

**SUZAN CAROLINE SUCH RIBEIRO**

**DESEMPENHO DE CRIANÇAS COM E SEM  
DIAGNÓSTICO DE TRANSTORNO DE DÉFICIT DE  
ATENÇÃO E HIPERATIVIDADE (TDAH) EM UMA  
BATERIA NEUROPSICOLÓGICA  
COMPUTADORIZADA**

Dissertação apresentada ao Programa de Pós-graduação em Neurociências, do Instituto de Ciências Biológicas da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de mestre.

Orientação: Prof. Dr. Arthur Melo e Kummer

**Belo Horizonte  
2013**

043 Ribeiro, Suzan Caroline Such.

Desempenho de crianças com e sem diagnóstico de Transtorno de Déficit de Atenção e Hiperatividade (TDAH) em uma bateria neuropsicológica computadorizada [manuscrito] / Suzan Caroline Such Ribeiro. – 2013.

177 f.; il. ; 29,5 cm.

Orientação: Prof. Dr. Arthur Melo e Kummer.

Dissertação (mestrado) – Universidade Federal de Minas Gerais, Instituto de Ciências Biológicas.

1. Neurociências - Teses. 2. Transtorno do Déficit de Atenção com Hiperatividade. 3. Função Executiva. 4. Testes neuropsicológicos - Teses. I. Kummer, Arthur Melo e. II. Universidade Federal de Minas Gerais. Instituto de Ciências Biológicas. III. Título.

CDU: 612.8



UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM NEUROCIÊNCIAS

UFMG

## FOLHA DE APROVAÇÃO

**Estudo do desempenho de crianças com e sem diagnóstico de Transtorno de Déficit de Atenção e Hiperatividade (TDAH) em uma bateria neuropsicológica computadorizada**

**SUZAN CAROLINE SUCH RIBEIRO**

Dissertação submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em NEUROCIÊNCIAS, como requisito para obtenção do grau de Mestre em NEUROCIÊNCIAS, área de concentração NEUROCIÊNCIAS CLÍNICAS.

Aprovada em 06 de agosto de 2013, pela banca constituída pelos membros:

Prof(a). Arthur Melo e Kummer - Orientador  
UFMG

Prof(a). José Ferreira Belizário Filho  
UFMG

Prof(a). Guilherme Nogueira Mendes de Oliveira  
UFMG

Belo Horizonte, 6 de agosto de 2013.

*Ao companheiro, amigo e tratador  
Fernando.*

## AGRADECIMENTOS

Muitas vezes a parte que mais ansiamos em escrever em um trabalho árduo são os agradecimentos. Este eu não ansiei. Este eu posterguei pois em muitos momentos pensei que pularia esta parte, ora por comodismo, ora por realmente não saber o que ou como escrever. Contudo, deixar de agradecer seria uma imensa ofensa à todos que fizeram parte desses 2 anos de caminhada. É um longo período compartilhado, suportado e alavancado por muitas pessoas que merecem algumas linhas.

Inicialmente gostaria de agradecer ao meu orientador de longa data, Arthur, pela dedicação, paciência, estímulo, apoio (acima de tudo apoio) e por, apesar de tudo, não ter desistido de mim. Espero ainda continuar aprendendo muito contigo. Obrigada é uma palavra pequena, mas é a que dispomos para expressar o agradecimento seja ele do tamanho que for. Este é grande!

Aos colegas de trabalho da Prefeitura de Nova Lima, que pacientemente ouviram minhas lamúrias, incentivaram, cobriram e entenderam minha ausência necessária.

A todas as instituições que participam desta pesquisa, pela colaboração e compreensão em todo o processo. E, claro, aos colaboradores e parceiros de pesquisa, Daniel, Beatriz, Douglas, Fernando e Natália. Muito obrigada pelo aprendizado.

Aos amigos, de perto, de longe e inomeáveis, que mesmo quando eu achava que não seria mais possível, declaravam uma fé inabalável em meu potencial. Vocês são um alicerce em minha vida.

Ao Núcleo Artístico Floresta que em meio a turbilhões ainda tem sido um ponto de equilíbrio, colocando a dança em minha vida. E aos muitos livros que preencheram lacunas...

Aos membros da banca, Fabrício Moreira, Guilherme Nogueira e José Belisário, que se prontificaram a apreciar esta dissertação.

A minha família, distante (pais, sogros, cunhados e etc) ou próxima (Fer), sendo base de todos os processos. Domo Arigatô! E, acima de tudo, agradeço a Deus que, de fato, escreveu por linhas muito tortas nos últimos tempos, mas certamente Ele sabe o que faz. 😊

## RESUMO

**Introdução:** O Transtorno do Déficit de Atenção/Hiperatividade (TDAH) é o distúrbio neurocomportamental mais comum na infância. Estima-se que 5,29% das crianças em todo o mundo têm o diagnóstico de TDAH. Os principais sintomas do TDAH incluem comprometimento da atenção concentrada, impulsividade e hiperatividade, e várias crianças com o transtorno têm prejuízos no rendimento acadêmico e na relação com colegas, professores e pais. Para explicar o TDAH, modelo como o déficit das funções executivas (FEs) têm sido proposto. Embora nem todas as crianças com TDAH apresentem problemas nas FEs, testes que as avaliam são importantes para refinar o diagnóstico e auxiliar no prognóstico.

**Objetivos:** Comparar o desempenho de crianças com e sem TDAH em uma bateria neuropsicológica computadorizada.

**Métodos:** Foram convidados a participar da pesquisa crianças entre 7 a 13 anos de escolas de Belo Horizonte e Nova Lima, além de pacientes do Ambulatório de Déficit de Atenção do Hospital das Clínicas da Universidade Federal de Minas Gerais (AMBDA- HC UFMG) e clínicas particulares. Após consentimento do participante e responsável, os participantes foram submetidos aos testes computadorizados (Go-No-Go, Teste de Dígitos direto e inverso, Donkey, Teste de Stroop, Torre de Londres, Teste de Classificação de Cartas), ao teste de inteligência Raven (para pareamento dos sujeitos) e a um questionário sobre uso de computadores e vídeo game. Além disso, os pais preencheram o SNAP-IV (questionário de rastreio de sintomas de TDAH), SDQ (questionário de rastreio de transtornos comportamentais), questionário breve sobre hábitos de uso de computador e vídeo game de seus filhos e critério socioeconômico.

**Resultados:** Participaram da pesquisa 50 crianças, das quais 54% possuem diagnóstico de TDAH e 84% obtiveram desempenho na média ou acima da média no teste Raven. Na análise feita com o programa SPSS (v. 20.0), crianças com TDAH apresentaram um desempenho pior em testes que avaliam flexibilidade cognitiva (Teste de Classificação de Cartas (BCST)- erros não perseverativos:  $p=0,018$ ), planejamento (Torre de Londres:  $p=0,047$ ), controle inibitório (Teste de Classificação de Cartas (BCST)- erros:  $p=0,010$ ) e atenção (Teste de Classificação de Cartas (BCST)- respostas corretas:  $p=0,019$ ). Porém não se observou diferença significativa quanto à memória de trabalho e tomada de decisão. Crianças com mais sintomas de desatenção apresentaram desempenho pior no Stroop ( $p=0,057$ ,  $r=0,279$ ) e Torre de Londres (pontuação:  $p=0,038$ ,  $r= -0,295$ ; tentativas:  $p=0,051$ ,  $r=0,277$ ).

**Conclusão:** Crianças com TDAH apresentaram desempenho pior principalmente no testes que avaliam planejamento e flexibilidade cognitiva. A avaliação neuropsicológica computadorizada pode se constituir em um importante aliado no diagnóstico do TDAH e fornecer dados relevantes para o acompanhamento e tratamento.

**Palavras chaves:** TDAH, Testes Neuropsicológicos, Funções Executivas, Testes Computadorizados, Stroop, Torre de Londres, BCST, Teste de Dígitos, Donkey

## ABSTRACT

**Introduction:** The Attention Deficit/Hyperactivity Disorder (ADHD) is the most common neurobehavioral disorder in childhood. It is estimated that 5.29% of children worldwide are diagnosed with ADHD. The main symptoms of ADHD include impaired focused attention, hyperactivity and impulsivity, and many children with the disorder have impairments in academic performance and relationships with peers, teachers and parents. To explain ADHD, the deficit model of executive functions (EF) have been proposed. Although not all children with ADHD have problems in EFs, tests that evaluate them are important to refine the diagnosis and in prognosis.

**Objectives:** To compare the performance of children with and without ADHD on a computerized neuropsychological battery.

**Methods:** We invited children aged 7-13 years old to participate in this research from schools in Belo Horizonte and Nova Lima, and patients from AMBDA HC-UFMG and private clinics. After the consent of the participant and their parents, the participants were subjected to computerized testing (Go-No-Go, Digit Span forward and backward, Donkey, Stroop Test, Tower of London, Card Sorting Test), the Raven (for pairing of subjects) and one questionnaire about use of computers and video games. In addition, parents completed the SNAP-IV (screening questionnaire of ADHD symptoms), SDQ (questionnaire screening behavioral disorders), brief questionnaire about use of computer and video games from their children and socioeconomic criteria.

**Results:** 50 children, participated and 84% were performing at or above average on Raven and 54% have a diagnosis of ADHD. In the analysis performed using the SPSS (v. 20.0), children with ADHD showed poorer performance on tests that assess cognitive flexibility (Card Sorting Test (BCST)- no perseverative error:  $p=0,018$ ), planning (Tower of London:  $p=0,047$ ), inhibitory control (Card Sorting Test (BCST)- error:  $p=0,010$ ) and attention Card Sorting Test (BCST)- correct answer:  $p=0,019$ ). However there was no significant difference in working memory and decision making. Children with more symptoms of inattention showed worse performance in the Stroop ( $p=0,057$ ,  $r=0,279$ ) and Tower of London (score:  $p=0,038$ ,  $r= -0,295$ ; tries:  $p=0,051$ ,  $r=0,277$ ).

**Conclusion:** Children with ADHD showed worse performance mainly on tests that assess planning and cognitive flexibility. The computerized neuropsychological assessment can be an important ally in the diagnosis of ADHD and provide relevant data for monitoring and treatment.

**Keywords:** ADHD, Neuropsychological Tests, Executive Functions, Computerized tests, Stroop, Tower of London, BCST, Digits Test, Donkey

## LISTA DE FIGURAS

<b>Figura 1-</b> Tela inicial do programa PEBL .....	29
<b>Figura 2-</b> Programa utilizado para alteração de áudio.....	30
<b>Figura 3-</b> Telas do Stroop Victoria.....	31
<b>Figura 4-</b> Exemplos de estímulos do Stroop Numérico. ....	31
<b>Figura 5-</b> Cartas-chave BCST. ....	33
<b>Figura 6-</b> Figuras da biblioteca do teste BCST. ....	34
<b>Figura 7-</b> Tela da instrução do teste BCST. ....	34
<b>Figura 8-</b> Telas do BCST.....	35
<b>Figura 9-</b> Instrução ToL.....	36
<b>Figura 10-</b> Tela inicial do problema de demonstração da ToL. ....	37
<b>Figura 11-</b> Tela da ToL.....	37
<b>Figura 12-</b> Tela de problema da ToL, mostrando a barra de movimentos. ....	38
<b>Figura 13-</b> Teste Hungry Donkey.....	39
<b>Figura 14-</b> Tela de instrução do Donkey. ....	40
<b>Figura 15-</b> Telas do teste Donkey.....	40
<b>Figura 16-</b> Teclado virtual utilizado para respostas do Teste de Dígitos. ....	42
<b>Figura 17-</b> Testes CPT e Go-no-go. ....	43

## LISTA DE TABELAS

<b>Tabela 1-</b> Notação para classificação e percentil do Raven .....	45
<b>Tabela 2-</b> Características dos Participantes. ....	50
<b>Tabela 3-</b> Comparações entre os grupos com e sem TDAH quanto a gênero, idade e critério socioeconômico. ....	53
<b>Tabela 4-</b> Comparações entre os grupos com e sem TDAH quanto a Idade, Critério Sócioeconômico, Raven, SNAP pontuação desatenção e hiperatividade, SDQ. ....	53
<b>Tabela 5-</b> Comparações significativas entre Diagnóstico SNAP-IV e Idade, Teste de Donkey, BCST, ToL. ....	54
<b>Tabela 6-</b> Correlações com a Pontuação do SNAP-IV nas subescalas de Desatenção e Hiperatividade .....	55
<b>Tabela 7-</b> Correlações com a Pontuação do SDQ nas subescalas de Hiperatividade, Conduta e Total.....	55

## LISTA DE GRÁFICOS

<b>Gráfico 1-</b> Distribuição dos participantes por uso de medicamento.....	50
<b>Gráfico 2-</b> Distribuição dos participantes por Local.....	51
<b>Gráfico 3-</b> Distribuição dos participantes por Idade e Critério Socioeconômico.....	52
<b>Gráfico 4 -</b> Desempenho no teste Raven. ....	52

## LISTA DE ABREVIATURAS E SIGLAS

**AAP:** American Academy of Pediatrics

**AMBDA/ HC-UFGM ou AMBDA:** Ambulatório de Déficit de Atenção do Hospital das Clínicas da Universidade Federal de Minas Gerais

**BCST:** Teste de Classificação de Cartas de Berg

**CPT:** Continuous Performance Task

**DFE:** Déficit de Funções Executivas

**Donkey:** Hungry Donkey

**DSM-IV:** Manual Diagnóstico e Estatístico de Transtornos Mentais 4ª. Edição

**DSM-5:** Manual Diagnóstico e Estatístico de Transtornos Mentais 5ª. Edição.

**IGT:** Iowa Gambling Task

**PEBL:** The Psychology Experiment Building Language

**Raven:** Teste das Matrizes Coloridas de Raven

**SDQ:** Strengths and Difficulties Questionnaire

**SNAP-IV:** Swanson, Nolan e Pelham-IV

**SPSS:** Statistical Package for the Social Sciences

**Stroop:** Teste de Stroop

**TAVIS:** Teste de Atenção Visual

**TDAH:** Transtorno de Déficit de Atenção e Hiperatividade

**TEACO:** Teste de Atenção Concentrada

**TEADI:** Teste de Atenção Dividida

**TEALT:** Teste de Atenção Alternada

**ToL:** Torre de Londres

**UFGM:** Universidade Federal de Minas Gerais

**WCST:** Teste de Classificação de Cartas de Wisconsin

**WCST-64:** Teste de Classificação de Cartas de Wisconsin, versão reduzida

**WISC:** Escala Wechsler de Inteligência para Crianças

## Sumário

Introdução.....	15
1 INTRODUÇÃO.....	14
1.1 Fundamentação teórica .....	14
1.1.1 Transtorno de Déficit de Atenção e Hiperatividade.....	14
1.1.2 Funções Executivas .....	15
1.2 Justificativa .....	21
1.3 Hipótese .....	21
Objetivos.....	22
2 OBJETIVOS.....	23
2.1 Objetivo Primário .....	23
2.2 Objetivo Secundário .....	23
.....	24
Métodos .....	24
3 MÉTODOS.....	25
3.1 Delineamento .....	25
3.2 Participantes.....	25
3.3 Materiais .....	27
3.3.1 Configuração dos testes computadorizados .....	28
3.3.2 Teste de Stroop.....	30
3.3.3 Teste de Classificação de Cartas de Berg.....	32
3.3.4 Torre de Londres (ToL).....	36
3.3.5 Hungry Donkey .....	38
3.3.6 Teste de Dígitos (Dígitos) .....	41
3.3.7 Outros testes .....	43
3.3.8 Teste das Matrizes Progressivas de Raven.....	44
3.3.9 Questionários utilizados .....	45
3.4 Procedimentos.....	46
3.5 Aspectos éticos .....	47
3.6 Análise estatística .....	48
Resultados.....	49
4 RESULTADOS .....	50
4.1 Características dos Participantes.....	50
4.2 Comparações entre grupos com e sem o Diagnóstico de TDAH.....	53
4.3 Correlações com Pontuação do SNAP-IV e SDQ .....	54

Discussão .....	57
5 DISCUSSÃO .....	58
Conclusão .....	64
6 CONCLUSÃO.....	65
Referências .....	67
Referências bibliográficas .....	68
Apêndices .....	71
APÊNDICE A .....	72
APÊNDICE B.....	74
Anexos.....	75
ANEXO A-Programação Teste de Dígitos.....	76
ANEXO B- Programação BCST .....	85
ANEXO C- Programação Torre de Londres .....	97
ANEXO D- Programação Donkey .....	122
ANEXO E- Programação Teste de Dígitos .....	129
ANEXO F- Protocolo .....	147
ANEXO G- Convite aos Pais .....	153
ANEXO H- Termo de Consentimento .....	154
ANEXO I- Principais Tabelas de Análises.....	155

# Introdução

# 1 INTRODUÇÃO

## 1.1 Fundamentação teórica

### 1.1.1 Transtorno de Déficit de Atenção e Hiperatividade

Um dos transtornos neurocomportamentais com maior prevalência na infância é o Déficit de Atenção/Hiperatividade (TDAH), podendo afetar tanto o aprendizado acadêmico quanto as interações sociais. De acordo com o manual para diagnóstico e tratamento do TDAH, elaborado pela American Academy of Pediatrics (AAP, 2011), crianças e adolescentes com idade entre 4 a 18 anos devem ser avaliadas com a suspeita diagnóstica de TDAH caso se observem problemas comportamentais ou acadêmicos e sintomas de desatenção, hiperatividade e impulsividade.

Os critérios do DSM-IV<sup>1</sup> (Manual Diagnóstico e Estatístico de Transtornos Mentais) são ainda os mais aceitos e utilizados para a confirmação diagnóstica (APÊNDICE A). Seguindo esses critérios, estima-se que 8% das crianças e adolescentes têm o diagnóstico de TDAH (AAP, 2011). Polanczyk e Rohde (2007), em pesquisa de revisão, identificaram uma prevalência de 5,29% de diagnósticos em crianças, sendo a maioria meninos. Em um estudo com crianças brasileiras que cursavam entre a 1ª e a 4ª séries, Fontana *et al.* (2007) encontraram uma prevalência do TDAH de 13%. Já em pesquisa realizada por Anselmi *et al.* (2009), também no Brasil, na avaliação de crianças com 11-12 anos 4,1 % apresentaram sintomas significativos de TDAH.

Segundo Malloy-Diniz *et al.* (2008a), o diagnóstico do TDAH é essencialmente clínico e envolve a investigação de sintomas como desatenção, impulsividade e hiperatividade. Conforme a predominância desses sintomas, pode-se diferenciar o TDAH em três subtipos: desatento, hiperativo-impulsivo e combinado. Esta classificação em subtipo, porém, foi reavaliado para o DSM-5<sup>2</sup>. Neste novo Manual, esses itens passam a ser chamados de apresentações, visando um entendimento de característica desenvolvimental e não permanente (MATTOS, 2013).

---

<sup>1</sup> DSM IV: Diagnostic and Statistical Manual of Mental Disorders ou Manual Diagnóstico e Estatístico de Transtornos Mentais.

<sup>2</sup> DSM 5: Novo Manual Diagnóstico que, apesar de já estar publicado, ainda não é o mais utilizado.

A presença de comorbidades<sup>3</sup> é significativamente elevada, tal como o transtorno de desafio e oposição (35-65%), o transtorno de conduta (20-50%) e a depressão (15-75%). O diagnóstico adequado e precoce de tais comorbidades favorece a escolha do tratamento apropriado (MALLOY-DINIZ *et al.*, 2008a).

Como ainda não há um único exame para o diagnóstico de TDAH, faz-se necessária a coleta de dados de fontes diversas, considerando que o portador tende a ter uma auto-observação imprecisa. Por isso, as observações parentais, dos professores e responsáveis são de grande relevância (AAP, 2011; MALLOY-DINIZ *et al.*, 2008a). Além disso, mais fontes reduzem a influência das diferenças culturais em aspectos da avaliação (AAP, 2011). Deve atentar-se, ainda, para que os métodos selecionados para a avaliação sejam preferencialmente com parâmetros baseado nas características culturais da população alvo. Outro elemento utilizado para refinar o diagnóstico é a avaliação neuropsicológica, caracterizando as dificuldades e potencialidades do indivíduo. Os principais aspectos dessa avaliação, com estudos apontando diferenças entre portadores de TDAH e população geral, são aspectos da inteligência (alguns subtestes da Escala Wechsler), vivência temporal, atenção e funções executivas (MALLOY-DINIZ *et al.*, 2008a).

### 1.1.2 Funções Executivas

As funções executivas são essenciais para o gerenciamento do comportamento humano. Por isso, são alvo da avaliação neuropsicológica de indivíduos normais ou com alterações cognitivas e/ou comportamentais de causas diversas, como o TDAH (SOUZA *et al.*, 2001). Tal gerenciamento ocorre através de habilidades que, de forma integrada, permitem que o indivíduo direcione seus comportamentos a objetivos, analise a adequação e eficácia desses comportamentos, modifique estratégias quando necessário e, assim, solucione problemas diversos. As funções executivas incluem planejamento, controle inibitório, tomada de decisões,

---

<sup>33</sup>Comorbidade: presença de mais de um transtorno em uma mesma pessoa. Exemplo: TDAH e depressão.

flexibilidade cognitiva, memória operacional, atenção, categorização e fluência (MALLOY-DINIZ *et al.*, 2008b). Para maior compreensão, segue descrição de alguns desses processos.

A capacidade de planejar envolve antecipação de eventos e consequências, assim como o monitoramento para o alcance de metas (KRIKORIAN, 1994). Para tanto, o indivíduo deve considerar a hierarquização de passos e a utilização de instrumentos necessários (MALLOY-DINIZ, 2008b). A resolução de problemas, por exemplo, tem como um componente central a capacidade de planejar, no sentido de antecipação de eventos e consequências, bem como o monitoramento de alcance de metas. Assim, em tarefas de resolução de problemas que exigem soluções novas ou rotineiras pode-se observar o planejamento mais facilmente (KRIKORIAN, 1994). A Torre de Londres é um dos testes mais utilizados para avaliação de planejamento (SOUZA *et al.*, 2001).

O controle inibitório é definido como a capacidade do indivíduo inibir respostas diante de estímulos distratores que interrompam o curso eficaz de sua ação ou frente a respostas para as quais o indivíduo apresenta uma forte tendência. Ou seja, o indivíduo deve ser capaz de inibir uma resposta prepotente. Dificuldades observadas neste processo são geralmente relacionadas com impulsividade. Para avaliação do controle inibitório são utilizados testes como CPT (Continuous Performance Task), Go-No-Go e Stroop (MALLOY-DINIZ, 2008b).

Segundo Crone e Molen (2007), muitas decisões da vida diária são feitas frente a circunstâncias incertas e necessitam de avaliação sobre possíveis consequências futuras. Para tanto, utiliza-se o processo de tomada de decisões. Um bom desempenho neste processo envolve análise de experiência, resultados esperados e sentimentos. Crianças e adolescentes passam por uma transformação nos padrões de tomada de decisão ou na capacidade de antecipar resultados futuros das escolhas. Isto pode ser observado no padrão de resposta de risco de crianças novas, ao passo que crianças mais velhas fazem escolhas mais conservadoras. Para avaliação deste aspecto pode-se utilizar o Iowa Gambling Task ou a versão infantil Hungry Donkey Task (CRONE; MOLEN, 2007).

O processo que envolve a capacidade de alternar o curso dos comportamentos (ações, comportamentos) de acordo com as exigências do ambiente é denominado flexibilidade cognitiva (MALLOY-DINIZ, 2008b). Assim, testes que avaliam esse processo promovem a alternância entre categorias cognitivas, como o Teste de Trilhas (parte B) e Teste de Classificação de Cartas (SOUZA *et al.*, 2001).

A memória operacional, segundo Malloy-Diniz (2008a, p.199)

“é um sistema temporário de armazenamento de informações que permite a sua monitoração e o seu manejo. Esse componente das funções executivas é responsável por manter ativado um delimitado volume de informações durante um determinado período de tempo, fornecendo inclusive base para outros processos cognitivos”

Para avaliação deste processo um dos testes utilizados é o Teste de Dígitos (HESTER; KINSELLA; ONG,2004).

A atenção é o processo que envolve habilidades de “concentração, esforço mental, manutenção do estado de observação ou alerta e a capacidade de focalizar, ignorar estímulos distratores ou irrelevantes e modificar o alvo da atenção quando necessário” (MALLOY-DINIZ *et al.*, 2008b). Em geral, a atenção é subdividida em: atenção focalizada, atenção sustentada, atenção seletiva e atenção alternada. Por ser um termo amplo e muito estudado, há inúmeros teste que avaliam atenção, como Stroop, CPT (Continuous Performance Test), TAVIS (Teste de Atenção Visual), TEADI (Teste de Atenção Dividida), TEALT (Teste de Atenção Alternada), TEACO (Teste de Atenção Concentrada), dentre outros (MALLOY-DINIZ *et al.*, 2008b).

A categorização, segundo Malloy-Diniz *et al.* (2008b), é um processo através do qual agrupam-se elementos com propriedade similares. Esta habilidade está envolvida na formação de conceitos, raciocínio dedutivo, indutivo e abstração (MALLOY-DINIZ *et al.*, 2008). O Teste de Classificação de Cartas de Wisconsin (WCST<sup>4</sup>), fluência verbal semântica, assim como os subtestes de semelhanças e compreensão da Escala Wechsler de Inteligência para Crianças

---

<sup>4</sup>WCST é a sigla do nome original deste teste: Wisconsin Card Sorting Test. A versão manual deste teste é validada no Brasil para crianças, adolescentes e idosos.

(WISC) são exemplos de testes que avaliam este processo. Pesquisas têm mostrado um baixo desempenho no WCST em sujeitos que apresentam diagnósticos como esquizofrenia, TDAH, epilepsia, autismo e depressão quando comparados ao resultado de grupos controles (TRENTINI *et al.*, 2006).

As funções executivas, assim, são constituídas pela integração desses processos, permitindo ao indivíduo coordenar seus comportamentos. Um indivíduo com déficit em alguns dos processos envolvidos (planejamento, flexibilidade cognitiva, atenção, etc.) apresentaria dificuldades na conclusão bem-sucedida da tarefa. Essa dificuldade nesses processos é denominada por alguns autores como disfunção executiva e pode ser decorrente de comprometimentos cerebrais adquiridos ou resultantes do desenvolvimento anormal do sistema nervoso (MALLOY-DINIZ, 2008b).

Biederman *et al.* sugerem que quando o indivíduo apresenta prejuízo em pelo menos duas medidas de funções executivas é considerado como Déficit de Função Executiva- DFE (BIEDERMAN *et al.*, 2004). Segundo Lambek *et al.* (2010), poucos estudos tentaram operacionalizar DFE e examinar seus correlatos em crianças e adolescentes diagnosticados com TDAH. Um desses estudos (BIEDERMAN *et al.*, 2004), realizou comparação ente 259 crianças com e 222 crianças sem diagnóstico de TDAH. Dentre os achados, os autores encontraram que 33% das crianças com TDAH também tiveram classificação de DFE, ao passo que dentre as crianças sem TDAH 12% tiveram classificação de DFE. Além disso, o diagnóstico de TDAH com DFE pode ser um fator de risco para retenção escolar e/ou prejuízos acadêmicos. Já no estudo de Lambek *et al.* (2010), crianças com DFE e TDAH obtiveram um QI menor do que o grupo controle. Para Knouse, Barkley e Murphy (2013), déficits em Funções Executivas estão associados ao TDAH em crianças, principalmente no que se refere aos processos de controle inibitório e memória de trabalho.

Em relação aos aspectos neuroanatômicos, apesar de muitos autores associarem as funções executivas ao córtex pré-frontal, é mais preciso “classificar tais funções como resultantes da atividade distribuída de diferentes circuitos neurais” (MALLOY-DINIZ *et al.*, p.190, 2008b). O circuito dorsolateral está relacionado a

“processos cognitivos de estabelecimento de metas, planejamento, solução de problemas, fluência, categorização, memória operacional, monitoração da aprendizagem e atenção, flexibilidade cognitiva, capacidade de abstração, auto-regulação, julgamento, tomada de decisões, foco e sustentação da atenção, e tomada de decisão” (MALLOY-DINIZ, p.191, 2008b).

Estudos de caso mostraram que um comprometimento deste circuito pode ocasionar dificuldade em recuperar material aprendido livremente (MALLOY-DINIZ *et al.*, 2008b).

No que se refere ao comportamento social, como empatia, cumprimento de regras sociais, controle inibitório e automonitoração, parece estar mais associado ao circuito lateral orbitofrontal. Portanto, comprometimento neste circuito pode gerar, por exemplo, comportamentos de risco (MALLOY-DINIZ *et al.*, 2008b).

Outro circuito que pode estar associado às funções executivas é o circuito do cíngulo anterior, que está relacionado à motivação, monitoração de comportamentos, controle executivo da atenção, seleção e controle de respostas. Dificuldades em controlar a atenção e em identificar e corrigir erros são comuns em pessoas que apresentam comprometimento neste circuito (MALLOY-DINIZ *et al.*, 2008b).

As funções executivas são estudadas também em seu aspecto neuroquímico. Segundo Malloy-Diniz *et al.* (2008b), os sistemas de neurotransmissão que se associam ao desempenho das funções executivas são o dopaminérgico, serotoninérgico e noradrenérgico. Falhas no sistema dopaminérgico podem ocasionar dificuldades de memória operacional, atenção, controle inibitório, planejamento, flexibilidade cognitiva e tomada de decisão. Já a diminuição da atividade serotoninérgica está associada ao aumento de impulsividade e agressividade. O sistema noradrenérgico, por sua vez, envolve atenção, memória e aprendizagem e motivação. O estudo das alterações nos sistemas de neurotransmissão é essencial para a manipulação e uso de medicamentos mais adequados para a melhora desses processos mencionados (MALLOY-DINIZ *et al.*, 2008b).

Para avaliação neuropsicológica das funções executivas, há uma ampla variedade de testes, conforme mencionado acima. O presente estudo selecionou versões computadorizadas dos testes Teste de Dígitos (Dígitos), Donkey, Teste de Classificação de Cartas de Berg<sup>5</sup> (BCST), Teste de Stroop (Stroop) e Torre de Londres (ToL) para integrar e compor uma bateria de avaliação de funções executivas.

A aplicação da versão computadorizada dos testes tem como benefícios a precisão na apresentação dos estímulos, *feedback*<sup>6</sup>, medidas temporais e registro das respostas, eliminando assim possíveis erros na obtenção dos dados e pontuação. Bauer *et al.* (2012), apontam outras vantagens como a avaliação de muitas pessoas rapidamente, redução no custo de aplicação e pontuação, facilidade para exportar dados para pesquisa, habilidade para integrar e interpretar dados automaticamente. Essas características fornecem maior rigor na aplicação dos testes, gerando dados para uma avaliação neuropsicológica mais confiável (CAPOVILLA, 2006).

Para Tien *et al.* (1996), os testes computadorizados, apesar de gerarem dados mais confiáveis, não são substitutos à aplicação manual. Em vez disso, o computador é um parceiro na aplicação do teste, que permite que o examinador fique livre para dar o suporte adequado ao sujeito em teste, encorajando-o e observando os aspectos qualitativos de seu desempenho.

Em pesquisa realizada anteriormente pelos presentes autores, 22 crianças (idade média= 9 anos) responderam a versões manual e computadorizada de três testes: WCST reduzido (WCST-64), Stroop e ToL. Ao final das avaliações, as crianças declaravam qual das versões preferiam. Apesar de terem obtido desempenho melhor nas versões manuais, a maioria gostou mais das versões computadorizadas dos testes WCST-64 e Stroop, mas preferiram a versão manual da ToL. Questionou-se ainda quanto à dificuldade das versões, podendo optar entre fácil, moderado ou difícil para cada teste. Assim como na avaliação sobre preferência, mesmo tendo desempenho melhor nas versões manuais, os participantes acharam mais difíceis as versões

---

<sup>5</sup> Tradução nossa. Termo original: Berg's Card Sorting Task.

<sup>6</sup> *Feedback*: informação dada após o comportamento do indivíduo, indicando como foi seu desempenho. No caso dos testes utilizados nesta pesquisa, o *feedback* é informar se a resposta do sujeito está correta ou errada.

manuais exceto para a ToL, cuja versão computadorizada foi considerada mais difícil. Estes dados demonstram uma tendência à preferência deste público por atividades tecnológicas, tornando mais prazerosa a execução dos testes.

## 1.2 Justificativa

Os testes Teste de Dígitos, Stroop, ToL, BCST e Donkey integrados podem compor uma bateria importante para avaliação das funções executivas, auxiliando no diagnóstico e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos. Acreditamos que o estudo de normatização desses instrumentos é significativo para ampliar os instrumentos de avaliação neuropsicológica na população brasileira. Para tanto, a pesquisa realizada visa cumprir com uma das etapas deste processo que é a comparação entre diferentes populações.

Além disso, esta pesquisa almeja contribuir com a divulgação institucional, tendo como metas o fortalecimento da UFMG como centro de referência em pesquisas sobre cognição e na normatização e validação de testes neuropsicológicos.

## 1.3 Hipótese

- Crianças com sintomas de TDAH apresentam desempenho inferior comparado com crianças sem sintomas em testes de avaliação neuropsicológica.

# Objetivos

## 2 OBJETIVOS

### 2.1 Objetivo Primário

- Comparar o desempenho de crianças com e sem TDAH na bateria computadorizada de testes neuropsicológicos.

### 2.2 Objetivo Secundário

- Verificar se há correlação entre o escore nas tarefas computadorizadas (bateria PEBL) e a pontuação nas escalas SDQ (Strengths and Difficulties Questionnaire<sup>7</sup>) e SNAP-IV (Swanson, Nolan e Pelham-IV) de todos os participantes;

---

<sup>7</sup>Strengths and Difficulties Questionnaire: Questionário de Facilidades e Dificuldades [tradução nossa]

# Métodos

## 3 MÉTODOS

### 3.1 Delineamento

Trata-se de pesquisa comparativa realizada no Ambulatório de Déficit de Atenção do Hospital das Clínicas da Universidade Federal de Minas Gerais (AMBDA/ HC-UFMG), além de Escolas e Clínicas das cidades de Belo Horizonte-MG e Nova Lima-MG, com os participantes que preencheram os critérios de inclusão do estudo. Nestes locais, foram explicados todos os procedimentos da pesquisa, esclarecidas eventuais dúvidas e obtido o consentimento dos participantes. Os participantes foram examinados nas instituições envolvidas na pesquisa, o que facilitou sua participação.

Foram examinados 50 indivíduos, número suficiente para o cálculo de correlação. A pesquisa não dispôs de financiamento e não acarretou gastos significativos.

### 3.2 Participantes

Para este estudo, foram avaliadas 50 crianças, com idade entre 7 a 13 anos. Foram incluídos participantes de ambos os gêneros, independentemente de cor, nível socioeconômico ou educacional. Para ser incluído no estudo, os pais e/ou responsáveis e o participante assinaram o termo de consentimento livre e esclarecido.

Os critérios de exclusão do estudo estabelecidos foram: crianças portadoras de retardo mental, com déficits motores ou sensoriais que dificultassem a realização da testagem, ou que tinham usado substâncias psicoativas (álcool e drogas ilícitas) recentemente. Nenhuma criança necessitou ser excluída.

No AMBDA/HC-UFMG, os pacientes com diagnóstico de TDAH foram convidados a participar da pesquisa no dia de sua consulta de rotina. O contato era realizado com os pais e/ou responsável e a confirmação diagnóstica verificada com o médico responsável e prontuário. Foram avaliadas 11 crianças deste local.

No Colégio Tiradentes da Polícia Militar de Minas Gerais/Santa Tereza, Belo Horizonte-MG, todos os alunos matriculados no 3º ano do período vespertino receberam convite para participar da pesquisa. Desta forma, foram convidados 140 crianças. Foram recebidas 80 autorizações para participação da pesquisa. Até o presente momento, foram avaliadas 16 crianças desta instituição.

Na cidade de Nova Lima-MG, a Secretaria Municipal de Educação autorizou a realização da pesquisa em todas as escolas municipais. Para esta etapa da pesquisa, foram selecionadas duas escolas. Na Escola Vicente Estevão dos Santos, foram convidados aleatoriamente 20 alunos matriculados no 3º ano período vespertino. Contudo, durante as avaliações, foi observado que a escola não cumpriu com o critério de aleatoriedade, selecionando parte dos alunos. Até o presente momento, avaliaram-se 3 crianças desta instituição.

Na Escola José Francisco da Silva, todos os alunos matriculados no 3º ano período matutino foram convidados, totalizando 22 alunos. Apesar de serem convidados todos os alunos, observou-se durante as avaliações que as crianças deste grupo são da turma com maior dificuldade da escola. Dentre as 22 crianças, 16 já foram avaliadas até então. Quando houve recusa na participação da pesquisa, outra criança matriculada na mesma série foi convidada para completar o número de avaliações a serem realizadas na instituição.

Foram avaliadas ainda outras crianças que preencheram critérios para participar da pesquisa e tomaram conhecimento da mesma por convite na clínica privada dos pesquisadores em que realizavam tratamento. Totalizaram 4 crianças até o momento.

### 3.3 Materiais

Os participantes foram avaliados em uma sala apropriada à aplicação dos testes. Foram administrados uma bateria de testes computadorizados composta pelos testes: BCST, Go-No-Go<sup>8</sup>, Hungry Donkey (Donkey), Torre de Londres (ToL), Teste de Stroop Numérico (Stroop), Teste de Dígitos (Dígitos)- versões direta e inversa. Juntamente com a aplicação dos testes computadorizados, foi aplicado o Teste das Matrizes Progressivas de Raven (Raven) para pareamento dos participantes, além de uma escala composta por sintomas de ansiedade (conforto, suor, temperatura) e motivação para os testes. As crianças também preencheram questionário sobre uso de computador e vídeo game.

Na pesquisa foram utilizados notebooks, com monitor LCD 15” e sistema operacional Microsoft Windows 7 ou superior. Além disso, optou-se em utilizar mouse e teclados separados do notebook para padronizar a sensibilidade e minimizar danos aos notebooks.

O *software* utilizado na elaboração dos testes computadorizados é o PEBL- The Psychology Experiment Building Language (versão 0.13), que possui código aberto e é um *software* livre. Utilizando este programa, os pesquisadores e o próprio Shane T. Mueller (criador do software) adaptaram as versões computadorizadas de maneira similar às versões manuais utilizadas destes testes (BCST, ToL, Dígitos) ou de acordo com o recomendado pela literatura (Go-No-Go, Donkey, Stroop). A versão 0.13 do PEBL permite ainda a criação de uma cadeia de testes, ordenando previamente a sequência dos testes que serão aplicados, não necessitando iniciar individualmente cada teste. Esta cadeia pode ser salva, facilitando a execução da pesquisa, pois ao iniciá-la os testes executarão na sequência programada. Caso necessite repetir algum teste, é possível abrir apenas o teste desejado.

---

<sup>8</sup> Apesar de ter sido utilizado o teste Go no-go durante as aplicações, o teste ficou invalidado devido a problema no registro das respostas.

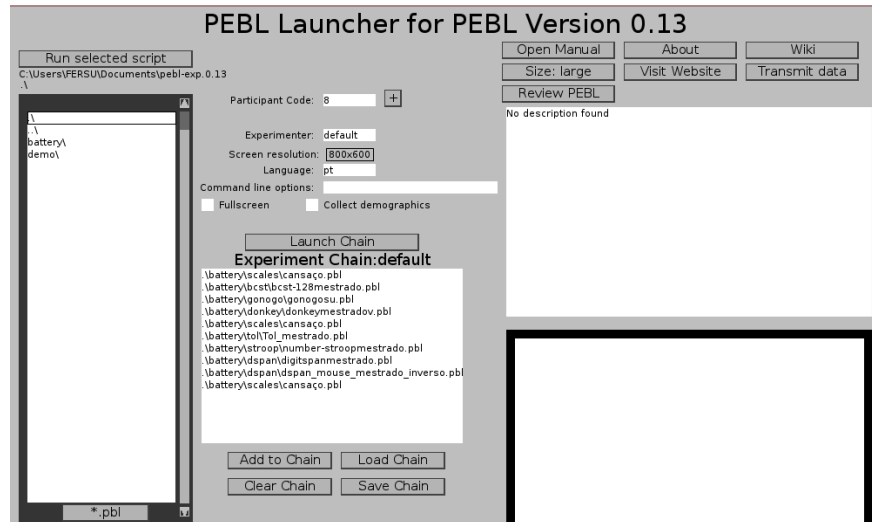
Na tela de início é possível selecionar a cadeia de testes que serão utilizados, bem como inserir o código do participante. Para garantir a identificação dos participantes, além de inserir o código na tela inicial do programa (número previamente selecionado), foi programado para que antes de cada teste iniciar fosse gerado uma nova tela para registro do participante (nome do participante). Assim, os resultados eram gerados tanto com o número quanto com o nome daquele participante.

Ao iniciar a cadeia de testes, os testes iniciam-se com o registro do participante, conforme mencionado, seguido pelas instruções dadas tanto de forma visual quanto auditiva na maioria dos testes. As versões foram elaboradas almejando que o avaliador interfira o mínimo possível. A seguir, segue descrição sobre as alterações de cada teste.

### 3.3.1 Configuração dos testes computadorizados

Para realizar as modificações no programa PEBL, utilizaram-se as bases fornecidos na bateria de teste, alterando-se via Bloco de notas e salvando em UTF-8 para manter os caracteres específicos da língua portuguesa tais como o uso de cedilha e acentos. O manual do programa fornece as principais instruções incluindo como instalar e comandos básicos para escrever ou alterar um programa, como por exemplo, orientação de como inserir áudio (MUELLER, 2011).

Nos estudos iniciais desta pesquisa, o programa PEBL estava na versão 0.11 e 0.12, podendo utilizar baterias de testes versões 0.6 e 0.7. No final de 2012, foi disponibilizada a versão 0.13 do programa (FIG. 1), que possui uma bateria de 63 testes embutida ao programa (sem necessidade de download a parte).

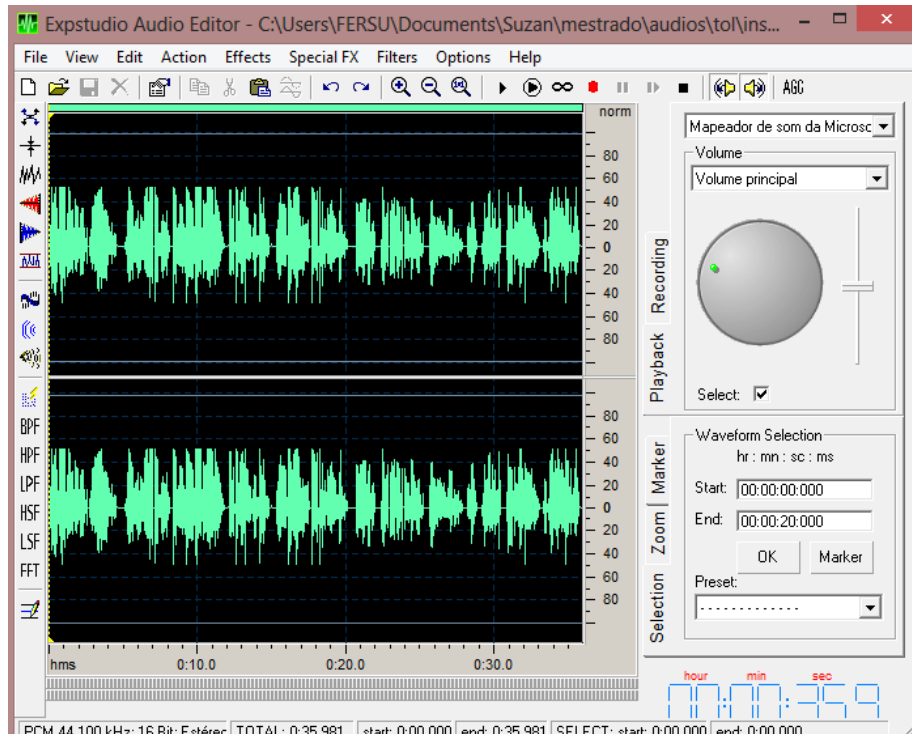


**Figura 1-** Tela inicial do programa PEBL

A versão 0.12 foi bastante explorada pelos pesquisadores. Ao ser liberada a versão 0.13, contudo, fizeram-se discretas adaptações e passou-se a utilizá-la. A seguir serão descritas as tentativas de alterações feitas em cada teste.

### *Gravação de áudio e instrução*

Os testes originalmente fornecem uma instrução escrita sobre como executar a atividade. Visando facilitar a compreensão, fornecer instruções mais claras e agilizar os *feedbacks*, optou-se em adicionar a instrução auditiva nos testes. Para tanto, são gravados áudios, correspondentes a instrução e *feedback* escrito, e inseridos na programação de cada teste (FIG. 2). Nem todos os testes, porém, permitem a inserção do áudio, por falha da programação base do PEBL. Este problema está sendo resolvido pelos criadores do programa.



**Figura 2-** Programa utilizado para alteração de áudio.

A imagem pertence ao áudio da instrução 1 do teste Torre de Londres.

### 3.3.2 Teste de Stroop

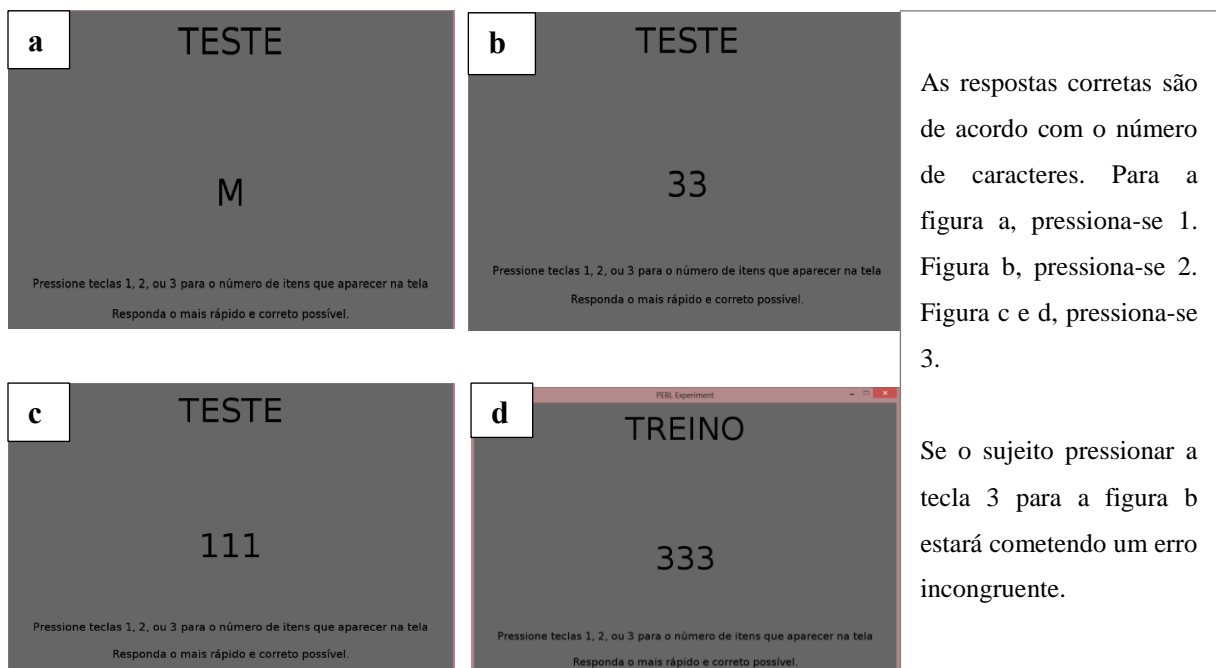
A aplicação manual do Teste de Stroop Victoria é composta por três cartões utilizando estímulos impressos em cores padrão: verde, vermelho, azul e amarelo. No primeiro cartão são apresentados pequenos retângulos, no segundo há palavras comuns coloridas e no último são apresentadas nomes de cores impressas em uma cor que nunca corresponde ao nome da cor (ex.: “vermelho” é impresso em azul). O sujeito deverá nomear as cores dos retângulos e das palavras o mais rápido possível, ignorando o conteúdo das palavras. Sendo assim, no terceiro cartão é onde poderá ser observado o efeito Stroop, pois são apresentados os nomes das cores (distrator), mas impressas com a cor que deverá ser nomeada (informação alvo). Durante a aplicação é registrado o tempo de resposta para cada cartão apresentado. O efeito de interferência é determinado calculando o tempo extra gasto para nomear as cores na última tarefa em comparação com o tempo gasto para nomear as cores na segunda tarefa (STRAUSS *et al.*, 2006).

Em uma primeira etapa da pesquisa, onde se comparou a aplicação manual e computadorizada deste teste, configurou-se o programa de forma similar a descrita acima (FIG. 3). Contudo, ao invés de nomear verbalmente as cores, o sujeito respondia pressionando as teclas 1, 2, 3 ou 4, conforme fosse solicitado.



**Figura 3-** Telas do Stroop Victoria.  
a) Tela parte I; b) Tela Parte II; c) Tela Parte III.

Após algumas tentativas de configuração para captura da resposta verbal terem falhado e considerando que esta pesquisa tem como objetivo testes que facilitem a aplicação e o registro, optou-se por manter a versão com resposta motora. Durante a fase piloto, observou-se ainda que principalmente as crianças com diagnóstico de TDAH não tinham domínio de leitura e por isso o teste poderia não estar avaliando o “efeito Stroop”. Desta forma, optou-se em utilizar a versão numérica (FIG. 4).



**Figura 4-** Exemplos de estímulos do Stroop Numérico.  
a) um caractere; b) dois caracteres; c) três caracteres; d) tela de treino.

A versão do Stroop numérico utilizada é a mesma que está anexa ao programa original do PEBL, alterando-se apenas para o português e acrescentando áudio nas instruções.

Este teste inicia-se com a escolha do tipo de resposta, se teclado *keyboard*<sup>9</sup>, *keypad*<sup>10</sup> ou microfone<sup>11</sup>. Para esta pesquisa, como se utilizaram teclados a parte, optou-se pelo *Keypad*. Em seguida, realiza-se o registro do nome. A instrução áudio/visual é mais extensa neste teste, Por ser um teste novo para os pesquisadores, ainda não se sabia qual seria a melhor formatação para a instrução. Em alguns momentos, observou-se necessidade de reler a instrução para as crianças.

Para a execução do teste, o sujeito é instruído a pressionar o mais rápido e preciso possível a tecla 1 toda vez que ver um caracter na tela, a tecla 2 toda vez que ver dois caracteres na tela e a tecla 3 quando surgirem três caracteres na tela. Os caracteres podem ser números ou letras. O efeito Stroop é investigado quando surge, por exemplo, dois números 1 na tela, gerando conflito entre significado e objetivo do teste (FIG. 4). O teste tem uma fase de treino, com 24 tentativas, e mais duas etapas com 90 tentativas cada. Caso o sujeito demore para emitir a resposta, surge na tela a mensagem “muito lento”. O programa registra total de erros, erros incongruentes, erros aleatórios, além de tempo de resposta e precisão. A programação deste teste está no ANEXO A.

### 3.3.3 Teste de Classificação de Cartas de Berg

A versão BCST selecionada neste estudo é similar a versão manual denominada Teste Wisconsin de Classificação de Cartas (WCST). O teste é realizado pela apresentação de 4 (quatro) cartas-chave (FIG. 5), compostas por um triângulo vermelho, duas estrelas verdes, três cruces amarelas e quatro círculos azuis, e duas pilhas idênticas de 64 cartas-resposta, gerando um total de 128 cartas-resposta. O sujeito tem como tarefas estabelecer categorias através de

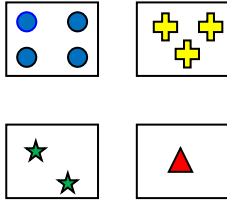
---

<sup>9</sup>*Keyboard*: números acima das letras.

<sup>10</sup>*Keypad*: números em teclado separado ao lado direito.

<sup>11</sup> Microfone: em fase de teste.

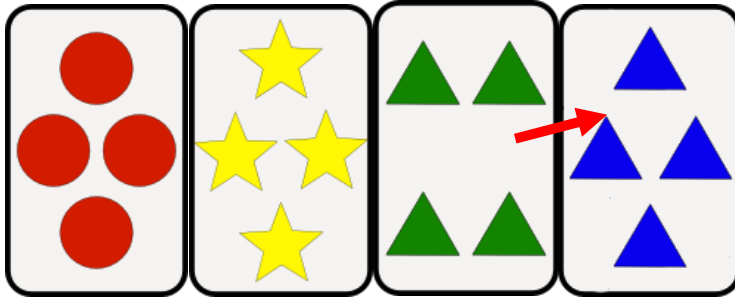
associações entre as cartas-resposta e as cartas-chave. As categorias são pré-determinadas e não são informadas ao sujeito, porém, o programa fornece *feedback* de certo ou errado a cada resposta emitida pelo sujeito.



**Figura 5-** Cartas-chave BCST.

Dentre as vantagens do teste BCST está a facilidade de ajuste do *software* para a versão desejada, podendo-se escolher diferentes momentos para encerrar o teste (ex.: após atingir 6 categorias ou após completar 32, 64 ou 128 cartas) ou o número de cartas a serem utilizadas. Entre as versões breves propostas, a versão de 64 cartas (apenas o primeiro *deck* de cartas) tem se mostrado a mais lógica e prática (GREVE, 2001), sendo superior a outras versões do teste e mantendo uma confiabilidade próxima à do teste completo (ROBINSON *et al.*, 1991). A versão reduzida de 64 cartas também foi desenvolvida pelos pesquisadores, porém como alguns autores afirmam que a versão completa apresenta um melhor entendimento para a faixa-etária desta pesquisa, optou-se em utilizar nesta etapa do estudo a versão com 128 cartas (SMITH-SEEMILLER; ARFFA; FRANZEN, 2001).

Outras alterações podem ser realizadas como, por exemplo, o teste original do PEBL apresentava uma diferença nas figuras com quatro números. Apenas as figuras com triângulos estavam disposta em forma de um quadrado, enquanto as demais estavam distribuídas em forma de losango (FIG. 6). Para evitar que esta diferença interferisse na escolha dos sujeitos, optou-se em alterar os triângulos para a mesma distribuição espacial das demais.



**Figura 6-** Figuras da biblioteca do teste BCST.

Todas as figuras com 4 triângulos foram alteradas para manter a mesma forma que as demais.

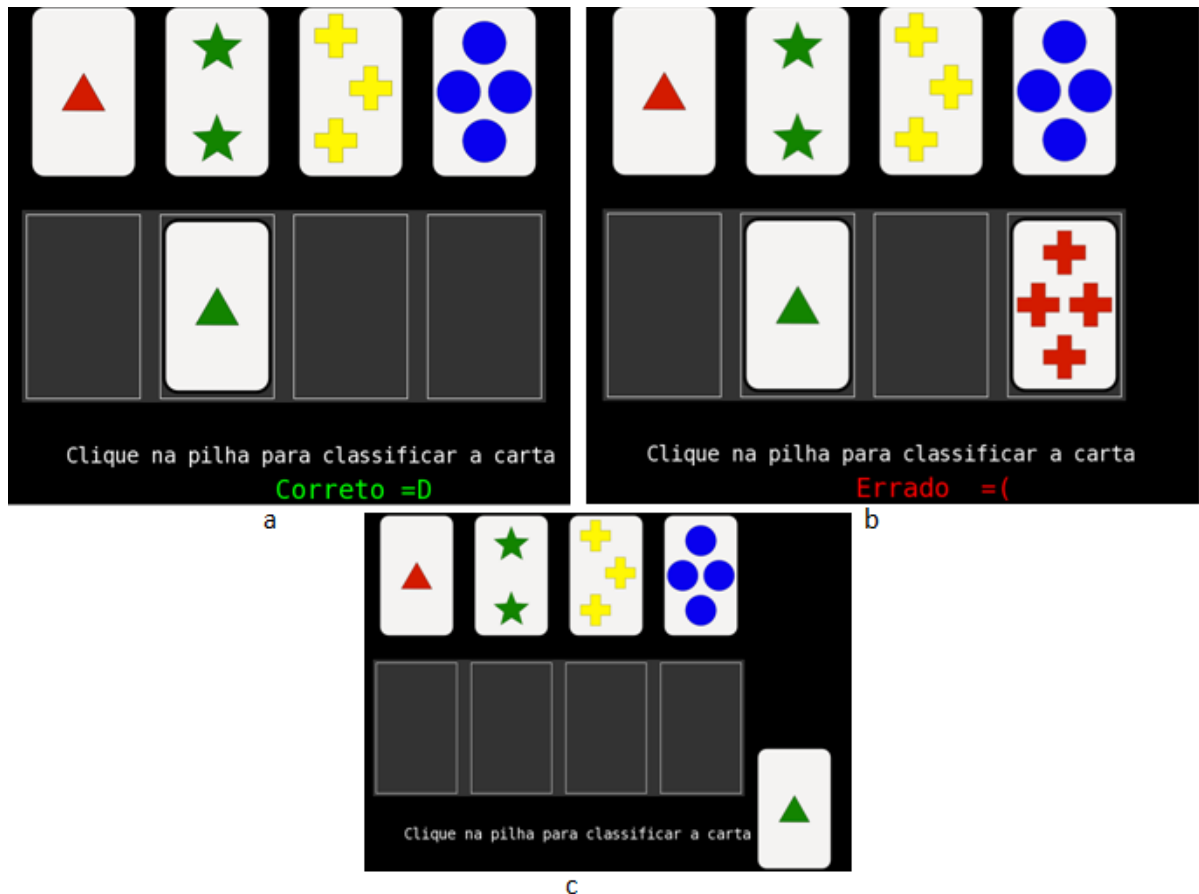
Conforme o desempenho do sujeito, os prejuízos podem ser identificados em diversos itens obtidos por este teste como total de erros, respostas perseverativas, erros perseverativos e o número de categorias completadas (APÊNDICE B).

Para realizar o teste, o sujeito utiliza o mouse. Inicialmente, registra-se o nome/sigla do sujeito. Logo após, o programa fornece instruções auditivas e visuais sobre o teste, divididas em três telas (FIG. 7).

Você está vendo 1 triângulo vermelho, 2 estrelas verdes, 3 cruces amarelas e 4 círculos azuis. A direita da tela, você verá uma série de cartas e precisa escolher a qual pilha elas pertencem. Clique no retângulo abaixo da pilha escolhida com o apontador do mouse. A carta nova aparecerá abaixo da pilha que escolheu. Clique com o mouse para continuar.

**Figura 7-** Tela da instrução do teste BCST.

Terminando as instruções, o sujeito iniciará o teste. As cartas-chave aparecerão no topo da tela, ao passo que a carta-resposta surgirá no canto inferior direito. Para realizar a resposta, o sujeito deverá clicar no retângulo abaixo da carta-chave com a qual deseja associar. Cada resposta emitida pelo sujeito é registrada pelo sistema. Os *feedbacks* das associações feitas pelo sujeito são dados tanto escrito quanto auditivo pelo programa (FIG. 8). O teste encerra-se se o sujeito completar 6 categorias ou ao terminarem as 128 cartas-resposta. O desempenho é registrado automaticamente pelo programa, fornecendo dados quanto ao número de acertos, número de erros, respostas perseverativas, erros perseverativos, erros não-perseverativos, nível conceitual, categorias completadas, ensaios para completar primeira categoria, fracasso em manter o contexto e aprendendo a aprender. No apêndice A encontram-se definições sobre termos específicos deste teste. Já no ANEXO B pode se verificar a programação desta versão.



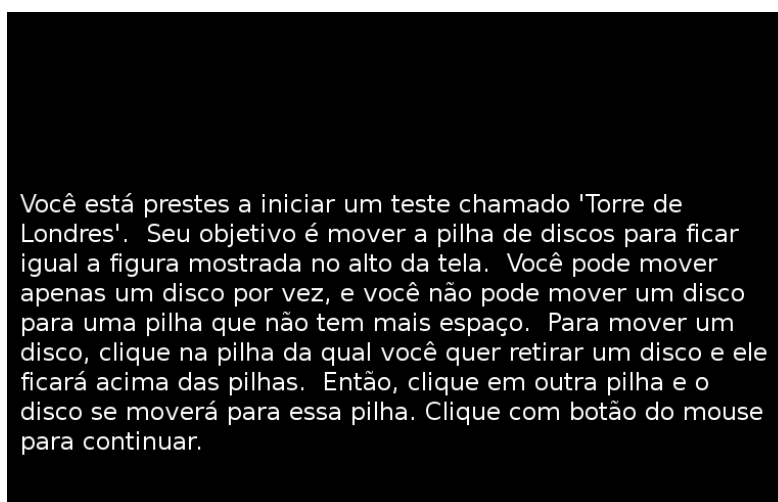
**Figura 8-** Telas do BCST.

a) feedback de acerto; b) feedback de erro; c) telacom a apresentação da carta-resposta

### 3.3.4 Torre de Londres (ToL)

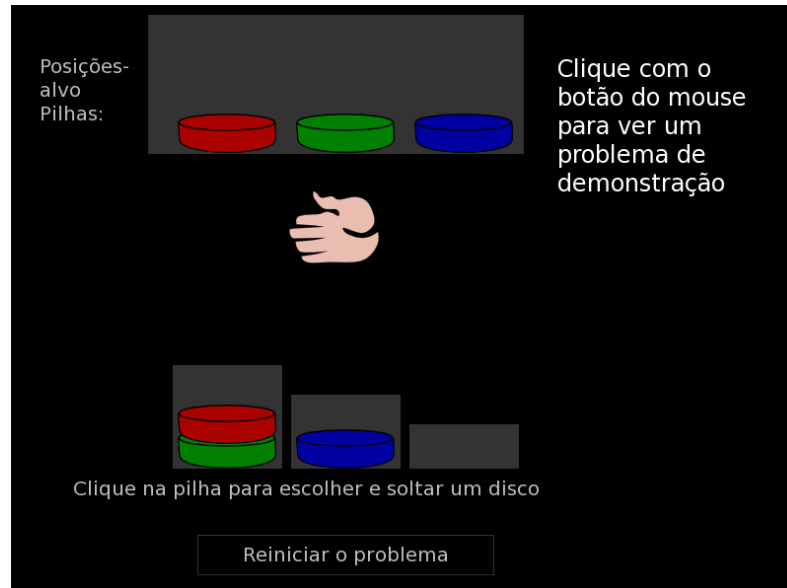
A versão da ToL foi configurada de forma similar a versão manual de Shallice-Krikorian. Em pesquisa anterior, a instrução deste teste no programa PEBL apresentava-se em uma única tela, o que foi uma das hipóteses para prejuízo dos sujeitos. Por isso, nesta versão da presente pesquisa a instrução está dividida em duas telas. As tentativas de configuração envolviam inserir imagens que auxiliassem a compreensão do teste, mas isto ainda não foi possível.

Para realizar o teste, o sujeito utiliza o mouse. Após registrar o nome/sigla do sujeito, o programa inicia a instrução auditiva/visual sobre o teste (FIG. 9). A instrução está subdividida em duas etapas.

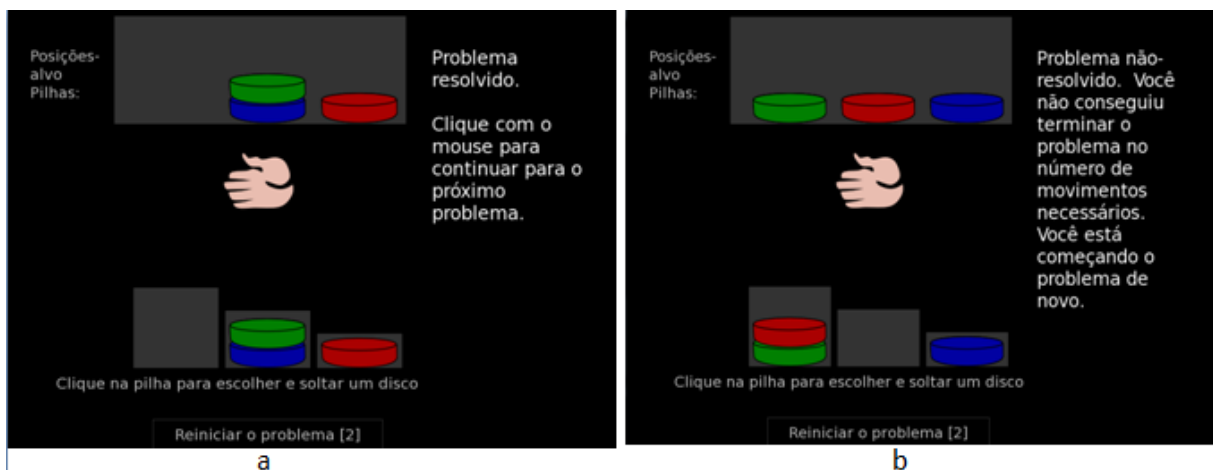


*Figura 9-* Instrução ToL.

Em seguida, para o sujeito se familiarizar com o teste, é realizado um problema de demonstração. Conforme se observa na FIG. 10, o “cartão problema” fica exposto no alto da tela. O objetivo é fazer com que a configuração mostrada no centro da tela fique igual ao cartão problema. São fornecidos *feedbacks* de acerto ou erro (FIG. 11). Para cada problema, o sujeito tem até 3 tentativas para resolver, sendo que quanto menos tentativas gastar, maior será sua pontuação.



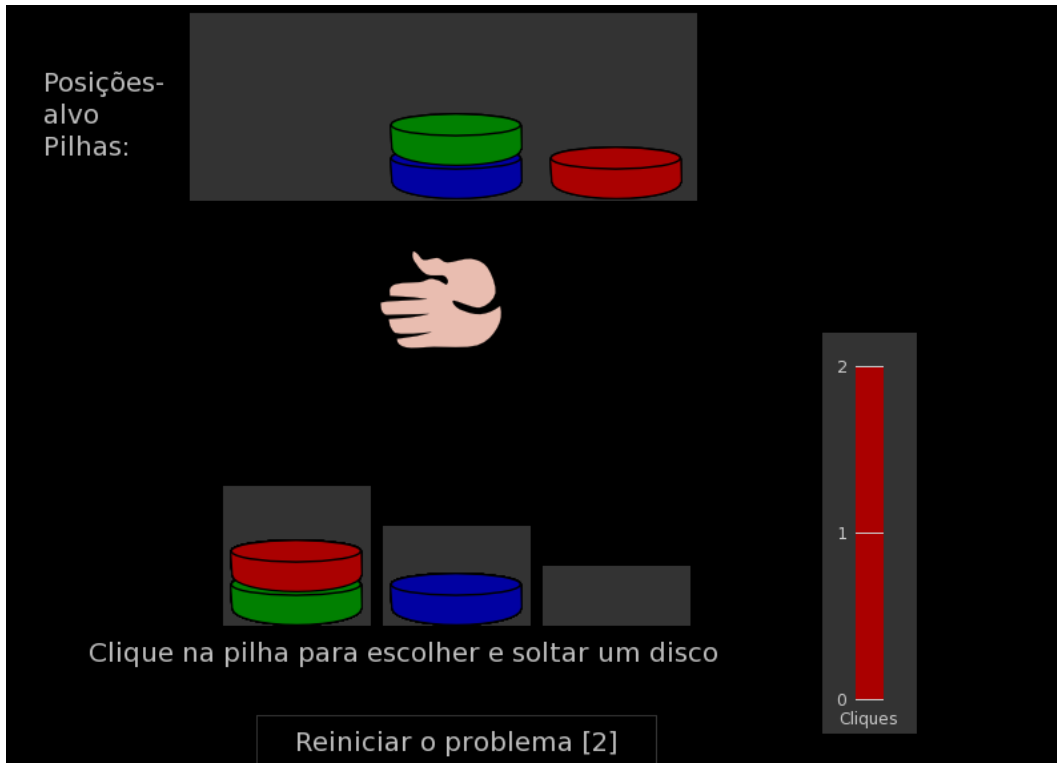
**Figura 10-** Tela inicial do problema de demonstração da ToL.



**Figura 11-** Tela da ToL

a) feedback acerto; b) feedback erro.

No lado direito aparece uma barra indicando o número de movimentos disponíveis para cada problema, sendo que é considerado um movimento quando o sujeito tira o disco de um lugar e coloca novamente em um local (mesmo sendo o local de onde tirou) (FIG. 12). São registrados pelo programa os movimentos realizados, pontuação e tempos em milissegundos, de cada problema e total. A configuração deste teste encontra-se no ANEXO C.



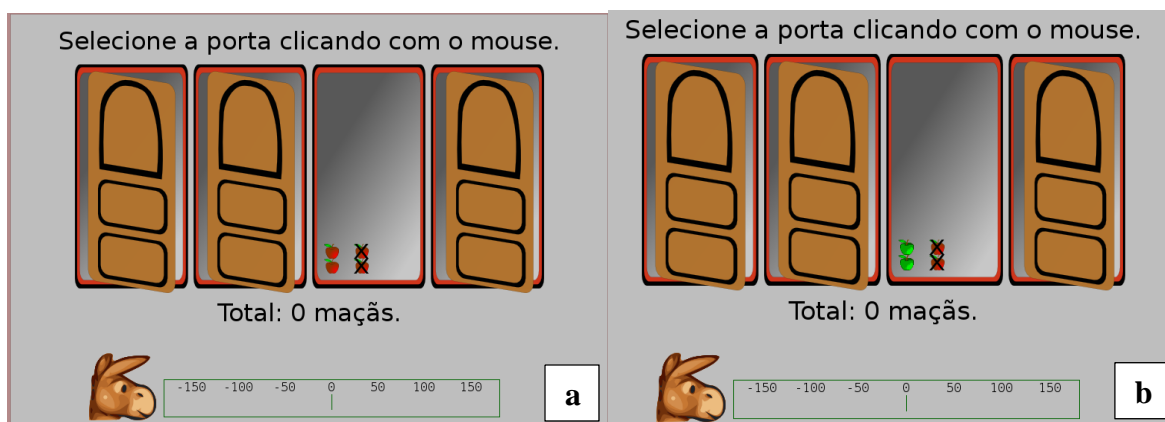
*Figura 12-* Tela de problema da ToL, mostrando a barra de movimentos.

### 3.3.5 Hungry Donkey

Durante a introdução, foi mencionado um dos testes mais utilizados para avaliação de tomada de decisões, o Iowa Gambling Task (IGT). Nesta tarefa, as probabilidades de recompensa e punição são manipuladas pelo sistema. Os participantes são apresentados a 4 (quatro) pilhas de cartas e instruídos a escolher cartas objetivando somar a maior quantidade de dinheiro possível. Duas pilhas de cartas gerarão uma maior ganho, porém maior perda. Ao passo que as outras duas pilhas fornecerão menor ganho, porém menor perda. A longo prazo, as pilhas com ganho menor fornecem maior ganho financeiro (CRONE; MOLEN, 2007).

Para estimular o interesse de crianças e tornar a tarefa mais significativa, foi elaborada a versão Hungry Donkey, no qual o sujeito deve auxiliar um burro a juntar maçãs (CRONE; MOLEN, 2007). Para tanto, são exibidas 4 portas nas quais 2 fornecerão mais maçãs, porém cobrarão mais maçãs como preço da porta; e outras 2 portas fornecerão menos maçãs, mas cobrarão menos maçãs como preço da porta.

Originalmente, o programa PEBL configurou telas com maçãs vermelhas para ganhos e maçãs vermelhas com um X para perdas (FIG. 13a). Em levantamento bibliográfico, observou-se que alguns autores utilizavam maçãs verdes para o reforço e vermelhas para a punição (CRONE & MOLEN, 2007; CARLSON *et al.*, 2009). Desta forma, configurou-se a versão do PEBL para maçãs verdes como ganhos e vermelhas como perdas (FIG. 13b), o que mostrou visualmente ser mais interessante e facilitando a discriminação entre reforço e punição.



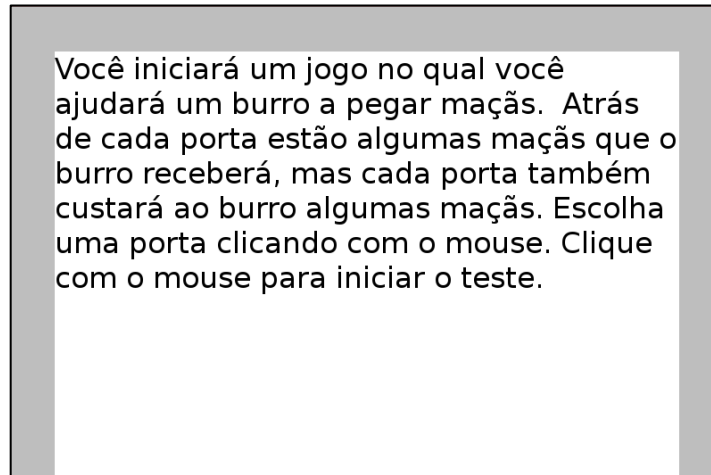
**Figura 13-** Teste Hungry Donkey.

a) Maçãs vermelhas: maçãs ganhas. Maçãs vermelhas com X: maçãs cobradas (perdas). b) Maçãs verdes: maçãs ganhas. Maçãs vermelhas com X: maçãs cobradas (perdas).

Outra alteração realizada na configuração do PEBL é o tempo de exposição do reforço e/ou punição, ou seja, o tempo que a porta permanecia aberta. Na versão original, o estímulo ficava exposto por 200 ms. Em outros estudos, observou-se variação entre 1,000 ms e 6000 ms (CRONE; MOLEN, 2007; CARLSON *et al.*, 2009). Optou-se em dobrar o tempo, 400 ms, e observou-se que o tempo de latência já era significativo.

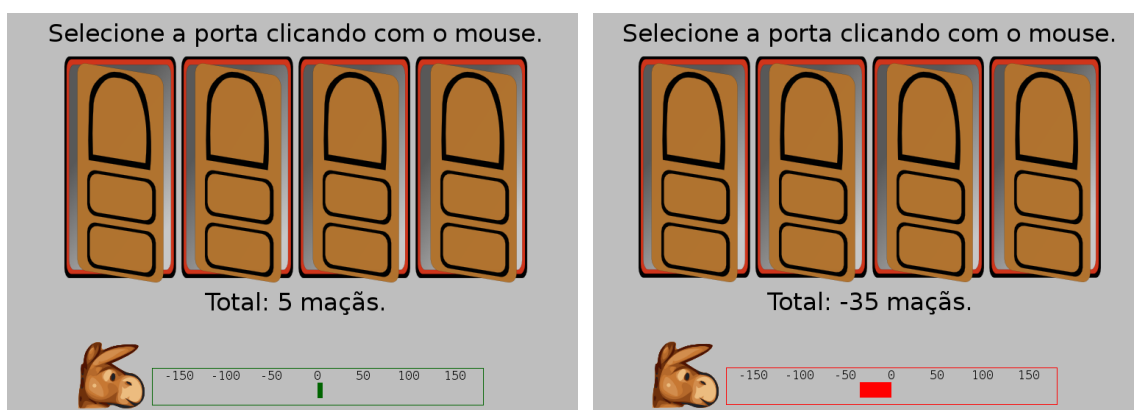
Uma última variação do teste refere-se ao número de tentativas. Optou-se em manter 150 tentativas, conforme programação original do PEBL. Porém, o número de tentativas também apresentou divergências nos estudos, variando entre 100, 200 e 280 tentativas (CRONE; MOLEN, 2007; CARLSON *et al.*, 2009; LAMBEK *et al.*, 2010).

O teste é executado com mouse. Após a tela inicial onde é inserido o nome do participante, aparece a tela de instrução. Neste teste não foi possível inserir o áudio, por isso os aplicadores liam a instrução da tela para os sujeitos (FIG. 14).



*Figura 14-* Tela de instrução do Donkey.

Após clicar, inicia-se o teste com uma tela onde o sujeito visualiza 4 portas: A, B, C e D. Abaixo delas é possível ver o total de maçãs tanto escrito quanto indicado por uma barra que quanto aumenta ou diminui de acordo com o número de maçãs total que o sujeito possui. Se for um valor positivo, a barra estará verde. Mas caso o valor seja negativo, a barra estará vermelha (FIG. 15). Conforme mencionado na configuração deste teste no capítulo introdutório, as maçãs verdes indicam ganho e as maçãs vermelhas indicam perda.



*Figura 15-* Telas do teste Donkey.

A esquerda observa-se a barra verde, indicativa de valor positivo. A direita, barra vermelha, indicativa de valor negativo.

Nesta configuração, as portas A e B fornecem um reforço maior imediato (4 maçãs por clique), porém “cobram” um valor maior, sendo que a cada 10 cliques perde-se 50 maçãs. A diferença entre estas portas é que na porta A, as perdas são divididas ao longo das 10 tentativas. Já a porta

B, durante as 10 tentativas, em algum momento definido de maneira aleatória, o sujeito perderá as 50 maçãs de uma vez. Um procedimento similar ocorre nas portas C e D, porém o reforço imediato é menor (2 maçãs por clique) e as perdas também são menores (máximo 10 maçãs a cada 10 tentativas). A porta C desconta as 10 maçãs em 5 vezes ao longo das 10 tentativas, ao passo que a porta D desconta as 10 maçãs em um único momento nesse mesmo intervalo de tentativas. Desta forma, para obter um bom desempenho, o sujeito deverá selecionar mais vezes as portas C e D. O total de tentativas é 150, sendo que após o clique a duração do estímulo (porta aberta) é de 400 ms. Optou-se em remover o resultado final do teste, deixando-se apenas um agradecimento pela participação na tela final. No ANEXO D pode se verificar a programação desta versão.

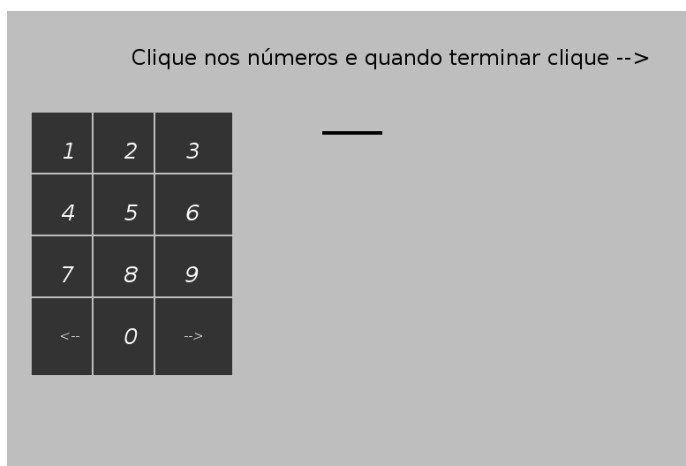
O programa gera apenas os dados brutos. Apesar de ter a programação de cálculo, o programa não gerava este arquivo. Desta forma, foi elaborada uma fórmula em programa de cálculo (Excel) onde se obtinha o número de total de cliques nas portas a, b, c e d; número de portas a+b (desvantajosas), números de portas c + d (vantajosas) e número de portas vantajosas menos desvantajosas.

### 3.3.6 Teste de Dígitos (Dígitos)

O Teste de Dígitos é um teste amplamente utilizado para a avaliação de memória de trabalho, sendo uma de suas versões mais conhecidas pertencente à Escala Wechsler de Inteligência. A tarefa possui duas etapas, uma direta e outra inversa. Em ambas as versões, o examinador fala uma sequência numérica, utilizando números de 2-9 e sem repetir o número durante uma lista. Os níveis são crescentes, iniciando com 2 dígitos e finalizando em 8 dígitos. Em cada nível o sujeito tem duas tentativas. Se acertar pelo menos uma tentativa do nível, passa para o nível seguinte. Se errar as duas tentativas, o teste se encerra (HESTER; KINSELLA; ONG, 2004).

Foram desenvolvidas duas versões deste teste: direta e inversa. Ambas são respondidas com o mouse (FIG. 16). A execução das duas versões são similares, diferenciando apenas na forma de

resposta que o sujeito deverá emitir. Aplicava-se sempre a versão direta e em seguida a inversa, tal como ocorre em outras aplicações (exemplo: versão WISC).



**Figura 16-** Teclado virtual utilizado para respostas do Teste de Dígitos.

Assim, após o preenchimento do nome, aparece a tela de instrução com áudio. Em seguida, surge áudio e escrito a palavra “prepare-se” como aviso ao estímulo que será falado. O objetivo da versão direta é clicar nos números que surgirem na tela exatamente na ordem que foi falada pelo programa. O programa gera lista de número aleatórias, mas sem repetir o número em uma mesma lista. As listas aumentam a cada duas tentativas, iniciando com 2 números e finalizando com 8 (níveis). Porém, o sujeito só prossegue no teste se acertar pelo menos um item da lista. Caso erre as duas tentativas do nível, o teste se encerrará. Ainda, é possível corrigir a resposta durante a execução: quando o sujeito percebe que errou, pode clicar na seta para a esquerda (na tela) e apagar o número selecionado<sup>12</sup>. Sempre que conclui sua lista, o sujeito deve clicar na seta para a direita para visualizar e ouvir o *feedback*.

Na versão direta, o sujeito deve repetir os números na mesma sequência escutada. Já na versão inversa, a diferença é que o sujeito deverá clicar nos números na ordem inversa da que escutou. O programa gera número de acertos, nível máximo atingido e tempo gasto para a execução de cada teste. A programação deste teste encontra-se no ANEXO E.

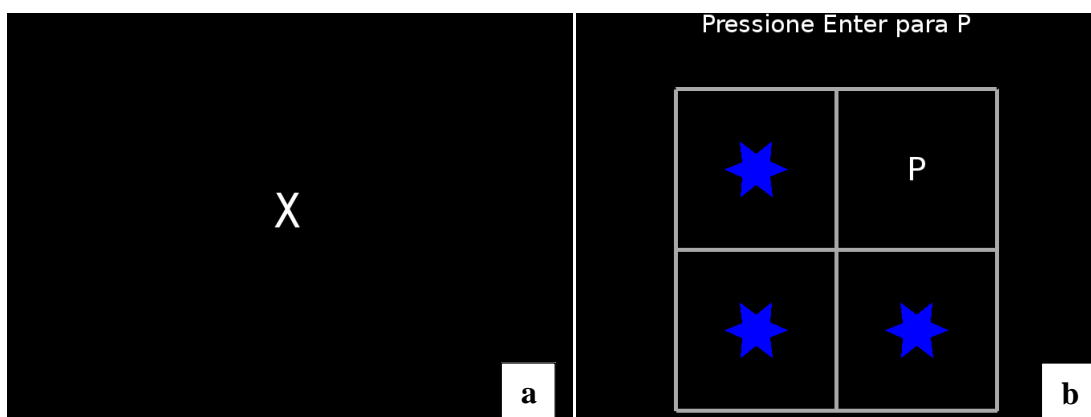
<sup>12</sup> Outro problema identificado no programa é que caso o sujeito volte mais itens do que os existentes na tela o programa encerra, tendo que reiniciar o teste.

### 3.3.7 Outros testes

Durante a configuração e seleção dos testes, alguns foram configurados e testados, mas por razões diversas não foram utilizados na aplicação.

O teste de trilhas foi configurado similar a versão manual conhecida, inclusive a distribuição dos números (parte A) e números e letras (parte B). O teste não foi utilizado por já haverem outros testes que abrangiam os aspectos avaliados por ele, tais como atenção e flexibilidade cognitiva.

Os testes Continuous Performance Task (CPT) e Go-No-Go também foram traduzidos e adaptados para uso. Inicialmente, utilizou-se o Teste CPT para avaliação de controle inibitório. Porém a aplicação dura em média 16 minutos, sendo 2 minutos de instrução e 14 minutos de execução do teste propriamente dito. O teste é composto por uma tela preta onde são exibidas letras, uma por vez, no centro da tela (FIG. 17a). O objetivo é pressionar a barra de espaço para todas as letras que surgirem, exceto a letra “X”. Observou-se que as crianças demonstravam inquietação durante a execução, olhando para o lado e clicando aleatoriamente.



**Figura 17**-Testes CPT e Go-no-go.  
a) Tela de teste do CPT. b) Tela de teste do Go no-go.

Por isso, optou-se em trocar para o teste Go-No-Go, que avalia os mesmos processos, contudo tem uma duração menor (em torno de 10 minutos) e sua visualização é mais atrativa (FIG17b).

A tela base deste teste é composta por 4 estrelas e as letras P e R aparecerão alternadamente no lugar de uma dessas estrelas. Aparecem mais letras P do que R. São executadas duas etapas com objetivos diferentes: na primeira etapa o sujeito deve clicar nas letras P que surgirem na tela; na segunda etapa o sujeito deve clicar nas letras R. Desta forma, na primeira etapa ocorre um treino para uma resposta preponderante e ao mesmo tempo inibição de um estímulo menos frequente. Ao passo que na segunda etapa, o sujeito deverá inibir a resposta preponderante. Segundo Pedron e Reppold (2010a) é “um paradigma de emissão e supressão de respostas, no qual, frente a um determinado grupo de estímulos, o sujeito deve emitir a resposta; frente a outros, deve suprimi-la”.

O teste Go-No-Go, contudo, apresentou erro no registro das respostas. O problema está sendo discutido com os autores do programa.

### 3.3.8 Teste das Matrizes Progressivas de Raven

Este teste foi aplicado com o intuito de nivelamento entre os participantes. Trata-se de um teste não verbal para avaliação da inteligência geral. Destina-se a crianças de 5 anos a 11 anos e meio, deficientes mentais e pessoas idosas, contemplando, assim, o público alvo desta pesquisa.

Os itens do teste são apresentados sob a forma de um desenho em que falta uma parte e a tarefa do sujeito é escolher, entre as alternativas colocadas, a que completa corretamente o desenho. A Escala contém 36 itens divididos em três séries (A, Ab e B) de 12 itens, em ordem de dificuldade crescente em cada uma delas. Foi adaptado o protocolo da resposta para anexar junto ao protocolo da pesquisa, que se encontra no ANEXO F.

Para análise do Teste Raven, faz-se necessário atentar-se para a forma correta de interpretação dos dados. Quando consideramos o critério pontuação, considera-se um bom desempenho quanto mais elevado for o valor. Já ao utilizar os itens de percentil e classificação, por uma

questão de nomenclatura do teste, quanto mais elevado o número, pior será o desempenho. Para facilitar, segue abaixo tabela com o significado do desempenho no Teste Raven conforme classificação e percentil (TAB. 1).

**Tabela 1-** Notação para classificação e percentil do Raven

Percentil	Classificação	Significado	Numeração estatística
Maior que 95	Maior que 95	I	0
75-94	90-94	II+	1
26-74	75-89	II	2
06-25	50-74	III+	3
Menor que 5	26-49	III-	4
	11-25	IV	5
	06-10	IV-	6
	Menor que 5	V	7

### 3.3.9 Questionários utilizados

Os pais preencheram alguns questionários, iniciando por um informando dados da criança, informações sobre uso de medicamento, se faz tratamento, hábitos de computador e videogame. Responderam também a versão 2013 do Critério socioeconômico Brasil. O questionário se utiliza de questões sobre itens de consumo e conforto que a família dispõe, além da escolaridade do chefe da família. A partir do seu escore final é definida a qual classe socioeconômica o indivíduo pertence (A1, A2, B1, B2, C1, C2, D ou E).

Outros dois questionários foram utilizados para investigar sintomas de TDAH e outros transtornos. O SNAP-IV (Swanson, Nolan e Pelham-IV) é um questionário de triagem para avaliar a presença de sintomas de desatenção e hiperatividade, de acordo com critérios do DSM-IV. Os pais preencheram o questionário SNAP-IV com perguntas adicionais para cumprir com todos os critérios diagnósticos. As perguntas adicionais referiam-se a:

- Os sintomas assinalados causam prejuízo;
- Presença de sintomas de desatenção e hiperatividade antes dos 7 anos;
- Presença dos sintomas em mais de um contexto.

Para sua correção, considera-se que 6 ou mais itens assinalados como bastante ou demais em uma das subescalas é significativo para suspeita diagnóstica de TDAH. Utilizou-se também

outra forma de correção para pontuação deste questionário, no qual cada item foi pontuado numa escala de 0 a 3 (0=Quase nunca; 1=Às vezes; 2= Bastante; 3=Demais). A soma dos itens gerou a pontuação de cada subescala, utilizada para comparação no decorrer das análises.

No presente trabalho, os participantes foram divididos em um grupo com TDAH e outro grupo sem TDAH de acordo com a definição do DSM-IV. Para tanto, o grupo “com TDAH” era definido através da aplicação do questionário SNAP-IV, sendo significativo quando obtivesse pelo menos 6 itens “bastante” ou “demais”, presença de sintomas antes dos 7 anos e em mais de um contexto, além de os sintomas causarem prejuízos.

Já o questionário SDQ (Strengths and Difficulties Questionnaire) é considerado um rastreio para problemas de saúde mental infantil, subdividido em cinco subescalas que são: problemas emocionais, hiperatividade, relacionamento, conduta e comportamento pró social, além de gerar uma pontuação total (CURY; GOLFETO, 2003). A classificação dos pontos, tanto das subescalas como total, pode ser interpretada como:

- Normal: não há sintomas significativos.
- Limítrofe: tendência a apresentar o problema.
- Anormal: há presença de sintomas significativos.

Apenas na subescala comportamento pró social uma pontuação elevada indica normalidade. Nas demais subescalas, quanto mais alta a pontuação pior será o desempenho do indivíduo.

Os questionários encontram-se no ANEXO F, junto com o protocolo.

### 3.4 Procedimentos

Os alunos que frequentam as escolas selecionadas foram convidados a participar da pesquisa por meio de um convite enviado aos pais ou responsável (ANEXO G). Aos participantes e aos

pais e/ou responsável que concordaram e não se encaixaram nos critérios de exclusão do estudo, foram explicados os procedimentos do estudo, sendo em seguida solicitado a ambos a assinatura do termo de consentimento livre e esclarecido (ANEXO H).

Após o consentimento dos pais e/ou responsável, estes preencheram um formulário referente aos dados sociodemográficos, questionário sobre dados da criança, a escala SNAP-IV adaptada e SDQ (ANEXO G). O participante respondeu a um questionário referente ao seu nível de conhecimento em informática e hábitos referente a jogos de vídeo-game (ANEXO G). A primeira parte do questionário foi elaborada baseando-se no Módulo I do questionário da “Pesquisa sobre o uso das Tecnologias da Informação e da Comunicação no Brasil 2009” (BARBOSA, 2010). Em seguida, os participantes foram examinados com a bateria de testes computadorizados. Foi aplicado, ainda, o Teste das Matrizes Progressivas de Raven para verificar se os grupos estavam homogêneos.

### 3.5. Aspectos éticos

Não houve interferência significativa nas atividades escolares. O risco que poderia acarretar aos participantes foi apenas a possibilidade de constrangimento em virtude dos testes propostos. Foram tomadas todas as medidas necessárias para evitar que isso ocorresse, bem como qualquer identificação dos alunos. A pesquisa pode trazer benefício ao aluno e a seus familiares, pois o instrumento utilizado revela dados de interesse clínico. O estudo também contribuirá para o desenvolvimento futuro de pesquisas na área de problemas cognitivos e comportamentais no Brasil.

Os dados coletados foram compilados e analisados em conjunto, mantendo-se o anonimato dos participantes. O uso dos dados restringiu-se aos objetivos da pesquisa.

Foi assegurado ao indivíduo que a participação no estudo não implicaria em vantagens para os voluntários e bem como não acarretaria em nenhum tipo de prejuízo pela não participação. Não

foi observado nenhum aspecto nocivo das avaliações aos participantes, por isso a pesquisa prosseguiu de acordo com o planejamento.

O projeto possui aprovação pela Comissão de Ética da Universidade Federal de Minas Gerais (ETIC nº. 0317.0.203.000-10).

### 3.6 Análise estatística

Foi utilizado o programa SPSS (Statistical Package for the Social Sciences) versão 20.0 para análise estatística. O teste de Kolmogorov-Smirnov foi utilizado para avaliar a normalidade das amostras. Como foi constatado se tratar de uma amostra que não segue distribuição normal (não Gauseana), utilizaram-se análises descritivas e os testes estatísticos não paramétricos de Mann-Whitney, correlações de Spearman e Chi quadrado. Um valor de p bilateral menor que 0,05 foi adotado como nível de significância estatística para todos os testes.

# Resultados

## 4 RESULTADOS

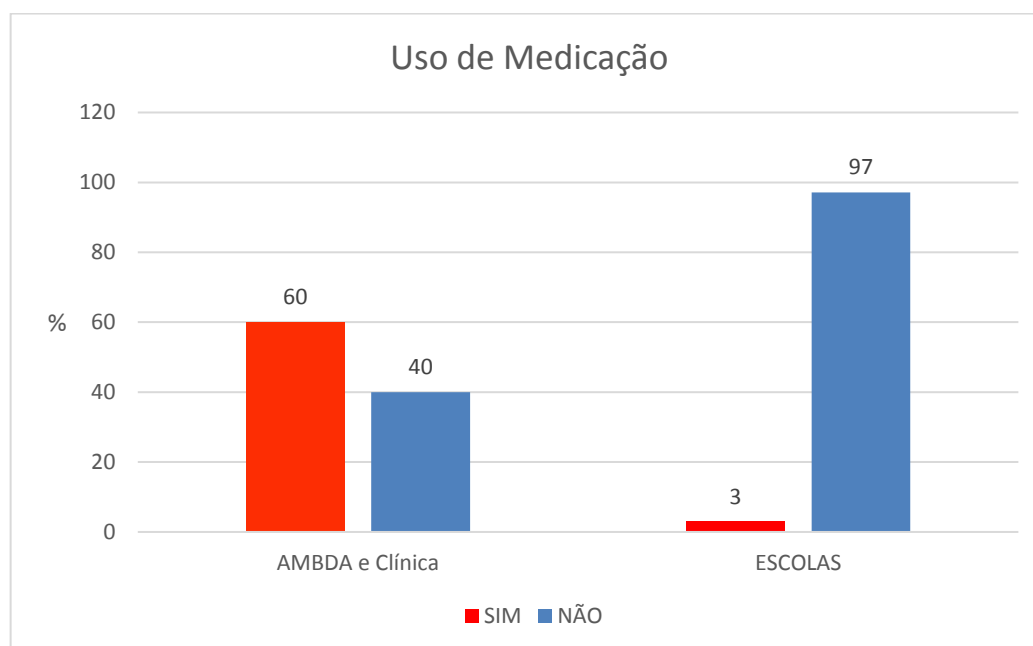
### 4.1 Características dos Participantes

Participaram desta pesquisa 50 crianças, sendo a maioria de escolas públicas (91,8%; n=45). Preencheram critérios diagnósticos<sup>13</sup> para TDAH 54% (n=27) dos participantes e 20% (n=10) utilizavam medicação, sendo mencionado uso de ritalina, clonazepan, imipramina e risperidona (TAB.2).

**Tabela 2-** Características dos Participantes.

Características	
Escola, Pública/Particular (%), n=49	45 (91,8)/ 4 (8,2)
Diagnóstico, Sim/Não(%), n=50	27(54) / 23 (46)
Medicação, Sim/Não(%), n=50	10 (20) / 40 (80)
Idade, média (desvio padrão), anos, n=50	8,62 ( $\pm$ 1,24)
Gênero, Masculino/Feminino (%), n=50	28 (56) / 22 (44)

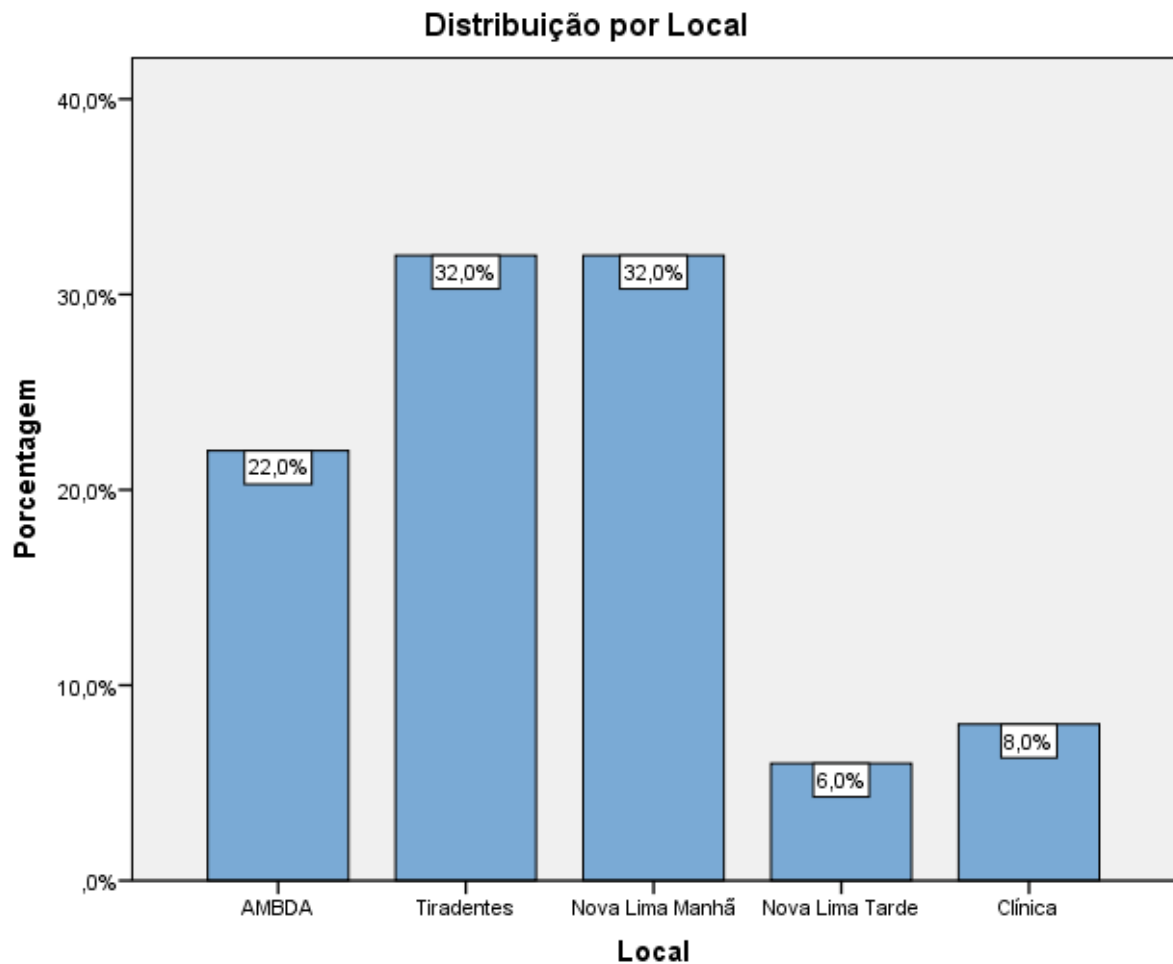
Além disso, a maior parte das crianças que utilizam medicamento pertencem a amostra do AMBDA e clínica (GRAF. 1).



**Gráfico 1-** Distribuição dos participantes por uso de medicamento.  
(Teste de Fisher:  $p < 0,001$ )

<sup>13</sup> Conforme já descrito previamente nos métodos, crianças “com TDAH” nesta pesquisa, são aquelas que obtiveram no SNAP –IV: pontuação significativa, presença de sintomas antes dos 7 anos e em mais de um contexto, além destes sintomas causarem prejuízos.

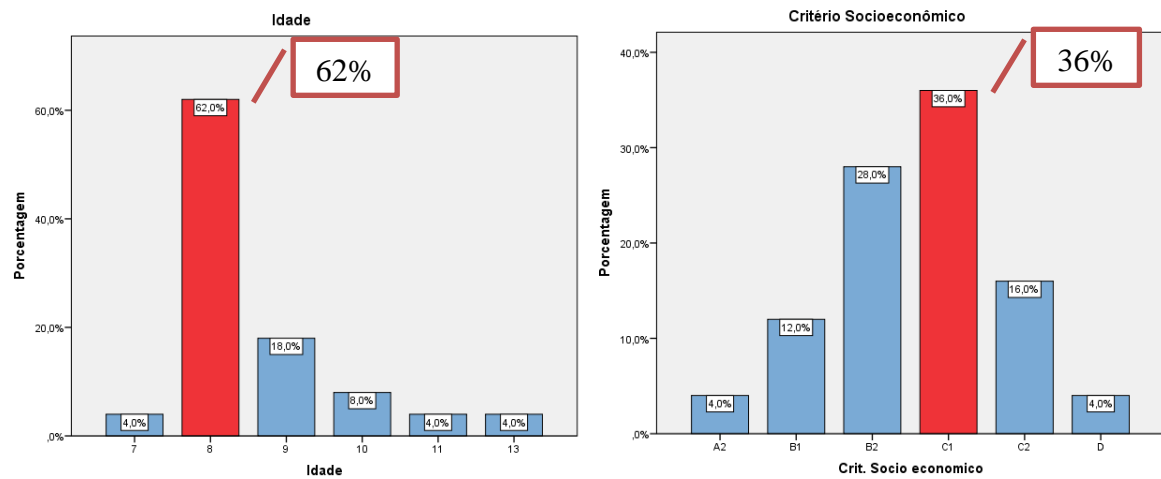
No Gráfico 2, pode-se observar o número de participantes de cada local, sendo 40% de Belo Horizonte, 38% de Nova Lima e 22% do AMBDA.



**Gráfico 2-** Distribuição dos participantes por Local.

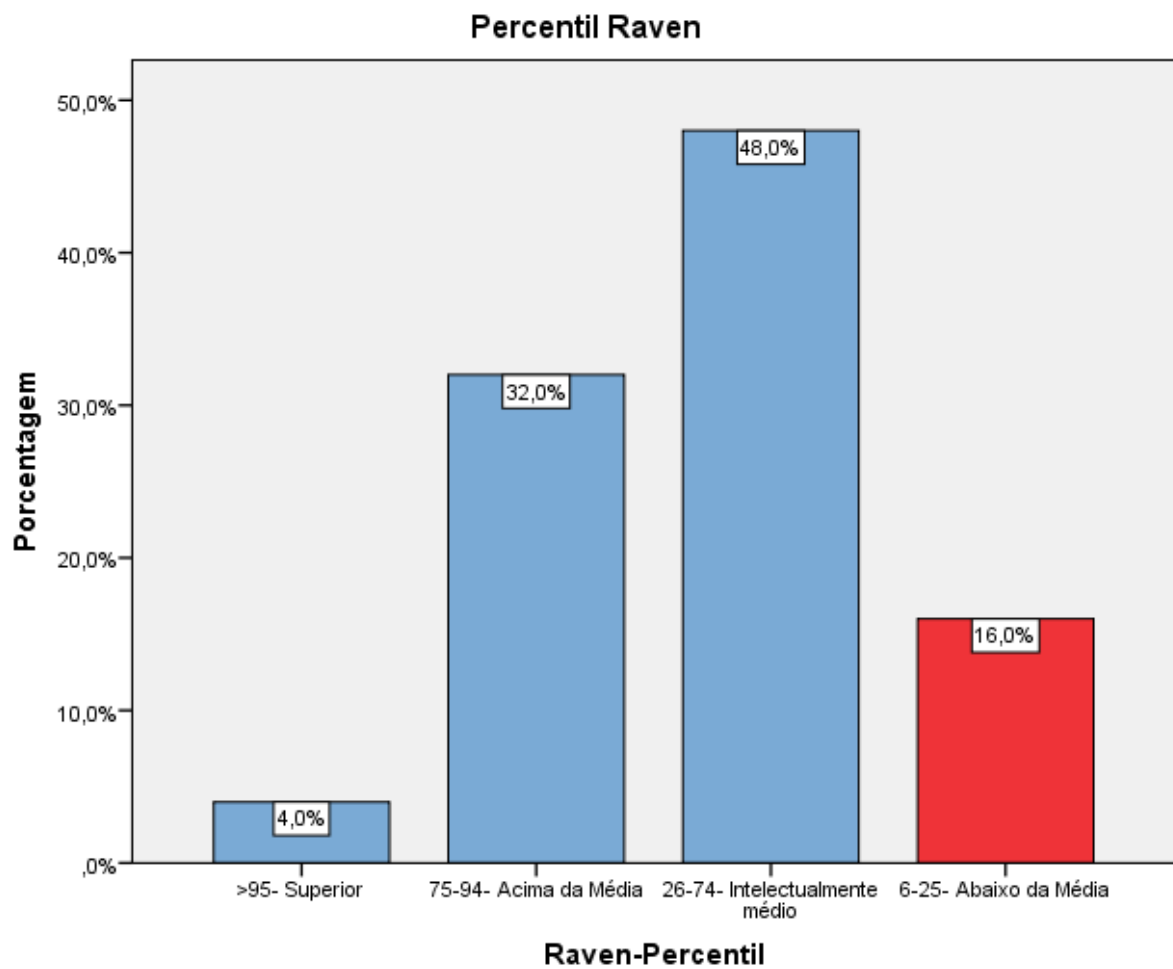
(AMBDA: 22%; Belo Horizonte (Tiradentes+Clínica): 40%; Nova Lima (Nova Lima Manhã+ Nova Lima Tarde): 38%)

Conforme descrito na Tabela 2, a idade média dos participantes foi de 8,62 anos (mínima=7, máxima=13, 62% com 8 anos, GRAF. 3), a maioria com pontos socioeconômicos equivalentes a Classe C1 (36%) e discreta preponderância do gênero masculino (56%).



**Gráfico 3-** Distribuição dos participantes por Idade e Critério Socioeconômico.

Quanto ao desempenho do Teste Raven, utilizado para pareamento intelectual, a maioria (84%, n=42) apresentou desempenho na média ou acima da média (GRAF. 4).



**Gráfico 4 -** Desempenho no teste Raven.

## 4.2 Comparações entre grupos com e sem o Diagnóstico de TDAH

Os grupos com e sem TDAH não diferiram em relação à proporção de gêneros, apesar de na amostra geral os meninos terem obtido pontuação mais elevada na subescala de desatenção do SNAP ( $p=0,015$ , média meninos= 5,04, DP=3,17; média meninas=2,91, DP=2,99) e na subescala problemas de conduta do SDQ ( $p=0,017$ , média meninos= 3,69, DP=2,49; média meninas=2,09, DP=1,97). O grupo de participantes com TDAH foi composto de crianças e adolescentes mais velhos, mas não houve diferença em relação ao nível socioeconômico (TAB. 3).

**Tabela 3-** Comparações entre os grupos com e sem TDAH quanto a gênero, idade e critério socioeconômico.

	Sem TDAH (n=23)	Com TDAH (n=27)	Valor de p
Gênero (M/F)	12 (42,9) /11 (50)	16 (57,1) /11 (50)	p=0,615
Idade**	8,26 (1,17)	8,93 (1,23)	p=0,004
Critério Socioeconômico**	23,96 (6,49)	21,48 (5,34)	p=0,099

\*\* Média (Desvio Padrão); Mann-Whitney U.

Analisando os dados dos grupos com e sem TDAH, notou-se que a média do grupo com TDAH no questionário SNAP-IV foi significativamente superior a média do grupo sem TDAH. Da mesma forma, na maioria das subescalas do SDQ o grupo de TDAH apresentou mais sintomas, exceto para comportamento pró social onde não houve diferença entre os grupos (TAB. 4).

**Tabela 4-** Comparações entre os grupos com e sem TDAH quanto a Idade, Critério Sócioeconômico, Raven, SNAP pontuação desatenção e hiperatividade, SDQ.

	Sem TDAH (n=23)*	Com TDAH (n=27)*	Valor de p
Idade	8,26 (1,17)	8,93 (1,23)	p=0,004*
Critério Socioeconômico	23,96 (6,49)	21,48 (5,34)	p=0,099
Raven	22,96 (5,39)	21,78 (5,32)	p=0,494
SNAP Desatenção	6,83 (4,16)	18,07 (5,82)	p<0,001*
SNAP Hiperatividade	6,43 (3,95)	17,41 (6,39)	p<0,001*
SDQ Total	19,70 (5,37)	27,40 (6,54)	p<0,001*
SDQ Emocional	3,61 (2,04)	5,24 (2,24)	p=0,016*
SDQ Problemas de Conduta	2,04 (1,89)	3,80 (2,52)	p=0,009*
SDQ Hiperatividade	4,09 (2,33)	7,40 (2,04)	p<0,001*
SDQ Problemas com Colegas	1,87 (1,69)	3,28 (1,97)	p=0,014*

SDQ Comportamento Pró Social 8,09 (1,76) 7,52 (2,22) p=0,377

\* Média (Desvio Padrão); Mann-Whitney U.

Crianças com TDAH obtiveram mais maçãs no teste Donkey. Já em relação aos demais testes, crianças sem TDAH apresentaram desempenho melhor no BCST, considerando-se os índices de acertos, erros, erros não-perseverativos e nível conceitual. Outro teste em que se observou diferença significativa foi na Torre de Londres, no qual crianças sem TDAH fizeram pontuação maior (TAB. 5).

**Tabela 5-** Comparações significativas entre Diagnóstico SNAP-IV e Idade, Teste de Donkey, BCST, ToL.

	Sem TDAH (n=23)*	Com TDAH (n=27)*	Valor de p
Teste de Donkey Total	-20,78 (26,84)	-6,70 (40,09)	p=0,048
BCST corretas	76,91 (11,53)	64,62 (18,34)	p=0,019
BCST erros	48,30 (14,09)	62,34 (17,66)	p=0,010
BCST erros não perseverativos	23,04 (11,85)	37,33 (22,37)	p=0,018
BCST nível conceitual	61,91 (14,84)	47,77 (21,33)	p=0,014
Torre de Londres pontuação	20,74 (6,84)	16,81 (7,62)	p=0,047

\* Média (Desvio Padrão); Mann-Whitney U.

Não se observou mais nenhuma associação significativa. Todas as comparações realizadas encontram-se no Anexo I.

### 4.3 Correlações com Pontuação do SNAP-IV e SDQ

Realizaram-se ainda análises considerando a pontuação nas subescalas de desatenção e hiperatividade do SNAP-IV (conforme descrito nos métodos), além das subescalas de hiperatividade, conduta e total do SDQ.

Nas análises com a pontuação do SNAP-IV, observou-se que quanto mais velhas as crianças, mais sintomas de desatenção (p=0,027, r=0,314) e hiperatividade (p=0,026, r=0,316) apresentaram. No teste Raven, crianças com menor pontuação obtiveram mais sintomas nas

duas subescalas (Raven classificação: desatenção- $p=0,038$ ,  $r=0,294$ ; hiperatividade- $p=0,046$ ,  $r=0,284$ ). Crianças com mais sintomas de desatenção, ainda, cometeram mais erros no Stroop ( $p=0,057$ ,  $r=0,279$ ) e tiveram um pior desempenho na Torre de Londres (pontuação:  $p=0,038$ ,  $r=-0,295$ ; tentativas:  $p=0,051$ ,  $r=0,277$ ). (TAB.6)

**Tabela 6-** Correlações com a Pontuação do SNAP-IV nas subescalas de Desatenção e Hiperatividade

	<b>Desatenção valor de p (r)</b>	<b>Hiperatividade valor de p (r)</b>
Idade	0,027 (0,314)*	0,026 (0,316)*
Raven	0,038 (0,294)*	0,046 (0,284)*
Stroop erros	0,057 (0,279)*	0,438 (0,116)
Torre de Londres pontuação	0,038 (-0,295)*	0,077 (-0,253)
Torre de Londres tentativas	0,051 (0,277)*	0,136 (0,214)

\*Houve correlação

As correlações com as pontuações do SDQ indicaram que quanto mais velhas as crianças mais sintomas de hiperatividade (SDQ hiperatividade:  $p=0,025$ ,  $r=0,324$ ) ou alteração em alguma subescala (SDQ total:  $p=0,020$ ,  $r=0,334$ ). Crianças de classes socioeconômicas mais alta apresentaram menos sintomas de hiperatividade ( $p=0,055$ ,  $r=-0,279$ ), problemas de conduta ( $p=0,013$ ,  $r=-0,355$ ) e alteração no SDQ geral ( $p=0,002$ ,  $r=-0,429$ ). (TAB. 7)

**Tabela 7-** Correlações com a Pontuação do SDQ nas subescalas de Hiperatividade, Conduta e Total

	<b>Conduta valor de p (r)</b>	<b>Hiperatividade valor de p (r)</b>	<b>Total valor de p (r)</b>
Idade	0,340 (0,141)	0,025 (0,324)*	0,020 (0,334)*
Critério Socioeconômico	0,013 (-0,355)*	0,055 (-0,279)*	0,002 (-0,429)*
Raven	0,082 (0,254)	0,069 (0,295)	0,017 (0,343)*
Dígitos nível	0,052 (-0,283)*	0,073 (-0,261)	0,022 (-0,330)*
Dígitos acertos	0,084 (-0,252)	0,286 (-0,157)	0,027 (-0,318)*
Dígitos Invertido acertos	0,031 (-0,312)*	0,347 (-0,139)	0,244 (-0,172)
Stroop erros	0,459 (0,113)	0,124 (0,233)	0,012 (0,373)*
Stroop aleatórios	0,435 (0,119)	0,176 (0,205)	0,028 (0,328)*
Torre de Londres pontuação	0,048 (-0,287)*	0,017 (-0,342)*	0,017 (-0,343)*
Torre de Londres tentativas	0,056 (0,277)*	0,030 (0,313)*	0,029 (0,314)*
BCST acerto	0,036 (-0,304)*	0,262 (-0,165)	0,165 (-0,204)
BCST erro	0,041 (0,297)*	0,160 (0,206)	0,073 (0,261)

\* Houve correlação

Crianças que apresentaram mais sintomas na escala SDQ como um todo (pontuação total), apresentaram também menor desempenho no Raven ( $p=0,017$ ,  $r=0,343$ ), atingiram menor nível

( $p=0,022$ ,  $r=-0,330$ ) fizeram menos acertos ( $p=0,027$ ,  $r=-0,318$ ) no Teste de Dígitos direto, no Stroop cometeram mais erros ( $p=0,012$ ,  $r=0,373$ ) e erros aleatórios ( $p=0,028$ ,  $r=0,328$ ) e apresentaram um desempenho pior na Torre de Londres (pontuação:  $p=0,017$ ,  $r= -0,343$ ; tentativas:  $p=0,029$ ,  $r=0,314$ ). (TAB.7)

O desempenho pior na Torre de Londres também está associado a sintomas de hiperatividade (pontuação:  $p=0,017$ ,  $r= -0,342$ ; tentativas:  $p=0,030$ ,  $r=0,313$ ) e a problemas de conduta (pontuação:  $p=0,048$ ,  $r=-0,287$ ; tentativas:  $p=0,056$ ,  $r=0,277$ ) avaliados pelo SDQ. Ainda, mais problemas de conduta também estão correlacionados a um desempenho pior no BCST considerando-se os índices de acerto ( $p=0,036$ ,  $r=-0,304$ ) e erro ( $p=0,041$ ,  $r= 0,297$ ), além de pior desempenho nas duas versões do Teste de Dígitos (nível de dígitos direto:  $p=0,052$ ,  $r= -0,283$ ; respostas correta inverso:  $p=0,031$ ,  $r=-0,312$ ). (TAB. 7)

Outras correlações não foram observadas, podendo ser conferidas nas tabelas do Anexo I.

# Discussão

## 5 DISCUSSÃO

O presente trabalho é um recorte de uma pesquisa que vem sendo realizada em escolas de Belo Horizonte e Nova Lima, além do AMBDA e crianças de clínicas particulares com diagnóstico de TDAH. Esta pesquisa visa ampliar e melhorar os testes de avaliação neuropsicológica para crianças e adolescentes no Brasil, tendo em vista a carência de testes validados e de fácil acesso para esta população.

Dentre os dados fornecidos pelos pais das crianças integrantes da pesquisa estão os questionários SNAP-IV e SDQ, amplamente utilizados como uma das fontes de informação para determinação do diagnóstico de TDAH. Assim, conforme critérios do DSM-IV, a amostra disponível para a presente análise foi composta por 50 crianças, sendo que 54% dessas preencheram critérios diagnósticos para TDAH. Analisando os dados dos grupos com e sem TDAH, observou-se que crianças com TDAH pontuaram mais tanto no SNAP-IV quanto na maioria das subescalas do SDQ. A única subescala em que ambos os grupos não apresentaram pontuação significativa e não houve diferença significativa entre eles foi comportamento pró-social.

As crianças do grupo de TDAH eram significativamente mais velhas. Além disso, quanto mais velhas, mais sintomas de desatenção e hiperatividade apresentaram. A relação com a desatenção e idade está compatível com a literatura, mas o mesmo não ocorre com a hiperatividade. Segundo estudo realizado com gêmeos por Larsson *et al.* (2011) a tendência é os sintomas de desatenção aumentarem proporcionalmente com a idade e os sintomas de hiperatividade diminuir. O resultado observado em nosso estudo, porém, pode estar associado com o fato das crianças do grupo com TDAH serem mais velhas.

Analisando os resultados, podemos observar que quando consideramos o diagnóstico, observa-se que o desempenho das crianças com TDAH apresentou diferenças significativas nos testes Donkey, BCST e Torre de Londres. De acordo com as comparações apresentadas, crianças com

TDAH tiveram um desempenho pior no BCST e Torre de Londres. Já ao considerarmos a pontuação no SNAP-IV, pode-se observar também melhor desempenho das crianças sem TDAH no Raven, Stroop e, tal como na análise com diagnóstico, na Torre de Londres. A maioria dos participantes, ainda, obteve desempenho na média ou acima da média no Raven (84%), mas crianças que obtiveram maior pontuação no SNAP-IV tiveram um desempenho mais baixo neste teste. Diferenças entre o desempenho de crianças com diagnóstico de TDAH e grupos controles em testes neuropsicológicos foram encontradas em outros estudos (BIEDERMAN, J. *et al.* 2004; CHAMBERLAIN, S.R. *et al.*, 2011; GUALTIERI, C. T., & JOHNSON, L. G., 2006; HONG *et al.*, 2010; LAMBEK *et al.*, 2010; SEIDMAN, L.J., 2006, dentre outros).

No teste BCST, que avalia principalmente flexibilidade cognitiva<sup>14</sup>, observamos que crianças com TDAH tiveram um desempenho pior nesse teste considerando-se o número de acertos, erros, erros não-perseverativos e nível conceitual. Para Hong *et al.* (2010), o baixo desempenho no BCST é comumente mencionado em pesquisas, sendo que tanto em pesquisas como em práticas clínicas é um dos testes mais mencionados como parte da bateria de avaliação. Ainda, os mesmos autores argumentaram em sua pesquisa que o baixo desempenho no BCST apresentado por crianças com TDAH, entre 7 a 9 anos, explicava-se melhor associando-se este dado com resultados de testes de memória de trabalho verbal e visual.

O Teste de Classificação de Cartas de Berg (BCST) do programa PEBL é uma versão similar a outro teste mais conhecido, o WCST, e é baseado nos procedimentos descritos por Berg em 1948. Segundo Fox *et al.* (2013), tais procedimentos mensuram funções executivas e podem ser usados para identificar prejuízos causados por danos ou transtornos cerebrais. Heaton *et al.* (2005) especifica que a avaliação abrange funções cognitivas superiores que estão relacionadas com os lobos frontais. Assim, um bom desempenho no teste WCST envolve planejamento, flexibilidade de pensamento, memória de trabalho, monitoração e inibição de respostas impulsivas (MALLOY-DINIZ *et al.*, 2008a). Quando o sujeito obtém um resultado baixo significa que pode haver uma disfunção executiva (SHAN *et al.*, 2008).

---

<sup>14</sup> O BCST avalia flexibilidade cognitiva, mas também utiliza processos como atenção, memória de trabalho e planejamento (HEATON, R. K. *et al.*, 2005).

De acordo com Reppold *et al.*, “a abrangência e a complexidade deste teste explicam por que o WCST foi convertido em uma prova por muitas vezes considerada padrão-ouro dentre os instrumentos destinados a avaliar funções executivas” (p.48, 2010b). Há muitas pesquisas comparando versões do WCST, com similaridade entre versão manual e computadorizada ou entre versão completa (128 cartas) e curta (64 cartas). Já o teste BCST possui ainda poucos estudos (FOX *et al.*, 2013).

Já em relação à Torre de Londres, observou-se que crianças com TDAH apresentaram um desempenho pior, tendo em vista que obtiveram menor pontuação. Ainda, quanto mais sintomas de desatenção (SNAP-IV), pior o desempenho neste teste. Um número maior de movimentos para resolver os problemas sugere algum prejuízo em córtex pré-frontal dorsolateral e núcleos da base, áreas que mostram ativadas em tarefas que exigem habilidades de planejamento e resolução de problemas (BAKER *et al.*, 1996). O teste da Torre de Londres (ToL), portanto, é amplamente utilizado para avaliar planejamento e resolução de problemas. Foi desenvolvido inicialmente na década de 80 por Shallice e colaboradores e desde então se tornou importante na prática clínica e na pesquisa para investigação de comprometimentos relacionados aos circuitos pré-frontais adquiridos ou decorrentes de distúrbios neuropsiquiátricos (SCHLOTTFELDT *et al.*, 2010).

Yáñez-Téllez *et al.* (2012) desenvolveram estudo descrevendo o funcionamento cognitivo em crianças com TDAH tipo hiperativo e combinado, sendo 26 meninos com TDAH, idade entre 7 e 12 anos (média 8,6 anos), e grupo controle composto por 25 meninos e idade média de 9 anos. Considerando a avaliação de funções executivas, utilizaram testes similares ao nosso estudo como Torre de Londres, Stroop Color Word Test (versão utilizando cores) e WCST (similar ao BCST). Considerando esses três testes, diferentemente de nosso estudo, os autores encontraram correlações significativas apenas com o teste WCST (total de erros, erros perseverativos e respostas perseverativas). Para os autores, tal desempenho indica que o déficit em funções executivas não explica os problemas de hiperatividade e impulsividade em crianças. Porém, a flexibilidade cognitiva (avaliada no BCST) pode ser um fator significativo para essa deficiência. Uma das diferenças entre nosso estudo e o descrito acima está na versão utilizada do Stroop e por isso pode ter ocorrido esta diferença na significância dos dados. Outra hipótese

é que o estudo de Yáñez- Téllez *et al.* (2012) referia-se a crianças com diagnóstico de TDAH com sintomas de hiperatividade. Em nosso estudo, as crianças com sintomas de desatenção é quem obtiveram pior desempenho no Stroop. Neste sentido, é possível que um mal desempenho neste teste esteja mais associado ao sintoma de desatenção.

Segundo Hong *et al.* (2010), em crianças, o “efeito Stroop” é mínimo quando elas estão começando a aprender a ler. Essa interferência no teste reduz ao passo que a leitura se torna automatizada. Por esta razão, ao observarmos que as crianças com diagnóstico de TDAH ainda não haviam desenvolvido a habilidade de leitura automatizada optou-se em aplicar a versão numérica do Stroop. Conforme observado, é possível que a opção por esta versão tenha reduzido a interferência do fator leitura e restringido a avaliação a outros aspectos, como atenção e controle inibitório. Nos dados obtidos, pode-se observar que crianças com maior pontuação em desatenção cometeram mais erros do que as demais neste teste. Considerando-se essa análise, é possível que crianças com TDAH apresentem desempenho mais baixo nesta versão do teste se Stroop.

Já em relação ao teste Donkey, em nossa pesquisa observou-se uma tendência a crianças com TDAH apresentarem um desempenho discretamente melhor no Donkey (total de maçãs). Lambek *et al.* (2010) utilizou este teste para avaliar tomada de decisão entre grupos TDAH com DFE, TDAH sem DFE e grupo controle sem DFE. Os participantes totalizaram 48 crianças com TDAH (divididas em TDAH com DFE e TDAH sem DFE) e 26 crianças do grupo controle. Apesar da versão de Lambek *et al.* ter um total de 200 tentativas, ou seja, 50 a mais do que nossa versão (total 150), não encontraram diferenças significativas entre os grupos de TDAH e controle. Ainda, questionaram as crianças quanto à compreensão do teste: 3 crianças declararam não entender o teste, 28 entenderam mas optaram pela opção mais desvantajosa e as demais 43 crianças formularam hipóteses sobre portas mais vantajosas e desvantajosas, mas sem interferir em seu desempenho no teste.

Corroborando este achado de Lambek *et al.* (2010), encontramos poucas correlações com o teste Donkey, conforme mencionamos acima. Crianças com diagnóstico de TDAH selecionaram menos a porta A e fizeram uma pontuação total melhor. Esses índices implicariam

que crianças com TDAH fizeram melhores escolhas do que crianças sem TDAH. Porém, não se observou nenhuma outra correlação nas demais análises que fortalecesse esta informação atípica. Além disso, apesar do desempenho significativamente melhor na pontuação total, ainda assim a média é apenas 3, o que demonstra um número muito baixo de maçãs e uma possível predileção por escolhas desvantajosas. Crone e Molen (2007), em seu estudo comparando desempenho no Donkey em faixa-etárias distintas, afirmaram que crianças ente 8-10 e 12-14 anos apresentam um desempenho ruim neste teste, pois falham em prever resultados antes de tomar uma decisão.

Já Carlson *et al.* (2009), apesar de não ter obtido dados significativos quanto a escolha de portas mais vantajosas do que desvantajosas em seu estudo com crianças de 8 anos, obtiveram uma curva de aprendizagem indicando que no decorrer do teste as crianças passaram a selecionar mais as portas vantajosas do que desvantajosas. Os autores justificaram tal desempenho pelo aumento no número de tentativas do teste para 280. Tais dados reforçam a necessidade de mais estudos e refinamento do teste Donkey, inclusive como fator diagnóstico para TDAH.

Dentre as limitações do presente estudo podemos citar que além de encontrar diferenças significativas para os testes BCST e Torre de Londres, a escala SDQ (principalmente subescala total) também apresentou correlações com a maioria das variáveis utilizadas na presente análise<sup>15</sup>. Quando nos referimos ao SDQ total, estamos tratando da soma total da pontuação que o indivíduo obteve. Conforme observado nos resultados, 81,3% (n=39) das crianças apresentaram um SDQ total significativo. As outras subescalas tiveram a seguinte porcentagem no item anormal: sintomas emocionais (56,3%, n=27), problemas de conduta (41,7%, n=20), problemas de relacionamento com os colegas (33,3%, n=16), hiperatividade (47,9%, n=23), este último utilizado nas análises. A única subescala que apresentou um valor menor no item anormal foi comportamento pró-social, onde apenas 8,3% tiveram significância (n=4). Yáñez-Téllez, G *et al.* (2012), consideraram em seus estudos a possibilidade dos prejuízos encontrados em diferentes estudos de TDAH estarem associados a comorbidades, como o distúrbio de aprendizagem. Desta forma, concluímos que se fazem necessárias outras análises dos dados,

---

<sup>15</sup> Idade, Critério socioeconômico, Raven, Teste de Dígitos, Stroop, Torre de Londres.

considerando as demais subescalas do SDQ para afirmar que os dados não sofreram influência de outros transtornos.

Outro ponto que pode ter interferido nos dados foi a seleção ocorrida nas escolas, principalmente de Nova Lima. Apesar dos esclarecimentos realizados quanto à importância de uma amostra aleatória, observou-se no decorrer da coleta que a escola da Tarde (n=3) selecionou os alunos com mais dificuldade nas salas para a avaliação. E a escola da manhã (n=16), selecionou a turma que apresenta mais dificuldades. Assim, uma amostra mais abrangente e aleatória poderá demonstrar se este fator da seleção interferiu nos resultados desta pesquisa.

Conclusão

## 6 CONCLUSÃO

Conforme mencionado na discussão, o presente trabalho é um recorte de uma pesquisa mais ampla que engloba outras variáveis. Dentre estas, pode-se destacar os questionários de uso de vídeo-game e habilidades no uso de computador e internet, além de dados da presença de possíveis outros transtornos. Ainda, alguns índices dos testes também foram omitidos como tempo de execução e índices detalhados dos testes, como pontuação em cada problema da Torre de Londres ou precisão da resposta no Stroop. Outro índice omitido foi escala elaborada para avaliar alguns sintomas de ansiedade (suor, temperatura, conforto) e motivação para realização dos testes. O recorte fez-se necessário pela amplitude de dados gerados.

Dentre os dados utilizados para a análise do presente estudo, pode-se observar que:

- Crianças com TDAH apresentaram um desempenho pior em testes que avaliam flexibilidade cognitiva, planejamento, controle inibitório e atenção. Porém não se observou diferença significativa quanto à memória de trabalho e tomada de decisão.
- Observaram-se diferenças relacionadas ao SNAP-IV, principalmente no Donkey, Stroop e Torre de Londres. Nestes dois últimos testes, as crianças com sintomas de desatenção apresentaram desempenho pior. Já no Donkey, no índice “total de maçãs”, obtiveram resultado melhor.
- No questionário SDQ observaram-se diferenças na maioria das variáveis, principalmente considerando a pontuação geral. Na subescala hiperatividade observou-se desempenho pior da Torre de Londres. O mesmo ocorreu na subescala de problemas de conduta, que apresentou também diferenças no Teste de Dígitos e BCST.
- Não se observou correlações significativas do Teste Raven com os testes neuropsicológicos. Porém crianças com maior pontuação no SNAP-IV tiveram um desempenho pior neste teste.
- Crianças mais velhas pontuaram mais na escala SDQ e SNAP-IV, porém o grupo de TDAH tinha crianças significativamente mais velhas.
- A pontuação no SDQ também foi influenciada pelo critério socioeconômico, sendo que pontuações mais elevadas estão associadas a classe mais baixas.

Esta pesquisa visa ampliar e melhorar os testes de avaliação neuropsicológica para crianças e adolescentes no Brasil, tendo em vista a carência de testes validados e de fácil acesso para esta população. A continuação deste estudo, portanto, melhorando os testes, ampliando o tamanho amostral e faixa-etária, poderão gerar dados relevantes para o estudo da avaliação neuropsicológica.

# Referências

## Referências bibliográficas

- American Academy of Pediatrics. ADHD: Clinical Practice Guideline for the Diagnosis, Evaluation, and Treatment of Attention-Deficit/Hyperactivity Disorder in Children and Adolescents. *Pediatrics*, 2011. Disponível em: <http://pediatrics.aappublications.org/content/early/2011/10/14/peds.2011-2654>
- ANSELMINI, L. *et al.* Prevalence of psychiatric disorders in a Brazilian birth cohort of 11-year-olds. *Soc. Psychiatr Epidemiol*, v. 45, p. 135-142, 2010.
- BAKER, S.C. *et al.* Neural systems engaged by planning: a PET study of the Tower of London task. *Neuropsychologia*, v. 34, p. 515-526, 1996.
- BARBOSA, A.F. (coord). Pesquisa sobre o Uso das Tecnologias da Informação e da Comunicação no Brasil: TIC Domicílios e TIC Empresas 2009. São Paulo: Comitê Gestor da Internet no Brasil, 2010
- BAUER, R.M. *et al.* Computerized Neuropsychological Assessment Devices: Joint Position Paper of the American Academy of Clinical Neuropsychology and the National Academy of Neuropsychology. *Archives of Neuropsychology*, v. 27 p. 362-373, 2012.
- BIEDERMAN, J. *et al.*, Impact of Executive Function Deficits and Attention-Deficit/Hyperactivity Disorder (ADHD) on Academic outcomes in children. *Journal of Consulting and Clinical Psychology*, v. 72, n. 5, p.757-766, 2004.
- CAPOVILLA, A. Desenvolvimento e validação de instrumentos neuropsicológicos para avaliar funções executivas. *Avaliação Psicológica*; v. 5, n. 2, p. 239-241, 2006.
- CARLSON, S.M.; ZAYAS, V.; GUTHORMSEN, A. Neural Correlates of Decision Making on a Gambling Task. *Child Development*, v. 80, n. 4, p. 1076-1096, 2009.
- CHAMBERLAIN, S. R. *et al.* Translational approaches to frontostriatal dysfunction in attention-deficit/hyperactivity disorder using a computerized neuropsychological battery. *Biological Psychiatry*, v. 69 n. 12, p.1192–1203, 2011.
- COELHO, L. *et al.* Transtorno de Déficit de Atenção e Hiperatividade (TDAH) na Criança - aspectos neurobiológicos, diagnóstico e conduta terapêutica. *Acta Med Port*, v 23, p. 689-696, 2010.
- CRONE, E.; MOLEN, M. Development of Decision Making in School- Aged Children and Adolescents: Evidence from heart rate and skin conductance analysis. *Child Development*, v. 78, n. 4, p. 1288-1301, 2007.
- CURY, C. R.; GOLFETO, J. H. Strengths and difficulties questionnaire (SDQ): a study of school children in Ribeirão Preto. *Rev. Bras. Psiquiatria*, São Paulo, v. 25, n.3, 2003. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1516-44462003000300005&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1516-44462003000300005&lng=en&nrm=iso)>. Acesso em 19 de Julho de 2013.
- DSM-IV - Manual diagnóstico e estatístico de transtornos mentais. Porto Alegre, Ed. Artes Médicas, 1995.
- FONTANA, R.S. *et al.* Prevalência de TDAH em quatro escolas públicas brasileiras. *Arq. Neuropsiquiatria*, v. 65 n. 1, p. 134-137, 2007.
- FOX, C.J. *et al.* Evaluation of a Short-Form of the Berg Card Sorting Test. *PLoS ONE*, v.8, n. 5, 2013.
- GREVE, K.W. The WCST-64: a standardized short-form of the Wisconsin Cards Sorting Test. *The Clinical Neuropsychologist*, v. 15 n. 2, p. 228-234, 2001.
- GUALTIERI, C. T.; JOHNSON, L. G. Efficient allocation of attentional resources in patients with ADHD: Maturational changes from age 10 to 29. *Journal of Attention Disorders*, v.9 n. 3, p. 534–542, 2006.
- HEATON, R.K. *et al.* Teste Wisconsin de Classificação de Cartas. São Paulo: Casa do Psicólogo, 2005.

HESTER, R.L.; KINSELLA, G.J.; ONG, B.. Effect of age on forward and backward span tasks. *Journal of the International Neuropsychological Society*, v.10, pp 475-481, 2004.

HONG, H.J *et al.* Impairment of concept formation ability in children with ADHD: comparisons between lower grades and higher grade. *Psychiatry Investigation*, v. 7, n. 3, p.177-188 2010.

KNOUSE, L.E.; BARKLEY, R.A.; MURPHY, K.R. Does executive functioning (EF) predict depression in clinic-referred adults?: EF tests vs. rating scales. *Journal of Affective Disorders*, v. 145, p. 270-275, 2013.

KRIKORIAN, R. *et al.* Tower of London procedure: A standard method and developmental data. *Journal of Clinical and Experimental Neuropsychology*, v.16, n. 6, p.840-850, 1994.

LAMBEK, R. *et al.* Validating neuropsychological subtypes of ADHD: how do children with and without and executive function deficit differ? *Journal of Child Psychology and Psychiatry*, v. 51, n. 8, p. 895-904, 2010.

LARSSON,H. *et al.* Developmental trajectories od DSM-IV symptoms of attention-deficit/hyperactivity disorder: genetic effects, family risk and associated psychopathology. *The Journal of Child Psychology and Psychiatry*, v.52, n. 9, p. 954-963, 2011.

MALLOY-DINIZ, L.F. *et al.* Neuropsicologia no Transtorno de Déficit de Atenção e Hiperatividade. In: FUENTES, D. *et al.* Neuropsicologia: teoria e prática. Porto Alegre: Artmed, cap 14, p. 187-206, 2008a.

MALLOY-DINIZ, L.F. *et al.* Neuropsicologia das funções executivas. In: FUENTES, D. *et al.* Neuropsicologia: teoria e prática. Porto Alegre: Artmed, cap 11, p. 187-206, 2008b.

MATTOS, P. Entenda o TDAH nos critérios do DSM-V: O que mudou no diagnóstico do TDAH com a nova edição do DSM-V, o Manual de Estatística e Diagnóstico de Transtornos Mentais. *Associação Brasileira do Déficit de Atenção*. 03 de Julho de 2013. Disponível em: <http://www.tdah.org.br/br/textos/textos/item/964-entenda-o-tdah-nos-crit%C3%A9rios-do-dsm-v.html>.

MATTOS, P. *et al.* Apresentação de uma versão em português para uso no Brasil do instrumento MTA-SNAP-IV de avaliação de sintomas de transtorno do déficit de atenção/hiperatividade e sintomas de transtorno desafiador e de oposição. *Rev. psiquiatr. Rio Gd. Sul, PortoAlegre*, v. 28, n.3, 2006. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0101-81082006000300008&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-81082006000300008&lng=en&nrm=iso)>. Acesso em 19 de Julho de 2013.

MUELLER, S. T. The PEBL Manual Programming and Usage Guide for The Psychology Experiment Building Language - PEBL (Version 0.12), 2011. Disponível em: <<http://pebl.sourceforge.net>>

POLANCZYK, G.; ROHDE L.A. Epidemiology of attention-deficit/hyperactivity disorder across the lifespan. *Current Opinion in Psychiatry*, v. 20, n. 4, p. 386-392, 2007.

REPPOLD, C.T.; PEDRON, A.C. Avaliação de mecanismos inibitórios através de tarefa computadorizada de paradigma go/no-go e tarefas que medem o tempo de reação com interrupção sinalizada (Stop-Signal Reaction-Time- SSRT). IN: JOLY, M.C.R.A.; REPPOLD, C.T.(org.). Estudos de testes informatizados: para avaliação psicológica. Casapsi Livraria e Editora Ltda: São Paulo, cap. 4, p. 85-102, 2010a.

REPPOLD, C.T.; PEDRON, A.C.; TRENTINI, C.M. Avaliação das funções executivas por meio do Teste Wisconsin de Classificação de Cartas- versão computadorizada. IN: JOLY, M.C.R.A.; REPPOLD, C.T.(org.). Estudos de testes informatizados: para avaliação psicológica. Casapsi Livraria e Editora Ltda: São Paulo, cap. 2, p. 45-62, 2010b.

ROBINSON, L.J. *et al.* Comparison of two short forms of the Wisconsin Card Sorting Test. *Archives of Clinical Neuropsychology*, v. 6, n. 1-2, p. 27-33, 1991.

ROHDE, L.A. *et al.* Attention deficit-hyperactivity disorder in a diverse culture: do research and clinical findings support the notion of a cultural construct for the disorder? *Biol Psychiatry*, v. 57, p. 1436-1441, 2005.

SCHLOTTFELDT, C.G. *et al.* Teste da Torre de Londres (TOL). In: Malloy-Diniz L.F. *et al.* (org.), Avaliação Neuropsicológica. Porto Alegre: Artmed, 2010.

- SEIDMAN, L.J. Neuropsychological functioning in people with ADHD across the life span. *Clin.Psychol.Rev.*, v. 26, p. 466-485, 2006.
- SHAN, K. *et al.* Adult normative data of the Wisconsin Card Sorting Test in Taiwan. *JournaloftheChinese Medical Association*, v.71, n. 10, p. 517-522, 2008.
- SMITH-SEEMILLER, L.;ARFFA, S.;FRANZEN, M.D. Use of Wisconsin Card Sorting Test short forms with school-age children. *Archives or Clinical Neuropsychology*, v.16, P. 489-499, 2001.
- SOUZA, R. O. *et al.* Contribuição à neuropsicologia do comportamento executivo. Torre de Londres e teste de Wisconsin em indivíduos normais. *Arquivos de Neuropsiquiatria*, v.59, n. 3-A, p.526-531, 2001.
- STRAUSS, E.; SHERMAN, E.M.S.; SPREEN, O. A compendium of neuropsychological tests: administration, norms, and commentary. 3º Ed. *Oxford University Press*: New York, 2006.
- TIEN, A. *et al.* Computerized Wisconsin Card Sorting Test: comparison with manual administration. *The Kaohsiun Journal of Medical Sciences*, v.12, p. 479-485, 1996.
- TRENTINI, C.M. *et al.* O desenvolvimento de normas para o Teste de Wisconsin de Classificação de Cartas (em andamento). *Avaliação Psicológica*, v. 5, n. 2, p. 247-250, 2006.
- YÁÑEZ-TÉLLEZ, G. *et al.* Cognitive and executive functions in ADHD. *Acta Esp Psiquiatr*, v.40, n. 6, p. 293-298, 2012.

# Apêndices

## APÊNDICE A

### Critérios Diagnósticos para TDAH segundo o DSM-IV (DSM-IV, 1995; Malloy-Diniz et al., 2008)

A. Ou (1) ou (2)

1) seis (ou mais) dos sintomas de desatenção persistiram por pelo menos 6 meses, em grau mal adaptativo e inconsistente com o nível de desenvolvimento:

2) seis (ou mais) dos sintomas de hiperatividade persistiram por pelo menos 6 meses, em grau mal adaptativo e inconsistente com o nível de desenvolvimento:

#### Desatenção:

(a) frequentemente deixa de prestar atenção a detalhes ou comete erros por descuido em atividades escolares, de trabalho ou outras

(b) com frequência tem dificuldades para manter a atenção em tarefas ou atividades lúdicas

(c) com frequência parece não escutar quando lhe dirigem a palavra

(d) com frequência não segue instruções e não termina seus deveres escolares, tarefas domésticas ou deveres profissionais (não devido a comportamento de oposição ou incapacidade de compreender instruções)

(e) com frequência tem dificuldade para organizar tarefas e atividades

(f) com frequência evita, antipatiza ou reluta a envolver-se em tarefas que exijam esforço mental constante (como tarefas escolares ou deveres de casa)

(g) com frequência perde coisas necessárias para tarefas ou atividades (por ex., brinquedos, tarefas escolares, lápis, livros ou outros materiais)

(h) é facilmente distraído por estímulos alheios à tarefa

(i) com frequência apresenta esquecimento em atividades diárias

#### Hiperatividade:

(a) frequentemente agita as mãos ou os pés ou se remexe na cadeira

(b) frequentemente abandona sua cadeira em sala de aula ou outras situações nas quais se espera que permaneça sentado

(c) frequentemente corre ou escala em demasia, em situações nas quais isto é inapropriado (em adolescentes e adultos, pode estar limitado a sensações subjetivas de inquietação)

(d) com frequência tem dificuldade para brincar ou se envolver silenciosamente em atividades de lazer

(e) está frequentemente "a mil" ou muitas vezes age como se estivesse "a todo vapor"

(f) frequentemente fala em demasia

#### Impulsividade:

(g) frequentemente dá respostas precipitadas antes de as perguntas terem sido completadas

(h) com frequência tem dificuldade para aguardar sua vez

(i) frequentemente interrompe ou se mete em assuntos de outros (por ex., intromete-se em conversas ou brincadeiras)

B. Alguns sintomas de hiperatividade-impulsividade ou desatenção que causaram prejuízo estavam presentes antes dos 7 anos de idade.

C. Algum prejuízo causado pelos sintomas está presente em dois ou mais contextos (por ex., na escola [ou trabalho] e em casa).

D. Deve haver claras evidências de prejuízo clinicamente significativo no funcionamento social, acadêmico ou ocupacional.

E. Os sintomas não ocorrem exclusivamente durante o curso de um Transtorno Invasivo do Desenvolvimento, Esquizofrenia ou outro Transtorno Psicótico e não são melhor explicados por outro transtorno mental (por ex., Transtorno do Humor, Transtorno de Ansiedade, Transtorno Dissociativo ou um Transtorno da Personalidade).

Codificar com base no tipo:

- Transtorno de Déficit de Atenção/Hiperatividade, Tipo Combinado: se tanto o Critério A1 quanto o Critério A2 são satisfeitos durante os últimos 6 meses.

- Transtorno de Déficit de Atenção/Hiperatividade, Tipo Predominantemente Desatento: Se o Critério A1 é satisfeito, mas o Critério A2 não é satisfeito durante os últimos 6 meses.

- Transtorno de Déficit de Atenção/Hiperatividade, Tipo Predominantemente Hiperativo-Impulsivo: Se o Critério A2 é satisfeito, mas o Critério A1 não é satisfeito durante os últimos 6 meses.

## APÊNDICE B

### **Definições de conceitos do Teste Wisconsin de Classificação de Cartas (WCST)**

Para conseguir um bom desempenho no teste, o indivíduo deve determinar qual é o princípio de classificação correto (ex: cor), com base no *feedback* fornecido pelo aplicador. Além disso, precisa manter este princípio, mesmo influenciado por outros distratores (ex: forma e número).

**Respostas correta-incorreta:** respostas que se associam com o princípio de classificação vigente (ex. cor) têm escore como **corretas**, enquanto respostas incorretas têm escore como **erros**.

**Respostas perseverativas e não-perseverativas:** quando o indivíduo persiste em responder a uma característica do estímulo incorreta. O percentual de erros perseverativos corresponde a concentração de erros perseverativos em relação ao desempenho total no teste.

**Números de categorias completadas:** é o número de categorias (cada dez respostas corretas) que o indivíduo conseguiu concluir durante o teste.

**Ensaio para completar a primeira categoria:** é o número de cartas (ensaio) que o indivíduo necessitou utilizar para completar a primeira categoria.

**Fracasso em manter o contexto:** é quando o indivíduo comete um erro após ter feito pelos menos cinco acertos consecutivos. Contabiliza-se o número de vezes em que o indivíduo fracassou.

**Nível conceitual:** é a soma das respostas corretas consecutivas ocorridas em curso de três ou mais em todo o teste. Contabilizam-se, assim, todas as respostas e não o número de vezes em que isto ocorreu, tal como ocorre no fracasso em manter o contexto.

**Aprendendo a aprender:** é calculado apenas se o indivíduo completar pelo menos 3 categorias. “Reflete a mudança média do cliente na eficiência conceitual, ao longo de categorias consecutivas do WCST” ( HEATON *et al.*, 2005).

Todos os conceitos definidos foram extraídos do Manual do Teste Wisconsin de Classificação de Cartas (Heaton, 2005).

Anexos

## ANEXO A-Programação Teste de Dígitos

```
#####
### Numerical Stroop Task
### (c) 2011 Shane T. Mueller and Alena Esposito
### smueller at obereed dot net
### PEBL Numerical Stroop Task, Version 0.1
### Modeled, roughly on Hernandez et al. (2010).
### Spanish translation by Alena Esposito
### Here is a description of the numerical Stroop directly from Hernandez
### et al. (2010). Participants were asked to indicate, as fast
### and accurately as possible, how many items appeared in each
### trial. The number of items ranged from 1 to 3, and participants had
### to press the keys 1, 2 or 3 on the keyboard with the index, middle
### and ring fingers of their dominant hand, respectively. There was a
### Distractor variable with three conditions:
###
### (a) alphabetic characters (neutral condition: e.g., Z, GGG, MM);
### (b) digits whose value matched the number of items
### (congruent condition: 1, 22, 333); and
### (c) digits whose value did not
### match the number of items (incongruent condition: e.g., 2, 33, 111,
### etc.).
###
### The experiment consisted of two blocks of 84 trials each
### (preceded by a training block of 24 trials). The three distractor
### conditions were represented the same number of times in each block in
### a random fashion. On each trial, a central fixation cross appeared for
### 1000 ms, immediately followed by the target, which was presented for
### 2000 ms or until a response was given. I believe the training block
### should tell the participant whether or not they are correct. I am
### also not concerned about having 2 separate blocks. One block with a
### practice of 24 and then 100 test trials should work. The trials need
### to be in the same order for each participant.

define Start(p)
{
  ##
  ## Options:
  ## Set gResponseType to either "keyboard" or "audio" to change the
  ## response method
  ##
  ## gResponseType <- "keyboard"
  ## gResponseType <- "audio"
  ## gResponseType <- "keypad"

  Initialize()

  gResponseType <- GetEasyChoice("Select input type ",
    ["1-2-3 Number Line on keyboard",
     "1-2-3 numbers on keypad",
     "Vocal input from microphone"],
    ["keyboard", "keypad", "audio"], gWin)

  ## for audio, set this to 1 to score real-time with keyboard.
  ## otherwise, you need to go back through and code responses afterward.
  gScoreRealtime <- 0

  ## The voice key will trip when 55% of the last
  ## gVoiceKeyDuration 1-ms bins are above the threshold.
  ## Based on input power and microphone noise, you may
  ## need to adjust these parameters.
  gVoiceKeyThreshold <- .35
  gVoiceKeyDuration <- 200
  ## Should the audio buffer be saved?
  gSaveAudio <- 1
  ## command to compress audio
  gCompressAudio <- 1
  gCompressor <- "oggenc "
  ## If set to 1, it will use the same sequence of stimuli every time.
  ## If set to 0, it will regenerate randomly on each run.
  usePreset <- 1
  gFixationTime <- 1000 #time for the fixation cross
  gResponseTimeout <- 2000 #Time limit given for responding
  ## Set gLanguage here to hard code
  gLanguage <- "BR"
}
```

```

GetStrings(gLanguage)
##Enter the subject code here if one wasn't already provided.
gSubNum <- GetSubNum(gWin)
if(gSubNum == 0)
{
  gSubNum <- GetSubNum(gWin)
}
MakeDirectory("data/"+gSubNum)
gFileOut <- FileOpenWrite("data/"+gSubnum+"/numstroop-"+gResponseType+"-"+gSubNum+".csv")
gPooledFileOut <- FileOpenAppend("data/numstroop-"+gResponseType+"-pooled.csv")
gSumFileOut <- FileOpenWrite("data/" + gSubNum + "/numstroop-"+gResponseType+"-summary-" + gSubNum+".txt")
gLog <- FileOpenAppend("data/nstroop-log.csv")
##Add headers onto files.
flen <- Length(FileReadList("data/numstroop-"+gResponseType+"-pooled.csv"))
if(flen==0)
{
  Print("empty pooled data file. Adding header")
  FilePrint(gPooledFileOut,
"subnum,response,code,block,trial,stim,cond,responded,response,correct,abstime,rt,speechduration,recordtime")
}
FilePrint(gFileOut, "subnum,response,code,block,trial,stim,cond,responded,response,correct,abstime,rt,speechduration,recordtime")
FilePrint(gLog, gSubNum + ", " + gLanguage + ", " + TimeStamp()+",NUMERICAL STROOP,"+gResponseType+",STARTED")
##Create a vocal buffer if we have one
if(gResponseType == "audio")
{
  Print("Making audio input buffer")
  ##3000/3500 doesn't seem to work in linux:???
  gResponseBuffer <- MakeAudioInputBuffer(5000)
}
#####
## Create the basic stimuli
#####
##This, by default, reads in a fixed
block0 <- PracticeTrials(usepreset)
block1 <- Round1Trials(usepreset)
block2 <- Round2Trials(usepreset)
#####
##Show the instructions.
gInstructions.text <- gInstructionsText
Show(gInstructions)
Hide(gHeader)
Draw()
instru001 <- LoadSound("instru001.wav")
PlayBackground(instru001)
Wait(41500)
Stop(instru001)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
gHeader.text <- gPracticeHeaderText
Show(gHeader)
Draw()
#Initialize data collectors
gErr <- []; gRT <- []; gCond <- []
gErrors <- 0 ##All errors
gCongruenceErrors <- 0 ##Errors indicating using the wrong dimension
gRandomErrors <- 0 ##Errors not attributable to incongruency
gHeader.text <- gPracticeHeaderText
##Do the practice round
##
gTrial <- 0
gBlock <- 0
loop(trial,block0)
{
  DoTrial(trial,gResponseType,"data/"+gSubNum+"/trial-"+ZeroPad(gTrial,3)+"-")
  gTrial <- gTrial + 1
}
#ReInitialize data collectors after practice
gErr <- []; gRT <- []; gCond <- []
gErrors <- 0 ##All errors
gCongruentErrors <- 0 ##Errors indicating using the wrong dimension
gRandomErrors <- 0 ##Errors not attributable to incongruency
gInstructions.text <- gBreak1Text
Show(gInstructions)
Hide(gFooter1);Hide(gFooter2)

```

```

Draw()
instru002 <- LoadSound("instru002.wav")
PlayBackground(instru002)
Wait(9300)
Stop(instru002)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
gHeader.text <- gTestHeaderText
gBlock <- 1
##Do Round 1
##
loop(trial,block1)
{
  DoTrial(trial.gResponseType,"data/"+gSubNum+"/trial-"+ZeroPad(gTrial,3)+"-")
  gTrial <- gTrial + 1
}
gInstructions.text <- gBreak2Text
Show(gInstructions)
Hide(gFooter1);Hide(gFooter2)
Draw()
instru003 <- LoadSound("instru003.wav")
PlayBackground(instru003)
Wait(10100)
Stop(instru003)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
gBlock <- 2
##Do Round 2
##
loop(trial,block2)
{
  DoTrial(trial.gResponseType,"data/"+gSubNum+"/trial-"+ZeroPad(gTrial,3)+"-")
  gTrial <- gTrial + 1
}
##These compute: condition, length, median, mean, stdev
statsCorr <- SummaryStats(gErr,gCCond)
statsRT <- SummaryStats(gRT,gCCond)
FilePrint(gSumFileOut,"-----")
FilePrint(gSumFileOut,"PEBL Numerical Stroop Task, Version 1.0")
FilePrint(gSumFileOut,"http://pebl.sf.net")
FilePrint(gSumFileOut,"Shane T. Mueller")
FilePrint(gSumFileOut,GetPEBLVersion())
FilePrint(gSumFileOut,"-----")
FilePrint(gSumFileOut, "Subject code: "+gSubNum)
FilePrint(gSumFileOut, "Time: "+TimeStamp())
FilePrint(gSumFileOut, "Response condition: " + gResponseType)
FilePrint(gSumFileOut,"-----")
FilePrint(gSumFileOut,"Total Errors: "+gErrors)
FilePrint(gSumFileOut,"Incongruency Errors: "+gCongruenceErrors)
FilePrint(gSumFileOut,"Random Errors: "+gRandomErrors)
FilePrint(gSumFileOut,"-----"+CR(2))
FilePrint(gSumFileOut," [C]ongruent, [I]ncongruent, [N]eutral")
FilePrint(gSumFileOut," Response time")
FilePrint(gSumFileOut," Cond Mean N")
FilePrint(gSumFileOut,"-----")
loop(i,statsRT)
{
  FilePrint(gSumFileOut, " "+First(i)+ " "+Nth(i,4)+ " "+Second(i))
}
FilePrint(gSumFileOut," Accuracy")
FilePrint(gSumFileOut," Cond Mean N")
FilePrint(gSumFileOut,"-----")
loop(i,statsCorr)
{
  FilePrint(gSumFileOut, " "+First(i)+ " "+Nth(i,4)+ " "+Second(i))
}
FilePrint(gSumFileOut,"-----")
filesum <- FileOpenAppend("data/numstroop-sumall.csv")
flen <- Length(FileReadList("data/numstroop-sumall.csv"))
if(flen==0)
{
  FilePrint(filesum,"sub,timestamp,response,duration,errors,congruence_errors,random_errors,RTC,RTI,RTN,AccC,AccI,AccN")
}

```

```

}
FilePrint(filesum, gSubNum + "," + TimeStamp() + "," + gResponseType+", "+ GetTime()+", "+ gErrors+ ", "+ gCongruenceErrors+", "+
gRandomErrors)
loop(i,statsRT)
{
FilePrint_(filesum,", "+Nth(i,4))
}
loop(i,statsCorr)
{
FilePrint_(filesum,", "+Nth(i,4))
}
FilePrint(filesum,"")

FilePrint(gLog, gSubNum + "," + gLanguage + "," + TimeStamp()+", "+ "NUMERICAL STROOP," + gResponseType+", ENDED")
MessageBox(gExitText, gWin)
}

#####
## basic function for a single trial

define DoTrial(stim,cond,code)
{

length <- StringLength(Second(stim))
fname <- code + "-" + length + ".wav"

##Condition is the response type.
type <- First(stim)
gFooter2.text <- gFooter2Text
if(cond == "keyboard" or cond=="keypad")
{
gFooter1.text <- gFooter1textManual
Show(gFooter1)
Show(gFooter2)
}else{
gFooter1.text <- gFooter1textVocal
Show(gFooter1)
Show(gFooter2)
}

gStimLabel.text <- "+"
Show(gStimLabel)
Draw()
Wait(1000)
gStimLabel.text <- Second(stim)
Draw()
time1 <- GetTime()
if(cond == "keyboard" or cond == "keypad")
{
resp <- WaitForListKeyPressWithTimeout(gKeys,2000,1)
time2 <- GetTime()
if(IsList(resp)) ##Check to see if there was a time-out.
{
gStimLabel.text <- "Muito Lento"
Draw()
Wait(500)
responded <- 0
} else{
responded <- 1
}

corr <- ToNumber(resp) == length
rt <- (time2-time1)

out <- gSubNum + "," + cond + "," + code + "," + gBlock + "," + gTrial + "," +
Second(stim)+","+First(stim)+"," +responded+", "+ resp + "," +corr+", "+ time1 + "," +
rt+ ",0"+ "," + (time2-time1)

}elseif(cond=="audio")
{
Print("time1")
resp0 <- GetVocalResponseTime(gResponseBuffer,gVoiceKeyThreshold, gVoiceKeyDuration)
# Wait(300)
# resp0 <- [0,0,0,0]
time2 <- GetTime()

```

```

        Print(resp0)
    if(Third(resp0)==0)
    {
        gStimLabel.text <- "Muito Lento"
        Draw()
        Wait(500)
    }

    rt <- First(resp0)
    endtime <- Second(resp0)
    responded <- Third(resp0)
if( gScoreRealtime )
{
    gFooter1.text <- gFooter1textManual2
    Hide(gStimLabel)
    Draw()

    ##use the following for keypad entry:
    ## resp <-WaitForListKeyPress(["<KP_1>","<KP_2>","<KP_3>","<KP_4>"])
    resp <-WaitForListKeyPress([1,2,3,4])
        corr <- ToNumber(resp)==length
    } else {
        #Use -1 as the non-score indicator.
        resp <- -1
        corr <- -1
    }

        out <- gSubNum + "," + cond + "," + code + "," + gBlock + "," + gTrial + "," +
        Second(stim)+","+First(stim)+"," +responded+","+ resp + ","+corr+","+ time1 + ","+
        rt + ","+(endtime-rt) + "," +(time2-time1)
    } else{
SignalFatalError("Unknown response type: "+cond)
}
Hide(gStimLabel)
Draw()
if(cond=="audio" and gSaveAudio)
{
    SaveAudioToWaveFile(fname,gResponseBuffer)
        if(gCompressAudio)
        {
            SystemCall(gCompressor + fname)
                if(GetSystemType()=="LINUX" or GetSystemType()=="OSX")
                {
                    SystemCall("rm "+fname)
                } else {
                    SystemCall("del "+fname)
                }
        }
    }
    FilePrint(gFileOut,out)
    FilePrint(gPooledFileOut,out)
##Update some counters:
gErr <- Append(gErr,corr)
gRT <- Append(gRT, rt)
gCond<- Append(gCond,First(stim))
if(not corr)
{
    gErrors <- gErrors + 1
    if(resp == (SubString(Second(stim),1,1)))
    {
        gCongruenceErrors <- gCongruenceErrors + 1
    }else{
        gRandomErrors <- gRandomErrors + 1
    }
}
}
Wait(500)
}
##
##
define PrintMe(file,text)
{
    FilePrint(file,text)
    Print(text)
}

```

```

define PrintMe_(file,text)
{
  FilePrint_(file,text)
  Print_(text)
}
define PracticeTrials(usepreset)
{
  if(usePreset)
  {
    x <- [{"N", "G"}, {"C", "333"}, {"I", "3"}, {"N", "MMM"}, {"N", "GG"}, {"I", "111"}, {"I", "2"}, {"C", "22"},
      {"I", "33"}, {"I", "11"}, {"C", "1"}, {"I", "222"}, {"C", "1"}, {"N", "ZZ"}, {"I", "111"}, {"N", "ZZZ"},
      {"C", "333"}, {"C", "22"}, {"N", "MM"}, {"N", "GGG"}, {"N", "M"}, {"I", "222"}, {"C", "1"}, {"C", "333"}]
  }
  else {
    x <- GenerateStimuli(8)
  }
  return x
}
define Round1Trials(usepreset)
{
  if(usePreset)
  {
    x <- [{"C", "1"}, {"C", "333"}, {"I", "222"}, {"C", "22"}, {"C", "22"}, {"I", "222"}, {"I", "33"},
      {"N", "G"}, {"C", "1"}, {"N", "GG"}, {"C", "333"}, {"N", "ZZZ"}, {"C", "22"}, {"N", "ZZZ"}, {"I", "111"},
      {"I", "2"}, {"N", "G"}, {"I", "11"}, {"I", "2"}, {"N", "MMM"}, {"C", "333"}, {"I", "11"}, {"C", "1"},
      {"N", "ZZ"}, {"I", "3"}, {"N", "MM"}, {"N", "GGG"}, {"I", "33"}, {"I", "33"}, {"C", "22"}, {"C", "333"},
      {"C", "333"}, {"C", "333"}, {"C", "1"}, {"N", "M"}, {"N", "GGG"}, {"C", "1"}, {"I", "3"}, {"N", "ZZ"}, {"C", "22"},
      {"C", "333"}, {"C", "1"}, {"I", "222"}, {"C", "1"}, {"I", "33"}, {"I", "3"}, {"N", "MMM"}, {"I", "111"}, {"C", "22"},
      {"N", "M"}, {"C", "22"}, {"C", "22"}, {"N", "GGG"}, {"I", "2"}, {"I", "222"}, {"N", "ZZZ"}, {"I", "111"}, {"N", "M"},
      {"I", "3"}, {"N", "Z"}, {"N", "MM"}, {"N", "GG"}, {"N", "G"}, {"N", "M"}, {"N", "MM"}, {"N", "MMM"}, {"I", "2"}, {"N", "GG"},
      {"C", "22"}, {"N", "Z"}, {"I", "11"}, {"N", "ZZ"}, {"N", "Z"}, {"I", "111"}, {"C", "1"}, {"I", "222"}, {"I", "33"},
      {"I", "111"}, {"C", "1"}, {"C", "333"}, {"C", "333"}, {"I", "2"}, {"I", "11"}, {"C", "333"}]
  }
  else {
    x <- GenerateStimuli(28)
  }
  return x
}
define Round2Trials(usepreset)
{
  if(usePreset)
  {
    x <- [{"N", "MM"}, {"C", "22"}, {"N", "MM"}, {"C", "333"}, {"C", "333"}, {"N", "G"}, {"N", "GG"}, {"I", "3"}, {"C", "22"}, {"I", "3"}, {"I",
      "11"}, {"I", "111"}, {"I", "111"}, {"N", "G"}, {"N", "ZZZ"}, {"N", "GG"}, {"N", "ZZ"}, {"C", "22"}, {"N", "MMM"}, {"N", "MMM"}, {"N",
      "GGG"}, {"I", "3"}, {"I", "2"}, {"N", "ZZ"}, {"N", "MM"}, {"C", "333"}, {"N", "M"}, {"I", "222"}, {"C", "22"}, {"I", "111"}, {"C", "333"}, {"C",
      "333"}, {"C", "333"}, {"N", "GGG"}, {"N", "GGG"}, {"I", "222"}, {"N", "G"}, {"C", "1"}, {"I", "33"}, {"C", "1"}, {"I", "33"}, {"N", "MMM"}, {"N",
      "G"}, {"I", "222"}, {"N", "ZZZ"}, {"C", "1"}, {"I", "111"}, {"N", "M"}, {"C", "1"}, {"C", "333"}, {"I", "11"}, {"C", "1"}, {"C", "1"}, {"I", "3"}, {"I",
      "2"}, {"I", "2"}, {"I", "11"}, {"I", "2"}, {"N", "GG"}, {"N", "Z"}, {"I", "11"}, {"I", "111"}, {"I", "33"}, {"N", "ZZZ"}, {"N", "Z"}, {"C", "1"}, {"N",
      "ZZ"}, {"C", "333"}, {"I", "222"}, {"C", "22"}, {"I", "3"}, {"C", "22"}, {"C", "22"}, {"N", "M"}, {"N", "Z"}, {"C", "22"}, {"C", "1"}, {"I", "33"}, {"C",
      "22"}, {"C", "333"}, {"C", "22"}, {"I", "11"}, {"I", "33"}, {"C", "1"}]
  }
  else {
    x <- GenerateStimuli(28)
  }
  return x
}
##create a shuffled set of stimuli with
##num stimili from each of the three sets.
define GenerateStimuli(num)
{
  neutrals <- RepeatEnough(gNeutralStim,num)
  incongruents <- RepeatEnough(gIncongruentStim, num)
  congruents <- RepeatEnough(gCongruentStim, num)

  type <- Flatten([Repeat("N",num),Repeat("I",num),Repeat("C",num)])
  return Shuffle(Transpose([type,Flatten([neutrals,incongruents,congruents])))
}
##for now, assumes the tables are the same size
define PasteTables(tab1,tab2)
{
  a <- Transpose(tab1)

```

```

b <- Transpose(tab2)
return Transpose(Merge(a,b))
}
## This is a standard initializer function that sets up typical
## objects used in experiments.
define Initialize()
{

## Uncomment to eliminate busy-wait (mostly for older machines or
## laptops that might overheat.)
gSleepEasy <- 1
##Initialize Window, etc.
gWin <- MakeWindow("grey40")

ShowCursor(0)
##Initialize Font and colors

gBG <- MakeColor("grey40")
fg <- MakeColor("black")

gInstructionsFont <- MakeFont(gPEBLBaseFont,0,22, fg,MakeColor("white"),1)
gHeaderFont <- MakeFont(gPEBLBaseFont, 0,20,fg,gBG,1)
gStimFont <- MakeFont(gPEBLBaseFont,0,60,fg,gBG,0)
## Make and place the instruction box, then hide it
gInstructions <- MakeTextBox("", gInstructionsFont, 600,400)
AddObject(gInstructions, gWin)
Move(gInstructions, gVideoWidth/2-300, 100)
Hide(gInstructions)
## Make and hide a header label
gHeader <- MakeLabel("",gStimFont)
AddObject(gHeader,gWin)
Move(gHeader, gVideoWidth/2, 50)
Hide(gHeader)
gStimLabel <- MakeLabel("",gStimFont)
AddObject(gStimLabel,gWin)
Move(gStimLabel, gVideoWidth/2, gVideoHeight/2)
Hide(gStimLabel)
gFooter1 <- MakeLabel("",gHeaderFont)
AddObject(gFooter1,gWin)
Move(gFooter1, gVideoWidth/2, gVideoHeight-150)
Hide(gFooter1)
gFooter2 <- MakeLabel("",gHeaderFont)
AddObject(gFooter2,gWin)
Move(gFooter2, gVideoWidth/2, gVideoHeight-100)
Hide(gFooter2)
if(not FileExists("data"))
{
  Print("Creating 'data' directory")
  MakeDirectory("data")
} elseif(not IsDirectory("data"))
{
  SignalFatalError("File named 'data' exists. Remove so data directory can be created.")
}
}
define GetStrings(language)
{
##stimuli are defined here because different languages may require different number or letter characters.
if (Uppercase(language) == "BR")
{
gNeutralStim <- ["Z","ZZ","ZZZ","M","MM","MMM","G","GG","GGG"]
gIncongruentStim <- ["11","111","2","222","3","33"]
gCongruentStim <- ["1","22","333"]
if(gResponseType=="keyboard")
{
gKeys <- [1,2,3]
keysting <- "teclas 1, 2, ou 3"
} elseif(gResponseType=="keypad")
{
gKeys <- ["<KP_1>","<KP_2>","<KP_3>"]
keysting <- "teclas 1, 2, ou 3"
}
}
else{
gKeys <- [1,2,3]
keysting <- "um, dois, ou três"
}
}

```

gInstructionsBase<- "Você participará de um teste no qual indicará quantos itens aparecerão na tela. Poderão ser um, dois ou três itens. Algumas vezes serão letras e outras serão números. "

```

    gKeyManual <- "Você responderá apertando as teclas 1-2-3.
Z   1  2  3   (responda pressionando '1')
MM  11 22 33   (responda pressionando '2')
GGG 111 222 333 (responda pressionando '3').
Lembre de ignorar o significado do item. Pressione qualquer tecla para iniciar o treino."
    gKeyVocal <- "You will need to respond by saying the number 1, 2, or 3, loudly and clearly, into the microphone:
Z   1  2  3   (respond by saying '1')
MM  11 22 33   (respond by saying '2')
GGG 111 222 333 (respond by saying '3').
Remember to ignore the actual identity of the character. Press any key to try a practice round."
if(gResponseType=="keyboard" or gResponseType=="keypad")
{
  gInstructionsText <- gInstructionsBase+gKeyManual
}else{
  gInstructionsText <- gInstructionsBase+gKeyVocal
}
gBreak1Text <- "O Treino está completo. Agora você fará o teste real. Lembre de responder o mais rápido e correto possível. Pressione qualquer tecla para começar."
gBreak2Text <- "Agora você terá um breve intervalo. Quando recomçar, lembre de responder o mais rápido e correto possível. Pressione qualquer tecla para começar."
gPracticeHeaderText <- "TREINO"
gTestHeaderText <- "TESTE"
gFooter1TextManual <- "Pressione "+keysting+" para o número de itens que aparecer na tela"
gFooter1TextManual2 <- "Pressione "+keysting+" ou '4' para outro erro."
gFooter1TextVocal <- "Say "+keysting+" for the number of characters that appear on the screen"
gFooter2Text <- "Responda o mais rápido e correto possível."
gExitText <- "Agora você terminou. Obrigada por sua participação."
} else {
#####
##Default to english
gNeutralStim <- ["Z","ZZ","ZZZ","M","MM","MMM","G","GG","GGG"]
gIncongruentStim <- ["11","111","2","222","3","33"]
gCongruentStim <- ["1","22","333"]
if(gResponseType=="keyboard")
{
  gKeys <- [1,2,3]
  keysting <- "teclas 1, 2, ou 3"
} elseif(gResponseType=="keypad")
{
  gKeys <- ["<KP_1>","<KP_2>","<KP_3>"]
  keysting <- "1, 2, or 3 key"
}else{
  gKeys <- [1,2,3]
  keysting <- "one, two, or three"
}
gInstructionsBase<- "You are about to take part in an study in which you will be asked to determine how many characters are displayed on the computer screen. There will either be one, two, or three characters. Sometimes the characters will be letters, and other times they will be numbers. "
    gKeyManual <- "You will need to respond with the 1-2-3 keys on the "+gResponsetype+":
Z   1  2  3   (respond by pressing '1')
MM  11 22 33   (respond by pressing '2')
GGG 111 222 333 (respond by pressing '3').
Remember to ignore the actual identity of the character. Press any key to try a practice round."
    gKeyVocal <- "You will need to respond by saying the number 1, 2, or 3, loudly and clearly, into the microphone:
Z   1  2  3   (respond by saying '1')
MM  11 22 33   (respond by saying '2')
GGG 111 222 333 (respond by saying '3').
Remember to ignore the actual identity of the character. Press any key to try a practice round."
if(gResponseType=="keyboard" or gResponseType=="keypad")
{
  gInstructionsText <- gInstructionsBase+gKeyManual
}else{
  gInstructionsText <- gInstructionsBase+gKeyVocal
}
gBreak1Text <- "Practice is complete. Now you will be tested for real. Remember to answer as quickly and accurately as possible. Press any key to begin."
gBreak2Text <- "You may now take a short break. When you resume, remember to answer as quickly and accurately as possible. Press any key to continue."
gPracticeHeaderText <- "PRACTICE"
gTestHeaderText <- "TEST"
gFooter1TextManual <- "Press "+keysting+" for the number of characters that appear on the screen"
gFooter1TextManual2 <- "Press "+keysting+" or '4' for other error."

```

```
gFooter1TextVocal <- "Say "+keysting+" in response to the number of characters you see appear on the screen"  
gFooter2Text <- "Responda p mais rápido e preciso possível."  
gExitText <- "You are now finished. Thank you for participating."  
}  
}  
## Repeat list enough times to make it length long.  
##  
define RepeatEnough(list,length)  
{  
l <- Length(list)  
times <- Ceiling(length/l)  
tmp <- Flatten(ShuffleRepeat(list,times))  
return SubList(tmp,l,length)  
}
```

## ANEXO B- Programação BCST

```
#####
## Berg's Card Sorting Task, Version 0.6
## (better known as the Wisconsin Card Sorting Test (tm)).
## For use with PEBL 0.10 or later
## http://pebl.sf.net
## Part of The PEBL Psychological Testing Battery
## 2010 Released into Public Domain
## by Shane T. Mueller, Ph.D. (smueller at obereed dot net)
##
## as found in:
## BCST-64 developed in conjunction with Arthur Kummer,
##
##
##
## Polish translation contributed by Artur Pilacinski
## Korean translation contributed by Hyunsoo ra ma
## Chinese translation contributed by Yu-Hsiang Wu
## Portuguese translation contributed by Arthur Kummer
#####
## Raw data file outputs:
## subnum, trialnum, category count, color, shape, number, response, accuracy
## perseveration count, absolute time, response time.
define Start(IPar)
{

## This is how long of a run is required before switching categories.
runlength <- 10

gSleepEasy <- 1
gUseMouse <- 1 #0 for keyboard 1-4, 1 for mouse entry.

if(gUseMouse)
{
tag <- "-mouse"
}else {

tag <- "-keyboard"
}

## The labels/instructions live in the file labels-en.txt. Change
## that file to translate to a new language.

textlist <- FileReadList(Lowercase("translations/labels-"+gLanguage+ tag + ".txt"))
gPromptText <- Nth(textlist,5)
gCorrectText <- Nth(textlist,6)
gIncorrectText <- Nth(textlist,7)
## results are stored in this file

MakeDirectory("data")
gFileOut <- FileOpenWrite("data/bcst128-" + gSubNum + ".csv")
gSleepEasy <- 1
persevs <- []
totCorr <- 0
totPers <- 0
totPersErr <- 0
totUnique <- 0 ##Counts number of 'unique' errors; errors that are not correct by another rule.

persrun <- [] ##computes length of 'perseverative runs'
persrunflag <- 1

## Initialize global objects for use later. Defined at end of file
Initialize()

gSubNum <- GetSubNum(gWin)

if(gUseMouse==0)
{
ShowCursor(0) #hide the cursor
}
}
```

```

gColors <- ["Red", "Green", "Yellow", "Blue"]
gShapes <- ["Triangle", "Star", "Cross", "Circle"]
gNumbers <- ["1", "2", "3", "4"]
##rule order is fixed: color-shape-number
rules <- [1,2,3]

##Get the cards from a special function.
csn <- GetStandardCards()
decksiz <- Length(csn)
stimcat <- csn

images<-[]

loop(i,stimcat)
{
  filename <- "png/" + ListToString(i) + ".png"
  x <- MakeImage(filename)
  images <- Append(images,x)
}
Draw()
stim <- (Transpose(Append( Transpose(stimcat),images)))

## Give instructions; wait for them to hit a key,
SetText(gInstructions,Nth(textlist,1))
Show(gInstructions)
Draw()
instru01 <- LoadSound("instru01.wav")
PlayBackground(instru01)
Wait(17100)
Stop(instru01)
WaitForIt()

## load canonical comparison cards. This is perhaps supposed to
## be randomized.

comps <- Transpose([gColors,gShapes,gNumbers])
comp1 <- MakeImage("png/"+ListToString(First(comps))+".png")
comp2 <- MakeImage("png/"+ListToString(Nth(comps,2))+".png")
comp3 <- MakeImage("png/"+ListToString(Nth(comps,3))+".png")
comp4 <- MakeImage("png/"+ListToString(Nth(comps,4))+".png")

AddObject(comp1,gWin)
AddObject(comp2,gWin)
AddObject(comp3,gWin)
AddObject(comp4,gWin)
Move(comp1, 28 + 125/2, 105)
Move(comp2, 28*2 + 125 + 125/2,105)
Move(comp3, 28*3 + 125*2 + 125/2,105)
Move(comp4, 28*4 + 125*3 + 125/2,105)

if(gUseMouse == 0)
{
  gHeader.text <- "1      2      3      4"
}
Move(gHeader,320,220)
Show(gHeader)

      gInstructions.text <- Nth(textlist,2)
Draw()
instru02 <- LoadSound("instru02.wav")
PlayBackground(instru02)
Wait(25100)
WaitForIt()

gInstructions.text <- Nth(textlist,3)
Draw()
instru03 <- LoadSound("instru03.wav")
PlayBackground(instru03)
Wait(16800)
WaitForIt()

Hide(gInstructions)

```

```

Show(gPrompt)
Draw()

tableau <- MakeCanvas(600,220,MakeColor("grey20"))
AddObject(tableau,gWin)
Move(tableau,320,350)

##add the click-on boxes

box1 <- Rectangle(28+125/2,350,130,210,MakeColor("white"),0);AddObject(box1,gWin)
box2 <- Rectangle(28*2+125+125/2,350,130,210,MakeColor("white"),0);AddObject(box2,gWin)
box3 <- Rectangle(28*3+125*2+125/2,350,130,210,MakeColor("white"),0);AddObject(box3,gWin)
box4 <- Rectangle(28*4+125*3+125/2,350,130,210,MakeColor("white"),0);AddObject(box4,gWin)
Hide(box1)
Hide(box2)
Hide(box3)
Hide(box4)

Show(box1);Show(box2);Show(box3);Show(box4)

Draw()

#####
## Begin presenting and collecting data
#####

FilePrint(gFileOut,
"Subnum,trial_run,rule,lastrule,color,shape,number,resp,corr,last_corr,corr_col,corr_shape,corr_num,persev,persev_err,abstime,rt")
#####
## Start the experimental trials.
##

rule <- First(rules)
rules <- Rotate(rules,1)
lastrule <- -1

failuretomaintain <- 0 ##failure to maintain set

trialsoncat <- 0 ##how long have you been working on this trial?
trialspercat <- [] ##Keep a list of the trials taken per category.
##This is used to measure learning-to-learn
erroroncat <- 0 ##how many errors on this category
errorspercat <- [] ##Keep a list of the errors made on the category
##This is used to measure learning-to-learn

conceptual <- 0 ##Record conceptual responses: 3-runs or greater

firstCat <- 0
run <- 0
persev <- 0
trialnum <- 1
numCats <- 0
numCatsExp <- 1

##Loop through each stimulus card
loop(s,stim)
{
  trialsoncat <- trialsoncat+1

  #Display the card
  card <- Nth(s,4)
  AddObject(card,gWin)
  Move(card,700,480)
  Draw()
  time0 <- GetTime()

  ## Get the response
  if(gUseMouse)
  {
    resp <- WaitForClickOnTarget([box1,box2,box3,box4],[ "1","2","3","4"])
  }else{
    resp <- WaitForListKeyPress(["1","2","3","4"])
  }
}

```

```

time1 <- GetTime()
rt <- time1 - time0

## Score the trial returns a pair. [cur-rule, last-rule]
##

corrBoth <- ScoreTrial(s,resp,rule,lastrule)
corrLast <- First(corrBoth)
corr <- Nth(corrBoth,2)

##Also compute whether the choice was correct for each given rule
corrCol <- Nth(gColors,ToNumber(resp)) == Nth(s,1)
corrShape <- Nth(gShapes,ToNumber(resp)) == Nth(s,2)
corrNum <- Nth(gNumbers,ToNumber(resp)) == Nth(s,3)

#####
##animate the card location.

#Get end location
x <- 125/2 + 28*ToNumber(resp) + 125*(ToNumber(resp)-1)
y <- 350
##compute distance to travel to scale frames to distance.
dist <- Sqrt((y-card.y)^2 + (x-card.x)^2)
#Increase stepsize to animate faster
stepsize <- 100
frames <- Ceiling(dist/stepsize)+1
ydiff <- (y - card.y)/(frames)
xdiff <- (x - card.x)/(frames)
steps <- Sequence(frames,0,-1)

#move toward end location
loop(i,steps)
{
  card.x <- x - xdiff*i
  card.y <- y - ydiff*i
  Draw()
}

RemoveObject(card,gWin)
AddObject(card,tableau)
Move(card,x-20,110)
Draw(tableau)
RemoveObject(card,tableau)
Draw()

#If it is correct, update run counters
if(corr == 1)
{
  run <- run + 1
  totCorr <- totCorr + 1
  feedback <- gCorrect
Draw()
  correto <- LoadSound("correto.wav")
  PlayBackground(correto)
  Wait(600)
  Stop(correto)
} else {
  ##It is incorrect.
  feedback <- gIncorrect
Draw()
  errado <- LoadSound("errado.wav")
  PlayBackground(errado)
  Wait(550)
  Stop(errado)
  ##Increment failuretomaintain if run>=5
  if(run>=5)
  {
    failuretomaintain <- failuretomaintain+1
  }

  erroroncat <- erroroncat + 1 ## increment category-specific
  ## error

  ##Increment conceptual responses here, when the run ends on an error

```

```

if(run>=3)
{
  conceptual <- conceptual + run
}

run <- 0

}

##If it was correct according to the last rule, increment counters.
if(corrLast == 1) {

  totPers <- totPers + 1
  persev <- persev + 1
  trialpersev <- 1
  trialPersevErr <- 0

  ##If it was also an error, this is a perseverative error.
  if(corr==0)
  {
    totPersErr <- totPersErr +1
    trialPersevErr <- 1
  }

  } else {

#####
## This trial was not a perseveration (it could be either correct or incorrect)

##Make sure these are defined
  trialPersev <- 0
  trialPersevErr <- 0
  ##If the runflag is still 1, that means this is the first non-perseverative response
  ## in the category. Unset the flag and record the length of the run.
  if(persrunflag)
  {
    persrunflag <- 0
    ##Add the current perseverative run count;
    ## only do so after the first category
    if(numCats >0)
    {
      persRun <- Append(persRun, persev)
    }
  }

}

##This determines if the response is a 'unique' response; i.e.,
  ## not correct according to any rule

rulesum <- (corrCol + corrShape + corrNum)

##It was incorrect, and not correct according to the previous rule, and not
##a match to another property of the card, increment the unique counter
if(corr==0 and corrLast == 0 and rulesum == 0)
{
  totUnique <- totUnique + 1
}

FilePrint_(gFileOut, gSubNum + "," + trialnum + "," + numCatsExp + "," + rule + "," + lastrule + ",")
index <- 1
##Print out the trial data.
loop(i,s)
{
  if(index !=4)
  {
    FilePrint_(gFileOut,i+",")
  }
}

```

```

    }
    index <- index + 1
  }

  FilePrint_(gFileOut, resp + "," + corr + "," + corrLast + "," + corrCol + "," + corrShape + "," + corrNum + ",")
  FilePrint( gFileOut, trialpersev + "," + trialpersever + "," + time0 + "," + rt )
  Show(feedback)
  Draw()
  Wait(500)
  Hide(feedback)
#   Hide(gHeader)
#   Hide(gPrompt)
  Draw()

##if you have a run of runlength correct, change rules.
if(run >= runlength)
{
  lastrule <- rule
  rule <- First(rules)
  rules <- Rotate(rules,1)

#Update conceptual responses when a category is complete
conceptual <- conceptual + run

run <- 0
numCats <- numCats + 1
numCatsExp <- numCatsExp + 1

lastswitch <- trialnum
##Keep track of the number of perseverations in this category.
persevs <- Append(persevs, persev)
persev <- 0

persrunflag <- 1 ##Reset the perseverative run flag.
##Record the number of trials needed to attain first category.
if(numCats == 1)
{
  firstcat <- trialnum
}
      ##update the trialsoncat/trialspercat counters
  trialspercat <-Append(trialspercat,trialsoncat)
  errorspercat <-Append(errorspercat, erroroncat)
  erroroncat <- 0
      trialsoncat <- 0

}

trialnum <- trialnum + 1
if(numCats >= 6 )
{
  numCatsExp <- 6
  break
}
}

#The last card may have ended before the end.
#If it were complete, run would have been reset to 0.
#Add partial run to conceptual responses
  if(run>=3)
  {
    conceptual <- conceptual + run
  }

##Do the final output and computations.

##compute hand things up-front
num <- trialnum - 1
err <- num - totCorr
nonp <- (err-totPersErr)

#####
##compute learning-to-learn.
if(Length(trialspercat)>2)
{

```

```

if((Length(trialspercat)>3) or
  (Length(trialspercat)==3 and Third(trialspercat)>=10))
  {
  #Compute the number of extra trials, and the

  diffs <- []
  lastpc <- "XXX"

  tt <- 1 ##trial counter
  loop(i,Transpose([trialspercat,errorpercat]))
  {
  ##Only use this if there were 10 or more trials
  if(i>=10)
    {
    pc <- Second(i)/First(i)*100
    }

  if(tt > 1)
    {
    diffs <- Append(diffs,pc-lastpc)

    }
  tt <- tt + 1
  lastpc <- pc
  }
  learningtolearn <- Mean(diffs)

  } else {
  ##Invalid to compute learning-to-learn
  learningtolearn <- "Not enough data"
  }} else{
  ##Invalid to compute learning-to-learn
  learningtolearn <- "Not enough data"
  }

FileClose(gFileOut)
report <-FileOpenWrite("Resultados/bcst128-report-" + gSubNum + ".txt")

FilePrint(report,"-----")
FilePrint(report," Report for PEBL's Berg's Card Sorting Task-64,")
FilePrint(report," a free version of the Wisconsin Card Sorting Task (tm)*")
FilePrint(report," (c) 2010 Shane T. Mueller, with guidance from Arthur Kummer")

FilePrint(report," Version 0.1")
FilePrint(report," "+GetPEBLVersion())
FilePrint(report," "+TimeStamp())
FilePrint(report," Participant Code: " + gSubNum)
FilePrint(report," http://pebl.sf.net")
FilePrint(report,"-----")
FilePrint(report,"Parameters:")
FilePrint(report,"Run length to move to next category: "+runlength)
FilePrint(report,"Deck Size: "+ decksize)

FilePrint_(report,"Order of Cards: Rule 1 (color): ")
loop(i,gColors)
{
  FilePrint_(report,i+" ")
}
FilePrint(report,"")

FilePrint_(report,"          Rule 2 (shape): ")
loop(i,gShapes)
{
  FilePrint_(report,i+" ")
}
FilePrint(report,"")

FilePrint_(report,"          Rule 3 (number): ")
loop(i,gNumbers)
{
  FilePrint_(report,i+" ")
}
FilePrint(report,"")

```

```

FilePrint(report,"-----")
FilePrint(report,"Statistic          Value")
FilePrint(report,"-----")
FilePrint(report,"1. Categorias Completadas/Experienciadas: "+ numCats + " / " +numCatsExp)
FilePrint(report,"2. Numero de Tentaivas:          " + num)

FilePrint(report,"          -----")
FilePrint(report,"          Total Percent Mean/Cat")
FilePrint(report,"3. Respostas Corretas:          " + totCorr + Tab(1) + Roundoff(totCorr/num,4)*100 + Tab(1) +
Roundoff(totCorr/numcatsExp,3))
FilePrint(report,"4. Total Erros:          " + err + Tab(1) + Roundoff(err/num,4)*100 + Tab(1) + Roundoff(err/numCatsExp,3))
FilePrint(report,"5. Respostas Perseverativas          " + totPers + Tab(1) + Roundoff(totPers/num,4)*100 + Tab(1) +
Roundoff(totPers/(numCatsExp-1),3))
FilePrint(report,"6. Erros Perseverativos          " + totPersErr + Tab(1) + Roundoff(totPersErr/num,4)*100 + Tab(1) +
Roundoff(totPersErr/(numCatsExp-1),3))
FilePrint(report,"7. Erros Não-Perseverativos          " + nonp + Tab(1) + Roundoff(nonp/num,4)*100 + Tab(1) +
Roundoff(nonp/(numCatsExp),3))
FilePrint(report,"8. Unique Errors          " + totUnique + Tab(1) + Roundoff(totUnique/num,4)*100 + Tab(1) +
Roundoff(totUnique/(numCatsExp),3))

FilePrint(report,"          -----")

FilePrint(report,"8. Tentativas para completar complete 1st cat: "+ firstcat)
FilePrint(report,"9. Falha em manter o contexto: "+failuretomaintain)
FilePrint(report,"10. Aprendendo a Aprender: "+learningtolearn)
FilePrint(report,"11. Respostas de Nivel Conceitual: "+conceptual)
FilePrint(report," Percent CLR:          "+Roundoff(100*(conceptual/num),1))
FilePrint(report,"12. Perseverative Runs: Min Max Mean Total")
if(Length(persrun)>0)
{
FilePrint(report,"          "+Min(persrun)+ " " + Max(persrun) + " " + Roundoff(Mean(persrun),3) + " " +
Sum(persrun))
} else{
FilePrint(report,"no perseverative runs")
}
FilePrint(report,"-----")

FilePrint(report,"Notes:")

FilePrint(report,"Stats 1-11 are standard statistics commonly
reported. #1 contains both categories completed (standard stat)
and categories experienced, whether or not the last one was
actually completed. Means are computed based on categories
experienced (not categories completed), and perseverative means
based on categories experienced AFTER the first category usually a
maximum of 8. Total and percent for 1-11 are computed like the
commonly-reported stats, to the best of our knowledge. #12,
Perseverative runs, computes the number of error perseverations in
a row at the beginning of each new category.")

FilePrint(report,"*PEBL, the PEBL test battery, and this test are unaffiliated with the Wells Printing Company or PAR Inc, the producers of
the 'Wisconsin Card Sorting Test' (tm)")
FileClose(report)
##Now, show debriefing info.

feedback <- Nth(textlist,4)

SetText(gInstructions, feedback)
Hide(gPrompt)
Hide(gHeader)
Hide(tableau)
Hide(box1)
Hide(box2)
Hide(box3)
Hide(box4)

Show(gInstructions)
Draw()
instru04 <- LoadSound("instru04.wav")
PlayBackground(instru04)
Wait(7200)
Wait(1000)
WaitForIt()

```

```

ShowCursor(1)
}

## This is a standard initializer function that sets up typical
## objects used in experiments.
define Initialize()
{
  ##Initialize Window, etc.
  gWin <- MakeWindow("black")

  ##Initialize Font and colors
  gSleepEasy <- 1

  ##Get subject code if we need to:
  if(gSubNum==0)
  {
    gSubNum <- GetSubNum(gWin)
  }

  bg <- MakeColor("black")
  fg <- MakeColor("white")
  bg2 <- MakeColor("black")

  colGreen <- MakeColor("green")
  colRed <- MakeColor("red")

  gStimFont <- MakeFont(gPEBLBaseFontMono, 0, 24,fg,bg,0)
  gCorrFont <- MakeFont(gPEBLBaseFontMono, 0, 30,colGreen, bg,0)
  gIncorrFont <- MakeFont(gPEBLBaseFontMono, 0, 30,colRed, bg,0)

  gInstructionsFont <- MakeFont(gPEBLBaseFont,0,18, fg,bg2,1)
  gHeaderFont <- MakeFont(gPEBLBaseFont, 0,36,fg,bg,1)

  ## Make and place the instruction box, then hide it
  gInstructions <- MakeTextBox("", gInstructionsFont, 600,200)
  AddObject(gInstructions, gWin)
  Move(gInstructions, gVideoWidth/2-300, gVideoHeight/2+5)
  Hide(gInstructions)

  ## Make and hide a header label
  gHeader <- MakeLabel("",gHeaderFont)
  AddObject(gHeader,gWin)
  Move(gHeader, gVideoWidth/2, 100)
  Hide(gHeader)

  #Create the instructions cue
  gPrompt <- MakeLabel(gPromptText,gStimFont)
  AddObject(gPrompt,gWin)
  Move(gPrompt,350,gVideoHeight-80)
  Hide(gPrompt)

  gCorrect <- MakeLabel(gCorrectText,gCorrFont)
  AddObject(gCorrect,gWin)
  Move(gCorrect,gVideoWidth/2,gVideoHeight-40)
  Hide(gCorrect)

  gIncorrect <- MakeLabel(gIncorrectText,gIncorrFont)
  AddObject(gIncorrect,gWin)
  Move(gIncorrect,gVideoWidth/2,gVideoHeight-40)
  Hide(gIncorrect)
}

## This returns two 1-0 correctness stats: [last-rule, cur-rule]
define ScoreTrial(triallist, resp, rule,lastrule)
{
  ##The piles are defined by:
  ## Rule 1: COLOR: red green blue yellow
  ## Rule 2: SHAPE: circle star triangle cross
  ## Rule 3: NUMBER: 1 2 3 4
  ## Depending upon the rule (triallist[5]),

```

```

## the correct response depends upon column 1,2,3

dimensions <- [gColors, gShapes, gNumbers]
dimension <- Nth(triallist, rule)
dimlist <- Nth(dimensions,rule)

if(lastrule == -1)
{
  lastdim <- 0
  lastdimlist <- [-1,-1,-1,-1]
} else {

  lastdim <- Nth(triallist, lastrule)
  lastdimlist <- Nth(dimensions,lastrule)
}
##So, the trial is correct if, along the chosen dimension,
## the response is the same as the pile.

corr <- (Nth(dimlist,ToNumber(resp)) == dimension)
lastcorr <- (Nth(lastdimlist,ToNumber(resp)) == lastdim)

if(corr)
{
  ##It was correct
  corCur <- 1
} else {
  corCur <- 0
}

if (lastcorr)
{
  corLast <- 1
} else {
  corLast <- 0
}

return [corLast, corCur]
}

define Roundoff(x,dec)
{
  power <- 10^dec
  return Round( x * power)/power
}

define WaitForIt()
{
  if(gUseMouse)
  {
    WaitForDownClick()
  } else {
    WaitforAnyKeyPress()
  }
}

define GetStandardCards()
{
  cards <- [ ["Green", "Triangle", "1"],
["Red", "Cross", "4"],
["Blue", "Triangle", "2"],
["Red", "Circle", "1"],
["Green", "Star", "4"],
["Yellow", "Cross", "1"],
["Blue", "Triangle", "4"],
["Red", "Circle", "3"],
["Green", "Cross", "4"],
["Yellow", "Circle", "2"],
["Blue", "Star", "1"],
["Red", "Triangle", "3"],

```

["Blue", "Cross", "2"],  
 ["Yellow", "Star", "1"],  
 ["Green", "Triangle", "3"],  
 ["Blue", "Circle", "4"],  
 ["Red", "Star", "2"],  
 ["Yellow", "Circle", "3"],  
 ["Red", "Triangle", "4"],  
 ["Yellow", "Circle", "1"],  
 ["Blue", "Star", "2"],  
 ["Green", "Cross", "3"],  
 ["Yellow", "Star", "2"],  
 ["Blue", "Triangle", "3"],  
 ["Red", "Star", "4"],  
 ["Yellow", "Triangle", "2"],  
 ["Blue", "Cross", "3"],  
 ["Red", "Circle", "4"],  
 ["Green", "Star", "2"],  
 ["Red", "Cross", "1"],  
 ["Green", "Circle", "4"],  
 ["Red", "Star", "1"],  
 ["Blue", "Circle", "3"],  
 ["Yellow", "Cross", "4"],  
 ["Green", "Star", "1"],  
 ["Yellow", "Triangle", "4"],  
 ["Blue", "Circle", "2"],  
 ["Yellow", "Star", "3"],  
 ["Green", "Circle", "1"],  
 ["Blue", "Cross", "4"],  
 ["Red", "Triangle", "1"],  
 ["Blue", "Star", "3"],  
 ["Yellow", "Circle", "4"],  
 ["Green", "Triangle", "2"],  
 ["Red", "Star", "3"],  
 ["Yellow", "Cross", "2"],  
 ["Blue", "Circle", "1"],  
 ["Yellow", "Triangle", "3"],  
 ["Green", "Cross", "2"],  
 ["Yellow", "Star", "4"],  
 ["Blue", "Triangle", "1"],  
 ["Green", "Star", "3"],  
 ["Red", "Circle", "2"],  
 ["Green", "Triangle", "4"],  
 ["Blue", "Cross", "1"],  
 ["Green", "Circle", "3"],  
 ["Red", "Cross", "2"],  
 ["Yellow", "Triangle", "1"],  
 ["Red", "Cross", "3"],  
 ["Green", "Circle", "2"],  
 ["Blue", "Star", "4"],  
 ["Yellow", "Cross", "3"],  
 ["Red", "Triangle", "2"],  
 ["Green", "Cross", "1"],  
 ["Green", "Triangle", "1"],  
 ["Red", "Cross", "4"],  
 ["Blue", "Triangle", "2"],  
 ["Red", "Circle", "1"],  
 ["Green", "Star", "4"],  
 ["Yellow", "Cross", "1"],  
 ["Blue", "Triangle", "4"],  
 ["Red", "Circle", "3"],  
 ["Green", "Cross", "4"],  
 ["Yellow", "Circle", "2"],  
 ["Blue", "Star", "1"],  
 ["Red", "Triangle", "3"],  
 ["Blue", "Cross", "2"],  
 ["Yellow", "Star", "1"],  
 ["Green", "Triangle", "3"],  
 ["Blue", "Circle", "4"],  
 ["Red", "Star", "2"],  
 ["Yellow", "Circle", "3"],  
 ["Red", "Triangle", "4"],  
 ["Yellow", "Circle", "1"],  
 ["Blue", "Star", "2"],  
 ["Green", "Cross", "3"],  
 ["Yellow", "Star", "2"],  
 ["Blue", "Triangle", "3"],

```

["Red", "Star", "4"],
["Yellow", "Triangle", "2"],
["Blue", "Cross", "3"],
["Red", "Circle", "4"],
["Green", "Star", "2"],
["Red", "Cross", "1"],
["Green", "Circle", "4"],
["Red", "Star", "1"],
["Blue", "Circle", "3"],
["Yellow", "Cross", "4"],
["Green", "Star", "1"],
["Yellow", "Triangle", "4"],
["Blue", "Circle", "2"],
["Yellow", "Star", "3"],
["Green", "Circle", "1"],
["Blue", "Cross", "4"],
["Red", "Triangle", "1"],
["Blue", "Star", "3"],
["Yellow", "Circle", "4"],
["Green", "Triangle", "2"],
["Red", "Star", "3"],
["Yellow", "Cross", "2"],
["Blue", "Circle", "1"],
["Yellow", "Triangle", "3"],
["Green", "Cross", "2"],
["Yellow", "Star", "4"],
["Blue", "Triangle", "1"],
["Green", "Star", "3"],
["Red", "Circle", "2"],
["Green", "Triangle", "4"],
["Blue", "Cross", "1"],
["Green", "Circle", "3"],
["Red", "Cross", "2"],
["Yellow", "Triangle", "1"],
["Red", "Cross", "3"],
["Green", "Circle", "2"],
["Blue", "Star", "4"],
["Yellow", "Cross", "3"],
["Red", "Triangle", "2"],
["Green", "Cross", "1"]

```

```

return cards
}

```



```

{
stim <- Phillips1999("A")
}elseif(type=="6")
{
stim <- Phillips1999("B")

}elseif(type=="7")
{
stim <- Phillips1999("C")      ##5-disk problems,
                                ## Argument specifies sets "A", "B", or "C"
                                ## unconstrained pole heights
                                ## which consist of different problems
}elseif(type=="8")
{
stim <- FimbelOld()           ##15 Shallice-like problems
}elseif(type=="9")
{
stim <- FimbelYoung()        ##35 Shallice-like problems

}elseif(type=="0")
{
stim <- TOLR()                ##TOL-R ala Schnirman
}elseif(Uppercase(type=="A")
{
##Culbertson & Zillmer      ##15 shallice-like problems, easy for adolescents.
stim <- TOLDX()
}
#####
## These are some global variables that override test-specific settings.
# gUseTimer <- 1 ##This hard-codes use of timer. Uncomment to
##override the defaults built into each particular test function.
gTimeLimit <- 120 ##default time limit, in seconds.
# gUseTurnLimit <- 1 ##this hard-codes turn limits, overriding
## setting in original definition
gAllowReset <- 1 ## Make a 'reset' button?
#####
if(gUseReverse)
{
stim <- ReverseStimuli(stim)
}

gInstructions.text <- gInstText1

if(gUseTimer)
{
gInstructions.text <- gInstructions.text + gInstText2
}
Draw()
instru01 <- LoadSound("instru01.wav")
PlayBackground(instru01)
Wait(35900)
Stop(instru01)
WaitForDownClick()
Draw()
if(gUseTurnLimit)
{
gInstructions.text <- gInstText3
}
gInstructions.text <- gInstructions.text + CR(1) + gBegin
gInstructions.font <- gFont2
Draw()
instru02 <- LoadSound("instru02.wav")
PlayBackground(instru02)
Wait(25600)
Stop(instru02)
Draw()
WaitForDownClick()
Hide(gInstructions)
Draw()
InitPoles()
FilePrint(gFileOut,"sub trial size current end step reset tries score abstime trialtime clicktime done")
FilePrint(gFileOutSum,"sub trial size shortest startlab endlab startconf endconf success tries score steps
roundstart starttime firsttime time")
gInstructions.width <- 220
gInstructions.height <- 520

```

```

Move(gInstructions,555,50)
#####
##Make a little demo animation:
c1 <- PlotDisksAnimate([[2,1],[3],[ ]] ,0,450,gDisks,10)
c2 <- PlotDisksAnimate([[1],[2],[3]] ,0,130,gDisks2,10)
gInstructions.text <- gSeeDemo
Show(gInstructions)
Draw()
gSeedemo <- LoadSound("gSeedemo.wav")
PlayBackground(gSeedemo)
Wait(3700)
Stop(gSeedemo)
WaitForDownClick()
  Wait(300)
c1 <- PlotDisksAnimate([[2],[3],[ ]], 1,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[2],[3,1],[ ]], 0,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[ ],[3,1],[ ]],2,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[ ],[3,1],[2]],0,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[ ],[3],[ ]],1,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1],[3],[2]],0,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1],[ ],[2]],3,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1,3],[ ],[2]],0,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1,3],[ ],[ ]],2,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1,3],[2],[ ]],0,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1],[2],[ ]],3,450,gDisks,10)
Wait(300)
c1 <- PlotDisksAnimate([[1],[2],[3]],0,450,gDisks,10)
Wait(300)
gInstructions.text <-gSeedemo2
Draw()
gSeedemo2 <- LoadSound("gSeedemo2.wav")
PlayBackground(gSeedemo2)
Wait(3500)
Stop(gSeedemo2)
WaitForDownClick()
Hide(gInstructions)
  HideAll()
#####
gTrial <- 1
##Keep track of a 'score' option.
gScore <- 0
loop(pair, stim)
{
  gReset <- 1
  gTries <- 1
  ##You should run a trial if gReset is both not 0 (0 signals completion)
  ## and gReset <3 (4+ indicates you ran out of turns)
  gStartTime <- GetTime()
  gEndTime <- gStartTime+ gTimeLimit*1000

  while(gReset == 1 and gTries <= 3)
  {

    gResetLabel.text <- gResetText + " ["+(3-gTries)+"]"

    gReset <- 0
    ##This runs a single trial attempt.
    success <- Trial(pair)
    ##### The following creates the proper feedback messages.
    if(success)
    {
      ##Success--use message about 'complete, click mouse to continue'
      gInstructions.text <- gContinue
      gReset <- 0
    } else {
      gInstructions.text <- gIncomplete
    }
  }
}

```

```

if(gReset)
{
  #the hit the reset button.
  gInstructions.text <- gInstructions.text + " " + gResetLab
  gTries <- gTries + 1
} else {
  if(gMovesRanOut)
  {
    ##Moves ran out on a single round
    ginstructions.text <- gInstructions.text + " " + gMovesOutLab
    if(gAllowReset and gTries < 3)
    {
      ##reset the
      gInstructions.text <- gInstructions.text + " " + gResetLab
      gTries <- gTries + 1
      gReset <- 1
    }
  }
  if(gTimeRanOut)
  {
    ginstructions.text <- gInstructions.text + gTimeOutLab
  }
}
if(gReset ==0)
{
  gInstructions.text <- gInstructions.text+CR(2)+ gContinueFail
}
}
##Do some cleanup after the trial is over.
# RemoveObject(gInstructions,gWin)
# AddObject(gInstructions,gWin)
Show(gInstructions)
Draw()
WaitForDownClick()
Hide(gInstructions)
HideAll()
}
gTrial <- gTrial + 1
}
##Record a final line
gInstructions.text <- gDebriefing
Show(gInstructions)
Draw()
WaitForAnyKeyPress()
}
define Trial(pair)
{
  ##Keep track of time to make first move.
  firstmovetime <- -1
  ##extract data from input
  config <- First(pair)
  config1 <- Second(pair)
  shortest <- Third(pair)
  startlab <- Fourth(pair)
  endlab <- Fifth(pair)
  numdisks <- Length(Flatten(config))
  origstring <- MakeStateString(config)
  ##reset flags that keep track if move/time runs out.
  gMovesranout <- 0
  gTimeranout <- 0
  ## Make standard and new configurations
  c1 <- PlotDisksAnimate(config,0,450,gDisks,10)
  c2 <- PlotDisksAnimate(config1,0,130,gDisks2,10)
  ##if appropriate, create the click graph
  if(shortest > 0 and gUseTurnLimit)
  {
    cGraph <- MakeGraph(shortest,shortest,650,400,gClickLab,shortest)
  }
  #Set some initial timing variables
  startTime <- GetTime()
  lasttime <- startTime
  time <- startTime
  ##record the starting configuration of the trial
  FilePrint(gfileout, gSubNum + " " + gTrial + " " + numdisks + " " +
    MakeStateString(config) + " " +
    MakeStateString(config1) + " 0 0 " + gTries + " " + gScore + " " +

```

```

        starttime + " 0  "+
        (time - lasttime) + " 0  ")
continue <- 1
step <- 1
##This part creates the timer graph
if(gUseTimer)
{
  ##Timegraph defaults to 120 sec.
  ##the ticks hould be every 10 sec, unless there are less than 4
  ## ticks. then every 5, then every 1 sec
  numticks <- gTimeLimit/10
  if(numticks < 4)
  {
    numticks <- gTimeLimit/5
  }
  if(numticks < 3)
  {
    numticks <- gTimeLimit
  }
}

gTimeGraph <- MakeGraph(gTimeLimit,gTimeLimit,710,400,gTimeLab,numticks)
}
Draw()
success <- 0
#Now, allow disk moves until the two have identical configurations.
while(continue)
{
  #Get the location we want to move the disk to
  config <- MoveDisk(config)
  time <- GetTime()
  ##record time to first move.
  if ( firstmovetime == -1)
  {
    firstmovetime <- time -starttime
  }
  ##Check to see if we have succeeded
  if(StackEqual(config,config1))
  {
    success <- 1
    continue <- 0

    #update the score; we are correct
    gScore <- gScore + (4-gTries)
  }

  ##record move time.
  FilePrint(gfileout, gSubNum + " " + gTrial + " " + numdisks + " " +
    MakeStateString(config) + " " + MakeStateString(config1) + " " +
    step + " " + gReset + " " + gTries + " " + gScore + " " +
    time + " " + (time - starttime) + " " +
    (time-lasttime) + " " + success)

  lasttime <- time

  ##only update the graph if we have a good measure of
  ## shortest path, and if the global flag specifies it.
  if(shortest > 0 and gUseTurnLimit)
  {
    rem <- Max([0,shortest-step])
    UpdateGraph(cgraph,rem, 650,400,shortest)
    #if no moves remain:
    if(rem == 0)
    {
      #if continue == 0, we have solved the puzzle already.
      if(continue == 0)
      {
        success <- 1
      }else{
        gMovesRanOut <- 1
        continue <- 0
        success <- 0
      }
    }
  }
}

```

```

}

##See if the move ended because time ran out
if(gUseTimer)
{
  if(gTimeRanOut)
  {
    success <- 0
    continue <- 0
  }
}

##see if the move ended because we hit the reset button.
if(gReset)
{
  continue <- 0
  success <- 0
}

Draw()
step <- step + 1
}

Wait(100)
FilePrint(gFileOutSum, gSubNum + " " + gTrial + " " + numdisks +
" " + shortest + " " + startlab + " " + endlab + " " +
origstring + " " + MakeStateString(config1) + " " + success +
" " + gTries + " " + gScore + " " + (step-1) + " " + gStartTime +
" " + starttime + " " + firstmovetime + " " + (time-starttime))

if(gUseTimer)
{
  RemoveComplex(gTimeGraph)
}

if(shortest > 0 and gUseTurnLimit)
{
  RemoveComplex(cGraph)
}
firstmovetime <- -1

return success
}

define HideAll()
{
  loop(i, Merge(gDisks,gDisks2))
  {
    Hide(i)
  }
}

define Init()
{
  gSleepEasy <- 1
  gVideoWidth <- 800
  gVideoHeight <- 600
# ShowCursor(0)
gWin <- MakeWindow("black")
gSubNum <- GetSubNum(gWin)
## gPEBLBaseFont <- "Vera.ttf" ##uncomment for version 0.10
##These may need to be changed for a particular language.
gFont <- MakeFont(gPEBLBaseFont,0,15,MakeColor("white"),MakeColor("black"),1)
gFont2 <- MakeFont(gPEBLBaseFont,0,24,MakeColor("white"),MakeColor("black"),1)
##Global flag keeping track of whether the trial was aborted.
gReset <- 0
}

define InitPoles()

```

```

{
##Make hot zones for each stack.

footerfont <- MakeFont(gPEBLBaseFont,0,20,MakeColor("grey"),MakeColor("black"),1)

gFooter <- MakeLabel(gFooterLab,footerfont)
AddObject(gFooter,gWin)
Move(gFooter,300,490)

height <- First(gPoleHeights)*30+15
y <- 470-height/2
r1 <- Rectangle(220,y,110,height,MakeColor("grey20"),1)
height <- Second(gPoleHeights)*30+15
y <- 470-height/2
r2 <- Rectangle(340,y,110,height,MakeColor("grey20"),1)
height <- Third(gPoleHeights)*30+15
y <- 470-height/2
r3 <- Rectangle(460,y,110,height,MakeColor("grey20"),1)

AddObject(r1,gWin)
AddObject(r2,gWin)
AddObject(r3,gWin)

##if we have a 'reset' button, add one.
if(gAllowReset)
{
  r0 <- Rectangle(340,gFooter.y+50,300,40,MakeColor("grey20"),0)
AddObject(r0,gWin)
  gZones <- [r0,r1,r2,r3]
gResetLabel <- MakeLabel(gResetText,footerfont)
  Move(gResetLabel,340,gFooter.y+50)
AddObject(gResetLabel,gWin)
} else {
  gZones <- [r1,r2,r3]
}

gDisks <- [MakeImage("png/disk1.png"),MakeImage("png/disk2.png"),
  MakeImage("png/disk3.png"),MakeImage("png/disk4.png"),
  MakeImage("png/disk5.png"),MakeImage("png/disk6.png")]
gDisks2 <- [MakeImage("png/disk1.png"),MakeImage("png/disk2.png"),
  MakeImage("png/disk3.png"),MakeImage("png/disk4.png"),
  MakeImage("png/disk5.png"),MakeImage("png/disk6.png")]

if(gShowHand)
{
  gHand <- MakeImage("png/hand.png")
  AddObject(gHand,gWin)
  Move(gHand,330,225)
}
gRect <- Rectangle(330,80,380,140,MakeColor("grey20"),1)
AddObject(gRect,gWin)

gHeader <- MakeTextBox(gHeaderLab,footerfont,105,95)
AddObject(gHeader,gWin)
Move(gHeader,30,50)
##Keep track of the image complex:
gComplex <- []

gFileOut <- FileOpenAppend("Resultados/tol-"+gsubnum+".csv")
gFileOutSum <- FileOpenAppend("Resultados/tol-summary-"+gsubnum+".csv")

}

define PlotDisks(config,inhand,ybase,disks)
{
  x <- 100
  complex <- []

```

```

loop(column,config)
{
  y <- ybase
  x <- x + 120
  loop(i,column)
  {
    disk <- Nth(disks,i)
    AddObject(disk,gWin)
    Move(disk,x, y)
    Show(disk)
    y <- y - 25
    complex <- Append(complex,disk)
  }
}
if(inhand >0)
{
  Move(Nth(disks,inhand),320,250)
  AddObject(Nth(disks,inhand),gWin)
}
return complex
}

define PlotDisksAnimate(config,inhand,ybase,disks,steps)
{

if(inhand > 0)
{
  diskIDs <- Flatten(Append(config,inhand))
}else{
  diskIDs <- Flatten(config)
}

disks2 <- SubSet(disks,Sort(diskIDs)) #Get the disks objects, in order

startxy <- []
loop(i, disks2)
{
  startxy <- Append(startxy, [i.x,i.y])
}

x <- 100
complex <- []

##Go through each column of the config string
endxy <- []
i <- 1
loop(column,config)
{
  y <- ybase
  x <- x + 120
  loop(i,column)
  {
    disk <- Nth(disks,i)
    AddObject(disk,gWin)
    Show(disk)

    pos <- [x,y]
    endxy <- Append(endxy,pos)
    y <- y - 25
    complex <- Append(complex,disk)
  }

  i <- i + 1
}
if(inhand > 0 )
{
  endXY <- Append(endXY, [320,225])
}
}

```

```

    disk <- Nth(disks,inhand)
    AddObject(disk,gWin)
    Show(disk)
  }
endXY <- SortBy(endXY,diskIDs)
startend <- Transpose([startxy, endxy])
stepxy <- []

loop(i, startend)
{
  start <- First(i)
  end <- Second(i)
  stepxy <- Append(stepxy,[(First(end)-First(start))/steps,
    (Second(end)-Second(start))/steps])
}

diskstep <- Transpose([disks2,endxy,stepxy])
step <- 1
while(step <= steps)
{
  loop(i, diskstep)
  {
    disk <- First(i)
    end <- Second(i)
    diff <- Third(i)

    disk.x <- First(end) - First(diff) * (steps-step)
    disk.y <- Second(end) - Second(diff) * (steps-step)
  }

  Draw()
  step <- step + 1
}

return complex
}

define MoveDisk(config)
{

  ##We need a different target set when reset button is enabled.
  if(gAllowReset)
  {
    out <- Sequence(0,3,1)
  } else {
    out <- Sequence(1,3,1)
  }

  reset <- 0
  ##We need to use separate code for timer/non-timer.
  if(gUseTimer)
  {
    cont <- 1
    while(cont)
    {
      first <- WaitForClickOnTargetWithTimeout(gZones, out,100)
      if(first=="<timeout>")
      {
        timeleft <- (gEndTime-GetTime())/1000
        if(timeleft < 0)
        {
          gTimeRanOut <- 1
          cont <- 0
        }
        UpdateGraph(gTimegraph,Max([0,timeleft]), 710,400,gTimeLimit)
        Draw()
      } else {
        if(first ==0)
        {
          ##We hit the 'abort' button.
          cont <- 0
          reset <- 1
        }
      }
    }
  }
}

```

```

    } else {
      ##Time is not out; but we need to make sure there is
      ##something in the pile
      if(Length(Nth(config,first))>0)
      {
        cont <- 0
      }
    }
  }
}

} else {
#####
##Non-timer branch, so use just standard click.

first <- WaitForClickOnTarget(gZones, out)

if(first == 0)
{
  reset <- 1
} else {
  while(Length(Nth(config,first))==0)
  {
    first <- WaitForClickOnTarget(gZones, out)
  }
}

##At this point, we either
##1. ran out of time: gtimeranout == 1
##2. picked up a valid disk gtimeranout == 0, reset == 0
##3. hit the reset button. reset == 1,

##Valid click:
if((not gTimeranout) and reset == 0)
{
  x <- RemoveDisk(first,config)

  config <- First(x)
  disk <- Nth(x,2)

  if(disk != 0)
  {
    ##Remove the disk complex so it can be re-drawn
    RemoveComplex(gComplex)
    gComplex <- PlotDisksAnimate(config,disk,450,gDisks,10)
    Draw()

#####
#####
## Now, get the pile to place the disk in.

cont <- 1
while(cont)
{
  #Now, get the target pile 1-3

  if(gUseTimer)
  {
    timecont <- 1
    while(timecont)
    {
      next <- WaitForClickOnTargetWithTimeout(gZones, out,100)

      if(next=="<timeout>")
      {
        timeleft <- (gEndTime-GetTime())/1000
        UpdateGraph(gTimegraph,Max([0,timeleft]), 710,400,gTimeLimit)
        Draw()

```

```

    if(timeleft < 0)
    {
        gTimeRanOut <- 1
    }

} else {

    ##We hit the 'abort' button
    if(next == 0)
    {
        reset <- 1
    }

    timecont <- 0
}
}

} else {
    ##No timer, so use just standard click.
    next <- WaitForClickOnTarget(gZones, out)
}

    ##Check to see if the new configuration violates the pole height
    ## limitation.
    if(Length(Nth(config ,next))<Nth(gPoleHeights,next))
    {
        cont <- 0
    }
}

RemoveComplex(gComplex)
config <- AddDisk(config,next,disk)
gComplex <- PlotDisksAnimate(config,0,450,gDisks,5)
Draw()
}
}

if(reset)
{
    gReset <- 1
}

return config
}

define RemoveComplex(x)
{
    loop(i,x)
    {
        RemoveObject(i,gWin)
    }
}

##Returns a string describing a state of the problem
define MakeStateString(config)
{

string <- "|"
loop(i,config)
{
    loop(j,i)
    {
        string <- string + j
    }
    string <- string + "|"
}

return string
}

```

```

##This works just on the data structure, not on the display.
define RemoveDisk(col,ind)
{
##Removing:

removed <- 0
a <- 1
newcomp <- []
loop(i,ind)
{

if(a == col)
{
l <- Length(i)
## Remove the last element of the list, if it exists
if(l == 0)
{
newcomp <- Append(newcomp,[])
} elseif(l==1){

newcomp <- Append(newcomp,[])
removed <- First(i)

}else {

##
newcomp <- Append(newcomp, SubList(i,1,l-1))
removed <- Nth(i,l)

}

} else {
newcomp <- Append(newcomp,i)
}

a <- a +1
}

return [newcomp,removed]
}

define AddDisk(config,nextcol,disk)
{

a <- 1
newcomp <- []
loop(i,config)
{
if(a == nextcol)
{
newcomp <- Append(newcomp, Append(i,disk))

} else {
newcomp <- Append(newcomp,i)
}

a <- a +1
}

return newcomp
}

##Makes a disk configuration, with the restriction that an individual
##stack must be shorter than its relevant pole height.
define ShuffleDisks(num)
{
order <- Shuffle(Sequence(1,num,1))
config <- [[],[],[]]

loop(i,order)
{
cont <- 1
while(cont)
{

```

```

stack <- RandomDiscrete(3) #is there room in the stack?
if(Length(Nth(config,stack)) < Nth(gPoleHeights,stack))
{
  cont <- 0
}
}

config <- AddDisk(config,stack,i)
}

return config
}

define StackEqual(stack1,stack2)
{
  equal <- 1

  loop(x, Transpose([stack1,stack2]))
  {
    a <- First(x)
    b <- Nth(x,2)

    if(Length(a) == Length(b) )
    {
      loop(j,Transpose([a,b]))
      {
        if(First(j) != Nth(j,2))
        {
          equal <- 0
        }
      }
    }
    } else {
      equal <- 0
    }
  }
  return equal
}

define Phillips1999(set)
{
  ##Don't use the timer or turn limits here.
  gUseTimer <- 0
  gUseTurnLimit <- 0

  gPoleHeights <- [5,5,5]

  trialsA <- [ [[[],[4,3,2,1],[5]], [[2,3],[4],[5,1]] ,-1,0,0],
               [[5,3,2,4,1],[,]], [[],[1,4,3],[2,5]],-1,0,0],
               [[4],[3,2],[5,1]], [[2,4,3,1],[,],[5]],-1,0,0],
               [[3],[1,2,5],[4]], [[],[4,3],[2,5,1]],-1,0,0,#orig. wiert
               [[],[3,5],[1,4,2]], [[3,5],[2],[1,4]] ,-1,0,0, #original wiert
               [[4,3],[5],[2,1]], [[3,5],[2],[1,4]] ,-1,0,0],
               [[2],[5,3],[1,4]], [[],[5,1,4,2,3]],-1,0,0],
               [[1,2,5,3,4],[,],[,]], [[2,3],[,],[5,4,1]],-1,0,0] #orig goal wiert

  trialsB <- [ [[4],[2,1,5],[3]], [[4,5,1,3],[2],[,]],-1,0,0],
               [[],[2,3,4,5],[1]], [[5,1,4],[,],[3,2]],-1,0,0],
               [[],[2],[4,3,1,5]], [[2,1,5,3],[,],[4]],-1,0,0],
               [[],[1,3],[4,5,2]], [[3,4,2],[1,5],[,]],-1,0,0, #orig goal wiert
               [[5,1],[,],[3,4,2]], [[2,3,4,5],[1],[,]],-1,0,0],
               [[],[,],[2,1,5,3,4]], [[1],[,],[5,3,4,2]],-1,0,0],
               [[],[3,2,4,1,5],[,]], [[5],[1,2,3],[4]],-1,0,0],
               [[],[2,5,3,4,1],[,]], [[3,2],[1],[5,4]],-1,0,0 ]

  trialsC <- [ [[4,3],[,],[2,1,5]], [[4],[5,1,3],[2]],-1,0,0],
               [[4,1],[5,3,2],[,]], [[2,1,4,5],[,],[3]],-1,0,0, #original wiert
               [[1,3],[4,5,2],[,]], [[,],[4,3,1,5],[2]],-1,0,0],
               [[2],[,],[5,4,3,1]], [[,],[2,1],[5,3,4]],-1,0,0, #orig goal wiert
               [[4,2,1],[3,5],[,]], [[,],[1],[5,2,3,4]],-1,0,0],
               [[5],[1,4,3],[2]], [[,],[2,5,3],[1,4]],-1,0,0],

```

```

[[[4,1,2,5],[],[3]], [[3,1,4],[],[5,2]],-1,0,0],
[[[],[],[5,1,3,2,4]], [[],[],[2,4,1,5,3]],-1,0,0]]

if(set == "A")
{
ret <- trialsA
} elseif(set=="B")
{
ret <- trialsB
} else {
ret <- trialsC
}
return ret
}

define Shallice()
{

##Don't use the timer or turn limits here.
gUseTimer <- 0
gUseTurnLimit <- 1

gPoleHeights <- [3,2,1]

## 1=RED, 2=YELL, 3=BLUE
##
##

s <- [
[[[2,1],[3],[],[],[3,2],[1]],2,63,66 ],
[[[2,1],[3],[],[2],[1],[3]] ,2,63,55],
[[[2,1],[3],[],[2,3],[1],[1]] ,3,63,53],
[[[2,1],[3],[],[2],[1,3],[1]] ,3,63,54],
[[[2,1],[3],[],[1,2],[3],[1]] ,4,63,13],
[[[2,1],[3],[],[3],[1,2],[1]] ,4,63,44],
[[[2,1],[3],[],[1,3],[1],[2]] ,4,63,22],
[[[2,1],[3],[],[1],[1,3],[2]] ,4,63,46],
[[[2,1],[3],[],[1,2,3],[1],[1]] ,5,63,11],
[[[2,1],[3],[],[1,3,2],[1],[1]] ,5,63,21],
[[[2,1],[3],[],[3,2],[1],[1]] ,5,63,43],
[[[2,1],[3],[],[3],[1],[2]] ,5,63,45]]

return s
}

###This mirror-reverses the stimuli and stacks.
define ReverseStimuli(stim)
{
gPoleheights <- Reverse(gPoleHeights)
newstim <- []
loop(i,stim)
{
line <- [Reverse(First(i)), Reverse(Second(i)), Third(i),Fourth(i),Fifth(i))
newstim <- Append(newstim,line)
}

return newstim
}

define GetTOLDist(matrix,a,b,c,d)
{
row <- (a-1)* 6 + b
col <- (c-1)* 6 + d
return ToNumber(M(matrix,row,col))
}

define AssembleTrial(a1,a2,b1,b2,states,dists)
{
return [M(states,a1,a2),M(states,b1,b2), GetTOLDist(dists,a1,a2,b1,b2), a1+""+a2, b1+""+b2 ]
}

```

```

## Computes the minimum distance between states, courtes of Fimbel
## http://sites.google.com/site/tolospace/
##
define StateDist()
{
  state <- FoldList([
0,1,1,2,2,3,4,3,3,3,2,3,6,6,5,4,5,4,7,6,7,8,7,8,6,5,6,7,7,8,6,6,5,4,4,3,
1,0,1,2,2,3,3,3,2,2,1,2,5,5,4,3,4,3,6,5,6,7,6,7,6,5,6,7,7,8,6,6,5,4,4,3,
1,1,0,1,1,2,3,2,3,3,2,3,6,6,5,4,5,4,7,6,7,8,7,7,5,4,5,6,6,7,5,5,4,3,3,2,
2,2,1,0,1,2,3,2,3,4,3,4,7,7,6,5,6,5,8,7,8,7,6,4,3,4,5,5,6,4,4,3,3,2,1,
2,2,1,1,0,1,2,1,2,3,3,4,7,7,6,5,5,4,7,6,7,7,8,7,5,4,5,6,5,6,4,4,3,2,3,2,
3,3,2,2,1,0,3,2,3,4,4,5,8,8,7,6,6,5,8,7,7,6,7,6,4,3,4,5,4,5,3,3,2,1,2,3,
4,3,3,3,2,3,0,1,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,7,6,7,8,7,8,6,6,5,4,5,4,
3,3,2,2,1,2,1,0,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,6,5,6,7,6,7,5,5,4,3,4,3,
3,2,3,3,2,3,1,1,0,1,1,2,5,5,4,3,3,2,5,4,5,6,6,7,7,6,7,8,7,7,6,6,5,4,5,4,

3,2,3,4,3,4,2,2,1,0,1,2,4,4,3,3,2,1,4,3,4,5,5,6,8,7,8,7,7,6,7,7,6,5,6,5,
2,1,2,3,3,4,2,2,1,1,0,1,4,4,3,2,3,2,5,4,5,6,5,6,7,6,7,7,8,7,7,6,5,5,4,
3,2,3,4,4,5,3,3,2,2,1,0,3,3,2,1,2,3,4,3,4,5,4,5,8,7,7,6,7,6,8,8,7,6,6,5,
6,5,6,7,7,8,6,6,5,4,4,3,0,1,1,2,2,3,4,3,3,2,3,6,6,5,4,5,4,7,6,7,8,7,8,
6,5,6,7,7,8,6,6,5,4,4,3,1,0,1,2,2,3,3,3,2,2,1,2,5,5,4,3,4,3,6,5,6,7,6,7,
5,4,5,6,6,7,5,5,4,3,3,2,1,1,0,1,1,2,3,2,3,2,3,6,6,5,4,5,4,7,6,7,8,7,7,
4,3,4,5,5,6,4,4,3,3,2,1,2,2,1,0,1,2,3,2,3,4,3,4,7,7,6,5,6,5,8,7,8,7,7,6,
5,4,5,6,5,6,4,4,3,2,3,2,2,1,1,0,1,2,1,2,3,3,4,7,7,6,5,5,4,7,6,7,7,8,7,
4,3,4,5,4,5,3,3,2,1,2,3,3,3,2,2,1,0,3,2,3,4,4,5,8,8,7,6,6,5,8,7,7,6,7,6,
7,6,7,8,7,8,6,6,5,4,5,4,4,3,3,2,3,0,1,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,
6,5,6,7,6,7,5,5,4,3,4,3,3,2,2,1,2,1,0,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,
7,6,7,8,7,7,6,6,5,4,5,4,3,2,3,3,2,3,1,1,0,1,1,2,5,5,4,3,3,2,5,4,5,6,6,7,
8,7,8,7,7,6,7,7,6,5,6,5,3,2,3,4,3,4,2,2,1,0,1,2,4,4,3,3,2,1,4,3,4,5,5,6,
7,6,7,7,8,7,7,6,5,5,4,2,1,2,3,3,4,2,2,1,1,0,1,4,4,3,2,3,2,5,4,5,6,5,6,
8,7,7,6,7,6,8,8,7,6,6,5,3,2,3,4,4,5,3,3,2,2,1,0,3,3,2,1,2,3,4,3,4,5,4,5,
6,6,5,4,5,4,7,6,7,8,7,8,6,5,6,7,7,8,6,6,5,4,4,3,0,1,1,2,2,3,4,3,3,3,2,3,
5,5,4,3,4,3,6,5,6,7,6,7,6,5,6,7,7,8,6,6,5,4,4,3,1,0,1,2,2,3,3,3,2,2,1,2,
6,6,5,4,5,4,7,6,7,8,7,7,5,4,5,6,6,7,5,5,4,3,3,2,1,1,0,1,1,2,3,2,3,3,2,3,
7,7,6,5,6,5,8,7,8,7,7,6,4,3,4,5,5,6,4,4,3,3,2,1,2,2,1,0,1,2,3,2,3,4,3,4,
7,7,6,5,5,4,7,6,7,7,8,7,5,4,5,6,5,6,4,4,3,2,3,2,2,2,1,1,0,1,2,1,2,3,3,4,
8,8,7,6,6,5,8,7,7,6,7,6,4,3,4,5,4,5,3,3,2,1,2,3,3,3,2,2,1,0,3,2,3,4,4,5,
6,6,5,4,4,3,6,5,6,7,7,8,7,6,7,8,7,8,6,6,5,4,5,4,4,3,3,3,2,3,0,1,1,2,2,3,
6,6,5,4,4,3,6,5,6,7,7,8,6,5,6,7,6,7,5,5,4,3,4,3,3,2,2,1,2,1,0,1,2,2,3,
5,5,4,3,3,2,5,4,5,6,6,7,7,6,7,8,7,7,6,6,5,4,5,4,3,2,3,3,2,3,1,1,0,1,1,2,
4,4,3,3,2,1,4,3,4,5,5,6,8,7,8,7,7,6,7,6,5,6,5,3,2,3,4,3,4,2,2,1,0,1,2,
4,4,3,2,3,2,5,4,5,6,5,6,7,7,8,7,7,6,5,5,4,2,1,2,3,3,4,2,2,1,1,0,1,
3,3,2,1,2,3,4,3,4,5,4,5,8,7,7,6,7,6,8,8,7,6,6,5,3,2,3,4,4,5,3,3,2,2,1,0],36)
  return state
}

define FimbelOld()
{
  ##Don't use the timer or turn limits here.
  gUseTimer <- 0
  gUseTurnLimit <- 0

  gPoleHeights <- [1,2,3]

  ## 1=RED, 2=YELL, 3=BLUE
  ##
  ##

  ##THIS IS ALL THE STATES
  s <- ([[[]],[1,2,3]],[[3],[1,2]],[[3],[1,2]],[[3,2],[1]],[[2],[3],[1]],[[2],[3,1],[[]],
[[[]],[1,3,2]],[[2],[1,3]],[[[]],[2],[1,3]],[[[]],[2,3],[1]],[[3],[2],[1]],[[3],[2,1],[[]],
[[[]],[3,1,2]],[[2],[3,1]],[[[]],[2],[3,1]],[[[]],[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[[]],
[[[]],[3,2,1]],[[1],[3,2]],[[[]],[1],[3,2]],[[[]],[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[[]],
[[[]],[2,3,1]],[[1],[2,3]],[[[]],[1],[2,3]],[[[]],[1,3],[2]],[[3],[1],[2]],[[3],[1,2],[[]],
[[[]],[2,1,3]],[[3],[2,1]],[[[]],[3],[2,1]],[[[]],[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[[]]]

  ##This is the minimal distance between states:
  dists <- StateDist()

```

```

trialsOld <- [AssembleTrial(1,2,1,3,s,dists),
              AssembleTrial(5,3,5,2,s,dists),
              AssembleTrial(2,3,2,4,s,dists),
              AssembleTrial(1,2,1,5,s,dists),
              AssembleTrial(2,3,3,6,s,dists),

              AssembleTrial(5,3,6,5,s,dists),
              AssembleTrial(2,3,3,5,s,dists),
              AssembleTrial(5,3,6,4,s,dists),
              AssembleTrial(1,2,1,6,s,dists),
              AssembleTrial(5,3,1,6,s,dists),

              AssembleTrial(1,2,6,4,s,dists),
              AssembleTrial(2,3,3,3,s,dists),
              AssembleTrial(1,2,6,3,s,dists),
              AssembleTrial(5,3,1,5,s,dists),
              AssembleTrial(2,3,3,2,s,dists)]

return trialsOld
}

define FimbelYoung()
{
  ##Don't use the timer or turn limits here.
  gUseTimer <- 0
  gUseTurnLimit <- 0

  gPoleHeights <- [1,2,3]

  ## 1=RED, 2=YELL, 3=BLUE
  ##
  ##

  ##THIS IS ALL THE STATES
  s <- [[[[[],[],[1,2,3]],[[3],[1],[1,2]],[[1],[3],[1,2]],[[1],[3,2],[1]],[[2],[3],[1]],[[2],[3,1],[1]],
        [[1],[1,3,2]],[[2],[1],[1,3]],[[1],[2],[1,3]],[[1],[2,3],[1]],[[3],[2],[1]],[[3],[2,1],[1]],
        [[1],[1,3,1,2]],[[2],[1],[3,1]],[[1],[2],[3,1]],[[1],[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[1]],
        [[1],[3,2,1]],[[1],[1],[3,2]],[[1],[1],[3,2]],[[1],[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[1]],
        [[1],[1],[2,3,1]],[[1],[1],[2,3]],[[1],[1],[2,3]],[[1],[1,3],[2]],[[3],[1],[2]],[[3],[1,2],[1]],
        [[1],[1],[2,1,3]],[[3],[1],[2,1]],[[1],[3],[2,1]],[[1],[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[1]] ]
  dists <- StateDist()

  trialsYoung <- [
    AssembleTrial(2,3,1,2,s,dists),
    AssembleTrial(2,1,1,5,s,dists),
    AssembleTrial(1,4,2,2,s,dists),
    AssembleTrial(2,6,1,2,s,dists),
    AssembleTrial(5,6,4,3,s,dists),

    AssembleTrial(3,3,4,4,s,dists),
    AssembleTrial(3,4,4,3,s,dists),
    AssembleTrial(5,3,6,3,s,dists),
    AssembleTrial(6,3,5,3,s,dists),
    AssembleTrial(6,4,1,4,s,dists),

    AssembleTrial(4,6,3,4,s,dists),
    AssembleTrial(2,2,6,5,s,dists),
    AssembleTrial(3,3,2,3,s,dists),
    AssembleTrial(2,2,3,4,s,dists),
    AssembleTrial(6,5,1,1,s,dists),

    AssembleTrial(2,2,6,2,s,dists),
    AssembleTrial(3,1,2,3,s,dists),
    AssembleTrial(5,2,3,2,s,dists),
    AssembleTrial(4,4,2,6,s,dists),
    AssembleTrial(3,2,2,3,s,dists),

    AssembleTrial(2,1,3,1,s,dists),
    AssembleTrial(3,2,1,1,s,dists),
    AssembleTrial(2,3,4,5,s,dists),
    AssembleTrial(2,2,3,1,s,dists),
  ]
}

```

```

    AssembleTrial(4,1,1,2,s,dists),

    AssembleTrial(1,4,3,1,s,dists),
    AssembleTrial(2,5,4,3,s,dists),
    AssembleTrial(3,6,6,5,s,dists),
    AssembleTrial(4,6,1,5,s,dists),
    AssembleTrial(6,3,3,1,s,dists),

    AssembleTrial(2,2,4,6,s,dists),
    AssembleTrial(2,4,5,1,s,dists),
    AssembleTrial(2,6,6,2,s,dists),
    AssembleTrial(3,5,6,5,s,dists),
    AssembleTrial(5,6,2,1,s,dists)]

return trialsYoung
}

## ShalliceRandom uses the fimbel space.
##
define ShalliceRandom(numtrials)
{

##Don't use the timer or turn limits here.
gUseTimer <- 0
gUseTurnLimit <- 0

gPoleHeights <- [1,2,3]

## 1=RED, 2=YELL, 3=BLUE
##
##

##THIS IS ALL THE STATES
s <- [[[[[],[],[1,2,3]],[[3],[],[1,2]],[[],[3],[1,2]],[[],[3,2],[1]],[[2],[3],[1]],[[2],[3,1],[[]],
[[[],[],[1,3,2]],[[2],[],[1,3]],[[],[2],[1,3]],[[],[2,3],[1]],[[3],[2],[1]],[[3],[2,1],[[]]],
[[[],[],[3,1,2]],[[2],[],[3,1]],[[],[2],[3,1]],[[],[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[[]]],
[[[],[],[3,2,1]],[[1],[],[3,2]],[[],[1],[3,2]],[[],[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[[]]],
[[[],[],[2,3,1]],[[1],[],[2,3]],[[],[1],[2,3]],[[],[1,3],[2]],[[3],[1],[2]],[[3],[1,2],[[]]],
[[[],[],[2,1,3]],[[3],[],[2,1]],[[],[3],[2,1]],[[],[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[[]]] ]
dists <- StateDist()

##Don't use the timer or turn limits here.
gUseTimer <- 0
gUseTurnLimit <- 0

##Generates a random set of trials using Shallice's original 123
##stack heights

id <- Sequence(1,6,1)

trials <- []
i <- 1
while(i <= numtrials)
{
  a1 <- Pick(id)
  a2 <- Pick(id)
  b1 <- Pick(id)
  b2 <- Pick(id)

##Make sure the two stacks are not identical at outset:
while(GetTOLDist(dists,a1,a2,b1,b2)<2) # Make sure you need at
# least 2 moves
{
  a1 <- Pick(id)
  a2 <- Pick(id)
  b1 <- Pick(id)
  b2 <- Pick(id)
}

trials <- Append(trials,AssembleTrial(a1,a2,b1,b2,s,dists))

```

```

    i <- i + 1
  }
  return trials
}

define TOLDX()
{
  ##Culbertson, W. C., & Zillmer, E. A. (1998). The
  ##Tower of LondonDX: A Standardized Approach to Assessing Executive
  ##Functioning in Children. Archives of Clinical Neuropsychology, 13(3),
  ##285-301. doi:10.1016/S0887-6177(97)00033-4

  ##Don't use the timer or turn limits here.
  gUseTimer <- 0
  gUseTurnLimit <- 1
  gUseReverse <- 1 ## Mirror reverse all stimuli and stacks.

  gPoleHeights <- [1,2,3]

  ## 1=RED, 2=YELL/(green), 3=BLUE
  ##
  ##

  ##THIS IS ALL THE STATES
  s <- [ [[[],[],[1,2,3]], [[3],[],[1,2]], [[],[3],[1,2]], [[],[3,2],[1]], [[2],[3],[1]], [[2],[3,1],[[]]],
        [[[],[],[1,3,2]], [[2],[],[1,3]], [[],[2],[1,3]], [[],[2,3],[1]], [[3],[2],[1]], [[3],[2,1],[[]]],
        [[[],[],[3,1,2]], [[2],[],[3,1]], [[],[2],[3,1]], [[],[2,1],[3]], [[1],[2],[3]], [[1],[2,3],[[]]],
        [[[],[],[3,2,1]], [[1],[],[3,2]], [[],[1],[3,2]], [[],[1,2],[3]], [[2],[1],[3]], [[2],[1,3],[[]]],
        [[[],[],[2,3,1]], [[1],[],[2,3]], [[],[1],[2,3]], [[],[1,3],[2]], [[3],[1],[2]], [[3],[1,2],[[]]],
        [[[],[],[2,1,3]], [[3],[],[2,1]], [[],[3],[2,1]], [[],[3,1],[2]], [[1],[3],[2]], [[1],[3,2],[[]]] ]

  dists <- StateDist()

  trials <- [AssembleTrial(6,3,5,3,s,dists),
            AssembleTrial(6,3,1,5,s,dists),
            AssembleTrial(6,3,5,4,s,dists),
            AssembleTrial(6,3,1,3,s,dists),
            AssembleTrial(6,3,4,4,s,dists),

            AssembleTrial(6,3,4,6,s,dists),
            AssembleTrial(6,3,4,5,s,dists),
            AssembleTrial(6,3,1,2,s,dists),
            AssembleTrial(6,3,4,3,s,dists),
            AssembleTrial(6,3,2,4,s,dists),

            AssembleTrial(6,3,4,2,s,dists),
            AssembleTrial(6,3,2,5,s,dists),
            AssembleTrial(6,3,3,3,s,dists),
            AssembleTrial(6,3,3,1,s,dists),
            AssembleTrial(6,3,3,5,s,dists)]

  return trials
}

define Pick(list)
{
  return First(Shuffle(list))
}

define ProgressiveDisks(numdisks,numtrials)
{
  gUseTimer <- 0
  gUseTurnLimit <- 0

  ##Generates a random set of trials using Shallice's original 123
  ##stack heights

```

```

gPoleHeights <- [6,6,6]

trials <- []

loop(size,numdisks)
{
  i <- 1
  while(i <= numtrials)
  {
    config1 <- ShuffleDisks(size)
    config2 <- ShuffleDisks(size)

    ##Make sure the two stacks are not identical at outset:
    while(StackEqual(config1,config2))
    {
      config1 <- ShuffleDisks(size)
      config2 <- ShuffleDisks(size)
    }

    trials <- Append(trials, [config1,config2,-1,0,0])
    i <- i + 1
  }
}
return trials
}

define RandomDisks(numdisks,numtrials)
{
  ##Generates a random set of trials using Shallice's original 123
  ##stack heights

  tmp <- ProgressiveDisks(numdisks,numtrials)
  return Shuffle(tmp)
}

define M(m,i,j)
{
  return Nth(Nth(m,i),j)
}

define MakeGraph(remaining,total,x,y,label,numticks)
{
  ##Sets up some basic parameters for plotting the graph at the bottom of the screen
  font <- MakeFont(gPEBLBaseFont,0,12,MakeColor("grey80"),MakeColor("black"),0)
  height <- 250
  dataheight <- remaining/total * height
  width <- 20
  scale <- height/total    ##pixels/unit

  ycenter <- y
  xcenter <- x

  bgcol <- MakeColor("grey20")
  # fgcol <- MakeColor("red")
  fgcol <- MakeColorRGB(170,0,0)
  black <- MakeColor("grey80")

  #Create rectangle for background and for data
  back <- Rectangle(xcenter,ycenter, 70, height+50, bgcol, 1)
  data <- Rectangle(xcenter,ycenter + height/2 - dataheight/2, width, dataheight, fgcol, 1)
  lab <- MakeLabel(label,font)
  Move(lab,xcenter, ycenter+height/2+14)

  AddObject(back,gWin)
}

```

```

AddObject(data,gWin)
AddObject(lab,gWin)
##Create all the labels, based on yvals. Add each label to the
##appropriate location, and combine them all into a single
##composite to make removal easier.
##These labels are all re-created each trial--they could be
##static, which would be a little more efficient, but it is really
##fast as is.
composite <-[back,data,lab]

yvals <- Sequence(0, total, total/numticks)

loop(i,yvals)
{
  yy <- MakeLabel(i+"" , font)
  AddObject(yy,gWin)
  Move(yy, xcenter-20, ycenter + height/2 - i*scale)
  Show(yy)
  composite <- Append(composite,yy)
  tick <- Line(Floor(xcenter-width/2),ycenter+height/2-i*scale,width,0,black)
  AddObject(tick,gWin)
  composite <- Append(composite,tick)
}
Draw()

return composite
}

define UpdateGraph(myGraph, remaining, x,y, total)
{

##Sets up some basic parameters for plotting the graph at the bottom of the screen

height <- 250
dataheight <- remaining/total * height

ycenter <- y
xcenter <- x

#Extract the data from graph.
data <- Second(myGraph)
data.height <- dataheight
data.y <- Floor(ycenter + height/2 -dataheight/2)

}

##targetlist is a set of graphical objects,
##keylist is a set of keys whose corresponding
##value should be returned when a graphical object is clicked upon.

define WaitForClickOnTargetWithTimeout(targetlist,keylist,time)
{

ret <- ""
testlist <- Transpose([targetlist,keylist])

wait1 <- 1
while(wait1)
{
  wait2 <- 1
  while(wait2)
  {
    pos <- WaitForMouseButtonWithTimeout(time)

    if(Length(pos)==4)
    {
      if(Nth(pos,4)=="<pressed>" and Nth(pos,3)==1)
      {
        wait2 <- 0
        gClick <- pos
      }
    }
  }
}
}

```

```

    }
  } else {
    wait2 <- 0
    wait1 <- 0

    gClick <- [-1,-1]
    ret <- "<timeout>"

  }
}

##only loop through the test if wait1 is 0; otherwise
##we timed out.
if(wait1)
{
  newtargs <- []

  loop(i,testlist)
  {
    if(Inside(pos,First(i)))
    {
      ret <- Nth(i,2)
      break
    }
  }
}
return ret
}

##
##
##

define TOLR()
{
  # Schnirman et al (1998) TOL-R

  ## Use the timer and turn limits here.
  gUseTimer <- 1
  gUseTurnLimit <- 1

  gPoleHeights <- [1,2,3]

  ## 1=RED, 2=YELL, 3=BLUE
  ##
  ##

  ##THIS IS ALL THE STATES
  s <- [[[[[],[],[1,2,3]],[[3],[],[1,2]],[[[],[3],[1,2]],[[[],[3,2],[1]],[[2],[3],[1]],[[2],[3],[1],[1]],
    [[[],[],[1,3,2]],[[2],[],[1,3]],[[[],[2],[1,3]],[[[],[2,3],[1]],[[3],[2],[1]],[[3],[2],[1],[1]],
    [[[],[],[3,1,2]],[[2],[],[3,1]],[[[],[2],[3,1]],[[[],[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[1]],
    [[[],[],[3,2,1]],[[1],[],[3,2]],[[[],[1],[3,2]],[[[],[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[1]],
    [[[],[],[2,3,1]],[[1],[],[2,3]],[[[],[1],[2,3]],[[[],[1,3],[2]],[[3],[1],[2]],[[3],[1,2],[1]],
    [[[],[],[2,1,3]],[[3],[],[2,1]],[[[],[3],[2,1]],[[[],[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[1]]]]]]]]

  ##This is the minimal distance between states:
  dists <- StateDist()

  trialsTOLR <- [
    AssembleTrial(1,1,6,5,s,dists),
    AssembleTrial(1,4,5,1,s,dists),
    AssembleTrial(3,5,1,2,s,dists),

    AssembleTrial(1,4,6,1,s,dists),

```

```

    AssembleTrial(1,4,5,3,s,dists),
    AssembleTrial(1,4,2,6,s,dists),

    AssembleTrial(6,3,4,4,s,dists),
    AssembleTrial(1,4,6,2,s,dists),
    AssembleTrial(6,3,4,6,s,dists),

    AssembleTrial(3,5,5,6,s,dists),
    AssembleTrial(3,5,5,4,s,dists),
    AssembleTrial(6,3,4,3,s,dists),

    AssembleTrial(1,1,3,5,s,dists),
    AssembleTrial(1,4,3,4,s,dists),
    AssembleTrial(6,3,4,5,s,dists),

    AssembleTrial(6,3,2,1,s,dists),
    AssembleTrial(6,3,1,2,s,dists),
    AssembleTrial(6,3,2,3,s,dists),

    AssembleTrial(1,4,5,5,s,dists),
    AssembleTrial(1,4,3,6,s,dists),
    AssembleTrial(1,1,6,1,s,dists),

    AssembleTrial(1,1,3,1,s,dists),
    AssembleTrial(6,3,2,5,s,dists),
    AssembleTrial(1,1,3,2,s,dists),

    AssembleTrial(6,3,4,1,s,dists),
    AssembleTrial(6,3,3,2,s,dists),
    AssembleTrial(1,4,4,6,s,dists),

    AssembleTrial(1,4,5,6,s,dists),
    AssembleTrial(2,1,5,1,s,dists),
    AssembleTrial(3,5,1,6,s,dists)]

return trialsTOLR
}

## trials from
## Anderson, P., Anderson, V., & Lajoie, G., (1996). The tower of
## London test: Validation and standardization for pediatric populations.
define TOLAnderson()
{
# Schnirman et al (1998) TOL-R

## Use the timer and turn limits here.
gUseTimer <- 1
gUseTurnLimit <- 1

gPoleHeights <- [1,2,3]

## 1=RED, 2=YELL, 3=BLUE
##
##

##THIS IS ALL THE STATES
s <- [[[[]],[1,2,3]],[[3],[1,2]],[[1,3],[1,2]],[[3,2],[1]],[[2],[3],[1]],[[2],[3,1],[1]],[
  [[[]],[1,3,2]],[[2],[1,3]],[[1,3]],[[2],[1,3]],[[2,3],[1]],[[3],[2],[1]],[[3],[2,1],[1]],[
  [[[]],[3,1,2]],[[2],[3,1]],[[2],[3,1]],[[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[1]],[
  [[[]],[3,2,1]],[[1],[3,2]],[[1],[3,2]],[[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[1]],[
  [[[]],[2,3,1]],[[1],[2,3]],[[1],[2,3]],[[1,1],[2,3]],[[3],[1],[2]],[[3],[1],[2]],[[3],[1,2],[1]],[
  [[[]],[2,1,3]],[[3],[2,1]],[[3],[2,1]],[[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[1]]]]]]]

##This is the minimal distance between states:
dists <- StateDist()

```

```

trialsTOLR <- [
  AssembleTrial(1,1,6,5,s,dists),
  AssembleTrial(1,4,5,1,s,dists),
  AssembleTrial(3,5,1,2,s,dists),

  AssembleTrial(1,4,6,1,s,dists),
  AssembleTrial(1,4,5,3,s,dists),
  AssembleTrial(1,4,2,6,s,dists),

  AssembleTrial(6,3,4,4,s,dists),
  AssembleTrial(1,4,6,2,s,dists),
  AssembleTrial(6,3,4,6,s,dists),

  AssembleTrial(3,5,5,6,s,dists),
  AssembleTrial(3,5,5,4,s,dists),
  AssembleTrial(6,3,4,3,s,dists),

  AssembleTrial(1,1,3,5,s,dists),
  AssembleTrial(1,4,3,4,s,dists),
  AssembleTrial(6,3,4,5,s,dists),

  AssembleTrial(6,3,2,1,s,dists),
  AssembleTrial(6,3,1,2,s,dists),
  AssembleTrial(6,3,2,3,s,dists),

  AssembleTrial(1,4,5,5,s,dists),
  AssembleTrial(1,4,3,6,s,dists),
  AssembleTrial(1,1,6,1,s,dists),

  AssembleTrial(1,1,3,1,s,dists),
  AssembleTrial(6,3,2,5,s,dists),
  AssembleTrial(1,1,3,2,s,dists),

  AssembleTrial(6,3,4,1,s,dists),
  AssembleTrial(6,3,3,2,s,dists),
  AssembleTrial(1,4,4,6,s,dists),

  AssembleTrial(1,4,5,6,s,dists),
  AssembleTrial(2,1,5,1,s,dists),
  AssembleTrial(3,5,1,6,s,dists)]

return trialsTOLR
}

## This provides basic translations. To translate to a new language, copy the text between
## elseif(lang== "es") and }else{
## and past before the final }else{, ensuring you have a single pair of matching brackets.
## Then translate each english string to your language of choice. Be sure to not include line breaks unless you mean it.
## also, make a 2-character
## label to specify your language. Then, change the 'Getstrings("en")' at the top of the file to
## be your language of choice. Finally, copy the function into an email and send to the pebl email list
## at pebl-list@lists.sourceforge.net, so others can enjoy your labor.

define GetStrings(language)
{
  lang <- Uppercase(language)
  if(lang == "EN")
  {
    gSelect <- "Type number of test to use:"+ CR(2) +
      "[1] Unconstrained pile heights, {3,4,5} disks, progressive difficulty, 24 trials"+CR(2)+
      "[2] Unconstrained pile heights, {3,4,5} disks, Random presentation, 24 trials"+CR(2)+
      "[3] Shallice test ([1,2,3] pile heights, 3 disks, Shallice's 12 problems."+CR(2)+
      "[4] Shallice pile heights [1,2,3], 3 disks, 30 random trials."+CR(2)+
      "[5] Phillips (1999) trials A (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" +CR(2)+
      "[6] Phillips (1999) trials B (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" +CR(2)+
      "[7] Phillips (1999) trials C (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" +CR(2)+
      "[8] Fimbel et al (2009) old: [1,2,3] pile heights, 3 disks, progressive difficulty, 15 trials"+CR(2)+
      "[9] Fimbel et al (2009) young: [1,2,3] pile heights, 3 disks, progressive difficulty, 35 trials"+CR(2)+
      "[10] TOL-R (Schnirman et al, 1998). [1,2,3] pile heights, 3 disks, 30 problems, time and move limit"+CR(2)+

```

```

[a] TOL-DX (Culbertson & Zillmer, 1998). [1,2,3] pile heights, 3 discs, 15 problems)+CR(3)+
>Edit TOL.pbl directly to specify a test configuration automatically."

gHeaderLab <- "Target"+CR(1)+"Stacks:"
gFooterLab <- "Click on pile to pick up and drop disk"

gInstText1 <- "You are about to perform a task called the 'Tower of London'. Your goal is to move a pile of disks from their original
configuration to the configuration shown on the top of the screen. You can only move one disk at a time, and you cannot move a disk onto a
pile that has no more room (indicated by the size of the grey rectangle). To move a disk, click on the pile you want to move a disk off of, and
it will move up above the piles. Then, click on another pile, and the disk will move down to that pile."
gInstText2 <- " You will have a time limit to finish each problem. Attempt to finish each problem within the allotted time."
gInstText3 <- " You will have only a limited number of moves to solve each problem. Before you make your first move, think about the
problem to make sure you can solve it within your move limit. If you do not finish the problem within the limit, the turn will end and you will
move on to the next problem."
gBegin <- "Click the mouse to begin."
gContinue <- "Trial Complete. "+CR(2)+ "Click mouse to continue to the next trial."
gIncomplete <- "Trial Incomplete. "
gClickLab <- "Clicks"
gTimeLab <- "Time (s)"
gMovesOutLab <- "You failed to complete the problem in the number of moves required."
gResetText <- "Reset problem"
gResetLab <- "Click screen to restart problem."
gTimeOutLab <- "You ran out of time."
gDebriefing <- "Thank you for participating. The experiment is now complete. Press any key to exit."
gContinueFail <- "Click screen to move on to the next problem."
} elseif(lang == "BR")
{
gSelect <- "Digite o número do teste a ser usado:"+ CR(2) +
"[1] Pilhas de alturas não-limitadas, {3,4,5} discos, dificuldade progressiva, 24 problemas"+CR(2)+
"[2] Pilhas de alturas não-limitadas, {3,4,5} discos, apresentação aleatória, 24 problemas"+CR(2)+
"[3] Teste de Shallice (pilhas de alturas [1,2,3], 3 discos, 12 problemas de Shallice."+CR(2)+
"[4] Pilhas nas alturas [1,2,3] de Shallice, 3 discos, 30 problemas aleatórios."+CR(2)+
"[5] Phillips (1999) trials A (pilhas não-limitadas, 5 discos, dificuldade progressiva, 8 problemas)" +CR(2)+
"[6] Phillips (1999) trials B (pilhas não-limitadas, 5 discos, dificuldade progressiva, 8 problemas)" +CR(2)+
"[7] Phillips (1999) trials C (pilhas não-limitadas, 5 discos, dificuldade progressiva, 8 problemas)" +CR(2)+
"[8] Fimbel et al (2009) antigo: [1,2,3] pile heights, 3 discos, dificuldade progressiva, 15 problemas"+CR(2)+
"[9] Fimbel et al (2009) novo: [1,2,3] pile heights, 3 discos, dificuldade progressiva, 35 problemas"+CR(2)+
"[0] TOL-R (Schnirman et al. 1998). Pilhas de alturas [1,2,3], 3 discos, 30 problemas, limite de tempo e movimentos)" +CR(2)+
[a] TOL-DX (Culbertson & Zillmer, 1998). novo: [1,2,3] pile heights, 3 discos, dificuldade progressiva, 15 problemas"+CR(3)+

"Edite TOL.pbl diretamente para especificar uma configuração automática do teste."

gHeaderLab <- "Posições-alvo"+CR(1)+"Pilhas:"
gFooterLab <- "Clique na pilha para escolher e soltar um disco"
gSeeDemo <- "Clique com o botão do mouse para ver um problema de demonstração"
gSeeDemo2 <- "Clique com o botão do mouse para continuar para o primeiro problema"
gInstText1 <- "Você está prestes a iniciar um teste chamado 'Torre de Londres'. Seu objetivo é mover a pilha de discos para ficar igual a
figura mostrada no alto da tela. Você pode mover apenas um disco por vez, e você não pode mover um disco para uma pilha que não tem mais
espaço. Para mover um disco, clique na pilha da qual você quer retirar um disco e ele ficará acima das pilhas. Então, clique em outra pilha e
o disco se moverá para essa pilha. Clique com botão do mouse para continuar."
gInstText2 <- " Você terá um limite de tempo para resolver cada problema. Tente terminar cada problema dentro do tempo estabelecido."
gInstText3 <- " Você terá apenas um número limitado de movimentos para resolver cada problema. Antes de fazer seu primeiro movimento,
pense sobre o problema para ter certeza de que você pode resolvê-lo dentro do seu limite de movimentos. Se você não terminar o problema
dentro do limite, sua vez terminará e você irá para o próximo problema."
gBegin <- "Clique com o botão do mouse para começar."
gContinue <- "Problema resolvido. "+CR(2)+ "Clique com o mouse para continuar para o próximo problema."
gIncomplete <- "Problema não-resolvido. "
gClickLab <- "Cliques"
gTimeLab <- "Tempo (s)"
gMovesOutLab <- "Você não conseguiu terminar o problema no número de movimentos necessários."
gResetText <- "Reiniciar o problema"
gResetLab <- "Você está começando o problema de novo."
gTimeOutLab <- "Você esgotou seu tempo."
gDebriefing <- "Obrigado por participar. O teste está agora terminado. Pressione qualquer tecla para sair."
gContinueFail <- "Clique com o mouse para continuar para o próximo problema."
} elseif(lang=="ES"){

## Note, this is not a proper Spanish translation.
##
gSelect <- "Type number of test to use:"+ CR(2) +
"[1] Unconstrained pile heights, {3,4,5} disks, progressive difficulty, 24 trials"+CR(2)+
"[2] Unconstrained pile heights, {3,4,5} disks, Random presentation, 24 trials"+CR(2)+
"[3] Shallice test ([1,2,3] pile heights, 3 disks, Shallice's 12 problems."+CR(2)+
"[4] Shallice pile heights [1,2,3], 3 disks, 30 random trials."+CR(2)+
"[5] Phillips (1999) trials A (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" +CR(2)+
"[6] Phillips (1999) trials B (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" +CR(2)+

```

```

"[7] Phillips (1999) trials C (unconstrained piles, 5 disks, progressive difficulty, 8 trials)" + CR(2) +
"[8] Fimbel et al (2009) old: [1,2,3] pile heights, 3 disks, progressive difficulty, 15 trials)" + CR(2) +
"[9] Fimbel et al (2009) young: [1,2,3] pile heights, 3 disks, progressive difficulty, 35 trials)" + CR(2) +
"[0] TOL-R (Schnirman et al, 1998). [1,2,3] pile heights, 3 disks, 30 problems, time and move limit)" + CR(2) +
"[a] TOL-DX (Culbertson & Zillmer, 1998). [1,2,3] pile heights, 3 discs, 15 problems)" + CR(3) +
"Edit TOL.pbl directly to specify a test configuration automatically."

gHeaderLab <- "Objetivo montÃ³ns:"
gFooterLab <- "Haga clic en pila para recoger y llevar a disco"
gInstText1 <- "EstÃ¡s a punto de realizar una tarea llamada 'Torre de Su Londres'. Objetivo es pasar un montÃ³n de discos desde su
configuraciÃ³n original a la configuraciÃ³n mostrada en la parte superior de la pantalla. SÃ³lo puedes mover un disco a la vez, y no se puede
mover un disco sobre un montÃ³n que no tiene mÃ¡s espacio (indicado por el tamaÃ±o del rectÃ¡ngulo gris). Para mover un disco, haga clic
en el montÃ³n que desea mover un disco fuera de, y se moverÃ¡ hacia arriba por encima de la pila. A continuaciÃ³n, haga clic en otra pila, y
el disco se moverÃ¡ hasta que el montÃ³n. "
gInstText2 <- "TendrÃ¡ un lÃmite de tiempo para terminar cada problema. Intento de terminar cada problema en el tiempo asignado."
gInstText3 <- " Usted tendrÃ¡ solamente un nÃmero limitado de movimientos para resolver cada problema. Antes de hacer su primer
movimiento, pensar en el problema para asegurarse de que puede resolver dentro de su lÃmite de movimiento. Si no termina el problema en
el lÃmite, el a su vez terminarÃ¡ y se trasladarÃ¡ al siguiente problema."
gBegin <- "Haga clic en el ratÃ³n para empezar."
gContinue <- "Prueba Completa. "+ CR(1) + " Haga clic en el ratÃ³n para continuar a la siguiente prueba. "
gIncomplete <- "Prueba Incompleta. "
gClickLab <- "Clicks"
gTimeLab <- "Tiempo (s)"
gMovesOutLab <- "Usted no pudo completar el problema del nÃmero de movimientos necesarios."
gResetText <- "Reset problem"
gResetLab <- "You are starting the problem again."
gTimeOutLab <- "Usted acabÃ³ el tiempo."
gDebriefing <- "Gracias por participar. El experimento ya estÃ¡ completa. Pulse cualquier tecla para salir."
gContinueFail <- " Haga clic en el ratÃ³n para continuar a la siguiente prueba. "

} else{
  Print("Language: ["+lang+"] unknown. Falling back on English translations.")
  GetStrings("br")
}
}

```

## ANEXO D- Programação Donkey

```
#####
## The PEBL Hungry Donkey Task, Version 0.1
## For use with PEBL 0.07 or later
## http://pebl.sf.net
## Part of The PEBL Psychological Testing Battery
## 2006-02 Released into Public Domain
## by Shane T. Mueller, Ph.D. (smueller at obereed dot net)
##
##
## This is an isomorph to the Bechara's "Iowa Gambling Task" (tm).
## It has been used to evaluate risk-seeking and gambling in children.
##
## As described by:
## Crone, E. A., & van der Molen, M. W. (2004). Developmental
## changes in real life decision making: Performance on a gambling task
## previously shown to depend on the ventomedial prefrontal cortex.
## Developmental neuropsychology, 25(3), 250-279.
##
##
## In this test, a keypress (cvbn) opens the door, and it remains
## open until the key is un-pressed. The software waits for all keys
## to be unpressed, which includes the numlock, capslock, etc. So,
## if the experiment 'freezes' with a door open, make sure capslock and
## numlock are off.
#####

define Start(IPar)
{

  gInputDevice <- "mouse" ##mouse or keyboard are valid.

  ## results are stored in tmpresults.dat
  MakeDirectory("data")
  gFileOut <-FileOpenWrite("data/donkey-" + gSubNum + ".csv")

  ##File header:
  FilePrint(gFileOut, "sub,trial,resp,reward,penalty,net,total,abstime,rt")

  gSleepEasy <- 1

  ## Initialize global objects for use later. Defined at end of file
  Initialize()
  gSubNum <- GetSubNum(gWin)

  if(gInputDevice == "keyboard")
  {
    ShowCursor(0)
    if(IsAnyKeyDown())
    {
      gInstructions.text <- "Please make sure no keys are pressed.
      Make sure that the numlock and scroll lock keys are untoggled."
      Show(gInstructions)
      Draw()

      WaitForAllKeysUp()
    }
  }
}

#####
## Now, generate stimuli/conditions
#####

deck1Pen <- Flatten(ShuffleRepeat([0,0,0,0,0,8,10,10,10,12],4))
deck2Pen <- Flatten(ShuffleRepeat([0,0,0,0,0,0, 0, 0, 0,50],4))
deck3Pen <- Flatten(ShuffleRepeat([0,0,0,0,0,1, 2, 2, 2, 3],4))
deck4Pen <- Flatten(ShuffleRepeat([0,0,0,0,0,0, 0, 0, 0,10],4))

deck1Rew <- 4
deck2Rew <- 4
deck3Rew <- 2
deck4Rew <- 2
```

```

#####
## Begin presenting and collecting data
#####

if(gINputDevice == "keyboard")
{
method <- " Choose a door by pressing the C V B or N key. Press any key to begin."
}elseif(gINputDevice == "mouse")
{
method <- " Escolha uma porta clicando com o mouse. Clique com o mouse para iniciar o teste."
} else {
method <- ""
}

## Give instructions; wait for them to hit a key,
SetText(gInstructions,"Você iniciará um jogo no qual você ajudará um burro a pegar maçãs. Atrás de cada porta estão algumas maçãs que o
burro receberá, mas cada porta também custará ao burro algumas maçãs."+method)

Show(gInstructions)
Draw()

WaitForContinue()
Hide(gInstructions)
Draw()

#####
#####
##
## Set up the basic graphical environment (decks, score, etc).
##

##Turn the header on during the task.
if(gInputDevice == "keyboard")
{
gHeader.text <- "Select door by pressing corresponding key (cvbn)"
}else{
gHeader.text <- "Selecione a porta clicando com o mouse."
}
Show(gHeader)

gTotal <- 0
earnings <- MakeEarningsGraph(gTotal)
gTotalLabel <- MakeLabel("Total: "+gTotal + " maçãs.",gHeaderFont)
AddObject(gTotalLabel,gWin)
Move(gTotalLabel,gVideoWidth/2,gVideoHeight/2+100)
Draw()

#####
#####
## Start the experimental trials.
##
deck <- 0
trialnum <- 1
while(trialnum <= 150)
{
time0 <- GetTime()
if(gInputDevice=="keyboard")
{
resp <- WaitForListKeyPress(["c","v","b","n"])
}else{
resp <- WaitForClickOnTarget(gDoors,["c","v","b","n"])
}

time1 <- GetTime()
rt <- time1 - time0

if(resp == "c")
{
reward <- deck1rew
penalty <- First(deck1pen)
net <- reward - penalty
}
}

```

```

        door <- 1
        ##Update the remainder of the deck.
        deck1pen <- Rotate(deck1pen,1)

    }elseif(resp == "v"){

        reward <- deck2rew
        penalty <- First(deck2pen)
        net <- reward - penalty
        door <- 2
            ##Update the remainder of the deck.
            deck2pen <- Rotate(deck2pen,1)

    }elseif(resp == "b")
    {

        reward <- deck3rew
        penalty <- First(deck3pen)
        net <- reward - penalty
        door <- 3
            ##Update the remainder of the deck.
            deck3pen <- Rotate(deck3pen,1)

    }else{

        reward <- deck4rew
        penalty <- First(deck4pen)
        net <- reward - penalty
        door <- 4

            ##Update the remainder of the deck.
            deck4pen <- Rotate(deck4pen,1)
    }

    Hide(Nth(gDoors,door))
    ShowPayoff(reward,penalty,Nth(gDoors,door))
    Draw()
    Wait(250)
    ##Figure out the +/- sign

    gTotal <- gTotal + net
    gTotalLabel.text <- "Total: "+gTotal+" maçãs."

    RemoveComplex(earnings)
        earnings <- MakeEarningsGraph(gTotal)

    ## Print out data to the file
    FilePrint_(gFileOut, gSubNum + "," + trialnum + "," + resp + "," + reward + "," + penalty)
    FilePrint(gFileOut, "," + net + "," + gTotal + "," + time1 + "," + rt )

    Wait(400)
    if(gInputdevice=="keyboard")
    {
        WaitForAllKeysUp()
    }
    Show(Nth(gDoors,door))
    HidePayoff(reward,penalty)
    Draw()

    trialnum <- trialnum + 1
}

Hide(gHeader)
Hide(gTotalLabel)
RemoveComplex(earnings)

FileClose(gFileOut)

##Now, show debriefing info.
feedback <- "Obrigada por participar do experimento.=D"

```

```
feedback <- feedback + CR(3) + " Por favor, avise ao experimentador que você terminou. Clique com mouse para terminar."
```

```
SetText(gInstructions, feedback)
Show(gInstructions)
Print(feedback)
  Draw()
  WaitForContinue()
ShowCursor(1)
}
```

```
## This is a standard initializer function that sets up typical
## objects used in experiments.
```

```
define Initialize()
{
```

```
  # gVideoWidth <- 640
  # gVideoHeight <- 480
```

```
  ##Initialize Window, etc.
  gWin <- MakeWindow("grey")
```

```
  ##Get subject code if we need to:
```

```
  if(gSubNum+""=="0")
  {
    gSubNum <- GetSubNum(gWin)
  }
```

```
  ##Initialize Font and colors
```

```
  bg <- MakeColor("grey")
  fg <- MakeColor("black")
  bg2 <- MakeColor("white")
```

```
  gColGreen <- MakeColor("darkgreen")
  gColRed  <- MakeColor("red")
```

```
  gStimFont    <- MakeFont(gPEBLBaseFontMono, 0, 16,fg,bg,1)
  gGraphFont   <- gStimFont
  gCorrFont    <- MakeFont(gPEBLBaseFontMono, 0, 24,gColGreen, bg,1)
  gIncorrFont  <- MakeFont(gPEBLBaseFontMono, 0, 24,gColRed,  bg,1)
```

```
  gInstructionsFont <- MakeFont(gPEBLBaseFont,0,33, fg,bg2,1)
  gHeaderFont      <- MakeFont(gPEBLBaseFont, 0,33,fg,bg,1)
```

```
  ## Make and hide a header label
  gHeader <- MakeLabel("",gHeaderFont)
  AddObject(gHeader,gWin)
  Move(gHeader, gVideoWidth/2, 35)
  Hide(gHeader)
```

```
  ##Load the donkey
```

```
  gDonkey <- MakeImage("png/donkey.png")
  AddObject(gDonkey,gWin)
  Move(gDonkey, gVideoWidth/2-250,gVideoHeight/2 + 200)
```

```
  gDoorways <- MakeImage("png/doorways.png")
  AddObject(gDoorways,gWin)
  Move(gDoorways,gVideoWidth/2,gVideoHeight/2-80)
```

```
  gDoors <- [ MakeImage("png/door.png"),
              MakeImage("png/door.png"),
              MakeImage("png/door.png"),
              MakeImage("png/door.png")]
```

```
  x <- gVideoWidth/2-160*1.5
  loop(i,gDoors)
  {
    AddObject(i,gWin)
    Move(i,x,gVideoHeight/2-80)
```

```

x <- x + 160
}

#Make a 'pool' of apples
gP2 <- MakeImage("png/appleg+2.png")
gP4 <- MakeImage("png/appleg+4.png")
gM1 <- MakeImage("png/apple-1.png")
gM2 <- MakeImage("png/apple-2.png")
gM3 <- MakeImage("png/apple-3.png")
gM8 <- MakeImage("png/apple-8.png")
gM10 <- MakeImage("png/apple-10.png")
gM12 <- MakeImage("png/apple-12.png")
gM50 <- MakeImage("png/apple-50.png")

gPlus <- [gP2,gP4]
gMinus <- [gM1, gM2, gM3, gM8, gM10,gM12,gM50]

loop(i,Merge(gMinus,gPlus))
{
  AddObject(i,gWin)
  Hide(i)
  MoveLower(i,100,140)
}

## Make and place the instruction box, then hide it
gInstructions <- MakeTextBox("", gInstructionsFont, gVideoWidth-100,gVideoHeight-100)
AddObject(gInstructions, gWin)
Move(gInstructions, 50,50)
Hide(gInstructions)

}

## This will display the apples in the correct door,
## depending on the plus/minus payoff.
define ShowPayoff(plus, minus, door)
{

  ## These numbers just automatically compute the best
  ## positions to display the feedback at
  x <- door.x-door.width/2+5
  y <- door.y + door.height/2-25

  if(plus != 0)
  {
    plusImage <- Nth(gPlus,ToInteger(plus/2))
    MoveLower(plusImage,x,y)
    Show(plusImage)
  }

  if(minus != 0)
  {
    orderMinus <- [1,2,3,8,10,12,50]
    loop(i, Transpose([orderMinus,gMinus]))
    {

      if(minus == First(i))
      {
        minusImage <- Nth(i,2)
        break
      }
    }

    MoveLower(minusImage,x+40,y)
    Show(minusImage)
  }

}

define HidePayoff(plus,minus)
{

  if(plus != 0)
  {

```

```

plusImage <- Nth(gPlus,ToInteger(plus/2))
Hide(plusImage)
}

if(minus != 0)
{
orderMinus <- [1,2,3,8,10,12,50]
loop(i, Transpose([orderMinus,gMinus]))
{
if(minus == First(i))
{
minusImage <- Nth(i,2)
break
}
}
Hide(minusImage)
}
}

define MoveLower(object, x,y)
{
s <- GetSize(object)
Move(object, x+Nth(s,2)/2, (y-Nth(s,1)/2))
}

define MakeEarningsGraph(earnings)
{
##Sets up some basic parameters for plotting the graph at the
##bottom of the screen
scale <- .8
height <- 20
width <- earnings / scale
ycenter <- gVideoHeight/2 + 220
xcenter <- gVideoWidth/2 + 30

#Determine which color to make it.
if(earnings < 0)
{
col <- gColRed
} else {
col <- gColGreen
}

##Define the locations of the labels
yvals <- [-150,-100,-50,0,50,100,150]

##Create the bargraph and the outline of the graph region.

x <- Rectangle(xcenter+width/2, ycenter, width, height, col, 1)
outline <- Rectangle(xcenter,ycenter-5, 450,50,col, 0)
AddObject(x,gWin)
AddObject(outline,gWin)

##Create all the labels, based on yvals. Add each label to the
##appropriate location, and combine them all into a single
##composite to make removal easier.
##These labels are all re-created each trial--they could be
##static, which would be a little more efficient, but it is really
##fast as is.
composite <-[outline,x]
loop(i,yvals)
{
yy <- MakeLabel(i+"" , ggraphFont)
AddObject(yy,gWin)
Move(yy, xcenter+(i/scale), ycenter-20)
Show(yy)
composite <- Append(composite,yy)
}

return composite
}

define RemoveComplex(x)

```

```
{
  loop(i,x)
  {
    RemoveObject(i,gWin)
  }
}

define WaitForContinue()
{
  if(gInputDevice=="keyboard")
  {
    WaitForAnyKeyPress()
  } else {
    WaitForDownClick()
  }
}
```

## ANEXO E- Programação Teste de Dígitos

### VERSÃO DIRETA:

```
#####
## Digit Span, Version 0.2
## For use with PEBL 0.06 or later
## http://pebl.sf.net
## Part of The PEBL Psychological Testing Battery
## 2006-02 Released into Public Domain
## by Shane T. Mueller, Ph.D. (smueller at obereed dot net)
##
##
## This conducts a digit span task with visual presentation
## and typed responses. Can be manipulated to start at a low
## length and increase until two erroneous trials in a row or 10;
## To start at length ten and decrease until two correct trials; or
## to use a staircase for X trials.
##
## This saves two files, a basic data file in regular format and one
## line per trial; and a second log file which lists stimuli, responses
## response times, etc.
##
## Version 0.2 enables both forward and backward digit span;
## saves as .csv. With encouragement and help from Reid Olsen.
#####

define Start(IPar)
{
  gLanguage <- "PT"
  ## results are stored in tmpresults.dat

  ## Initialize global objects for use later. Defined at end of file
  Initialize()
  gSubNum <- GetSubNum(gWin)
  ##Get subject code if one has not been supplied
  if(gSubNum+"=="0")
  {
    gSubNum <- GetSubNum(gWin)
  }

  gDirection <- "forward" #change "forward" to "backward" to do
  #backward scoring

  ## currently, three devices are supported:
  ## keyboard, which uses the top numbers for data entry
  ## keypad, which uses the keypad numbers
  ## mouse, which creates a screen ala virtualkeyboard.
  gInputDevice <- "mouse"

  MakeDirectory("data") ##make the data directory if it doesn't exist.

  Print("Data are saved in [data/"+"dspan-"+gDirection+"-" + gSubNum + ".csv]")

  gFileOut <- FileOpenWrite("data/dspan-"+gDirection+"-" + gSubNum + ".csv")
  gReportFile <- FileOpenWrite("data/dspan-report"+gDirection+"-" + gSubNum + ".txt")
  gPooledFile <- FileOpenAppend("data/pooled-dspan-"+gDirection+"-" + gSubNum + ".csv")

  ShowCursor(1)
  ## This determines whether audio accompanies the visual
  ## presentation.
  gUseAudio <- 1

  gAllowBackspace <- 1 ##Allow users to backspace over responses?
  gAllowSkip <- 1 ##Allow users to skip over cells?

  ## Version can be hard-coded here, or entered with the -v command-line
  ## argument. 1 indicates traditional low-to-high; 2 indicates high-to-low;
```

```

## 3 indicates a staircase.
gVersion <- 1

## End Early determines whether the number of trials at each length
## should continue only until the criterion, or if all the
## lengths should be tested. I.e., if 2/3 correct are needed to
## pass up to the next level, we should always do 3. Furthermore,
## if endearly is 0, we will continue until we get 0/3 (or 3/3 if
## using gVersion <-2

endearly <- 0

upperbound <- 8      ## The longest list length
lowerbound <- 2      ## The shortest list length
perlength <- 2       ## The number of trials at each length
criterion <- 1       ## The number that must be correct to proceed
staircasestart <- 4   ## starting point for staircasestart.
staircasetrials <- 16 ## Number of staircase trials.
gISI <- 1000         ## Inter-stimulus interval in ms
iti <- 1500          ## inter-trial interval.

if(gVersion == 1){
  startLength <- 2
}elseif(gVersion == 2){
  startLength <- 8
}elseif(gVersion == 3){
  startLength <- 2
}

#####
## Now, generate stimuli/conditions
#####

## gStim <- FileReadList("DigitNames.txt")
gStim <- FileReadList("Digits.txt")

gSounds <- []
loop(i,gStim)
{
  x <- LoadSound("stim/"+i+".wav")
  gSounds <- Append(gSounds,x)
}

gReady <- LoadSound("stim/ready.wav")
gBeep <- LoadSound("stim/beep.wav")
gCorrect <- LoadSound("stim/correct.wav")
gIncorrect <- LoadSound("stim/incorrect.wav")

stim <- Transpose([gStim,gSounds])
#####
## Begin presenting and collecting data
#####

## Give instructions; wait for them to hit a key,
if(gDirection=="forward")
{
  inst <- "Você participará de um teste de memória. Serão faladas algumas listas de números, uma de cada vez. Cada número aparecerá
somentemente uma vez por lista. Você deverá clicar nos números que aparecem na tela. "
}
else{
  inst <- "Você participará de um teste de memória. Serão faladas algumas listas de números, uma de cada vez. Cada número aparecerá
somentemente uma vez por lista. Você deverá clicar nos números que aparecem na tela. "
}

if(gAllowSkip)
{

```

```

inst <- inst + " Mas ATENÇÃO: clique nos números EXATAMENTE NA ORDEM QUE VOCÊ ESCUTOU. "
}else{
  inst <- inst + " "
}

if(gAllowBackspace)
{
  inst <- inst + "Se cometer um erro, você pode clicar na seta para a esquerda <-- e corrigir. "
} else {
  inst <- inst + ""
}

lthInst <- inst + " Quando terminar de clicar nos números clique na seta para a direita --> para seguir para a próxima sequência.
Clique com o mouse para começar"
htlInst <- inst + " "

scsInst <- inst + " "

SetText(gInstructions,lthInst)
Show(gInstructions)
Draw()

instru01 <- LoadSound("instru01.wav")
PlayBackground(instru01)
Wait(38100)
Stop(instru01)
WaitForIt()

Hide(gInstructions)
Draw()

FilePrint(gFileOut, "subnum,version,direction,endeary,trialnum,length,try,input,response,correct,resptime")

#####
#####
## Start the experimental trials.
##

trialnum <- 1
continue <- 1
currcorrect <- 0
try <- 0
length <- startLength
lengths <- []
sumCorrect <- 0
while(continue)
{
  Wait(iti)
  FilePrint_(gFileOut, gSubNum + "," + gVersion + "," + gDirection+"," + endeary + "," + trialnum + "," + length + "," + try + ",")
  stimuli <- SampleN(stim,length)
  corr <- Trial(stimuli)

  sumCorrect <- sumcorrect + corr
  #####
  ##Standard Memory Span

if(gVersion == 1)
{
  try <- try + 1
  currcorrect <- currcorrect + corr

  if(endeary)
  {
    if(currcorrect >= criterion)
    {
      length <- length + 1

```

```

currcorrect <- 0
try <- 0
} elseif(try >= perlength) {

  ##We did not pass the threshold; we are done.
  continue <- 0
}

} else {
  if(try >= perlength)
  {
    if(currcorrect >= criterion)
    {
      length <- length + 1
      currcorrect <- 0
      try <- 0
    } else {
      continue <- 0
    }
  }
}

if(length > upperbound)
{
  continue <- 0
}

} elseif(gVersion == 2) {

  #####
  ## Version 2 uses from longest to short list lengths

  try <- try + 1
  currcorrect <- currcorrect + corr

  if(endearly) {
    ##In this situation, only do enough trials to pass criterion.
    if(currcorrect >= criterion)
    {
      continue <- 0

    } elseif(try >= perlength) {
      ##We have used all the tries, go one length lower.
      length <- length - 1
      currcorrect <- 0
      try <- 0
    } else {
      ##In this situation, do all the trials per length, until we get them all correct.
      if(try >= perlength) {
        if(currcorrect == perlength)
        {
          continue <- 0
        } else {
          length <- length -1
          currcorrect <- 0
          try <- 0
        }
      }
    }

    if(length < lowerbound)
    {
      continue <- 0
    }

  } else {
    #####
    ## Staircase
    if(corr)
    {

```

```

length <- Min([length + 1,upperbound])
} else {
length <- Max([lowerbound,length-1])
}
if(trialnum >= staircasetrials)
{
continue <- 0
}
}

if(trialnum > 4) {
lengths <- Append(lengths, length)
}

trialnum <- trialnum + 1

}
Hide(gHeader)
Draw()
FileClose(gFileOut)

##Now, show debriefing info.
feedback <- "Obrigada por sua participação. "

report <- "PEBL Digit Span Task" +CR(1)+
"Direction: " + gDirection + CR(1)+
Timestamp() + CR(1)+
"Memory Span: " + (length-1) + CR(1)+
"Total words correct: "+sumcorrect + CR(1)+
"Total time: " +Round(GetTime()/1000/60,2)+" min"

FilePrint(gReportFile,report)
FilePrint(gPooledFile, gSubNum+", "+TimeStam()+", "+ gDirection+", "+
(length-1)+", "+sumcorrect)

feedback <- feedback + CR(2) + "Clique com o mouse para terminar."

SetText(gInstructions, feedback)
Show(gInstructions)
Draw()

instru02 <- LoadSound("instru02.wav")
PlayBackground(instru02)
Wait(5200)
Stop(instru02)
WaitForIt()

WaitForIt()
ShowCursor(1)
}

define Trial(stimpairs)
{

stim <- First(Transpose(stimpairs))
length <- Length(stimpairs)
string <- ListToString(stimpairs)
dummy <- Repeat("_",length)
responselist <- []

gStimLabel.text <- "Prepare-se"
Show(gStimLabel)

Draw()
if(gUseAudio)
{
PlayForeground(gReady)
}
Wait(800)
Hide(gStimLabel)

```

```

Draw()
Show(gStimLabel)

Wait(1000)

loop(x,stimpairs)
{
  digit <- First(x)
  sound <-Nth(x,2)

  FilePrint_(gFileOut,digit)
  gStimLabel.text <- digit
  t1 <- GetTime()
  if(gUseAudio){PlayForeground(sound)}
  t2 <- GetTime()
  Wait(gISI - (t2-t1))
}
FilePrint_(gFileOut, ",")
##Play a response signal.

Hide(gStimLabel)
gStimLabel.text <- ListToString(dummy)
Draw()
if(gUseAudio){PlayForeground(gBeep)}

##Now, collect the responses.
gHeader.text <- "Clique nos números e quando terminar clique --> "

Show(gHeader)
Show(gStimLabel)
Draw()
corr <- 0

if(gInputDevice == "keyboard")
{
  responses <- ["1","2","3","4","5","6","7","8","9","0"]

}elseif(gInputDevice == "keypad")
{
  responses <- ["<kp_1>","<kp_2>","<kp_3>","<kp_4>","<kp_5>","<kp_6>","
    "<kp_7>","<kp_8>","<kp_9>","<kp_0>","<kp_enter>","<return>"]
} elseif (gInputDevice == "mouse"){

gStimLabel.y <- 120
##Make response screen here.

##basic display parameters
ybase <- 100
xbase <- 10
gap <- 70
keysize <- 90

keys <- [["1","2","3"],
  ["4","5","6"],
  ["7","8","9"],
  ["<--","0","<-->"]]

  ##we should automatically determine length/height of keys here.
  height <- 4
  width <- 3

font <- MakeFont(gPEBLBaseFont,2,25,MakeColor("white"),MakeColor("white"),0)
wordfont<-MakeFont(gPEBLBaseFont,2,15,MakeColor("lightgrey"),MakeColor("white"),0)

rownum <- 1
clicks <- []
  stuff <- []
  loop(row,keys)
  {
    colnum <- 1
    loop(col,row)

```

```

{
  back1 <- Square(xbase+colnum*gap,ybase+rownum*gap,keysiz,MakeColor("grey"),1)
  back2 <- Square(xbase+colnum*gap,ybase+rownum*gap,keysiz-4,MakeColor("grey20"),1)
  if(StringLength(col)>1)
  {
    lab <- MakeLabel(col,wordfont)
  } else {
    lab <- MakeLabel(col,font)
  }
  AddObject(back1,gwin)
  AddObject(back2,gwin)
  AddObject(lab,gwin)
  Move(lab,xbase+colnum*gap,ybase+rownum*gap)

  colnum <- colnum + 1
  clicks <- Append(clicks,back1)
  stuff <- Append(stuff,[lab,back1,back2])
}
rownum <- rownum + 1

}

##End mouse keyboard setup.

responses <- clicks
}

if(gInputDevice=="keyboard")
{
  if(gAllowBackspace)
  {
    responses <- Append(responses,"<backspace>")
  }

  if(gAllowSkip)
  {
    responses <- Append(responses,"-")
  }
}
Draw()

position <- 1
resp <- ""

time1 <- GetTime()
while(not (resp == "<return>" or resp == "<enter>" or resp == "<kp_enter>"))
{
  if(gInputDevice=="keyboard" or gInputDevice == "keypad")
  {
    resp <- WaitForListKeyPress(responses)

  }elseif(gInputDevice == "mouse")
  {
    resp <- WaitForClickOnTarget(responses, ["1","2","3","4","5","6","7","8","9","<backspace>","0","<return>"])
  }
  if(resp == "<backspace>")
  {
    if(Length(responseList) == 1)
    {
      responseList <- []
      position <- 1
    } else {
      if(gDirection=="forward")
      {
        responseList <- SubList(responseList, 1, Length(responseList)-1)
        position <- position - 1
      }else{
        responseList <- SubList(responseList, 2, Length(responseList))
        position <- position - 1
      }
    }
  }
}

```

```

} elseif(resp == "<return>" or resp == "<kp_enter>") {
  ##
  33
} else {

  if(position <= length)
  {
    position <- position + 1
    if(gDirection == "forward")
    {
      responseList <- Append(responseList, resp)
    } else {
      responseList <- Merge([resp],responseList)
    }
  }
}

  if(gDirection == "forward")
  {
    respText <- ListToString(Merge(responseList,Repeat("_",(length - position + 1))))
  }else{
    respText <- ListToString(Merge(Repeat("_",(length - position + 1)),responseList))
  }

gStimLabel.text <- respText
Draw()

##See if things are correct.
if( ListToString(stim) == respText)
{
  corr <- 1
} else{
  corr <- 0
}
}
time2 <- GetTime()

##finish up the data output.
FilePrint(gFileOut,respText+","corr+", "+(time2-time1))

Wait(200)

##Give Feedback Here
gFeedback.text <- ListToString(stim)
Show(gFeedback)
Draw()
Wait(1500)
if(corr)
{
  gFeedback.text <- " Correto!=D"
  feedbacksound <- gCorrect
} else {
  gFeedback.text <- " Errado =(("
  feedbacksound <- gIncorrect
}
Draw()
if(gUseAudio){PlayForeground(feedbacksound)}
Wait(1500)

Hide(gFeedback)
Hide(gStimLabel)
Draw()
return corr
}

## This is a standard initializer function that sets up typical
## objects used in experiments.
define Initialize()
{
  ##Initialize Window, etc.
  gWin <- MakeWindow("grey")

```

```

##Get subject code if we need to:
if(gSubNum==0)
{
    gSubNum <- GetSubNum(gWin)
}

##Initialize Font and colors

bg <- MakeColor("grey")
bg1 <- MakeColor("lightgrey")
fg <- MakeColor("black")
bg2 <- MakeColor("white")

colGreen <- MakeColor("darkgreen")
colRed <- MakeColor("red")

gStimFont <- MakeFont(gPEBLBaseFontMono, 0, 56, fg,bg,1)

gInstructionsFont <- MakeFont(gPEBLBaseFont,0,18, fg,bg2,1)
gHeaderFont <- MakeFont(gPEBLBaseFont, 0,24,fg,bg,1)

## Make and place the instruction box, then hide it
gInstructions <- MakeTextBox("", gInstructionsFont, 600,300)
AddObject(gInstructions, gWin)
Move(gInstructions, gVideoWidth/2-300, gVideoHeight/2-150)
Hide(gInstructions)

## Make and hide a header label
gHeader <- MakeTextBox("",gHeaderFont, 600,200)
AddObject(gHeader,gWin)
Move(gHeader, gVideoWidth/2-250, 50)
Hide(gHeader)

gStimLabel <- MakeLabel("",gStimFont)
AddObject(gStimLabel,gWin)
Move(gStimLabel, gVideoWidth/2, gVideoHeight/2)
Hide(gStimLabel)

gFeedback <- MakeLabel("",gStimFont)
AddObject(gFeedback,gWin)
Move(gFeedback, gVideoWidth/2, gVideoHeight/2+70)
Hide(gFeedback)

}

define WaitForIt()
{
    if(gInputDevice == "keyboard" or gInputDevice == "keypad")
    {
        WaitForAnyKeyPress()
    } elseif(gInputDevice=="mouse")
    {
        WaitForDownClick()
    }
}

```

## VERSÃO INVERSA:

```

#####
## Digit Span, Version 0.2
## For use with PEBL 0.06 or later
## http://pebl.sf.net
## Part of The PEBL Psychological Testing Battery
## 2006-02 Released into Public Domain
## by Shane T. Mueller, Ph.D. (smueller at obereed dot net)
##
##
## This conducts a digit span task with visual presentation
## and typed responses. Can be manipulated to start at a low
## length and increase until two erroneous trials in a row or 10;
## To start at length ten and decrease until two correct trials; or
## to use a staircase for X trials.
##

```

```

## This saves two files, a basic data file in regular format and one
## line per trial; and a second log file which lists stimuli, responses
## response times, etc.
##
## Version 0.2 enables both forward and backward digit span;
## saves as .csv. With encouragement and help from Reid Olsen.
#####

define Start(IPar)
{
gLanguage <- "PT"
  ## results are stored in tmpresults.dat

  ## Initialize global objects for use later. Defined at end of file
  Initialize()
gSubNum <- GetSubNum(gWin)
  ##Get subject code if one has not been supplied
  if(gSubNum+""=="0")
  {
    gSubNum <- GetSubNum(gWin)
  }

  gDirection <- "backward" #change "forward" to "backward" to do
    ##backward scoring

  ## currently, three devices are supported:
  ## keyboard, which uses the top numbers for data entry
  ## keypad, which uses the keypad numbers
  ## mouse, which creates a screen ala virtualkeyboard.
  gInputDevice <- "mouse"

  MakeDirectory("data") ##make the data directory if it doesn't exist.

  Print("Data are saved in [data/"+dspan-"+gDirection+"-" + gSubNum + ".csv]")

  gFileOut <-FileOpenWrite("data/dspan-"+gDirection+"-" + gSubNum + ".csv")
  gReportFile <- FileOpenWrite("data/dspan-report"+gDirection+"-" + gSubNum + ".txt")
  gPooledFile <- FileOpenAppend("data/pooled-dspan-"+gDirection+"-" + gSubNum + ".csv")

  ShowCursor(1)
  ## This determines whether audio accompanies the visual
  ## presentation.
  gUseAudio <- 1

  gAllowBackspace <- 1 ##Allow users to backspace over responses?
  gAllowSkip <- 1 ##Allow users to skip over cells?

  ## Version can be hard-coded here, or entered with the -v command-line
  ## argument. 1 indicates traditional low-to-high; 2 indicates high-to-low;
  ## 3 indicates a staircase.
  gVersion <- 1

  ## End Early determines whether the number of trials at each length
  ## should continue only until the criterion, or if all the
  ## lengths should be tested. I.e., if 2/3 correct are needed to
  ## pass up to the next level, we should always do 3. Furthermore,
  ## if endearly is 0, we will continue until we get 0/3 (or 3/3 if
  ## using gVersion <-2

  endearly <- 0

  upperbound <- 8 ## The longest list length
  lowerbound <- 2 ## The shortest list length
  perlength <- 2 ## The number of trials at each length
  criterion <- 1 ## The number that must be correct to proceed
  staircasestart <- 4 ## starting point for staircasestart.
  staircasetrials <- 16 ## Number of staircase trials.
  gISI <- 1000 ## Inter-stimulus interval in ms
  iti <- 1500 ## inter-trial interval.

```

```

if(gVersion == 1){
  startLength <- 2
}elseif(gVersion == 2){
  startLength <- 8
}elseif(gVersion == 3){
  startLength <- 2
}

#####
## Now, generate stimuli/conditions
#####

## gStim <- FileReadList("DigitNames.txt")
gStim <- FileReadList("Digits.txt")

gSounds <- []
loop(i,gStim)
{
  x <- LoadSound("stim/"+i+".wav")
  gSounds <- Append(gSounds,x)
}

gReady <- LoadSound("stim/ready.wav")
gBeep <- LoadSound("stim/beep.wav")
gCorrect <- LoadSound("stim/correct.wav")
gIncorrect <- LoadSound("stim/incorrect.wav")

stim <- Transpose([gStim,gSounds])
#####
## Begin presenting and collecting data
#####

## Give instructions; wait for them to hit a key,
if(gDirection=="forward")
{
  inst <- "Você participará de um teste de memória. Serão faladas algumas listas de números, uma de cada vez. Cada número aparecerá somente uma vez por lista. Você deverá clicar nos números que aparecem na tela. "
}
else{
  inst <- "Você participará de um teste de memória. Serão faladas algumas listas de números, uma de cada vez. Cada número aparecerá somente uma vez por lista. Você deverá clicar nos números que aparecem na tela. "
}

if(gAllowSkip)
{
  inst <- inst + " Mas ATENÇÃO: clique nos números na ordem INVERSA da que você escutou. Se a sequência falada foi 1 2, você deverá clicar 2 1. "
}
else{
  inst <- inst + " "
}

if(gAllowBackspace)
{
  inst <- inst + "Se cometer um erro, você pode clicar na seta para a esquerda <-- e corrigir. "
} else {
  inst <- inst + ""
}

lthInst <- inst + " Quando terminar de clicar nos números clique na seta para a direita --> para seguir para a próxima sequência. Clique com o mouse para começar"
htlInst <- inst + " "

```

```

scsInst <- inst + " "

SetText(gInstructions,lthInst)
Show(gInstructions)
Draw()

instru03 <- LoadSound("instru03.wav")
PlayBackground(instru03)
Wait(47500)
Stop(instru03)
WaitForIt()

Hide(gInstructions)
Draw()

FilePrint(gFileOut, "subnum,version,direction,endeary,tialnum,length,try,input,response,correct,respime")

#####
#####
## Start the experimental trials.
##

trialnum <- 1
continue <- 1
currcorrect <- 0
try <- 0
length <- startLength
lengths <- []
sumCorrect <- 0
while(continue)
{

  Wait(iti)
  FilePrint_(gFileOut, gSubNum + "," + gVersion + "," + gDirection + "," + endeary + "," + trialnum + "," + length + "," + try + ",")
  stimuli <- SampleN(stim,length)
  corr <- Trial(stimuli)

  sumCorrect <- sumcorrect + corr
  #####
  ##Standard Memory Span

  if(gVersion == 1)
  {
    try <- try + 1
    currcorrect <- currcorrect + corr

    if(endeary)
    {
      if(currcorrect >= criterion)
      {

        length <- length + 1
        currcorrect <- 0
        try <- 0
      } elseif(try >= perlenght ) {

        ##We did not pass the threshold; we are done.
        continue <- 0
      }

    } else {
      if(try >= perlenght)
      {
        if(currcorrect >= criterion)
        {
          length <- length + 1
          currcorrect <- 0
          try <- 0
        } else {
          continue <- 0
        }
      }
    }
  }
}

```

```

    }
}
if(length > upperbound)
{
  continue <- 0
}
} elseif(gVersion == 2) {

#####
## Version 2 uses from longest to short list lengths

try <- try + 1
currcorrect <- currcorrect + corr

if(endeary) {
  ##In this situation, only do enough trials to pass criterion.
  if(currcorrect >= criterion)
  {
    continue <- 0

  } elseif(try >= perlength ) {
    ##We we have used all the tries, go one length lower.
    length <- length - 1
    currcorrect <- 0
    try <- 0
  }
} else {
  ##In this situation, do all the trials per length, until we get them all correct.
  if(try >= perlength) {
    if(currcorrect == perlength)
    {
      continue <- 0
    } else {
      length <- length -1
      currcorrect <- 0
      try <- 0
    }
  }
}

  if(length < lowerbound)
  {
    continue <- 0
  }

} else {
#####
## Staircase
if(corr)
{
  length <- Min([length + 1,upperbound])
} else {
  length <- Max([lowerbound,length-1])
}
if(trialnum >= staircasetrials)
{
  continue <- 0
}
}

if(trialnum > 4) {
  lengths <- Append(lengths, length)
}

trialnum <- trialnum + 1

}
Hide(gHeader)

```

```

Draw()
FileClose(gFileOut)

##Now, show debriefing info.
feedback <- "Obrigada por participar deste experimento. "

report <- "PEBL Digit Span Task" +CR(1)+
"Direction: " + gDirection + CR(1)+
Timestamp() + CR(1)+
"Memory Span: " + (length-1) + CR(1)+
"Total words correct: "+sumcorrect + CR(1)+
"Total time: " +Round(GetTime()/1000/60,2)+" min"

FilePrint(gReportFile,report)
FilePrint(gPooledFile, gSubNum+", "+TimeStamp()+", "+ gDirection+", "+
(length-1)+", "+sumcorrect)

feedback <- feedback + CR(2) + "Pressione qualquer tecla para sair."

SetText(gInstructions, feedback)
Show(gInstructions)
Draw()

instru02 <- LoadSound("instru02.wav")
PlayBackground(instru02)
Wait(5200)
Stop(instru02)
WaitForIt()

WaitForIt()
ShowCursor(1)
}

define Trial(stimpairs)
{

stim <- First(Transpose(stimpairs))
length <- Length(stimpairs)
string <- ListToString(stimpairs)
dummy <- Repeat("_",length)
responselist <- []

gStimLabel.text <- "Prepare-se"
Show(gStimLabel)

Draw()
if(gUseAudio)
{
PlayForeground(gReady)
}
Wait(800)
Hide(gStimLabel)
Draw()
Show(gStimLabel)

Wait(1000)

loop(x,stimpairs)
{
digit <- First(x)
sound <-Nth(x,2)

FilePrint_(gFileOut,digit)
gStimLabel.text <- digit
t1 <- GetTime()
if(gUseAudio){PlayForeground(sound)}
t2 <- GetTime()
Wait(gISI - (t2-t1))
}
FilePrint_(gFileOut, ", ")
##Play a response signal.

```

```

Hide(gStimLabel)
gStimLabel.text <- ListToString(dummy)
Draw()
if(gUseAudio){PlayForeground(gBeep)}

##Now, collect the responses.
gHeader.text <- "Clique nos números e quando terminar clique -->"

Show(gHeader)
Show(gStimLabel)
Draw()
corr <- 0

if(gInputDevice == "keyboard")
{
  responses <- ["1","2","3","4","5","6","7","8","9","0"]

}

}elseif(gInputDevice == "keypad")
{
  responses <- ["<kp_1>","<kp_2>","<kp_3>","<kp_4>","<kp_5>","<kp_6>","
    "<kp_7>","<kp_8>","<kp_9>","<kp_0>","<kp_enter>","<return>"]
} elseif (gInputDevice == "mouse"){

gStimLabel.y <- 250
##Make response screen here.

##basic display parameters
ybase <- 100
xbase <- 10
gap <- 70
keysize <- 90

keys <- [[["1","2","3"],
  ["4","5","6"],
  ["7","8","9"],
  ["<-->","0","<-->"]]]

  ##we should automatically determine length/height of keys here.
  height <- 4
  width <- 3

font <- MakeFont(gPEBLBaseFont,2,25,MakeColor("white"),MakeColor("white"),0)
wordfont<-MakeFont(gPEBLBaseFont,2,15,MakeColor("lightgrey"),MakeColor("white"),0)

rownum <- 1
clicks <- []
stuff <- []
loop(row,keys)
{
  colnum <- 1
  loop(col,row)
  {
    back1 <- Square(xbase+colnum*gap,ybase+rownum*gap,keysize,MakeColor("grey"),1)
    back2 <- Square(xbase+colnum*gap,ybase+rownum*gap,keysize-4,MakeColor("grey20"),1)
      if(StringLength(col)>1)
      {
        lab <- MakeLabel(col,wordfont)
      } else {
        lab <- MakeLabel(col,font)
      }
      AddObject(back1,gwin)
      AddObject(back2,gwin)
      AddObject(lab,gwin)
      Move(lab,xbase+colnum*gap,ybase+rownum*gap)

    colnum <- colnum + 1
    clicks <- Append(clicks,back1)
    stuff <- Append(stuff,[lab,back1,back2])
  }
  rownum <- rownum + 1

```

```

}
##End mouse keyboard setup.

responses <- clicks
}

if(gInputDevice=="keyboard")
{
if(gAllowBackspace)
{
responses <- Append(responses,"<backspace>")
}

if(gAllowSkip)
{
responses <- Append(responses,"-")
}
}
Draw()

position <- 1
resp <- ""

time1 <- GetTime()
while(not (resp == "<return>" or resp== "<enter>" or resp== "<kp_enter>"))
{
if(gInputDevice=="keyboard" or gInputDevice == "keypad")
{
resp <- WaitForListKeyPress(responses)

}elseif(gInputDevice == "mouse")
{

resp <- WaitForClickOnTarget(responses, ["1","2","3","4","5","6","7","8","9","<backspace>","0","<return>"])
}
if(resp == "<backspace>")
{
if(Length(responseList) == 1)
{
responseList <- []
position <- 1
} else {
if(gDirection=="forward")
{
responseList <- SubList(responseList, 1, Length(responseList)-1)
position <- position - 1
} else {

responseList <- SubList(responseList, 2, Length(responseList))
position <- position - 1
}
}
}

} elseif(resp == "<return>" or resp == "<kp_enter>") {
##
33
} else {

if(position <= length)
{
position <- position + 1
if(gDirection == "forward")
{
responseList <- Append(responseList, resp)
} else {
responseList <- Merge([resp],responseList)
}
}
}

if(gDirection == "forward")
{
resptext <- ListToString(Merge(responseList,Repeat("_",(length - position + 1))))
} else {
resptext <- ListToString(Merge(Repeat("_",(length - position + 1)),responseList))
}
}

```

```

    }

    gStimLabel.text <- respstext

    Draw()

##See if things are correct.
    if( ListToString(stim) == respstext)
    {
        corr <- 1
    }else{
        corr <- 0
    }
}
time2 <- GetTime()

##finish up the data output.
FilePrint(gFileOut,respstext+", "+corr+", "+(time2-time1))

Wait(200)

##Give Feedback Here
gFeedback.text <- ListToString(stim)
Show(gFeedback)
Draw()
Wait(1500)
if(corr)
{
    gFeedback.text <- " Correto!=D"
    feedbacksound <- gCorrect
} else {
    gFeedback.text <- " Errado =(("
    feedbacksound <- gIncorrect
}
Draw()
if(gUseAudio){PlayForeground(feedbacksound)}
Wait(1500)

Hide(gFeedback)
Hide(gStimLabel)
Draw()
return corr
}

## This is a standard initializer function that sets up typical
## objects used in experiments.
define Initialize()
{
    ##Initialize Window, etc.
    gWin <- MakeWindow("grey")

    ##Get subject code if we need to:
    if(gSubNum==0)
    {
        gSubNum <- GetSubNum(gWin)
    }

##Initialize Font and colors

    bg <- MakeColor("grey")
    bg1 <- MakeColor("lightgrey")
    fg <- MakeColor("black")
    bg2 <- MakeColor("white")

    colGreen <- MakeColor("darkgreen")
    colRed <- MakeColor("red")

    gStimFont <- MakeFont(gPEBLBaseFontMono, 0, 56, fg,bg,1)

    gInstructionsFont <- MakeFont(gPEBLBaseFont,0,18, fg,bg2,1)
    gHeaderFont <- MakeFont(gPEBLBaseFont, 0,24,fg,bg,1)

```

```
## Make and place the instruction box, then hide it
gInstructions <- MakeTextBox("", gInstructionsFont, 600,300)
AddObject(gInstructions, gWin)
Move(gInstructions, gVideoWidth/2-300, gVideoHeight/2-150)
Hide(gInstructions)

## Make and hide a header label
gHeader <- MakeTextBox("",gHeaderFont, 600,200)
AddObject(gHeader,gWin)
Move(gHeader, gVideoWidth/2-250, 50)
Hide(gHeader)

gStimLabel <- MakeLabel("",gStimFont)
AddObject(gStimLabel,gWin)
Move(gStimLabel, gVideoWidth/2, gVideoHeight/2)
Hide(gStimLabel)

gFeedback <- MakeLabel("",gStimFont)
AddObject(gFeedback,gWin)
Move(gFeedback, gVideoWidth/2, gVideoHeight/2+70)
Hide(gFeedback)

}

define WaitForIt()
{
  if(gInputDevice == "keyboard" or gInputDevice == "keypad")
  {
    WaitForAnyKeyPress()
  } elseif(gInputDevice=="mouse")
  {
    WaitForDownClick()
  }
}
```

## ANEXO F- Protocolo

### Protocolo de Aplicação

Questionário elaborado com base no "Questionário TIC Domicílios"

Código criança: \_\_\_\_\_ ( ) AMBDA ( ) ESCOLA

Data teste 1: \_\_\_\_/\_\_\_\_/\_\_\_\_ Hora início: \_\_ \_\_: \_\_ \_\_ Hora de término: \_\_ \_\_: \_\_ \_\_

Aplicador: \_\_\_\_\_

Data teste 2: \_\_\_\_/\_\_\_\_/\_\_\_\_ Hora início: \_\_ \_\_: \_\_ \_\_ Hora de término: \_\_ \_\_: \_\_ \_\_

Aplicador: \_\_\_\_\_

Total horas 1 e 2: \_\_\_\_\_

Tempo: Escalas: 1 \_\_\_\_\_ 2 \_\_\_\_\_ 3 \_\_\_\_\_ Bcst: \_\_\_\_\_

Donkey: \_\_\_\_\_

Dspan: \_\_\_\_\_ Dspaninverso: \_\_\_\_\_ ToL: \_\_\_\_\_ Go/nogo: \_\_\_\_\_

Stroop: \_\_\_\_\_

#### Dados pessoais: (coleta com responsável)

Entrevistado: \_\_\_\_\_

Data de nascimento: \_\_\_\_/\_\_\_\_/\_\_\_\_ Idade: \_\_\_\_\_ Sexo: \_\_\_\_\_ ( ) Destro ( )

Canhoto

Escolaridade (\*anos completos de estudo): \_\_\_\_\_ Escola: ( ) pública ( ) privada

Série atual: \_\_\_\_\_ ( ) manhã ( ) tarde ( ) noite

#### Conhecimento de informática:

1. Você já usou um computador? ( ) Sim ( ) Não

2. Você gosta de computador? [0- não gosta, 10- gosta muito]

01	02	03	04	05	06	07	08	09	10
Não gosta				Normal					Gosta muito

3. Com que idade utilizou computador pela primeira vez? \_\_\_\_\_

4. Tem acesso a um computador em casa? ( ) Sim ( ) Não

5. Quando você utilizou um computador pela última vez?

( ) Há menos de 3 meses ( ) Entre 3 meses e 12 meses atrás ( ) Há mais de 12 meses

6. Quando você costuma usar o computador na maioria das vezes?

( ) durante a semana ( ) fins de semana ( ) manhã ( ) tarde ( ) noite

7. Nos últimos 3 meses, quantas horas aproximadamente você gastou no Computador/semana?

| \_\_\_\_\_ | hora(s) por semana

8. Quais das seguintes atividades ligadas ao computador você já executou?

Habilidades	Pontuação*
Usar um <i>mouse</i>	(0) (1) (2)
Copiar ou mover um arquivo ou uma pasta	(0) (1) (2)
Usar o paint (programa para colorir)	(0) (1) (2)
Usar um editor de texto (Word)	(0) (1) (2)
Usar uma planilha de cálculo ( Excel)	(0) (1) (2)
Comprimir arquivos no computador (WinZip)	(0) (1) (2)
Conectar ou Instalar periféricos (impressora, câmera, microfone)	(0) (1) (2)
Usar programas de som e imagem/multimídia	(0) (1) (2)
Executar um jogo no computador (sem internet)	(0) (1) (2)
Escrever um programa de computador usando alguma linguagem de programação	(0) (1) (2)

\*Legendas: 0-nunca, 1- executou com ajuda, 2- executou sozinho

9. Você já utilizou a Internet? ( ) Sim ( ) Não (*pule para a questão 14*)

10. Em média, com que frequência você usou a Internet nos últimos 3 meses?

( ) Todos os dias ou quase todos os dias ( ) Pelo menos uma vez por semana

( ) Pelo menos uma vez por mês ( ) Menos de uma vez por mês

11. Quando você costuma utilizar a internet na maioria das vezes?  
 durante a semana  fins de semana  manhã  tarde  noite

12. Nos últimos 3 meses, quantas horas aproximadamente você gastou na Internet/semana?

| \_\_\_\_\_ | hora(s) por semana

13. Quais das seguintes atividades ligadas à Internet você já executou?

Habilidades	Pontuação*
Abrir um programa para navegar na internet (Internet Explorer, Google Chrome, Mozilla, etc)	(0) (1) (2)
Usar um mecanismo de busca para achar informação (Google, Bing)	(0) (1) (2)
Enviar <i>e-mails</i> com arquivos anexados (documentos, fotos)	(0) (1) (2)
Enviar mensagens em salas de bate-papo e fóruns de discussão <i>online</i>	(0) (1) (2)
Participar de <i>sites</i> de relacionamento (Orkut, Facebook, Linked IN, Twitter)	(0) (1) (2)
Trocar mensagens de voz, usando programas - Skype/videoconferência	(0) (1) (2)
Jogar jogos <i>online</i> (conectados à Internet). Quais: _____	(0) (1) (2)
Assistir a filmes ou vídeos (como os do Youtube)	(0) (1) (2)
Baixar/ fazer <i>download</i> arquivos	(0) (1) (2)
Criar ou atualizar <i>blogs</i> e/ou páginas na Internet ( <i>websites</i> )	(0) (1) (2)
Outra atividade _____	

\***Legendas:** 0-nunca, 1- executou com ajuda, 2- executou sozinho

14. Você tem aulas de informática na escola?  sim  não

a) A partir de qual série: \_\_\_\_\_ b) Qual a carga horária: \_\_\_\_\_ (aulas/hora)

15. Você utiliza o computador para trabalhos acadêmicos?  sim  não

**Entretenimento:**

16. Você já jogou vídeo-game?  sim  não

Qual:  Playstation  X-box  Nintendo  Outro: \_\_\_\_\_

17. Você possui algum vídeo-game?  sim  não

Qual:  Playstation  X-box  Nintendo  Outro: \_\_\_\_\_

18. Quais tipos de jogos costuma jogar

ação  RPG  futebol  corrida  musicais  aventura  luta  tiro  esportes  estratégia

Outro: \_\_\_\_\_

19. Quantas horas costuma jogar por semana: \_\_\_\_\_ hora(s) por semana

**MATRIZES PROGRESSIVAS DE RAVEN**

A			AB			B				
01			01			01				
02			02			02				
03			03			03				
04			04			04			Idade cronológica	
05			05			05			Pontos	
06			06			06			Nível	
07			07			07			Percentil	
08			08			08				
09			09			09				
10			10			10				
11			11			11				
12			12			12				

**Questionários Pais:****Formulário:**

Nome da criança: \_\_\_\_\_ Sexo: ( ) Feminino ( ) Masculino

Data de nascimento: \_\_\_\_/\_\_\_\_/\_\_\_\_ Idade: \_\_\_\_\_ Escolaridade(\*): \_\_\_\_\_

Endereço: \_\_\_\_\_

Bairro: \_\_\_\_\_ CEP: \_\_\_\_\_-\_\_\_\_

Telefone: \_\_\_\_\_

Nome do pai: \_\_\_\_\_

Profissão: \_\_\_\_\_

Escolaridade(\*): \_\_\_\_\_

Nome da mãe: \_\_\_\_\_

Profissão: \_\_\_\_\_

Escolaridade(\*): \_\_\_\_\_

Nome do responsável: \_\_\_\_\_

Parentesco: \_\_\_\_\_

**\* escolaridade= anotar total de anos que estudou****Histórico:**

1. Toma algum remédio controlado? ( ) Não ( ) Sim-Qual(is): \_\_\_\_\_

\_\_\_\_\_

2. A criança já fez algum tratamento especializado? ( ) Não ( ) Sim

Fisioterápico ( ) Fonoaudiólogo ( ) Terapeuta Ocupacional ( ) Psicológico ( ) Neurológico  
( ) Ortopédico ( ) Outros ( ) Tipo/motivo: \_\_\_\_\_

\_\_\_\_\_

3. Seu filho(a) já usou um computador? Não ( ) Sim ( ).

4. Se sim, com que idade seu/sua filho(a) utilizou um computador pela primeira vez? \_\_\_\_\_

5. Nos últimos 3 meses, quantas horas aproximadamente ele(a) gastou no computador *por semana?*

\_\_\_\_\_ horas por semana.

6. Seu/sua filho(a) joga videogame? ( ) Sim ( ) Não

7. Se sim, nos últimos 3 meses, quantas horas aproximadamente ele(a) gastou jogando *por semana?*

\_\_\_\_\_ horas por semana.

## SNAP-IV

NOME:

SÉRIE:

IDADE:        anos

Para cada item, escolha a coluna que **melhor** descreve o(a) seu(sua) filho/filha (marque com um X).  
Considere

	Nem um pouco	Só um pouco	Bastante	Demais
1. Não consegue prestar muita atenção a detalhes ou comete erros por descuido nos trabalhos da escola ou tarefas.				
2. Tem dificuldade de manter a atenção em tarefas ou atividades de lazer.				
3. Parece não estar ouvindo quando se fala diretamente com ele.				
4. Não segue instruções até o fim e não termina deveres de escola, tarefas ou obrigações.				
5. Tem dificuldade para organizar tarefas e atividades.				
6. Evita, não gosta ou se envolve contra a vontade em tarefas que exigem esforço mental prolongado.				
7. Perde coisas necessárias para atividades (p. ex.: brinquedos, deveres da escola, lápis ou livros).				
8. Distrai-se com estímulos externos.				
9. É esquecido em atividade do dia a dia.				
10. Mexe com as mãos ou os pés ou se remexe na cadeira.				
11. Sai do lugar na sala de aula ou em outras situações em que se espera que fique sentado.				
12. Corre de um lado para outro ou sobe demais nas coisas em situações em que isto é inapropriado.				
13. Tem dificuldade em brincar ou envolver-se em atividades de lazer de forma calma.				
14. Não para ou frequentemente está a “mil por hora”.				
15. Fala em excesso.				
16. Responde as perguntas de forma precipitada antes de elas terem sido terminadas.				
17. Tem dificuldade de esperar sua vez.				
18. Interrompe os outros ou se intromete (p. ex., mete-se nas conversas/jogos).				
<i>Versão em Português validada por Mattos P et al, 2005.</i>				

19. Alguns desses sintomas estavam presentes antes dos 7 anos de idade? ( ) Sim ( ) Não  
Quais (anote o número da questão que descreve o sintoma)?

---

20. Existem problemas causados pelos sintomas acima em pelo menos 2 contextos diferentes (na escola, na vida social ou em casa)? ( ) Sim ( ) Não

21. Esses sintomas causam prejuízo? ( ) Sim ( ) Não

## Questionário de Capacidades e Dificuldades (SDQ-Port)

Instruções: Por favor, em cada item marque com uma cruz o quadrado que melhor descreva a criança. Responda a todas as perguntas da melhor maneira possível, mesmo que você não tenha certeza absoluta ou se a pergunta lhe parecer estranha. Dê suas respostas com base no comportamento da criança nos últimos seis meses ou durante o ano escolar em curso.

Nome da Criança .....

Masculino/Feminino

Data de Nascimento .....

	Mais ou menos		
	Falso	verdadeiro	Verdadeiro
Tem consideração pelos sentimentos de outras pessoas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Não consegue parar sentado quando tem que fazer a lição ou comer; mexe-se muito, esbarrando em coisas, derrubando coisas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Muitas vezes se queixa de dor de cabeça, dor de barriga ou enjôo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tem boa vontade em compartilhar doces, brinquedos, lápis ... com outras crianças	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frequentemente tem acessos de raiva ou crises de birra	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
É solitário, prefere brincar sozinho	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Geralmente é obediente e faz normalmente o que os adultos lhe pedem	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tem muitas preocupações, muitas vezes parece preocupado com tudo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tenta ser atencioso se alguém parece magoado, aflito ou se sentindo mal	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Está sempre agitado, balançando as pernas ou mexendo as mãos	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tem pelo menos um bom amigo ou amiga	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frequentemente briga com outras crianças ou as amendronta	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frequentemente parece triste, desanimado ou choroso	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Em geral, é querido por outras crianças	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Facilmente perde a concentração	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fica inseguro quando tem que fazer alguma coisa pela primeira vez, facilmente perde a confiança em si mesmo	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
É gentil com crianças mais novas	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frequentemente engana ou mente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Outras crianças 'pegam no pé' ou atormentam-no	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Frequentemente se oferece para ajudar outras pessoas (pais, professores, outras crianças)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pensa nas coisas antes de fazê-las	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rouba coisas de casa, da escola ou de outros lugares	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Se dá melhor com adultos do que com outras crianças	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tem muitos medos, assusta-se facilmente	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Completa as tarefas que começa, tem boa concentração	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Nome completo (em letra de forma) .....

Data .....

Mãe/pai/professor/outro (especifique):

**Muito obrigado pela sua colaboração**

**Dados sócio-demográficos:** (Marque um X na opção de acordo com o que possui. Por exemplo, se tem uma televisão em casa, marque um X na coluna “Um”; se tiver duas, marque um X na coluna “Dois” e assim por diante.)

O(A) senhor(a) tem em sua casa...	Quantidade de itens				
	Não tem	Um	Dois	Três	Quatro ou +
Televisão em cores					
Rádio					
Banheiro					
Automóvel					
Empregada mensalista					
Máquina de lavar					
Videocassete e/ou DVD					
Geladeira					
Freezer (aparelho independente ou parte de geladeira duplex)					

Qual a instrução do chefe da família?

Marque X	Nomenclatura Antiga	Nomenclatura Atual
	Analfabeto/ Primário incompleto	Analfabeto/ Até 3ª série Fundamental (ou 1º. Grau)
	Primário completo/ Ginásial incompleto	Até 4ª série Fundamental (ou 1º. Grau)
	Ginásial completo/ Colegial incompleto	Fundamental (ou 1º. Grau) completo
	Colegial completo/ Superior incompleto	Médio (ou 2º. Grau) completo
	Superior completo	Superior completo

**Agradecemos sua colaboração. Por gentileza, coloque este formulário, os questionários em anexo (QDQ, SNAP-IV, SDQ e SSRS) e o Termo de Consentimento devidamente assinado e datado no envelope e envie pelo(a) seu/sua filho(a) para a escola.**

Atenciosamente,

\_\_\_\_\_  
Daniel Foschetti Gontijo  
Psicólogo CRP 04/31961

\_\_\_\_\_  
Suzan Caroline Such Ribeiro  
Psicóloga CRP 04/30382

## ANEXO G- Convite aos Pais

Prezados Pais e/ou Responsáveis,

Seu(sua) filho(a) está sendo convidado(a) a participar de um estudo em que vamos aplicar alguns testes computadorizados e manuais. Esses testes servem para avaliar algumas habilidades do(a) aluno(a) como atenção, memória e autocontrole. O(a) aluno(a) se ausentará por pouco tempo da sala de aula, não sendo prejudicado por isso. A direção da escola estará ciente da participação do aluno. Somente a equipe responsável pelo projeto tem acesso aos dados do estudo. O resultado final estará disponível na escola.

Para que seu(sua) filho(a) possa participar, solicitamos que leia com atenção o Termo de Consentimento em anexo e envie pelo seu filho para escola devidamente datado e assinado. Além disso, preencha os questionários que estão junto com o Termo de Consentimento sobre informações da sua família e seu filho(a).

Desde já, agradecemos por sua colaboração.  
Atenciosamente,

---

Daniel Foschetti Gontijo  
Psicólogo CRP 04/31961

---

Suzan Caroline Such Ribeiro  
Psicóloga CRP 04/30382

## ANEXO H- Termo de Consentimento

### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

(Para pais e/ou responsáveis)

#### Avaliação neuropsicológica e de habilidades sociais de crianças do ensino fundamental

Seu/sua filho(a) está sendo convidado a participar da pesquisa intitulada “Avaliação neuropsicológica e de habilidades sociais de crianças do ensino fundamental”. Antes de aceitar participar desta pesquisa, é importante que você leia e compreenda a explicação sobre os procedimentos propostos. Esta declaração descreve o objetivo, os procedimentos, os benefícios e riscos do estudo, bem como o seu direito de sair do estudo a qualquer momento. Nenhuma garantia ou promessa pode ser feita sobre o resultado do estudo. Estas informações estão sendo fornecidas para esclarecer quaisquer dúvidas sobre a pesquisa antes de você aceitar o convite para participar.

O objetivo deste estudo é avaliar as habilidades sociais (empatia, evitação de problemas, responsabilidade etc.), o autocontrole e as funções executivas (tomada de decisão, memória, atenção etc.) de crianças, investigando possíveis relações entre esses domínios. A pesquisa também rastreará crianças com suspeita de apresentarem Transtorno de Déficit de Atenção/Hiperatividade (TDAH), que é caracterizado por sintomas de desatenção, hiperatividade e/ou impulsividade.

Este estudo requer que você preencha alguns questionários a respeito do comportamento de seu/sua filho(a) que responda algumas perguntas sobre sua relação com ele(a). A avaliação de seu/sua filho(a) será dividida em dois dias, tendo duração média de 60 (sessenta) minutos cada.

As informações obtidas nesta pesquisa contribuirão para validar testes computadorizados mais eficazes para a avaliação e acompanhamento de diversas dificuldades intelectuais e comportamentais de crianças. Além do mais, o conhecimento adquirido será importante para verificar possíveis relações entre as habilidades sociais, as funções executivas e o autocontrole, o que contribuirá para a promoção futura de intervenções com crianças que têm problema em algum desses domínios.

Os riscos que essa pesquisa pode causar aos participantes são apenas a possibilidade de algum constrangimento em virtude das questões propostas (por exemplo, quando indagados sobre alguns de seus comportamentos), mas faremos o máximo para evitar que isso ocorra.

Os registros de sua participação neste estudo serão mantidos confidencialmente até onde é permitido por lei, e todas as informações estarão restritas à equipe responsável pelo projeto. No entanto, o pesquisador e, sob certas circunstâncias, o Comitê de Ética em Pesquisa/UFMG poderão verificar e ter acesso aos dados confidenciais que identificam o participante pelo nome. Contudo, qualquer publicação dos dados não o identificará. Ao assinar este formulário de consentimento, você autoriza o pesquisador a fornecer os registros de seu/sua filho(a) para o Comitê de Ética em Pesquisa/UFMG.

A participação neste estudo de seu/sua filho(a) é voluntária e sua recusa em participar ou seu desligamento do estudo não os prejudicará de nenhuma forma. Você não receberá qualquer compensação financeira por sua participação no estudo.

Durante o estudo, se você tiver qualquer dúvida ou apresentar qualquer problema médico, entre em contato com o Dr. Arthur Melo e Kummer pelo telefone 3409-9785, ou com a Comissão de Ética da Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627, Unidade Administrativa II – 2º andar, Campus Pampulha, Belo Horizonte, MG – Brasil, Cep: 31270-901, pelo telefone 3409-4592.

**Consentimento:** Declaro que li e entendi as informações precedentes. Este formulário está sendo assinado voluntariamente por mim e meu/minha filho(a), indicando o nosso consentimento para ele(a) participar do estudo, até que nós decidamos o contrário.

Belo Horizonte, \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_  
DIA MÊS ANO

Assinatura da criança \_\_\_\_\_

Assinatura do responsável pelo participante \_\_\_\_\_

Assinatura do pesquisador \_\_\_\_\_

## ANEXO I- Principais Tabelas de Análises

### Comparações com Diagnóstico SNAP-IV:

#### Diagnostico SNAP

	Frequência	Porcentual	Porcentagem válida	Porcentagem acumulativa
Válido Não	23	46,0	46,0	46,0
Válido Sim	27	54,0	54,0	100,0
Total	50	100,0	100,0	

#### Idade e Critério Socioeconômico

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
Idade	Não	23	8,26	1,176	,245
	Sim	27	8,93	1,238	,238
Crit. Socio econ pontos	Não	23	23,96	6,498	1,355
	Sim	27	21,48	5,345	1,029

#### Idade e Critério Socioeconômico

	Idade	Crit. Socio econ pontos
Mann-Whitney U	183,000	226,000
Wilcoxon W	459,000	604,000
Z	-2,855	-1,649
Sig. Assint. (2 caudas)	,004	,099
Sig exata (2 extremidades)	,004	,100
Sig exata (1 extremidade)	,002	,050
Probabilidade de ponto	,000	,001

a. Variável de agrupamento: Diagnostico SNAP

#### RAVEN

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
Raven-Pontos	Não	23	22,96	5,389	1,124
	Sim	27	21,78	5,323	1,024
Raven-Percentil	Não	23	1,65	,775	,162
	Sim	27	1,85	,770	,148
Raven-Classificação	Não	23	2,74	1,544	,322
	Sim	27	3,37	1,445	,278

#### RAVEN- Mann-Whitney

	Raven-Pontos	Raven-Percentil	Raven-Classificação
Mann-Whitney U	273,000	264,500	238,000
Wilcoxon W	651,000	540,500	514,000
Z	-,732	-,970	-1,441
Sig. Assint. (2 caudas)	,464	,332	,150
Sig exata (2 extremidades)	,471	,357	,151
Sig exata (1 extremidade)	,235	,179	,075
Probabilidade de ponto	,003	,023	,003

#### BCST

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
BCST-N categorias	Não	23	3,57	1,472	,307
	Sim	27	7,19	24,200	4,657
BCST-N cartas	Não	23	125,22	9,342	1,948
	Sim	27	125,15	14,819	2,852
BCST-R corretas	Não	23	76,91	11,532	2,405
	Sim	27	64,62	18,339	3,529
BCST-% R corretas	Não	23	61,76	10,124	2,111
	Sim	27	52,23	14,634	2,816
BCST-erros	Não	23	48,30	14,089	2,938

	Sim	27	62,34	17,663	3,399
BCST-% erros	Não	23	38,24	10,123	2,111
	Sim	27	47,58	14,985	2,884
BCST-R perseverativas	Não	23	44,78	12,674	2,643
	Sim	27	40,97	22,516	4,333
BCST-% R perseverativas	Não	23	35,69	9,359	1,952
	Sim	27	32,11	17,515	3,371
BCST-E perseverativos	Não	23	25,26	9,588	1,999
	Sim	27	25,01	15,644	3,011
BCST-% E perseverativos	Não	23	20,08	7,197	1,501
	Sim	27	21,62	14,707	2,830
BCST-E não perseverativos	Não	23	23,04	11,853	2,472
	Sim	27	37,33	22,369	4,305
BCST-% E não perseverativos	Não	23	18,13	9,026	1,882
	Sim	27	28,60	17,395	3,348
BCST-Erros unicos	Não	23	7,91	6,302	1,314
	Sim	27	14,21	13,194	2,539

## BCST

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
BCST-% Erros unicos	Não	23	6,21	4,892	1,020
	Sim	27	14,23	19,854	3,821
BCST-Cartas 1 categoria	Não	23	30,70	27,418	5,717
	Sim	27	21,11	20,767	3,997
BCST-Falha manter contexto	Não	23	2,52	1,780	,371
	Sim	26	2,00	1,414	,277
BCST-Aprendendo a aprender	Não	16	,56	7,764	1,941
	Sim	12	1,11	9,669	2,791
BCST-Nivel conceitual	Não	23	61,91	14,835	3,093
	Sim	26	47,77	21,332	4,184

## BCST- Mann-Whitney

	BCST-N categorias	BCST-N cartas	BCST-R corretas	BCST-% R corretas	BCST-erros	BCST-% erros	BCST-R perseverativas	BCST-% R perseverativas	BCST-E perseverativos
Mann-Whitney U	226,000	283,000	190,500	195,000	178,500	196,000	294,000	287,000	304,000
Wilcoxon W	604,000	559,000	568,500	573,000	454,500	472,000	672,000	665,000	580,000
Z	-1,694	-1,138	-2,337	-2,249	-2,571	-2,230	-,321	-,458	-,127
Sig. Assint. (2 caudas)	,090	,255	,019	,024	,010	,026	,748	,647	,899
Sig exata [2*(Sig. de 1-cauda)]									
Sig exata (2 extremidades)	,091	,322	,019	,024	,009	,025	,754	,653	,904
Sig exata (1 extremidade)	,046	,246	,009	,012	,005	,013	,377	,327	,452
Probabilidade de ponto	,001	,052	,000	,000	,000	,000	,004	,003	,004

## BCST- Mann-Whitney

	BCST-% E perseverativos	BCST-E não perseverativos	BCST-% E não perseverativos	BCST-Erros unicos	BCST-% Erros unicos	BCST-Cartas 1 categoria	BCST-Falha manter contexto	BCST-Aprendendo a aprender	BCST-Nível conceitual
Mann-Whitney U	288,000	189,500	198,500	222,500	223,000	238,000	252,500	91,500	176,000
Wilcoxon W	564,000	465,500	474,500	498,500	499,000	616,000	603,500	169,500	527,000
Z	-,438	-2,356	-2,181	-1,716	-1,705	-1,416	-,947	-,209	-2,465
Sig. Assint. (2 caudas)	,661	,018	,029	,086	,088	,157	,344	,834	,014
Sig exata [2*(Sig. de 1-cauda)]								,837	
Sig exata (2 extremidades)	,667	,018	,029	,087	,089	,160	,348	,846	,013
Sig exata (1 extremidade)	,334	,009	,014	,044	,045	,080	,174	,423	,006
Probabilidade de ponto	,004	,000	,000	,001	,001	,001	,003	,009	,000

a. Variável de agrupamento: Diagnostico SNAP

b. Não corrigido para vínculos.

## Donkey

	Diagnostico SNAP	N	Média	Desvio padrão	Erro padrão da média
Donkey-Total maçãs	Não	23	-20,78	26,841	5,597
	Sim	27	-6,70	40,087	7,715
Donkey-Portas c	Não	23	28,04	10,209	2,129
	Sim	27	25,78	10,653	2,050
Donkey-Portas v	Não	23	52,96	13,590	2,834
	Sim	27	53,48	11,882	2,287
Donkey-Portas b	Não	23	32,57	7,223	1,506
	Sim	27	33,00	11,365	2,187
Donkey-Portas n	Não	23	36,17	12,791	2,667
	Sim	27	37,74	12,066	2,322
Donkey-Portas c+v	Não	23	81,00	14,206	2,962

	Sim	27	79,26	16,559	3,187
Donkey-Portas b+n	Não	23	68,74	13,755	2,868
	Sim	27	70,74	16,559	3,187
Donkey-Vantajosas - Desvantajosas	Não	23	-12,26	27,937	5,825
	Sim	27	-8,52	33,118	6,374

**Donkey- Mann Whitney**

	Donkey-Total maçãs	Donkey-Portas c	Donkey-Portas v	Donkey-Portas b	Donkey-Portas n	Donkey-Portas c+v	Donkey-Portas b+n	Donkey-Vantajosas - Desvantajosas
Mann-Whitney U	209,000	244,500	288,000	293,000	263,500	273,000	272,500	273,000
Wilcoxon W	485,000	622,500	564,000	569,000	539,500	651,000	548,500	549,000
Z	-1,976	-1,287	-,438	-,341	-,916	-,731	-,740	-,731
Sig. Assint. (2 caudas)	<b>,048</b>	,198	,661	,733	,360	,465	,459	,465
Sig exata (2 extremidades)	,048	,202	,667	,739	,365	,471	,465	,471
Sig exata (1 extremidade)	,024	,101	,334	,369	,183	,236	,233	,236
Probabilidade de ponto	,001	,002	,004	,004	,003	,003	,003	,003

a. Variável de agrupamento: Diagnostico SNAP

**Teste de Dígitos**

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
DspanD-Memory span	Não	23	4,13	1,180	,246
	Sim	27	4,00	1,144	,220
DspanD-Total corretos	Não	23	5,30	2,225	,464
	Sim	27	5,19	2,149	,414
DspanD-Tempo (min)	Não	23	267,87	72,685	15,156
	Sim	26	294,27	81,416	15,967
DspanI-Memory span	Não	23	3,04	1,224	,255
	Sim	27	2,74	1,095	,211
DspanI-Total corretos	Não	23	3,30	2,098	,438
	Sim	27	2,59	1,866	,359
DspanI-Tempo (min)	Não	23	262,91	115,886	24,164
	Sim	27	230,30	88,820	17,093

**Teste de Dígitos- Mann-Whitney**

	DspanD-Memory span	DspanD-Total corretos	DspanD-Tempo (min)	DspanI-Memory span	DspanI-Total corretos	DspanI-Tempo (min)
Mann-Whitney U	277,000	294,500	254,500	269,500	250,500	260,000
Wilcoxon W	655,000	672,500	530,500	647,500	628,500	638,000
Z	-,680	-,315	-,892	-,823	-1,181	-,983
Sig. Assint. (2 caudas)	,497	,753	,373	,411	,238	,326

Sig exata (2 extremidades)	,508	,759	,379	,420	,242	,331
Sig exata (1 extremidade)	,250	,379	,189	,210	,121	,166
Probabilidade de ponto	,006	,004	,003	,006	,002	,002

a. Variável de agrupamento: Diagnostico SNAP

### Stroop

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
StroopE-Total	Não	23	45,30	23,338	4,866
	Sim	24	47,71	32,047	6,542
StroopE-Incroguentes	Não	23	14,48	7,153	1,492
	Sim	24	15,00	9,873	2,015
StroopE-Aleatorios	Não	23	33,22	20,427	4,259
	Sim	24	34,92	29,228	5,966

### Stroop- Mann- Whitney

	StroopE-Total	StroopE-Incroguentes	StroopE-Aleatorios
Mann-Whitney U	265,500	270,500	261,500
Wilcoxon W	541,500	570,500	561,500
Z	-,224	-,117	-,309
Sig. Assint. (2 caudas)	,823	,907	,758
Sig exata (2 extremidades)	,829	,912	,764
Sig exata (1 extremidade)	,414	,456	,382
Probabilidade de ponto	,004	,004	,004

a. Variável de agrupamento: Diagnostico SNAP

### Torre de Londres

Diagnostico SNAP		N	Média	Desvio padrão	Erro padrão da média
Pontuação Final	Não	23	20,74	6,837	1,426
	Sim	27	16,81	7,621	1,467
Total Tentativas	Não	23	23,91	4,284	,893
	Sim	27	25,96	4,587	,883

### Torre de Londres – Mann-Whitney

	Pontuação Final	Total Tentativas
Mann-Whitney U	208,500	220,000
Wilcoxon W	586,500	496,000
Z	-1,988	-1,767
Sig. Assint. (2 caudas)	,047	,077
Sig exata (2 extremidades)	,047	,078
Sig exata (1 extremidade)	,023	,039
Probabilidade de ponto	,001	,001

### Comparações com Pontuação SNAP desatenção:

#### Correlações Idade e Critério Socioeconômico

			SNAP Pont Desatenção	SNAP Pont Hiperatividade	Idade	Crit. Socio econ pontos	
Rô de Spearman	SNAP Pont Desatenção	Correlações de	1,000	,660**	,314*	-,267	
		coeficiente					
		Sig. (2		,000	,027	,061	
			N	50	50	50	50
	SNAP Pont Hiperatividade	Correlações de	,660**	1,000	,316*	-,212	
		coeficiente					
		Sig. (2	,000	.	,026	,140	
			N	50	50	50	50
	Idade	Correlações de	,314*	,316*	1,000	-,249	
		coeficiente					
		Sig. (2	,027	,026	.	,081	
			N	50	50	50	50
Crit. Socio econ pontos	Correlações de	-,267	-,212	-,249	1,000		
	coeficiente						
	Sig. (2	,061	,140	,081	.		
		N	50	50	50	50	

#### Correlações Raven

			SNAP Pont Desatenção	SNAP Pont Hiperatividade	Raven- Pontos	Rave n- Perc entil	Raven- Classificaçã o
Rô de Spearman	SNAP Pont Desatenção	Correlações de	1,000	,660**	-,222	,234	,294*
		coeficiente					
		Sig. (2		,000	,121	,102	,038
			N	50	50	50	50
	SNAP Pont Hiperatividade	Correlações de	,660**	1,000	-,134	,243	,284*
		coeficiente					
Sig. (2							

	Sig. (2 extremidades)	,000	.	,352	,089	,046
	N	50	50	50	50	50
Raven-Pontos	Correlações de coeficiente	-,222	-,134	1,000	,805**	-,883**
	Sig. (2 extremidades)	,121	,352	.	,000	,000
	N	50	50	50	50	50
Raven-Percentil	Correlações de coeficiente	,234	,243	-,805**	1,000	,943**
	Sig. (2 extremidades)	,102	,089	,000	.	,000
	N	50	50	50	50	50
Raven-Classificação	Correlações de coeficiente	,294*	,284*	-,883**	,943**	1,000
	Sig. (2 extremidades)	,038	,046	,000	,000	.
	N	50	50	50	50	50

## Correlações BCST

		SNAP Pont Desatenção	SNAP Pont Hiperatividade	BCST-N categorias	BCST-N cartas	BCST-R corretas	BCST-% R corretas	
Rô de Spearman	SNAP Pont Desatenção	Correlações de coeficiente	1,000	,660**	-,179	,120	-,193	-,154
		Sig. (2 extremidades)	.	,000	,212	,408	,179	,286
		N	50	50	50	50	50	50
	SNAP Pont Hiperatividade	Correlações de coeficiente	,660**	1,000	-,243	,223	-,172	-,243
		Sig. (2 extremidades)	,000	.	,089	,119	,231	,090
		N	50	50	50	50	50	50
	BCST-N categorias	Correlações de coeficiente	-,179	-,243	1,000	-,484**	,687**	,854**
		Sig. (2 extremidades)	,212	,089	.	,000	,000	,000
		N	50	50	50	50	50	50
	BCST-N cartas	Correlações de coeficiente	,120	,223	-,484**	1,000	-,025	-,466**
		Sig. (2 extremidades)	,408	,119	,000	.	,861	,001
		N	50	50	50	50	50	50
BCST-R corretas	Correlações de coeficiente	-,193	-,172	,687**	-,025	1,000	,845**	
	Sig. (2 extremidades)	,179	,231	,000	,861	.	,000	
	N	50	50	50	50	50	50	

BCST-% R corretas	Correlações de coeficiente	-,154	-,243	,854**	-,466**	,845**	1,000
	Sig. (2 extremidades)	,286	,090	,000	,001	,000	.
	N	50	50	50	50	50	50
BCST-erros	Correlações de coeficiente	,221	,243	-,804**	,321*	-,937**	-,958**
	Sig. (2 extremidades)	,123	,089	,000	,023	,000	,000
	N	50	50	50	50	50	50

## Correlações

			BCST-erros	BCST-% erros	BCST-R perseverativ as	BCST-% R perseverativ as	BCST-E perseverativ os	BCST- % E persev erativos
Rô de Spearman	SNAP Pont Desatenção	Correlações de coeficiente	,221	,150**	-,034	-,062	,012	,094
		Sig. (2 extremidades)	,123	,299	,812	,670	,935	,515
		N	50	50	50	50	50	50
	SNAP Pont Hiperatividade	Correlações de coeficiente	,243**	,241	,042	,019	-,006	-,023
		Sig. (2 extremidades)	,089	,092	,770	,896	,970	,874
		N	50	50	50	50	50	50
	BCST-N categorias	Correlações de coeficiente	-,804	-,854	,140	,173**	-,062**	,063**
		Sig. (2 extremidades)	,000	,000	,331	,229	,667	,665
		N	50	50	50	50	50	50
	BCST-N cartas	Correlações de coeficiente	,321	,466	,238**	,151	,269	-,019**
		Sig. (2 extremidades)	,023	,001	,096	,297	,059	,898
		N	50	50	50	50	50	50
	BCST-R corretas	Correlações de coeficiente	-,937	-,843	,046**	,038	-,204	-,316**
		Sig. (2 extremidades)	,000	,000	,749	,791	,155	,025
		N	50	50	50	50	50	50
	BCST-% R corretas	Correlações de coeficiente	-,958	-1,000	-,117**	-,085**	-,349**	-,233

BCST-erros	Sig. (2 extremidades)	,000	,000	,419	,557	,013	,103
	N	50	50	50	50	50	50
	Correlações de coeficiente	1,000	,955	,058**	,027*	,297**	,310**
	Sig. (2 extremidades)	.	,000	,688	,851	,036	,028

## Correlações

			BCST-E não perseverativos	BCST-% E não perseverativos	BCST- Erros unicos	BCST-% Erros unicos	BCST-Cartas 1 categoria	BCST- Falha manter contexto
Rô de Spearman	SNAP Desatenção	Correlações de coeficiente	,254	,215**	,328	,332	-,127	,049
		Sig. (2 extremidades)	,075	,134	,020	,018	,378	,740
		N	50	50	50	50	50	49
	SNAP Hiperatividade	Correlações de coeficiente	,183**	,168	,131	,121	-,372	,052
		Sig. (2 extremidades)	,204	,243	,364	,404	,008	,723
		N	50	50	50	50	50	49
	BCST-N categorias	Correlações de coeficiente	-,752	-,793	-,584	-,545**	-,115**	,149**
		Sig. (2 extremidades)	,000	,000	,000	,000	,426	,306
		N	50	50	50	50	50	49
	BCST-N cartas	Correlações de coeficiente	,251	,305	,208**	,147	,282	,154**
		Sig. (2 extremidades)	,079	,031	,148	,309	,048	,291
		N	50	50	50	50	50	49
	BCST-R corretas	Correlações de coeficiente	-,732	-,720	-,647**	-,648	,124	,661**
		Sig. (2 extremidades)	,000	,000	,000	,000	,389	,000
		N	50	50	50	50	50	49
	BCST-% R corretas	Correlações de coeficiente	-,704	-,739	-,624**	-,586**	-,035**	,549

BCST-erros	Sig. (2 extremidades)	,000	,000	,000	,000	,811	,000
	N	50	50	50	50	50	49
	Correlações de coeficiente	,775	,776	,689**	,673*	-,029**	-,549**
	Sig. (2 extremidades)	,000	,000	,000	,000	,844	,000

## Correlações

		BCST-Aprendendo a aprender	BCST-Nível conceitual	
Rô de Spearman	Correlações de coeficiente	,168	-,187**	
	SNAP Pont Desatenção	Sig. (2 extremidades)	,391	,198
	N	28	49	
	Correlações de coeficiente	,108**	-,217	
	SNAP Pont Hiperatividade	Sig. (2 extremidades)	,584	,135
	N	28	49	
	Correlações de coeficiente	,394	,825	
	BCST-N categorias	Sig. (2 extremidades)	,038	,000
	N	28	49	
	Correlações de coeficiente	-,349	-,279	
	BCST-N cartas	Sig. (2 extremidades)	,068	,052
	N	28	49	
	Correlações de coeficiente	-,090	,965	
	BCST-R corretas	Sig. (2 extremidades)	,650	,000
	N	28	49	
	Correlações de coeficiente	,195	,957	
BCST-% R corretas	Sig. (2 extremidades)	,321	,000	
N	28	49		
BCST-erros	Correlações de coeficiente	-,007	-,957	

Sig. (2 extremidades)

,972

,000

**Correlações Donkey**

		SNAP Pont Desatenção	SNAP Pont Hiperatividade	Donkey-Total maças	Donkey-Portas c	Donkey-Portas v	Donkey-Portas b
Rô de Spearman	Correlações de coeficiente	1,000	,660**	,127	-,115	,092	-,038
	SNAP Pont Desatenção	Sig. (2 extremidades)	.	,380	,428	,525	,793
	N	50	50	50	50	50	50
	Correlações de coeficiente	,660**	1,000	-,037	-,059	,125	-,100
	SNAP Pont Hiperatividade	Sig. (2 extremidades)	,000	.	,797	,683	,387
	N	50	50	50	50	50	50
	Correlações de coeficiente	,127	-,037	1,000	-,267	-,419**	,398**
	Donkey-Total maçãs	Sig. (2 extremidades)	,380	,797	.	,061	,002
	N	50	50	50	50	50	50
	Correlações de coeficiente	-,115	-,059	-,267	1,000	-,442**	,231
	Donkey-Portas c	Sig. (2 extremidades)	,428	,683	,061	.	,001
	N	50	50	50	50	50	50
	Correlações de coeficiente	,092	,125	-,419**	-,442**	1,000	-,597**
	Donkey-Portas v	Sig. (2 extremidades)	,525	,387	,002	,001	.
	N	50	50	50	50	50	50
	Correlações de coeficiente	-,038	-,100	,398**	,231	-,597**	1,000
	Donkey-Portas b	Sig. (2 extremidades)	,793	,490	,004	,107	,000
	N	50	50	50	50	50	50
	Correlações de coeficiente	,101	-,016	,460**	-,477**	-,274	-,250
	Donkey-Portas n	Sig. (2 extremidades)	,485	,913	,001	,000	,054

**Correlações**

			Donkey-Portas n	Donkey-Portas c+v	Donkey-Portas b+n	Donkey-Vantajosas - Desvantajosas
Rô de Spearman	SNAP Pont Desatenção	Correlações de coeficiente	,101	-,086**	,086	,086
		Sig. (2 extremidades)	,485	,554	,550	,554
		N	50	50	50	50
	SNAP Pont Hiperatividade	Correlações de coeficiente	-,016**	,035	-,034	-,035
		Sig. (2 extremidades)	,913	,810	,817	,810
		N	50	50	50	50
	Donkey-Total maçãs	Correlações de coeficiente	,460	-,694	,694	,694
		Sig. (2 extremidades)	,001	,000	,000	,000
		N	50	50	50	50
	Donkey-Portas c	Correlações de coeficiente	-,477	,293	-,293	-,293
		Sig. (2 extremidades)	,000	,039	,039	,039
		N	50	50	50	50
	Donkey-Portas v	Correlações de coeficiente	-,274	,620	-,620**	-,620**
		Sig. (2 extremidades)	,054	,000	,000	,000
		N	50	50	50	50
	Donkey-Portas b	Correlações de coeficiente	-,250	-,405	,407**	,405
		Sig. (2 extremidades)	,080	,004	,003	,004
		N	50	50	50	50
	Donkey-Portas n	Correlações de coeficiente	1,000	-,695	,694**	,695**
		Sig. (2 extremidades)	.	,000	,000	,000

## Correlações Dspan

			SNAP Pont Desatenção	SNAP Pont Hiperatividade	DspanD-Memory span	DspanD-Total corretos	DspanD-Tempo (min)	DspanI-Memory span
Rô de Spearman	SNAP Pont Desatenção	Correlações de coeficiente	1,000	,660**	-,199	-,154	,105	-,159
		Sig. (2 extremidades)	.	,000	,166	,286	,474	,270

	N	50	50	50	50	49	50
	Correlações de coeficiente	,660**	1,000	-,019	,042	,102	-,105
SNAP Pont Hiperatividade	Sig. (2 extremidades)	,000	.	,895	,774	,487	,468
	N	50	50	50	50	49	50
	Correlações de coeficiente	-,199	-,019	1,000	,927**	,646**	,515**
DspanD-Memory span	Sig. (2 extremidades)	,166	,895	.	,000	,000	,000
	N	50	50	50	50	49	50
	Correlações de coeficiente	-,154	,042	,927**	1,000	,550**	,522**
DspanD-Total corretos	Sig. (2 extremidades)	,286	,774	,000	.	,000	,000
	N	50	50	50	50	49	50
	Correlações de coeficiente	,105	,102	,646**	,550**	1,000	,407**
DspanD-Tempo (min)	Sig. (2 extremidades)	,474	,487	,000	,000	.	,004
	N	49	49	49	49	49	49
	Correlações de coeficiente	-,159	-,105	,515**	,522**	,407**	1,000
DspanI-Memory span	Sig. (2 extremidades)	,270	,468	,000	,000	,004	.
	N	50	50	50	50	49	50
	Correlações de coeficiente	-,219	-,128	,547**	,530**	,415**	,952**
DspanI-Total corretos	Sig. (2 extremidades)	,126	,374	,000	,000	,003	,000
	N	50	50	50	50	49	50

## Correlações

			DspanI-Total corretos	DspanI-Tempo (min)
Rô de Spearman		Correlações de coeficiente	-,219	-,119**
	SNAP Pont Desatenção	Sig. (2 extremidades)	,126	,409
		N	50	50
	SNAP Pont Hiperatividade	Correlações de coeficiente	-,128**	-,054
		Sig. (2 extremidades)	,374	,707

	N	50	50
	Correlações de coeficiente	,547	,364
DspanD-Memory span	Sig. (2 extremidades)	,000	,009
	N	50	50
	Correlações de coeficiente	,530	,322
DspanD-Total corretos	Sig. (2 extremidades)	,000	,023
	N	50	50
	Correlações de coeficiente	,415	,429
DspanD-Tempo (min)	Sig. (2 extremidades)	,003	,002
	N	49	49
	Correlações de coeficiente	,952	,818
DspanI-Memory span	Sig. (2 extremidades)	,000	,000
	N	50	50
	Correlações de coeficiente	1,000	,757
DspanI-Total corretos	Sig. (2 extremidades)	.	,000
	N	50	50

## Correlações Stroop

		SNAP Pont Desatenção	SNAP Pont Hiperatividade	StroopE-Total	StroopE- Incroguentes	StroopE- Aleatorios
Rô de Spearman	Correlações de coeficiente	1,000	,660**	,279	,241	,177
	<b>SNAP Pont Desatenção</b>	.	,000	,057	,103	,233
	N	50	50	47	47	47
	Correlações de coeficiente	,660**	1,000	,116	,179	,036
	SNAP Pont Hiperatividade	,000	.	,438	,228	,808
	N	50	50	47	47	47
StroopE-Total	Correlações de coeficiente	,279	,116	1,000	,532**	,959**

	Sig. (2 extremidades)	,057	,438	.	,000	,000
	N	47	47	47	47	47
	Correlações de coeficiente	,241	,179	,532**	1,000	,351*
StroopE-Incroguentes	Sig. (2 extremidades)	,103	,228	,000	.	,016
	N	47	47	47	47	47
	Correlações de coeficiente	,177	,036	,959**	,351*	1,000
StroopE-Aleatorios	Sig. (2 extremidades)	,233	,808	,000	,016	.
	N	47	47	47	47	47

## Correlações Torre de Londres

		SNAP Pont Desatenção	SNAP Pont Hiperatividade	Pontuação Final	Total Tentativas	
Rô de Spearman	Correlações de coeficiente	1,000	,660**	-,295*	,277	
	SNAP Pont Desatenção	Sig. (2 extremidades)	.	,000	,038	,051
		N	50	50	50	50
	Correlações de coeficiente	,660**	1,000	-,253	,214	
	SNAP Pont Hiperatividade	Sig. (2 extremidades)	,000	.	,077	,136
		N	50	50	50	50
	Correlações de coeficiente	-,295*	-,253	1,000	-,983**	
	Pontuação Final	Sig. (2 extremidades)	,038	,077	.	,000
		N	50	50	50	50
	Correlações de coeficiente	,277	,214	-,983**	1,000	
	Total Tentativas	Sig. (2 extremidades)	,051	,136	,000	.
		N	50	50	50	50

**Comparações com SDQ Problemas de Conduta, Hiperatividade e Total:**

## Correlações Idade e Critério Socioeconômico

			SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	Idade	Crit. Socio econ pontos
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	,141	-,355*
		Sig. (2 extremidades)	.	,000	,000	,340	,013
		N	48	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,536**	1,000	,796**	,324*	-,279
		Sig. (2 extremidades)	,000	.	,000	,025	,055
		N	48	48	48	48	48
	SDQ total	Correlações de coeficiente	,671**	,796**	1,000	,334*	-,429**
		Sig. (2 extremidades)	,000	,000	.	,020	,002
		N	48	48	48	48	48
	Idade	Correlações de coeficiente	,141	,324*	,334*	1,000	-,249
		Sig. (2 extremidades)	,340	,025	,020	.	,081
		N	48	48	48	50	50
	Crit. Socio econ pontos	Correlações de coeficiente	-,355*	-,279	-,429**	-,249	1,000
		Sig. (2 extremidades)	,013	,055	,002	,081	.
		N	48	48	48	50	50

## Correlações Raven

			SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	Raven- Pontos	Raven- Percentil	Raven- Classificação
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	-,151	,210	,254
		Sig. (2 extremidades)	.	,000	,000	,306	,151	,082
		N	48	48	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,536**	1,000	,796**	-,143	,236	,265
		Sig. (2 extremidades)	,000	.	,000	,333	,106	,069
		N	48	48	48	48	48	48
	SDQ total	Correlações de coeficiente	,671**	,796**	1,000	-,245	,270	,343*
		Sig. (2 extremidades)	,000	,000	.	,093	,063	,017
		N	48	48	48	48	48	48
	Raven-Pontos	Correlações de coeficiente	-,151	-,143	-,245	1,000	-,805**	-,883**
		Sig. (2 extremidades)	,306	,333	,093	.	,000	,000
		N	48	48	48	50	50	50
	Raven-Percentil	Correlações de coeficiente	,210	,236	,270	-,805**	1,000	,943**
		Sig. (2 extremidades)	,151	,106	,063	,000	.	,000

	N	48	48	48	50	50	50
Raven-Classificação	Correlações de coeficiente	,254	,265	,343*	-,883**	,943**	1,000
	Sig. (2 extremidades)	,082	,069	,017	,000	,000	.
	N	48	48	48	50	50	50

## Correlações BCST

		SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	BCST-N categorias	BCST-N cartas	BCST-R corretas	BCST-% R corretas	
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	-,258	,241	-,206	-,304*
		Sig. (2 extremidades)	.	,000	,000	,076	,098	,161	,036
		N	48	48	48	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,536**	1,000	,796**	-,142	,186	-,156	-,165
		Sig. (2 extremidades)	,000	.	,000	,336	,206	,290	,262
		N	48	48	48	48	48	48	48
	SDQ total	Correlações de coeficiente	,671**	,796**	1,000	-,222	,197	-,201	-,204
		Sig. (2 extremidades)	,000	,000	.	,129	,179	,172	,165
		N	48	48	48	48	48	48	48
	BCST-N categorias	Correlações de coeficiente	-,258	-,142	-,222	1,000	-,484**	,687**	,854**
		Sig. (2 extremidades)	,076	,336	,129	.	,000	,000	,000
		N	48	48	48	50	50	50	50
	BCST-N cartas	Correlações de coeficiente	,241	,186	,197	-,484**	1,000	-,025	-,466**
		Sig. (2 extremidades)	,098	,206	,179	,000	.	,861	,001
		N	48	48	48	50	50	50	50
	BCST-R corretas	Correlações de coeficiente	-,206	-,156	-,201	,687**	-,025	1,000	,845**
		Sig. (2 extremidades)	,161	,290	,172	,000	,861	.	,000
		N	48	48	48	50	50	50	50
	BCST-% R corretas	Correlações de coeficiente	-,304*	-,165	-,204	,854**	-,466**	,845**	1,000
		Sig. (2 extremidades)	,036	,262	,165	,000	,001	,000	.
		N	48	48	48	50	50	50	50

## Correlações

		BCST-erros	BCST-% erros	BCST-R perseverativas	BCST-% R perseverativas	BCST-E perseverativas	BCST-% E perseverativas	
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	,297	,304**	,198**	,176	,255	,219
		Sig. (2 extremidades)	,041	,036	,176	,232	,080	,134
		N	48	48	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,206**	,161	,055**	,029	,062	,104
		Sig. (2 extremidades)	,160	,274	,709	,843	,678	,481
		N	48	48	48	48	48	48

	N	48	48	48	48	48	48
SDQ total	Correlações de coeficiente	,261**	,199**	-,041	-,083	,021	,082
	Sig. (2 extremidades)	,073	,174	,784	,576	,889	,582
	N	48	48	48	48	48	48
BCST-N categorias	Correlações de coeficiente	-,804	-,854	,140	,173	-,062**	,063**
	Sig. (2 extremidades)	,000	,000	,331	,229	,667	,665
	N	50	50	50	50	50	50
BCST-N cartas	Correlações de coeficiente	,321	,466	,238	,151**	,269	-,019
	Sig. (2 extremidades)	,023	,001	,096	,297	,059	,898
	N	50	50	50	50	50	50
BCST-R corretas	Correlações de coeficiente	-,937	-,843	,046	,038**	-,204	-,316
	Sig. (2 extremidades)	,000	,000	,749	,791	,155	,025
	N	50	50	50	50	50	50
BCST-% R corretas	Correlações de coeficiente	-,958*	-1,000	-,117	-,085**	-,349**	-,233**
	Sig. (2 extremidades)	,000	,000	,419	,557	,013	,103

## Correlações

		BCST-E não perseverativos	BCST-% E não perseverativos	BCST- Erros unicos	BCST-% Erros unicos	BCST-Cartas 1 categoria	BCST-Falha manter contexto	
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	,118	,111**	,204**	,184	-,182	-,064
		Sig. (2 extremidades)	,425	,452	,164	,212	,217	,669
		N	48	48	48	48	48	47
	SDQ Hiperatividade	Correlações de coeficiente	,161**	,129	,188**	,190	-,216	-,009
		Sig. (2 extremidades)	,275	,383	,201	,197	,140	,951
		N	48	48	48	48	48	47
	SDQ total	Correlações de coeficiente	,261**	,217**	,205	,200	-,118	,113
		Sig. (2 extremidades)	,073	,138	,163	,174	,423	,451
		N	48	48	48	48	48	47
	BCST-N categorias	Correlações de coeficiente	-,752	-,793	-,584	-,545	-,115**	,149**
		Sig. (2 extremidades)	,000	,000	,000	,000	,426	,306
		N	50	50	50	50	50	49
	BCST-N cartas	Correlações de coeficiente	,251	,305	,208	,147**	,282	,154
		Sig. (2 extremidades)	,079	,031	,148	,309	,048	,291
	N	50	50	50	50	50	49	
BCST-R corretas	Correlações de coeficiente	-,732	-,720	-,647	-,648**	,124	,661	
	Sig. (2 extremidades)	,000	,000	,000	,000	,389	,000	

	N	50	50	50	50	50	49
BCST-% R corretas	Correlações de coeficiente	-,704*	-,739	-,624	-,586**	-,035**	,549**
	Sig. (2 extremidades)	,000	,000	,000	,000	,811	,000

**Correlações**

			BCST-Aprendendo a aprender	BCST-Nível conceitual
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	,338	-,230**
		Sig. (2 extremidades)	,084	,119
		N	27	47
	SDQ Hiperatividade	Correlações de coeficiente	,194**	-,163
		Sig. (2 extremidades)	,331	,274
		N	27	47
	SDQ total	Correlações de coeficiente	,004**	-,193**
		Sig. (2 extremidades)	,986	,194
		N	27	47
	BCST-N categorias	Correlações de coeficiente	,394	,825
		Sig. (2 extremidades)	,038	,000
		N	28	49
	BCST-N cartas	Correlações de coeficiente	-,349	-,279
		Sig. (2 extremidades)	,068	,052
		N	28	49
	BCST-R corretas	Correlações de coeficiente	-,090	,965
		Sig. (2 extremidades)	,650	,000
		N	28	49
BCST-% R corretas	Correlações de coeficiente	,195*	,957	
	Sig. (2 extremidades)	,321	,000	

**Correlações Donkey**

			SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	Donkey-Total maçãs	Donkey-Portas c	Donkey-Portas v	Donkey-Portas b
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	-,138	-,114	,256	-,191
		Sig. (2 extremidades)	.	,000	,000	,349	,440	,079	,194
		N	48	48	48	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,536**	1,000	,796**	,004	-,137	,104	-,058
		Sig. (2 extremidades)	,000	.	,000	,977	,355	,483	,695
		N	48	48	48	48	48	48	48
	SDQ total	Correlações de coeficiente	,671**	,796**	1,000	,050	-,074	,106	-,015

	Sig. (2 extremidades)	,000	,000	.	,735	,618	,474	,917
	N	48	48	48	48	48	48	48
Donkey-Total maçãs	Correlações de coeficiente	-,138	,004	,050	1,000	-,267	-,419**	,398**
	Sig. (2 extremidades)	,349	,977	,735	.	,061	,002	,004
	N	48	48	48	50	50	50	50
Donkey-Portas c	Correlações de coeficiente	-,114	-,137	-,074	-,267	1,000	-,442**	,231
	Sig. (2 extremidades)	,440	,355	,618	,061	.	,001	,107
	N	48	48	48	50	50	50	50
Donkey-Portas v	Correlações de coeficiente	,256	,104	,106	-,419**	-,442**	1,000	-,597**
	Sig. (2 extremidades)	,079	,483	,474	,002	,001	.	,000
	N	48	48	48	50	50	50	50
Donkey-Portas b	Correlações de coeficiente	-,191	-,058	-,015	,398**	,231	-,597**	1,000
	Sig. (2 extremidades)	,194	,695	,917	,004	,107	,000	.

## Correlações

			Donkey-Portas n	Donkey-Portas c+v	Donkey-Portas b+n	Donkey-Vantajosas - Desvantajosas
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	,012	,069**	-,067**	-,069
		Sig. (2 extremidades)	,934	,644	,649	,644
		N	48	48	48	48
	SDQ Hiperatividade	Correlações de coeficiente	,067**	-,039	,040**	,039
		Sig. (2 extremidades)	,651	,795	,786	,795
		N	48	48	48	48
	SDQ total	Correlações de coeficiente	,034**	-,054**	,056	,054
		Sig. (2 extremidades)	,819	,716	,706	,716
		N	48	48	48	48
	Donkey-Total maçãs	Correlações de coeficiente	,460	-,694	,694	,694
		Sig. (2 extremidades)	,001	,000	,000	,000
		N	50	50	50	50
	Donkey-Portas c	Correlações de coeficiente	-,477	,293	-,293	-,293
		Sig. (2 extremidades)	,000	,039	,039	,039
		N	50	50	50	50
	Donkey-Portas v	Correlações de coeficiente	-,274	,620	-,620	-,620**
Sig. (2 extremidades)		,054	,000	,000	,000	
N		50	50	50	50	
Donkey-Portas b	Correlações de coeficiente	-,250	-,405	,407	,405**	

Sig. (2 extremidades) | ,080 | ,004 | ,003 | ,004 |

**Correlações**

			SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	DspanD-Memory span	DspanD-Total corretos	DspanD-Tempo (min)	DspanI-Memory span
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	-,283	-,252	-,011	-,222
		Sig. (2 extremidades)	.	,000	,000	,052	,084	,944	,129
		N	48	48	48	48	48	47	48
	SDQ Hiperatividade	Correlações de coeficiente	,536**	1,000	,796**	-,261	-,157	-,108	-,115
		Sig. (2 extremidades)	,000	.	,000	,073	,286	,472	,435
		N	48	48	48	48	48	47	48
	SDQ total	Correlações de coeficiente	,671**	,796**	1,000	-,330*	-,318*	-,002	-,170
		Sig. (2 extremidades)	,000	,000	.	,022	,027	,989	,248
		N	48	48	48	48	48	47	48
	DspanD-Memory span	Correlações de coeficiente	-,283	-,261	-,330*	1,000	,927**	,646**	,515**
		Sig. (2 extremidades)	,052	,073	,022	.	,000	,000	,000
		N	48	48	48	50	50	49	50
	DspanD-Total corretos	Correlações de coeficiente	-,252	-,157	-,318*	,927**	1,000	,550**	,522**
		Sig. (2 extremidades)	,084	,286	,027	,000	.	,000	,000
		N	48	48	48	50	50	49	50
	DspanD-Tempo (min)	Correlações de coeficiente	-,011	-,108	-,002	,646**	,550**	1,000	,407**
		Sig. (2 extremidades)	,944	,472	,989	,000	,000	.	,004
		N	47	47	47	49	49	49	49
	DspanI-Memory span	Correlações de coeficiente	-,222	-,115	-,170	,515**	,522**	,407**	1,000
		Sig. (2 extremidades)	,129	,435	,248	,000	,000	,004	.
		N							

**Correlações**

			DspanI-Total corretos	DspanI-Tempo (min)
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	-,312	,036**
		Sig. (2 extremidades)	,031	,808
		N	48	48
	SDQ Hiperatividade	Correlações de coeficiente	-,139**	-,219
		Sig. (2 extremidades)	,347	,135
		N	48	48
	SDQ total	Correlações de coeficiente	-,172**	-,153**
		Sig. (2 extremidades)	,244	,299
		N	48	48

	Correlações de coeficiente	,547	,364
DspanD-Memory span	Sig. (2 extremidades)	,000	,009
	N	50	50
	Correlações de coeficiente	,530	,322
DspanD-Total corretos	Sig. (2 extremidades)	,000	,023
	N	50	50
	Correlações de coeficiente	,415	,429
DspanD-Tempo (min)	Sig. (2 extremidades)	,003	,002
	N	49	49
	Correlações de coeficiente	,952	,818
DspanI-Memory span	Sig. (2 extremidades)	,000	,000

### Correlações Stroop

		SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	StroopE- Total	StroopE- Incroguentes	StroopE- Aleatorios
Rô de Spearman	Correlações de coeficiente	1,000	,536**	,671**	,113	,003	,119
	Sig. (2 extremidades)	.	,000	,000	,459	,985	,435
	N	48	48	48	45	45	45
	Correlações de coeficiente	,536**	1,000	,796**	,233	,132	,205
	Sig. (2 extremidades)	,000	.	,000	,124	,389	,176
	N	48	48	48	45	45	45
	Correlações de coeficiente	,671**	,796**	1,000	,373*	,099	,328*
	Sig. (2 extremidades)	,000	,000	.	,012	,517	,028
	N	48	48	48	45	45	45
	Correlações de coeficiente	,113	,233	,373*	1,000	,532**	,959**
	Sig. (2 extremidades)	,459	,124	,012	.	,000	,000
	N	45	45	45	47	47	47
	Correlações de coeficiente	,003	,132	,099	,532**	1,000	,351*
	Sig. (2 extremidades)	,985	,389	,517	,000	.	,016
	N	45	45	45	47	47	47
	Correlações de coeficiente	,119	,205	,328*	,959**	,351*	1,000
	Sig. (2 extremidades)	,435	,176	,028	,000	,016	.
	N	45	45	45	47	47	47

### Correlações Torre de Londres

			SDQ Prob. Conduta	SDQ Hiperatividade	SDQ total	Pontuação Final	Total Tentativas
Rô de Spearman	SDQ Prob. Conduta	Correlações de coeficiente	1,000	,536**	,671**	-,287*	,277

	Sig. (2 extremidades)	.	,000	,000	,048	,056
	N	48	48	48	48	48
	Correlações de coeficiente	,536**	1,000	,796**	-,342*	,313*
SDQ Hiperatividade	Sig. (2 extremidades)	,000	.	,000	,017	,030
	N	48	48	48	48	48
	Correlações de coeficiente	,671**	,796**	1,000	-,343*	,314*
SDQ total	Sig. (2 extremidades)	,000	,000	.	,017	,029
	N	48	48	48	48	48
	Correlações de coeficiente	-,287*	-,342*	-,343*	1,000	-,983**
Pontuação Final	Sig. (2 extremidades)	,048	,017	,017	.	,000
	N	48	48	48	50	50
	Correlações de coeficiente	,277	,313*	,314*	-,983**	1,000
Total Tentativas	Sig. (2 extremidades)	,056	,030	,029	,000	.
	N	48	48	48	50	50