

**UNIVERSIDADE FEDERAL
DE MINAS GERAIS**



UFMG

**COMPARAÇÃO ENTRE AS VERSÕES MANUAIS E
COMPUTADORIZADAS DOS TESTES TORRE DE
LONDRES (ToL), TESTE DE STROOP (STROOP) E
TESTE WISCONSIN DE CLASSIFICAÇÃO DE
CARTAS (WCST)**

**Belo Horizonte
2011**

SUZAN CAROLINE SUCH RIBEIRO

**COMPARAÇÃO ENTRE AS VERSÕES MANUAIS E
COMPUTADORIZADAS DOS TESTES TORRE DE
LONDRES (ToL), TESTE DE STROOP (STROOP) E
TESTE WISCONSIN DE CLASSIFICAÇÃO DE
CARTAS (WCST)**

Monografia apresentada ao Programa de Especialização em Neurociência e Comportamento, com ênfase em Neuropsicologia, do Instituto de Ciências Biológicas da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de especialista.

Orientação: Prof. Dr. Arthur Melo e Kummer

**Belo Horizonte
2011**

*Esta pesquisa é dedicada a quem
pacientemente me apoiou, ensinou,
incentivou e, acima de tudo, está ao meu
lado a cada momento: meu marido
Fernando.*

AGRADECIMENTOS

Inicialmente, gostaria de agradecer a Deus por mais essa oportunidade de aprendizagem, aprimoramento e crescimento.

Para que esta pesquisa obtivesse o êxito de sua conclusão, contou com participações essenciais no seu desenvolvimento.

A escolha desta especialização não foi fácil, assim como percorrê-la, mas foi uma decisão sensata e de grande contribuição para minha formação profissional e pessoal. Agradeço a todos os funcionários da secretaria de neurociências, desde 2009 até então, que foram sempre solícitos e auxiliaram naquilo que lhes era possível.

Aos coordenadores do curso, pelo intermédio dos quais tive a oportunidade de cursar excelentes módulos, conhecendo profissionais competentes e dispostos a partilhar seu conhecimento.

Dentre estes, destaco meu orientador, Prof. Dr. Arthur Melo e Kummer, que gentilmente aceitou essa difícil função de orientar uma “desorientada aluna”. Obrigada pelas inúmeras conversas, emails, sugestões, sutis puxões de orelhas, ensinamentos e, acima de tudo, acreditar em mim. Agradeço ainda por agregar a nossa pesquisa a Estefânia, que em muito contribuiu com ricas sugestões, como a “aplicadora manual” e ouvindo minhas lamúrias de “mais um desistiu, faltou consentimento, fulano ficou doente”. Esta pesquisa é nossa.

Ao Profº Antonio Lúcio Teixeira e a Guilherme Nogueira, por gentilmente aceitarem compor a banca examinadora, disponibilizando tempo e atenção para este trabalho.

A cada aula, a cada fim de semana sem dormir, a cada seminário elaborado a base de muita coca-cola e pão de queijo (devidamente “feitos” pelo Fernando), a cada grupo de amigos que frequentaram o Q.G., a cada texto enigmático decifrado, aos lanchinhos da “Mãeção”, ao bandeirão, as inúmeras caronas, as histórias (e quantas histórias...), as charges, ao curso de inglês, ao Google, ao campus da UFMG, que tornaram cada momento único e especial.

A minha turma de especialização. Obrigada por me apresentarem um dos lados mais belos dos mineiros (e capixabas, risos): a amizade.

Agradeço ao Professor Leandro Malloy-Diniz que apesar de sua correria, arrumou tempinho para contribuir com idéias e sugestões.

Ao Alexandre Ferreira Campos, desculpas por retirar os títulos, mas após esse um ano de contato, trabalho e amizade, tomei essa liberdade. Obrigada por auxiliar no contato com a Escola Estadual Afonso Pena e por conviver contigo no ACRIAR. Tem sido muito rico para mim.

Obrigada aos alunos, pais e equipe da Escola Estadual Afonso Pena pela participação nesta pesquisa.

A Mãeção ou Conceição, que tem sido a melhor “chefa” do mundo, oportunizando aprendizado, amizade, parceria e preocupação de mãe mesmo. Obrigada pela confiança e por proporcionar minha convivência com a Carla. Vocês são demais.

As meninas do grupo de estudos de neurociências, inspirado na especialização e que propiciou uma continuidade neste aprendizado. A amizade de vocês faz parte das conquistas da pós.

Aos colegas do Núcleo Artístico Floresta que fazem da dança a melhor atividade física, agregando a amizade como fator essencial e local de “desstress”.

As minhas amigas que de alguma forma fizeram parte desse processo, seja no incentivo (Vivi, Karen, Iris, Fer), na cooperação direta (Pri), nas conversas rotineiras (Ale, madrinhas) e em pensamento. Vocês estarão sempre perto de mim, mesmo que a física diga não.

A meus pais, que investiram e acreditaram em meu potencial.

E, por fim, ao Fernando. Obrigada por fazer da parte minha vida e pagar a pós (risos).

RESUMO

O presente estudo teve como objetivo comparar as aplicações manual e computadorizada do Teste de Stroop (Stroop), Torre de Londres (ToL) e Teste Wisconsin de Classificação de Cartas (WCST). Tais testes são utilizados na avaliação das funções executivas, consideradas essenciais para gerenciar o comportamento humano e, por isso, pertencem à avaliação neuropsicológica de indivíduos normais ou com alterações cognitivas ou neurocomportamentais de causas diversas. A aplicação computadorizada destes testes tem como benefícios a precisão na apresentação dos estímulos, feedback, medidas temporais (reação e duração das respostas) e registro preciso das respostas, eliminando assim possíveis erros na obtenção dos dados e pontuação. A aplicação pelo examinador permite atividades como encorajamento do sujeito, esclarecimento das regras, auxiliá-lo a focar na tarefa, além de obter dados qualitativos durante o teste que não são possíveis observar apenas com a pontuação obtida. Os dados obtidos demonstraram similaridades significativas entre as versões manual e computadorizada do WCST-64 considerando os índices número de acertos, categorias completadas e número e percentual de erros, indicando possível validade convergente entre as versões deste teste, tal como foi mencionado em estudos similares, porém com diferenças na metodologia. O mesmo não pode ser observado no teste ToL, onde comparou-se o número de movimentos e pontuação das versões. No Stroop, apenas o índice tempo II apresentou similaridade, tendo divergências em todos os demais (tempo e erro I, erro II, tempo e erro III). Observou-se, ainda, que as versões computadorizadas dos testes WCST-64 e o Stroop foram mais motivadores para a faixa etária avaliada (8 a 10 anos) e que crianças com maior conhecimento em informática, em geral, obtiveram melhor desempenho nas versões computadorizadas. Ainda, quanto mais estimuladas as crianças eram (uso de computador, internet), melhor seu desempenho no Raven. Tais dados demonstram a relevância de pesquisas envolvendo testes computadorizados para o aprimoramento na avaliação das funções executivas, auxiliando no diagnóstico e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos. A presente pesquisa de validade convergente desses instrumentos foi um primeiro passo para a realização de estudos futuros que objetivem normatizar as versões computadorizadas na população brasileira.

Palavras chaves: Teste de Stroop, Teste Wisconsin de Classificação de Cartas, Torre de Londres, testes computadorizados, validação.

ABSTRACT

This study aimed to verify the functional analogy between manual and computer applications in the Stroop Test (Stroop), Tower of London (ToL) and Wisconsin Card Sorting Test (WCST). These tests are used in the assessment of executive functions, considered essential for managing human behavior and therefore belong to the neuropsychological evaluation of normal subjects or with cognitive or neurobehavioral alterations several causes. The implementation of computerized testing has benefits such as accuracy in the presentation of stimuli, feedback, temporal measures (reaction and duration of responses) and accurate record of the responses, thus eliminating possible errors in data collection and scoring. The application by the examiner allows activities such as encouragement of the subject, clarification of the rules, help you focus on the task, and to obtain qualitative data during the test that are not possible to observe the score only. The data showed significant similarities between manual and computerized versions of the WCST-64, considering the number of correct, categories completed and number and percentage of errors indexes, indicating a possible convergent validity between the versions of this test, as was mentioned in similar studies, but with differences in methodology. The same cannot be seen in the ToL test, which compared the number of moves and scores of versions. In the Stroop, only the time II index showed similarity, and differences in all other (time and error I, error II, time and error III). It was noted further that the computerized versions of the WCST-64 and Stroop tests were more motivating for the age group assessed (8-10 years) and that children with greater computer knowledge in general, performed better on computerized versions. Still, the more stimulated the kids were (use of computer, internet), better was their performance in Raven. These data demonstrate the relevance of research involving computerized tests to improve the assessment of executive functions, aiding in the diagnosis and monitoring of several deficits of behavioral and psychiatric disorders. This study of convergent validity of these instruments was a first step towards future studies that aim to standardize computerized versions in Brazilian population.

Keywords: Stroop test, Wisconsin Cards Sorting Test, Tower of London, computerized tests, validation.

LISTA DE ILUSTRAÇÕES

FIGURA 1- Tela inicial do programa PEBL	25
FIGURA 2- Cartões da versão manual do Stroop	26
FIGURA 3- Teclado adaptado.....	26
FIGURA 4- Tela de treino	27
FIGURA 5- Tela da primeira parte do Stroop computadorizado	27
FIGURA 6- Tela da parte II do Stroop computadorizado	28
FIGURA 7- Tela da parte III do teste Stroop computadorizado	28
FIGURA 8- Torre de Londres Manual	29
FIGURA 9- Cartões problemas da Torre de Londres manual	30
FIGURA 10- Instrução ToL computadorizada	30
FIGURA 11- Tela inicial do problema de demonstração ToL	31
FIGURA 12- Tela da ToL computadorizada	31
FIGURA 13- Tela de problema da ToL, mostrando a barra de movimentos	32
FIGURA 14- Cartas-chave Wisconsin manual	32
FIGURA 15- Tela da instrução do Wisconsin computadorizado	33
FIGURA 16- Tela do Wisconsin computadorizado	34
GRÁFICO 1- Horas de uso de computador por semana	39
GRÁFICO 2- Porcentagem de participantes que já utilizaram internet	40
GRÁFICO 3- Porcentagem de horas gastas jogando vídeo-game por semana	40
GRÁFICO 4- Tempo gasto nas aplicações dos testes	41
GRÁFICO 5- Percentual da preferência entre as versões dos testes	43

LISTA DE TABELAS

1- Sexo	38
2- Dados sobre utilização do computador	39
3- Dados em relação ao uso da internet	40
4- Dados da aplicação dos testes	41
5- Desempenho no teste Raven	41
6- Resultados gerais dos testes	42
7- Grau de dificuldade das versões computadorizada e manual	43
8- Análise de correlações entre variáveis do teste Torre de Londres	43
9- Correlações entre as versões manual e computadorizada do Wisconsin Cards Sorting Test (WCST-64)	44
10- Correlações entre as versões manual e computadorizada do Stroop	44
11- Correlações entre o Teste Raven e Dados do Questionário	45
12- Comparações do Teste Raven com ToL, WCST-64 e Stroop	46
13- Comparações entre os dados dos questionários	47
14- Comparações dos dados dos questionários e ToL	48
15- Comparações dos dados dos questionários e WCST-64	49
16- Comparações dos dados dos questionários e Stroop	50
17- Comparações relacionadas ao gênero	51
18- Modelo de tabela de preferência e grau de dificuldade (ANEXO H)	116
19- Correlações com o Teste Raven (ANEXO I)	117
20- Correlações entre os dados dos questionários (ANEXO J)	120
21- Comparações dos dados dos questionários e ToL, WCST-64 e Stroop (ANEXO L)	121

LISTA DE ABREVIATURAS E SIGLAS

Computadorizada (o)- versão computadorizada do teste

EUA- Estados Unidos da América

Manual- versão manual do teste

MG – Minas Gerais

PEBL- The Psychology Experiment Building Language

Raven- Teste das Matrizes Progressivas de Raven

SPSS- Statistical Package for the Social Sciences

Stroop – Teste de Stroop

TDAH- Transtorno de déficit de atenção e hiperatividade

ToL- Teste de Torre de Londres

WCST- Teste Wisconsin de Classificação de Cartas

WCST- 64- Teste Wisconsin de Classificação de Cartas versão 64 cartas

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Fundamentação Teórica	13
1.2 Justificativa	19
1.3 Hipóteses	19
2 OBJETIVOS	21
2.1 Objetivo primário	21
2.2 Objetivos secundário	21
3 MÉTODOS	23
3.1 Delineamento	23
3.2 Participantes	23
3.3 Materiais	24
3.4 Procedimentos	35
3.5 Aspectos éticos	35
3.6 Análise Estatística	36
4 RESULTADOS	38
4.1 Características dos participantes	38
4.2 Resultados gerais dos testes	41
4.3 Correlações entre as versões	43
4.4 Correlações com Teste Raven e os dados dos questionários	45
4.5 Comparações entre gêneros	51
5 DISCUSSÃO	53
6 CONCLUSÃO	60
REFERÊNCIAS	62

APÊNDICE 66

ANEXOS 68

Introdução

1 INTRODUÇÃO

1.1 Fundamentação teórica

As funções executivas são consideradas essenciais para gerenciar o comportamento humano e, por isso, pertencem à avaliação neuropsicológica de indivíduos normais ou com alterações cognitivas ou comportamentais de causas diversas (SOUZA *et al.*, 2001). Tal gerenciamento ocorre através de habilidades que, de forma integrada, permitem o indivíduo direcionar os comportamentos a objetivos, analisar a adequação e eficácia deste comportamento, modificar estratégias quando necessário e, desta forma, solucionar problemas diversos.

Neste sentido, quando se menciona o termo função executiva, está se referindo a um conjunto complexo de processos que tem sido definido de diversas maneiras. Para Lezak (2004), as funções executivas contêm quatro componentes principais: volição, planejamento, ação proposital e o desempenho efetivo. A volição refere-se ao comportamento intencional do indivíduo direcionado à realização de metas. O planejamento está relacionado à identificação de etapas e elementos necessários para a concretização de determinado objetivo. A transição da intenção e do planejamento para a ação em si é a denominada ação proposital. À medida que executa essas ações, o indivíduo deve monitorar o desempenho efetivo de seus comportamentos (MALLOY-DINIZ, 2008a).

Já Goldberg (MALLOY-DINIZ, 2008a) compara a atuação das funções executivas a um maestro que coordena as demais estruturas neurais. Ainda, enfatiza que são resultado da atividade dos lobos frontais, mais especificamente da região pré-frontal.

Malloy-Diniz (2008a) resume as diversas definições, considerando-as complementares e, desta forma, considera as funções executivas como gerenciadoras do comportamento humano, utilizando, para tanto, processos como planejamento, controle inibitório, tomada de decisões, flexibilidade cognitiva, memória operacional e atenção. 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, 2008a). 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).

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 (MALLOY-DINIZ, 2008a).

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 (*ibidem*).

A memória operacional, segundo Malloy-Diniz

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 (2008a, p.199).

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, 2008a).

Para avaliação das funções executivas há uma ampla variedade de testes. Alguns objetivam avaliar de maneira global (bateria de avaliação de síndrome disexecutiva) e outros se atentam a aspectos específicos como controle inibitório (Teste de Stroop), planejamento (Torre de Londres) e flexibilidade (Teste Wisconsin de Classificação de Cartas) (MALLOY-DINIZ *et al.*, 2008a).

O Teste de Stroop foi desenvolvido por John Ridley Stroop em 1935 e é utilizado para avaliar controle inibitório (impulsividade), atenção seletiva e flexibilidade (DUNCAN,

2006). O desempenho comprometido no teste tem sido associado por alguns autores ao transtorno de déficit de atenção e hiperatividade (TDAH) (CAPOVILLA *et al.* 2007), transtorno obsessivo-compulsivo (FONTENELLE, 2001), dentre outros.

O teste baseia-se no efeito Stroop, o qual pressupõe que as pessoas possuem dificuldades em processar informações simultâneas com significado conflitante, mesmo quando uma delas não tem relevância na tarefa. Inúmeras versões têm sido utilizadas como a versão Victoria, uma das mais citadas na literatura (STROOP, 1935). Esta versão tem apresentado inúmeras vantagens. Comparada com outras versões, é considerada breve, possuindo 24 itens em cada etapa. Estudos demonstram que versões mais curtas favorecem a identificação de dificuldades nesta tarefa (KLEIN *et al.*, 1997 *apud* STRAUSS, 2006). Além disso, a versão Victoria é de domínio público, ou seja, cada pesquisador pode elaborar os seus estímulos de acordo com as normas previstas (STRAUSS, 2006).

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). Um maior tempo nessa tarefa sugere um comprometimento das regiões pré-frontais dorsolateral e cíngulo, estruturas que possivelmente estão envolvidas em monitoração e resolução de conflito (LEUNG *et al.*, 2000). No Brasil foi publicado um estudo sobre a normatização para adolescentes entre 12 e 14 anos do Teste de Stroop aplicação manual baseado na versão Victoria, porém utilizando as cores azul, verde, rosa e marrom (DUNCAN, 2006).

Existem diversas versões computadorizadas do Teste de Stroop, que podem corresponder ou não as linhas gerais das principais versões dos testes manuais. Em muitas versões computadorizados, os retângulos e as palavras costumam surgir um por um, enquanto o indivíduo que está sendo testado pressiona determinada tecla do computador que

corresponde à cor que está sendo mostrada (ex.: tecla 1 para vermelho, tecla 2 para verde, tecla 3 para azul e tecla 4 para amarelo). Entretanto, o software pode ser programado de tal forma que em uma primeira parte do teste apareça uma sequência de 24 retângulos coloridos, numa segunda fase uma sequência de 24 palavras coloridas e por fim uma sequência de 24 nomes de cores impressas em cores não correspondentes. Programado dessa forma, o teste corresponderia, em teoria, à versão Victoria do Teste de Stroop.

O teste da Torre de Londres (ToL) 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). No Brasil, existem estudos publicados sobre o desempenho de adultos e crianças.

Há algumas formas de aplicação envolvendo distintas versões (número de questões, manual, computadorizada) e formas de registro (marcação do tempo ou não). A versão manual proposta por Krikorian e colaboradores utiliza uma base (placa lisa) com três pinos de tamanhos diferentes e equidistantes e três esferas de cores distintas (azul, verde e vermelho) (*ibidem*). Na instrução inicial, o sujeito é informado que deverá mover as esferas de forma a montar a mesma posição dos cartões-problema apresentados. Além disso, deverá executar na menor quantidade de movimentos possível e só poderá mover uma esfera de cada vez encaixando-a em um dos pinos. São no total 12 cartões-problema apresentados em ordem crescente de dificuldade. O sujeito tem três tentativas para cada cartão, alterando sua pontuação de acordo com a quantidade de tentativas gastas para resolução do problema. Se o sujeito não conseguir resolver o problema com o número mínimo de movimentos, o problema é reiniciado. Quanto menos tentativas, maior a pontuação. O máximo de pontos possível será 36. O examinador deverá registrar cada movimento emitido pelo sujeito, ou seja, cada vez que tirar a bola de um pino e colocar em outro pino. Algumas pesquisas têm sugerido um possível “efeito teto” desta versão da ToL, mas há a necessidade de maiores estudos.

As versões computadorizadas da ToL também podem ser configuradas de diversas formas, podendo-se escolher o número de esferas (3 a 6), o tamanho dos pinos, o número de cartões-problema e a forma de apresentação dos cartões-problema (randomizada ou em dificuldade progressiva). Pode-se ainda escolher que o teste siga os 12 cartões-problema clássicos de Shallice ou o de outros autores, como os cartões-problema de Phillips (1999) ou de Fimbel *et al* (2009) (MUELLER, 2009). 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 Wisconsin de Classificação de Cartas (WCST-64- Wisconsin Card Sorting Test) é um instrumento elaborado em 1948, utilizado para avaliar funções cognitivas superiores que estão relacionadas com os lobos frontais (HEATON *et al.*, 2005). 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). Sendo assim, quando o sujeito obtém um resultado baixo significa que pode haver uma disfunção executiva (SHAN *et al.*, 2008).

Pesquisas têm mostrado um baixo desempenho em sujeitos que apresentam diagnósticos como esquizofrenia, transtorno de déficit de atenção e hiperatividade (TDAH), epilepsia, autismo e depressão quando comparados ao resultado de grupos controles. O instrumento tem sua normatização norte-americana para a população de 6 (seis) anos e 6 (seis) meses a 89 anos e 11 meses (TRENTINI *et al.*, 2006). No entanto, alguns estudos demonstram a influência do fator cultural nos resultados do WCST e para que não haja indicação falso-positiva de déficit da função executiva faz-se necessário o estabelecimento de normas de acordo com a população alvo (SHAN *et al.*, 2008). No Brasil, as normas padronizadas para a aplicação manual do teste abrangem a faixa-etária dos 6 (seis) anos e 6 (seis) meses aos 17 anos e 11 meses (crianças e adolescentes) (TRENTINI *et al.*, 2006) e a versão para idosos de 60 a 89 anos (TRENTINI *et al.*, 2010).

A aplicação manual do teste é realizada pela apresentação de 4 (quatro) cartas-chave, compostas por um triângulo vermelho, duas estrelas verdes, três cruzeiras amarelas e quatro círculos azuis, e 128 cartas-resposta (dois *decks*¹ de 64 cartas). O sujeito tem como tarefas estabelecer categorias através de associações entre as cartas-resposta e as cartas-chave. As categorias são pré-determinadas e de conhecimento apenas do examinador que fornece feedback² de certo ou errado a cada resposta emitida pelo sujeito. Além disso, o examinador necessita registrar cada resposta emitida pelo sujeito para análises futuras. O teste se encerra quando o sujeito atinge as 6 (seis) categorias pré-determinadas ou terminam as cartas-resposta. Versões breves do teste têm sido propostas recentemente, com o intuito de diminuir o tempo de administração do instrumento, mas sem prejuízo da qualidade da avaliação. Entre

¹ Deck: baralho de cartas.

² 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.

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 aplicação computadorizada permite ajuste do software para a versão desejada, podendo-se escolher diferentes momentos para se encerrar o teste (ex.: após se atingir 6 categorias ou após se completar 32, 64 ou 128 cartas). Estudos demonstraram haver correlação entre as versões manual e computadorizada do teste em adultos (SHAN *et al.*, 2008; TIEN *et al.*, 1996; WAGNER; TRENTINI, 2009).

A motivação do sujeito durante a aplicação dos testes é essencial para que seu desempenho sofra poucas interferências. De acordo com Cunha (2004), o ambiente de testagem deve ser manejado de forma não só a diminuir a ansiedade natural do sujeito, mas levá-lo a uma atitude de cooperação. Neste sentido, acredita-se que os testes computadorizados possuam elementos favorecedores quanto à motivação. Os computadores estão cada vez mais presentes no cotidiano, principalmente das crianças e adolescentes, tendo em vista que nasceram junto com a popularização deste recurso (SANTOS, 2010). De acordo com Barbosa (2010), as faixas-etárias de 10 a 24 anos de idade apresentam percentuais muito mais elevados de domínio de habilidades no uso de computadores e da internet do que as demais. Assim, ter um elemento comum ao cotidiano pode facilitar a aplicação do teste.

A aplicação da versão computadorizada dos testes tem como benefícios a precisão na apresentação dos estímulos, feedback, medidas temporais (reação e duração das respostas) e registro preciso das respostas, eliminando assim possíveis erros na obtenção dos dados e pontuação. Essas características fornecem maior rigor na aplicação dos testes gerando uma avaliação neuropsicológica mais confiável (CAPOVILLA, 2006). Já a aplicação pelo examinador permite atividades como encorajamento do sujeito, esclarecimento das regras, auxiliá-lo a focar na tarefa, além de obter dados qualitativos durante o teste que não são possíveis observar apenas com a pontuação obtida. Ressalta-se ainda que o examinador deve ser uma pessoa hábil, atenta e muito prática para obter eficientemente tais dados sem interferir na coleta de dados. Entretanto, os poucos estudos que comparam testes manuais e computadorizados observaram que os testes computadorizados, apesar de gerarem dados mais confiáveis, não são substitutos à aplicação manual. Ao invés disso, o computador é um parceiro na aplicação do teste, que permite o examinador ficar livre para dar o suporte adequado ao sujeito em teste, encorajando-o e observando os aspectos qualitativos de seu desempenho (TIEN *et al.*, 1996).

Dessa forma, o presente estudo teve por objetivo comparar as aplicações manual e computadorizada do testes de Stroop, da ToL e do WCST na população brasileira.

1.2 Justificativa

Os testes Teste de Stroop (Stroop), Torre de Londres (ToL) e Teste Wisconsin de Classificação de Cartas (WCST) têm relevante importância na avaliação das funções executivas, auxiliando no diagnóstico e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos. Acreditamos que o estudo da validade convergente desses instrumentos é o primeiro passo para a realização de estudos futuros que objetivem normatizar as versões computadorizadas na população brasileira. Além disso, o projeto é de extrema importância 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 Hipóteses

- Há validade convergente entre as aplicações manual e computadorizada do Teste de Stroop (Stroop), Torre de Londres (ToL) e Teste Wisconsin de Classificação de Cartas (WCST).
- O conhecimento em informática favorece o desempenho nos testes computadorizados.
- Há maior motivação na realização dos testes computadorizados do que nos testes de aplicação manual.

Objetivos

2 OBJETIVOS

2.1 Objetivo Primário

O objetivo deste estudo foi comparar as versões manual e computadorizada dos testes Teste de Stroop, Torre de Londres e Teste Wisconsin de Classificação de Cartas.

2.2 Objetivos Secundário

- Avaliar se o conhecimento em informática do sujeito interfere no desempenho da versão computadorizada dos testes;
- Investigar se a presença do computador influencia a motivação do sujeito;
- Examinar a influência de idade, sexo e nível de escolaridade no desempenho dos participantes nas versões manual e computadorizada dos instrumentos.

Métodos

3 MÉTODOS

3.1 Delineamento

Trata-se de pesquisa comparativa realizada na Escola Estadual Afonso Pena em Belo Horizonte- MG com os participantes que preencheram os critérios de inclusão do estudo. Neste local, foram explicados todos os procedimentos da pesquisa, esclarecidas eventuais dúvidas e obtido o consentimento dos participantes. A instituição localiza-se na região central de Belo Horizonte e recebe alunos de diversas regiões da cidade, proporcionando heterogeneidade da amostra. Os participantes foram examinados na instituição envolvida na pesquisa, o que facilitou sua participação.

Foram examinados 22 indivíduos, número suficiente para o cálculo de correlação dos instrumentos.

A pesquisa não dispôs de financiamento e não acarretou gastos significativos.

3.2 Participantes

Para este estudo, foram sorteadas a partir dos diários de classe 22 crianças de oito a dez anos regularmente matriculadas na Escola Estadual Afonso Pena, cursando o 4º ano do Ensino Fundamental I, no período entre Março a Abril de 2011. Foram incluídos participantes de ambos os sexos, independente 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 tenham usado substâncias psicoativas (álcool e drogas ilícitas) recentemente. Nenhuma criança necessitou ser excluída.

3.3 Materiais

Os participantes foram avaliados em uma sala apropriada à aplicação dos testes. Foram administrados os testes manuais do Teste de Stroop versão Victoria, da Torre de Londres (versão de Shallice-Krikorian) e do Wisconsin Cards Sorting Test-64 (WCST-64), bem como as versões computadorizadas desses testes em versões similares às versões manuais. Juntamente com a aplicação manual dos testes, foi aplicado o Teste das Matrizes Progressivas de Raven (Raven) para pareamento dos participantes.

O software utilizado na elaboração dos testes computadorizados é o PEBL- The Psychology Experiment Building Language (versão 0.11 e bateria de teste 0.6), que possui código aberto e é um software livre (FIG. 1), sendo possível obtê-lo no site <http://pebl.sourceforge.net/>. O programa foi originalmente criado por Shane T. Mueller e permite algumas adaptações no sistema de acordo com a necessidade do pesquisador.

O PEBL é escrito primariamente em linguagem C++, mas utiliza outras ferramentas (flex, yacc) e bibliotecas (SDL, imagem SDL, SDL gfx e SDL ttf) para ser usada. Pode ser rodado nos principais sistemas operacionais comumente utilizados: Linux, Mac OS X e Microsoft Windows. Na pesquisa foi utilizado um notebook da marca Acer, com monitor LCD 15'', processador Intel Pentium dual core, 1.60 GHz, memória RAM 1.00 GB, HD 160 GB e sistema operacional Microsoft Windows 7. Para realizar as modificações no programa, utilizaram-se os programas 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 um programa, por exemplo, orientação de como inserir áudio (MUELLER, 2010).

Utilizando este programa, os pesquisadores e o próprio Shane T. Mueller adaptaram as versões computadorizadas de maneira similar às versões manuais utilizadas. Os testes iniciam-se com o registro do participante (nome ou iniciais), seguido pelas instruções dadas tanto de forma auditiva quanto visual. As versões foram elaboradas almejando que o avaliador interfira apenas para iniciar e encerrar cada teste. A seguir, segue descrição de cada teste.

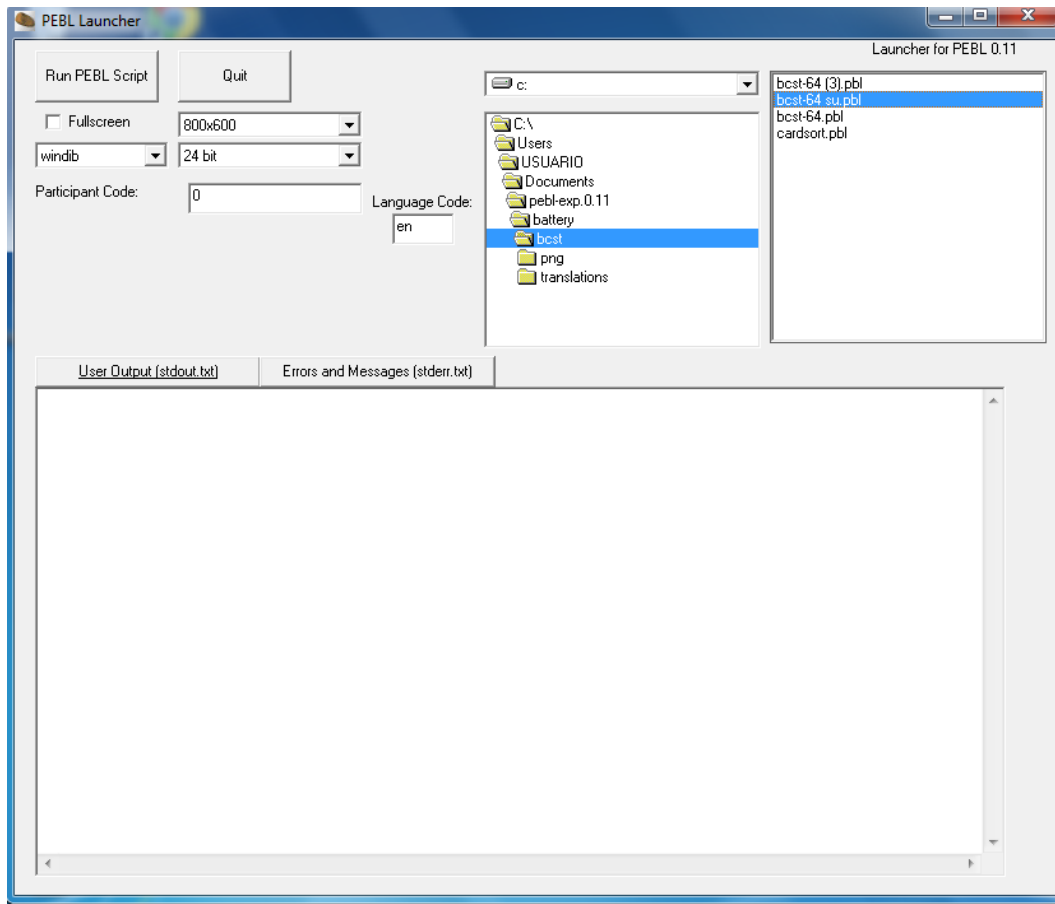


FIGURA 1- Tela inicial do programa PEBL onde o teste aplicado é selecionado.

- **Teste de Stroop versão Victoria**

A versão manual deste teste aplicado na pesquisa é composta por 3 cartões, tamanho 12 cm x 20 cm (altura x largura), todos com 24 estímulos, nas cores verde, azul, vermelho ou amarelo. Os estímulos ficam dispostos em 4 colunas, com intervalo de 1,5 cm entre elas, e 6 linhas, intervalo de 1 cm. O primeiro cartão, contém retângulos como estímulos, os quais o sujeito deve nomear o mais rápido que puder a cor na qual está impresso. O segundo possui como estímulos as palavras “cada, nada, nunca, tudo” e o sujeito deve nomear as cores na qual estas palavras estão impressas. O último é composto por nomes das cores “amarelo, verde, vermelho, azul” impressos em cores distintas (ex: “amarelo”, impresso em azul) e, da mesma forma que na segunda parte, a atividade é nomear a cor na qual a palavra está impressa. Um modelo dos cartões encontra-se na FIG. 2.



FIGURA 2- Cartões da versão manual do Stroop.

A versão computadorizada foi aplicada utilizando um teclado adaptado, no qual estavam disponíveis apenas as teclas 1, 2, 3, 4 e a barra de espaço (FIG. 3). A adaptação foi feita encaixando-se uma “capa” de papel cartão ao teclado, deixando exposto apenas as teclas mencionadas acima (DUNCAN, 2006; STROOP, 1935).

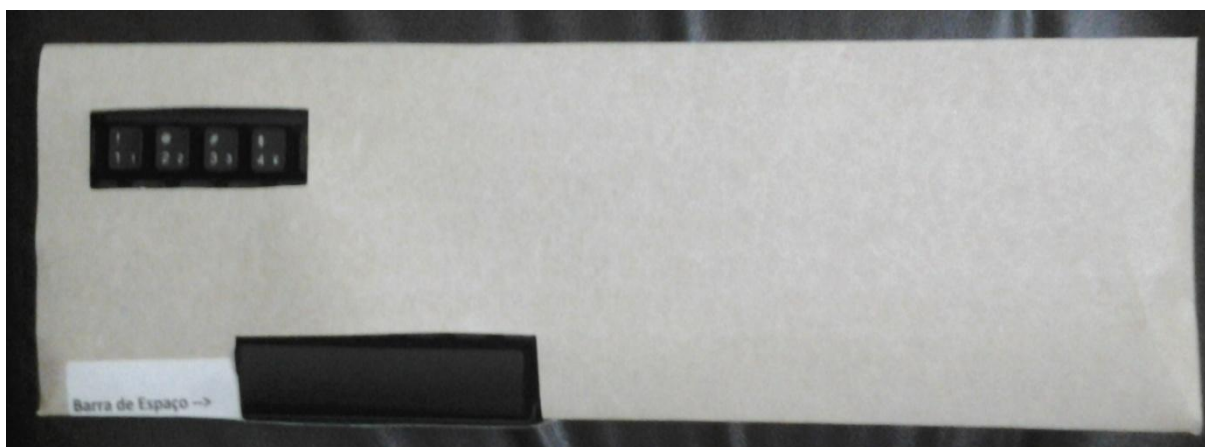


FIGURA 3- Teclado adaptado para utilização no Teste de Stroop manual.

O teste computadorizado inicia com o registro do nome/sigla do sujeito. Em seguida, o programa fornece instruções quanto ao teste de forma visual e auditiva. Antes do teste propriamente dito, há uma etapa na qual o sujeito pode se familiarizar com as teclas que

são utilizadas para a execução da atividade (FIG. 4). A tela esclarece como será a “nomeação de cores” apresentando uma barra para que o sujeito pressione as teclas e observe as cores respectivas. Durante todo o teste, as palavras correspondentes a cada tecla (1, 2, 3 e 4) ficam expostas na parte inferior da tela, como pode ser observado na FIG. 4.

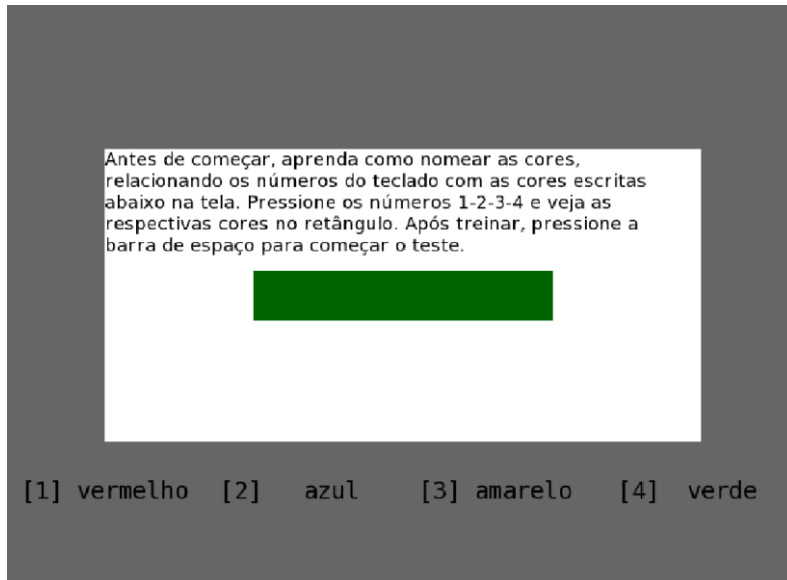


FIGURA 4- Tela de treino.

O teste é composto por três etapas, tal como na versão manual, com 24 estímulos, nas cores verde, azul, vermelho ou amarelo, dispostos em 4 colunas e 6 linhas. Na primeira etapa, após a instrução, aparece na tela 24 círculos e o sujeito deve pressionar a tecla correspondente a cor destacada por um retângulo (FIG. 5).



FIGURA 5- Tela da primeira parte do Stroop computadorizado.

Na segunda parte, após a instrução, a tarefa é pressionar a tecla correspondente à cor da palavra destacada pelo retângulo. Nesta etapa aparecem as palavras: cada, nunca, tudo, nada, conforme observado na FIG. 6.



FIGURA 6- Tela da parte II do Stroop computadorizado.

Na última parte, após a instrução, a tarefa é pressionar a tecla correspondente à cor da palavra destacada pelo retângulo, tal como na parte II. Contudo, aparecem as palavras estímulos que são nome das cores: amarelo, azul, verde, vermelho, conforme observado na FIG. 7. A ordem das cores relacionadas aos números 1, 2, 3 e 4 ocorre de forma aleatória, assim, como a distribuição das cores dos estímulos. O desempenho é registrado em milissegundos e fornecido em planilha do Excel. No Anexo A, encontra-se descrito a programação deste teste.



FIGURA 7- Tela da parte III do teste Stroop computadorizado.

- **Teste da Torre de Londres**

A versão manual utilizada na pesquisa possui uma base (placa lisa), tamanho 35 cm x 10 cm (comprimento x largura), com três pinos de tamanhos diferentes (15 cm, 10 cm e 5 cm), equidistantes, e três esferas (5 cm de altura) de cores distintas (azul, verde e vermelho), conforme a FIG. 8.

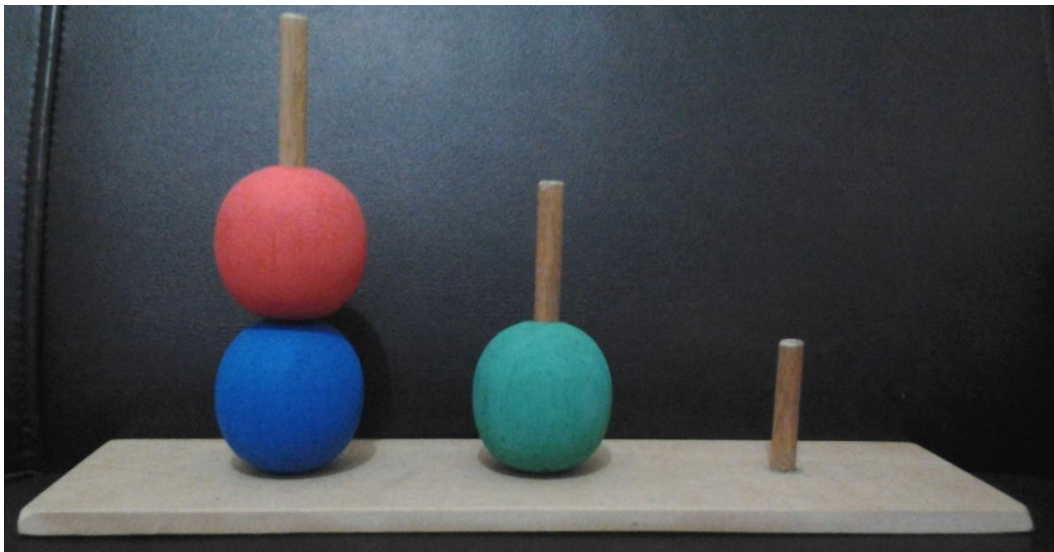


FIGURA 8- Torre de Londres Manual

Na instrução inicial, o sujeito é informado que deverá mover as esferas de forma a montar a mesma posição dos cartões-problema apresentados (FIG. 9). Além disso, deverá executar na menor quantidade de movimentos possível e só poderá mover uma esfera de cada vez encaixando-a em um dos pinos. São no total 12 cartões-problema apresentados em ordem crescente de dificuldade. O sujeito tem três tentativas para cada cartão, alterando sua pontuação de acordo com a quantidade de tentativas gastas para resolução do problema. Se o sujeito não conseguir resolver o problema com o número mínimo de movimentos, o problema é reiniciado. Quanto menos tentativas, maior a pontuação (KRIKORIAN, 1994).

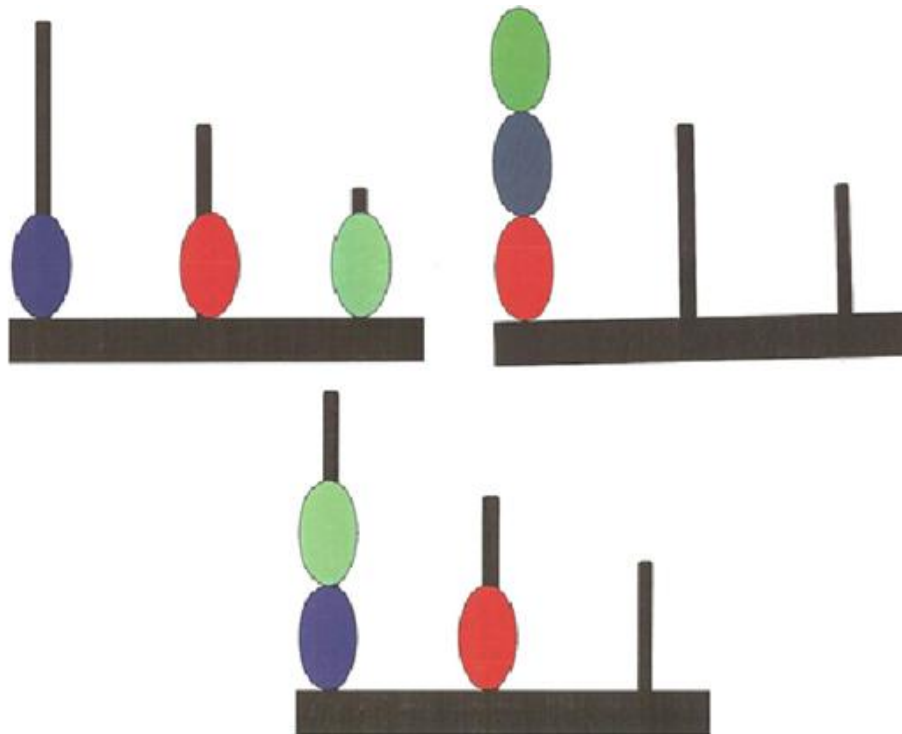


FIGURA 9- Cartões problemas da Torre de Londres manual.

A versão computadorizada da ToL foi configurada de forma similar a versão manual descrita acima. Contudo, ao invés de esferas, são discos. As demais características permaneceram iguais: cores, número de cartões-problema, ordem apresentada dos cartões. Para realizar o teste, os sujeitos utilizaram um mouse óptico sem fio (FIG. 10 a).

Após registrar o nome/sigla do sujeito, o programa inicia a instrução auditiva/visual sobre o teste (FIG. 10 b).



a

Você está prestes a iniciar um teste chamado 'Torre de Londres'. Seu objetivo é mover a pilha de discos de sua configuração original para a configuração 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 (indicado pelo tamanho do retângulo cinza). 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. Você terá apenas um número limitado de movimentos para resolver cada problema, indicado pela barra que aparecerá a direita da tela. 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. Clique no mouse para ver a demonstração.

b

FIGURA 10- a) Mouse óptico utilizado na pesquisa; b) Instrução Tol computadorizada.

Em seguida, para o sujeito se familiarizar com o teste, é realizado um problema de demonstração. Conforme se observa na FIG. 11, o “cartão problema” fica exposto no alto da tela. Abaixo na tela, encontram-se as pilhas onde o sujeito poderá movimentar os discos, de acordo com o tamanho dos retângulos. Cada vez que um disco é clicado, ele aparece na mão no centro da tela até ser clicado na pilha (retângulo) para qual deverá se deslocar (FIG. 11).

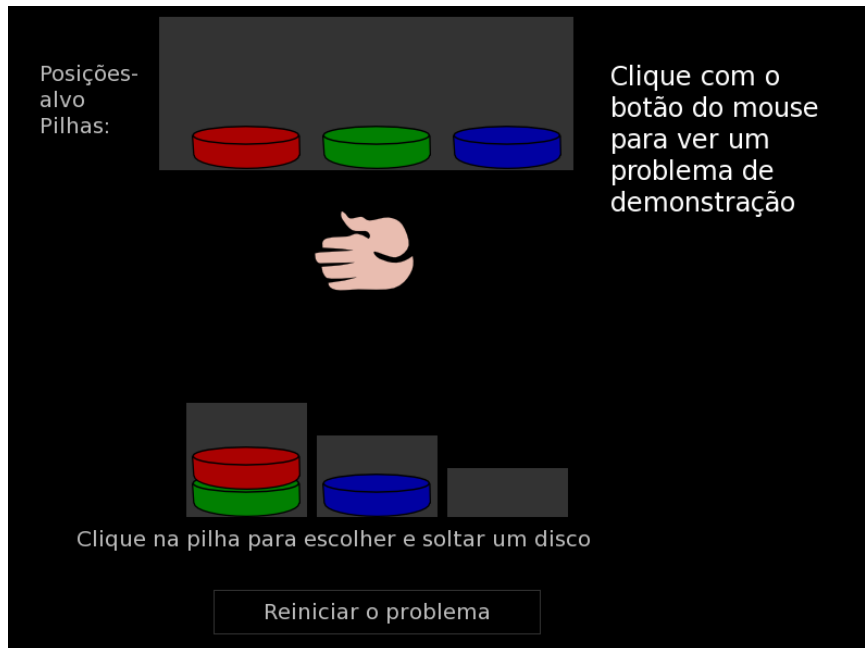


FIGURA 11- Tela inicial do problema de demonstração Tol.

Após a demonstração, o sujeito inicia o teste. No canto superior direito aparecem os feedbacks, indicando se a resolução está correta ou errada (FIG. 12).

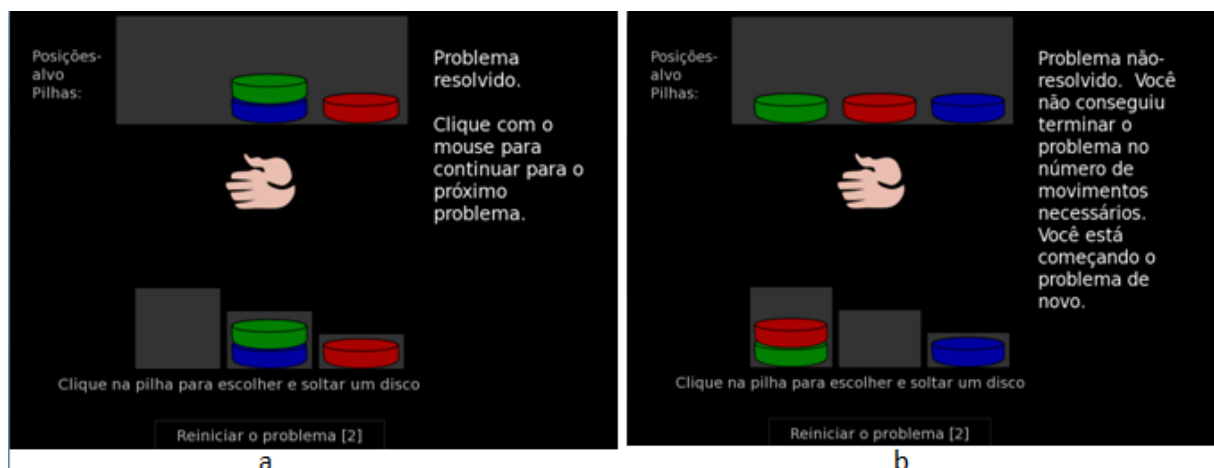


FIGURA 12- Tela da Tol computadorizada: a) feedback acerto; b) feedback erro.

Na lateral direita, ainda, há uma barra que demonstra o número de movimentos que o sujeito possui. A barra inicia-se vermelha e com o número de movimentos disponíveis, reduzindo a cada movimento realizado (FIG. 13). São registrados pelo programa os

movimentos realizados, pontuação e tempos em milissegundos. A descrição da programação deste teste encontra-se no Anexo B.

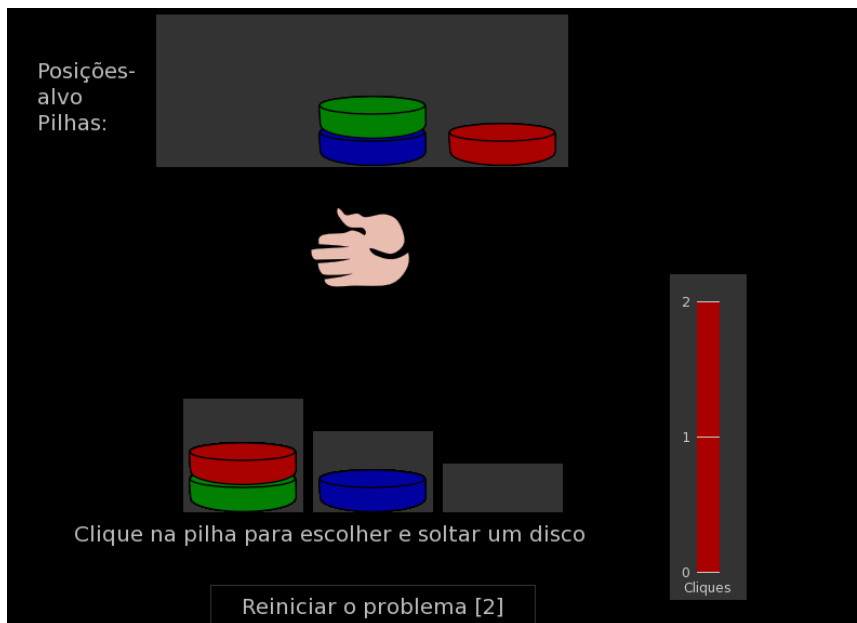


FIGURA 13- Tela de problema da Tol, mostrando a barra de movimentos.

- **Teste Wisconsin Cards Sorting Test-64 (WCST-64)**

A versão manual selecionada neste estudo é realizada pela apresentação de 4 (quatro) cartas-chave (FIG. 14), compostas por um triângulo vermelho, duas estrelas verdes, três cruzes amarelas e quatro círculos azuis, e 64 cartas-resposta. Tal material foi elaborado por Heaton e colaboradores (2005), sendo sua adaptação e padronização brasileira de uso exclusivo para psicólogos.

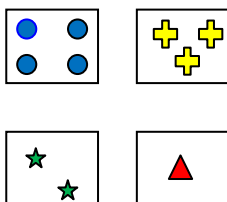


FIGURA 14- Cartas-chave Wisconsin manual

O sujeito tem como tarefa estabelecer categorias através de associações entre as cartas-resposta e as cartas-chave. As categorias são pré-determinadas e de conhecimento apenas do examinador que fornece feedback de certo ou errado a cada resposta emitida pelo sujeito. Além disso, o examinador necessita registrar cada resposta emitida pelo sujeito para

análises futuras. O teste se encerra quando acabarem as 64 cartas do deck. No protocolo pode ser visualizada a folha de registro utilizada pelo aplicador.

Quanto a versão computadorizada do WCST-64, foi ajustado o software para a uma versão similar a manual descrita acima. Assim, o teste é composto pelas mesmas cartas-chave e outras 64 cartas-resposta. Para realizar o teste, o sujeito utiliza um mouse óptico sem fio. 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. 15).



FIGURA 15- Tela da instrução do Wisconsin computadorizado.

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. Os feedbacks das associações feitas pelo sujeito são dados tanto visual quanto auditivo pelo programa (FIG. 16). O teste encerra-se ao terminarem as cartas. 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 encontram-se definições sobre termos específicos deste teste. A descrição da programação deste teste encontra-se no Anexo C.

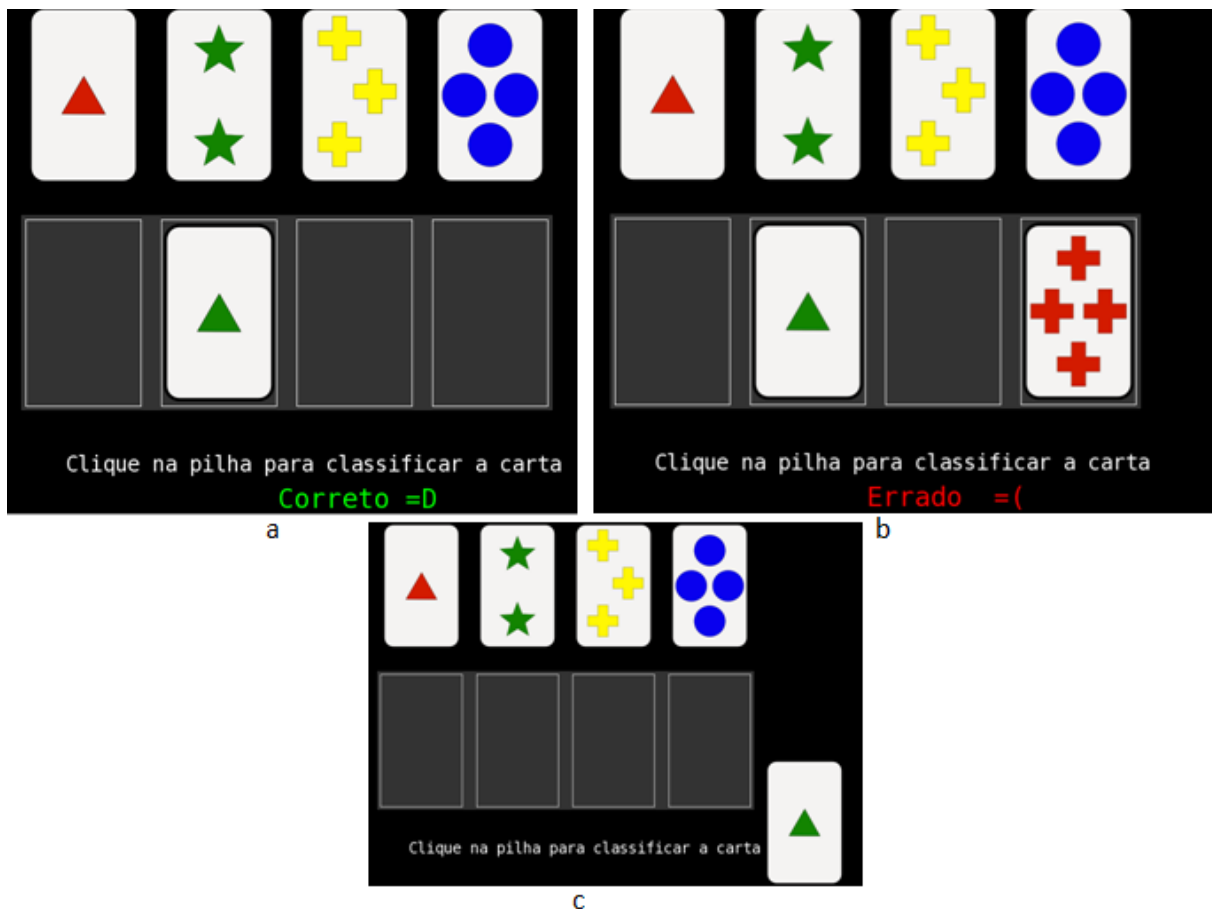


FIGURA 16- Tela do Wisconsin computadorizado: a) feedback de acerto; b) feedback de erro; c) tela com a apresentação da carta-resposta.

- **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 também de uso exclusivo para psicólogos. Destina-se a crianças de 5 anos a 11 anos e meio, deficientes mentais e pessoas idosas, contemplando, assim, o público alvo desta pesquisa (BANDEIRA, 2004).

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.

3.4 Procedimentos

Os alunos que frequentavam a Escola Estadual Afonso Pena foram convidados a participar da pesquisa por meio de um convite enviado aos pais ou responsável (ANEXO D). 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 E).

Após o consentimento dos pais e/ou responsável, estes preencheram um formulário referente aos dados sociodemográficos (ANEXO F). 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 as versões manuais e computadorizadas dos testes. Foi aplicado, ainda, o Teste das Matrizes Progressivas de Raven para verificar se os grupos estavam homogêneos. A ordem de administração foi escolhida de forma aleatória, sendo que 10 participantes responderam primeiro à versão manual e 12 responderam à versão computadorizada. As versões manuais e computadorizadas foram administradas por pesquisadoras independentes que desconheciam o resultado da avaliação que não estavam administrando. Um intervalo de 24 horas entre uma aplicação e outra foi estabelecido para que o aluno não fosse prejudicado em seu desempenho escolar. Os participantes responderam ainda a um último questionário contendo perguntas sobre o grau de motivação e dificuldade gerado pelas duas formas de administração (ANEXO H).

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. Embora a pesquisa não traga benefício direto ao

aluno e a seus familiares, 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) v.17.0 para análise estatística. O teste de Shapiro-Wilk foi utilizado para avaliar a normalidade das amostras. Utilizaram-se análises descritivas e os testes estatísticos Mann-Whitney e de correlações de Spearman. 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

O número total de participantes foram 22 crianças com discreta preponderância de meninas (TAB. 1). Os participantes eram na maioria crianças pré-púberes, com pontos socioeconômicos em média equivalentes a classe C1, segundo o Critério de Classificação Econômica Brasil.

TABELA 1- Sexo

Características	n=22
Gênero, feminino/ masculino (%)	12(54,5%)/ 10 (45,5%)
Idade, média (desvio padrão), anos	9,15 ($\pm 0,35$)
Pontos socioeconômicos, média (desvio padrão)	22,50 ($\pm 4,53$)
Número de membros na família, média (desvio padrão)	3,91 ($\pm 1,11$)

No questionário aplicado com os participantes obtiveram-se dados quanto ao uso do computador. Todos já utilizaram computador alguma vez, sendo que mais de dois terços possuem computador em casa (TAB. 2). A idade média para a primeira vez que utilizaram computador foi 6,36 ($\pm 1,09$). Na avaliação sobre quanto gostam do computador, a opinião dos participantes foi bastante positiva, com notas próximas à nota máxima. Dentre os participantes, dois terços já utilizaram computador para realizar trabalho escolar e a maioria (59,1%) gasta menos de uma hora por semana no computador (GRAF. 1).

Ainda neste questionário, os participantes foram questionados quanto a já ter executado certas habilidades³ relacionadas ao uso do computador, tais como usar o mouse e copiar ou mover um arquivo ou um pasta. Assim, de um total de 9 habilidades, em média os participantes declaram já ter executado 3,05 ($\pm 1,56$) destas.

³ Habilidades avaliadas: usar um mouse, copiar ou mover um arquivo ou uma pasta, usar um editor de texto, usar uma planilha de cálculo, comprimir arquivos no computador, escrever um programa de computador usando alguma linguagem de programação, conectar ou instalar periféricos (impressoras, câmera), usar programas de som e imagem/multimídia, abrir um programa para navegar na internet.

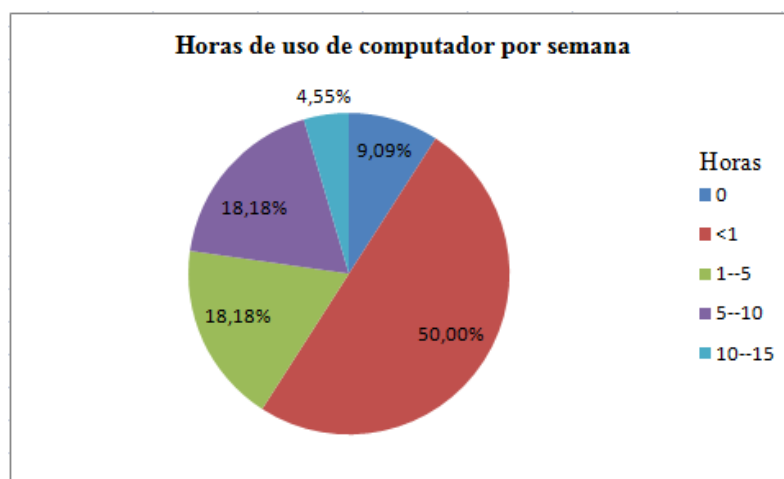


GRÁFICO 1- Horas de uso de computador por semana.

TABELA 2- Dados sobre utilização do computador

Uso do Computador	n=22
Já usou computador, n (%)	22 (100%)
Possui computador	
Sim, n (%)	15 (68,2%)
Não, n (%)	5 (22,7%)
Possui, mas não utiliza, n (%)	2 (9,1%)
Idade que usou a 1ª vez, média (desvio-padrão), anos	6,36 (±1,09)
Gosta de computador*, média (desvio-padrão)	8,91 (±1,60)
Habilidades com computador**, média (desvio-padrão)	3,05 (±1,56)
Já utilizou para trabalho escolar	
Sim, n (%)	14 (63,6%)
Não, n (%)	8 (36,4%)

* Notas atribuídas de 0 a 10, sendo 0 não gosta e 10 gosta muito.

** Foram avaliadas um total de 9 habilidades no uso do computador.

Em relação à internet, 77,3% já utilizou (GRAF. 2). Dentre as atividades realizadas na internet, 27,3% utilizam sites de busca (ex: Google) e 50% participa de sites de relacionamento (ex: Orkut). Quanto a jogos no computador, 77,3% acessam sites de jogos, 13,6% jogam no computador sem acesso a internet e 9,1% não jogam (TAB. 3).

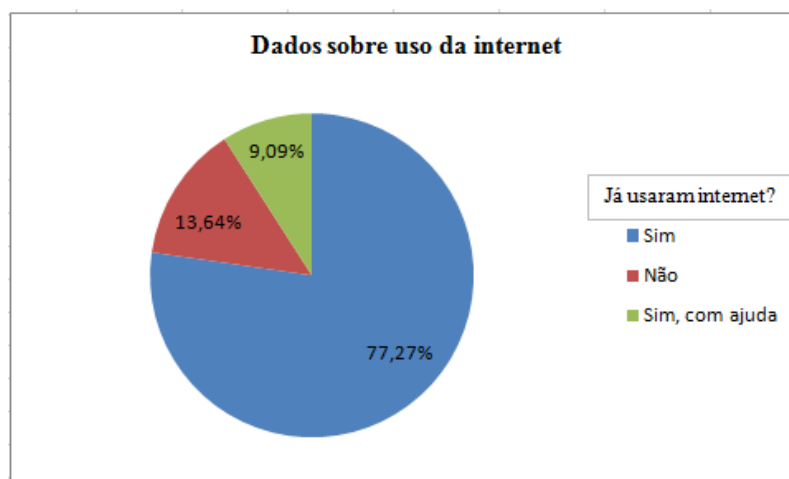


GRÁFICO 2- Porcentagem de participantes que já utilizaram internet.

TABELA 3- Dados em relação ao uso da internet

Uso da Internet	n=22
Acessou sites de informação (ex: Google)	
Sim, n(%)	6 (27,3%)
Não, n(%)	16 (72,7%)
Acessou sites de relacionamento (ex: Orkut)	
Sim, n(%)	11 (50%)
Não, n(%)	11 (50%)
Jogos	
Acessa sites de jogos, n(%)	17 (77,3%)
Joga no computador sem acesso a internet, n(%)	3 (13,6%)
Não joga no computador, n(%)	2 (9,1%)

A maioria dos participantes já jogou video-game (81,8%) e gastam em média menos de uma hora por semana nesta atividade (GRAF. 3).

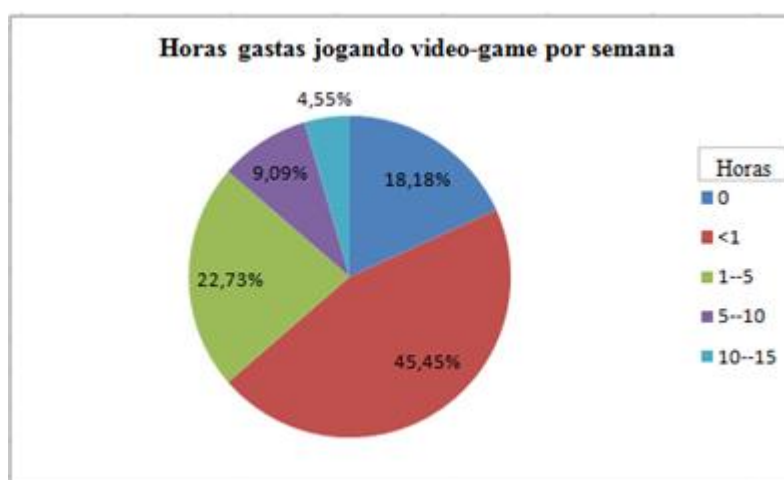


GRÁFICO 3- Porcentagem de horas gastas jogando vídeo-game por semana.

4.2 Resultados gerais dos testes

A distribuição dos participantes quanto à ordem de aplicação dos testes e do tempo gasto em cada dia foi relativamente homogênea (TAB. 4). O tempo gasto nos testes manuais foi inferior ao gasto na aplicação nos testes computadorizados (GRAF. 4). Contudo, os testes manuais necessitavam ser analisados posteriormente, sendo que os computadorizados já forneciam as análises logo após a aplicação.

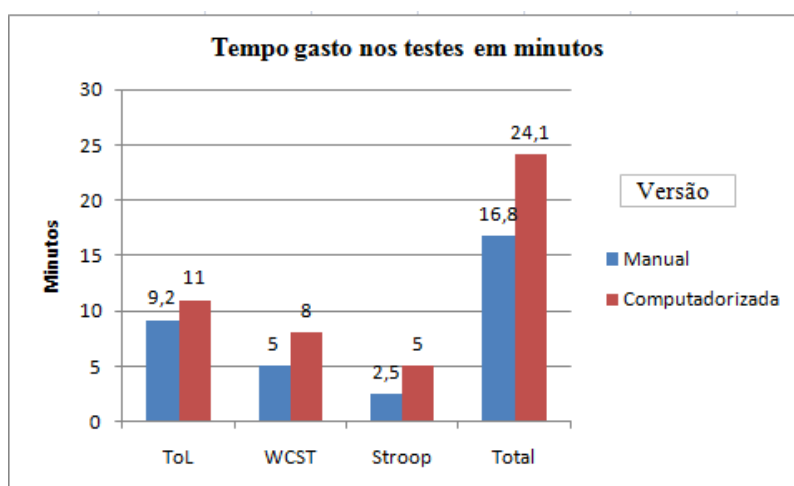


GRÁFICO 4- Tempo gasto nas aplicações dos testes.

Tabela 4- Dados da aplicação dos testes

Dados da aplicação	n=22
Primeiro dia computadorizado, n (%)	12 (54,5%)
Primeiro dia manual, n (%)	10 (45,5%)

No teste aplicado para pareamento dos participantes, Teste das Matrizes Progressivas de Raven (Raven), a maioria obteve desempenho médio ou acima da média (95,5%) (TAB. 5).

Tabela 5- Desempenho no teste Raven

Resultados	n=22
Total de Pontos , média (desvio padrão)	23,64 ($\pm$ 5,16)
Percentil , média (desvio padrão)	66,59 ($\pm$ 22,38)
Classificação	
Superior, n (%)	2 (9,1%)
Médio superior, n (%)	9 (40,9%)
Médio, n (%)	10 (45,5%)
Médio inferior	1 (4,5%)

Em geral, o desempenho nas versões computadorizadas foi pior do que nas versões manuais (TAB. 6). No teste da Torre de Londres (ToL), os participantes fizeram mais movimentos e menos pontos na versão computadorizada. Já no Teste de Classificação de Cartas Wisconsin (WCST-64), os participantes obtinham, em média, menos acertos e mais erros na versão computadorizada. No Teste de Stroop (Stroop), o tempo gasto e o número de erros nas três partes do teste também foram maior na versão computadorizada.

Tabela 6- Resultados gerais dos testes

Resultados	Manual	Computadorizada
	n=22	n=22
ToL		
Total de Pontos, média (desvio padrão)	29,09 ($\pm 2,29$)	19,82 ($\pm 9,24$)
Total de Movimentos, média (desvio padrão)	61,59 ($\pm 4,88$)	94,23 ($\pm 22,94$)
WCST-64		
Total de acerto, média (desvio padrão)	43,00 ($\pm 9,11$)	38,55 ($\pm 11,16$)
Total de erro, média (desvio padrão)	21,00 ($\pm 9,11$)	25,45 ($\pm 11,16$)
Respostas perseverativas, média (desvio padrão)	20,67 ($\pm 16,28$)	27,84 ($\pm 12,75$)
Erros perseverativos, média (desvio padrão)	11,36 ($\pm 7,49$)	10,27 ($\pm 6,15$)
Erros não- perseverativos, média (desvio padrão)	8,86 ($\pm 4,67$)	15,18 ($\pm 11,29$)
Categorias completadas, média (desvio padrão)	2,91 ($\pm 1,23$)	1,95 ($\pm 1,13$)
Stroop		
Tempo I, média (desvio padrão), milissegundos	27924,55 ($\pm 36903,95$)	81651,36 ($\pm 68968,24$)
Erro I, média (desvio padrão), milissegundos	0,64 ($\pm 0,902$)	4,59 ($\pm 10,38$)
Tempo II, média (desvio padrão), milissegundos	28227,27 ($\pm 4791,32$)	56147,14 ($\pm 24906,32$)
Erro II, média (desvio padrão), milissegundos	0,50 ($\pm 0,740$)	2,00 ($\pm 4,80$)
Tempo III, média (desvio padrão), milissegundos	39032,27 ($\pm 9374,50$)	67712,23 ($\pm 21898,02$)
Erro III, média (desvio padrão), milissegundos	3,45 ($\pm 3,03$)	4,27 ($\pm 7,04$)

Ao final das aplicações, os participantes responderam a um questionário sobre sua preferência e dificuldade em relação as versões dos testes. 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 (GRAF. 5). Em relação à dificuldade, os participantes selecionaram entre fácil (0), moderado (1) ou difícil (2) para cada teste. Assim como na avaliação de preferência, mesmo tendo desempenho melhor nas versões manuais, acharam mais difíceis as versões manuais do WCST-64 e Stroop, exceto para a ToL cuja versão computadorizada foi considerada mais difícil (TAB. 7).

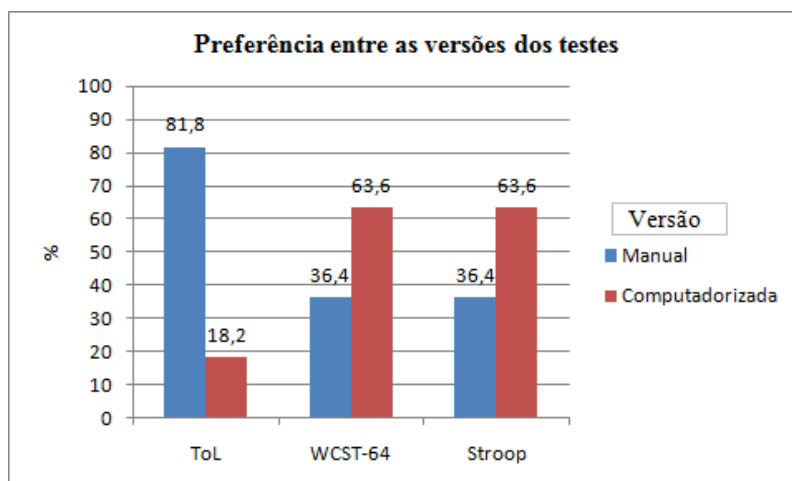


GRÁFICO 5- Percentual da preferência entre as versões dos testes.

Tabela 7- Grau de dificuldade das versões computadorizada e manual

Resultados	Manual	Computadorizada
	n=22	n=22
Dificuldade		
ToL, média (desvio padrão)	0,27 ($\pm 0,46$)	0,91 ($\pm 0,68$)
WCST-64, média (desvio padrão)	0,82 ($\pm 0,73$)	0,50 ($\pm 0,67$)
Stroop, média (desvio padrão)	0,45 ($\pm 0,74$)	0,41 ($\pm 0,59$)

4.3 Correlações entre as versões⁴

No teste ToL, não houve correlação entre as versões. Assim, o bom desempenho na versão manual não implicava no mesmo sucesso na versão computadorizada. Isso pode ser observado tanto na avaliação da correlação entre o número total de movimentos quanto entre a pontuação de ambas as versões (TAB. 8).

Tabela 8- Análise de correlações entre variáveis do teste Torre de Londres (ToL)

ToL	r	Valor de p
Manual x Computadorizada		
Número de movimentos x Número de movimentos	0,056	0,804
Pontuação x Pontuação	-0,043	0,851

⁴Interpretação dos dados de correlação: valor positivo- há correlação direta entre os dados, ou seja, quanto melhor o desempenho em um item na comparação, melhor no outro item; valor negativo- há correlação inversa entre os dados, ou seja, quanto melhor o desempenho em um item da comparação, pior no outro item.

Ao analisarmos as versões manual⁵ e computadorizada⁶ do teste WCST-64, observaram-se diversas correlações demonstrando boa consistência entre as versões (TAB. 9).

Tabela 9- Correlações entre as versões manual e computadorizada do Wisconsin Cards Sorting Test (WCST-64)

WCST-64 Manual x Computadorizada	r	Valor de p
Número de acertos	0,561	0,007*
Número e percentual de erros	0,561	0,007*
Número e percentual respostas perseverativas	-0,058	0,796
Número e percentual de erros perseverativos	0,127	0,572
Número e percentual de erros não perseverativos	0,274	0,217
Categorias completadas	0,521	0,013*
Ensaio para completar a 1ª categoria	0,114	0,615
Fracasso em manter o contexto	-0,283	0,201

* houve correlação

Desta forma, nos itens com asterísco (por exemplo: número de acertos) se o participante obteve bom desempenho na versão manual, também foi bem sucedido no mesmo item computadorizado. A análise dos fatores nível conceitual⁷ (número e percentual) e aprendendo aprender⁸ não foram consideradas nesta análise por ter ocorrido divergência na forma de correção das versões manual e computadorizada, não sendo possível, assim, garantir que os dados se referiam ao mesmo aspecto.

Na comparação entre as versões manual e computadorizada do Teste de Stroop pode-se observar forte correlação no Tempo II (em negrito na TAB. 10).

Tabela 10- Correlações entre as versões manual e computadorizada do Stroop

Stroop Manual x Computadorizado	r	Valor de p
Tempo I	0,292	0,187
Erro I	0,048	0,831
Tempo II	0,625	0,002*
Erro II	0,268	0,229
Tempo III	0,259	0,244
Erro III	-0,252	0,258

* houve correlação

⁵ A versão manual do teste passará a ser denominada apenas como “manual”.

⁶ A versão computadorizada do teste passará a ser denominada apenas como “computadorizada (o)”

⁷ Nível conceitual: descrição deste fator encontra-se no apêndice.

⁸ Aprendendo a aprender: descrição deste fator encontra-se no apêndice.

4.4 Correlações com Teste Raven e os dados dos questionários

Nesta parte, foram analisados o desempenho obtido no Raven e os dados coletados nos questionários respondidos pelos responsáveis e participantes. O teste Raven fornece três formas de resultado: pontuação, percentual e nível. Quanto maior a pontuação e o percentual, melhor será o desempenho. Já quanto menor o nível (classificação de 1 a 5)⁹, melhor será o desempenho.

De acordo com os dados obtidos, quanto mais tempo gasto na internet ou computador, maior o desempenho no Teste Raven. Houve tendência a correlação entre o percentual obtido no teste e a idade que utilizou o computador pela primeira vez, sendo quanto mais novo melhor o desempenho no Raven. Outra tendência a correlação foi entre o nível definido pelo Raven e o critério socioeconômico, indicando relação entre baixo critério socioeconômico com melhor desempenho no teste (TAB. 11).

Tabela 11- Correlações entre o Teste Raven e Dados do Questionário

Teste Raven	r	Valor de p
Pontuação Raven		
Tempo gasto na internet	0,648	0,001*
Tempo gasto no computador	0,583	0,004*
Percentual no Raven		
Tempo gasto na internet	0,670	0,001*
Tempo gasto no computador	0,610	0,003*
Idade que usou computador pela 1ª vez	-0,387	0,075**
Nível no Raven		
Tempo gasto na internet	-0,557	0,007*
Tempo gasto no computador	-0,504	0,017*
Critério socioeconômico	-0,394	0,069**

* houve correlação

**tendência a correlação

Na comparação do Teste Raven com as versões de ToL, WCST-64 e Stroop, observou-se correlações em diversos itens descritos a Tabela 12. Conforme descrito, a maioria dos itens em que houve correlação são das versões computadorizadas. Na ToL houve desempenho melhor na versão computadorizada dos participantes que obtiveram desempenho

⁹ Classificação por nível no Teste Raven: 1- superior, 2- acima da média, 3- média, 4- abaixo da média, 5- deficiente.

melhor no Raven. No WCST-64, aqueles que obtiveram melhor desempenho no Raven também acertaram mais, cometeram menos erros, completaram mais categorias e fracassaram menos em manter o contexto na versão computadorizada. Em anexo segue tabela completa com os demais itens que não tiveram correlação (ANEXO I).

Tabela 12- Comparações do Teste Raven com ToL, WCST-64 e Stroop

ToL	r	Valor de p
Pontuação Raven x		
Número de movimentos computadorizado	-0,412	0,057*
Pontuação computadorizado	0,403	0,063**
Percentual Raven x		
Número de movimentos computadorizado	-0,477	0,025*
Pontuação computadorizado	0,473	0,026*
Nível Raven x		
Número de movimentos computadorizado	0,463	0,030*
Pontuação computadorizado	-0,469	0,028*
WCST-64		
Pontuação Raven x		
Categorias completadas computadorizado	0,505	0,017*
Fracasso em manter o contexto computadorizado	-0,428	0,047*
Percentual Raven x		
Número e percentual de respostas perseverativas manual	-0,376	0,085**
Número de acertos computadorizado	0,501	0,018*
Número e percentual de erros computadorizado	-0,501	0,018*
Categorias completadas computadorizado	0,582	0,005*
Nível Raven x		
Número de acertos computadorizado	-0,549	0,008*
Número e percentual de erros computadorizado	0,549	0,008*
Categorias completadas computadorizado	-0,527	0,012*
Stroop		
Pontuação Raven x		
Tempo II manual	-0,434	0,043*
Tempo I computadorizado	-0,455	0,034*
Tempo II computadorizado	-0,425	0,049*
Percentual Raven x		
Tempo I manual	-0,418	0,053*
Tempo II manual	-0,508	0,016*
Tempo I computadorizado	-0,448	0,037*
Tempo II computadorizado	-0,409	0,058*

Nível Raven x		
Tempo I manual	0,530	0,011*

* houve correlação
 **tendência a correlação

Já em relação aos dados coletados pelo questionário, observou-se que participantes que gostam de computador passam mais horas jogando video-game e quanto mais novos utilizaram pela primeira vez o computador, maior o tempo que gastam na internet ou no computador por semana. Ainda, os participantes que costumam passar mais horas no computador por semana, possuem mais habilidades no uso do computador e gastam mais horas na internet (TAB. 13).

Além das correlações acima descritas, houve algumas tendências a correlação. Participantes que gostam mais de computador tenderam a possuir mais habilidades no o uso deste. Tenderam também a possuir mais habilidades aqueles que eram mais novos quando usaram pela primeira vez o computador (TAB. 13). A tabela completa, incluindo os dados que não apresentaram correlação, encontra-se no Anexo J.

Tabela 13- Comparações entre os dados dos questionários

Dados dos questionários	r	Valor de p
Gostar de computador		
Habilidades com computador	0,381	0,080**
Tempo gasto com vídeo-game	0,481	0,027*
Idade que utilizou computador pela 1ª vez		
Tempo gasto no computador	-0,564	0,006*
Habilidades com computador	-0,379	0,082**
Tempo gasto na internet	-0,556	0,007*
Tempo gasto no computador		
Habilidades com computador	0,528	0,012*
Tempo gasto na internet	0,955	<0,001*
Habilidade com computador		
Tempo gasto na internet	0,585	0,004*

* houve correlação

**tendência a correlação

A seguir, será abordado as comparações realizadas entre os dados dos questionários e as versões manual e computadorizada dos testes. No Anexo L encontram-se todos os dados, inclusive os que não apresentaram correlação.

Em relação a ToL, observou-se que quanto maior a pontuação na versão manual, menor a idade que o participante possuía quando utilizou computador pela primeira vez. Já na versão computadorizada deste teste, quanto mais horas os participantes gastavam com a internet, realizaram menos movimentos e tenderam a pontuar mais (TAB. 14).

Tabela 14- Comparações dos dados dos questionários e ToL

Dados do questionário x	r	Valor de p
Idade que usou computador pela 1ª vez x		
Pontuação manual	-0,527	0,012*
Tempo gasto no computador x		
Número de movimentos computadorizado	-0,374	0,087**
Tempo gasto na internet x		
Número de movimentos computadorizado	-0,429	0,047*
Pontuação computadorizado	0,407	0,060**

* houve correlação

**tendência a correlação

Na comparação com a aplicação manual do WCST-64, os participantes mais velhos obtiveram pior desempenho nesta versão do teste (TAB. 15). Ainda, os participantes de classe socioeconômica menor tenderam a cometer mais respostas perseverativas e mais erros perseverativos.

Já na versão computadorizada do WCST-64, essas correlações ocorreram de forma mais evidente, ou seja, quanto menor a classe econômica do participante, pior o seu desempenho. Curiosamente, participantes que gostam de computador obtiveram desempenho pior nesta versão. Ressalta-se que também ocorreu uma fraca tendência similar na versão manual (ANEXO L). Já tempo gasto no computador e na internet e possuir mais habilidades no uso do computador, tiveram correlações com melhores desempenhos em determinados itens da versão computadorizada. Desta forma, apenas gostar de computador não interfere no desempenho do sujeito, mas utilizar o computador relaciona-se com bom desempenho deste teste tal como ocorreu no Raven (TAB. 15).

Tabela 15- Comparações dos dados dos questionários e WCST-64

Dados do questionário x	r	Valor de p
Manual		
Idade x		
Número de acertos	-0,498	0,018*
Número e percentual de erros	0,498	0,018*
Número e percentual de respostas perseverativas	1,463	0,030*
Número e percentual de erros perseverativos	0,509	0,016*
Categorias completadas	-0,558	0,007*
Critério Socioeconômico x		
Número e percentual de respostas perseverativas	-0,386	0,076**
Número e percentual de erros perseverativos	-0,415	0,055**
Computadorizado		
Critério Socioeconômico x		
Número de acertos	0,485	0,022*
Número e percentual de erros	-0,485	0,022*
Número e percentual de erros não perseverativos	-0,466	0,029*
Ensaio para completar a 1ª categoria	-0,459	0,032*
Gostar de computador x		
Número de acertos	-0,449	0,036*
Número e percentual de erros	0,449	0,036*
Idade que usou computador pela 1ª vez x		
Fracasso em manter o contexto	0,504	0,017*
Tempo gasto no computador x		
Número de acertos	0,416	0,054*
Número e percentual de erros	-0,416	0,054*
Categorias completadas	0,552	0,008*
Fracasso em manter o contexto	-0,437	0,042*
Habilidades com computador x		
Fracasso em manter o contexto	-0,458	0,032*
Tempo gasto na internet x		
Número de acertos	0,464	0,030*
Número e percentual de erros	-0,464	0,030*
Número e percentual de erros não perseverativos	-0,399	0,066**
Categorias completadas	0,534	0,010*
Fracasso em manter o contexto	-0,394	0,069**

* houve correlação

**tendência a correlação

E, por fim, comparando-se os dados dos questionários com o Teste de Stroop observou-se que os participantes com nível socioeconômico menor gastaram mais tempo na parte I de ambas as versões e nos tempos das partes II e III computadorizado. Quanto mais novos eram os participantes quando usaram computador pela primeira vez, melhor o desempenho na parte III e no tempo da parte II. Os participantes que gastam mais horas no computador cometeram menos erros na parte III computadorizada neste teste. Já aqueles que gastam mais horas na internet tenderam a gastar menos tempo na parte III manual e a cometer menos erros na mesma parte só que da versão computadorizada (TAB. 16).

Tabela 16- Comparações dos dados dos questionários e Stroop

Dados do questionário x	r	Valor de p
Critério Socioeconômico x		
Manual		
Tempo I	-0,513	0,015*
Computadorizado		
Tempo I	-0,436	0,042*
Tempo II	-0,429	0,046*
Tempo III	-0,493	0,020*
Gostar de Computador x		
Tempo III computadorizado	0,379	0,082**
Idade que usou computador pela 1ª vez x		
Manual		
Tempo II	0,379	0,082**
Tempo III	0,444	0,039*
Erro III	0,371	0,089**
Tempo Gasto no computador x		
Erro III computadorizado	-0,473	0,026*
Tempo gasto na internet x		
Tempo III manual	-0,373	0,087**
Erro III computadorizado	-0,402	0,063**

* houve correlação

**tendência a correlação

4.5 Comparações entre gêneros

Nas comparações entre gêneros, ocorreram poucos dados significativos. Os meninos jogavam bem mais video-game do que as meninas. Já na versão computadorizada do WCST-64 as meninas tenderam a pontuar mais do que os meninos. E, na versão computadorizada do Stroop, os meninos, em geral, foram mais rápidos sem que isso incorresse em prejuízo no desempenho (TAB. 17).

Tabela 17- Comparações relacionadas ao gênero

	Média	DP*
Horas jogando video-game (p=0,003)		
Meninas	0,82	0,182
Meninos	2,10	0,314
WCST-64 computadorizado- aprendendo a aprender (p=0,031)		
Meninas	1,9075	1,12116
Meninos	-0,3570	0,35700
Stroop computadorizado		
Tempo II (p= 0,035)		
Meninas, milissegundos	64331,75	8813,708
Meninos, milissegundos	46325,60	3340,634
Tempo III (p= 0,008)		
Meninas, milissegundos	79243,08	5996,192
Meninos, milissegundos	53875,20	4489,321

* DP= Desvio-padrão

Discussão

5. DISCUSSÃO

Conforme descrito nos resultados, nossos dados confirmam a hipótese sobre similaridade entre as versões manual e computadorizada do WCST-64, considerando-se que houve correlação entre os itens número de acertos, número e percentual de erros e categorias completadas. Tal resultado foi mencionado em estudos similares, porém com diferenças na metodologia (SHAN *et al.*, 2008; TIEN *et al.*, 1996; WAGNER; TRENTINI, 2009).

No estudo de Tien *et al.* (1996), realizado com 28 pacientes psiquiátricos do “Johns Hopkins Hospital Phipps Psychiary Service” (EUA) e 5 indivíduos normais, com idades entre 17 e 65 anos, aplicou-se a versão de 128 cartas, sendo 64 cartas na versão manual e 64 cartas na versão computadorizada. O grupo foi dividido de forma que metade respondesse a versão manual primeiro e a outra metade a computadorizada. Enquanto o sujeito respondia a versão manual, o aplicador registrava as respostas no computador de forma que ao chegar à carta 64 o participante pudesse retomar o teste de onde parou no computador. Ainda, utilizaram na versão computadorizada um teclado adaptado com 4 botões associados as cartas-chave e o feedback aparecia escrito na tela do computador, permanecendo por um segundo. Os resultados encontraram similaridade entre as versões nos itens: resposta perseverativa, erros perseverativos e fracasso em manter o contexto.

Da mesma forma, na pesquisa conduzida no Brasil por Wagner e Trentini (2009) com 54 indivíduos, idades entre 60 e 82 anos, utilizaram na versão computadorizada os botões 1, 2, 3 e 4 do teclado associados às cartas-chave. Neste estudo, metade dos sujeitos responderam a versão completa manual (128 cartas) e a outra metade a versão completa computadorizada. Os dois grupos foram pareados por idade e grau de escolaridade. Na análise estatística realizada não observaram diferenças significativas entre os grupos nos itens: respostas corretas, erros perseverativos, percentual de resposta de nível conceitual e número de categorias completadas.

Já no estudo de Shan *et al.* (2008) em Taiwan, com 475 participantes adultos, o objetivo inicial era a obtenção de dados normativos para adultos neste teste. Contudo, na mesma pesquisa realizaram a comparação entre as versões computadorizada e manual. No primeiro estágio da pesquisa aplicaram com 148 sujeitos a versão manual e no segundo estágio, a versão computadorizada foi aplicada em 327 sujeitos. Ainda, o aplicador auxiliava na versão computadorizada, clicando a resposta apontada pelo sujeito e fornecendo o

feedback oralmente. Nos resultados, os autores afirmam não terem encontrado diferença significativa entre as versões nos itens avaliados: percentual de erros, percentual de respostas perseverativas, percentual de erros perseverativos, percentual de erros não perseverativos, percentual de respostas de nível conceitual e número de categorias completadas;

Assim, apesar das diferenças metodológicas, todos esses estudos consideraram não haver diferenças significativas entre as suas versões.

Em nosso estudo, o público selecionado foram crianças com faixa etária entre 8 e 10 anos, sendo utilizado a versão curta do teste com 64 cartas. Apesar da ordem de aplicação ser similar ao estudo de Tien *et al.*(1996), diferencia-se pelas versões não terem sido aplicadas no mesmo dia e, portanto, não serem uma sequência. Outro ponto distinto do presente estudo refere-se a forma do feedback e autonomia do participante, pois esse utilizava um mouse óptico para clicar na resposta desejada e recebia um feedback visual e auditivo do próprio programa. Tais aspectos podem ter contribuído para uma maior proximidade entre as versões.

De forma geral, o desempenho na versão manual foi superior ao da versão computadorizada. Contudo, é incerto o que essa diferença de desempenho significa em termos clínicos. Pesquisas futuras devem investigar se essa provável diferença do grau de dificuldade do teste se reflete, por exemplo, em maior sensibilidade para disfunções executivas.

Na versão manual, ainda, indivíduos mais velhos obtiveram desempenho inferior. Como todas as crianças são pertencentes ao 4º ano do ensino fundamental, temos como hipótese de que as crianças mais velhas são aquelas com mais dificuldade de aprendizagem.

Gostar de computador não interferiu no desempenho do teste em ambas as versões, porém, horas de utilização do computador/internet correlacionaram-se a um melhor desempenho na versão computadorizada. Desta forma, é possível que a hipótese sobre uso de computador influenciar no desempenho das versões computadorizadas esteja correta. Observou-se também que indivíduos de classe socioeconômica menor obtiveram pior desempenho em ambas as versões, e com maior prevalência em categorias avaliadas no teste computadorizado. Tal desempenho pode estar associado a estas crianças serem menos estimuladas, por exemplo, utilizam menos computador. Entretanto, a correlação entre pontos socioeconômicos e uso de horas de computador ou video-game não atingiu significância estatística.

Na ToL também houve boa correlação entre a versão computadorizada e horas de uso na internet. Já no teste de Stroop não se observou esta mesma correlação. Ao contrário dos demais testes, na versão manual do Teste de Stroop, quanto mais novos os sujeitos utilizaram computador pela primeira vez, melhor foi seu desempenho em algumas etapas. Já

na versão computadorizada deste teste, quanto mais horas gastas no computador, melhor o desempenho na parte III.

Outro dado interessante relaciona-se ao melhor desempenho não só nos testes computadorizados tal como WCST-64, e ToL, mas principalmente no Raven que apresentou influência significativa do uso de computador com o desempenho.

Em relação a comparação entre as versões da Torre de Londres, não houve correlação entre as duas versões. Sendo assim, obter um bom desempenho na versão manual não implicava em obter bom desempenho na versão computadorizada. Algumas hipóteses foram elencadas a respeito desses dados. A versão computadorizada foi a primeira a ser aplicada dos três testes. O nível de ansiedade dos sujeitos poderia estar mais elevado nesta etapa do processo. Outra observação refere-se a crianças mais estimuladas (pontuação maior no Raven, tempo de internet e uso do computador) terem obtido melhor desempenho na ToL computadorizada. Durante as avaliações, a aplicadora observou que crianças com menos contato com computadores tendiam a errar mais neste teste, inclusive apresentando inicialmente dificuldades no manuseio do mouse. Contudo, essas dificuldades reduziam no decorrer da aplicação. Uma alternativa para reduzir a discrepância, então, poderia ser criar uma etapa prévia de treinamento, reduzindo a ansiedade causada pelo teste e melhorando as habilidades com os instrumentos utilizados.

A instrução inicial do teste também pode ter influenciado no desempenho pior na ToL computadorizada. Isto pode ter ocorrido devido a instrução ter sido dada em uma única tela e, apesar de elas serem dadas tanto auditiva quanto verbalmente, pode ter sido cansativa para a faixa etária avaliada. Qualitativamente, observou-se que, durante a fase de instrução, as crianças frequentemente se distraíam olhando para os lados ou para aplicadora, balançando as pernas e mudando a postura corporal (encurvando-se mais na cadeira, debruçando-se). Essa dificuldade pode ser minimizada alterando a forma de instrução, tornando-a mais atrativa para esta faixa etária e/ou aplicando com outras faixa etárias (14 anos para cima) para discernir se de fato a instrução influenciou no baixo desempenho.

Outro dado interessante refere-se a ToL ter sido o único dos 3 testes em que as crianças preferiram a aplicação manual do que a computadorizada. Como mencionado no referencial teórico, a motivação do sujeito durante a aplicação dos testes é essencial para que seu desempenho sofra poucas interferências. Neste sentido, uma das hipóteses tratava-se de que os testes computadorizados possuam elementos favorecedores quanto à motivação principalmente na faixa etária avaliada. Assim, na avaliação das crianças, o Teste de Stroop e o WCST-64 de fato foram mais motivadores, mas o mesmo não ocorreu com a ToL.

Uma hipótese pode estar relacionada com a teoria de Jean Piaget, o primeiro a dividir em estágios o desenvolvimento cognitivo. Segundo este autor, há 4 estágios para o desenvolvimento: sensório motor (0 a 2 anos), pré operatório (2 a 7 anos), operações concretas (7 a 12 anos) e operações formais (12 anos em diante). A faixa etária avaliada na pesquisa, assim, encontra-se no estágio de operações concretas. Tais fases coincidem com os denominados surtos de crescimento [“growth spurts”- tradução livre]. Estudado por Epstein, os surtos de crescimento referem-se a um aumento do peso do cérebro que ocorre desde os 3 aos 10 meses (aumentando cerca de 30% até 1 ano e meio) até aproximadamente 16 anos. Esses surtos, ocorrem em períodos similares as fases descritas por Piaget, sendo que entre as idades de 2 a 4 anos, 6 a 8 anos, 10 a 12 anos são idênticas, acrescentando apenas uma nova fase não avaliada por Piaget que abrange as idades de 14 a 16 ou mais anos de idade. Ainda, nestas últimas fases o peso do cérebro aumenta aproximadamente de 5% a 10% em cada etapa ocorrida de dois em dois anos (KOLB; WHISHAW, 2003).

No estágio de operações concretas do desenvolvimento cognitivo, apesar da criança ter potencial para raciocinar de forma coerente, tanto os esquemas conceituais como as ações mentalmente se referem a objetos ou situações conceituais que possam ser tocadas ou imaginadas de forma concreta (*ibidem*). Desta forma, como a aplicação manual da ToL envolve a manipulação das esferas, pode ter sido mais motivador para os participantes, tendo em vista que crianças desta faixa etária tendem a ter maior facilidade com objetos concretos.

Apesar disso, crianças que usaram o computador mais precocemente obtiveram maior pontuação na ToL manual ao passo que tempo na internet e o desempenho no Raven estavam relacionados a maior desempenho na ToL computadorizada. Assim, complementando o mencionado anteriormente, crianças mais estimuladas tiveram desempenho superior em ambas as versões.

Ainda, houve correlações inesperadas como bom desempenho na ToL computadorizada e demora no estabelecimento da primeira categoria do WCST-64 manual e mais erros cometidos na parte III do Stroop manual. Mais estudos são necessários para esclarecer tais aspectos.

No teste Stroop observou-se correlação entre as versões apenas quanto ao tempo gasto na execução das três etapas, contudo não houve correlação quanto aos registros de erros. Uma hipótese para tal ausência de correlação pode estar relacionado a forma de registro de erros destas versões. Na versão manual, é registrado como erro cada nomeação verbal inadequada sem correção espontânea (ex: disse amarelo ao invés de azul) e a criança poderia prosseguir no teste, mesmo tendo emitido uma resposta errada. Já na versão computadorizada,

considerava-se um erro cada vez que se apertava o botão, passando para o próximo estímulo apenas quando a criança emitisse a resposta correta. Por isso, o número de erros registrado na versão computadorizada foi superior a versão manual, e, conseqüentemente houve discrepância no desempenho das versões.

Ainda, observou-se que duas meninas não compreenderam a instrução do teste apertando aleatoriamente os botões. Uma delas compreendeu ao final da primeira parte, mas a outra persistiu cometendo erros excessivos até o final do teste. Assim, os erros registrados nestes dois casos foram muito superior aos erros emitidos pelos demais sujeitos. Em análise realizada sem os dados dessas duas meninas observou-se que houve mais correlações entre as versões manual e computadorizada, tal como: Tempo II manual com os Tempos computadorizados I ($p=0,006$, $r=0,591$), II ($p=0,003$, $r=0,624$) e III ($p=0,031$, $r=0,483$); entre os Erros II das duas versões ($p=0,048$, $r=0,448$); Tempo III manual com os Tempos computadorizados I ($p=0,016$, $r=0,532$), II ($p=0,028$, $r=0,492$) e III ($p=0,053$, $r=0,439$); e entre o Erro III manual com Erro II computadorizado ($p=0,031$, $r=0,483$) e tendência com o Tempo II computadorizado ($p=0,068$, $r=0,417$). Nesta nova análise, observa-se maior semelhança entre as versões e, portanto, necessidade de mais estudos.

Quando analisamos os dados dos questionários aplicados e do Raven obtemos diversos dados dignos de nota. Os participantes que gostam mais de computador passam mais horas jogando video-game e os meninos jogam mais video-game do que as meninas. Já no teste WCST manual, passar mais horas no video-game correlacionou-se com pior desempenho na categoria aprendendo a aprender. Este dado pode ter sido influenciado pela ausência de informações mais específicas como, por exemplo, o tipo de jogo que o sujeito costuma jogar, e, conseqüentemente desenvolver habilidades distintas. Algumas pesquisas comparando o desempenho em testes atencionais com grupos de jogadores de video-game e não jogadores de video-game demonstraram que aqueles que jogam video-game tem desempenho melhor nestes testes (ALVES; CARVALHO, 2010; CASTEL; PRATT; DRUMMOND, 2005; DUNBAR; HILL; LEWIS, 2001; GREEN; BAVELIER, 2003). Ainda, os autores Green e Bavelier (2003), apontam para um desempenho diferenciado entre jogos de ação comparados a outros jogos, sendo aqueles melhores para atenção visual. Assim, uma limitação do presente estudo pode estar relacionado a ausência de discriminação entre os tipos de jogos utilizados pelos participantes.

As comparações com o desempenho no Raven destacaram-se significativamente das análises. Por exemplo, crianças que passam mais tempo no computador ou internet por semana e começaram a utilizá-lo mais precocemente, além de possuir mais habilidades no uso

do computador, obtiveram melhor desempenho no Raven. Ainda, o bom desempenho no Raven estava relacionado a melhor desempenho em todas versões computadorizadas e alguns itens da aplicação do Stroop Manual. Esses dados corroboram a hipótese de que o conhecimento em informática favorece no desempenho dos testes computadorizados e, mais do que isso, melhoram o desempenho no teste de inteligência geral (Raven).

Conclusão

6 CONCLUSÃO

De acordo com os dados obtidos nesta pesquisa, o WCST- 64 computadorizado possui características similares com a versão manual utilizada. O mesmo não pode ser verificado nas versões da ToL e Stroop. Contudo, mais estudos são necessários para que as limitações deste estudo em relação a comparação entre as versões sejam esclarecidas.

Considerando-se as hipóteses levantadas, observou-se que as versões computadorizadas dos testes WCST-64 e o Stroop foram mais motivadores para a faixa etária avaliada e que crianças com maior conhecimento em informática, em geral, obtiveram melhor desempenho nas versões computadorizadas. Ainda, quanto mais estimuladas as crianças, melhor seu desempenho no Raven. Tais dados demonstram a relevância de pesquisas envolvendo testes computadorizados para o aprimoramento na avaliação das funções executivas, auxiliando no diagnóstico e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos.

A presente pesquisa de validade convergente desses instrumentos foi um primeiro passo para a realização de estudos futuros que objetivem normatizar as versões computadorizadas na população brasileira.

Referências

REFERÊNCIAS

- ALVES, L; CARVALHO, A.M. Video-game e sua influência em teste de atenção. *Psicologia em Estudo*, Maringá, v.15, n.3, p. 519-525, 2010.
- BANDEIRA, D.R *et al.* Matrizes progressivas coloridas de Raven- escala especial: normas para Porto Alegre, RS. *Psicologia em estudo*, v.9, n. 3, p. 479-486, 2004.
- 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. SãoPaulo: Comitê Gestor da Internet no Brasil, 2010.
- BARON, I.S. *Neuropsychological evaluation of the child*. Nova York: Oxford University Press, Inc, 2004. 451p.
- CAPOVILLA, A. Desenvolvimento e validação de instrumentos neuropsicológicos para avaliar funções executivas. *Avaliação Psicológica*, v.5, n.2, p. 239-241, 2006.
- CAPOVILLA, A. *et al.* Avaliação Neuropsicológica das funções executivas e relação com desatenção e hiperatividade. *Avaliação Psicológica*, v.6, n.1, p.51-60, 2007.
- CASTEL, A. D.; PRATT, J.; DRUMMOND, E. The effects of action video game experience on time course of inhibition of return and efficiency of visual search. *Acta Psychologica*, v.119, p 217-30, 2005.
- CERNICH, A. *et al.* Sources of error in computerized neuropsychological assessment. *Archives of Clinical Neuropsychology*, v.22, suplemento 1, p.S32-48, 2007.
- CUNHA, J.A. *Psicodiagnóstico – V*. Porto Alegre: ARTMED, 2004.
- DUNBAR, G.; HILL, R.; LEWIS, V. Children's attentional skill's and road behavior. *Journal of Experimental Psychology: Applied*, v. 7, n.3, p. 227-34, 2001.
- DUNCAN, M.T. Obtenção de dados normativos para desempenho no teste de Stroop num grupo de estudantes do ensino fundamental em Niterói. *Jornal Brasileiro de Psiquiatria*, v.55, n1, p 42-48, 2006.
- FONTENELLE, L. Aspectos neuropsicológicos do transtorno obsessivo-compulsivo. *Revista Brasileira de Psiquiatria*, v.23, suppl. 2, 2001.
- GOLDBERG, E. *O cérebro executivo: lobos frontais e a mente civilizada*. Rio de Janeiro: Imago, 2002.

GREEN, C. S.;BAVELIER, D.. Action Video game modifies visual selective attention. *Nature*, v.423, p. 534-37, 2003.

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.

HEATON, R.K *et al.* *Teste Wisconsin de Classificação de Cartas*. São Paulo: Casa do Psicólogo, 2005.

KLEIN, M. *et al.* Effect of test duration on age-related differences in Stroop interference. *Journal of Clinical and Experimental Neuropsychology*, 19, p. 77–82, 1997 *apud* STRAUSS, E; SHERMAN, E.M.S.; SPREEN, O. *A compendium of neuropsychological tests: administration, norms, and commentary*. 3° Ed. Oxford University Press: New York, 2006.

KOLB, B; WHISHAW, I.C. *Fundamentals of human neuropsychology*. 5 ed. New York: Worth Publ, 2003.

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.

LEUNG, H.C. *et al.* An event-related functional MRI study of the Stroop color word interference task. *Cerebral Cortex*, v. 10, p.552-560, 2000.

LEZAK, M.D. *Neuropsychological assessment*. 4 ed. New York: Oxford University Press, 2004.

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

MALLOY-DINIZ, L.F. *et al.* Planning abilities of children aged 4 years and 9 months to 8 ½ years. Effects of age, fluid intelligence and school type on performance in the Tower of London test. *Dementia & Neuropsychologia*, v.2, n. 1, p.26-30, 2008b.

MASSON, J.D.; DAGNAN,D; EVANS, J. Adaptation and validation of the Tower of London test of planning and problem solving in people with intellectual disabilities. *Journal of Intellectual Disability Research*, v. 54, part 5, p 457-467, maio 2010.

MUELLER, S. T. *PEBL: The psychology experiment building language (Version 0.10)* [Computer experiment programming language], 2009. Disponível em: <http://pebl.sourceforge.net>. Acesso em: maio 2010.

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.

SANTOS, B. S. *et al.* Informática educativa e o processo motivacional dos adolescentes. 200-. Disponível em:<http://www.oei.es/CongresoCiudadania/Comunicaciones/INFORMATICAEDUCATIVAEOPROCESSOMOTIVACIONALDOSADOLESCENTES.pdf>. Acesso em: maio 2010.

- 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, cap. 35, p. 344-349.
- SHAN, K. *et al.* Adult normative data of the Wisconsin Card Sorting Test in Taiwan. *Journal of the Chinese Medical Association*, v. 71, n.10, p.517-522, outubro 2008.
- 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.
- STROOP, J.R. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18, p. 643-662, 1935.
- 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.
- TRENTINI, C.M. *et al.* *Teste Wisconsin de Classificação de Cartas: versão para idosos*. São Paulo: Casa do Psicólogo, 2010.
- WAGNER, G.P.; TRENTINI, C.M. Assessing executive functions in older adults: a comparison between the manual and the computer-based versions of the Wisconsin Card Sorting Test. *Psychology & Neuroscience*, v.2, n.2, p. 195-198, 2009.

Apêndice

APÊNDICE

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 Stroop

```

define Start(p)
{
## "Victoria" implementation of the stroop task.
##
##
##
gLanguage <- "PT"
##The number of trials per block (and condition)
reps <- 6 #24 items per test
Initialize()
gSubNum <- GetSubNum(gWin)
## Uncomment to eliminate busy-wait (mostly for older machines or
## laptops that might overheat.)
gSleepEasy <- 1
##Create the response order at the bottom
colororder <- Shuffle([1,2,3,4])
gColorNames <- SortBy(gColorBaseNames,colororder)
gColors <- SortBy(["red","green","blue","yellow"],colororder)
#####
## Create the basic stimuli
#####
gBlack <- MakeColor("black")
gRed <- MakeColor("red")
gGreen <- MakeColor("darkgreen")
gBlue <- MakeColor("navy")
gYellow <- MakeColor("yellow1")
colors <- [gRed, gGreen, gBlue, gYellow]
gFontBlack <- MakeFont(gPEBLBaseFontMono, 0, 24, gBlack,gbg,0)
gFontRed <- MakeFont(gPEBLBaseFontMono, 0, 24, gRed,gbg,0)
gFontGreen <- MakeFont(gPEBLBaseFontMono, 0, 24, gGreen,gbg,0)
gFontBlue <- MakeFont(gPEBLBaseFontMono, 0, 24, gBlue,gbg,0)
gFontYellow <- MakeFont(gPEBLBaseFontMono, 0, 24, gYellow,gbg,0)
## There are 4 blocks; color & name with colored targets;
## color & name with black targets.
gFileOut <- FileOpenWrite("vstroop-"+gSubNum+".txt")
gSumFileOut <- FileOpenWrite("vstroop-summary-"+gSubNum+".txt")
FilePrint(gFileOut,"subnum    block trial    word    color    part    xpos    ypos    resp    rname    correct
intrusion numresponses    time0    timea    timeend    trialtime    responsetime")
gTrial <- 0
gBlock <- 0
##This works by presenting a specific stimulus, and giving the option of four responses.
##The stimulus must be categorized either by color or by name.
##There are two control conditions for comparison purposes.
#####
## round 1 is "Part D": colored dots
stim1 <- PasteTables(ShuffleMe([0,0,0,0],gColors,1),
    Transpose([Repeat("d",reps*4)]))
#####
## Round 2 is "Part W": colored words
stim2 <- PasteTables(ShuffleMe(gWords,gColors,1),
    Transpose([Repeat("w",reps*4)]))
#####
## some trials where all colors are black
stim3 <- PasteTables(ShuffleMe(gColorNames,gColors,0),
    Transpose([Repeat("c",reps*4)]))
#####
## Begin experiment
gInstructions.text <- gInst1
Show(gInstructions)
Hide(gHeader)
Draw()
gInst1 <- LoadSound("gInst1.wav")
PlayBackground(gInst1)
Wait(24200)
Stop(gInst1)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
#Initialize data collectors
gErr <- []; gRT <- []; gCond <- []

```

```

gHead <- gHeadBlock
#####
##Do the Dot naming (block 1)
gBlock <- 1
out1 <- Block(stim1,1)
resp1 <- First(out1)
time1 <- Second(out1)
#####
##Do the word naming (block 2)
gInstructions.text <- gInst2
Show(gInstructions)
Hide(gHeader)
Draw()
gInst2 <- LoadSound("gInst2.wav")
PlayBackground(gInst2)
Wait(19600)
Stop(gInst2)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
gBlock <- 2
out2 <- Block(stim2,0)
resp2 <- First(out2)
time2 <- Second(out2)
#####
##Do the color naming (block 3)
gInstructions.text <- gInst3
Show(gInstructions)
Hide(gHeader)
Draw()
gInst3 <- LoadSound("gInst3.wav")
PlayBackground(gInst3)
Wait(17300)
Stop(gInst3)
WaitForAnyKeyPress()
Hide(gInstructions)
Show(gHeader)
Draw()
gBlock <- 3
out3 <- Block(stim3,0)
resp3 <- First(out3)
time3 <- Second(out3)
intrusions <- Third(out3)
text <- CR(3)+
  "Round      Time taken  Responses made  Intrusions"+CR(1)+
  "-----"+CR(1)+
  "Part D:      "+Format(Round(time1/10)/100,10) + "    " + resp1+ CR(1) +
  "Part W:      "+Format(Round(time2/10)/100,10) + "    " + resp2+CR(1) +
  "Part C:      "+Format(Round(time3/10)/100,10) + "    " + resp3+ "          " + intrusions+CR(1)+
  "Efficiency(C/D): "+Format((time3/time1),10) + "    " + resp3/resp1+CR(1)+
  "Efficiency(C/W): "+Format((time3/time2),10) + "    " + resp3/resp2
  FilePrint(gSumFileOut,"-----")
  FilePrint(gSumFileOut,"PEBL Victoria-Stroop Task, Version 1.0")
  FilePrint(gSumFileOut,"http://pebl.sf.net")
  FilePrint(gSumFileOut,"Shane T. Mueller, Arthur Kümmer, Suzan Such")
  FilePrint(gSumFileOut,GetPEBLVersion())
  FilePrint(gSumFileOut,"-----")
  FilePrint(gSumFileOut,"Response Order:"+gColors)
  FilePrint(gSumFileOut, "Subject code: "+gSubNum)
  FilePrint(gSumFileOut, "Time:      "+TimeStamp())
  FilePrint(gSumFileOut,"-----")
  FilePrint(gSumFileOut,text)
  FilePrint(gSumFileOut,"-----")
  filesum <- FileOpenAppend("stroop-v-all.csv")
  FilePrint(filesum, gSubNum + "    " , " + TimeStamp() + "    " , " + GetTime()+ "    " , " + time1+ "    " , " + resp1+ "    " , " + time2+ "    " ,
    "+resp2+ "    " , " + time3+ "    " , " + resp3+ "    " , " + intrusions)
#  MessageBox("This output only provided for testing.  Uncomment for deployment"+CR(1)+text,gWin)
  MessageBox(gExitText,gWin)
}
##
##
define PrintMe(file,text)
{
  FilePrint(file,text)
}

```

```

Print(text)
}
define PrintMe_(file,text)
{
  FilePrint_(file,text)
  Print_(text)
}
define PrintTable(file,table)
{
  PrintMe(file,"-----")
  loop(i,table)
  {
    loop(j,i)
    {
      PrintMe_(file,RoundOff(j,3) + Tab(1))
    }
    PrintMe(file,"")
  }
  PrintMe(file,"-----")
}
##The btype variable controls whether the block is a color or a word block.
## It should only affect scoring; it assumes stim is made correctly.
## btype == 1: by name
## btype == 2: by shade
## If colorresp!=0, then the responses are labeled with their
## corresponding colors.
define Block(stim,dotrainning)
{
  red   <- gFontBlack
  green <- gFontBlack
  blue  <- gFontBlack
  yellow <- gFontBlack
  foot1 <- MakeLabel("[1] "+First(gColorNames),red)
  foot2 <- MakeLabel("[2] "+Second(gColorNames),green)
  foot3 <- MakeLabel("[3] "+Third(gColorNames),blue)
  foot4 <- MakeLabel("[4] "+Fourth(gColorNames),yellow)
  feet <- [foot1,foot2,foot3,foot4]
  AddObject(foot1,gWin); AddObject(foot2,gWin);
  AddObject(foot3,gWin); AddObject(foot4,gWin);
  Move(foot1, gVideoWidth/2-300, gVideoHeight-100)
  Move(foot2, gVideoWidth/2-100, gVideoHeight-100)
  Move(foot3, gVideoWidth/2+100, gVideoHeight-100)
  Move(foot4, gVideoWidth/2+300, gVideoHeight-100)
  if(dotrainning)
  {
    ##Do the practice here:
    ##Lets do a little learning about the key mappings
    gInstructions.text <- gPracticeText
    Show(gInstructions)
    Hide(gHeader)
    Draw()
    gPracticeText<- LoadSound("gPracticeText.wav")
    PlayBackground(gPracticeText)
    Wait(24700)
    patch <- Rectangle(gVideoWidth/2,gVideoHeight/2,300,50,MakeColor("black"),1)
    AddObject(patch,gWin)
    Draw()
    resp <- ""
    while(not (resp == " "))
    {
      resp <- WaitForListKeyPress([1,2,3,4," "])
      ##translate response to a color
      colname <- Nth(gColors, ToNumber(resp))
      patch.color <- ChooseColor(colname)
      Draw()
    }
    RemoveObject(patch,gWin)
    Hide(gInstructions)
  }
  Show(gHeader)
  ##Create the grid layout.
  xypoints <- LayoutGrid(-20,gVideoWidth+20,
    80,450, 6,4,0)

  ##Wait at this point for the subject to press a key to begin.

```

```

gHeader.text <- gHeadBegin
Draw()
WaitForAnyKeyPress()
gHeader.text <- gHead
time1 <- GetTime()
#Add the highlight rectangle BEHIND the stimulus
highlight <- Rectangle(0,0,120,60,MakeColor("grey"),0)
AddObject(highlight,gWin)
##Add the stimuli
tmp <- []
complex <- PasteTables(stim,xypoints)
loop(id,complex)
{
  targ <- First(id)
  color <- Second(id)
  type <- Third(id)
  x <- Fourth(id)
  y <- Fifth(id)
  if(targ == 0)
  {
    stim <- Circle(x,y,20,ChooseColor(color),1)
  } else {
    stim <- MakeLabel(targ,ChooseFont(color))
    Move(stim,x,y)
  }
  AddObject(stim,gWin)
  tmp <- Append(tmp,stim)
}
##Create an 'error' highlight
highlight2 <- Rectangle(0,0,120,60,MakeColor("grey"),1)
AddObject(highlight2,gWin)
Hide(highlight2)
trial <- 1
responses <- 0
intrusions <- 0
##Collect responses
loop(id,complex)
{
  Move(highlight,Fourth(id),Fifth(id))
  Draw()

  corr <- 0
  time0 <- GetTime() #time 0 is start of trial
  while(not corr)
  {
    timea <- GetTime() #timea is start of last presentation
    resp <- WaitForListKeyPress([1,2,3,4])
    timeend <- GetTime() #time1 is end response
    responses <- responses + 1
    corr <- ScoreTrial(resp,"",Second(id),2)
    colname <- Nth(gColorNames, ToNumber(resp))
    if(not corr)
    {
      Move(highlight2,highlight.x,highlight.y)
      Show(highlight2)
      Draw()
      Wait(250)
      Hide(highlight2)
      Draw()
      ## If it was an error, was it a match to the word text?
      intrusion <- colname == First(id)
    } else{
      intrusion <- 0
    }
  }
  FilePrint(gFileOut,gSubNum+ " " + gBlock + " " + trial + " " + First(id) + " " + Second(id)+
    " " + Third(id) + " " + Fourth(id) + " " + Fifth(id)+ " " + resp + " " + colname + " " + corr + " "
    +intrusion + " " +responses + " " +
    time0+ " " + timea + " " + timeend + " " +(timeend-time0) + " " + (timeend-timea))
  intrusions <- intrusions + intrusion
}
trial <- trial + 1
}
time2 <- GetTime()
return [responses,time2-time1,intrusions]
}

```

```

##for now, assumes the tables are the same size
define PasteTables(tab1,tab2)
{
  a <- Transpose(tab1)
  b <- Transpose(tab2)
  return Transpose(Merge(a,b))
}
define ChooseColor(name)
{
  if(name == "black") {col <- gBlack}
  if(name == "blue") {col <- gBlue}
  if(name == "red") {col <- gRed}
  if(name == "green") {col <- gGreen}
  if(name == "yellow"){col <- gYellow}
  return col
}
define ChooseFont(name)
{
  if(name == "black") {font <- gFontBlack}
  if(name == "blue") {font <- gFontBlue}
  if(name == "red") {font <- gFontRed}
  if(name == "green") {font <- gFontGreen}
  if(name == "yellow"){font <- gFontYellow}
  return font
}
define LayoutGrid(xmin,xmax,ymin,ymax, wide,tall, horvert)
{
  xskip <- (xmax-xmin)/(wide+1)
  yskip <- (ymax-ymin)/(tall+1)
  pts <- []
  if(horvert == 1)
  {
    x <- xmin+xskip
    loop(i,Sequence(1,wide,1))
    {
      y <- ymin+yskip
      loop(j,Sequence(1,tall,1))
      {
        pts <- Append(pts, [x,y])
        y <- y + yskip
      }
      x <- x + xskip
    }
  } else {
    y <- ymin+yskip
    loop(i,Sequence(1,tall,1))
    {
      x <- xmin+xskip
      loop(j,Sequence(1,wide,1))
      {
        pts <- Append(pts, [x,y])
        x <- x + xskip
      }
      y <- y + yskip
    }
  }
  return pts
}
##depending upon condition,
define ScoreTrial(resp,wordname,colname,btype)
{
  ## responses are: 1.r e d 2.gre en 3.bl u e 4.yellow
  corr <- 0
  ##There are really just two ways of scoring:
  ## by color and by name.
  if(btype==1)
  {
    if( Nth(gColors, ToNumber(resp)) ==wordname)
    {
      corr <- 1
    }
  } else {
    if( Nth(gColors, ToNumber(resp)) ==colname)
    {
      corr <- 1
    }
  }
}

```

```

    }
  }
  return corr
}
## This is a standard initializer function that sets up typical
## objects used in experiments.
define Initialize()
{
  gVideoWidth <- 800
  gVideoHeight <- 600
  ##Initialize Window, etc.
  gWin <- MakeWindow("grey40")
  ShowCursor(0)
  ##Initialize Font and colors
  gBG <- MakeColor("grey40")
  fg <- MakeColor("black")
  gInstructionsFont <- MakeFont(gPEBLBaseFont,0,18, fg,MakeColor("white"),1)
  gHeaderFont <- MakeFont(gPEBLBaseFont, 0,36,fg,gBG,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)
  Show(gInstructions)
  ## Make and hide a header label
  gHeader <- MakeLabel("",gHeaderFont)
  AddObject(gHeader,gWin)
  Move(gHeader, gVideoWidth/2, 50)
  Hide(gHeader)
  gStimLabel <- MakeLabel("",gHeaderFont)
  AddObject(gStimLabel,gWin)
  Move(gStimLabel, gVideoWidth/2, gVideoHeight/2)
  Hide(gStimLabel)
  gFeedback <- MakeLabel("INCORRECT",gHeaderFont)
  AddObject(gFeedback,gWin)
  Move(gFeedback, gVideoWidth/2, gVideoHeight/2+70)
  Hide(gFeedback)
  if(Uppercase(gLanguage)=="PT") {
    gColorBaseNames <- ["vermelho"," verde ", " azul ", "amarelo "]
    gWords <- ["cada ", "hoje ", "nunca", "tudo "]
    gPracticeText <- "Antes de começar, aprenda como nomear as cores, relacionando os números do teclado com as cores escritas abaixo na
tela. Pressione os números 1-2-3-4 e veja as respectivas cores no retângulo. Após treinar, pressione a barra de espaço para começar o teste."
    gTreino<-"Treino"
    gInst1 <- "Você está prestes a iniciar um teste no qual deverá nomear as cores dos círculos o mais rápido que você puder. Siga o retângulo
cinza e pressione a tecla correta [1-2-3-4] apropriadamente.
      Pressione qualquer tecla para continuar."
    gInst2 <- "Dessa vez, nomeie as cores das palavras o mais rápido que puder. Siga o retângulo cinza e pressione a tecla correta [1-2-3-4]
apropriadamente.
      Pressione qualquer tecla para começar."
    gInst3 <- "Novamente, nomeie as cores nas quais as palavras estão impressas o mais rápido que puder. Não leia a palavra! Em vez disso,
nomeie a cor na qual a palavra foi impressa.
      Pressione qualquer tecla para começar."
    gHeadBegin <- "Pressione qualquer tecla para começar"
    gHeadBlock <- "Pressione a tecla para indicar a cor."
    gExitText <- "Você terminou. Obrigado por participar."
  } else {
    ##Default to ENGLISH
    gColorBaseNames <- ["red","green","blue","yellow"]
    gWords <- ["when","hard","and","over"]
    gPracticeText <- "Before we begin, take a moment to learn the mapping between colors and the keyboard numbers. Press the numbers 1-4
consecutively to practice the stimuli, and space bar to begin the test."
    gTreino<-"Treino"
    gInst1 <- "You are about to be tested regarding how well you can read words and the colors they appear in. In the first part, you must name
the colors of the dots as quickly as you can. Follow the gray rectangle and press the correct key [1-4] accordingly.
      Press any key to begin."
    gInst2 <- "This time, name the colors of the words as quickly as you can. Follow the gray rectangle and press the correct key [1-4]
accordingly.
      Press any key to begin."
    gInst3 <- "Again, name the colors in which the words are printed as quickly as you can. Don't read the word! Instead, name the color in
which the word is printed.
      Press any key to begin."
    gHeadBegin <- "Press any key to begin"
    gHeadBlock <- "Press key to indicate color."
    gExitText <- "You are now finished. Thank you for participating."
  }
}

```

```

## This finds mean and SD RT for
## each of the conditions specified in delays
##
define Aggregate(vals, cond)
{
  ## start by sorting values by condition,
  ## then move through them and analyze subparts.
  valX <- SortBy(vals,cond)
  condX <- Sort(cond)
  trials <- Transpose([condX,valX])
  stats <- []
  lastVal <- First(condX)
  tmpVals <- []
  loop(i, trials)
  {
    ## if the current delay differs from the previous delay,
    ## we should analyze what is in tmp right now.
    if(First(i) != lastVal)
    {
      stats <- Append(stats,
        [lastVal, Length(tmpVals), Median(tmpVals),
          Mean(tmpVals), StdDev(tmpVals)])
      tmpVals <- []
    }
    lastVal <- First(i)
    tmpVals <- Append(tmpVals, Nth(i, 2))
  }
  stats <- Append(stats,
    [lastVal, Length(tmpVals),
      Median(tmpVals), Mean(tmpVals), StdDev(tmpVals)])
  return stats
}
define RoundOff(val,sig)
{
  s <- 10 ^ sig
  return Round( val *s)/s
}
##Finds difference between two lists.
define ListDiff(l1,l2)
{
  tmp <- Transpose([l1,l2])
  new <- []
  loop(i,tmp)
  {
    new <- Append(new,First(i)-Second(i))
  }
  return new
}
##This creates a stimulus deck with the special
## constraints of vic-stroop. It creates a balanced list
## 4x6=24 items long of pairs of words and colors
define ShuffleMe(words, colors, matchesOK)
{
  ##If matchesOK==1, then matching is OK (blocks 1 and 2)
  ##In that case, we can just create individual lists 24-long and shuffle.
  if(matchesOK)
  {
    ##Create baseline shuffled templates
    basewords <- RepeatEnough([1,2,3,4],24)
    basecols <- RepeatEnough([1,2,3,4],24)
    outwords <- Replace(basewords, Transpose([1,2,3,4],words))
    outcols <- Replace(basecols, Transpose([1,2,3,4],colors))
    outlist <- Transpose([outwords,outcols])
  } else {
    list <- []
    i <- 1
    while(i <= 6)
    {
      ##we need to be sure words and colors do not match
      matchany <- 1
      while(matchany)
      {
        trywords <- Shuffle([1,2,3,4])
        trycols <- Shuffle([1,2,3,4])
        matchany <- MatchAny(trywords,trycols)
      }
    }
  }
}

```

```

    }
    out <- Transpose([trywords,trycols])
    ##Make sure the transition doesn't repeat
    if(i>1)
    {
        lastpair <- Last(list)
        lastword <- First(lastpair)
        lastcol <- Second(lastpair)
        ##Make sure the transition between sublists is good.
        badtransition <- 1
        while(badtransition)
        {
            out <- Rotate(out,1)
            nextpair <- First(out)
            nextword <- First(nextpair)
            nextcol <- Second(nextpair)
            badtransition <- (nextword==lastword) or (nextcol==lastcol)
        }
    }
    list <- Merge(list,out)
    i <- i + 1
}
##list just contains numbers. Recode into the proper labels
tmp <- Transpose(list)
outwords <- Replace(First(tmp), Transpose([[1,2,3,4],words]))
outcols <- Replace(Second(tmp), Transpose([[1,2,3,4],colors]))
outlist <- Transpose([outwords,outcols])
}
return outlist
}
define RepeatEnough(list,length)
{
    l <- Length(list)
    times <- Ceiling(length/l)
    tmp <- Flatten(ShuffleRepeat(list,times))
    return SubList(tmp,1,length)
}
define MatchAny(list1,list2)
{
    match <- 0
    loop(i,Transpose([list1,list2]))
    {
        if(First(i)==Second(i))
        {
            match <- 1
            if(match)
            {
                break
            }
        }
    }
    return match
}
##Note that shufflerepeat ensures that there are no
##duplicates at transitions between lists. This could lead
##to infinite loops if you shuffle a list of identical items
define ShuffleRepeat(list, n)
{
    ##Shuffle the first copy:
    tmp <- [Shuffle(list)]
    ##Add additional sets to the end
    if(n>1)
    {
        loop(i, Sequence(1,n-1,1))
        {
            try <- Shuffle(list)
            while(Last(Last(tmp)) == First(try))
            {
                try <- Shuffle(list)
            }
            tmp <- Append(tmp,try)
        }
    }
    return tmp
}

```

ANEXO B- Programação Teste Torre de Londres

```
#####
##
## PEBL's Tower of London Test
##
## Version 0.5
## Released August 2010
##
##
## see
## Shallice T. (1982), Philosophical Transactions of the Royal Society of
## London, B, 298, 199-209.
##
##
## Problem sets available from:
## Fimbel, E., Lauzon, S., & Rainville, C. (2009). Performance of
## Humans vs. Exploration Algorithms on the Tower of London Test.
## (J. Bongard, Ed.) PLoS ONE, 4(9),
## e7263. doi:10.1371/journal.pone.0007263
## Phillips, L. H., Wynn, V., Gilhooly, K. J., Della Sala, S., &
## Logie, R. H. (1999). The role of memory in the Tower of London
## task. Memory, 7(2), 209-231.
##
## Schnirman, G. M., Welsh, M. C., & Retzlaff, P. D. (1998).
## Development of the Tower of London-Revised. Assessment,
## 5(4), 355-360. Retrieved from
##
##
## Special thanks to Adam Thomas of NIH/NIMH for implementing the
## TOL-R trials.
define Start(p)
{
## Basic translations strings are assigned in GetStrings (see bottom of file).
##
##
## gLanguage defaults to 'en', but can be set at command line.
## using the --language flag.
gLanguage <- "BR"
GetStrings(gLanguage)
gShowHand <- 1 #Show the hand graphic?
Init()
gInstructions <- MakeTextBox("", gFont, 760, 560)
AddObject(gInstructions, gWin)
Move(gInstructions, 20, 20)
Draw()
type <- "3"
stim <- Shallice()
#####
##
##
## 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?
#####
gInstructions.text <- gInstText1
if(gUseTimer)
{
gInstructions.text <- gInstructions.text + gInstText2
}
if(gUseTurnLimit)
{
gInstructions.text <- gInstructions.text + gInstText3
}
gInstructions.text <- gInstructions.text + CR(1) + gBegin
gInstructions.font <- gFont2
```

```

Draw()
instru01t <- LoadSound("instru01t.wav")
PlayBackground(instru01t)
Wait(83800)
Stop(instru01t)
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(4800)
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(5200)
Stop(gSeeDemo2)
WaitForDownClick()
#####
gInstructions.text <- gContinue
Hide(gInstructions)
Draw()
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
    Hide(gInstructions)
    Show(gInstructions)
    Draw()
    gReset <- 0
  } else {
    gInstructions.text <- gIncomplete
    Hide(gInstructions)
    Show(gInstructions)
    Draw()
    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)
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,y+110,300,40,MakeColor("grey20"),0)
AddObject(r0,gWin)
gZones <- [r0,r1,r2,r3]
gResetLabel <- MakeLabel(gResetText,footerfont)
Move(gResetLabel,340,y+110)
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)
gFooter <- MakeLabel(gFooterLab,footerfont)
AddObject(gFooter,gWin)
Move(gFooter,300,490)
gHeader <- MakeTextBox(gHeaderLab,footerfont,105,95)
AddObject(gHeader,gWin)
Move(gHeader,30,50)
##Keep track of the image complex:
gComplex <- []
gFileOut <- FileOpenAppend("tol-"+gsubnum+".txt")
gFileOutSum <- FileOpenAppend("tol-summary-"+gsubnum+".txt")
}
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. wierd
               [[[],[3,5],[1,4,2]], [[3,5],[2],[1,4]] ,-1,0,0], #original wierd
               [[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 wierd

  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 wierd
               [[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 wierd
               [[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 wierd
               [[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],[ ]],3,63,53],
  [[2,1],[3],[ ]],[[2],[1,3],[ ]],3,63,54],
  [[2,1],[3],[ ]],[[1,2],[3],[ ]],4,63,13],
  [[2,1],[3],[ ]],[[3],[1,2],[ ]],4,63,44],
  [[2,1],[3],[ ]],[[1,3],[ ],[2]],4,63,22],
  [[2,1],[3],[ ]],[[ ],[1,3],[2]],4,63,46],
  [[2,1],[3],[ ]],[[1,2,3],[ ],[ ]],5,63,11],
  [[2,1],[3],[ ]],[[1,3,2],[ ],[ ]],5,63,21],
  [[2,1],[3],[ ]],[[3,2],[1],[ ]],5,63,43],
  [[2,1],[3],[ ]],[[3],[1],[2]],5,63,45]]
return s
}
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/tolSPACE/
##
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,
2,2,1,0,1,2,3,2,3,4,3,4,7,7,6,5,6,5,8,7,8,7,7,6,4,3,4,5,5,6,4,4,3,3,2,1,
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,
2,1,2,3,3,4,2,2,1,1,0,1,4,4,3,2,3,2,5,4,5,6,5,6,7,7,8,7,7,7,6,5,5,4,
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,
5,4,5,6,5,6,4,4,3,2,3,2,2,2,1,1,0,1,2,1,2,3,3,4,7,7,6,5,5,4,7,6,7,8,7,
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,
7,6,7,8,7,8,6,6,5,4,5,4,4,3,3,3,2,3,0,1,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,
6,5,6,7,6,7,5,5,4,3,4,3,3,3,2,2,1,2,1,0,1,2,2,3,6,6,5,4,4,3,6,5,6,7,7,8,
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,
7,6,7,8,7,7,7,6,5,5,4,2,1,2,3,3,4,2,2,1,1,0,1,4,4,3,2,3,2,5,4,5,6,5,6,
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,
7,7,6,5,5,4,7,6,7,8,7,5,4,5,6,5,6,4,4,3,2,3,2,2,2,1,1,0,1,2,1,2,3,3,4,
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,
6,6,5,4,4,3,6,5,6,7,7,8,6,5,6,7,6,7,5,4,3,4,3,3,3,2,2,1,2,1,0,1,2,2,3,
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,
4,4,3,3,2,1,4,3,4,5,5,6,8,7,8,7,7,6,7,7,6,5,6,5,3,2,3,4,3,4,2,2,1,0,1,2,
4,4,3,2,3,2,5,4,5,6,5,6,7,6,7,8,7,7,7,6,5,5,4,2,1,2,3,3,4,2,2,1,1,0,1,
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],[],[1,2]],[[[],[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()
  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]],
        [[[],[],[1,3,2]],[[2],[1,3]],[[],[2],[1,3]],[[],[2,3],[1]],[[3],[2],[1]],[[3],[2,1],[1]],
        [[[],[],[3,1,2]],[[2],[1,3],[1]],[[],[2],[3,1]],[[],[2,1],[3]],[[1],[2],[3]],[[1],[2,3],[1]],
        [[[],[],[3,2,1]],[[1],[1,3],[2]],[[],[1],[3,2]],[[],[1,2],[3]],[[2],[1],[3]],[[2],[1,3],[1]],
        [[[],[],[2,3,1]],[[1],[1,2],[3]],[[],[1],[2,3]],[[],[1,3],[2]],[[3],[1],[2]],[[3],[1,2],[1]],
        [[[],[],[2,1,3]],[[3],[1,2],[1]],[[],[3],[2,1]],[[],[3,1],[2]],[[1],[3],[2]],[[1],[3,2],[1]] ]
    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 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>")
        {
          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]],
[[[],[],[1,3,2]],[[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()
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, 30 trials"+CR(2)+
      "[2] Unconstrained pile heights, {3,4,5} disks, Random presentation, 30 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(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"
    gSeeDemo <- "Click mouse button to see demo problem"
    gSeeDemo2 <- "Click mouse button to continue to the first problem"
    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, 30 problemas"+CR(2)+
"[2] Pilhas de alturas não-limitadas, {3,4,5} discos, apresentação aleatória, 30 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(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 de sua
configuração original para a configuração 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 (indicado pelo tamanho do retângulo cinza). 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."
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, indicado pela barra que aparecerá a
direita da tela. 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 no mouse para ver a demonstração."
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, 30 trials"+CR(2)+
"[2] Unconstrained pile heights, {3,4,5} disks, Random presentation, 30 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(3)+
"Edit TOL.pbl directly to specify a test configuration automatically."
gHeaderLab <- "Objetivo montóns:"
gFooterLab <- "Haga clic en pila para recoger y llevar a disco"
gSeeDemo <- "Click mouse button to see demo problem"
gSeeDemo2 <- "Click mouse button to continue to the first problem"
gInstText1 <- "Estás a punto de realizar una tarea llamada 'Torre de Su Londres'. Objetivo es pasar un montón de discos desde su
configuración original a la configuración mostrada en la parte superior de la pantalla. Sólo puedes mover un disco a la vez, y no se puede
mover un disco sobre un montón que no tiene más espacio (indicado por el tamaño del rectángulo gris). Para mover un disco, haga clic en el

```

montón que desea mover un disco fuera de, y se moverá hacia arriba por encima de la pila. A continuación, haga clic en otra pila, y el disco se moverá hasta que el montón. "

```
gInstText2 <- " Tendrá un límite de tiempo para terminar cada problema. Intento de terminar cada problema en el tiempo asignado."
```

```
gInstText3 <- " Usted tendrá solamente un número 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 límite de movimiento. Si no termina el problema en el límite, el a su vez terminará y se trasladará al siguiente problema."
```

```
gBegin <- "Haga clic en el ratón para empezar."
```

```
gContinue <- "Prueba Completa. "+ CR(1) +" Haga clic en el ratón para continuar a la siguiente prueba. "
```

```
gIncomplete <- "Prueba Incompleta. "
```

```
gClickLab <- "Clics"
```

```
gTimeLab <- "Tiempo (s)"
```

```
gMovesOutLab <- "Usted no pudo completar el problema del número de movimientos necesarios."
```

```
gResetText <- "Reset problem"
```

```
gResetLab <- "You are starting the problem again."
```

```
gTimeOutLab <- "Usted acabó el tiempo."
```

```
gDebriefing <- "Gracias por participar. El experimento ya está completa. Pulse cualquier tecla para salir."
```

```
gContinueFail <- " Haga clic en el ratón para continuar a la siguiente prueba. "
```

```
} else{
```

```
  Print("Language: ["+lang+"] unknown. Falling back on English translations.")
```

```
  GetStrings("en")
```

```
}
```

```
}
```

ANEXO C- Programação Teste Wisconsin de Classificação de Cartas-64

```
#!/usr/local/bin/pebl
#####
## 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.
  gLanguage <- "pt"
  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
  gFileOut <-FileOpenWrite("bcst64-" + gSubNum + ".txt")
  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()
  decksize <- 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()
instru10 <- LoadSound("instru10.wav")
PlayBackground(instru10)
Wait(23700)
Stop(instru10)
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(31800)
Stop(instru02)
WaitForIt()
gInstructions.text <- Nth(textlist,3)
Draw()
instru03 <- LoadSound("instru03.wav")
PlayBackground(instru03)
Wait(17000)
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()
    correto2 <- LoadSound("correto2.wav")
    PlayBackground(correto2)
    Wait(700)
  Stop(correto2)
} else {
  ##It is incorrect.
  feedback <- gIncorrect
  Draw()
    errado2 <- LoadSound("errado2.wav")
    PlayBackground(errado2)
    Wait(800)
    Stop(errado2)
  ##Increment failure to maintain if run >= 5
  if(run >= 5)
  {
    failure to maintain <- failure to maintain + 1
  }
  error on cat <- error on cat + 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 + " " + trialperseverr + " " + 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)
  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 >= 9 )
{
  numCatsExp <- 9
  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)
Print(trialspercat)
Print(erroroncat)
#####
##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,errorspercat]))
    {

```

```

##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("bcst64-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. Categories Completed/Experienced: "+ numCats + " / " + numCatsExp)
FilePrint(report,"2. Number of Trials:          "+ num)
FilePrint(report,"-----")
FilePrint(report,"          Total Percent Mean/Cat")
FilePrint(report,"3. Correct Responses:          "+ totCorr + Tab(1) + Roundoff(totCorr/num,4)*100 + Tab(1) +
Roundoff(totCorr/numcatsExp,3))
FilePrint(report,"4. Total Errors:          "+ err + Tab(1) + Roundoff(err/num,4)*100 + Tab(1) + Roundoff(err/numCatsExp,3))
FilePrint(report,"5. Perseverative Responses          "+ totPers + Tab(1) + Roundoff(totPers/num,4)*100 + Tab(1) +
Roundoff(totPers/(numCatsExp-1),3))
FilePrint(report,"6. Perseverative Errors          "+ totPersErr + Tab(1) + Roundoff(totPersErr/num,4)*100 + Tab(1) +
Roundoff(totPersErr/(numCatsExp-1),3))
FilePrint(report,"7. Non-Perseverative Errors          "+ 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. Trials to complete 1st cat: "+ firstcat)
FilePrint(report,"9. Failure to maintain set: "+failuretomaintain)
FilePrint(report,"10. Learning to Learn:          "+learningtolearn)

```

```

FilePrint(report,"11. Conceptual Level Responses: "+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()
gInstructions.text <- Nth(textlist,4)
Draw()
instru04 <- LoadSound("instru04.wav")
PlayBackground(instru04)
Wait(9500)
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
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"]
return cards
}

```

ANEXO D- Solicitação para a escola

Solicitação para a escola

Belo Horizonte, 17 de fevereiro de 2011.

Sra. Rosana Mol Lana

Diretora da Escola Estadual Afonso Pena
Avenida João Pinheiro, 450
Bairro Funcionários - Belo Horizonte/MG

Prezada Diretora,

Solicitamos sua autorização para realizarmos, na Escola Estadual Afonso Pena, parte da pesquisa intitulada **“Analogia funcional entre as versões manuais e computadorizadas dos testes Torre de Londres (TOL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST)”**.

Participação da pesquisa:

- **Crianças entre 9 e 10 anos, que não tenham participado de outras pesquisas que envolvam os testes aplicados.**
- **Serão incluídas crianças de ambos os sexos, independente de cor ou nível sócio-econômico.**

Os responsáveis por essa pesquisa são o Dr. Arthur Melo e Kummer, professor do Departamento de Saúde Mental da Faculdade de Medicina da UFMG e Suzan Caroline Such Ribeiro, psicóloga e aluna da especialização de Neurociências da UFMG.

O objetivo deste estudo é verificar se há semelhança entre as versões manual e computadorizada dos testes Torre de Londres (ToL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST), que são importantes para avaliar as funções executivas. Essas funções são consideradas essenciais para gerenciar o comportamento humano e, por isso, é parte fundamental da avaliação neuropsicológica de pessoas normais ou com problemas cognitivos ou comportamentais.

A pesquisa consiste no preenchimento de um formulário pelos pais e aluno e um exame do aluno para a avaliação se ele/ela preenche os critérios para participar da pesquisa. O exame consistirá em duas sessões com duração de 40 minutos, em média, nas quais serão aplicados um questionário e os testes Torre de Londres (ToL), Teste de Stroop, Teste Wisconsin de Classificação de Cartas (WCST) e Teste das Matrizes Progressivas de Raven. As informações obtidas nesta pesquisa contribuirão para

estudos futuros de normatização das versões computadorizadas destes testes na população brasileira, ampliando, assim, disponibilidade de testes mais eficazes para avaliação e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos.

Todos os dados coletados serão arquivados e poderão ser utilizados nesta pesquisa, em divulgações científicas e publicações em revistas da área de saúde, psicologia e educação. As crianças não serão identificadas e os responsáveis pela pesquisa se comprometem a manter sigilo sobre a identidade das pessoas envolvidas e sobre as informações que possam identificá-las, assim como a cumprir os demais requisitos éticos, de acordo com o previsto em lei.

Esclarecemos que a participação na pesquisa é voluntária e os pais podem se recusar a participar ou podem retirar seu consentimento, quando desejarem ou precisarem, sem nenhum prejuízo ou penalidade. Se concordarem que a criança participe desse estudo, as avaliações serão agendadas com antecedência e realizadas na escola. Será necessário, também, que os pais assinem um Termo de Consentimento.

Serão avaliadas em torno de 20 crianças, a partir do mês de março de 2011, em dia e horário a ser combinado, durante um seguimento de aproximadamente três meses. Necessitaremos de um local para aplicarmos os testes, um pouco distante das atividades das demais crianças da escola.

Agradecemos sua colaboração e nos colocamos à disposição para qualquer informação adicional.

Atenciosamente,

Psicóloga Suzan Caroline Such Ribeiro
CRP 04/30382

Prof. Dr Arthur Melo e Kummer
RG: 1.424.287 SSP/AL

Suzan Caroline Such Ribeiro
Av. Flávio dos Santos, 330 ap. 201 – Bairro Floresta – BH/MG
Tel.: (31) 2552-6164 / (31) 9881-1028

Prof. Dr Arthur Melo e Kummer
Departamento de Saúde Mental da Faculdade de Medicina da UFMG
Telefone: (31) 3909-9786
Belo Horizonte, 17 de fevereiro de 2011.

Psicóloga Suzan Caroline Such Ribeiro
Prof. Dr. Arthur Melo e Kummer

Prezados pesquisadores,

Confirmo-lhes meu acordo para realizarem avaliações/testes na Escola Estadual Afonso Pena, em crianças entre nove e dez anos de idade, como parte da pesquisa “**Analogia funcional entre as versões manuais e computadorizadas dos testes Torre de Londres (ToL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST)**”. Colocaremos à disposição dos senhores um local apropriado para essas avaliações. Os professores serão comunicados e divulgaremos a pesquisa para os pais.

Atenciosamente,

Rosana Mol Lana
Diretora da Escola Estadual Afonso Pena

Escola Estadual Afonso Pena
Avenida João Pinheiro, 450
Bairro Funcionários - Belo Horizonte/MG
Tel.: (31) 3226-3764

ANEXO E- Termo de Consentimento Livre e Esclarecido**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**

(Para pais)

Analogia funcional entre as versões manuais e computadorizadas dos testes Torre de Londres (TOL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST)

Seu/sua filho/a está sendo convidado a participar da pesquisa intitulada “Analogia funcional entre as versões manuais e computadorizadas dos testes Torre de Londres (ToL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST)”. Antes de aceitar participar desta pesquisa clínica, 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, e 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 dadas para esclarecer quaisquer dúvidas sobre a pesquisa proposta, antes de você aceitar o convite para participar.

O objetivo deste estudo é verificar se há semelhança entre as versões manual e computadorizada dos testes Torre de Londres (ToL), Teste de Stroop e Teste Wisconsin de Classificação de Cartas (WCST), que são importantes para avaliar as funções executivas. Essas funções são consideradas essenciais para gerenciar o comportamento humano e, por isso, é parte fundamental da avaliação neuropsicológica de pessoas normais ou com problemas cognitivos ou comportamentais.

Este estudo consiste no preenchimento de um formulário por você e seu/sua filho/a e um exame de seu/sua filho/a para a avaliação se ele/ela preenche os critérios para participar da pesquisa. O trabalho estuda crianças de 9 e 10 anos. O exame com seu/sua filho/a deverá durar 1 (uma) hora e 30 (trinta) minutos e este tempo poderá ser dividido em dois dias.

As informações obtidas nesta pesquisa contribuirão para estudos futuros de normatização das versões computadorizadas destes testes na população brasileira, ampliando, assim, disponibilidade de testes mais eficazes para avaliação e acompanhamento de diversos déficits e transtornos comportamentais e psiquiátricos.

Os riscos que essa pesquisa pode causar aos participantes são apenas a possibilidade de algum constrangimento em virtude das questões propostas, 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 o identificam pelo nome. É importante informar que qualquer publicação dos dados não o identificará. Ao assinar este formulário de consentimento, você autoriza o pesquisador a fornecer os registros obtidos no estudo 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 poderá interromper sua participação a qualquer momento sem afetar seu acompanhamento médico em andamento. Você não receberá qualquer compensação financeira por sua participação no estudo.

Durante o estudo, se você tiver qualquer dúvida, contate o Dr. Arthur Melo e Kummer pelo telefone 3409-9785, ou 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. Tive a oportunidade de fazer perguntas e todas as minhas dúvidas foram respondidas a contento. Este formulário está sendo assinado voluntariamente por mim e meu/minha filho/a, indicando o nosso consentimento para ele/ela participar do estudo, até que nós decidamos o contrário.

Belo Horizonte, ____ de _____ de ____.

Assinatura do responsável pelo participante_____

Assinatura do(a) aluno(a)_____

Assinatura do pesquisador_____

ANEXO F – Formulário para pais

Formulário:

Nome da criança: _____
 Data de nascimento: __/__/__ Idade: _____ Sexo: _____
 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:

- Toma algum remédio controlado? () não () sim- Qual(is): _____

- A criança já fez algum tratamento especializado? () não () sim

Fisioterápico () Fonoaudiólogo () Terapeuta Ocupacional () Psicológico

() Neurológico () Ortopédico () Outros

() Tipo/motivo: _____

Dados sócio-demográficos: (marque um X na opção de acordo com o que possui, por exemplo, se tem uma televisão na casa, marque um X na coluna “Um”)

O(A) senhor(a) tem em sua casa...	Quantidade de itens				
	Não tem	Um	Dois	Três	4 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?

- () Analfabeto/ Primário incompleto () Primário completo/ ginasial incompleto
 () Ginasial completo/ Colegial incompleto () Colegial completo/ Superior incompleto
 () Superior Completo

Perfil Familiar:

Anotar o primeiro nome de cada pessoa que mora na mesma casa que o participante, o sexo (F=feminino,M= masculino), grau de parentesco com o participante (pai, mãe, irmão, etc), Idade, formação (até que série estudou).

Moradores	Sexo	Parentesco	Idade	Formação
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			
	()F () M			

Agradecemos sua colaboração. Por gentileza coloque este formulário e o Termo de Consentimento devidamente assinado e datado no envelope e envie pelo(a) seu(sua) filho(a) para a escola.

Atenciosamente,

Suzan Caroline Such Ribeiro
Psicóloga CRP04/30382

ANEXO G- Protocolo de Aplicação

Protocolo de Aplicação

Questionário elaborado com base no “Questionário TIC Domicílios”

Data de aplicação: __/__/__ Entrevistador: _____

Hora de início: __:__:__ Hora de término: __:__:__

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 () Não sabe () Não respondeu
2. Você gosta de computador? [0- não gosta, 10- gosta muito] _____
3. Com que idade utilizou computador pela primeira vez? _____
4. Tem acesso a um computador em casa? () Sim () Não
5. Em qual desses locais você utilizou o computador com mais frequência?

Você utilizou o computador...	8. Usou-RM		8A- mais frequente RU
	Sim	Não	
Em casa			
No trabalho (quando fora de casa)			
Na escola (ou estabelecimento de ensino)			
Na casa de outra pessoa (amigo, vizinho ou familiar)			
Centro público de acesso gratuito (telecentro, biblioteca, entidade comunitária, Correios etc.)			
Centro público de acesso pago (Internet Café, lanhouse, hotspot)			
Em algum outro lugar? _____			

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. Quais das seguintes atividades ligadas ao computador você já executou?

	Sim	Não
Usar um <i>mouse</i>		
Copiar ou mover um arquivo ou uma pasta		
Usar um editor de texto (Word)		
Usar uma planilha de cálculo (Excel)		
Comprimir arquivos no computador (WinZip)		
Escrever um programa de computador usando alguma linguagem de programação		
Conectar ou Instalar periféricos (impressora, câmera, microfone)		
Usar programas de som e imagem/multimídia		
Abrir um programa para navegar na Internet (Internet Explorer, Mozilla, Firefox)		

8. Você tem aulas de informática na escola? () sim () não
a) A partir de qual série: _____ b) Qual a carga horária: _____ (aulas/hora)
9. Você utiliza o computador para trabalhos acadêmicos? () sim () não
10. Você já utilizou a Internet? () Sim () Não () Não sabe () Não respondeu
11. Em média, com que frequência você usou a Internet nos últimos 3 meses?

() Todos os dias ou quase todos os dias
semana

() Pelo menos uma vez por

() Pelo menos uma vez por mês

() Menos de uma vez por mês

12. Nos últimos 3 meses, quantas horas aproximadamente você gastou na Internet/semana?

		hora(s) (espontânea)	
	Menos de 1h por semana		Mais de 20h até 30h por semana
	Entre 1h e 5h por semana		Mais de 30h até 40h por semana
	Mais de 5h até 10h por semana		Mais de 40h por semana
	Mais de 10h até 15h por semana		Não sabe
	Mais de 15h até 20h por semana		Não respondeu

13. Quando você costuma utilizar a internet na maioria das vezes?

() durante a semana () fins de semana () manhã () tarde () noite

14. Quais das seguintes atividades ligadas à Internet você já executou?

	Sim	Não
Usar um mecanismo de busca para achar informação	☐	☐
Enviar e-mails com arquivos anexados (documentos, fotos)	☐	☐
Enviar mensagens em salas de bate-papo e fóruns de discussão online	☐	☐
Usar a Internet para realizar ligações telefônicas	☐	☐
Trocar mensagens de voz, usando programas - Skype/videoconferência	☐	☐
Usar um programa de compartilhamento de arquivos para trocar filmes, música etc.	☐	☐
Criar uma página na Internet	☐	☐
Baixar/ fazer download arquivos	☐	☐
Baixar e instalar softwares	☐	☐
Participar de sites de relacionamento (Orkut, Facebook, Linked IN)	☐	☐
Criar ou atualizar blogs e/ou páginas na Internet (websites)	☐	☐
Jogar jogos online (conectados à Internet)	☐	☐
Assistir a filmes ou vídeos (como os do Youtube)	☐	☐
Assistir televisão (em tempo real)	☐	☐
Outra atividade _____	☐	☐

15. Qual a principal razão para não ter procurado aprimorar suas habilidades com o computador?

☐	Falta de oportunidade	☐	Os cursos são muito difíceis
☐	Desconhecimento de como fazê-lo	☐	Tenho conhecimento suficiente
☐	Falta de tempo	☐	Não sabe
☐	Custo do curso	☐	Nenhum dos motivos anteriores
☐	Nenhuma oferta adequada	☐	

Entretenimento:

- Você já jogou vídeo-game? () sim () não

Qual: () Playstation () X-box () Nintendo () Outro: _____

- Você possui algum vídeo-game? () sim () não

Qual: () Playstation () X-box () Nintendo () Outro: _____

- Quais tipos de jogos costuma jogar:

() ação () RPG () futebol () corrida () musicais () aventura

() Outro: _____

Quantas horas costuma jogar por semana:_____ hora(s) (espontânea)			
☐	Menos de 1h por semana	☐	Mais de 20h até 30h por semana
☐	Entre 1h e 5h por semana	☐	Mais de 30h até 40h por semana
☐	Mais de 5h até 10h por semana	☐	Mais de 40h por semana
☐	Mais de 10h até 15h por semana	☐	Não sabe
☐	Mais de 15h até 20h por semana	☐	Não respondeu

TESTE DA TORRE DE LONDRES

Data de aplicação: __/__/__ Hora de início: __:__:__ Hora de término: __:__:__

Examinador: _____ Sujeito: _____

Problem a	No. mov	Tempo	Resposta correta	Tentativas	Pontos
1	2		Vm1/Vd2	1. _____ 2. _____ 3. _____	
2	2		A1/Vm2	1. _____ 2. _____ 3. _____	
3	3		A1/Vm2/A3 ou Vm1/A3/Vm2	1. _____ 2. _____ 3. _____	
4	3		A1/Vm2/A2	1. _____ 2. _____ 3. _____	
5	4		Vm2/Vd1/Vm3/Vd3 ou Vm1/Vd2/Vm3/Vd3	1. _____ 2. _____ 3. _____	
6	4		A1/Vm2/Vd2/A3	1. _____ 2. _____ 3. _____	
7	4		Vm2/Vd1/Vm3/A3	1. _____ 2. _____ 3. _____	
8	4		A1/Vm2/A2/Vd1	1. _____ 2. _____ 3. _____	
9	5		Vm1/Vd2/Vm3/Vd3/A3 ou Vm2/Vd1/Vm3/Vd3/A3	1. _____ 2. _____ 3. _____	
10	5		Vm2/Vd1/Vm3/A3/Vd3	1. _____ 2. _____ 3. _____	
11	5		A1/Vm2/Vd2/A3/Vd3	1. _____ 2. _____ 3. _____	
12	5		A1/Vm2/Vd2/A3/Vd1 ou A1/Vm2/A2/Vd1/A3	1. _____ 2. _____ 3. _____	
Total					_____

TESTE WISCONSIN DE CLASSIFICAÇÃO DE CARTAS (WCST)

Data de aplicação: __/__/__

Hora de início: __ __: __ __

Hora de término: __ __: __ __

Examinador _____ Sujeito _____

C F N C F N

Categoria	Nº	Resposta	C	F	N	O	P	Categoria	Nº	Resposta	C	F	N	O	P
	1		C	F	N	O			33		C	F	N	O	
	2		C	F	N	O			34		C	F	N	O	
	3		C	F	N	O			35		C	F	N	O	
	4		C	F	N	O			36		C	F	N	O	
	5		C	F	N	O			37		C	F	N	O	
	6		C	F	N	O			38		C	F	N	O	
	7		C	F	N	O			39		C	F	N	O	
	8		C	F	N	O			40		C	F	N	O	
	9		C	F	N	O			41		C	F	N	O	
	10		C	F	N	O			42		C	F	N	O	
	11		C	F	N	O			43		C	F	N	O	
	12		C	F	N	O			44		C	F	N	O	
	13		C	F	N	O			45		C	F	N	O	
	14		C	F	N	O			46		C	F	N	O	
	15		C	F	N	O			47		C	F	N	O	
	16		C	F	N	O			48		C	F	N	O	
	17		C	F	N	O			49		C	F	N	O	
	18		C	F	N	O			50		C	F	N	O	
	19		C	F	N	O			51		C	F	N	O	
	20		C	F	N	O			52		C	F	N	O	
	21		C	F	N	O			53		C	F	N	O	
	22		C	F	N	O			54		C	F	N	O	
	23		C	F	N	O			55		C	F	N	O	
	24		C	F	N	O			56		C	F	N	O	
	25		C	F	N	O			57		C	F	N	O	
	26		C	F	N	O			58		C	F	N	O	
	27		C	F	N	O			59		C	F	N	O	
	28		C	F	N	O			60		C	F	N	O	
	29		C	F	N	O			61		C	F	N	O	
	30		C	F	N	O			62		C	F	N	O	
	31		C	F	N	O			63		C	F	N	O	
	32		C	F	N	O			64		C	F	N	O	

TESTE DE STROOP**Folha de Resultados**

Data de aplicação: __/__/__ Hora de início: __ __: __ __ Hora de término: __ __: __ __

Examinador _____ Sujeito _____

FIGURAS				CORES				NOMES			
VD	AZ	AM	VM	VD	AZ	AM	VM	VD	AZ	AM	VM
AM	VM	VD	AZ	AM	VM	VD	AZ	AM	VM	VD	AZ
AZ	VD	AM	VM	AZ	VD	AM	VM	AZ	VD	AM	VM
AZ	AM	VM	VD	AZ	AM	VM	VD	AZ	AM	VM	VD
VM	VD	AZ	AM	VM	VD	AZ	AM	VM	VD	AZ	AM
AM	VD	AZ	VM	AM	VD	AZ	VM	AM	VD	AZ	VM

	Tempo (s)	Nº. de Erros
I		
II		
III		

⇒ **Após aplicação dos testes, aplicar o seguinte questionário:**

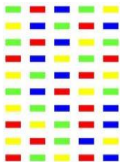
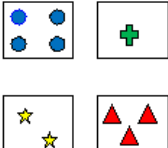
1. Qual das formas de aplicação foi mais agradável de realizar.
2. Como você classificaria a execução do **[nome do teste]** na versão **[tipo da versão]**?

		Classificação			
		Agradável	Fácil	Moderado	Difícil
Teste de Stroop	Manual				
	Computadorizado				
Torre de Londres	Manual				
	Computadorizado				
Teste Wisconsin de Classificação de Cartas	Manual				
	Computadorizado				

Fim 😊

ANEXO H- Modelo de tabela de preferência e grau de dificuldade

Tabela 18- Preferência e grau de dificuldade

Teste de Stroop			
Manual 	Fácil 	Moderado 	Difícil 
Computadorizada 	Fácil 	Moderado 	Difícil 
Torre de Londres (TOL)			
Manual 	Fácil 	Moderado 	Difícil 
Computadorizada 	Fácil 	Moderado 	Difícil 
Teste Wisconsin de Classificação de Cartas (WCST)			
Manual 	Fácil 	Moderado 	Difícil 
Computadorizada 	Fácil 	Moderado 	Difícil 

ANEXO I- Correlações entre o Teste Raven e ToL, WCST-64 e Stroop

Tabela 19- Correlações com o Teste Raven

ToL	r	Valor de p
Pontuação Raven x		
Número de movimentos manual	-0,310	0,161
Pontuação manual	0,242	0,279
Número de movimentos computadorizado	-0,412	0,057
Pontuação computadorizado	0,403	0,063
Percentual Raven x		
Número de movimentos manual	-0,329	0,135
Pontuação manual	0,173	0,443
Número de movimentos computadorizado	-0,477	0,025
Pontuação computadorizado	0,473	0,026
Nível Raven x		
Número de movimentos manual	0,220	0,325
Pontuação manual	-0,168	0,455
Número de movimentos computadorizado	0,463	0,030
Pontuação computadorizado	-0,469	0,028
WCST-64	r	Valor de p
Pontuação Raven x		
Manual		
Número de acertos	0,158	0,458
Número e percentual de erros	-0,158	0,482
Número e percentual de respostas perseverativas	-0,211	0,345
Número e percentual de erros perseverativos	-0,180	0,423
Número e percentual de erros não perseverativos	-0,182	0,419
Categorias completadas	0,075	0,740
Ensaio para completar a 1ª categoria	0,174	0,439
Fracasso em manter o contexto	0,078	0,729
Computadorizado		
Número de acertos	0,368	0,092
Número e percentual de erros	-0,368	0,092
Número e percentual de respostas perseverativas	0,107	0,637
Número e percentual de erros perseverativos	0,083	0,714
Número e percentual de erros não perseverativos	-0,294	0,184
Categorias completadas	0,505	0,017
Ensaio para completar a 1ª categoria	-0,323	0,143
Fracasso em manter o contexto	-0,428	0,047
Percentual Raven x		
Manual		
Número de acertos	0,338	0,124
Número e percentual de erros	-0,338	0,124
Número e percentual de respostas perseverativas	-0,376	0,085
Número e percentual de erros perseverativos	-0,344	0,117
Número e percentual de erros não perseverativos	-0,183	0,414
Categorias completadas	0,252	0,258
Ensaio para completar a 1ª categoria	0,141	0,531
Fracasso em manter o contexto	0,073	0,748
Computadorizado		
Número de acertos	0,501	0,018
Número e percentual de erros	-0,501	0,018
Número e percentual de respostas perseverativas	0,039	0,862

Número e percentual de erros perseverativos	-0,049	0,827
Número e percentual de erros não perseverativos	-0,379	0,082
Categorias completadas	0,582	0,005
Ensaio para completar a 1ª categoria	-0,286	0,196
Fracasso em manter o contexto	-0,359	0,101
Nível Raven x		
Manual		
Número de acertos	-0,301	0,173
Número e percentual de erros	0,301	0,173
Número e percentual de respostas perseverativas	0,302	0,172
Número e percentual de erros perseverativos	0,267	0,230
Número e percentual de erros não perseverativos	0,146	0,518
Categorias completadas	-0,274	0,217
Ensaio para completar a 1ª categoria	-0,095	0,673
Fracasso em manter o contexto	0,047	0,836
Computadorizado		
Número de acertos	-0,549	0,008
Número e percentual de erros	0,549	0,008
Número e percentual de respostas perseverativas	0,011	0,963
Número e percentual de erros perseverativos	0,153	0,497
Número e percentual de erros não perseverativos	0,403	0,063
Categorias completadas	-0,527	0,012
Ensaio para completar a 1ª categoria	0,245	0,271
Fracasso em manter o contexto	0,210	0,348
Stroop	r	Valor de p
Pontuação Raven x		
Manual		
Tempo I	-0,274	0,216
Erro I	-0,49	0,827
Tempo II	-0,434	0,043
Erro II	0,171	0,446
Tempo III	-0,201	0,369
Erro III	0,034	0,881
Computadorizado		
Tempo I	-0,455	0,034
Erro I	0,193	0,390
Tempo II	-0,425	0,049
Erro II	-0,286	0,197
Tempo III	-0,348	0,113
Erro III	-0,283	0,202
Percentual Raven x		
Manual		
Tempo I	-0,418	0,053
Erro I	-0,157	0,485
Tempo II	-0,508	0,016
Erro II	0,069	0,760
Tempo III	-0,315	0,153
Erro III	0,010	0,963
Computadorizado		
Tempo I	-0,448	0,037
Erro I	0,155	0,492
Tempo II	-0,409	0,058
Erro II	-0,307	0,164
Tempo III	-0,327	0,137
Erro III	-0,285	0,198
Nível Raven x		
Manual		
Tempo I	0,530	0,011
Erro I	0,224	0,316

Tempo II	0,284	0,200
Erro II	-0,059	0,795
Tempo III	0,327	0,137
Erro III	-0,076	0,738
Computadorizado		
Tempo I	0,321	0,145
Erro I	-0,136	0,547
Tempo II	0,355	0,105
Erro II	0,360	0,100
Tempo III	0,306	0,167
Erro III	0,307	0,164

ANEXO J- Correlações entre o Teste Raven e ToL, WCST-64 e Stroop

Tabela 20- Correlações entre os dados dos questionários

Dados dos questionários	r	Valor de p
Idade x		
Critério Socioeconômico	-0,369	0,091
Gostar de computador	0,327	0,137
Idade que utilizou computador pela 1ª vez	0,037	0,869
Tempo gasto no computador	-0,066	0,772
Habilidades com computador	0,009	0,970
Tempo gasto na internet	0,020	0,928
Tempo gasto com vídeo-game	0,117	0,615
Critério Socioeconômico x		
Gostar de computador	-0,337	0,125
Idade que utilizou computador pela 1ª vez	-0,337	0,125
Tempo gasto no computador	0,214	0,339
Habilidades com computador	-0,037	0,869
Tempo gasto na internet	0,247	0,268
Tempo gasto com vídeo-game	0,129	0,577
Gostar de computador		
Idade que utilizou computador pela 1ª vez	0,049	0,829
Tempo gasto no computador	-0,087	0,699
Habilidades com computador	0,381	0,080**
Tempo gasto na internet	0,022	0,923
Tempo gasto com vídeo-game	0,481	0,027*
Idade que utilizou computador pela 1ª vez		
Tempo gasto no computador	-0,564	0,006*
Habilidades com computador	-0,379	0,082**
Tempo gasto na internet	-0,556	0,007*
Tempo gasto com vídeo-game	0,098	0,673
Tempo gasto no computador		
Habilidades com computador	0,528	0,012*
Tempo gasto na internet	0,955	<0,001*
Tempo gasto com vídeo-game	-0,115	0,618
Habilidade com computador		
Tempo gasto na internet	0,585	0,004*
Tempo gasto com vídeo-game	0,018	0,938
Tempo gasto na internet		
Tempo gasto com vídeo-game	-0,119	0,608

* houve correlação

**tendência a correlação

ANEXO L- Correlações entre itens do Questionário e ToL, WCST-64 e Stroop

Tabela 21- Comparações dos dados dos questionários e ToL, WCST-64 e Stroop

Dados do questionário		
ToL	r	Valor de p
Idade x		
Número de movimentos manual	-0,105	0,643
Pontuação manual	0,317	0,151
Número de movimentos computadorizado	0,006	0,978
Pontuação computadorizado	-0,027	0,904
Critério Socioeconômico x		
Número de movimentos manual	0,071	0,752
Pontuação manual	0,265	0,233
Número de movimentos computadorizado	-0,159	0,480
Pontuação computadorizado	0,139	0,539
Gostar de computador x		
Número de movimentos manual	-0,177	0,431
Pontuação manual	-0,015	0,947
Número de movimentos computadorizado	0,072	0,751
Pontuação computadorizado	-0,113	0,617
Idade que usou computador pela 1ª vez x		
Número de movimentos manual	0,345	0,116
Pontuação manual	-0,527	0,012*
Número de movimentos computadorizado	0,167	0,458
Pontuação computadorizado	-0,101	0,655
Tempo gasto no computador x		
Número de movimentos manual	-0,281	0,206
Pontuação manual	0,219	0,327
Número de movimentos computadorizado	-0,374	0,087**
Pontuação computadorizado	0,352	0,108
Habilidades com computador x		
Número de movimentos manual	-0,020	0,928
Pontuação manual	-0,096	0,670
Número de movimentos computadorizado	-0,338	0,124
Pontuação computadorizado	0,280	0,208
Tempo gasto na internet x		
Número de movimentos manual	-0,343	0,118
Pontuação manual	0,275	0,216
Número de movimentos computadorizado	-0,429	0,047*
Pontuação computadorizado	0,407	0,060**
Tempo gasto com vídeo-game x		
Número de movimentos manual	0,017	0,940
Pontuação manual	0,255	0,264
Número de movimentos computadorizado	0,281	0,218
Pontuação computadorizado	-0,289	0,204
WCST-64	r	Valor de p
Manual		
Idade		
Número de acertos	-0,498	0,018*
Número e percentual de erros	0,498	0,018*
Número e percentual de respostas perseverativas	1,463	0,030*
Número e percentual de erros perseverativos	0,509	0,016*
Número e percentual de erros não perseverativos	0,098	0,663
Categorias completadas	-0,558	0,007*
Ensaio para completar a 1ª categoria	0,251	0,260
Fracasso em manter o contexto	0,010	0,956

Critério Socioeconômico

Número de acertos	0,320	0,147
Número e percentual de erros	-0,320	0,147
Número e percentual de respostas perseverativas	-0,386	0,076**
Número e percentual de erros perseverativos	-0,415	0,055**
Número e percentual de erros não perseverativos	0,010	0,964
Categorias completadas	0,333	0,130
Ensaio para completar a 1ª categoria	-0,114	0,615
Fracasso em manter o contexto	0,191	0,394

Gostar de computador

Número de acertos	-0,368	0,092**
Número e percentual de erros	0,368	0,092**
Número e percentual de respostas perseverativas	0,327	0,137
Número e percentual de erros perseverativos	0,308	0,163
Número e percentual de erros não perseverativos	-0,028	0,901
Categorias completadas	-0,325	0,140
Ensaio para completar a 1ª categoria	0,195	0,385
Fracasso em manter o contexto	-0,128	0,570

Idade que usou computador pela 1ª vez

Número de acertos	-0,059	0,794
Número e percentual de erros	0,059	0,794
Número e percentual de respostas perseverativas	0,076	0,736
Número e percentual de erros perseverativos	0,099	0,660
Número e percentual de erros não perseverativos	-0,286	0,196
Categorias completadas	0,065	0,775
Ensaio para completar a 1ª categoria	0,298	0,178
Fracasso em manter o contexto	-0,170	0,448

Tempo gasto no computador

Número de acertos	0,176	0,433
Número e percentual de erros	-0,176	0,433
Número e percentual de respostas perseverativas	-0,125	0,581
Número e percentual de erros perseverativos	-0,109	0,630
Número e percentual de erros não perseverativos	-0,004	0,987
Categorias completadas	0,043	0,849
Ensaio para completar a 1ª categoria	-0,104	0,644
Fracasso em manter o contexto	0,293	0,186

Habilidades com computador

Número de acertos	-0,052	0,817
Número e percentual de erros	0,052	0,817
Número e percentual de respostas perseverativas	0,035	0,879
Número e percentual de erros perseverativos	-0,005	0,983
Número e percentual de erros não perseverativos	-0,026	0,909
Categorias completadas	-0,155	0,490
Ensaio para completar a 1ª categoria	-0,269	0,226
Fracasso em manter o contexto	0,253	0,256

Tempo gasto na internet

Número de acertos	0,164	0,465
Número e percentual de erros	-0,164	0,465
Número e percentual de respostas perseverativas	-0,150	0,504
Número e percentual de erros perseverativos	-0,138	0,540
Número e percentual de erros não perseverativos	-0,051	0,821
Categorias completadas	-0,013	0,954
Ensaio para completar a 1ª categoria	-0,059	0,795
Fracasso em manter o contexto	0,358	0,102

Tempo gasto com vídeo-game

Número de acertos	-0,258	0,260
Número e percentual de erros	0,258	0,260
Número e percentual de respostas perseverativas	0,203	0,378
Número e percentual de erros perseverativos	0,144	0,532
Número e percentual de erros não perseverativos	0,178	0,440

Categorias completadas	-0,088	0,703
Ensaio para completar a 1ª categoria	0,104	0,655
Fracasso em manter o contexto	-0,204	0,376
Computadorizado		
Idade		
Número de acertos	-0,370	0,091
Número e percentual de erros	0,370	0,091
Número e percentual de respostas perseverativas	0,043	0,850
Número e percentual de erros perseverativos	0,283	0,202
Número e percentual de erros não perseverativos	0,157	0,485
Categorias completadas	-0,203	0,364
Ensaio para completar a 1ª categoria	-0,101	0,656
Fracasso em manter o contexto	-0,107	0,636
Critério Socioeconômico		
Número de acertos	0,485	0,022*
Número e percentual de erros	-0,485	0,022*
Número e percentual de respostas perseverativas	0,044	0,844
Número e percentual de erros perseverativos	-0,322	0,145
Número e percentual de erros não perseverativos	-0,466	0,029*
Categorias completadas	0,304	0,169
Ensaio para completar a 1ª categoria	-0,459	0,032*
Fracasso em manter o contexto	0,085	0,708
Gostar de computador		
Número de acertos	-0,449	0,036*
Número e percentual de erros	0,449	0,036*
Número e percentual de respostas perseverativas	0,076	0,738
Número e percentual de erros perseverativos	0,263	0,237
Número e percentual de erros não perseverativos	0,369	0,091
Categorias completadas	-0,204	0,363
Ensaio para completar a 1ª categoria	-0,224	0,317
Fracasso em manter o contexto	-0,351	0,109
Idade que usou computador pela 1ª vez		
Número de acertos	-0,007	0,974
Número e percentual de erros	0,007	0,974
Número e percentual de respostas perseverativas	-0,297	0,179
Número e percentual de erros perseverativos	-0,181	0,420
Número e percentual de erros não perseverativos	0,128	0,571
Categorias completadas	-0,307	0,164
Ensaio para completar a 1ª categoria	0,353	0,107
Fracasso em manter o contexto	0,504	0,017*
Tempo gasto no computador		
Número de acertos	0,416	0,054*
Número e percentual de erros	-0,416	0,054*
Número e percentual de respostas perseverativas	0,357	0,103
Número e percentual de erros perseverativos	0,149	0,507
Número e percentual de erros não perseverativos	-0,362	0,098
Categorias completadas	0,552	0,008*
Ensaio para completar a 1ª categoria	-0,325	0,140
Fracasso em manter o contexto	-0,437	0,042*
Habilidades com computador		
Número de acertos	0,122	0,587
Número e percentual de erros	-0,122	0,587
Número e percentual de respostas perseverativas	0,285	0,199
Número e percentual de erros perseverativos	0,184	0,411
Número e percentual de erros não perseverativos	-0,072	0,749
Categorias completadas	0,179	0,426
Ensaio para completar a 1ª categoria	-0,186	0,407
Fracasso em manter o contexto	-0,458	0,032*
Tempo gasto na internet		
Número de acertos	0,464	0,030*

Número e percentual de erros	-0,464	0,030*
Número e percentual de respostas perseverativas	0,240	0,281
Número e percentual de erros perseverativos	0,059	0,795
Número e percentual de erros não perseverativos	-0,399	0,066**
Categorias completadas	0,534	0,010*
Ensaio para completar a 1ª categoria	-0,320	0,146
Fracasso em manter o contexto	-0,394	0,069**
Tempo gasto com vídeo-game		
Número de acertos	-0,237	0,301
Número e percentual de erros	0,237	0,301
Número e percentual de respostas perseverativas	0,230	0,316
Número e percentual de erros perseverativos	0,226	0,324
Número e percentual de erros não perseverativos	0,070	0,763
Categorias completadas	-0,290	0,203
Ensaio para completar a 1ª categoria	-0,137	0,555
Fracasso em manter o contexto	0,079	0,735
Stroop		
Idade x		
Manual		
Tempo I	0,349	0,111
Erro I	0,027	0,904
Tempo II	0,245	0,273
Erro II	0,257	0,248
Tempo III	0,195	0,386
Erro III	0,068	0,764
Computadorizado		
Tempo I	0,003	0,990
Erro I	0,765	0,464
Tempo II	0,061	0,787
Erro II	-0,121	0,592
Tempo III	0,223	0,319
Erro III	0,191	0,394
Critério Socioeconômico x		
Manual		
Tempo I	-0,513	0,015*
Erro I	0,137	0,543
Tempo II	-0,370	0,090
Erro II	-0,013	0,953
Tempo III	-0,127	0,574
Erro III	0,043	0,848
Computadorizado		
Tempo I	-0,436	0,042*
Erro I	-0,133	0,556
Tempo II	-0,429	0,046*
Erro II	0,141	0,533
Tempo III	-0,493	0,020*
Erro III	-0,111	0,622
Gostar de Computador x		
Manual		
Tempo I	0,053	0,814
Erro I	0,325	0,140
Tempo II	-0,017	0,939
Erro II	0,123	0,585
Tempo III	0,303	0,170
Erro III	0,061	0,789
Computadorizado		
Tempo I	0,265	0,233
Erro I	-0,172	0,444
Tempo II	0,207	0,356
Erro II	-0,248	0,265

Tempo III	0,379	0,082**
Erro III	-0,027	0,906
Idade que usou computador pela 1ª vez x		
Manual		
Tempo I	-0,042	0,852
Erro I	-0,074	0,744
Tempo II	0,379	0,082**
Erro II	0,206	0,358
Tempo III	0,444	0,039*
Erro III	0,371	0,089**
Computadorizado		
Tempo I	0,322	0,144
Erro I	0,053	0,816
Tempo II	0,206	0,358
Erro II	0,153	0,498
Tempo III	0,164	0,465
Erro III	-0,098	0,666
Tempo Gasto no computador x		
Manual		
Tempo I	0,174	0,438
Erro I	-0,223	0,318
Tempo II	-0,299	0,177
Erro II	-0,091	0,686
Tempo III	-0,365	0,095
Erro III	-0,352	0,108
Computadorizado		
Tempo I	-0,215	0,336
Erro I	0,156	0,487
Tempo II	-0,179	0,426
Erro II	-0,324	0,141
Tempo III	-0,289	0,193
Erro III	-0,473	0,026*
Habilidades com computador x		
Manual		
Tempo I	0,241	0,280
Erro I	-0,029	0,899
Tempo II	-0,079	0,726
Erro II	-0,273	0,218
Tempo III	-0,010	0,965
Erro III	-0,175	0,437
Computadorizado		
Tempo I	-0,022	0,923
Erro I	-0,083	0,713
Tempo II	0,177	0,430
Erro II	-0,277	0,212
Tempo III	0,109	0,628
Erro III	-0,267	0,229
Tempo gasto na internet x		
Manual		
Tempo I	0,083	0,714
Erro I	-0,126	0,576
Tempo II	-0,347	0,114
Erro II	-0,133	0,554
Tempo III	-0,373	0,087**
Erro III	-0,331	0,132
Computadorizado		
Tempo I	-0,214	0,340
Erro I	0,172	0,444
Tempo II	-0,182	0,418
Erro II	-0,308	0,164

Tempo III	-0,244	0,273
Erro III	-0,402	0,063**
Tempo gasto no vídeo-game x		
Manual		
Tempo I	-0,069	0,766
Erro I	0,233	0,309
Tempo II	-0,199	0,387
Erro II	0,327	0,148
Tempo III	0,086	0,711
Erro III	-0,207	0,369
Computadorizado		
Tempo I	-0,223	0,332
Erro I	-0,238	0,300
Tempo II	-0,292	0,199
Erro II	-0,105	0,651
Tempo III	-0,254	0,267
Erro III	0,041	0,861

* houve correlação

**tendência a correlação