

SOFTWARE CONTINGÊNCIA PROGRAMADA: UTILIDADE E FUNCIONALIDADES

PROGRAMMED CONTINGENCY SOFTWARE: ITS UTILITY AND FUNCTIONALITIES¹

ELENICE S. HANNA

LUIZ ANÍSIO VIEIRA BATITUCCI

JASSANÃ DA SILVA LACERDA BATITUCCI

UNIVERSIDADE DE BRASÍLIA, BRASIL E

INSTITUTO DE CIÊNCIA E TECNOLOGIA SOBRE COMPORTAMENTO, COGNIÇÃO E ENSINO, BRASIL

RESUMO

O objetivo do presente artigo é descrever as características e o contexto de criação do *software* Contingência Programada (CP), desenvolvido para programar contingências de ensino ou avaliação de discriminações simples e condicionais com consequências diferenciais para o comportamento. O *software* está disponível para download gratuito para a comunidade científica. Foi idealizado com interface amigável com o usuário e não requer experiência em programação para arranjar e testar os protocolos experimentais. Permite a utilização de nove estímulos e chaves de respostas. O arquivo de saída fornece as informações sobre tempo que os eventos ocorrem, latência e número de respostas, com a especificação dos parâmetros das tentativas. O formato dos registros facilita a conferência e a análise dos resultados e permite a exportação para MS Excel. A descrição de características de um *software* flexível, utilizando o CP como modelo, pode reduzir insucessos na interação entre pesquisadores e profissionais de tecnologia da informação.

Palavras-chave: *software* para pesquisa, discriminação simples, discriminação condicional, pesquisa, ensino, análise do comportamento, psicologia cognitiva.

ABSTRACT

This article describes characteristics and the context of creation of the Programmed Contingency software (CP), which allows the programming of contingencies to teach and test simple and conditional discriminations with differential behavior consequences. The software is downloading free for the scientific community. It was planned with user-friendly interface for researchers with no experience in computer programming for setting and testing experimental protocols. CP allows the use of nine stimuli and response windows. The output file provides information about time of events, latency and number of responses, with the corresponding parameters of each trial. Records' formatting facilitates the conference and analysis of results and allows export to MS Excel. The description of characteristics of flexible software, taking CP as a model, may reduce failures in the interaction between researchers and information technology professionals.

Keywords: software, simple discrimination contingency, conditional discrimination contingency, research, teaching, behavior analysis, cognitive psychology.

¹ Durante a redação do artigo a primeira autora era pesquisadora do INCT sobre Comportamento, Cognição e Ensino, Edital MCT 15/2008 (CNPq 573972/2008-7, FAPESP 2008/57705-8). Os autores agradecem aos revisores que contribuíram para melhorar o trabalho com suas sugestões. Endereço para correspondência: elenicehanna@gmail.com.

O pesquisador experimental das ciências humanas tem necessidade de desenvolver tecnologia de informação (TI) voltada para seu trabalho como o de outras áreas. A tecnologia pode ser voltada para a instrumentalização das suas atividades de pesquisa ou para automatizar os procedimentos e protocolos desenvolvidos com a finalidade de aplicar os conhecimentos. As dificuldades para desenvolver a tecnologia de informação, entretanto, começam pelos menores recursos dedicados às ciências humanas, aliadas ao alto custo do acesso a esta tecnologia. Além disso, as dificuldades na interação com os profissionais de TI, que utilizam a lógica de programação e o jargão não familiar aos pesquisadores das ciências humanas, podem gerar produtos com utilidade limitada e pouco flexíveis. Este artigo relata uma experiência no desenvolvimento de tecnologia da informação para uso em pesquisa psicológica, em especial para o estudo do controle de estímulos. O objetivo é descrever e divulgar o *software* “Contingência Programada” para o uso de outros pesquisadores, bem como fornecer um modelo de produto que pode aumentar as chances de sucesso de outros pesquisadores.

O *software* “Contingência Programada” (CP) foi desenvolvido em 2006 pelo segundo autor desse trabalho, quando a terceira autora estava realizando seu curso de Mestrado na Universidade de Brasília, sob a orientação da primeira autora. As pesquisas para estudar o ensino e a aprendizagem de rudimentos de leitura musical precisavam de um *software* com características não identificadas em outros que a equipe tinha acesso devido ao trabalho colaborativo e à cordialidade de seus autores. Algumas dificuldades estavam relacionadas ao trabalho para confeccionar arquivos de sons e imagens com exigências muito específicas (e.g., GEIC – Gerenciador de Ensino Individualizado por Computador, desenvolvido por D. Capobiano, C. Teixeira, R. E. Bela, A. F. Orlando, D. G. de Souza e J. C. de Rose, 2009), outras à limitação de uso apenas em equipamento Macintosh (e.g., MTS – *Match to sample program*, desenvolvido por William Dube e Eric Hiris, 1996). Todos exigiam um treinamento inicial para sua utilização e algum conhecimento em programação para arranjar as contingências desejadas. A forma de programar as muitas sessões diferentes que o projeto exigia e a pouca ou nenhuma experiência em programação de alunos de psicologia pediam uma ferramenta fácil para arranjar e testar parâmetros e contingências que podiam mudar entre e intra sessões experimentais. Além disso, as experiências anteriores de interações pouco produtivas com profissionais de TI fizeram com que a meta fosse um *software* flexível para a utilização em diversos outros estudos e não planejado para o estudo da aluna.

O tempo disponível para o desenvolvimento e teste da ferramenta foi outro enorme desafio, devido a curta duração do mestrado. Havia se passado um semestre, a coleta era relativamente longa e, portanto, o programador tinha apenas seis meses para todas as etapas que envolvem o desenvolvimento de um

software complexo. A formação diferenciada do programador, com conhecimento sobre análise do comportamento, juntamente com a proximidade e compromisso dos membros da equipe durante o processo de criação, foram fundamentais para que o desafio fosse aceito e superado no tempo disponível.

O CP foi inicialmente planejado para programar tarefas de treino discriminativo simples simultâneo ou sucessivo (Harrison, 1991) e pareamento ao modelo com e sem atraso (Mackay, 1991). Apesar dos objetivos específicos iniciais, pessoas com experiência em programação conseguirão identificar outros propósitos.

O *software* foi desenvolvido em linguagem Delphi 5 na plataforma Microsoft Windows. Ele permite a utilização de tipos de arquivos de uso universal (wav, bmp, jpg) que podem ser criados e editados com a utilização de ferramentas gratuitas disponíveis na Internet. Para utilizar o *software*, é necessário um equipamento com sistema Windows 98/ME/2000/XP/Vista ou superior², processador de pelo menos 800MHz e memória RAM de 128Mb. Pesquisas que utilizam estímulos sonoros precisam de máquina equipada com dispositivo de som. Pode ser utilizado com tela sensível ao toque (e.g., *Elo TouchSystems 2700* da marca *IntelliTouch*; Filgueiras, 2011) e com equipamento de rastreamento ocular (e.g., ISCAN ETL-300 da ISCAN Inc.; Oliveira, 2014).

A distribuição do PC é gratuita, após envio de e-mail (ver nota do autor) para a primeira autora, mas não é livre para troca entre pessoas não autorizadas. Os autores querem manter o registro de quem utiliza e para qual finalidade o software está sendo utilizado.

O CP deixa à disposição do pesquisador a possibilidade de apresentar até nove estímulos na tela do monitor, dispostos em matriz 3 x 3, que são facilmente planejados e de maneira interativa. Os estímulos podem ser figuras e sons³.

Como exemplo, suponha a programação de uma tarefa de pareamento ao modelo arbitrário, em que se deve escolher partituras correspondentes aos estímulos modelos auditivos. Uma das tentativas é exemplificada na Figura 1. Na tela que inicia a tentativa, um modelo auditivo é tocado junto com um retângulo branco apresentado no monitor com fundo azul (Painel a). O som é repetido a cada 5 s, até que uma resposta sobre o retângulo ocorra. A resposta produz a tela que apresenta os estímulos de comparação, que são três partituras (Painel b). As respostas sobre um dos estímulos de comparação têm consequências diferentes: apresentação de *smiley* se o estímulo corresponder ao modelo (Painel c); e apresentação de um xis se o estímulo não corresponder ao modelo (Painel d).

² Testes realizados até Windows 7.

³ A versão CP 2.0 foi criada recentemente para permitir a apresentação de vídeos nas diferentes janelas. Esta versão não foi suficientemente testada, mas o acesso para pesquisadores com interesse em testá-la pode ser solicitado.

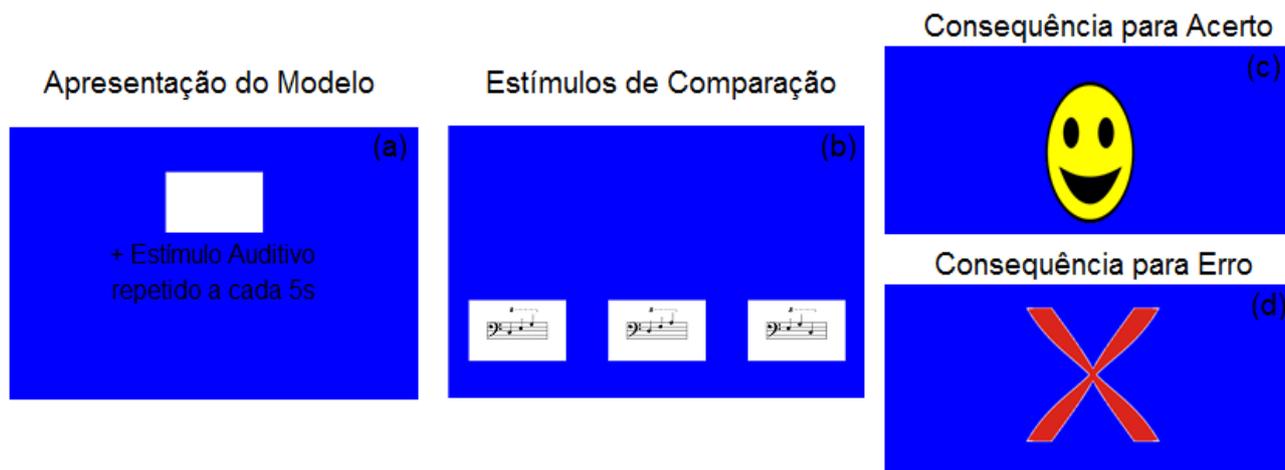


Figura 1. Exemplo de uma tentativa de pareamento ao modelo arbitrário auditivo-visual programável pelo software CP: tela de apresentação do modelo auditivo repetido a cada 5 s e retângulo branco (Painel a); tela com três estímulos de comparação visuais para escolha (Painel b); telas de consequências para respostas no estímulo correto (Painel c) e incorreto (Painel d).

Respostas em cada janela/estímulo e a duração da exposição a cada tela são registradas automaticamente, informando qual o estímulo escolhido em cada tentativa.

Em diferentes formas de programar as contingências relativas ao modelo, pode-se, por exemplo, apresentá-lo junto com os estímulos de comparação (pareamento simultâneo), escolher uma cor de fundo diferente de azul para a tela do modelo, ajustar o tempo de repetição do modelo auditivo, apresentar alguma imagem como consequência de resposta ao modelo (uma dica adicional, por exemplo), usar mais de um estímulo/janela como modelo composto, mudar a posição do estímulo modelo, e utilizar apenas o componente auditivo do modelo com tempo definido de exposição a esta tela.

Na tela com os estímulos de comparação (Figura 1, Painel b) são possíveis diversas variações, como: apresentar até nove comparações, aumentar o número de comparações gradualmente ao longo das tentativas, misturar tipos diferentes de estímulos (outros sons e outros estímulos visuais), definir mais de um estímulo como correto ou incorreto, e utilizar parte das janelas para emissão de resposta de observação de estímulos visuais ou auditivos.

As consequências diferenciais utilizadas como exemplo (*smiley* e o *xis*, Figura 1, painéis c e d) podem variar de posição. O pesquisador escolhe as imagens e/ou sons que preferir, com durações que podem ser definidas no contexto da tentativa. Cada tentativa pode ter consequências diferentes para acerto e erro. Em situações de teste, as consequências podem ser omitidas.

Além das telas do exemplo mostradas na Figura 1, estudos sobre controle de estímulos usualmente programam um intervalo que separa as tentativas (ITI, Mackay, 1991), durante o qual nenhum estímulo é apresentado. O CP deixa à disposição a possibilidade do uso de tela para ITI, com a sua duração e formatação facilmente configuráveis.

Existem também outros dois tipos de tela especiais que podem ter programação específica: a tela de instrução e a tela de finalização da sessão. A primeira pode ser inserida a qualquer tempo na sessão e permite a inclusão de instruções sobre o estudo ou tarefa. Este tipo de tela é útil para automatizar e padronizar as instruções e, em especial, quando a sessão é composta por mais de uma tarefa (e.g., de ensino e de avaliação) e requer o fornecimento de avisos ou de informações adicionais ao longo das tentativas. A tela final pode ser também configurada de maneira diferente das demais (e.g., cor de fundo, inclusão de alguma informação ou agradecimento ao participante).

Programar e configurar as tentativas e telas é uma tarefa feita de forma interativa em janela do CP chamada de *script*. A Figura 2 apresenta a tela do monitor com um exemplo de *script* do CP. A metade superior da figura mostra alguns parâmetros que o pesquisador pode configurar para cada sessão, bloco ou tentativa, e a metade inferior lista as tentativas e telas⁴ já programadas.

A seguir são apresentados os principais parâmetros que podem ser configurados no CP na ordem que aparecem na tela de um *script*. A informação sobre cada campo é fornecida quando o usuário mantém o cursor em um campo por 1s. Os parâmetros para configuração foram divididos em seis tipos: (1) configuração geral da tela ou tentativa (título usado no *script*: Código/Nome da Tela); (2) instrução [Instrução (opcional)]; (3) consequências para respostas (Consequência do S+, Consequência do S-, O que acontece ao escolher S+ ou S-); (4) critérios de saída (Critérios de Saída); (5) contadores; e (6) disposição e configuração das janelas (Disposição das chaves). A Figura 2 mostra os tipos 1, 2, 3 e parte do 4 e a Figura 3 mostra o tipo 6.

⁴ Uma tentativa pode ser programada em uma tela apenas ou pode exigir duas ou mais telas, por isso os termos são utilizados como sinônimos em algumas ocasiões.

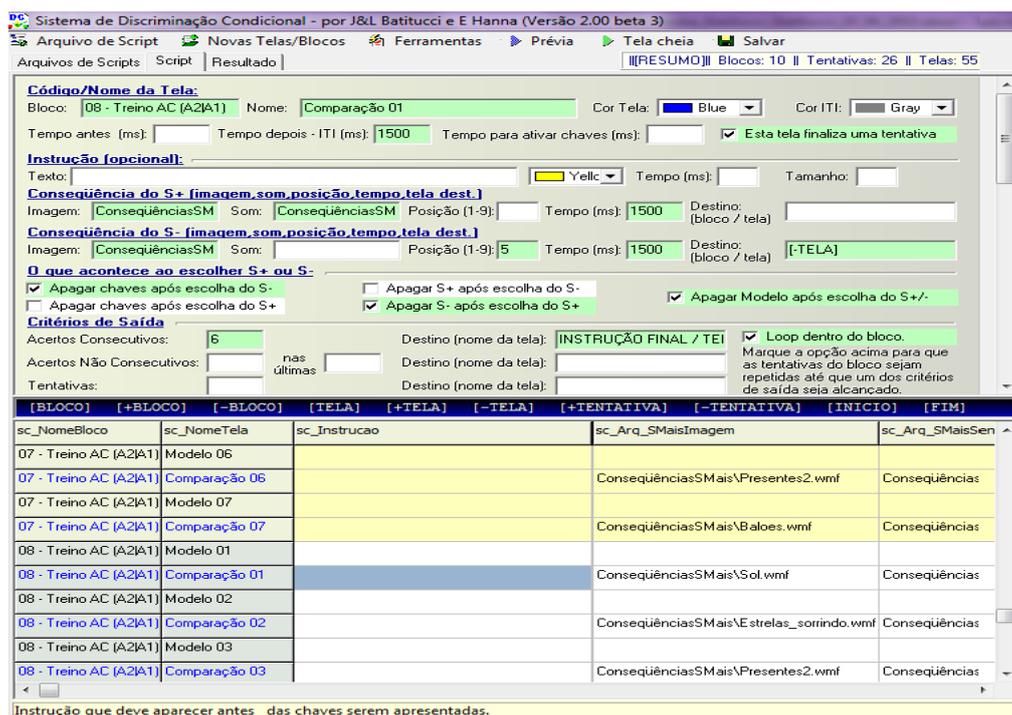


Figura 2. Tela de parte de um script de uma sessão experimental, com parte dos parâmetros que podem ser configurados (metade superior) e a lista de telas/tentativas programadas (metade inferior).

Configuração geral da tela/tentativa

Para cada tela é necessário informar o nome do bloco e o nome da tela/tentativa. Os parâmetros configuráveis da tela são: cor de fundo da tela, cor da tela durante o ITI, tempo antes de iniciar a tela/tentativa, duração do ITI e tempo para ativar as janelas a partir do início da tentativa. É necessário indicar se a tela finaliza uma tentativa, quando duas ou mais telas compõem uma tentativa. A definição do nome do bloco, composto por uma ou mais tentativas, tem especial importância na programação. Os critérios para avanço e repetição de tentativas e blocos (campos do tipo 5) são aplicados sobre as tentativas de um bloco, que por sua vez é diferenciado pelo nome fornecido como o primeiro parâmetro do *script*. Todas as telas que devem obedecer ao critério devem fazer parte do mesmo bloco. Qualquer mudança no nome do bloco em tentativas sucessivas é considerada como um novo bloco, sendo sinalizado automaticamente pela alteração na cor de preenchimento das linhas que compõem cada bloco (ora amarelo ora branca, ver metade inferior da Figura 2).

Para a tentativa utilizada como exemplo (Figura 1) as configurações da tela com os estímulos de comparação são as mostradas na Figura 2. O nome da tela utilizado foi “Comparação 01” do bloco “Treino AC (A2\A1)”, cor da tela azul, cor da tela durante o ITI cinza com duração de 1,5 s e a tela foi definida como aquela que finaliza a tentativa.

Instrução

Em telas de instrução, o texto é digitado diretamente no campo “Texto”. A cor e tamanho da fonte podem ser escolhidos, bem como o tempo que a instrução

permanecerá na tela. O exemplo acima não incluiu tela de instrução, por isso nenhum parâmetro foi configurado.

Consequências para respostas

Para as escolhas de estímulos definidos como corretos (S+) e incorretos (S-) é possível definir a imagem e/ou som que deve(m) ser apresentado(s), a posição, o tempo de apresentação e qual deve ser a próxima tela após as consequências. Pode-se também definir se as janelas com os estímulos se apagarão como consequência da escolha.

Para a tentativa utilizada como exemplo, foram definidos como consequências do S+ a apresentação da imagem do *smiley* por 1,5 s e de um som (ver Figuras 1 e 2). Como a posição da imagem (janela) foi deixada em branco (Figura 2), ela será definida aleatoriamente entre as nove janelas. Após essas consequências, o ITI entrará em vigor com tela cinza e a próxima tela será mostrada, uma vez que o campo de “Destino” foi deixado em branco. Para respostas no S- foi definido apenas a apresentação de uma imagem (Xis) como consequência, na posição da Janela 5 por 1,5 s. Como destino após a apresentação da consequência para erro foi definida a tela anterior [-Tela], que significa repetir a tentativa, uma vez que a tela anterior é a tela de apresentação do modelo.

Critérios de saída

A programação destes parâmetros permite definir critério de aprendizagem ou de finalização de um bloco de tentativas a partir de: número de acertos consecutivos ou não consecutivos, número de tentativas, tempo máximo na tela e máximo de repetições do bloco. Pode-se também indicar a tela de destino quando o critério escolhido for

alcançado. Se o critério implicar na repetição do bloco, é possível definir se haverá mudança na ordem das telas/tentativas, apenas marcando a opção “loop dentro do bloco”.

Para o exemplo utilizado, o critério definido (ver Figura 2) foi de seis acertos consecutivos com repetição do bloco por no máximo três vezes (parâmetro não mostrado na figura) com mudança na ordem das tentativas a cada repetição (“Loop dentro do bloco” foi marcado). Quando o critério for atingido, o destino será uma tela de Instrução Final.

Contadores

Pode-se mostrar contadores de acertos ou erros consecutivos e não consecutivos, de número de tentativas e de tempo. No exemplo, nenhum contador foi utilizado.

Os parâmetros apresentados a seguir se referem aos estímulos que são apresentados nas janelas e suas funções.

Disposição e configuração das janelas

O CP dispõe de uma configuração de tela com seis janelas ou *chaves* (termo utilizado no software) e outra com nove janelas. A diferença entre elas está na disposição das janelas na tela do monitor. A escolha de uma configuração ou outra é feita com um clique sobre uma das opções que aparecem na área de programação do script, conforme Figura 3. A configuração escolhida é sinalizada em verde escuro e imediatamente o programa apresenta a lista de janelas/chaves que podem ser configuradas. Para o exemplo, foi utilizada a configuração com seis chaves.

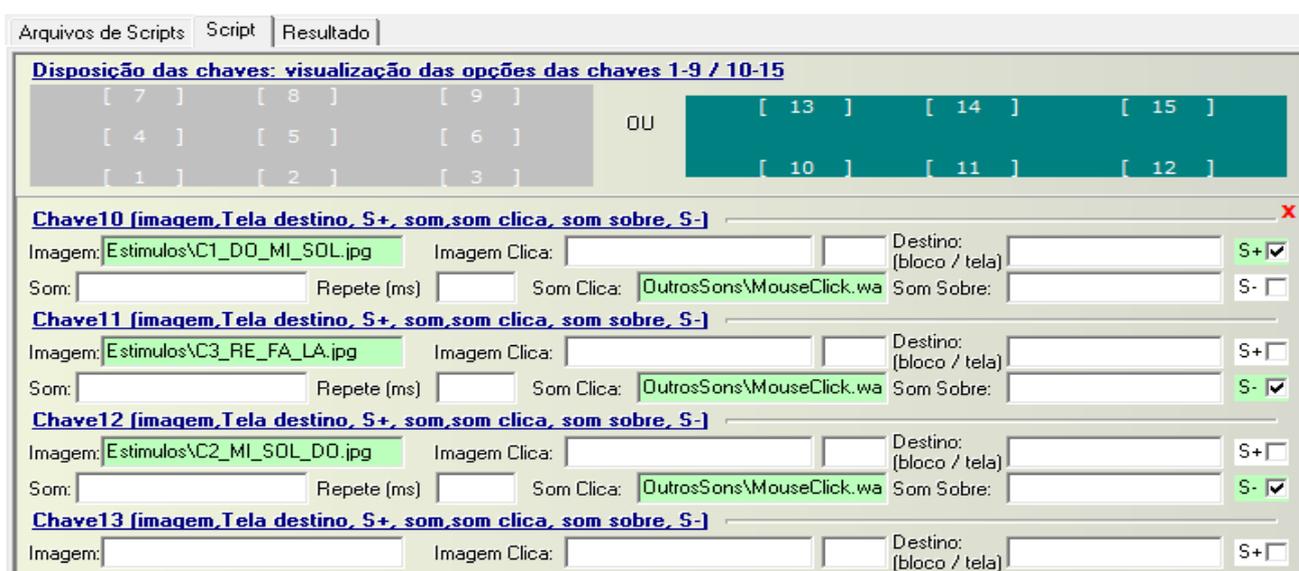


Figura 3. Print da tela do software CP que permite configurar a disposição das janelas/chaves de resposta, os estímulos correspondentes e suas funcionalidades no *script* de cada sessão experimental.

Cada janela/chave de resposta pode conter um estímulo visual e um estímulo auditivo correspondente, como na tela do modelo do exemplo (Figura 1, Painel a). Os estímulos devem ser confeccionados pelo pesquisador e organizados em pastas. A inclusão de um estímulo em uma das janelas é feita com um clique no campo correspondente de cada chave (Figura 3), que permitirá a escolha do arquivo que está em uma pasta do seu computador. No exemplo acima, para programar a repetição do modelo auditivo a cada 5 s, coloca-se 5000 no campo “Repete (ms)” da Chave 14 (configuração não mostrada na figura que se refere apenas à tela dos estímulos de comparação).

Pode-se programar a substituição do estímulo visual por outro quando ocorrer uma resposta naquela janela (incluindo o nome do arquivo no campo “Imagem Clica”) e do tempo que esta imagem deve permanecer na janela. Clicar na janela ou passar o cursor sobre ela pode também produzir um *feedback*

sonoro, cujo arquivo deve estar na pasta de estímulos (e.g., som de clique para indicar que a resposta foi registrada) e deve ser inserido no campo “Som Clica” ou “Som Sobre”. Para algumas pesquisas, pode ser importante definir em cada tentativa ou para cada janela/chave uma tela de destino diferente quando a chave é escolhida, o que é feito no campo “Destino (bloco/tela)”. Clicar neste campo produz a lista de telas programadas para a escolha interativa.

Por fim, deve-se definir a função da janela/estímulo como S+ (correto) ou S- (incorreto), mesmo que não se deseje apresentar consequências diferenciais, para fins de registro do número de acertos e erros. Esta definição é feita marcando-se a opção S+ ou S- de cada chave.

Para o exemplo, foram configuradas as chaves 10, 11 e 12, com os três arquivos de imagem que devem ser apresentados (as partituras mostradas na Figura 1, Painel b) e o som de clique quando uma resposta é

emitida na janela/chave. Os estímulos apresentados nas janelas 11 e 12 foram definidos como S- e o apresentado na janela 10 foi definido como S+ (ver Figura 3).

Todas as tentativas programadas e os parâmetros correspondentes a cada tela aparecem na metade inferior da tela de script (Figura 2). O pesquisador pode acompanhar e revisar a configuração que foi feita para cada tela e mudar manualmente qualquer dos parâmetros da tentativa/tela de interesse. Uma vez configurada uma tentativa ou bloco, é possível fazer a duplicação (no menu principal: *Novas Telas/Blocos*), bem como apagar. Existem também ferramentas para facilitar a programação de parâmetros temporais, sequências, critérios de saída e contadores, aplicados a um bloco ou a todo o *script* (menu principal: *Ferramentas*, Figura 2).

Em resumo, todas as configurações são feitas especificando-se o parâmetro numérico (e.g., número de acertos consecutivos), inserindo o texto (e.g., instrução) ou com duplo clique para a localização e escolha de uma das alternativas disponíveis (e.g., tela de destino) ou do arquivo desejado (e.g., figura, som).

O pesquisador poderá também fazer a programação do script em Excel ou outro software de edição, desde que mantenha a formatação do arquivo com extensão *.dc* salvando como arquivo texto. Esta alternativa economiza muito tempo, mas requer alguma experiência com o CP e Excel para evitar erros na definição dos parâmetros e, depois de programados, requer o teste cuidadoso das sessões antes do início da pesquisa.

Uma funcionalidade também extremamente útil para o pesquisador é o teste da programação da sessão em modo *Prévia* (menu principal, Figura 2). Ao clicar nesta função, o software executa a sessão em uma janela pequena (de tamanho ajustável), a partir da tela/tentativa escolhida pelo pesquisador. No modo *Prévia*, pode-se conferir quais estímulos estão programados em cada janela para cada tentativa, quais funções foram atribuídas a eles (S+ aparece com borda verde, S- com borda vermelha e sem função com borda cinza), os demais parâmetros e o fluxo da sessão. Em caso de necessidade de mudança, interrompe-se o modo *Prévia* desativando a janela adicional (com a tecla ESC), faz-se a modificação na tela que requer alteração e imediatamente se pode testar para verificar a nova programação.

Para executar uma sessão/script no modo pleno é necessário escolher o modo *Tela cheia* no menu principal (Figuras 2 e 3). A sessão executada nesse modo desabilita outras funcionalidades do Windows, exceto a tecla “Esc” que interrompe a execução do programa.

Os resultados são registrados com contadores para todos os eventos nas chaves de resposta, inclusive

os cliques sem efeito. O registro é feito a cada resposta, sem perda de dados em caso de falta de energia e com possibilidade de continuação da sessão posteriormente.

A Figura 4 mostra uma parte do arquivo de dados registrados pelo CP quando o script usado como exemplo foi executado. Para a figura foi feita a impressão da tela de parte das informações de um bloco com seis tentativas programadas. Como o participante escolheu o S- nas tentativas 1 e 3, essas tentativas foram repetidas (programação do destino para “-Tela” se clique em S-, Figura 2). Além disso, para atingir o critério programado de seis acertos consecutivos foram necessárias duas tentativas adicionais. Os dados extraídos incluem para cada tela: o nome do bloco (8 - Treino AC); o nome das etapas da tentativa (Modelo e Comparação); o evento (clique no S+ ou no S-); as informações relativas a cada chave de resposta (o estímulo programado, sua função e a latência da resposta em milissegundos); o número da tentativa; e os acertos acumulados. Erros e acertos são preenchidos com cores diferentes para facilitar a visualização.

A partir do arquivo de saída é possível plotar gráficos de acertos e erros ao longo da sessão e visualizar o desempenho do participante ao longo das tentativas. A Figura 5 apresenta os acertos acumulados nas 10 tentativas mostradas na Figura 4. Análises comparativas que envolvem várias sessões necessitam que o arquivo de saída seja exportado para o programa que o pesquisador está mais familiarizado para realizar suas análises.

O *software* CP tem suprido as necessidades para programação de pesquisas sobre ensino de leitura textual, leitura musical, superseletividade, repertório recombinativo, discriminação simples e condicional com adultos e crianças sob a orientação da primeira autora. Representou um avanço para a equipe programar pesquisas complexas com relativa rapidez e para os alunos participarem desta etapa dos projetos com maior independência do orientador. Desde o seu desenvolvimento, diversos estudos realizados por alunos de pós-graduação sob a orientação da primeira autora utilizaram o CP para a programação das contingências experimentais sem necessidade de alterações no código (e.g., Batitucci, 2007; de Paula, 2013; Fernandes, 2013; Filgueiras, 2011; Huber, 2010; Martins, 2010; Oliveira, 2012; Oliveira, 2014; Pereira, 2012; Verneque, 2011). A descrição dos procedimentos desses estudos apresentam exemplos detalhados da utilização e das funcionalidades do CP. Outros pesquisadores básicos de análise experimental do comportamento e psicologia cognitiva, principalmente os que estudam fenômenos de Controle de Estímulos, como discriminação e generalização, categorização, equivalência de estímulos, para citar alguns, podem ter benefícios com o compartilhamento desta ferramenta.

Bloco	Tela	Evento	Chave_10	Chave_11	Chave_12	Tentativa	Acertos Não Consecutivos
08 - Treino AC	Modelo 01					1	
08 - Treino AC	Comparação 01	ClickChaveS-	(S+) [DO_MI_SOL.jpg]	(Click 2153ms) (S-) [RE_FA_LA.jpg]	(S-) [MI_SOL_DO.jpg]	1	
08 - Treino AC	Modelo 01					2	
08 - Treino AC	Comparação 01	ClickChaveS+	(Click 2372ms) (S+) [DO_MI_SOL.jpg]	(S-) [RE_FA_LA.jpg]	(S-) [MI_SOL_DO.jpg]	2	1
08 - Treino AC	Modelo 02					3	1
08 - Treino AC	Comparação 02	ClickChaveS+	(S-) [MI_SOL_DO.jpg]	(Click 2589ms) (S+) [DO_MI_SOL.jpg]	(S-) [RE_FA_LA.jpg]	3	2
08 - Treino AC	Modelo 03					4	2
08 - Treino AC	Comparação 03	ClickChaveS-	(S-) [DO_MI_SOL.jpg]	(S+) [MI_SOL_DO.jpg]	(Click 3134ms) (S-) [RE_FA_LA.jpg]	4	2
08 - Treino AC	Modelo 03					5	2
08 - Treino AC	Comparação 03	ClickChaveS+	(S-) [DO_MI_SOL.jpg]	(Click 2121ms) (S+) [MI_SOL_DO.jpg]	(S-) [RE_FA_LA.jpg]	5	3
08 - Treino AC	Modelo 04					6	3
08 - Treino AC	Comparação 04	ClickChaveS+	(S-) [RE_FA_LA.jpg]	(S-) [MI_SOL_DO.jpg]	(Click 4476ms) (S+) [DO_MI_SOL.jpg]	6	4
08 - Treino AC	Modelo 05					7	4
08 - Treino AC	Comparação 05	ClickChaveS+	(Click 2340ms) (S+) [MI_SOL_DO.jpg]	(S-) [RE_FA_LA.jpg]	(S-) [DO_MI_SOL.jpg]	7	5
08 - Treino AC	Modelo 06					8	5
08 - Treino AC	Comparação 06	ClickChaveS+	(S-) [RE_FA_LA.jpg]	(S-) [DO_MI_SOL.jpg]	(Click 4273ms) (S+) [MI_SOL_DO.jpg]	8	6
08 - Treino AC	Modelo 01					9	6
08 - Treino AC	Comparação 01	ClickChaveS+	(Click 2496ms) (S+) [DO_MI_SOL.jpg]	(S-) [RE_FA_LA.jpg]	(S-) [MI_SOL_DO.jpg]	9	7
08 - Treino AC	Modelo 02					10	7
08 - Treino AC	Comparação 02	ClickChaveS+	(S-) [MI_SOL_DO.jpg]	(Click 5008ms) (S+) [DO_MI_SOL.jpg]	(S-) [RE_FA_LA.jpg]	10	8
						10	8

Figura 4. Parte de um arquivo de resultados gerado pelo CP com as informações para um bloco de tentativas sobre a ocorrência de resposta no estímulo correto (ClickChaveS+) ou incorreto (ClickChaveS-), a função (S+ ou S-) e nome do estímulo programado em cada chave, o número da tentativa e os acertos acumulados.

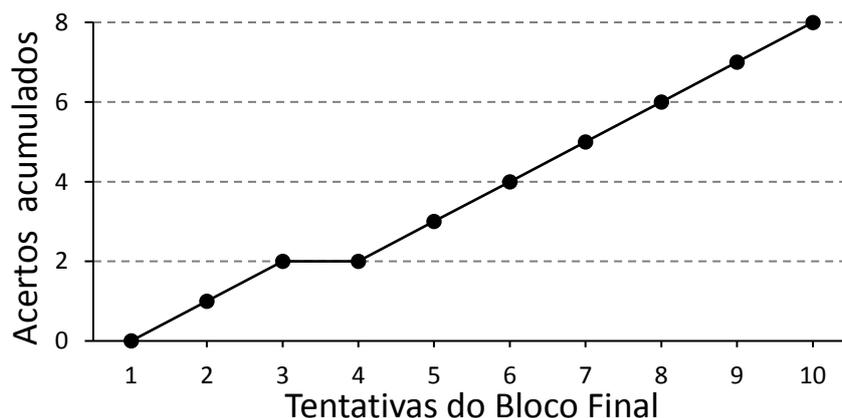


Figura 5. Frequência acumulada de respostas no S+ ao longo das tentativas do treino discriminativo.

REFERÊNCIAS

- Batitucci, J. S. L. (2007). *Paradigma de equivalência de estímulos no ensino de leitura de seqüências de notas musicais*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/2804>.
- Batitucci, L. A. V., Batitucci, J. S. L. & Hanna, E. S. (2006). *Contingência Programada*. Software não registrado. Brasília.
- Capobianco, D., Teixeira, C., Bela, R. E., Orlando, A. F., de Souza, D. G., & de Rose, J. C. (2009). LECH-

- GEIC. Sistema web gerenciador de ensino individualizado por computador. Recuperado em <http://geic.ufscar.br:8080/site/>
- de Paula, R. M. (2013). *Avaliação da utilização de fragmentos melódicos com e sem significado sobre a formação de classes, o desempenho recombinaivo e o tocar teclado*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/13116>.
- Dube, W. V., & Hiris, E. J. (1996). MTS v 11.6 [Software for research and education]. Waltham, MA: E. K. Shriver Center for Mental Retardation.
- Fernandes, A. M. A. A. (2013). *Avaliação de um procedimento de ensino de leitura adaptado a alunos com transtorno global do desenvolvimento*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/13655>
- Filgueiras, J. T. Q. (2011). *Efeitos do treino discriminativo com resposta de seleção ou de tocar teclado sobre a leitura musical*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/8862>.
- Harrison, J. M. (1991). Stimulus control. In I. H. Iverson & K. A. Lattal (Eds.), *Experimental Analysis of Behavior*, Part 1 (pp. 251-299). Amsterdam: Elsevier.
- Huber, E. R. (2010). *Avaliação do ensino cumulativo de relações entre estímulos musicais sobre a formação de classes, o desempenho recombinaivo e o tocar teclado*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/7758>.
- Mackay, H. A. (1991). Conditional stimulus control. In I. H. Iverson & K. A. Lattal (Eds.), *Experimental Analysis of Behavior*, Part 1 (pp. 301-350). Amsterdam: Elsevier.
- Martins, G. T. (2010). *Avaliação de dois protocolos de ensino de relações de identidade generalizada em indivíduos com atraso severo no desenvolvimento*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/5855>.
- Oliveira, A. (2012). *Efeito de reforço específico na formação de classes de estímulos a partir de discriminações simples entre estímulos compostos e procedimento com mancha redundante*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/10432>.
- Oliveira, J. M. (2014). *Mudanças no treino discriminativo de pseudopalavras e seus efeitos sobre a observação dos estímulos e o controle pelas letras*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/15879>.
- Pereira, E. S. (2012). *Discriminação de Diferença de Frequência de Sons e Aprendizagem de Leitura Musical*. Dissertação aprovada pelo Programa de Pós-Graduação em Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/10457>.
- Verneque, L. (2011). *Aprendizagem de Frações Equivalentes: Efeito do Ensino de Discriminações Condicionais Minimizando o Erro e da Possibilidade de Consulta a Dicas*. Tese aprovado pelo Programa de Pós-Graduação de Ciências do Comportamento, Universidade de Brasília, Brasília. Recuperado em <http://repositorio.unb.br/handle/10482/8635>.