

Capítulo

3

Game Development for Researchers: From Concept to User Experience Data Collection

Marcio Maestrello Funes¹, Leandro Agostini do Amaral¹,
Rudinei Goularte¹, Renata Pontin Mattos Fortes¹

¹Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação (ICMC/USP)

{marciofunes, leandroagostini}@usp.br, {rudinei, renata}@icmc.usp.br

Abstract

This chapter aims to present game development for researchers who wish to analyze user experience by means of data collection in their research, but find difficulties in the development of their applications. The chapter is divided into two parts: the first one presents game design theory highlighting definitions and examples about data collection involving guidelines and usability, the second one presents a game development approach by means of the Construct 2 tool using multimedia resources (audio, videos and photos) and variables (quantitative and qualitative) aiming to store data generated during users interaction.

Resumo

Este capítulo visa apresentar o desenvolvimento de jogos para pesquisadores que desejem analisar a experiência do usuário por meio da coleta de dados em suas pesquisas mas encontram dificuldades no desenvolvimento de suas aplicações. O capítulo está dividido em duas partes: a primeira parte apresenta a teoria sobre a concepção de jogos destacando as definições e exemplos sobre a coleta de dados envolvendo diretrizes e usabilidade, a segunda parte apresenta uma abordagem para o desenvolvimento de jogos por meio da ferramenta Construct 2 utilizando recursos multimídia (áudio, vídeos e fotos) e variáveis (quantitativas e qualitativas) visando o armazenamento dos dados gerados durante a interação do usuário.

3.1. Introdução

Desenvolvimento de jogos se refere a uma área de conhecimentos que abrange diversas linhas de estudos em Computação, Educação, *Design*, Simulação, Entretenimento, entre

outras. Por envolver diversas possibilidades de integração de conteúdos e modos de interação avançados, muitas pesquisas podem se beneficiar de seus mecanismos para atender demandas específicas dos procedimentos metodológicos científicos, em especial para coleta de dados.

Acredita-se que a pesquisa científica possa obter muitas vantagens a partir da utilização de jogos [Covert et al. 2017], porém este cenário se configurava, por bastante tempo, como de difícil acesso, por apresentar muitos obstáculos que tornaram tímida a imersão dos jogos no campo de pesquisas científicas, incluindo custos de desenvolvimento exorbitantes e representações gráficas inexatas que limitavam a fidelidade de simulação. Os avanços nas *engines* de jogos (por exemplo, CryEngine¹, Unity² e Unreal Engine³) diminuíram muito os problemas associados a sua adoção. Assim, é hora de considerar seriamente a utilidade de *serious games* para a área de pesquisa [Covert et al. 2017].

Diante dos inúmeros domínios de conhecimento que abordaram o tema de jogos e atendo-se ao segmento que detém o entretenimento como segundo plano, emerge a denominação *serious games*. De modo geral, a expressão “*serious games*” (em Português, jogos sérios) é adotada com o significado de que os jogos desse tipo possuem uma motivação educacional explícita, cuidadosamente pensada, e não se destinam a ser jogados exclusivamente para diversão ou entretenimento. Porém, essa interpretação não significa que jogos sérios não sejam, ou não devam ser, divertidos; ao contrário, a ideia é de que jogos sérios possuam elementos criativos e agradáveis como seus componentes.

Os *serious games* são projetados para ter um impacto no público-alvo, que está além do entretenimento puro [Gee 2003, Greitzer et al. 2007, Michael and Chen 2005]. Um dos domínios de aplicação mais importantes está no campo da Educação, dado seu potencial reconhecido, tendo em conta a necessidade contínua de aprimoramento das técnicas e atualização de recursos educacionais [Frederik et al. 2010, Van Eck 2006].

Outra iniciativa mais recente para uso dos jogos sérios é sua aplicação com o público idoso. Esse público representa uma comunidade de potenciais usuários que podem se beneficiar muito de jogos digitais [Souza and Trevisan 2014], por representarem uma ferramenta que contribui para a qualidade de vida durante o processo de envelhecimento, amenizando o declínio de alguns aspectos decorrentes deste processo, como aspectos motores, perceptivos, cognitivos e psicossociais [Chen et al. 2012].

Nesse contexto, segundo [Valladares-Rodríguez et al. 2016], uma das áreas mais promissoras para aplicação de jogos sérios voltados para o público idoso é a avaliação neuropsicológica. Os jogos podem revolucionar a avaliação cognitiva dos idosos em ambientes clínicos, permitindo que as avaliações sejam mais frequentes, mais acessíveis e mais agradáveis [Tong et al. 2016]. Em uma pesquisa sobre diferentes momentos e situações clínicas dos idosos, os estudos de [Gonçalves 2012] revelam que *serious games* têm se mostrado muito importantes para manter e desenvolver as competências cognitivas e sociais dos idosos. Na literatura especializada, estudos apontam que jogos sérios para idosos podem ser uma opção para melhorar a deterioração das funções cognitivas [Torres

¹<http://www.crytek.com/cryengine>

²<https://unity3d.com/pt>

³<https://www.unrealengine.com/>

2011], a manutenção da autoimagem, das funções motoras [Wiemeyer and Kliem 2011] e o convívio social (afeto e solidão) [Jung et al. 2009].

Diante do potencial relevante das pesquisas envolvendo jogos, este capítulo agrega conceitos sobre a utilização de jogos em pesquisa; em especial, é apresentado o procedimento de pesquisa de coleta de dados adotando-se jogos, e exemplos práticos de desenvolvimento.

3.2. Termos e Definições

São muitos os termos e definições utilizados em pesquisa para desenvolvimento e aplicações de jogos. Assim, neste capítulo fazemos um recorte, considerando os conceitos da área de jogos mais relacionados com a atividade de coleta de dados em ambiente de pesquisa, seja na produção de jogos completos ou no uso de elementos de *gamificação*, explicados mais adiante, para contribuir no engajamento do usuário dos jogos.

O engajamento do jogador está associado diretamente aos elementos de construção de um jogo, como também ao enriquecimento da experiência do usuário. Diante da diversidade de habilidades dos usuários (tanto limitações, quanto capacidades intensificadas), é essencial o uso de diretrizes que contribuam para a inclusão do maior número de especificidades, contribuindo com sua usabilidade e acessibilidade.

3.2.1. Jogos & Coleta de Dados

A utilização de jogos no ambiente de pesquisa visa enriquecer o objetivo fim de pesquisas aplicadas, em especial no contexto de coleta de dados, na obtenção de evidências científicas que promovam comprovação a questionamentos sob investigação. Assim, uso de jogos por pesquisadores não está associado apenas ao planejamento e desenvolvimento de jogos na forma de produtos acabados, ou completos, mas também no uso de elementos de *gamificação*, de modo a oferecer suporte às tarefas requeridas durante as atividades desenvolvidas na pesquisa.

Entende-se por *gamificação*, o uso de mecânicas, ideias e estética de jogos para engajar pessoas, motivar ações, promover o aprendizado e solucionar problemas [Kapp 2012]. Portanto, ao realizar *gamificação*, faz-se uso de elementos e técnicas de *design* de jogos em situações fora do ambiente de jogos, com o intuito de obter maior participação e envolvimento do usuário [Pedro 2016].

No contexto de auxílio computacional a pesquisadores, o conceito por trás do termo “jogo” pode ser resumido, conforme definido por [Salen and Zimmerman 2004]: refere-se a um sistema, no qual jogadores se envolvem em um conflito artificial, definido por regras e que possui um resultado quantificável.

De modo geral, o desenvolvimento de jogos fundamenta-se em determinados elementos, que podem caracterizar e até mesmo gerar o engajamento do usuário, tais como: o sistema, os jogadores, abstração, desafio, regras, interatividade, *feedback*, resultados quantificáveis, reações emocionais e história [Kapp 2012]. Esses elementos, definidos a seguir, devem ser considerados no âmbito de quaisquer tipos de jogos.

- **Sistema:** conjunto de elementos interligados que ocorrem no contexto do jogo;

- **Jogadores:** pessoas interagindo com o jogo ou com outras pessoas;
- **Abstração:** refere-se a uma imagem mental da realidade para definir o contexto do jogo;
- **Desafio:** o jogo incorpora atividades e objetivos que instigam os jogadores a atingirem metas e resultados em diferentes níveis de dificuldade;
- **Regras:** compreendem a estrutura de fluidez do jogo, definindo o ambiente para uma jogabilidade adequada;
- **Interatividade:** é o processo de relação do jogador com o conteúdo, com o jogo, ou com outros jogadores;
- **Feedback:** retorno do sistema às ações realizadas pelos jogadores;
- **Resultados quantificáveis:** o jogo deve reproduzir de maneira clara, ao usuário, o conceito de ganho ou perda, a partir da interação com o jogador;
- **Reações emocionais:** o jogo deve proporcionar sensações de prazer ou frustração, dependendo do resultado das atividades pelo jogador;
- **História:** o uso da narrativa de uma história contextualiza o jogador ao cenário, introduzindo um significado para o jogo.

Em especial, no contexto de coleta de dados, jogos propiciam que formas inovadoras de aplicação de seus elementos possam ser utilizadas nos procedimentos para coletar dados de modo isento e criterioso, conforme rigor prescrito nas investigações científicas. Na perspectiva de métodos de pesquisa em geral, tem-se que a coleta de dados é considerada uma das fases mais importantes em uma pesquisa [Pádua 2000], [Marconi and Lakatos 2001], [Marconi and Lakatos 2002], [da Costa and da Costa 2009], seja ela qualitativa ou quantitativa. Entre os principais instrumentos utilizados durante a coleta de dados, tem-se:

- 1. Observação** – quando os fatos são percebidos diretamente pelo pesquisador, sem qualquer intermediação. Deve ser planejada, registrada, submetida a verificação e controles de validade e precisão.
Entre as desvantagens, tem-se: a presença do pesquisador pode provocar alterações no comportamento dos observados.
- 2. Entrevista** – requer formulação de perguntas ao investigado, tratando-se de uma interação social. Possibilita a obtenção de dados referentes a um espectro de informações. Dentre as vantagens: proporciona obtenção de esclarecimentos e captura das expressões dos entrevistados (corporal, gestos, voz...).
Entre as desvantagens, tem-se: pode sofrer influência das opiniões pessoais do entrevistador, além de envolver custos para treinamento e aplicação das entrevistas.
- 3. Questionário** – conjunto de questões apresentadas por escrito aos investigados, permite que um grande número de respondentes possa participar / colaborar.

Entre as desvantagens, tem-se: exclusão dos que não sabem ler e escrever, além de requerer um tratamento diferenciado para as pessoas com deficiência visual, por exemplo; impede o conhecimento das circunstâncias do respondente, quando respondeu ao questionário.

4. Documentos – são fontes em “papel” (arquivos, registros estatísticos, diários, biografias, jornais, revistas, relatórios, cartas memórias, boletins...) que armazenam dados diversos, os quais possibilitam aquisição de conhecimentos ali representados.

Entre as desvantagens, tem-se: requer minucioso rigor na obtenção e interpretação do material juntado.

Vale destacar que esses instrumentos podem ser utilizados de modo complementar, conforme o tipo da pesquisa. Em qualquer tipo de levantamento (coleta de dados) com o público-alvo de uma pesquisa, exige-se o máximo de cuidado no momento de coletar informações. Dentre os cuidados, no contexto do tema apresentado neste capítulo, os elementos de jogos devem ser adequadamente projetados, conforme discutimos nas seções a seguir.

3.2.2. Experiência do usuário & Jogos

Dentre as definições para a expressão “*User experience (UX)*” (em Português Experiência do Usuário - EU), muitos pesquisadores convergem, considerando que UX abrange todos os aspectos da interação do usuário final com determinado sistema, serviço ou produto [Hassenzahl and Tractinsky 2006], [Goodman et al. 2013].

Para [Goodman et al. 2013], um aspecto relevante a ser considerado no desenvolvimento de jogos, relacionado a experiência do usuário, é que eles podem ser tediosos, quando muito fáceis, e frustrantes, quando muito difíceis. A maioria dos jogos *single-player* (jogador sozinho) permite ao jogador apenas um ajuste grosseiro de nível de dificuldade, geralmente possibilitando a escolha entre os modos de jogo: fácil, médio, difícil ou muito difícil [Hunicke 2005]. Tal abordagem mostra-se estática frente à interação do jogador com o jogo e pode apresentar uma divergência entre o nível de habilidade do jogador e a dificuldade geral do jogo.

Poucos desenvolvedores de jogos comerciais implementaram sistemas de ajuste dinâmico para seus jogos, dentre estes, pode-se citar God Hand® (Clover Studio e Capcom, 2006), no qual um medidor regula, ao longo do jogo, a inteligência e força dos adversários. Este medidor aumenta sempre que o jogador consegue, com sucesso, se esquivar de ataques inimigos ou golpear seus oponentes, diminuindo a medida que o jogador é atingido. Uma grande recompensa é oferecida para os jogadores que vencem oponentes que, de acordo com o medidor, são muito difíceis para seu nível de habilidade. Entretanto, nos jogos comerciais, no campo dos *serious games*, não foi identificada nenhuma iniciativa neste sentido.

Não tão distante da adaptabilidade dos jogos aos diferentes níveis de habilidades, a experiência do usuário pode estar relacionada também com questões de interface. Tais questões são relativamente mais complexas para pessoas (potenciais jogadores) que apresentem algum declínio nas suas funções cognitivas ou motoras, incentivando pesquisas na concepção de jogos planejados para uso universal, para todos.

É importante ponderar sobre a perspectiva de se desenvolver um software tradicional, que não seja um jogo, considerando-se a experiência de usuário. O desenvolvimento com UX para software tradicional visa a remoção de problemas ao usuário. Por outro lado, no desenvolvimento de um jogo, a experiência do usuário se refere a oferecer “problemas” / desafios ao usuário. Em ambos os casos, deve-se examinar o raciocínio exigido dos usuários e sua capacidade cognitiva.

Em geral, o trabalho de um desenvolvedor que se preocupa com a experiência do usuário, de software tradicional, é direcionado para criar uma ótima experiência para o usuário ao usar o produto para um propósito. Assim, esse desenvolvedor faz o máximo possível para que a interface seja transparente (o usuário precise pensar o mínimo possível), sugestiva (o usuário saiba quais possibilidades de interação ele possui), flexível (que se adapte à medida que o usuário desenvolve habilidades), rica em informações e *feedback* (o usuário saiba mais sobre algo ou saiba situações que lhe traga problemas). Trata-se de conduzir os usuários a construir um modelo mental que os ajude a fazer o seu trabalho. Exemplos clássicos: o do volante de um carro ou a metáfora da área de trabalho inteira. O volante está escondendo do usuário o fato de que existe um mecanismo bastante complexo entre a roda e os pneus. Em vez disso, está tentando levar o usuário a pensar “girar a roda, virar o carro”. Esta é uma ilusão intencional. O trabalho dos desenvolvedores de jogos, geralmente, é voltado a criar uma ótima experiência ao jogador ao jogar um jogo. Portanto, eles fazem o máximo possível para criar o jogo:

- **Desafiador** – muitas vezes, é requerido que o jogo faça com que o usuário pense, com atenção, use a memória e sua capacidade de raciocínio.
- **Explorável** – geralmente, é requerido que o usuário pense que há sempre mais possibilidades a serem descobertas.
- **Escalável** – para que os jogadores aprendam a jogar melhor, enquanto jogam.
- **Surpreendente** – para que os jogadores sejam cativados pelo espetáculo que o jogo apresente. Com esta relação de empolgação, tem-se inclusive um convite ao erro pois, muitas vezes, é requerido que os jogadores aprendam através de seus erros.

Finalmente, enquanto a experiência do usuário em software tradicional visa claramente ocultar toda a complexidade do produto, em jogos UX visa claramente ensinar a lidar com a complexidade. Ainda assim, acessibilidade em jogos é requisito de qualidade importante, e na próxima seção encontra-se descrito.

3.2.3. Diretrizes de acessibilidade para jogos

Com a finalidade de produzir uma referência objetiva aos desenvolvedores de jogos, e no sentido de incluir o maior número de pessoas possível, considerando as diversas necessidades apresentadas por pessoas com deficiências, um esforço colaborativo entre produtoras, especialistas e pesquisadores foi realizado, surgindo as diretrizes de acessibilidade para jogos [Ellis et al. 2012].

As diretrizes não devem ser utilizadas apenas durante o processo de desenvolvimento do jogo, mas também na avaliação contínua da aplicação, por especialistas e

usuários finais. Diante da variedade de abordagens existentes, no tocante à avaliação de acessibilidade para jogos digitais, o uso de mais de um método de avaliação é importante para se obter maior qualidade nos resultados [Fortes et al. 2017].

As diretrizes foram categorizadas em três tipos: básico, intermediário e avançado. Nessas categorias são contempladas as diferenças quanto ao número de pessoas beneficiadas, o impacto provido para os utilizadores e o custo de implementação.

As diretrizes são também agrupadas em subcategorias, relacionadas com as diferentes habilidades: motora, cognitiva, visual, auditiva e de fala. A seguir, esses **Conjuntos de Diretrizes de Acessibilidade para Jogos (CDAJ)** são apresentados.

[CDAJ.1] para USO GERAL

- Básico
 - Fornecer detalhes sobre os **recursos de acessibilidade** no pacote e/ou jogo
 - Oferecer uma grande variedade de níveis de dificuldade
 - Certificar que todas as configurações são salvas/lembradas
- Intermediário
 - Permitir que o nível de dificuldade possa ser alterado durante o jogo, seja através de configurações ou dificuldade adaptativa
 - Incluir algumas **pessoas com deficiência** entre os participantes que testarão o jogo
 - Oferecer um meio de contornar elementos de jogabilidade que não fazem parte da mecânica central, através das definições ou uma opção de ignorar durante o jogo
 - Incluir modos auxiliares, tais como mira automática e direção assistida
 - Fornecer possibilidade manual de salvar
 - Fornecer a possibilidade de salvar automático
 - Permitir uma preferência para ser definida a jogabilidade *multiplayer* online, com ou sem outros que estão usando recursos de acessibilidade, que poderiam dar uma vantagem competitiva
- Avançado
 - Incluir **todas as categorias relevantes de deficiência** (motora, cognitiva etc.) entre os participantes dos testes de jogo, em números representativos com base na idade e dados demográficos do público-alvo
 - Permitir que o jogo possa ser aperfeiçoado, expondo tantas variáveis quanto possível
 - Permitir salvar configurações de diferentes perfis, em qualquer jogo ou plataforma

[CDAJ.2] para usuários com DEFICIÊNCIA MOTORA

- Básico
 - Que os controles possam ser remapeados / reconfigurados
 - Certificar que todas as áreas da interface do usuário possam ser acessadas usando o mesmo método de entrada com mesma jogabilidade
 - Incluir uma opção para ajustar a sensibilidade dos controles
 - Garantir controles tão simples quanto possível, ou fornecer uma alternativa mais simples
 - Certificar que os elementos interativos / controles virtuais são grandes e bem espaçados, especialmente em telas pequenas ou sensíveis ao toque
- Intermediário
 - Suportar mais do que um dispositivo de entrada
 - Faça elementos interativos que exigem precisão (ex. cursor/toque controlados pelas opções do menu) estacionária
 - Certificar que várias ações simultâneas (ex. clicar/arrastar ou deslizar) não sejam necessárias, e incluir apenas como um método de entrada alternativa / complementar
 - Certificar que todas as ações chave possam ser realizadas por controles digitais (pad / chaves / ação de pressionar), com mais de uma entrada complexa (ex. discurso, gesto) não obrigatória, e incluir apenas como métodos de entrada suplementares/alternativas
 - Incluir uma opção para ajustar a velocidade do jogo
 - Evitar entradas repetidas (eventos em tempo rápido)
 - Se produzir um jogo para PC, apoiar o modo de janela para compatibilidade com teclados virtuais sobrepostos
 - Evitar/fornecer alternativas de exigência de botões a serem pressionados
 - Permitir que as interfaces possam ser reorganizadas
 - Permitir que as interfaces possam ser redimensionadas
- Avançado
 - Permitir que o jogo possa ser colocado em modos retrato e paisagem
 - Não faça que o sincronismo preciso seja essencial para a jogabilidade - oferecer alternativas, ações que possam ser realizadas durante a pausa, ou um mecanismo de salto
 - Incluir um período (pós atraso de aceitação) de 0,5 segundos entre as entradas
 - Fornecer esquemas de controle muito simples que são compatíveis com dispositivos de Tecnologia Assistiva, tais como interruptor ou *eye tracking*

[CDAJ.3] para usuários com DEFICIÊNCIA COGNITIVA

- Básico
 - Permitir que o jogo seja iniciado sem a necessidade de navegar através de múltiplos níveis de menus
 - Usar um tamanho de fonte padrão de fácil leitura
 - Usar uma linguagem simples e clara
 - Usar a formatação de texto simples e clara
 - Incluir tutoriais
 - Permitir aos jogadores que possam progredir através de interface de texto ao seu próprio ritmo
 - Evitar piscar imagens e padrões repetitivos

- Intermediário
 - Incluir ajuda contextual no jogo / orientações / dicas
 - Indicar / permitir lembrete dos objetivos atuais durante o jogo
 - Indicar / permitir lembrete de controles durante o jogo
 - Incluir um meio de praticar sem falhas, tais como um nível de prática ou *sandbox*
 - Empregar uma estrutura narrativa simples, clara
 - Se estiver usando uma narrativa global longa, fornecer resumos de progresso
 - Garantir que não haja informações essenciais (especialmente instruções) que sejam transmitidas apenas via texto, reforçar com recursos visuais e/ou voz
 - Dar uma clara indicação de que elementos interativos são interativos
 - Fornecer uma opção para desligar/ocultar movimento de fundo
 - Possibilitar chat via voz e texto para jogos *multiplayer*
 - Fornecer imagens do jogo (*thumbnails*) com os jogos salvos
 - Fornecer controles de volume separados ou de mudo para efeitos, fala e música de fundo
 - Certificar que escolhas de som/música para cada objeto-chave / eventos são distintos um do outro
 - Incluir uma opção para ajustar a velocidade do jogo
 - Fornecer uma escolha de cor de texto, escolha de baixo/alto contraste, no mínimo

- Avançado
 - Fornecer dublagens pré-gravadas para todo o texto, incluindo menus e instaladores

- Evitar qualquer movimento ou eventos súbitos
- Permitir que todas as narrativas e instruções possam ser repetidas
- Usar chat baseado em símbolo (*smileys* etc.)
- Fornecer uma opção para desligar/ocultar todos os elementos não interativos

[CDAJ.4] para usuários com DEFICIÊNCIA VISUAL

- Básico

- Garantir que não haja nenhuma informação essencial transmitida por cores apenas
- Se o jogo utiliza o campo de visão (*engine* 3D apenas), definir um padrão apropriado para o ambiente de visualização esperado
- Usar um tamanho de fonte padrão de fácil leitura
- Usar a formatação de texto simples e clara
- Fornecer alto contraste entre o texto e o fundo
- Certificar que elementos interativos/controles virtuais são grandes e bem espaçados, especialmente em telas pequenas ou sensíveis ao toque

- Intermediário

- Se o jogo utiliza o campo de visão (*engine* 3D apenas), permitir um meio para que ele possa ser ajustado
- Evitar (ou fornecer opção para desativar) qualquer diferença entre o movimento do controlador e o movimento da câmera, como arma/balanço do caminhar ou a suavização do mouse
- Usar o som *surround*
- Fornecer uma opção para desligar/esconder a animação de fundo
- Garantir o apoio do leitor de tela para dispositivos móveis
- Fornecer uma opção para ajustar o contraste
- Certificar escolher som/música para objetos-chave / eventos são distintos um do outro
- Fornecer uma escolha de cursor / conjunto de cores / design
- Dar uma clara indicação de que elementos interativos são interativos
- Assegurar que o manual / website são fornecidos em um formato amigável ao leitor de tela
- Fornecer controles de volume separados ou mudos para efeitos, de fala e de música de fundo
- Evitar colocar informações temporárias essenciais fora da linha do olho do jogador

- Permitir interfaces redimensionáveis
- Avançado
 - Permitir que o tamanho da fonte seja ajustado
 - Fornecer um mapa de áudio *pingable*, estilo sonar
 - Fornecer dublagens pré-gravadas para todo o texto, incluindo menus e instaladores
 - Fornecer um GPS sonoro
 - Permitir orientação fácil / movimento ao longo dos pontos cardeais
 - Certifique-se de que todas as ações chave possam ser realizadas por controles digitais (pad / chaves / ação de pressionar), com mais de uma entrada complexa (ex. discurso, gesto) não obrigatória, e incluir apenas como métodos de entrada suplementares/alternativas
 - Garantir o apoio do leitor de tela, incluindo menus e instaladores
 - Use design de som / música distinta para todos os objetos e eventos
 - Simular a gravação binaural

[CDAJ.5] para usuários com DEFICIÊNCIA AUDITIVA

- Básico
 - Fornecer controles de volume separados ou mudos para efeitos, de expressão e fundo de música
 - Garantir que não haja informação essencial transmitida apenas por áudio, reforçar com texto / visual
 - Se quaisquer legendas forem fornecidas, utilize um tamanho de fonte padrão de fácil leitura, formatação de texto simples e clara e forneça alto contraste entre o texto e o fundo
- Intermediário
 - Manter o ruído de fundo mínimo durante o discurso
 - Fornecer uma alternativa em texto para toda a fala (legendas)
 - Permitir alternativas em texto a serem exibidas antes de qualquer som reproduzido
 - Fornecer uma descrição em texto da narrativa / atmosfera significativa, com sons de fundo
 - Fornecer uma indicação visual de quem está falando
 - Possibilitar chat via voz e texto para jogos *multiplayer*
 - Fornecer meios visuais de comunicação no *multiplayer*

- Permitir uma preferência para ser definida a jogabilidade *multiplayer* online, com ou sem outros que estão usando recursos de acessibilidade que poderiam dar uma vantagem competitiva
- Garantir que todas as informações suplementares importantes (por exemplo: a direção que você está sendo filmado) veiculada por áudio são reproduzidas em texto / visuais
- Proporcionar uma alternância entre estéreo / mono
- Avançado
 - Certificar que as legendas são cortadas e apresentadas em uma frequência apropriada para a faixa etária alvo
 - Fornecer cadastro

[CDA.6] para usuários com DEFICIÊNCIA de FALA

- Básico
 - Certificar que a entrada de fala não é necessária, e é incluída apenas como um método de entrada alternativa / complementar
- Intermediário
 - Permitir uma preferência para ser definida a jogabilidade *multiplayer* online, com ou sem outros que estão usando recursos de acessibilidade que poderiam dar uma vantagem competitiva
 - Possibilitar *chat* via voz e texto para jogos *multiplayer*
 - Fornecer meios visuais de comunicação no *multiplayer*
 - Reconhecimento de fala, com base em palavras individuais, a partir de um pequeno vocabulário (ex. 'sim', 'não', 'abrir') em vez de frases longas ou palavras multi-silábicas
- Avançado
 - Para o reconhecimento de fala tomar por base chegar a um limite de volume (por exemplo, 50%) em vez de palavras

Tais diretrizes norteiam a composição de jogos adaptados às inúmeras dificuldades inerentes às pessoas com deficiência e mesmo aos idosos, que muitas vezes apresentam declínios em suas funções cognitivas e motoras.

3.3. Coletando dados por meio de um jogo: Um exemplo

O estudo de caso aqui descrito exemplifica uma coleta de dados realizada por meio de jogos. Nesse exemplo, a coleta de dados desempenha papel de vital importância na pesquisa e foi realizada como parte de uma investigação sobre a conversão de baterias de testes cognitivos para o meio digital. O aplicativo de teste foi desenvolvido na forma de um jogo. O teste cognitivo digital foi submetido a um experimento com usuários, com base nas heurísticas de usabilidade [de Lima Salgado et al. 2017]. Heurísticas são processos que visam encontrar problemas que recorrentemente ocorrem, e em geral, são reconhecidas as possíveis soluções.

Em virtude de o usuário final do teste cognitivo digital ser o público idoso, foram consideradas questões de acessibilidade desde seu planejamento. Um exemplo foi a escolha da paleta de cores, a qual foi feita respeitando sua compreensão por indivíduos que apresentem daltonismo, mais especificamente nas 3 possíveis situações⁴: Tritanopia, Protanopia e Deuteranopia.

A ferramenta Color Oracle⁵ foi utilizada para averiguar a paleta nas três situações e então foram eliminadas cores que em um ou mais casos se aproximavam muito umas das outras, se tornando quase indistinguíveis. Mesmo assim ainda é necessária cautela ao confeccionar os *assets* do jogo, isto é, *feedbacks* e objetivos não podem contar apenas com a cor para diferenciar objetos. Sons e formas também devem ser utilizados.

Na bateria de testes cognitivos, foram tratados oito aspectos relacionados com as exigências intelectuais dos usuários, a saber: **(Teste 01)** - Percepção Visual, Memória Incidental e Memória Imediata, **(Teste 02)** - Praxia, **(Teste 03)** - Abstração, **(Teste 04)** - Teste de Extensão de Dígitos (Direto), **(Teste 05)** - Teste de Extensão de Dígitos (Inverso), **(Teste 06)** - Atenção, **(Teste 07)** - Teste de Fluência Verbal e **(Teste 08)** - Reconhecimento Facial. A partir da definição desses aspectos, buscou-se uma correlação entre eles, a fim de se identificar / comprovar o nível de alcance que a portabilidade para o meio digital permitiria, sem que houvesse uma alteração significativa no contexto.

As coletas de dados no Estudo de Caso foram realizadas, tendo um conjunto de variáveis catalogadas, referentes aos aspectos relacionados com as cargas cognitivas requeridas dos usuários. A seguir, são listadas variáveis catalogadas, que exemplificam como foi planejado o experimento com usuários usando o jogo (jogando).

1. **(Teste 01) - Percepção Visual, Memória Incidental e Memória Imediata**

- (a) Quantidade de Objetos: número de objetos que aparecem na tela simultaneamente. Esta variável pode ter impacto direto no desempenho relativo de um mesmo jogador. Por exemplo, uma iteração com menos objetos na tela pode ter melhores resultados.
- (b) Quantidade de Formas: número de diferentes formas que aparecem na tela. Esta variável pode ter impacto direto no desempenho relativo de um mesmo

⁴Os tipos de Daltonismo analisados foram Tritanopia - insensibilidade às ondas curtas (os azuis); Protanopia - insensibilidade às ondas longas (os vermelhos); Deuteranopia - insensibilidade às ondas médias (os verdes)

⁵<http://www.colororacle.org/>

jogador. Por exemplo, um teste onde apareçam triângulos e quadrados pode ter resultados inferiores a um em que apareçam apenas quadrados.

- (c) Elementos de Confusão: número de objetos que serão inseridos para criar confusão, relevante apenas na versão anterior do teste, em que o jogador precisava marcar quais objetos eram diferentes da matriz memorizada. Esta variável pode ter impacto direto no desempenho relativo do mesmo jogador. Espera-se que em um cenário com mais elementos de confusão o jogador cometa mais erros.
- (d) *Score*: pontuação do jogador, de 0 a 100. Este dado armazena, apenas, quantos dos espaços foram preenchidos corretamente, ignorando espaços que foram preenchidos sem necessidade. Para contornar a possibilidade do jogador preencher toda a matriz o número de objetos que podem ser colocados foi limitado para o mesmo número da variável Quantidade de Objetos. Este dado é importante para se comparar a matriz inserida pelo jogador e a matriz original, avaliando se o jogador de fato memorizou a matriz original.
- (e) Tempo Total: soma do tempo desde que o jogador iniciou o teste até concluí-lo. Este dado pode ser um indicativo de dificuldade para interagir com a tela do tablet, dificuldade de compreensão do teste bem como limitações motoras ou distrações.
- (f) Cliques:
 - i. Forma Clicada: dado que armazena qual foi a forma tocada pelo usuário, relevante apenas em casos com mais de uma forma disponível. Este dado pode indicar dificuldades para interagir com a tela do tablet, dificuldade de compreensão do teste bem como limitações motoras.
 - ii. Cor: dado que armazena a cor do objeto tocado, relevante apenas em casos em que haverá mais de uma cor de objeto. Este dado pode indicar dificuldade para distinguir as cores ou dificuldade de compreensão do teste.
 - iii. Tempo: tempo entre o clique atual e o anterior. Este dado pode indicar dificuldades para interagir com a tela do tablet, dificuldade de compreensão do teste, bem como limitações motoras.
 - iv. Tipo de clique: dado que marca se o jogador tocou um botão, um objeto de interesse ou nenhum dos dois anteriores, configurando então um *mis-click*. Este dado pode indicar dificuldades para interagir com a tela do tablet, objetos muito pequenos e limitações motoras.
 - v. Acerto: dado que marca se o toque feito foi um acerto ou um erro, dentro das definições de acerto e erro do teste. Este dado pode indicar dificuldades para interagir com a tela do tablet, dificuldade de compreensão do teste bem como limitações motoras.

2. (Teste 02) - Praxia

- (a) *Score*: pontuação de 0 a 100, marcando quantos objetos foram encaixados da forma correta. Este dado armazena a avaliação das conexões dois a dois, com

pequena margem de erro para cada conexão, dividindo a pontuação máxima pelo número de conexões totais que serão verificadas. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.

- (b) Tempo Total: soma do tempo desde que o jogador iniciou o teste até concluí-lo. Este dado pode ser um indicativo de dificuldade para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras ou distrações.
- (c) Cliques:
 - i. Forma: a forma tocada pelo jogador. Este dado pode indicar que o jogador tem maior dificuldade de interagir com uma certa peça do Tangram, seja pelo seu tamanho ou formato.
 - ii. Cor: a cor da forma tocada pelo jogador, importante pois existem mais de uma cópia de algumas formas, mas de cores e/ou rotações diferentes. Este dado pode indicar que o jogador tem dificuldade em distinguir certas cores.
 - iii. Tempo: tempo entre o clique atual e o anterior. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.
 - iv. Tipo de clique: este dado marca se o jogador tocou um botão, um objeto de interesse ou nenhum dos dois anteriores, configurando então um *mis-sclick*. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, objetos muito pequenos e limitações motoras.

3. (Teste 03) – Abstração

- (a) *Score*: pontuação de 0 a 100, marcando quantas iterações do teste de abstração foram respondidas corretamente. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.
- (b) Tempo Total: soma do tempo desde que o jogador iniciou o teste até concluí-lo. Este dado pode ser um indicativo de dificuldade para interagir com a tela do *tablet*, dificuldade de compreensão do teste, bem como limitações motoras ou distrações.
- (c) Cliques:
 - i. Tempo: tempo entre o clique atual e o anterior. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.
 - ii. Tipo de clique: este dado marca se o jogador tocou um botão, um objeto de interesse ou nenhum dos dois anteriores, configurando então um *mis-sclick*. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, objetos muito pequenos e limitações motoras.

- iii. Acerto: este dado marca se o toque feito foi um acerto ou um erro, dentro das definições de acerto e erro do teste. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.

4. (Teste 06) – Atenção

- (a) Tamanho da Sequência: variável que armazena quão longa é a sequência apresentada na tela. Esta variável pode ter impacto direto no desempenho relativo de um mesmo jogador. Por exemplo, sequências mais longa podem causar maior confusão ao reposicionar os diversos números pela tela.
- (b) Total de Erros: dado que armazena quantas vezes o jogador tocou em um número que não era o próximo na sequência. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.
- (c) Tempo Total: soma do tempo desde que o jogador iniciou o teste até concluí-lo. Este dado pode ser um indicativo de dificuldade para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras ou distrações.
- (d) Cliques:
 - i. Número Pressionado: dado que armazena qual foi o dígito pressionado. Este dado pode indicar dificuldade de compreensão do teste e confusão com os números mostrados na tela.
 - ii. Tempo: tempo entre o clique atual e o anterior. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste bem como limitações motoras.
 - iii. Acerto: este dado marca se o toque feito foi um acerto ou um erro, dentro das definições de acerto e erro do teste. Este dado pode indicar dificuldades para interagir com a tela do *tablet*, dificuldade de compreensão do teste, distração bem como limitações motoras.

3.4. Desenvolvimento de jogos para coleta de dados utilizando *Construct 2*

Como pode ser observado no estudo de caso da Seção 3.3, apresentado como exemplo, é grande a quantidade de aspectos que podem ser estudados nos trabalhos científicos experimentais, bem como as suas variáveis catalogadas, que podem variar bastante dependendo do teste e do objetivo que se deseja alcançar ao realizar procedimentos de coleta de dados.

Muitas vezes, a falta de familiaridade com ferramentas, técnicas e linguagens utilizadas no desenvolvimento de jogos, bem como o desconhecimento dos elementos relacionados à *gamificação*, podem desencorajar profissionais e pesquisadores envolvidos com investigações experimentais a utilizarem formas alternativas para coleta de dados, em seus estudos de casos por exemplo. Abordagens envolvendo estudo de caso e jogos foram publicadas na literatura, recentemente [Barendregt et al. 2017] e [Sung and Berland 2017].

Segundo [Samodelkin et al. 2016], é cada vez mais importante explorar novas formas de interação, em particular, para coleta de dados de experiências de usuários. Nesse sentido, [Wood et al. 2017] afirmam que o desenvolvimento de jogos e o uso de *gamificação* permitem atingir uma parcela significativa de usuários, principalmente usuários mais jovens, uma vez que o contato direto com jogos pode ocorrer com maior frequência em seu dia a dia. A adoção de elementos de *gamificação* em estudos de casos, não necessariamente relacionados com a área de jogos, pode fornecer novas perspectivas de coleta de dados, novas experiências aos usuários e novos métodos para realização de pesquisas acadêmicas.

Nesse contexto, surge a necessidade de uma ferramenta que satisfaça aos seguintes requisitos: (a) ofereça diversas possibilidades de apoio ao desenvolvimento de jogos, à manipulação de elementos de *gamificação* de forma intuitiva, (b) que seja de fácil uso mesmo para pesquisadores sem experiências prévias com a área de jogos e, principalmente, (c) que possibilite o *design* de estudos de casos, bem como a coleta e armazenamento de variáveis para futuras análises.

A ferramenta *Construct 2* foi estudada, pois auxilia no desenvolvimento de jogos, e atende aos três requisitos especificados. De propriedade da empresa *Scirra*⁶, seu lançamento ocorreu no ano de 2011 e um exemplo de sua interface pode ser visualizada na Figura 3.1.

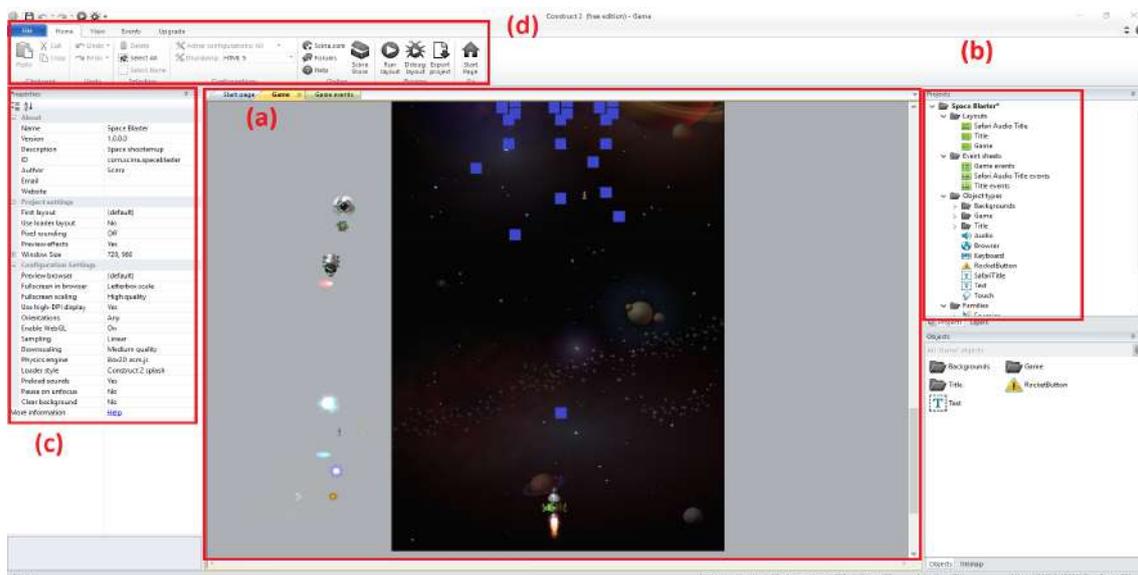


Figura 3.1. Visão geral da ferramenta *Construct 2*: (a) cenário, (b) lista de objetos (imagens, sons, controles), (c) painel de configuração de comportamentos e (d) menu de opções

Na Figura 3.1, a região em destaque rotulada por (a) mostra os elementos gráficos, chamados de *sprites*, que irão compor o jogo. Podem ser adicionados ao *layout* e manipulados, formando o cenário no qual o jogo irá ser executado. A região no destaque (b) mostra que é possível adicionar facilmente outros elementos como som, vídeo, controles

⁶<https://www.scirra.com/>

e etc. Na região de destaque (c), encontram-se as diversas opções de configuração que permitem ajustar o comportamento que cada elemento presente no jogo irá receber. Por fim, no destaque (d) está o menu principal que permite compilar e testar o jogo bem como realizar a exportação do projeto criado.

Uma das principais características da ferramenta *Construct 2* se refere a não impor ao desenvolvedor o conhecimento prévio de alguma técnica ou linguagem de programação, tornando-se assim uma ferramenta ideal para profissionais e pesquisadores sem experiência prévia no desenvolvimento de jogos. Sua programação ocorre por meio de eventos que informam ao sistema como será a relação e o comportamento que todos os elementos presentes no jogo devem/podem realizar, além da interação com variáveis e controles. Com essa ferramenta, também é possível exportar o jogo para múltiplas plataformas, como: *Web*, *mobile* e *desktop*, além de comunicar com banco de dados para armazenamento de variáveis. Na Figura 3.2 encontra-se um exemplo da programação por eventos, na ferramenta *Construct 2*.

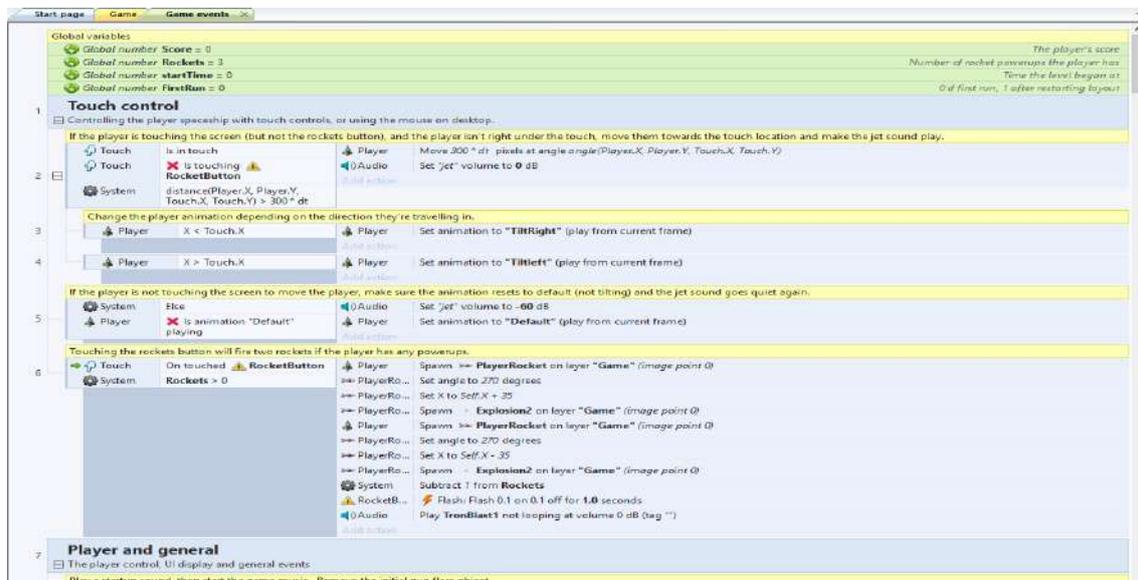


Figura 3.2. Interface que lista os eventos necessários para execução de um jogo em *Construct 2*

Devido ao foco deste capítulo estar em fornecer possibilidades de concepção de jogos orientados à coleta de dados, não serão abordados elementos básicos como o *download* e instalação. Assim, será apresentado um conjunto de exemplos nas Seções a seguir, os quais, a partir de uma abordagem para auxiliar pesquisadores no desenvolvimento de seus próprios experimentos, possibilita a adoção dos elementos de jogos. A abordagem está representada na Figura 3.3 e os elementos que a compõem são os seguintes:

1. **Design do experimento:** se refere à concepção dos elementos do jogo a ser construído, deverá ser realizada durante o desenvolvimento do jogo na ferramenta *Construct 2*, na fase inicial. Para auxiliar o pesquisador nesse *design* do seu experimento, deve-se considerar os conceitos teóricos sobre jogos, a experiência do usuário e diretrizes de acessibilidade para jogos, como apresentados na Seção 3.2. Na Seção

3.3 foi apresentado um exemplo de coleta de dados, por meio de heurísticas de usabilidade.

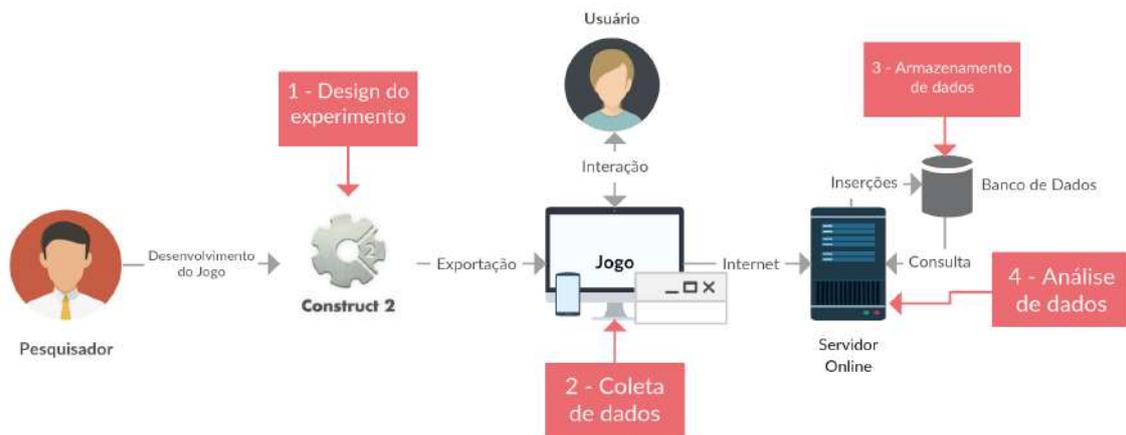


Figura 3.3. Abordagem para auxiliar pesquisadores no desenvolvimento de seus experimentos, envolvendo jogos e coleta de dados

2. **Coleta de dados:** ocorre durante a interação do usuário com o jogo, o qual pode ser projetado para multiplataformas (*Web, mobile e desktop*). A Seção 3.4.1 apresenta o uso de recursos multimídia para coleta de dados e a Seção 3.4.2 descreve a coleta de dados referente a variáveis quantitativas e qualitativas.
3. **Armazenamento de dados coletados:** para que o pesquisador possa realizar análises sobre os dados coletados, é importante que haja armazenamento dos dados em banco de dados hospedado em um servidor *online*. A Seção 3.4.3 descreve um jogo que utiliza diretrizes relativas a habilidade motora, para exemplificar o processo de armazenamento.
4. **Análise de dados:** após os dados coletados terem sido armazenados em um banco de dados, consultas aos dados podem ser efetuadas pelo pesquisador utilizando o servidor *online*. Como este tópico requer estudos mais aprofundados, recomenda-se as orientações de [MacKenzie 2012] para realização das análises estatísticas dos dados coletados.

3.4.1. Utilizando recursos multimídia na coleta de dados

3.4.1.1. Coleta do áudio do usuário para reconhecimento de voz

Na Seção 3.2.3 foram descritas diretrizes de acessibilidade que devem ser utilizadas também para melhorar a experiência do usuário. Dentre elas, utilizando como exemplo a habilidade de “Fala”, uma diretriz, de categoria intermediária, recomenda que: “o reconhecimento de fala deve ocorrer com base em palavras individuais, a partir de um pequeno vocabulário (ex. 'sim', 'não', 'abrir’), em vez de frases longas ou palavras multisilábicas”. Com o intuito de coletar dados relacionados com a habilidade de “Fala”, os seguintes eventos podem ser programados na ferramenta *Construct 2* como mostrados na Figura 3.4.



Figura 3.4. Programação em eventos sobre reconhecimento de fala

Após o projeto do jogo receber o objeto chamado *"UserMedia"*, os eventos listados na Figura 3.4 podem ser criados e manipulados. Na linha 1, o evento *"UserMedia is recognising speech"* define que o microfone deverá ser acessado sempre que o jogo for executado no navegador e fornece um *feedback* ao usuário com o texto *"Currently recognising speech..."*, o conteúdo do texto pode ser alterado ou substituído por uma imagem. Entre as linhas 2 e 4 são definidos os eventos, nos casos de não existir microfone ou o navegador não oferecer tal suporte.

Entre as linhas 5 e 7 está definido o evento principal de reconhecimento de fala, no qual existe a condição de clicar em um botão para iniciar ou pausar o reconhecimento. Na linha 7, a configuração *"Request speech recognition (language 'pt-br', Continuous mode, Interim results"* define o idioma utilizado no reconhecimento, bem como a configuração *"Continuous mode"* cumpre a diretriz, citada anteriormente, de reconhecer palavras individuais (para o reconhecimento de frases inteiras a configuração *"Single phrase mode"* deve ser utilizada). Na Figura 3.5 é apresentada a tela com o resultado do reconhecimento de fala.

3.4.1.2. Coleta do vídeo da tela de interação do usuário

Capturar a tela na qual ocorre a interação com um jogo pode ser um recurso valioso quando se trata de analisar a experiência do usuário. Segundo [MacKenzie 2012], a captura de tela é uma abordagem criativa para coletar dados, principalmente quando existe interação com algum *software*.



Figura 3.5. Resultado do reconhecimento de fala de palavras individuais

Um problema que pode inviabilizar a captura de tela, como recurso de coleta de dados, está nos *softwares* deste segmento, disponíveis no mercado. Em alguns casos, *softwares* especializados possuem licenças de uso, que podem inviabilizar pesquisas que não dispõem de verba ou exigem um nível de conhecimento específico para sua correta utilização.

A ferramenta *Construct 2* permite realizar a captura da tela, na qual o jogo está sendo executado, e possibilita, como saída, gerar um vídeo que pode ser gravado em formatos variados. A Figura 3.6 mostra a programação utilizada para tal objetivo.

1	UserMedia	Is canvas recording supported	Text	Append newline & "Canvas recording is not supported - try Firefox"
			Add action	
2	System	On start of layout	Record	Set Enabled
	UserMedia	Is canvas recording supported	Add action	
3	Record	On clicked	UserMedia	Start recording canvas (WebM VP8 @ 2500 kbps, 0 FPS)
			Record	Set Disabled
			Stop	Set Enabled
			Add action	
4	Stop	On clicked	UserMedia	Stop recording canvas
			Record	Set Enabled
			Stop	Set Disabled
			Add action	
5	UserMedia	On canvas recording ready	Browser	Invoke download of UserMedia.CanvasRecordingURL with filename "video.avi"
			Add action	

Figura 3.6. Programação de captura de tela na ferramenta *Construct 2*

Na linha 1 da Figura 3.6, é definido o teste de compatibilidade com o navegador; o *feedback* gerado no exemplo é dado de forma textual na interface, mas pode ser alterado.

A linha 2 informa ao sistema que, quando o jogo for iniciado ("*On start of layout*") e o navegador oferecer suporte para captura de tela, o botão de gravação ficará ativo. Na linha 3, é configurado o evento que inicia a gravação, que está condicionado ao clique de um botão. Na linha 4, é prevista a ação de interromper a gravação por meio de um botão e, na linha 5, é previsto o *download* imediato do resultado da captura em formato de vídeo que pode ter seu formato (avi, mp4 e etc.) alterado.

Outro recurso que a ferramenta *Construct 2* possui é o que permite realizar a gravação do usuário jogando, utilizando uma ação do usuário como condição. Na Figura 3.7, por exemplo, a gravação somente ocorre quando o usuário obtiver 20 pontos e será interrompida quando o usuário atingir 30 pontos.

Line	Event	Action
1	UserMedia: Is canvas recording supported	Text: Append newline & "Canvas recording is not supported - try Firefox"
2	System: On start of layout	Record: Set Enabled
3	System: userPoints_variable = 20	UserMedia: Start recording canvas (WebM VP8 @ 2500 kbps, 0 FPS)
4	System: userPoints_variable = 30	Browser: Invoke download of UserMedia.CanvasRecordingURL with filename "video.avi"

Figura 3.7. Exemplo de programação de captura de tela condicionada a ações do usuários

Na linha 3 da Figura 3.7, está a configuração para que o sistema apenas inicie a gravação quando o valor da variável "*userPoints_variable*" for igual a 20 e a linha 4 indica que a finalização da gravação ocorra quando a mesma variável atingir o valor 30. Programações como esta permitem maior flexibilidade para que estudos de caso possam ser realizados de modo mais preciso e alinhados aos dados que se espera obter.

3.4.1.3. Coleta de fotos durante a interação do usuário

Durante a interação de um usuário com o jogo, a ferramenta *Construct 2* oferece também a possibilidade de captura de fotos antes, durante e depois da coleta de dados. A gravação de vídeo do usuário, na versão disponível atualmente (r244), ainda não oferece tal suporte. A Figura 3.8 mostra um exemplo da programação necessária para o acesso da *webcam*.

Para a captura de fotos de um usuário, as seguintes linhas definidas na ferramenta *Construct 2*, apresentadas na Figura 3.8, mostram as principais definições a serem realizadas:

- (i) Linha 1: verifica se existe uma *webcam* conectada e devidamente instalada quando o jogo é carregado.
- (ii) Linhas 2 e 3: permitem a listagem e seleção das *webcams* detectadas.
- (iii) Linha 4: permite, por meio de um botão, que a *webcam* selecionada anteriormente seja iniciada; também exibe a resolução detectada da *webcam*. Com esta programação já é possível visualizar no jogo a filmagem da *webcam* em tempo real.

1	System	On start of layout	UserMedia	Get media sources
			Add action	
2	UserMedia	On retrieved media sources	CameraSourceList	Set Visible
			CameraSourceList	Clear all items
			Add action	
3	System	Repeat UserMedia.CameraSourceCount times	CameraSourceList	Add item "Camera " & loopindex & " (" & UserMedia.CameraSourceLabelAt(loopindex) & " " & UserMedia.CameraSourceFacingAt(loopindex) & ")"
			Add action	
4	GetWebcam	On clicked	UserMedia	Request camera source CameraSourceList.SelectedIndex (prefer 0 x 0)
			Add action	
5	UserMedia	On media request approved	StatusText	Set text to "Approved!"
			SnapButton	Set Enabled
			StopWebcam	Set Enabled
			GetWebcam	Set Disabled
			Add action	
6	UserMedia	On media request declined	StatusText	Set text to "Declined!"
			Add action	
7	System	Every tick	VideoSizeText	Set text to UserMedia.VideoWidth & "x" & UserMedia.VideoHeight
			Add action	
8	StopWebcam	On clicked	UserMedia	Stop
			SnapButton	Set Disabled
			StopWebcam	Set Disabled
			GetWebcam	Set Enabled
			Add action	
9	UserMedia	Supports user media	StatusText	Set text to "Not supported! Try using Chrome 21+"
			Add action	
10	SnapButton	On clicked	UserMedia	Take snapshot (PNG, quality 700)
			SnapshotSprite	Load image from UserMedia.SnapshotURL (Keep current size, cross-origin anonymous)
			DownloadButton	Set Enabled
			Add action	
11	DownloadButton	On clicked	Browser	Invoke download of UserMedia.SnapshotURL with filename "mypicture.png"
			Add action	

Figura 3.8. Programação utilizada na captura de fotos do usuário

- (iv) Linha 5: caso a *webcam* seja reconhecida, ativa-se os botões que possibilitam tirar uma foto, e interromper o uso da *webcam*.
- (v) Linha 6: exibe o *status* caso a *webcam* selecionada não esteja funcionando.
- (vi) Linha 7: pede ao sistema que exiba o conteúdo da captura a todo momento.
- (vii) Linha 8: prevê uma ação, no caso em que o botão que interrompe a gravação for acionado.
- (viii) Linha 9: prevê uma ação, no caso em que o navegador não tenha suporte multimídia.
- (ix) Linha 10: permite a captura da foto do usuário, por meio de um botão; também permite configurar a qualidade da imagem e armazena em memória.
- (x) Linha 11: permite fazer o *download* da foto capturada anteriormente.

É possível também condicionar a captura da imagem do usuário, associada a uma ação. Essa possibilidade propicia a captura de um momento ou a sequência de momentos em que o usuário realiza determinada ação ou que alguma condição seja alcançada. Na

Figura 3.9, tem-se um exemplo de que uma foto do usuário é tirada, quando a variável "userPoints_variable" é igual ao valor 10.



Figura 3.9. Exemplo de programação de captura de foto condicionada a ações do usuário.

3.4.2. Coletando dados de variáveis quantitativas e qualitativas

Segundo [MacKenzie 2012], uma atividade muito importante no planejamento de um experimento com usuários consiste na decisão sobre quais variáveis experimentais a pesquisa deverá considerar. Segundo o autor, pensar em quais variáveis experimentais podem ser obtidas e no *design* do experimento auxilia na transição de questões genéricas e muito amplas (e.g., *Será que o usuário vai gostar do meu jogo?*) para questões de pesquisa focadas e possíveis de serem testadas (e.g., *Uma tarefa pode ser realizada com mais rapidez ao comparar duas interfaces?*)

Nesta seção é apresentada a abordagem (seguindo o esquema apresentado na Figura 3.3) de desenvolvimento de jogos para a coleta de dados de experimento com usuário, por meio da ferramenta *Construct 2*, utilizando variáveis quantitativas e qualitativas. A seguir, tem-se a Seção 3.4.2.1 que apresenta um estudo de caso no qual são utilizados jogos para coleta de dados do usuário. Por meio deste estudo, é possível identificar exemplos de variáveis comumente utilizadas em pesquisas acadêmicas. A Seção 3.4.2.2 discute modos de utilizar a ferramenta *Construct 2* para replicar algumas etapas do estudo de caso apresentado, fornecendo ao desenvolvedor exemplos para realizar seus próprios experimentos e coletas de dados.

3.4.2.1. Planejando coleta de dados: Exemplo do jogo *Flappy Bird*

Utilizaremos como exemplo para o planejamento da coleta de dados, o jogo *Flappy Bird*. Lançado no ano de 2013, este jogo foi desenvolvido para dispositivos móveis, tornou-se muito popular principalmente pela característica de ser muito difícil, na opinião dos jogadores, que se sentiam desafiados a conquistar maiores pontuações.

Devido a sua popularidade, pesquisas utilizaram-no como base para compreender melhor a experiência do usuário. Uma dessas pesquisas, de [Isaksen et al. 2015], foi conduzida com os seguintes objetivos:

- Auxiliar *designers* a explorar os elementos presentes no espaço do jogo, como (*layouts*, cenários e *sprites*)
- Compreender melhor a relação entre *design* e experiência do usuário

A partir do foco definido nos objetivos elencados, o próximo passo da pesquisa consistiu em delimitar a análise estabelecendo uma questão de pesquisa, no contexto dos objetivos, que deve ser discutida e respondida.

Questão de pesquisa:

– Como os parâmetros, sem alteração das regras, podem afetar a dificuldade que um jogo possui?

Uma das formas encontradas para responder esta questão de pesquisa foi realizar um estudo de caso contendo as duas seguintes fases:

Fase 1 - Aplicação de Questionário - foco na coleta de variáveis independentes: sexo, idade, experiência como jogador e exposição ao jogo *Flappy Bird* e suas versões genéricas.

Fase 2 - Teste de Habilidade - foco na coleta de variáveis dependentes: precisão, tempo de resposta e ações por segundo.

3.4.2.2. Realizando coleta de dados: Exemplo do jogo *Flappy Bird*

Uma vez estabelecidas a questão de pesquisa e quais variáveis experimentais serão utilizadas no processo de coleta de dados, o próximo passo consiste no desenvolvimento do jogo. Utilizando o estudo de caso apresentado anteriormente, esta seção apresenta a abordagem para replicar a coleta de dados, utilizando a ferramenta *Construct 2*. Para replicar a coleta de dados, será utilizado um modelo genérico do jogo *Flappy Bird* que acompanha a versão gratuita da ferramenta, como mostrado na Figura 3.10.

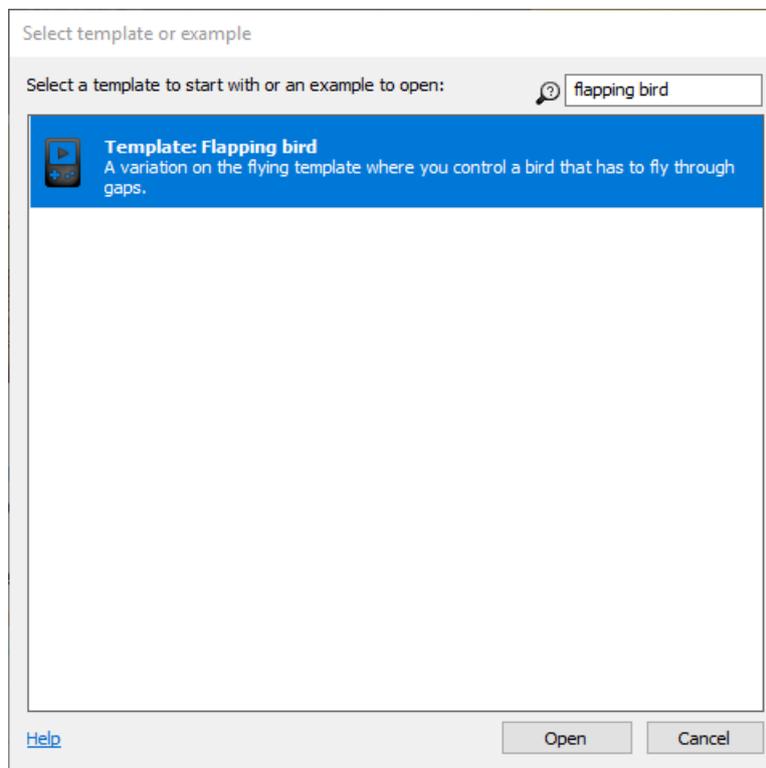


Figura 3.10. Seleção de *template* na ferramenta *Construct 2*

A partir de selecionado o *template* em *Construct 2*, a tela de desenvolvimento do referido jogo é mostrada (Figura 3.11), contendo os elementos necessários para o desenvolvimento do modo de coleta de dados nas variáveis.

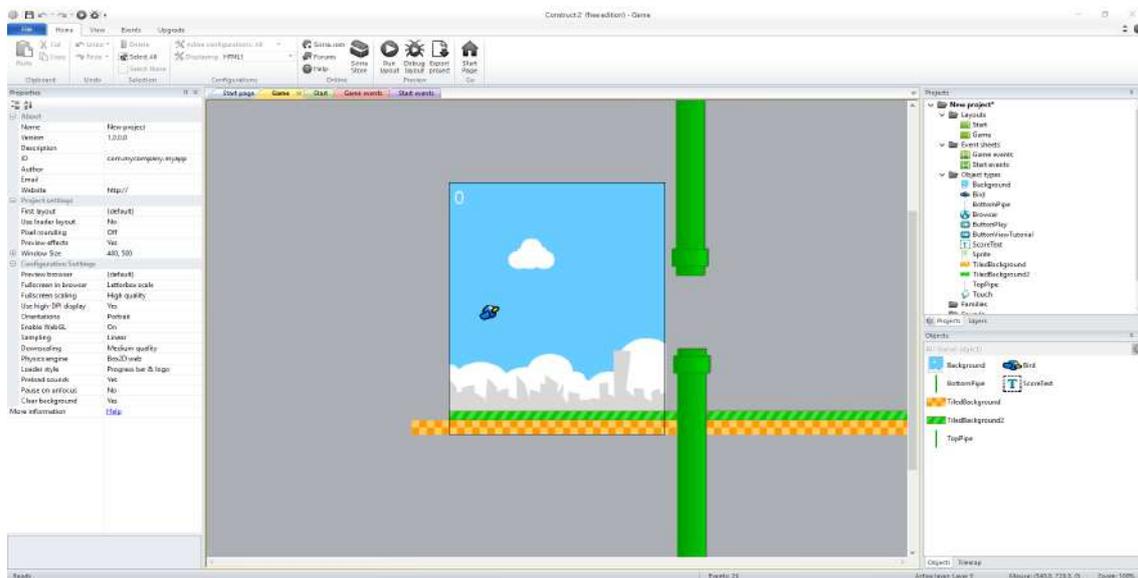


Figura 3.11. *Template Flapping bird* aberto no *Construct 2*

Fase 1 - Questionário

A primeira fase de coleta de dados, do exemplo citado anteriormente, utilizou as seguintes variáveis independentes: **sexo**, **idade**, **experiência como jogador** e **exposição ao jogo**. É muito comum que em estudos de caso, um questionário contendo perguntas sobre o perfil e opiniões dos usuários seja aplicado em algum momento.

Para coletar as mesmas variáveis na ferramenta *Construct 2*, deverá ser adicionado um novo *layout* ao *template*, no qual serão inseridos os campos em que o jogador deverá responder o questionário. Os campos adicionados podem ser observados na Figura 3.12.

Na três telas de interação pelo usuário (Figura 3.12), tem-se: item (a) apresenta ao usuário a tela inicial do jogo, item (b) são apresentadas as opções para coletar as informações relativas à fase de questionário. Nota-se que é possível inserir tanto alternativas preestabelecidas como na opção "Sexo"(masculino e feminino) e campos de texto como na opção "Idade"(anos). Para exemplificar as possibilidades de utilização nas opções "Experiência como jogador"e "Exposição ao *game*", foram oferecidas ao usuário opções listadas, utilizando a escala *Likert*, e também um campo de texto. Independente de qual estratégia o pesquisador irá adotar, para obter dados sobre o perfil do usuário, a ferramenta permite inserção de uma ou mais opções de coleta de dados.

No item (c) é apresentada a tela na qual o jogo será executado. Estratégias podem ser adotadas tanto na ordem sequencial como na repetição de telas de interação com o usuário. A ferramenta *Construct 2* permite criar condições que, ao serem satisfeitas, diferentes formulários ou telas podem ser apresentados ao usuário. Um exemplo dessa possibilidade é o fator tempo: após determinado tempo de jogo, automaticamente outro

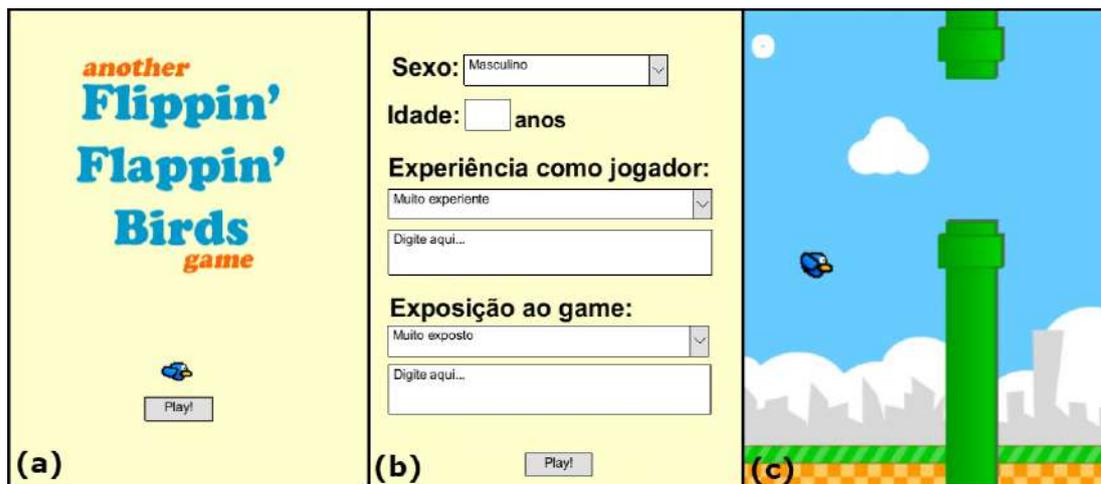


Figura 3.12. Sequências de interfaces do jogo: (a) tela inicial, (b) tela onde será coletado as informações do questionário e (c) tela do jogo

formulário de coleta de dados pode ser apresentado ao usuário. Outro exemplo está em cumprir certos objetivos ou tarefas durante a execução do jogo, dependendo dos resultados dos usuários, como coletar certa quantidade de moedas ou alcançar certa quantidade de pontos, formulários diferentes podem surgir aumentando assim a precisão na coleta de dados sobre a experiência do usuário.

Outra vantagem ao utilizar formulários para coleta de dados, durante a execução de jogos, se refere à possibilidade de utilizar as próprias informações inseridas pelo usuário para alterar a experiência de interação. Se um usuário se diz experiente em certo assunto, uma possibilidade (dependendo do que se deseja coletar) se encontra no nivelamento de dificuldade que pode ser ajustado para cada usuário e, assim, ao final dos experimentos, os dados coletados podem refletir, de forma mais fiel, a real experiência do usuário.

Fase 2 - Habilidade

A segunda fase de coleta de dados, do exemplo citado anteriormente, utilizou as variáveis dependentes: **precisão**, **tempo de resposta** e **ações por segundo**. Para essas variáveis, dados quantitativos devem ser devidamente coletados e apresentados de forma clara ao pesquisador para a devida análise posterior.

Coleta da variável "Precisão": para coletar os dados sobre precisão, aos usuários foi instruído que apenas clicassem em um botão sempre que uma linha, que se movia constantemente, estivesse horizontalmente alinhada a um alvo na tela. O teste foi repetido 20 vezes com três velocidades diferentes. Para verificar a possibilidade de a ferramenta replicar a mesma coleta de dados, o *template* foi modificado (Figura 3.13).

Na Figura 3.13, o item (a) mostra a posição do alvo vermelho adicionado juntamente a uma linha preta que realiza movimentos horizontais, no item (b) é apresentada a configuração utilizada para atribuir o comportamento (*Behavior*) de movimento da linha. Esse item é importante pois permite a fácil mudança da velocidade da linha, apenas indicando um valor no item *Period*. No item (c) é mostrada a posição do botão que deverá ser pressionado (no trabalho de [Isaksen et al. 2015] não é especificado qual o nome atribuído

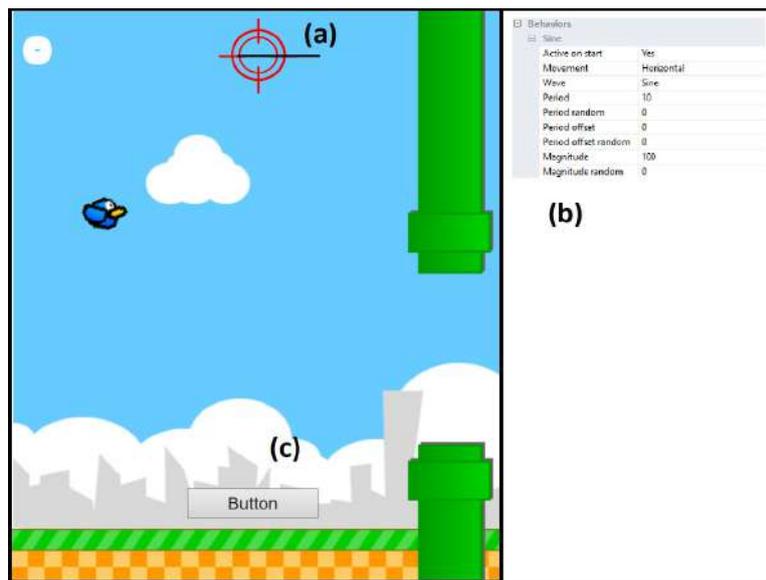


Figura 3.13. Elementos gráficos necessários para replicar coleta da variável precisão

ao botão). Esses elementos gráficos foram adicionados na interface do jogo, utilizando a ferramenta *Construct 2*, e podem ser reposicionados e arrastados facilmente com o uso do *mouse*, permitindo foco na coleta de dados e não na programação desse elementos por meio de alguma linguagem ou método específico. Para verificar se a regra de precisão foi respeitada, um programa foi adicionado ao exemplo (Figura 3.14).

21	Ability (data collection)		
	OK buttonTarget	On clicked	System Add 1 to variable_targetAligned
22	line	$X \geq 190$	Add action
	line	$X \leq 210$	

Figura 3.14. Programa “desenvolvido” para coleta da variável precisão

Uma condição foi definida na linha 22 da Figura 3.15, de forma que uma variável chamada "*variable_targetAligned*" apenas receba a adição de um valor, quando o botão "*buttonTarget*" for clicado e a linha "*line*" estiver entre os valores $X \geq 190$ e $X \leq 210$, valores estes que representam o alinhamento do alvo e da linha na interface do jogo. Esse intervalo de valores no eixo X pode ser configurado para maior flexibilidade do alinhamento da linha com o alvo. Assim, ao final da interação, é possível saber, com precisão, quantas vezes o usuário conseguiu realizar a condição de pressionar o botão no alinhamento da linha com o alvo.

Essa condição também pode ser reajustada durante a interação do usuário. Por exemplo, na linha 23, é possível aumentar a velocidade da linha "*line*" criando a seguinte condição "quando a variável '*variable_targetAligned*' for maior que 5, o período de movimento será 8", como mostra a Figura 3.15.

22	OK buttonTarget	On clicked	System	Add 1 to variable_targetAligned
	line	$X \geq 190$	Add action	
	line	$X \leq 210$		
23	System	variable_targetAligned > 5	line	Set Sine period to 8

Figura 3.15. Programação realizada para coleta da variável precisão e condições de velocidade para o objeto linha

Coleta da variável "Tempo de reação": para medir o tempo de reação, os usuários foram instruídos a pressionar o botão tão rápido quanto visualizassem a linha horizontal ao lado direito da janela do jogo. A média de atraso, que permitiu calcular o tempo de reação, foi obtida utilizando o momento em que a linha se posicionava a direita da janela e o momento que o usuário pressionava o botão. Para coletar esta variável, uma programação foi adicionada, como mostra a Figura 3.16.

24	OK buttonTarget	On clicked	System	Add 1 to variable_Button_lineRight
	line	$X \geq 300$	System	Set variable_Button_lineRight_Time to time
Add action				
25	line	$X \geq 300$	System	Set variable_System_lineRight_Time to time

Figura 3.16. Programação realizada para verificar o "click" e a posição da linha horizontal

Na Figura 3.16, a linha 24 contém a programação que apenas adiciona um valor à variável "variable_Button_lineRight" e captura o tempo (segundos) na variável "variable_Button_lineRight_Time" quando a seguinte condição é satisfeita: "Quando o botão "buttonTarget" é clicado e a posição da linha "line" for maior ou igual a 300"(o valor do eixo X ≥ 30 indica se a linha está à direita da janela do jogo). Na programação da linha 25, o momento (tempo em segundos) em que a linha se encontra à direita da janela ($x \geq 300$), é adicionado na variável "variable_System_lineRight_Time", subtraindo as duas variáveis de tempo é possível calcular a média de atraso como objetivado no *design* da coleta do "Tempo de reação".

Coleta da variável "Ações por segundo": para coleta dos dados referentes a uma certa variável, os usuários foram instruídos a manter o botão pressionado por 10 segundos; no trabalho de [Isaksen et al. 2015] não foi especificado quando os usuários deveriam realizar esta tarefa. Para replicar esta coleta, foi adicionado ao exemplo, na linha 26, a seguinte programação: "se o botão buttonTarget for pressionado por 10 segundos, adicione um valor à variável variable_touchButton"(Figura 3.17).

A ferramenta *Construct 2* possui outras formas de ações relacionadas ao toque do tipo "touch screen", permitindo ao pesquisador, que não possui familiarização com desenvolvimento de jogo, diferentes alternativas para coletar dados por meio da realização de tarefas.

Por meio da análise do trabalho de [Isaksen et al. 2015] e o uso da ferramenta



Figura 3.17. Programação realizada para verificar o tempo de toque no botão

de desenvolvimento *Construct 2*, foi possível replicar algumas etapas do estudo de caso, exemplificando uma abordagem para coleta de dados no contexto de jogos. Desta forma, pesquisadores que não possuem conhecimentos prévios, sobre desenvolvimento de jogos, dispõem de uma abordagem para replicar ou desenvolver experimentos para coleta de dados da experiência do usuário.

3.4.3. Armazenamento de dados coletados

Uma das características mais importantes na coleta de dados relativos à experiência do usuário está em desenvolver estratégias de armazenamento de dados gerados durante a interação, para análises posteriores. Os dados devem, portanto, ser armazenados de forma a permitir fácil manipulação e consulta posterior.

Nesta seção é descrito um modo de armazenamento dos dados coletados em uma base de dados *online*, durante a execução de um jogo desenvolvido por meio da ferramenta *Construct 2*. A possibilidade de armazenar dados em uma base de dados *online* visa assegurar que os dados estarão seguros e poderão ser acessados e analisados de qualquer lugar.

3.4.3.1. Exemplo de jogo com habilidade motora

Utilizando como exemplo a habilidade "Motora", na Seção 3.2.3 no [CDAJ.2] foi apresentada a seguinte diretriz, de categoria intermediária, que recomenda: “faça elementos interativos que exigem precisão (ex. cursor/toque controlados pelas opções do menu) estacionária”. Utilizando essa diretriz como base, a Figura 3.18 mostra uma interface de jogo, na qual os usuários deverão arrastar, com auxílio do *mouse*, as maçãs até a cesta, permitindo coletar duas variáveis: a quantidade de maçãs e o tempo exato em que a maçã foi colocada na cesta.

A interface da Figura 3.18 foi desenvolvida utilizando a ferramenta *Construct 2*, para Web, e poderá ser acessada localmente ou hospedada em um servidor. Para armazenar as variáveis “quantidade de maçãs na cesta” e “tempo em que a maçã e colocada” foi necessário um servidor *online*, seguindo o esquema LAMP⁷, um código na linguagem PHP para realizar o acesso e inserção no banco de dados.

Na Figura 3.19 é apresentado o código-fonte para criar a tabela chamada “*apples*” que irá receber os dados do jogo. Na linha 3, o *score* irá receber a quantidade de maçãs colocada na cesta e na linha 4, o *time* irá receber o tempo exato em que cada maçã foi colocada na cesta.

⁷Linux, Apache, MySQL e PHP



Figura 3.18. Exemplo de jogo para coleta de dados relacionados a habilidade Motora

```

1 CREATE TABLE 'apples' (
2   'id' int(11) NOT NULL,
3   'score' int(11) NOT NULL,
4   'time' time NOT NULL
5 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

Figura 3.19. Código SQL utilizado para o desenvolvimento da tabela que irá receber os dados do jogo

Para que as variáveis fossem armazenadas no banco de dados, um código na linguagem PHP foi hospedado no servidor *online*. A programação do jogo faz uma requisição utilizando o método *POST* ao código-fonte PHP (Figura 3.20). Na linha 2 desse código, foi adicionado um *header* para permitir que o recurso *online* fosse acessado de forma cruzada, isso é necessário devido a forma em que o *Construct 2* permite configurar o acesso a recursos externos; entre as linhas 4 e 7 são criadas variáveis de conexão com o banco de dados, e na linha 7, a variável *\$dblink* realiza a conexão com o banco utilizando as variáveis anteriores.

Por fim, as linhas 11 e 12 contêm as variáveis *score* e *time* que irão receber os valores enviados pelo jogo por meio do método *POST*. A linha 13 executa a *query* (consulta) para inserção de dados no banco e, a linha 15 encerra a conexão. O código-fonte exemplifica apenas uma forma simples de inserção de dados no banco, outros recursos em SQL ou a utilização de estruturas existentes podem ser utilizadas para obtenção de melhores resultados no armazenamento de dados.

A Figura 3.21 mostra uma programação para envio de dados para um banco de dados externo. Para realizar esse envio, a ferramenta *Construct 2* utiliza o AJAX⁸ e,

⁸Asynchronous Javascript and XML

```

1 <?php
2 header('Access-Control-Allow-Origin: *');
3
4 $db = "database";//Your database name
5 $dbu = "username";//Your database username
6 $dbp = "password";//Your database users' password
7 $host = "localhost";//Your host
8
9 $dblink = mysqli_connect($host,$dbu,$dbp,$db);
10
11 $score = $_GET['score'];
12 $time = $_GET['time'];
13 $sql = mysqli_query($dblink, "INSERT INTO '$db'.'apples' ('id','score','time') VALUES ('','$score','$time');");
14
15 mysqli_close($dblink);

```

Figura 3.20. Código PHP utilizado para intermediação do jogo com o banco de dados

portanto, ele deve ser adicionado durante o desenvolvimento do jogo na ferramenta.

The screenshot shows a programming interface with four lines of logic:

- Line 1:** Triggered by a collision between a 'basket' sprite and an 'apple' sprite. Actions include: 'Destroy' the 'apple' sprite; 'System' action to 'Add 1 to variable_Basket'; and an 'AJAX' action to 'Send "" to URL variable_Url & "savescores.php?score=" & variable_Basket & "&time=" & variable_Time (method 'POST', tag 'PostScore')'.
- Line 2:** Triggered by the 'AJAX' action from line 1. Actions include: 'System' action to 'Add 1 to variable_Save'; and a 'save' action to 'Set text to "Save: " & variable_Save'.
- Line 3:** Triggered by a 'System' 'Every tick' event. Action: 'Score' action to 'Set text to variable_Basket'.
- Line 4:** Triggered by a 'System' 'Every 1.0 seconds' event. Actions include: 'Clock' action to 'Set text to "Time: " & zeropad(int(time/60), 1) & ":" & zeropad(int(time%60), 2)'; and a 'System' action to 'Set variable_Time to time'.

Figura 3.21. Programação utilizada para armazenamentos de dados em um banco de dados externo

Na linha 1 da Figura 3.21, uma condição foi criada para que, quando ocorre uma colisão entre as *sprites* "basket" e "apple", a *sprite* "apple", desapareça (*apple Destroy*) da interface do jogo para que o usuário entenda que a maçã foi colocada na cesta; também deverá ser adicionado o valor 1 à variável *variable_Basket*. A parte mais importante da programação ocorre ainda na linha 1, com uma chamada em AJAX que é feita da seguinte forma: "*variable_Url* & "*savescores.php?score="* & *variable_Basket* & "&*time="* & *variable_Time*, na qual:

- *variable_Url*: uma variável global, que contém o endereço do servidor *online*, o qual inclui o código PHP e o banco de dados. Um detalhe importante é que, ao final do endereço atribuído a esta variável, deverá ser incluído "/" para que a concatenação utilizando o símbolo "&" possa ser realizada.

- *"savescores.php?score="*: esse trecho é a continuação do endereço que será concatenado com a variável do endereço do servidor. Nele deverá ser colocado o nome do arquivo que contém o código PHP hospedado no servidor; no exemplo, o arquivo possui o nome "savescores.php", como a requisição utilizará o método *POST* a continuação da requisição passa o valor que a variável *score* irá receber no banco de dados.
- *variable_Basket*: esta variável possui o valor de qual maçã foi colocada na cesta; como é desejado saber também o tempo em que a maçã é colocada na cesta; toda vez que uma maçã é colocada, uma nova entrada no banco é realizada. Dessa forma, será possível saber quanto tempo se passou entre o usuário colocar a maçã 1 na cesta e a maçã 2, por exemplo.
- *"&time=" & variable_Time*: o segundo valor que irá ser enviado ao banco é o da variável *time* que, neste exemplo, está concatenado com a variável *variable_Time*. Esta variável recebe o tempo exato que uma maçã é colocada na cesta.
- *(method "POST", tag "PostScore")*: configura qual é o método de requisição que a chamada AJAX deverá efetuar e atribui uma *"tag"* para esta chamada; esse recurso permite verificar se a chamada foi realizada com sucesso.

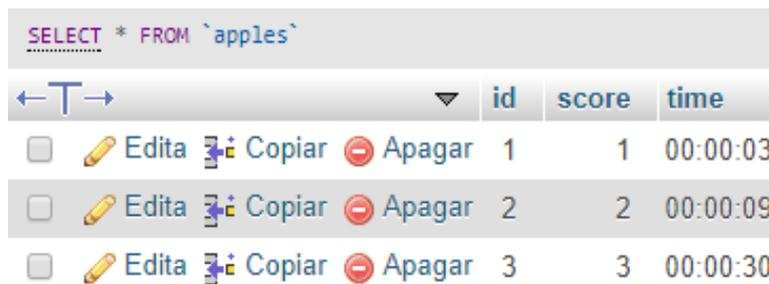
A linha 2 da Figura 3.21 diz ao sistema do jogo que a todo momento (*"Every Tick"*) o valor da variável *variable_Basket* deverá ser atualizado na interface. Uma caixa de texto com o nome *Score* foi criada e fica sobre a cesta na interface do jogo.

A linha 3 informa ao sistema que a cada 1.0 segundo, a caixa de texto chamada *Clock*, deverá receber o tempo e ser exibido na interface. A ferramenta trabalha, por padrão, com milésimos de segundos sendo necessário realizar a conversão para minutos e segundos, para isso foi utilizada a expressão: *zeropad(int(time/60% 60), 1) & ":" & zeropad(int(time%60), 2)*. Ainda nesta linha, o valor de tempo é adicionado à variável *variable_Time* a cada 1,0 segundo para que o valor exibido na interface e o valor que irá ser adicionado ao banco sempre sejam os mesmos.

Na linha 4 da Figura 3.21, é verificado se a chamada AJAX nomeada de *"PostScore"* foi executada com sucesso. Foi possível atribuir uma ação (também é possível atribuir ações caso a chamada não seja executada com sucesso). A ação atribuída no exemplo adiciona o valor 1 à variável *variable_Save* e exibe este valor na interface por meio de uma caixa de texto chamada *save*, essa variável foi criada para informar que os dados estão sendo salvos com sucesso no banco de dados.

Na Figura 3.22, é apresentado o resultado da consulta dos dados armazenados. Assim, tem-se que a primeira maçã foi colocada na cesta aos 3 segundos e a terceira maçã foi colocada aos 30 segundos. Dessa forma, é possível verificar a velocidade que cada usuário, utilizando suas habilidades motoras, está interagindo com o jogo. Com outras possibilidades de experimentos seria viável controlar também a velocidade em que novas maçãs aparecem na tela, verificar a ordem em que os usuários colocam maçãs na cesta, mudar a posição da cesta e etc.

Por meio do exemplo de jogo aqui descrito, utilizando conceitos sobre habilidades motoras e o uso da ferramenta de desenvolvimento *Construct 2*, foi possível observar um



	id	score	time
<input type="checkbox"/>  Edita  Copiar  Apagar	1	1	00:00:03
<input type="checkbox"/>  Edita  Copiar  Apagar	2	2	00:00:09
<input type="checkbox"/>  Edita  Copiar  Apagar	3	3	00:00:30

Figura 3.22. Consulta na tabela *apples* retornando os valores armazenados

modo de armazenar os dados gerados durante a interação do usuário em uma base de dados online. Existem outros modos para armazenar dados utilizando a ferramenta, porém, dentre os existentes atualmente, este modo se mostrou eficiente para análises posteriores.

3.5. Conclusões

Este capítulo abordou o tema de desenvolvimento de jogos para pesquisadores de áreas relacionadas com experimentos que consideram experiência do usuário. Em especial, é discutida a adoção de jogos nas pesquisas científicas, para auxiliar coleta de dados e estudos de caso. Este capítulo atende a um amplo conjunto de pesquisadores, mesmo os que encontram dificuldades ou não possuam familiaridade com o desenvolvimento de elementos de jogos. O uso de jogos em pesquisas se apresenta como um tema de grande relevância para profissionais e pesquisadores envolvidos com experiência do usuário.

Já a experiência dos autores mostrou que o desenvolvimento de jogos pode ser tanto um desafio quanto uma inovação para pesquisas nos diversos campos da Ciência. Pesquisadores dispõem do conhecimento para desenvolver experimentos contendo coleta de dados provenientes da interação do usuário. No entanto, muitas vezes, a falta de familiaridade com ferramentas, técnicas e linguagens utilizadas no desenvolvimento de jogos podem desencorajá-los a adotar estratégias inovadoras envolvendo elementos de *gamificação*.

Finalmente, neste capítulo, é apresentada uma abordagem de desenvolvimento de jogos por meio da ferramenta *Construct 2* que permite diversas possibilidades de apoio ao desenvolvimento de jogos e, principalmente, possibilita o *design* de estudos de casos, bem como a coleta de dados e o armazenamento de variáveis, para futuras análises. Para melhor aproveitamento da abordagem, foi apresentada uma visão geral sobre a área de jogos visando coleta de dados, contendo referências sobre a experiência do usuário, considerando-se os tópicos de usabilidade e acessibilidade, bem como exemplos de estudos relacionados.

3.6. Agradecimentos

Agradecemos ao suporte dado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processo n. 2016/01009-0.

Referências

- Barendregt, W., Torgersson, O., Eriksson, E., and Börjesson, P. (2017). Intermediate-level knowledge in child-computer interaction: A call for action. In *Proceedings of the 2017 Conference on Interaction Design and Children, IDC '17*, pages 7–16, New York, NY, USA. ACM.
- Chen, S.-T., Huang, Y.-G. L., and Chiang, I.-T. (2012). Using somatosensory video games to promote quality of life for the elderly with disabilities. In *Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2012 IEEE Fourth International Conference on*, pages 258–262. IEEE.
- Coovert, M. D., Winner, J., Bennett Jr, W., and Howard, D. J. (2017). Serious games are a serious tool for team research. *INTERNATIONAL JOURNAL OF SERIOUS GAMES*, 4(1):41–55.
- da Costa, M. A. F. and da Costa, M. d. F. B. (2009). *Metodologia da pesquisa: conceitos e técnicas*. Interciência, Rio de Janeiro, 2 edition.
- de Lima Salgado, A., do Amaral, L. A., de Mattos Fortes, R. P., Chagas, M. H. N., and Joyce, G. (2017). Addressing mobile usability and elderly users: Validating contextualized heuristics. In *International Conference of Design, User Experience, and Usability*, pages 379–394. Springer.
- Ellis, B., Ford-Williams, G., Graham, L., Grammenos, D., and Hamilton, I. (2012). *Game Accessibility Guidelines* - <http://www.gameaccessibilityguidelines.com>.
- Fortes, R. P. M., de Lima Salgado, A., de Souza Santos, F., do Amaral, L. A., and da Silva, E. A. N. (2017). Game accessibility evaluation methods: A literature survey. In *International Conference on Universal Access in Human-Computer Interaction*, pages 182–192. Springer.
- Frederik, D. G., Peter, M., and Jan, V. L. (2010). Uncharted waters?: exploring experts' opinions on the opportunities and limitations of serious games for foreign language learning. In *Proceedings of the 3rd International Conference on Fun and Games*, pages 107–115. ACM.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in Entertainment (CIE)*, 1(1):20–20.
- Gonçalves, C. (2012). Programa de estimulação cognitiva em idosos institucionalizados. *O Portal dos Psicólogos*, 18:1–18.
- Goodman, E., Kuniavsky, M., and Moed, A. (2013). Observing the user experience: A practitioner's guide to user research. *IEEE Transactions on Professional Communication*, 56(3):260–261.
- Greitzer, F. L., Kuchar, O. A., and Huston, K. (2007). Cognitive science implications for enhancing training effectiveness in a serious gaming context. *Journal on Educational Resources in Computing (JERIC)*, 7(3):2.

- Hassenzahl, M. and Tractinsky, N. (2006). User experience - a research agenda. *Behaviour & Information Technology*, 25(2):91–97.
- Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433. ACM.
- Isaksen, A., Gopstein, D., and Nealen, A. (2015). Exploring game space using survival analysis. In *FDG*.
- Jung, Y., Li, K. J., Janissa, N. S., Gladys, W. L. C., and Lee, K. M. (2009). Games for a better life: effects of playing wii games on the well-being of seniors in a long-term care facility. In *Proceedings of the Sixth Australasian Conference on Interactive Entertainment*, page 5. ACM.
- Kapp, K. M. (2012). *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons.
- MacKenzie, I. (2012). *Human-Computer Interaction: An Empirical Research Perspective*. Elsevier Science.
- Marconi, M. A. and Lakatos, E. M. (2001). *Metodologia do Trabalho Científico*. Atlas, 6 edition.
- Marconi, M. A. and Lakatos, E. M. (2002). *Técnicas de Pesquisa*. Atlas, 5 edition.
- Michael, D. R. and Chen, S. L. (2005). *Serious games: Games that educate, train, and inform*. Muska & Lipman/Premier-Trade.
- Pedro, L. Z. (2016). *Uso de gamificação em ambientes virtuais de aprendizagem para reduzir o problema da externalização de comportamentos indesejáveis*. PhD thesis, Universidade de São Paulo.
- Pádua, E. M. M. (2000). *Metodologia da Pesquisa: Abordagem Teórico-Prática*. Papi-rus, 6 edition.
- Salen, K. and Zimmerman, E. (2004). *Rules of play: Game design fundamentals*. MIT press.
- Samodelkin, A., Alavesas, P., and Voroshilov, A. (2016). A platform for pervasive games for research. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia, MUM '16*, pages 335–337, New York, NY, USA. ACM.
- Souza, G. R. and Trevisan, D. G. (2014). Estudo investigativo sobre idosos, jogos e sua motivações. *Cadernos de Informática*, 8(3):35–40.
- Sung, I. and Berland, M. (2017). Forest friends demo: A game-exhibit to promote computer science concepts in informal spaces. In *Proceedings of the 2017 Conference on Interaction Design and Children, IDC '17*, pages 701–704, New York, NY, USA. ACM.

Tong, T., Chignell, M., Tierney, M. C., and Lee, J. (2016). A serious game for clinical assessment of cognitive status: validation study. *JMIR serious games*, 4(1).

Torres, A. C. S. (2011). Cognitive effects of video games on old people. *International Journal on Disability and Human Development*, 10(1):55–58.

Valladares-Rodríguez, S., Pérez-Rodríguez, R., Anido-Rifón, L., and Fernández-Iglesias, M. (2016). Trends on the application of serious games to neuropsychological evaluation: A scoping review. *Journal of biomedical informatics*, 64:296–319.

Van Eck, R. (2006). Digital game-based learning: It's not just the digital natives who are restless. *EDUCAUSE review*, 41(2):16.

Wiemeyer, J. and Kliem, A. (2011). Serious games in prevention and rehabilitation—a new panacea for elderly people? *European Review of Aging and Physical Activity*, 9(1):41.

Wood, M., Wood, G., and Balaam, M. (2017). Sex talk: Designing for sexual health with adolescents. In *Proceedings of the 2017 Conference on Interaction Design and Children*, IDC '17, pages 137–147, New York, NY, USA. ACM.

Biografia resumida dos autores



Marcio Maestrello Funes: é mestrando no programa de Pós-Graduação em Ciências da Computação e Matemática Computacional no Instituto de Ciências Matemáticas e de Computação (ICMC), na Universidade de São Paulo (USP). Tem experiência na área de jogos tendo atuado como docente e desenvolvedor. Atualmente realiza pesquisas em avaliação de experiência do usuário com interfaces baseadas em gestos, acessibilidade e Web. Atua principalmente nos seguintes temas: Interface Humano-Computador, Computação Ubíqua, Interfaces Naturais, realidade aumentada e virtual, design e acessibilidade.



Leandro Agostini do Amaral: é doutorando no programa de Pós-Graduação em Ciências da Computação e Matemática Computacional no Instituto de Ciências Matemáticas e de Computação (ICMC), na Universidade de São Paulo (USP) e Pesquisador Responsável em um projeto PIPE/FAPESP que utiliza jogos para treinamento cognitivo voltado ao público acima de 50 anos. Desde 2010 pesquisa assuntos relacionados à Interação Humano-Computador, priorizando questões de acessibilidade.



Rudinei Goularte: é professor do Departamento de Ciências de Computação na Universidade de São Paulo, campus de São Carlos. Possui graduação em Ciências da Computação pela Universidade Federal de Mato Grosso do Sul (1995). Possui mestrado (1998), doutorado (2003) e livre-docência (2011) pela Universidade de São Paulo [São Carlos], todos em Ciência da Computação. Atualmente é professor associado do ICMC/USP em regime de dedicação integral à docência e à pesquisa e orientador pleno de mestrado e doutorado. Atua como consultor ad hoc da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Desenvolve pesquisa em Multimídia nas linhas: codificação de vídeo digital, vídeo 3D, recuperação baseada em conteúdo, personalização de conteúdo e tv interativa.



Renata Pontin M. Fortes: é professora do Departamento de Ciências de Computação na Universidade de São Paulo, campus de São Carlos. Possui graduação em Bacharelado em Ciência da Computação pela Universidade de São Paulo (1982), mestrado em Ciência da Computação e Matemática Computacional pela Universidade de São Paulo (1991) e doutorado em Física pela Universidade de São Paulo (1996). Atualmente é professora associada da Universidade de São Paulo, consultor do Ministério da Educação e consultor ad hoc da Fundação de Amparo à Pesquisa do Estado de São Paulo. Tem experiência na área de Ciência da Computação, com ênfase em Engenharia de Software, atuando principalmente nos seguintes temas: engenharia de web, projetos de software livre, acessibilidade na web e processo de software.