

UNIVERSIDADE FEDERAL DO PAMPA

BRUNO OLIVEIRA DE ARAÚJO

**IMPACTOS DE REQUISITOS DE ACESSIBILIDADE NA EVOLUÇÃO DO JOGO
DIGITAL PROGRAMMER**

**Alegrete
2021**

BRUNO OLIVEIRA DE ARAÚJO

**IMPACTOS DE REQUISITOS DE ACESSIBILIDADE NA EVOLUÇÃO DO JOGO
DIGITAL PROGRAMMER**

Trabalho de Conclusão de Curso
apresentado ao Curso de Engenharia
de Software da Universidade Federal do
Pampa, como requisito parcial para
obtenção do Título de Bacharel em
Engenharia de Software.

Orientadora: Amanda Meincke Melo

**Alegrete
2021**

Ficha catalográfica elaborada automaticamente com os dados fornecidos pelo(a) autor(a) através do Módulo de Biblioteca do Sistema GURI (Gestão Unificada de Recursos Institucionais).

A481o Araujo, Bruno Oliveira de
Impactos de Requisitos de Acessibilidade na Evolução do Jogo Digital
Programmer/ Bruno Oliveira de Araujo.
98 p.

Orientador: Amanda Meincke Melo
Trabalho de Conclusão de Curso (Graduação) – Universidade
Federal do Pampa, ENGENHARIA DE SOFTWARE, 2021.

1. Jogos Digitais. 2. Evolução de Software. 3. Acessibilidade. I. Título.



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal do Pampa

BRUNO OLIVEIRA DE ARAÚJO

**IMPACTOS DE REQUISITOS DE ACESSIBILIDADE NA EVOLUÇÃO DO JOGO DIGITAL
PROGRAMMER**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 03 de maio de 2021.

Banca examinadora:

Profa. Dra. Amanda Meincke Melo

Orientadora

Unipampa

Prof. Dr. João Pablo Silva da Silva

Unipampa

Prof. Dra. Claudia Camerini Correa Perez

Unipampa



Assinado eletronicamente por **JOAO PABLO SILVA DA SILVA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/05/2021, às 22:13, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **AMANDA MEINCKE MELO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/05/2021, às 22:13, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **CLAUDIA CAMERINI CORREA PEREZ, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/05/2021, às 22:13, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0512674** e o código CRC **BAB3D4E7**.

Universidade Federal do Pampa, Campus Alegrete
Av. Tiarajú, 810 – Bairro: Ibirapuitã – Alegrete – RS CEP: 97.546-550

Telefone: (55) 3422-8400

RESUMO

A adoção de jogos no ensino de programação é uma estratégia bastante utilizada nos dias atuais. Considerando a proposta contemporânea de promoção de uma educação inclusiva, para todos, é de suma importância que esses jogos sejam amplamente acessíveis. Uma possível solução para realizar o desenvolvimento desses requisitos de acessibilidade é a utilização de padrões de interface de usuário. Utilizados como base para desenvolver o jogo digital Programmer, os padrões de interface de usuário podem ser utilizados para a manutenção da coerência das telas ajudando na criação da imersão do jogo. Entretanto, o desenvolvimento de jogos, de forma geral, raramente visa contemplar requisitos de acessibilidade, criando uma vasta gama de jogos que excluem pessoas com algum tipo de deficiência. Além disso, a revisão do estado da arte sugere a ausência de processos de evolução de jogos digitais que contemplem a acessibilidade entre seus requisitos. Dessa forma, este Trabalho de Conclusão de Curso tem como objetivo geral promover acessibilidade em jogos educacionais pela aplicação de padrões de projeto de interface de usuário no desenvolvimento de jogos digitais. A metodologia deste trabalho define como se dá a adaptação do processo de evolução de *software*, como são criados os documentos necessários para o trabalho e a utilização de um processo adaptado do Scrum para realizar o desenvolvimento. Como resultado foi possível compreender o estado atual da acessibilidade do jogo digital Programmer, a partir da aplicação de um *checklist* e produzir seu *Game Design Document*, obtendo uma visão geral sobre o jogo. Ademais, além de desenvolver um Catálogo de Padrões de Interface de Usuário contemplando requisitos de acessibilidade, também foram gerados protótipos de tela para adaptação do jogo a uma nova tecnologia, que contempla os requisitos de acessibilidade definidos. Como trabalhos futuros, propõe-se a implementação desses requisitos em uma plataforma diferente da adotada originalmente.

Palavras-Chave: Jogos Digitais, Evolução de Software, Acessibilidade.

ABSTRACT

Adopting games in programming teaching is a strategy widely used nowadays. Given the contemporary proposal to promote inclusive education for all, it is of paramount importance that these games can be widely accessible. One possible solution to the development of accessibility requirements is applying user interface patterns. Used as a basis for the development of the Programmer digital game, user interface patterns can be used to maintain consistency of the screens that helps create game immersion. However, the development of games, in general, rarely take into account accessibility requirements, creating a wide range of games that exclude people with some type of disability. In addition, the state of the art review suggests the absence of digital game evolution processes that include accessibility among their requirements. Thus, this study has as its general objective to promote accessibility in educational games by applying user interface design patterns in the development of digital games. The methodology of this work defines how to adapt the software evolution process, how the necessary documents for the work and the use of a process adapted from Scrum to carry out the development. As a result it was possible to understand the current state of accessibility of the digital game Programmer, from the application of a checklist and produce its Game Design Document, obtaining an overview about the game. In addition, besides developing a User Interface Standards Catalog covering accessibility requirements, screen prototypes were also generated to adapt the game to a new technology, which includes the defined accessibility requirements. As future work, it is proposed to implement these requirements on a different platform than the one originally adopted.

Keywords: Digital Games, Software Evolution, Accessibility.

LISTA DE FIGURAS

Figura 1 – Representação gráfica do processo de evolução de software	21
Figura 2 – Metodologia para conduzir a evolução do jogo digital Programmer	36
Figura 3 - Representação gráfica do subprocesso de analisar Programmer	37
Figura 4 – Subprocesso de evolução do jogo digital Programmer	39
Figura 5 – Representação gráfica do processo de implementação de mudança	41
Figura 6 – Protótipo de tela do menu do jogo digital Programmer	54
Figura 7 – Protótipo de tela de um desafio selecionado do jogo Programmer	55

LISTA DE QUADROS

Quadro 1 – Bases de Busca	25
Quadro 2 – Strings de Busca e Resultados da Base ACM Digital Library	25
Quadro 3 – Strings de Busca e Resultados da Base IEEE Xplore Digital Library	26
Quadro 4 – Strings de Busca e Resultados da Base Scopus	26
Quadro 5 – Contribuição dos artigos coletados para o Estado da Arte	44
Quadro 6 – Backlog do Produto	53

LISTA DE SIGLAS

GDD – *Game Design Document*

IGDA – *International Game Developers Association*

TCC – Trabalho de Conclusão de Curso

PO – *Product Owner*

W3C – *World Wide Web Consortium*

HTML5 - Hypertext Markup Language, version 5

UNIPAMPA – Universidade Federal do Pampa

SUMÁRIO

1 INTRODUÇÃO	11
2 FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA	14
2.1 Acessibilidade	14
2.2 Jogos Digitais	15
2.2.1 Jogos Digitais na Educação	17
2.2.2 Acessibilidade em Jogos Digitais	18
2.3 Padrões de Projeto de Interface de Usuário	20
2.4 Evolução de Software	21
2.5 Scrum	22
2.6 Considerações Finais do Capítulo	23
3 ESTADO DA ARTE	25
3.1 Questões de Pesquisa	25
3.2 Bases de Busca	26
3.4 Critérios de Inclusão e Exclusão	28
3.5 Análise dos Resultados	29
3.5.1 Como cada etapa do processo de desenvolvimento de jogos digitais educacionais contempla a acessibilidade?	29
3.5.2 Como a evolução de software desenvolve padrões de acessibilidade?	31
3.5.3 Que padrões de projeto de interface de usuário que contribuem à promoção da acessibilidade em jogos digitais?	32
3.5.4 Como padrões de interface de usuário para jogos digitais têm contribuído à promoção de sua acessibilidade?	34
3.6 Considerações Finais do Capítulo	35
4 METODOLOGIA	37
4.1 O jogo digital Programmer	37
4.1.1 Regras do Jogo	37
4.1.2 Personagens	38

4.1.3 Controles.....	38
4.2 Metodologia para conduzir a evolução do jogo digital Programmer	38
4.2.1 Analisar Programmer	39
4.2.2 Catalogar padrões de projeto de interface de usuário.....	41
4.2.3 Evolução de Software	41
4.2.4 Implementação de Mudança.....	43
4.3 Considerações Finais do Capítulo.....	45
5 RESULTADOS.....	46
5.1 Analisar Programmer	46
5.2 Catalogar padrões de projeto de interface de usuário.....	48
5.3 Evolução de Software	50
5.4 Implementação de Mudança.....	50
5.5 Considerações Finais do Capítulo.....	53
6. CONSIDERAÇÕES FINAIS	55
APÊNDICES	60
ANEXOS	84

1 INTRODUÇÃO

A utilização de jogos como ferramentas de ensino tem diversos fatores facilitadores para adaptar essa estratégia dentro do contexto de sala de aula: o crescimento no número de títulos; as diversas plataformas utilizadas, sejam jogos para computador ou para dispositivos móveis; os diferentes estilos de jogabilidade e de gênero; e, principalmente, a grande popularidade de tanto entre jovens como também entre adultos (BRINCHER; SILVA, 2011).

De acordo com Cheiran (2013, p. 23), jogos digitais possuem “potencial para transportar recursos que corroboram/ com o desenvolvimento de habilidades importantes”, tais como solução de problemas, melhora na comunicação e do trabalho em equipe. Há, porém, dificuldades para aplicá-los no ensino, que vão desde a estrutura de escolas e universidades, a flexibilização dos métodos de ensino dos professores, até efeitos psicológicos temporários e duradouros em jogadores.

Outro problema enfrentado pela aplicação de jogos digitais como ferramentas de ensino é a exclusão de pessoas com algum tipo de deficiência. Como citam Yuam e Folmer (2008), a popularidade de jogos é causada pelo alto grau de interação oferecida, porém é importante notar que atualmente jogos digitais dependem da habilidade dos jogadores de enxergarem telas. Mesmo recursos sonoros, que ajudam na construção da imersão, na maioria dos jogos, não conseguem garantir que uma pessoa consiga jogar apenas ouvindo o *feedback* proporcionado pelos sons.

No contexto de desenvolvimento de jogos digitais, a abordagem geralmente utilizada para trabalhar com a acessibilidade é a de utilizar recursos de Tecnologia Assistiva¹. Porém, apenas a utilização desses recursos, como, por exemplo, em jogos que não foram desenvolvidos para serem compatíveis com eles, pode afetar não só a jogabilidade do jogo como também a imersão do jogador e causar mais problemas do que solucionar dificuldades. Entretanto, é necessário pontuar também que o desenvolvimento de jogos digitais, especificamente para algum

¹ Tecnologia Assistiva é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação, de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social. (COMITÊ DE AJUDAS TÉCNICAS, 2009).

grupo de pessoas com determinada deficiência pode acabar se tornando mais custoso e ainda contribuir para uma exclusão social, determinando uma separação entre jogadores com e sem deficiência (GRAMMENOS; STEPHANIDIS; SAVIDIS, 2009).

No Brasil, o Decreto nº 6.949 de 25 de agosto de 2009 determina que serão executadas inteiramente as normas da Convenção Internacional dos Direitos das Pessoas com Deficiência, que cita “a importância da acessibilidade à educação para possibilitar às pessoas com deficiência o pleno gozo de todos os direitos humanos e liberdades fundamentais”. Esse decreto valida também o que citam Hersh e Leporini (2012) sobre a importância particular de o ensino garantir que todos os estudantes possam concentrar-se no processo de aprendizado sem precisar superar obstáculos desnecessários.

Nesse contexto, jogos digitais educacionais devem prever entre seus requisitos a acessibilidade. Entre as estratégias para atender a requisitos de acessibilidade, mantendo uma coerência entre as diversas possíveis soluções, é a adoção de recomendações internacionais de acessibilidade e de padrões de projeto de interface de usuário. No caso de jogos já desenvolvidos, consideramos a evolução de *software*, como no jogo digital educacional Programmer, que pode ser usado como uma ferramenta para o ensino de programação voltado ao ensino superior, mas que ainda não contempla requisitos de acessibilidade. Contudo, a revisão do estado da arte sugere a ausência de processos de evolução de jogos digitais que contemplem a acessibilidade entre seus requisitos.

Este Trabalho de Conclusão de Curso (TCC) tem como objetivo geral, portanto, promover acessibilidade em jogos educacionais pela aplicação de recomendações de acessibilidade e padrões de projeto de interface de usuário no desenvolvimento de jogos digitais. São propostos também como objetivos específicos a catalogação de padrões de projeto de interface de usuário que promovam a acessibilidade em jogos digitais e a evolução do jogo educativo Programmer de modo que este contemple requisitos de acessibilidade. A metodologia deste trabalho define os processos e os documentos necessários para realizar a evolução do jogo Programmer, bem como a ordem necessária para executar essas tarefas.

Este texto está organizado como segue. A fundamentação teórico-metodológica é apresentada no Capítulo 2, onde são caracterizados jogos de forma geral e jogos digitais na educação, além da acessibilidade nessa área, também são apresentados os principais elementos que compõem padrões de projeto de interface de usuário, assim como conceitos e estratégias sobre evolução de *software*. No Capítulo 3, é investigado e discutido o estado da arte. A metodologia deste trabalho é apresentada no Capítulo 4, onde é mostrado todo o processo definido para realizar a evolução do jogo Programmer. No Capítulo 5, são apresentados e discutidos os resultados deste TCC. As considerações finais são apresentadas no Capítulo 6, onde são apontadas as contribuições do trabalho e possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

Neste Capítulo, são caracterizados jogos digitais e problematizada sua utilização no ensino. Além disso, são discutidas formas de desenvolver acessibilidade em jogos digitais e os padrões de projeto de interface de usuário. Finalmente, é apresentada a evolução de *software* e suas estratégias, assim como as considerações finais sobre os assuntos abordados no Capítulo.

2.1 Acessibilidade

A acessibilidade é um conceito relativamente conhecido e segundo comunicado em fórum da Organização das Nações Unidas, Estados deveriam reconhecer sua importância no processo de diminuir as diferenças de oportunidades em todas as esferas da sociedade (NAÇÕES UNIDAS, 1993).

Segundo Iwarsson e Ståhl (2003), a abordagem mais comum à acessibilidade está relacionada aos ambientes físicos. Isso denota uma questão a ser trabalhada: a negligência à acessibilidade de serviços, porém é interessante notar também que há um interesse crescente em tratar sobre esse assunto.

Normas e recomendações existentes podem ser adotadas para solucionar determinado problema de acessibilidade. Isso ajuda a entender, segundo Iwarsson e Ståhl (2003), sobre o que a acessibilidade realmente trata: cumprir requisitos necessários para manter um ambiente acessível para todas as pessoas. A acessibilidade vista de forma individual também pode ser trabalhar, fazendo com que seja colocada em perspectiva as necessidades específicas de uma pessoa, deixando essa relação centrada ao “cliente” o máximo possível. Mesmo assim, é importante não tomar decisões baseadas em convicções pessoais e sim em dados reais e confiáveis, preferencialmente tendo passado por uma camada de testes para validar sua eficácia. Já a perspectiva de grupo deixa as decisões mais abrangentes e trabalhadas em cima da diversidade de grupos com deficiência existentes.

Segundo Sasaki (2009), pode-se classificar a acessibilidade do seguinte modo:

- Acessibilidade arquitetônica;
- Acessibilidade atitudinal;

- Acessibilidade comunicacional;
- Acessibilidade instrumental;
- Acessibilidade programática;
- Acessibilidade web.

Sendo a acessibilidade um meio importante para garantir a equidade de oportunidades, ao promovê-la, é possível melhorar a experiência até mesmo de pessoas sem algum tipo de deficiência, mas com alguma limitação específica como tecnológica, cultura ou socioeconômica.

2.2 Jogos Digitais

Jogos, de forma geral, são bastante comuns em nossa sociedade. Schell (2014) define jogos de maneira mais formal como um exercício voluntário e sistemático de ações onde ocorre uma disputa de poderes em um contexto definido por regras que visam a produzir um resultado positivo para um dos jogadores. O desenvolvimento de um jogo, ou *game design*, pode ter o mesmo processo, independentemente do tipo de jogo que se deseja criar, seja ele digital ou analógico. Dessa forma, o autor define jogos em dez aspectos principais:

1. **Jogos são jogados voluntariamente:** Envolvem um “exercício voluntário e sistemático de ações”;
2. **Jogos têm objetivos:** Uma disputa de poderes mostra que vencer essa disputa é um dos objetivos do jogo;
3. **Jogos têm conflitos:** Essa mesma disputa de poderes deixa claro um conflito existente, apesar de em alguns jogos *single-player* isso não ocorrer;
4. **Jogos têm regras:** Brinquedos não têm regras, mas jogos as têm;
5. **Jogos têm vitórias e derrotas:** Em geral, ou se vence o jogo ou se é derrotado;
6. **Jogos são interativos:** O jogador é um agente ativo e não passivo. Dessa forma, a partir das regras pré-definidas é possível ter uma interação entre jogo e jogador;
7. **Jogos têm desafios:** A quantidade de desafios dentro de um jogo pode, em muitos casos, definir sua qualidade. Jogos ruins têm poucos ou muitos

desafios. Bons jogos apresentam equilíbrio na proposta de desafios para não entediar o jogador ou fazê-lo desistir;

8. **Jogos podem criar seu próprio valor internamente:** É possível medir o quão interessante um jogo pode ser a partir dessa ideia de “valor interno” de um jogo. Esse valor é gerado dentro do jogo e só é válido dentro do jogo. E isso pode variar entre jogos de azar como Roleta, que pode ser jogado com dinheiro falso, mas não é tão interessante, até jogos onde itens que só têm valor dentro do jogo podem ser comprados com dinheiro de verdade fora dele;

9. **Jogos envolvem seus jogadores:** A imersão é, tecnicamente, um bom argumento para avaliar a qualidade de um jogo. Em muitos casos, entretanto, não é uma característica necessária;

10. **Jogos são sistemas formais e fechados:** Jogos são feitos de elementos inter-relacionados que trabalham junto. Ser formal denota que jogos são bem definidos e que possuem regras. O conceito de fechado delimita esse sistema formal.

De acordo com Schell (2014), diversos aspectos compõem o processo de *game design*. Esse processo de desenvolvimento é feito inicialmente, na maioria das vezes, por uma grande quantidade de decisões. Trata-se de decisões sobre a história do jogo, suas regras, o seu ritmo, sua aparência e tudo aquilo que o jogador possa experimentar.

Segundo Petrillo (2008), existem duas etapas no processo de desenvolvimento de jogos digitais que podem ser classificadas como pontos essenciais: a definição das regras do jogo e a criação do Documento do Jogo ou Documento do Projeto. Schell (2014) menciona que as regras são a mecânica mais fundamental de um jogo, definindo espaço, objeto, ações e consequências, ou seja, permitem a realização das mecânicas citadas até aqui, como a definição de como será realizada a imersão do jogo ou quais os principais desafios do jogador dentro do mundo criado, e adicionam partes cruciais que complementam o jogo e determinam um objetivo a ser alcançado pelo jogador. O Documento do Projeto, do inglês *Game Design Document* (GDD), em muitos casos, é a única documentação criada pelo desenvolvedor de um jogo. Essa documentação é

especificada com o objetivo de descrever e detalhar todas as mecânicas do jogo, além dos desafios, fases, ambientes e objetivos gerais de cada etapa do jogo.

Segundo Hira *et al.* (2016), o GDD é um documento que serve de referência para todas as pessoas envolvidas no desenvolvimento de um jogo. Schell (2014) cita que existem vários tipos desse documento que atendem várias necessidades. Contendo as principais características do jogo como personagens, cenários e mecânicas, esse artefato é fundamental durante o processo de desenvolvimento. Desenvolver um Documento do Projeto conciso e sólido é importante não só para documentar o jogo, mas também para a fase de testes, onde as informações descritas serão melhor utilizadas.

2.2.1 Jogos Digitais na Educação

Brincher e Silva (2011) fazem contribuições interessantes sobre a utilização de jogos digitais na educação. Eles mencionam que a utilização de jogos em um contexto educacional com o objetivo de criar novos panoramas para o ensino e a aprendizagem vem se provando como uma saída eficiente. Além disso, jogos conseguem trazer uma perspectiva diferente entre os alunos pelo seu poder de criar mais interesse e de conseguir manter interações diretas sobre diversos temas.

Os autores apontam também que o uso desses aparatos, não só para a educação infantil, mas também para o ensino superior, tem tido um crescimento nos últimos anos. Outra questão para compreender o contexto da utilização de jogos digitais na educação é que, como mencionam Brincher e Silva (2011), diferentemente dos jogos que têm como principal propósito o entretenimento, a interação em jogos educativos é construída a partir da ideia de estimular o aprendizado e a curiosidade no jogador.

Outro ponto importante no uso de jogos como ferramenta de ensino é a necessidade de assimilação por parte dos alunos de conteúdos que não se referem estritamente a conceitos de matemática, química ou biologia, mas também que necessitam de habilidades cognitivas (WHITTON, 2010).

Para Brincher e Silva (2011), regras podem definir jogos. Usualmente, em jogos educativos, regras são organizadas para servir o contexto do tema de ensino. Isso faz com que possam ser muito mais complexas e específicas que

regras de jogos de entretenimento. A concepção do ambiente é outro fator determinante em um jogo. Jogos educativos buscam, na maioria dos casos, um ambiente baseado num contexto de domínio específico, buscando sempre a maior verossimilhança possível com a realidade.

Por outro lado, conforme pontuam os autores, jogos criados apenas para o entretenimento tendem a apresentar ambientes mais fantasiosos relacionando-se apenas ao mundo, geralmente imaginário, onde a história acontece.

2.2.2 Acessibilidade em Jogos Digitais

A acessibilidade, de acordo com Iwarsson e Ståhl (2003), pode ser definida como todos os parâmetros que influenciam as ações do ser humano dentro de um contexto específico, ou seja, a acessibilidade é um conceito relativo ao ambiente. Dessa forma podemos avaliar, por exemplo, como um estabelecimento fornece alternativas para o deslocamento de pessoas com deficiência visual ou de mobilidade de forma independente. Alternativamente, se algum serviço atende às expectativas de acesso para todas as pessoas.

Um desenvolvimento que contemple apropriadamente os aspectos visuais e auditivos de jogos digitais é essencial para determinar sua qualidade. Esses aspectos são o que fazem o jogador sentir-se no controle da situação e imerso na história, como cita Schell (2014). Porém, precisam de um cuidado específico quando o assunto é a acessibilidade em jogos digitais. Principalmente, quando o objetivo é desenvolver jogos digitais acessíveis para o maior número de pessoas possível. Para isso é necessário analisar soluções para pessoas com qualquer deficiência seja ela motora, visual, auditiva ou intelectual.

Cheiran (2013, p. 27) lembra que, frequentemente, “o feedback sonoro fornecido pelos jogos não é suficiente para indicar todas as informações necessárias para o entendimento de um cenário”. Outro aspecto que impede pessoas com deficiência de ter uma experiência completa com jogos digitais é a falta de legendas de diálogos e narrações, dificultando a compreensão do contexto atual da história do jogo. Isso também afeta o *feedback* dos eventos do jogo, pois não há representação visual para eventos sonoros importantes. Há, ainda, a questão das dificuldades impostas a pessoas com deficiência pela comunicação entre os jogadores em um jogo *multi-player*. Visto os problemas supracitados de

feedback, mesmo em jogos onde é possível realizar a comunicação por áudio ou teclado, essa comunicação pode atrapalhar a tomada de ações quando o jogador tiver que parar de jogar para digitar.

Diretrizes para a acessibilidade em *websites* já estão bem documentadas e conceituadas dentro da área de desenvolvimento *web*. A IGDA (do inglês, International Game Developers Association) desenvolveu, em 2004, uma lista com o que considera as dez principais diretrizes para aplicar acessibilidade em jogos digitais e, assim, atender às necessidades de jogadores com algum tipo de deficiência, descrevendo-as de forma simples para facilitar o entendimento dos desenvolvedores:

1. **Permitir a configuração de controles:** Oferecer aos jogadores a opção de configuração de controles para melhor adequar-se as suas próprias necessidades de forma confortável.
2. **Prover suporte a controles alternativos:** Não limitar a jogabilidade a controles padrões, como *joysticks*, *mouse* e teclado. Algumas pessoas com deficiência apresentam dificuldades no uso de controles muito complexos.
3. **Oferecer alternativas ao som:** A sonoridade de um jogo é um dos aspectos mais importantes para a imersão na história. Porém, para pessoas surdas e com deficiência auditiva, essa experiência pode ser comprometida. Assim, é importante desenvolver essa imersão de outra forma, com legendas e *closed captions*, além do uso criativo na forma de mostrar o *feedback* visualmente ou pela vibração dos controles.
4. **Permitir a configuração dos sons do jogo:** A possibilidade de configurar, de forma separada, os sons do jogo, como música, efeitos sonoros e diálogo, facilita o entendimento do que está acontecendo na tela.
5. **Gráficos com alta visibilidade:** Devem-se evitar fontes pequenas ou de difícil distinção na tela, prover a opção de um esquema de cores em alto-contraste e destacar itens importantes na tela.
6. **Design amigável a pessoas com daltonismo:** Certas combinações de cores e tons de cores podem impossibilitar a distinção de elementos na tela por pessoas com daltonismo. Deve-se evitar depender apenas de cores para apresentar alguma ação dentro do jogo ou, pelo menos, garantir que elas possam ser entendidas por todos.

7. **Oferecer diversas opções de modos de dificuldade/velocidade:** É necessário entender que não existe um modo “fácil demais”. Jogadores com dificuldades visuais podem ter a experiência facilitada com modos mais lentos ou mais fáceis do jogo.
8. **Oferecer modos de tutorial e treinamento:** Esses modos podem ajudar na compreensão, nos ajustes do controle e no desenvolvimento das habilidades dentro do jogo.
9. **Menus acessíveis:** Menus com a opção de leitura dos textos da tela são uma ferramenta importante para acessibilidade.
10. **Listar a acessibilidade como uma das funcionalidades do jogo:** É importante que as informações sobre a acessibilidade do jogo sejam de fácil acesso e de claro entendimento.

Baseado em diretrizes para jogos digitais, o trabalho de Cheiran (2013) apresenta um *checklist* adaptado para a avaliação de acessibilidade (ANEXO A). Esse *checklist* pode ser aplicado em um jogo digital com o objetivo de conhecer seus principais problemas de acessibilidade.

2.3 Padrões de Projeto de Interface de Usuário

Segundo Talarico Neto *et al.* (2004, p. 4), um padrão é “uma solução comprovada escrito por pessoas experientes que já enfrentaram um problema várias vezes em determinadas instâncias de um contexto”. Folmer (2015) define que um padrão de interface de usuário é uma solução repetível de forma geral para um problema frequente de usabilidade. Esse padrão, geralmente, consiste de sete elementos:

1. **Problema:** O problema relacionado à usabilidade do sistema;
2. **Quando usar:** Em qual contexto do sistema o problema ocorre;
3. **Princípio:** Um padrão é baseado, geralmente, em princípios como consistência ou gerenciamento de erro;
4. **Solução:** Uma solução comprovada para o problema. Essa solução deve descrever apenas o problema principal, dando, dessa forma, maior liberdade para o desenvolvedor implementar de diversas maneiras;

5. **Como e Por que:** Inclusão de uma análise de impacto nos atributos de usabilidade e definir como o padrão funciona;
6. **Exemplos:** Apresentação de exemplos dos padrões aplicados em sistemas reais;
7. **Implementação:** Alguns padrões apresentam detalhes de implementação.

Outros autores também apresentam elementos que podem constar em um padrão de projeto de interface de usuário, como Talarico Neto *et al.* (2004) e Kultsova *et al.* (2017), apresentando os mesmos elementos configurados de formas diferentes.

2.4 Evolução de Software

A Evolução de Software é uma das grandes áreas da Engenharia de Software. Sommerville (2011) argumenta que é uma fase inevitável da vida útil de um sistema, pois mudanças podem partir da necessidade de atualização de requisitos pedidos pelos usuários, das mudanças de negócios em empresas ou para que o *software* se adapte a plataforma de *hardware* ou *software*.

É importante notar, contudo, que a Evolução de Software é um processo que, em muitos casos, pode ter sua abrangência reduzida se o desenvolvimento for realizado de forma correta. Existem diversas alterações que podem ser aplicadas ao *software* depois de sua entrega, desde alterações no código, mudança de tecnologia ou a necessidade de novas funcionalidades serem implementadas. Essas alterações podem ser classificadas em três tipos, segundo Sommerville (2011):

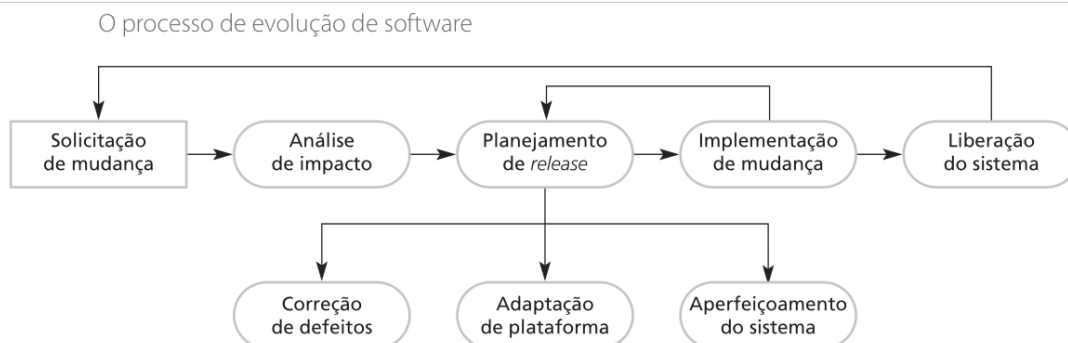
1. **Correção de defeitos:** Os erros de codificação, geralmente, tendem a ser a correção menos custosa, com correções de projeto e de requisitos sendo a parte mais custosa, pelo tamanho dos artefatos a serem avaliados.
2. **Adaptação ambiental:** Essa mudança ocorre quando algum elemento que propicia o uso do *software*, como *hardware* ou o sistema operacional sofre alguma alteração e é necessário alterar o *software* para se adaptar a essa nova condição.
3. **Adição de funcionalidade:** Ocorre geralmente quando novas formas de negócio são criadas em um contexto empresarial ou no *feedback* de usuários

sobre funcionalidades que são necessárias para que o sistema cumpra seu objetivo.

Contemplar a acessibilidade em um *software* que não foi projetado e desenvolvido com esse requisito em perspectiva é uma tarefa desafiadora. É preciso identificar claramente os problemas que podem impedir seu uso por alguns usuários e definir estratégias para corrigi-los. Um modelo sistemático para a Evolução do Software deve colaborar a esse propósito.

Sommerville (2011) apresenta um modelo de processo de Evolução de Software (Figura 1), com etapas definidas e que pode ser replicado diversas vezes. Esse processo inclui as etapas de análise de impacto, planejamento de *release*, implementação de sistema e liberação do sistema para os usuários. Nesse processo, são avaliados o custo e o impacto de mudanças, além de serem discutidas quais mudanças serão implementadas.

Figura 1 – Representação gráfica do processo de evolução de *software*



Fonte: Sommerville (2011, p. 167).

2.5 Scrum

Segundo Schwaber e Sutherland (2020), o Scrum é um *framework* que tem como objetivo auxiliar pessoas e equipes a trabalhar com soluções adaptativas para um desenvolvimento iterativo. Essa abordagem, ao contrário de outros métodos ágeis não determina práticas de programação específicas, tornando-a mais flexível para ser aplicada independente do contexto de tecnologia utilizado no desenvolvimento.

O Scrum funciona a partir da coordenação de um Scrum Master, papel que tem a função de gerenciar um ambiente onde o Product Owner (PO) define as tarefas para resolver um problema, a partir de um Backlog do Produto. Após isso, a equipe de desenvolvimento realiza as atividades pré-definidas durante a Sprint, para que os *stakeholders* do projeto possam avaliar os resultados e definir os objetivos para a próxima Sprint, sucessivamente, até a entrega final do projeto.

Os princípios do Scrum giram em torno da simplicidade de sua implementação e do empirismo. Dessa forma, os diversos processos, técnicas e métodos empregados com este *framework* fazem com que ele seja amplamente flexível, sendo desenvolvido a partir da experiência das pessoas que o utilizam, em um determinado contexto.

Os Sprints são uma unidade de planejamento onde o software é implementado, eles são planejados em um *Sprint Planning*, onde o Product Owner e os desenvolvedores discutem quais itens do Backlog do Produto serão desenvolvidos. O principal evento do Scrum são os Sprints e geralmente duram de duas a quatro semanas, e utilizam o Backlog do Produto, que tem o Product Owner, como responsável pelo seu conteúdo, disponibilidade e avaliação, podendo ser organizadas dentro de prioridades e com riscos identificados.

Durante a execução dos *sprints* a comunicação é realizada apenas pelo Scrum Master, fazendo com que a equipe foque no desenvolvimento do produto. Reuniões diárias são realizadas para analisar o progresso e, se necessário, alterar alguma prioridade. O Scrum Master é responsável por garantir que todas as atividades básicas do Scrum funcionem. Durante essa etapa é criado o *Sprint Backlog*, documento que contém o que deve ser feito no Sprint que se inicia. Após o fim de cada Sprint, a funcionalidade completa deve ser entregue e uma revisão do *sprint* deve ser realizada. Quando todas as tarefas forem realizadas a partir dos sprints, o sistema deverá estar pronto.

2.6 Considerações Finais do Capítulo

Com os jogos digitais sendo bastante utilizados em contextos de ensino, é fundamental que haja uma preocupação maior com sua acessibilidade, de modo que mais estudantes possam desenvolver aprendizagens significativas, de forma lúdica.

Para contemplar adequadamente requisitos de acessibilidade, é necessário que o desenvolvimento de *software* tenha etapas bem definidas com instruções de como abordá-los. Um Game Design Document robusto, que possa esclarecer como as mecânicas e principais delimitações do jogo devem ser executadas, é importante para esse processo. Além disso, diretrizes de acessibilidade para jogos digitais são ferramentas essenciais para o desenvolvimento de jogos amplamente acessíveis (IGDA, 2004). É possível, através dessas diretrizes, desenvolver jogos que tenham consistência e coerência com os padrões de acessibilidade para usuários com qualquer tipo de deficiência, seja ela motora, visual, auditiva ou intelectual.

Uma ferramenta que pode auxiliar nesse processo são os padrões de projeto de interface de usuário. Como jogos digitais compartilham conceitos específicos, que podem ser dependentes do gênero do jogo ou mesmo de sua plataforma, os padrões de projeto de interface podem colaborar para lidar com possíveis problemas de interação. Utilizando esses padrões, problemas de interação com o usuário, como também os de acessibilidade, podem ser corrigidos de forma sistemática e organizada, além de manter uma coerência dentro do jogo.

A Evolução de Software é uma importante ferramenta para contemplar a acessibilidade em um jogo já existente. Sommerville (2011) afirma que a evolução de um sistema raramente pode ser considerada de forma isolada. Portanto, é necessário entender como esse processo e suas atividades podem auxiliar na construção da acessibilidade em jogos digitais.

3 ESTADO DA ARTE

Neste Capítulo, são apresentadas as principais características da pesquisa bibliográfica deste trabalho, que se utilizou de algumas etapas do processo de Revisão Sistemática de Literatura, desenvolvido por Kitchenham (2004). Também é realizado um esforço para responder as questões de pesquisas e uma análise dos resultados encontrados.

Primeiramente, as questões de pesquisa foram enunciadas. Então, pesquisas preliminares foram realizadas com palavras e expressões relacionadas a essas questões, procurando-se ter acesso não somente a revisões de literatura já existentes sobre os temas, mas também para saber qual o potencial volume de estudos relevantes encontrados. Com o resultado dessas pesquisas preliminares em mãos, foi realizado o processo de definição de *strings* de busca de cada uma das questões de pesquisa definidas. Os critérios de inclusão e exclusão foram definidos para filtrar os artigos encontrados a partir do que se busca pesquisar.

Por fim, foi realizado, durante a primeira parte do mês de setembro, a aplicação dessas *strings* nas bases de dados, e com o artigos retornados foi efetuada uma seleção desses estudos utilizando os critérios de inclusão e exclusão.

3.1 Questões de Pesquisa

As questões enunciadas são as seguintes:

1. Como cada etapa do processo de desenvolvimento de jogos digitais educacionais contempla acessibilidade?
2. Como a evolução de *software* desenvolve padrões de acessibilidade?
3. Que padrões de projeto de interface de usuário contribuem à promoção da acessibilidade em jogos digitais?
4. Como padrões de projeto de interface de usuário para jogos digitais têm contribuído à promoção de sua acessibilidade?

3.2 Bases de Busca

As bases de dados consultadas foram a ACM Digital Library, IEEE Xplore Digital Library e Scopus. Os anais do evento SBGames (Simpósio Brasileiro de Games e Entretenimento Digital) foram considerados para a pesquisa, porém, como em alguns anos seus artigos não estavam disponíveis e por abordarem vagamente o tema do trabalho, os artigos desse evento foram descartados. No Quadro 1, a seguir, estão organizadas as bases utilizadas e seus respectivos *websites* na Internet.

Quadro 1 – Bases de Busca

Nome	URL
ACM Digital Library	https://dl.acm.org
IEEE Xplore Digital Library	https://ieeexplore.ieee.org
Scopus	https://www.scopus.com
SB Games	https://www.sbgames.org/

Fonte: O próprio autor.

3.3 Strings de Busca

Foram definidas *strings* de busca específicas para cada questão de pesquisa e ajustadas para cada base de busca utilizada. A ferramenta Thoth (LESSE, 2018) foi utilizada para auxiliar na modificação das *strings*, de acordo com os critérios de formatação das bases de busca selecionadas para consulta.

Nos Quadros 2, 3 e 4, a seguir, estão organizados o número de resultados da busca com as *strings* para cada questão de pesquisa em suas respectivas bases de dados, antes dos critérios de inclusão e exclusão serem aplicados.

Quadro 2 – Strings de Busca e Resultados da Base ACM Digital Library

Questão de Pesquisa	String de busca	Número de Resultados
QP1	(accessibility) AND ("digital games") AND (education educative educational instructional)	25
QP2	(accessibility) AND ("software evolution")	45
QP3	(accessibility) AND ("design patterns") AND ("user interface")	17
QP4	(accessibility) AND ("user interface") AND ("digital games")	5

Fonte: O próprio autor.

Quadro 3 – Strings de Busca e Resultados da Base IEEE Xplore Digital Library

Questão de Pesquisa	String de busca	Número de Resultados
QP1	(accessibility AND digital games AND education)	14
QP2	(accessibility AND software evolution)	59
QP3	(accessibility AND design patterns AND user interface)	17
QP4	(accessibility AND digital games AND user interface)	6

Fonte: O próprio autor.

Quadro 4 – Strings de Busca e Resultados da Base Scopus

Questão de Pesquisa	String de busca	Número de Resultados
QP1	TITLE-ABS-KEY ((accessibility) AND ("digital games") AND (education OR educative OR educational OR instructional))	19
QP2	TITLE-ABS-KEY ((accessibility) AND ("software evolution"))	4
QP3	TITLE-ABS-KEY ((accessibility) AND ("design patterns") AND ("user interface"))	18
QP4	TITLE-ABS-KEY ((accessibility) AND ("digital games") AND ("user interface"))	5

Fonte: O próprio autor.

O Simpósio Brasileiro de Games e Entretenimento Digital (SB Games) também foi utilizado para a coleta de artigos. Porém, a busca foi realizada de forma manual nos anais do evento. A indisponibilidade do texto completo de

alguns artigos fez com que o número de artigos selecionados fosse bem menor em relação às outras bases de dados. E como esses estudos não citavam o tema do trabalho de forma específica, foram descartados.

3.4 Critérios de Inclusão e Exclusão

Os critérios de inclusão e exclusão foram definidos com o intuito de identificar os estudos que relatam diretamente sobre o tema de pesquisa como também para evitar que artigos que se distanciem do tema possam ser coletados para análise. Geralmente, mais de um pesquisador determina os critérios de exclusão e inclusão para auxiliar na consistência dos resultados obtidos pelos filtros, utilizando opiniões diferentes para validar os estudos pesquisados. Segundo Kitchenham (2004), porém, quando apenas um pesquisador define os critérios, é recomendável que ele os debata com alguém que tenha experiência. No caso deste trabalho, os critérios de inclusão e exclusão foram discutidos com a orientadora do TCC.

Os critérios de inclusão foram aplicados para validar os estudos coletados e identificar quais seriam utilizados pelo pesquisador. Sua aplicação se deu após a coleta de todos os artigos nas bases de dados. São eles:

1. Artigos completos de eventos e periódicos;
2. Artigos em português ou inglês;
3. Artigos que abordam acessibilidade em jogos digitais;
4. Artigos que abordam acessibilidade no desenvolvimento de *software*;
5. Artigos que abordam padrões de acessibilidade em evolução de *software*;
6. Artigos que abordam padrões de projeto de interface de usuário para jogos digitais e acessibilidade.

Com o objetivo de descartar estudos que não condizem com o tema do trabalho, os critérios de exclusão foram aplicados logo após a aplicação dos critérios de inclusão. São eles:

1. Artigos resumidos, resumos ou capítulos de livro ou editorais;
2. Artigos cujo texto completo não está disponível;
3. Artigos duplicados;
4. Artigos que não abordem acessibilidade em interfaces de usuário;

5. Artigos que abordam soluções de acessibilidade voltadas a um público específico (ex.: pessoas cegas, pessoas surdas, pessoas com mobilidade reduzida, pessoas com transtorno do espectro autista, pessoas com deficiência intelectual etc.).

3.5 Análise dos Resultados

Após a coleta e a seleção dos estudos nas bases de dados uma análise foi realizada com o objetivo de responder às questões de pesquisa. Para isso, foram analisados os vinte e dois artigos selecionados após os critérios de inclusão e exclusão terem sido aplicados. Dessa forma, foi possível compreender qual o atual estado da arte no assunto abordado neste trabalho.

3.5.1 Como cada etapa do processo de desenvolvimento de jogos digitais educacionais contempla a acessibilidade?

Segundo estudos de Szykman (2015), o interesse de pesquisadores em estudar e desenvolver jogos para pessoas com deficiência vem aumentando. Diversas formas de oferecer a acessibilidade em jogos são aplicadas em jogos digitais de diversos formatos, porém em muitos casos não atingem funcionalidades que tornem o jogo mais atrativo nem conseguem simular a imersão requerida (ARAÚJO *et al.*, 2017). Nesses estudos, não há especificação das etapas de desenvolvimento de um jogo com foco na acessibilidade. Porém, apresentam métodos, como de aperfeiçoar o *feedback* e tornar a imersão mais completa para usuários com deficiência, que podem ser utilizados para aplicar diretrizes de acessibilidade.

Torrente *et al.* (2012) citam a importância das configurações flexíveis dentro do jogo para tornar a experiência do jogador mais confortável, além de ter pessoas com deficiência durante os experimentos e durante a construção do jogo para receber um *feedback* mais preciso. Nesse estudo, um dos exemplos de diretrizes pré-estabelecidas que necessitaram de uma avaliação melhor é a do modo de cores de alto-contraste. No experimento, pessoas com deficiência intelectual puderam jogar o jogo “My First Day at Work”, que foi desenvolvido com o auxílio de pessoas que normalmente usam configurações de cores em alto-contraste para

interagir com tecnologia digital. Porém, nem todos os participantes se sentiram confortáveis enquanto jogavam.

Na avaliação de interfaces acessíveis de jogos digitais do tipo apontar e clicar (do inglês, *Point-and-Click*), Torrente *et al.* (2013) apresentam alguns pontos importantes para o entendimento de como diferentes tipos de interfaces e tecnologias recomendadas para a criação de jogos digitais acessíveis afetam a jogabilidade. Tecnologias como sonar e a navegação por teclado podem apresentar problemas com o dinamismo necessário para deixar um jogo atrativo. Primeiro, o sistema de leitura de texto do jogo não lia a descrição de elementos quando o sonar identificava a proximidade do cursor duas vezes seguidas. Já com a navegação por teclado, o sistema de leitura de texto do jogo lia por cima as descrições de elementos do jogo, ou seja, lia os dois elementos selecionados pela navegação do teclado, quando essa navegação ocorria de maneira rápida.

Conforme citam Hauge *et al.* (2018) é imprescindível para o baixo custo de se implementar acessibilidade em qualquer *software* o seu desenvolvimento desde o início, porém é importante que não especificam um modelo de processo específico ou suas etapas de trabalho. Ou seja, para tornar um sistema acessível é necessário pensar a acessibilidade junto com as primeiras ideias de desenvolvimento das interfaces, ou até mesmo em como o usuário irá interagir com o sistema.

Hersh e Leporini (2012) citam a importância de desenvolver jogos digitais acessíveis que sejam compatíveis com uma variedade de recursos de Tecnologia Assistiva. Entre os principais recursos estão:

1. Teclado adaptados para o uso com apenas uma mão;
2. Teclado virtual;
3. Dispositivos que utilizam apenas um movimento, como um selecionador binário;
4. Controles *joystick*;
5. Teclado em Braille;
6. Software de reconhecimento de voz;
7. Leitores de tela.

Os autores desse artigo levantam também a questão das diferentes experiências de usuários com deficiências distintas e como isso deve ser levado em conta durante a busca por soluções efetivas que abranjam o maior número possível de *gaps* de acessibilidade. Um exemplo disso é o uso de línguas de sinais para usuários surdos. Como pessoas surdas estão mais habituadas com a comunicação por línguas de sinais, a leitura durante o jogo pode retirar o dinamismo que uma atividade necessita e quebrar a imersão proporcionada durante a experiência do jogo. Porém, é necessário ressaltar, como diz Hersh e Leporini (2012, p. 13), em tradução livre: “Assim como ações rápidas são uma funcionalidade importante de um jogo, saber como deixar o jogo atrativo durante um momento mais lento é uma característica importante do desenvolvedor”.

Outra questão importante citada pelos autores é a dificuldade de expandir jogos digitais acessíveis com novas funcionalidades ou módulos. Chegou-se a essa conclusão após uma pesquisa de escala pequena sobre a experiência de estudantes do ensino superior com deficiência em universidades do Reino Unido, em 2008, dentro de um contexto de ambiente de aprendizado virtual.

3.5.2 Como a evolução de software desenvolve padrões de acessibilidade?

Nos estudos selecionados não houve menções ao desenvolvimento de padrões de acessibilidade durante o processo de evolução de *software*. Porém, foram citados avaliações de acessibilidade durante a evolução de *software*. Segundo Kobayashi, Letizio e Tanaka (2011), existem duas formas de se avaliar a acessibilidade: revisões de diretrizes e testes com usuários. Apesar de a utilização de diretrizes ser a forma de avaliação mais difundida, por ser sistemática e usualmente mais barata por não precisar de um número grande de pessoas para ser realizada, os testes de usuário são essenciais para uma boa avaliação de acessibilidade. Isso ocorre porque muitos recursos acessíveis são relativos a uma condição específica que pode passar despercebida ou não ser apontada por uma diretriz. Outro ponto importante é realizar essa avaliação com pessoas com deficiência, porque um usuário sem deficiência pode apresentar vícios de interação com o *software* que interferem na avaliação de usabilidade e diferem do cenário utilizado por quem possui algum tipo de deficiência.

Oliveira, Mota e Leite (2018) aplicam uma estratégia de reengenharia com o objetivo de evoluir um *software*, um aplicativo real chamado PhoneAdapter, e torná-lo mais acessível. Essa estratégia, baseada no conceito de Consciência de Software, foi organizada em três atividades: Recuperar o Modelo de Metas, Reespecificar o Sistema Usando o SIG (do inglês, Softgoal Interdependency Graphs) Híbrido e, por fim, Reimplementar o Sistema. Apesar de essa estratégia ter funcionado, algumas questões ainda precisam ser melhor trabalhadas. Para realizar a evolução do sistema, o arquiteto de *software* e o desenvolvedor trabalharam com os artefatos gerados durante a aplicação das etapas da estratégia citada. Isso faz com que um conhecimento prévio sobre os conceitos envolvidos, não apenas de acessibilidade, mas também de Consciência de Software, seja um pré-requisito importante. Outro ponto está relacionado à validade de aplicação, pois o estudo foi realizado utilizando uma quantidade pequena de aplicativos e, além disso, apenas os autores participaram da aplicação da abordagem.

Os estudos encontrados não apresentaram nenhum processo de evolução de *software* para aplicar a acessibilidade, que aborde jogos digitais ou mesmo etapas que possam ser adaptadas para esse contexto. Isso pode ser explicado pelo fato de desenvolvedores ainda pensarem o tópico de acessibilidade como uma funcionalidade complexa e custosa, como citam Letizio e Tanaka (2012).

3.5.3 Que padrões de projeto de interface de usuário que contribuem à promoção da acessibilidade em jogos digitais?

Nos estudos de Kultsova *et al.* (2017) é possível encontrar uma lista, referenciada de um manual, contendo as melhores práticas, com base em recomendações de acessibilidade, para serem aplicadas em padrões de projeto de interface de usuário para o desenvolvimento *web*. Essas recomendações, em sua maioria, podem ser utilizadas como base para o desenvolvimento de interfaces de usuário para jogos digitais, assim como para *websites*. Entre elas estão:

- 1. Forneça textos alternativos para elementos não textuais:** É importante garantir que todos os botões, imagens e ícones tenham uma descrição com sentido e concisa possibilitando sua leitura por leitores de tela.

2. **Evite imagens de textos:** Leitores de tela, geralmente, não conseguem ler textos dentro de imagens. Logo, utilizar textos no lugar de imagens ou, no mínimo, fornecer textos alternativos, é considerado uma boa prática.
3. **Possibilite configuração de tamanho do texto:** É importante garantir que o tamanho do texto possa ser aumentado ou diminuído sem comprometer seu conteúdo.
4. **Codifique os elementos da tela em ordem lógica:** Se o conteúdo precisa ser lido em uma ordem específica, é necessário garantir que a codificação dos elementos mantenha essa ordem.
5. **Evite depender somente de formas, elementos visuais ou elementos auditivos para navegar na tela:** Além de identificar textualmente os elementos da tela, é importante fornecer textos alternativos para uma compreensão completa de pessoas com deficiência visual, que possam usar um leitor de telas, ou pessoas com deficiência auditiva.
6. **Evite depender somente de cores para transmitir informações relevantes:** Pessoas com baixa visão ou com daltonismo têm dificuldades em distinguir cores, logo, uma informação que pode ser clara para pessoas sem esses tipos de deficiência pode passar despercebida para aqueles que a têm.
7. **Utilize cores com contraste suficiente para facilitar a visualização:** Escolha as cores de fundo e de texto apropriadas para terem um contraste suficiente para pessoas com daltonismo terem clareza do texto informado.
8. **Utilize mais de uma forma de *feedback* ao usuário como som, vibração e visualização:** É necessário fornecer mais de uma forma de notificar o usuário que possam ser percebidas por pessoas com deficiência visual ou auditiva.
9. **Forneça descrição de vídeos:** Para pessoas com deficiência visual, um vídeo pode ser ouvido, porém sem uma descrição do que acontece durante ele pode ser que informações importantes se percam, como descrição das ações, personagens, mudança de cenário e textos na tela.
10. **Forneça legenda para vídeos:** Para pessoas com deficiência auditiva a legenda é uma ferramenta importante para o entendimento do conteúdo apresentado.
11. **Forneça língua de sinais para vídeos:** A língua de sinais é um método universal de comunicação para pessoas com deficiência auditiva. Além de

melhorar a dinâmica das informações fornecidas, a língua de sinais pode ser mais fácil de compreender do que o texto da legenda.

12. Forneça alternativas para informações sonoras: Informações que só aparecem em áudio necessitam de uma alternativa para ser comunicada. Texto descritivo é uma boa solução para este problema.

13. Forneça uma estrutura da interface consistente e simples: Mudanças bruscas de *layout* e configuração de botões ou ações a serem executadas podem confundir o usuário.

14. Forneça identificação de erros: Se o usuário errar, utilize meios de esclarecer onde o erro ocorreu como solucioná-lo.

15. Forneça visibilidade no foco de um elemento importante: Quando um elemento necessita de atenção para executar uma ação, é importante que o sistema foque nele de forma clara para auxiliar o usuário a identificá-lo.

3.5.4 Como padrões de interface de usuário para jogos digitais têm contribuído à promoção de sua acessibilidade?

Padrões de interface de usuário auxiliam na construção de interfaces utilizando as melhores práticas e diretrizes de usabilidade para cada contexto. Em alguns casos, isso facilita o desenvolvimento de requisitos de acessibilidade (FOLMER, 2007). Nesses padrões são definidas as formas de configuração da interface em diversos níveis, como alteração do tamanho da fonte, que auxiliam pessoas com dificuldades visuais, e o modo de apresentação de informações que pode facilitar o entendimento para pessoas com deficiência mental ou visual. Quando esses requisitos são bem trabalhados e desenvolvidos, é possível ter uma solução para contemplar a acessibilidade já no projeto de software, focando em sua interface de usuário. Reunir todas essas configurações, porém, pode ser um obstáculo. Como citam Kultsova *et al.* (2017), criar um banco de dados com esses padrões de interface é essencial para a aplicação dessa abordagem ser coerente.

Dessa forma, é possível construir funcionalidades que sejam acessíveis para o maior número possível de pessoas com as mais variadas deficiências, sem precisar desenvolver soluções para cada grupo específico. Também é importante definir como esses padrões de interface podem ser catalogados de forma clara

para a rápida identificação do problema que está sendo trabalhado e da solução que ele visa resolver.

3.6 Considerações Finais do Capítulo

Durante a construção deste Capítulo, foi possível identificar os principais aspectos envolvidos na realização da acessibilidade em etapas da Engenharia de Software. Também foi possível observar a necessidade de abordar explicitamente a acessibilidade em processos de desenvolvimento de *software* e de evolução, visto que a acessibilidade geralmente é tratada apenas como um componente menos importante durante a construção de um sistema. Esse tratamento, usualmente, deixa a construção do *software* mais cara quando há a necessidade de torná-lo acessível. Porém, não foi possível compreender como cada etapa de desenvolvimento trata a acessibilidade. Nos estudos pesquisados, não houve uma especificação de como levantar esses requisitos, apesar de testes de usuário e *checklists* serem apresentados como ferramentas para validação.

A acessibilidade, no contexto de evolução de *software*, é citada apenas como uma consequência do processo. Se um sistema foi desenvolvido com requisitos mínimos de acessibilidade, a tendência é que durante a evolução desse sistema a acessibilidade venha a evoluir também. Porém, não há uma preocupação em identificar quais os problemas mais comuns por quem utiliza o sistema e assim definir como solucioná-los. Dessa forma, é necessário entender quais problemas de acessibilidade são mais importantes de resolver para facilitar o acesso a pessoas com deficiência e, principalmente, como aplicar esses requisitos de forma coerente e padronizada.

Outra questão importante é a necessidade de definição de padrões de projeto de interface de usuário que possam ser utilizados tanto em *websites* como em jogos digitais, facilitando a utilização de recomendações de acessibilidade. Além disso, entender como esses padrões resolvem problemas de acessibilidade pode auxiliar na adaptação de modelos de desenvolvimento de *software*. A seguir, no Quadro 5, são apresentados os artigos utilizados neste trabalho e suas características que, apesar de não terem respondido às questões de pesquisa diretamente, ajudaram a entender o contexto das áreas de jogos, acessibilidade e evolução de *software* em conjunto. Além disso, esses trabalhos subsidiam a

construção da metodologia deste trabalho para realizar a evolução do jogo digital Programmer.

Quadro 5 – Contribuição dos artigos coletados para o Estado da Arte

Artigo	Questões Respondidas	Contribuição
Szykman (2015)	QP1	Dados sobre o crescimento do interesse de pessoas em pesquisar e desenvolver jogos para pessoas com algum tipo de deficiência.
Araújo <i>et al.</i> (2017)	QP1	Como aspectos de acessibilidade interferem na jogabilidade e imersão do jogo.
Torrente <i>et al.</i> (2012)	QP1	A importância de uma opção vasta de configurações para o usuário sentir-se confortável dentro da experiência do jogo.
Torrente <i>et al.</i> (2013)	QP1	Apresenta diferenças na jogabilidade de jogos acessíveis entre tecnologias e dispositivos.
Hauge <i>et al.</i> (2018)	QP1	Demonstra a diferença entre tratar funcionalidades acessíveis em qualquer software desde o seu início e após sua liberação, em relação aos custos do projeto.
Hersh e Leporini (2012)	QP1	Cita os principais recursos de Tecnologia Assistiva.
Kobayashi, Letizio e Tanaka (2011)	QP2	Cita as formas de se avaliar a acessibilidade em um software.
Oliveira, Mota e Leite (2018)	QP2	Apresenta uma estratégia de evolução de software com intuito de tornar um sistema mais acessível.
Letizio e Tanaka (2012)	QP2	Apresenta que muitos desenvolvedores ainda têm um conceito antiquado da acessibilidade.
Kultsova <i>et al.</i> (2017)	QP3, QP4	Apresenta uma lista com melhores práticas baseada em recomendações de acessibilidade para serem aplicadas em projeto de interface de usuário. Cita a importância da criação de um banco dados com padrões de interface de usuário para o desenvolvimento de software acessível.

Fonte: O próprio autor.

4 METODOLOGIA

A metodologia para o desenvolvimento deste TCC, apresentada neste Capítulo, utilizou conhecimentos de processos de Evolução de Software (SOMMERVILLE, 2011), conceitos apresentados sobre acessibilidade (CHEIRAN, 2013; IGDA, 2004), as definições de jogos digitais (SCHELL, 2014) e sua utilização em contextos de ensino (BRINCHER; SILVA, 2011), assim como as principais diretrizes para se trabalhar com acessibilidade no desenvolvimento de software (HAUGE *et al*, 2018).

Neste Capítulo, é descrita uma visão geral sobre o jogo Programmer, assim como são detalhados os processos definidos para o desenvolvimento de sua evolução. Também são apresentados os motivos para adaptação do processo de evolução de Sommerville (2011). A terceira etapa deste processo consiste na execução do processo de evolução definido para este trabalho, que é detalhado na seção 4.2.3 Evolução de Software.

4.1 O jogo digital Programmer

Programmer é um jogo digital educacional para plataforma *web*, do gênero *puzzle*. O jogo é dividido em desafios, sendo que os desafios representam algoritmos que o usuário deve solucionar para resolver um problema proposto. O objetivo do jogador no desafio é montar um código que resolve um problema. Para isso, deverá escolher entre os blocos de códigos e montá-los na ordem correta. Isso deve ser realizado dentro de um tempo determinado.

4.1.1 Regras do Jogo

Existem apenas duas regras que delimitam as ações do jogador:

- 1. Tempo limite para resolver os desafios:** O jogador tem um número delimitado de minutos para responder as questões propostas. Esse tempo é contado de forma crescente começando em zero segundo até dez minutos.
- 2. Soluções pré-definidas:** O jogador já terá as soluções construídas previamente, logo só poderá organizar o código da forma correta, sem poder realizar alterações nas linhas de código.

4.1.2 Personagens

Os personagens existentes no jogo são o “Programador”, personagem controlado pelo jogador, e o “Chefe”, personagem que apresenta os enunciados dos desafios a serem realizados.

4.1.3 Controles

O jogador utiliza apenas o cursor para realizar as ações do jogo, utilizando como ferramenta o *mouse* ou o *touchpad* para isso. A navegação por teclado ainda não está habilitada. Os movimentos realizados são de selecionar os desafios antes de iniciá-los e arrastar as soluções corretamente, organizando as respostas.

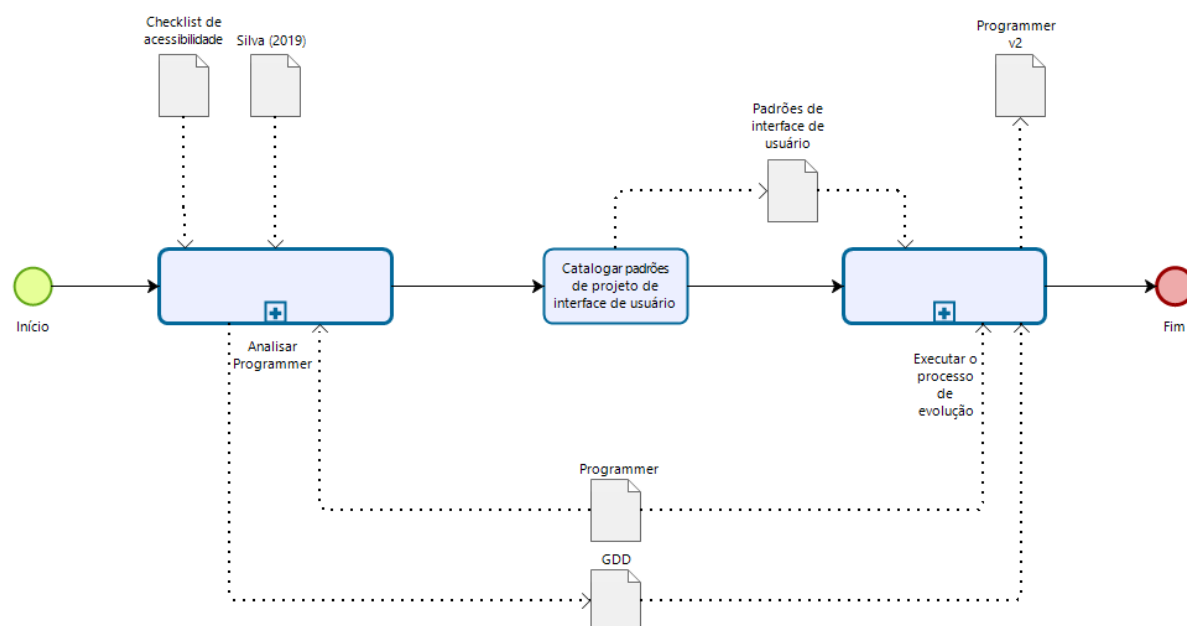
4.2 Metodologia para conduzir a evolução do jogo digital Programmer

Após a verificação do estado da arte, constatou-se a necessidade de delimitar um processo para definir especificamente os passos para realizar a evolução, com requisitos de acessibilidade, do jogo digital Programmer.

Os principais objetivos do trabalho são a promoção da acessibilidade em jogos educacionais pela aplicação de padrões de projeto de interface de usuário no desenvolvimento de jogos digitais, a catalogação de padrões de projeto de interface de usuário e a evolução do jogo digital Programmer. Para realizar esses objetivos, primeiramente foi realizado um estudo sobre os principais pontos que compõem o desenvolvimento de software com requisitos de acessibilidade, o desenvolvimento de jogos digitais e como aplicar esses conceitos dentro de um processo de evolução de software. Em seguida, foi determinado um modelo de catalogação dos padrões de projeto de interface de usuário para, por último, auxiliar no desenvolvimento dos requisitos de acessibilidade do jogo Programmer.

A Figura 2, a seguir, ilustra a metodologia para conduzir a evolução do jogo digital Programmer.

Figura 2 – Metodologia para conduzir a evolução do jogo digital Programmer

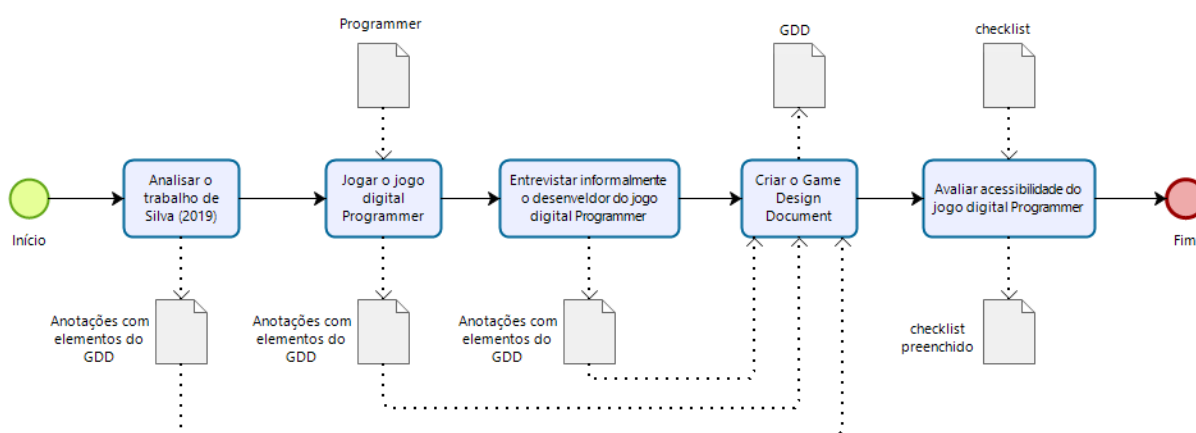


Fonte: O próprio autor.

4.2.1 Analisar Programmer

Como ilustrado na Figura 3, o desenvolvimento do *Game Design Document* (GDD) foi realizado utilizando três abordagens: A análise do trabalho de Silva (2018), a utilização do jogo Programmer e uma entrevista informal com o desenvolvedor do jogo.

Figura 3 - Representação gráfica do subprocesso de analisar Programmer



Fonte: O próprio autor.

A análise de Silva (2019) teve como objetivo conhecer as principais mecânicas do jogo e entender, de forma sucinta, quais são seus objetivos. Para utilizar o jogo Programmer foi necessário acessar sua versão *online*, disponibilizada pelo desenvolvedor do jogo, que esclareceu questões sobre a jogabilidade. Após a utilização do jogo, uma entrevista informal com o desenvolvedor foi realizada com questionamentos sobre os principais elementos que compõem um *Game Design Document* (HIRA; *et al.*, 2016). Esses questionamentos serviram de base, assim como a avaliação do jogo e a leitura do trabalho de Silva (2019), para construir o GDD.

A primeira etapa da metodologia foi definida para realizar a análise do jogo digital Programmer. Essa atividade consiste em avaliar o jogo utilizando o *checklist* de acessibilidade desenvolvido por Cheiran (2013), apresentado no ANEXO A, e com o auxílio de diretrizes desenvolvidos pelo autor, com o objetivo de entender e documentar seus principais problemas de acessibilidade. A escolha desse *checklist* foi feita a partir da análise do trabalho de Cheiran (2013, p.83), que cita a adaptação do *checklist* como uma “verificação da consistência e da abrangência das diretrizes compiladas” pelo estudo.

Como complemento dessa atividade está a análise do trabalho de Silva (2019), procurando possíveis recomendações de melhoria, recomendações de trabalhos futuros e os conceitos utilizados para a criação do jogo baseado nos padrões de interface. A partir da realização dessas tarefas, foi possível criar o *Game Design Document* (GDD), considerado importante artefato de análise.

4.2.2 Catalogar padrões de projeto de interface de usuário

Após a primeira etapa, a atividade de “Catalogar padrões de projeto de interface de usuário” foi realizada. Esta etapa configura a identificação e a anotação, em forma de catálogo, de padrões de projeto de interface de usuário com requisitos de acessibilidade, isto é, que especifiquem soluções para o projeto de interfaces de usuários acessíveis a pessoas com algum tipo de deficiência. O modelo utilizado para a catalogação foi o mesmo utilizado pelo trabalho de Silva (2019), que também utilizou o modelo descrito por Folmer (2007). No catálogo desenvolvido neste trabalho foram realizadas adaptações necessárias para abordar os requisitos de acessibilidade.

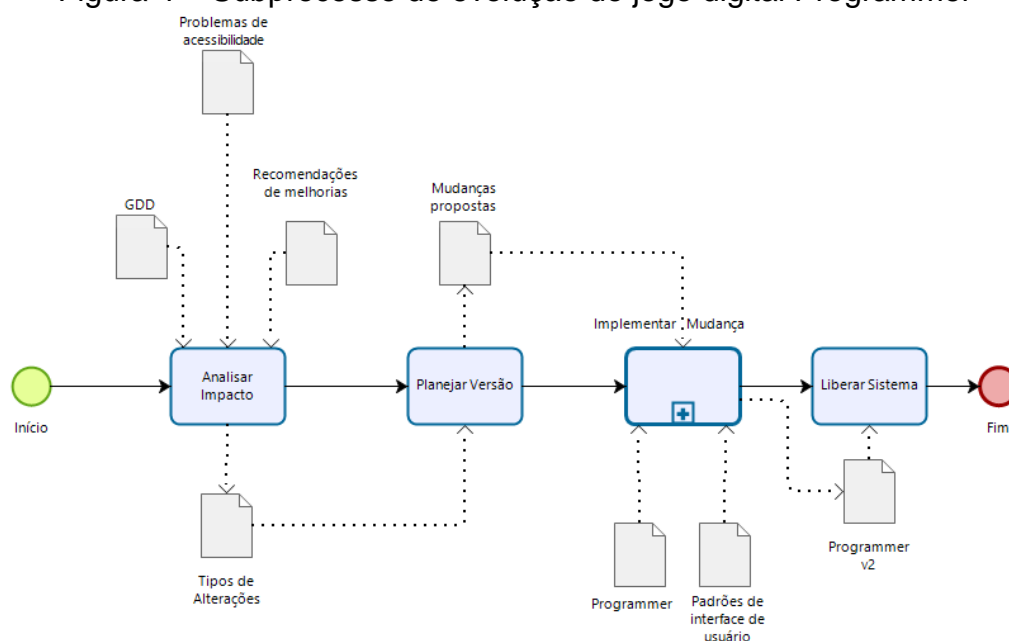
Contemplar soluções acessíveis foi o critério utilizado para adicionar padrões neste catálogo e é importante salientar que, na revisão do estado da arte, foram identificadas boas práticas de acessibilidade para serem aplicadas em padrões de projeto de interface de usuário. Dessa forma, optou-se por uma adaptação de padrões de projeto utilizados no trabalho de Silva (2019) utilizando essas recomendações e diretrizes. Também foram utilizadas as diretrizes apresentadas no trabalho de Cheiran (2013), visto seu detalhamento e da abordagem da acessibilidade em jogos. Com a execução dessa atividade, foi criado um documento contendo padrões de projeto de interface de usuário, contendo as principais informações sobre os padrões de projeto de interface coletados.

4.2.3 Evolução de Software

Este processo de Evolução de Software é uma adaptação do processo que Sommerville (2011) apresenta. Foi necessário realizar uma alteração na etapa do processo intitulada “Planejamento de *release*”, como forma de simplificá-la e aplicá-la ao contexto do estudo. Dessa forma, a etapa de “Planejamento de *release*”, que era subdividida em três opções de mudança – Correção de defeitos, Adaptação de plataforma e Aperfeiçoamento do sistema –, teve seu subdesenvolvimento retirado do processo utilizado neste trabalho. Além disso, a etapa de “Solicitação de mudança”, que ocorria também como forma de *loop* foi

retirada completamente. A primeira etapa apresentada foi alterada, pois o trabalho não tinha como objetivo a correção de um defeito específico e nem de alterar a plataforma *web* atual, sendo assim, o aperfeiçoamento do sistema era a única solução vislumbrada para essa etapa. Já a segunda etapa foi retirada com a justificativa de que como o objetivo de evoluir o *software* só ocorrerá uma vez não há a necessidade de um *loop* dentro do processo, e também pelo motivo de que as solicitações de mudanças já foram identificadas anteriormente, logo, não há a necessidade de essa etapa ocorrer novamente. A Figura 4 ilustra o subprocesso de evolução.

Figura 4 – Subprocesso de evolução do jogo digital Programmer



Fonte: O próprio autor.

Dessa forma, o processo adaptado foi definido com as seguintes etapas: “Analisar impacto”, “Planejar versões”, “Implementar mudança” e “Liberar sistema”. A primeira etapa consiste em analisar o impacto que os arquivos contendo os problemas de acessibilidade e as recomendações de melhoria irão causar no jogo, desde a dinâmica de jogo até as regras impostas para cada desafio do Programmer. Também foi utilizado o GDD. Nessa etapa foi criado o documento contendo os tipos de alteração necessários dentro do jogo.

A segunda etapa consiste na criação do documento de mudanças propostas, utilizando como base o documento de tipos de alteração, para

compreender o que é possível realizar dentro do escopo do projeto e do tempo disponível.

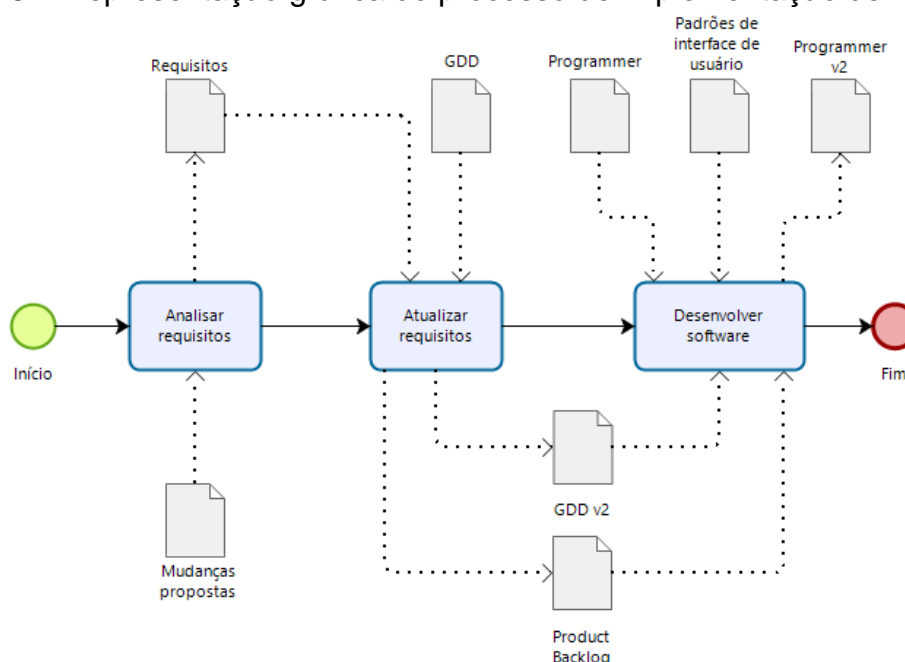
A terceira etapa consiste na implementação de mudanças, que é descrita detalhadamente na seção 4.2.4 Implementação de Mudança.

A última etapa do processo propunha a liberação do sistema para sua utilização. Essa atividade tem como entrada o *software* gerado neste trabalho.

4.2.4 Implementação de Mudança

O processo de implementação de mudança, como pode ser visto na Figura 5, tem como sua primeira etapa “Analisar requisitos”. Essa análise é feita com base nas mudanças propostas e, após esse processo, novos requisitos de software devem ser identificados para serem utilizados na etapa seguinte. Na segunda etapa, “Atualizar requisitos”, ocorre a atualização dos documentos criados anteriormente, como o GDD, para uma nova versão que contemple os requisitos de acessibilidades definidos antes, além de criar o Backlog do Produto, utilizado durante o desenvolvimento. Por fim, ocorre o desenvolvimento de *software* utilizando os requisitos definidos, além dos documentos de padrões de interface de usuário e do jogo Programmer que, em um primeiro momento teria, seu código-fonte reaproveitado.

Figura 5 – Representação gráfica do processo de implementação de mudança



Fonte: O próprio autor.

Propôs-se para esse desenvolvimento a adoção do método ágil Scrum de forma adaptada, com o envolvimento de apenas uma pessoa. A escolha desse processo se dá pelo fato de ser o mesmo processo utilizado por Silva (2019) para desenvolver o jogo Programmer. Segundo Sommerville (2011), métodos ágeis são baseados em desenvolvimento incremental, dessa forma, então, a transição da etapa de desenvolvimento para a evolução pós-entrega deveria ser imperceptível.

Na adaptação do Scrum, o primeiro aspecto altera o número de papéis dentro do processo. Não há um Scrum Master dentro do projeto, visto que não há um time de desenvolvedores para dividir este papel. A orientadora do trabalho assumiu o papel de Product Owner e o aluno, o de desenvolvedor. O desenvolvedor criou inicialmente o documento de Backlog do Produto, com a PO como responsável pela sua manutenção, classificação e avaliação. A duração dos *sprints* deveria variar de duas até três semanas, dependendo da tarefa definida. O Trello² foi utilizado para fazer o gerenciamento desse processo.

A utilização do catálogo de padrões de projeto de interface ocorre como um guia com exemplos práticos de possíveis alterações realizadas no Programmer. Ao se utilizar das diretrizes definidas por Cheiran (2013), a nova versão do catálogo de padrões especifica o que pode ser alterado em cada contexto de desenvolvimento onde os padrões são aplicados. É possível exemplificar o padrão de “Opções” onde é determinada a necessidade utilizar textos alternativos para imagens que são adicionadas nas opções de um menu. Outro ponto é a importância de possibilitar ao usuário mais de uma forma de apresentar informações relevantes para a configuração do jogo. Esses dois pontos eram negligenciados na primeira versão do jogo digital Programmer e foram alterados a partir da consulta do Backlog do Produto e do Catálogo de Padrões.

Após realizar uma investigação mais profunda na *engine* utilizada para desenvolver o jogo digital Programmer, onde foram consultadas a documentação apresentada no *website* da ferramenta e uma troca de mensagens de *e-mails* com os próprios desenvolvedores, contudo, foi constatado que ela não apresentava os recursos necessários para realizar a evolução de software completa contemplando requisitos de acessibilidade. Dessa forma, uma solução proposta para a etapa de

² Trello é um aplicativo de gerenciamento de projeto baseado na web desenvolvido em 2011. Ele opera um modelo de negócio gratuito e com opção de assinatura para recursos avançados (PRYOR, 2021).

desenvolvimento foi a criação de protótipos de interface de alta fidelidade, utilizando as tecnologias HTML5 e CSS. Esses protótipos, desenvolvidos para serem funcionais, servem como exemplo das possíveis melhorias a serem feitas no jogo digital Programmer, envolvendo a alteração de sua tecnologia de desenvolvimento, mas sem a necessidade de alterar sua plataforma de utilização.

4.3 Considerações Finais do Capítulo

A construção da metodologia deste trabalho foi embasada por diversas referências relacionadas aos temas abordados. Com isso, foi possível ter uma visão clara de como abordar todas as etapas de evolução propostas como objetivos do TCC.

É importante destacar que durante o desenvolvimento foi necessário ajustar a tecnologia na qual seriam desenvolvidas as melhorias do jogo digital Programmer por uma limitação em atender os requisitos de acessibilidade. Como esse impedimento foi identificado na fase final do trabalho pelo desenvolvedor, não foi possível implementar em sua totalidade todas as tarefas do Backlog do Produto.

A partir da geração dos protótipos, contudo, foi possível identificar como os principais requisitos seriam atendidos utilizando a nova abordagem. Dessa forma, o desenvolvimento dos protótipos de tela de alta fidelidade pode servir, de certa forma, como validação do catálogo de padrões de projeto de interface de usuário gerado neste trabalho, visto que este foi utilizado como guia para a implementação das soluções do Backlog do Produto.

5 RESULTADOS

Neste Capítulo, observando-se a mesma estrutura de apresentação da metodologia, são apresentados os resultados deste TCC.

5.1 Analisar Programmer

A primeira etapa da metodologia, Analisar o jogo digital Programmer, foi desenvolvida utilizando o *checklist* desenvolvido por Cheiran (2013), o trabalho de Silva (2019) e o próprio jogo Programmer. Como resultados obtidos, têm-se os artefatos de *Game Design Document* (GDD), descrito no Apêndice A, e o *checklist* com suas questões respondidas, descrito no Apêndice B, além de uma descrição sobre os principais problemas de acessibilidade.

O GDD produzido neste trabalho apresenta uma visão geral sobre alguns elementos que compõem o jogo. Além de apresentar as telas utilizadas pelo jogador para a interação, é possível também compreender como algumas mecânicas do jogo funcionam, qual é o seu principal objetivo, quais são suas regras e controles. Todos esses pontos foram utilizados para realizar a análise dos requisitos do jogo durante o processo de evolução.

O jogo Programmer apresenta diversos problemas relacionados à falta de acessibilidade. Esses problemas são apresentados, a seguir, associados aos princípios definidos pela WCAG 2.0 (W3C, 2008):

- **Percepção:** Entre os principais problemas de acessibilidades do jogo Programmer, está a ausência do *feedback* sonoro, ou seja, indicativos sonoros imediatos para as informações essenciais como eventos dentro do jogo, interação de elementos com o jogador e o progresso da tarefa atual. A estrutura visual do jogo é também um problema identificado pelo *checklist*. Não há a possibilidade de ajustar elementos como a resolução do jogo, a redução de detalhes, como por exemplo, diminuir a qualidade de texturas ou sombras. O tamanho do cursor também não é possível alterar. Após a realização do *checklist*, foi identificado que não há a possibilidade de configuração entre o som monoaural e estereofônico, que é utilizado para causar a impressão de posicionamento de fontes sonoras mais à esquerda ou mais à direita. O *feedback* visual também apresenta problemas no

Programmer, como a seleção das opções de resposta para os desafios do jogo que não é clara. A utilização apenas da cor para apresentar informação é outro elemento não recomendável em termos de acessibilidade que foi encontrado no jogo;

- **Operabilidade:** Outro ponto importante negligenciado é a opção de mapeamento e personalização dos controles do jogo. Atualmente, só é possível realizar os comandos do jogo pelo cursor, utilizando um *mouse* ou *touchpad*, o que limita muito as possibilidades para o jogador. Não há também a possibilidade de configurar a sensibilidade dos dispositivos de entrada, no caso do programa, o cursor do *mouse/touchpad*;
- **Compreensibilidade:** Além disso, seria interessante apresentar informações essenciais do jogo por outros meios além do visual, como meios sonoros. O jogo não apresenta linguagem simples, porém isso ocorre pelo tema apresentado pelos desafios. Dessa forma, apesar de o *checklist* ter apontando como uma questão que não atende ao critério, não é necessariamente um problema. A funcionalidade de definir foco em elementos do jogo também não está contemplada. Outro problema identificado com o *checklist* é a falta de um manual online acessível para o jogo. Um modo de treinamento guiado não está disponível para os jogadores, essa opção ajudaria os usuários a utilizarem as mecânicas do jogo com instruções de um passo a passo;
- **Robustez:** Após a revisão do *checklist* foi identificada a falta de um apoio coerente em relação aos recursos de acessibilidade de todos os elementos do jogo.

É importante, porém, pontuar que apesar de negligenciar pontos essenciais para contemplar a acessibilidade, o jogo Programmer contém alguns elementos que foram aceitos no *checklist*. Estes também são apresentados com base nos princípios definidos pela WCAG 2.0 (W3C, 2008):

- **Percepção:** É possível citar o critério de fonte e formatação legível que estão presentes no jogo, apresentando fontes com padrão claro e em tamanho grande;
- **Operabilidade:** O menu é simples e sem a necessidade de configurações desnecessárias. Além disso, em relação aos controles do jogo é possível

defini-los como simples, facilitando sua jogabilidade. Não há a necessidade de comandos complexos demais para a realização das atividades dentro do jogo. A possibilidade de início rápido do jogo é outro critério de acessibilidade atendido pelo jogo. A opção de pausar e repetir animações ou eventos que tenham informações essenciais para a progressão do jogo também estão disponíveis, como a possibilidade de ler o enunciado de um desafio iniciado mais de uma vez;

- **Compreensibilidade:** Não foram encontradas funcionalidades que contemplavam este princípio;
- **Robustez:** Para atender um dos requisitos da robustez é necessário que os elementos tenham coerência na sua usabilidade. Apesar da problemática do jogo de ser jogável apenas com o cursor, no quesito homogeneidade, que consiste em realizar todos os comandos com o mesmo controle, o jogo possibilita esse requisito.

Um ponto que não foi possível avaliar e que consta no *checklist* é a do salvamento e recuperação de opções, que possibilita que todas as configurações alteradas pelo jogador no comportamento padrão do jogo, como alteração de controles, recursos de fala, volume e outros, sejam recuperadas automaticamente quando o jogo for reaberto. O que impossibilitou essa avaliação foi a indisponibilidade de alteração desses recursos dentro do jogo. Todos os outros elementos do *checklist* foram classificados com a nota mais baixa em relação à dificuldade de testar, ou seja, foram fáceis de testar no jogo.

5.2 Catalogar padrões de projeto de interface de usuário

Para a criação do catálogo de padrões de projeto de interface de usuário (APÊNDICE C), foi necessário adaptar os padrões utilizados no trabalho de Silva (2019) com requisitos de acessibilidade identificados a partir do *checklist* de Cheiran (2013), seguindo as boas práticas identificadas de Kultsova (2015) e as diretrizes encontradas no trabalho de Cheiran (2013). Dessa forma, além das características previamente descritas nos padrões de projeto de interface de usuário, como Nome do Padrão, Aplicabilidade, Intenção e Consequências, uma nova característica visando solucionar problemas de acessibilidade foi introduzida.

Como alguns padrões são descritos de forma mais genérica, sua solução de acessibilidade precisou especificar algumas formas de aplicar esse padrão no desenvolvimento de jogos digitais. Dicas de Loading, um exemplo de padrão de projeto de interface de usuário pode ser visto abaixo:

- **Nome do Padrão:** Dicas de Loading;
- **Intenção:** Mostrar dicas ao jogador enquanto ele espera o carregamento do jogo terminar;
- **Motivação:** Em vários jogos é necessário que seja feito o carregamento do conteúdo do jogo para que o jogador possa começar a jogar. No entanto, nesse meio tempo o jogador não interage com o jogo de nenhuma forma. Para aliviar isso, durante as telas de carregamento o jogo pode dar dicas que irão ajudar o jogador quando ele for realmente jogar;
- **Aplicabilidade:** Esse padrão deve ser usado quando o jogador precisar esperar o carregamento de algum conteúdo sem que ele interaja com o jogo de nenhuma forma;
- **Consequências:** O jogador pode ganhar conhecimento das mecânicas do jogo enquanto espera. O jogador pode ficar entretido dependendo do conteúdo das dicas.
- **Acessibilidade:** Na tela de dicas é possível utilizar a recomendação da diretriz 1.4.4 de apresentação visual de textos possuindo um contraste mínimo de 4:5:1, exceto textos com tamanho “muito grande”. Também é importante verificar se os textos apresentados não são imagens, facilitando assim a utilização de leitores de tela, e a aplicação de texto alternativo para repassar informações.

Sendo assim, o catálogo visa apresentar as principais orientações para o desenvolvedor ter a capacidade aplicar os requisitos de acessibilidade quando utilizar determinado padrão de interface de usuário para construir um jogo digital. Neste trabalho, porém, esse catálogo será utilizado para conduzir a evolução do sistema contemplando requisitos de acessibilidade.

5.3 Evolução de Software

A etapa de evolução de software teve início com a atividade de “Analisar Impacto” utilizando os artefatos de gerados anteriormente como o GDD e também a identificação dos principais problemas de acessibilidade do jogo digital Programmer após a aplicação de uma ficha de avaliação de acessibilidade em jogos, desenvolvida por Cheiran (2013). Essa atividade teve como saída um documento intitulado “Tipos de Alterações” (APÊNDICE D), descrevendo as possíveis mudanças necessárias para tornar o jogo digital Programmer mais acessível e como elas poderiam afetar a estrutura do projeto. Como essas principais modificações visam reconfigurar interações com o jogador quando ocorre a transmissão de informações do jogo, não há necessidade de uma alteração estrutural no código, porém é necessário construir novas formas de realizar uma ação dentro do jogo para tornar a utilização do teclado pelo jogador possível.

Na segunda etapa, denominada “Planejar Versão”, ocorre a criação do documento “Mudanças Propostas” (APÊNDICE E), que define as alterações a serem realizadas no projeto. A análise para definir o que poderá ser feito foi realizada utilizando o documento de Tipos de Alterações, onde trata como essas mudanças poderiam afetar o estado atual do jogo. Outro critério utilizado também foi o tempo disponível para a conclusão do trabalho dentro do prazo de entrega do TCC.

5.4 Implementação de Mudança

O subprocesso de Implementação de Mudança é dividido em três partes, são elas: Analisar Requisitos, Atualizar Requisitos e Desenvolver Software. Na primeira parte, foi realizada uma análise do documento de “Mudanças Propostas” e, então, novos requisitos foram definidos. Essa análise funciona como uma forma de redigir as mudanças previamente definidas para um formato técnico, facilitando assim sua utilização na próxima parte do processo.

Na segunda parte, ocorre a etapa de Atualização de Requisitos, onde foi utilizada a primeira versão do Game Design Document e a descrição dos problemas de acessibilidade no documento de Mudanças Propostas para atualizar

o GDD e criar o Backlog do Produto. Este novo documento é definido pelos requisitos a serem implementados, sua prioridade (baixa, média ou alta), seu status (a fazer, fazendo ou feito) e o sprint em que foi desenvolvido. A orientadora do trabalho, junto com o aluno, definiram as prioridades de cada requisito. Esses requisitos foram descritos em forma de história de usuário. Dentro do contexto de metodologia ágil, esse documento é designado como sendo o Backlog do Produto, que serve de guia para o desenvolvedor. O Quadro 6, a seguir, apresenta um extrato do Backlog do Produto (APÊNDICE F).

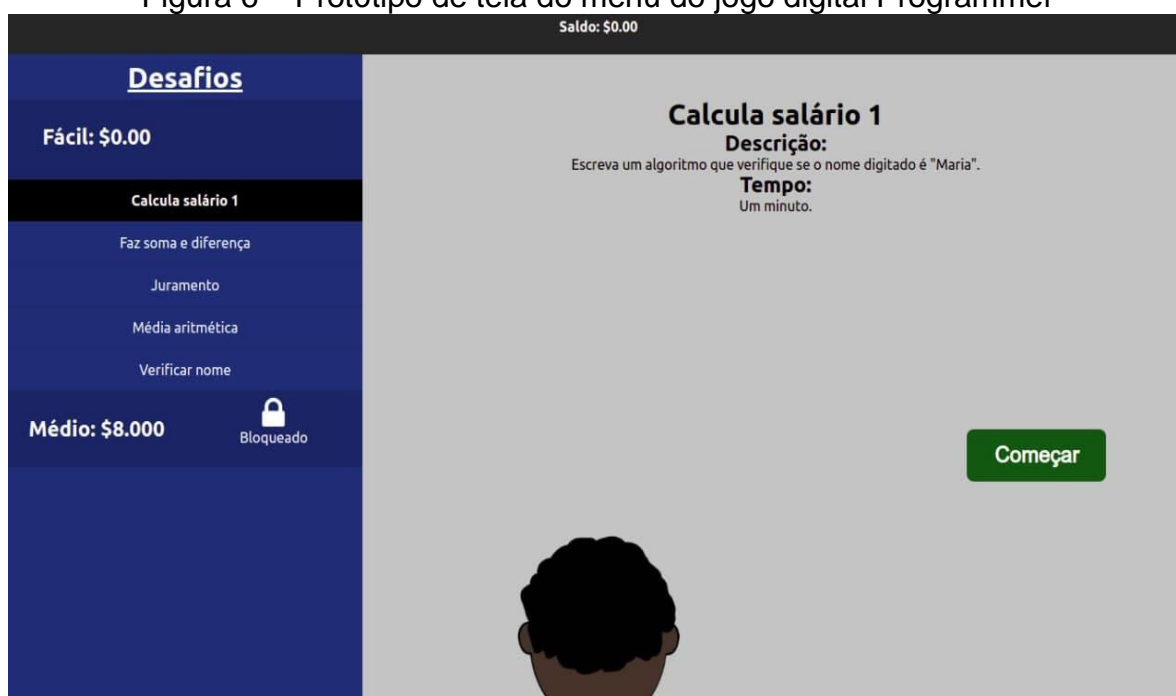
Quadro 6 – Backlog do Produto

ID	Requisito	Prioridade	Status	Sprint
1	Como jogador quero usá-lo pelo teclado e/ou pelo mouse.	alta	a fazer	1
2	Como jogador quero ativar ou desativar as animações do jogo.	alta	a fazer	2
3	O sistema deve apresentar informações utilizando texto além de cor.	alta	a fazer	1

Fonte: O próprio autor.

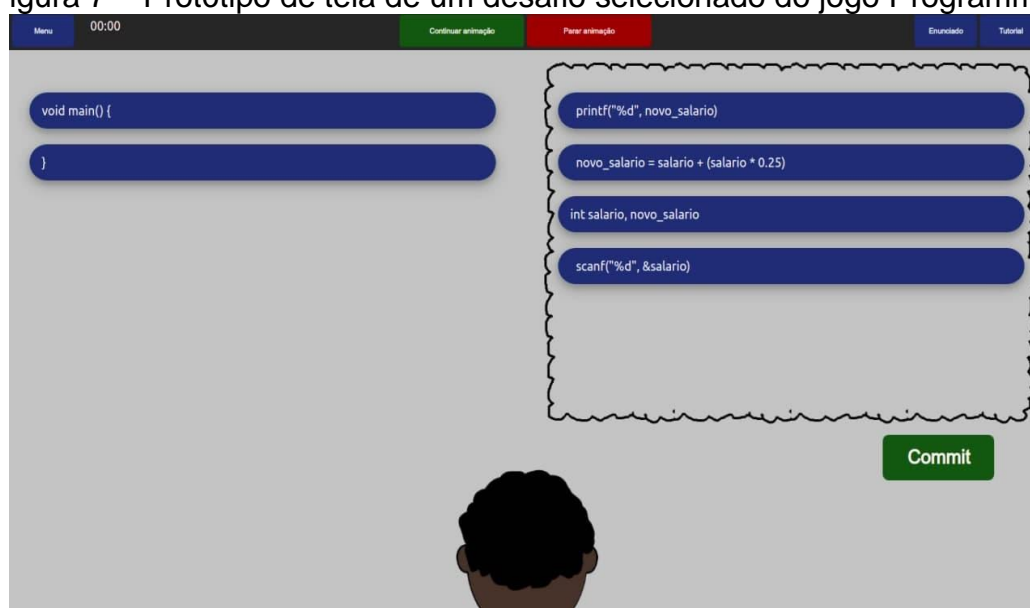
Na terceira e última etapa, ocorre o desenvolvimento de software. Nesta etapa, foram utilizados todos os documentos refinados gerados anteriormente. Desde o catálogo de padrões de projeto de interface, passando pela nova versão do GDD (APÊNDICE G), o Backlog do Produto e o próprio jogo digital Programmer. Durante a execução desta etapa foram identificados pelo desenvolvedor alguns gargalos técnicos e de estrutura para a conclusão de todas as tarefas. Como a *engine* utilizada não apresenta meios de se trabalhar a acessibilidade, a solução proposta foi a de alteração da linguagem de programação utilizada no desenvolvimento do jogo digital Programmer originalmente para o HTML5. Essa linguagem foi escolhida pois oferece um grande suporte à acessibilidade, como nas possíveis formas de se implementar as funcionalidades do jogo, até nas formas de avaliação do código-fonte conforme a WCAG 2.0 (W3C, 2008). Dessa forma, foram produzidos protótipos de tela que apresentam as duas principais interações do usuário com o jogo. Esses protótipos estão representados nas Figuras 6 e 7, a seguir.

Figura 6 – Protótipo de tela do menu do jogo digital Programmer



Fonte: O próprio autor.

Figura 7 – Protótipo de tela de um desafio selecionado do jogo Programmer



Fonte: O próprio autor.

Na construção desses protótipos desenvolvidos utilizando HTML5, foi possível contemplar alguns dos requisitos do Backlog do Produto, que não foram possíveis serem implementados utilizando a *engine* Cocos Creator. Alguns deles relacionados à interação entre usuário e sistema como os requisitos “Como jogador quero usar o jogo pelo teclado e/ou pelo mouse”, “Como jogador quero

ativar ou desativar as animações do jogo” e “Como jogador, ao navegar pelo teclado, quero saber minha localização na interface”. O requisito “O código-fonte deverá estar de acordo com os padrões de web de acessibilidade” também pode ser implementado por causa da alteração de tecnologia de desenvolvimento. Essa validação pode ser feita utilizando ferramentas disponíveis na web que apontam problemas no código-fonte a partir das diretrizes do W3C.

A implementação dessas funcionalidades foi realizada seguindo as orientações definidas no catálogo de padrões de interface de usuário. Como, por exemplo, a utilização do padrão Skeuomorphic, que define a importância de apresentar informações ao usuário sem afetar a suspensão de descrença do jogador ao acessar as interfaces. Como as soluções de acessibilidade na interface de menu não interferiam nos elementos que compõe o cenário de um ambiente de desenvolvimento, como a representação gráfica do desenvolvedor ou da nuvem de pensamentos com as soluções de um desafio, foi possível alterar a forma de apresentar dessas informações seguindo esse padrão. Outro padrão adotado foi o de Opções. Neste padrão é necessário apresentar opções de configurações, e com a introdução das diretrizes de acessibilidade, foi possível também manter as opções dispostas da forma original.

5.5 Considerações Finais do Capítulo

Como resultado deste trabalho, destacam-se alguns artefatos. A partir da compreensão dos problemas de acessibilidade do jogo digital Programmer e da utilização de um processo bem definido, foi elaborado seu Game Design Document. Este documento apresenta as principais funcionalidades do jogo, suas regras e objetivos.

Além disso, após um estudo sobre padrões de projeto de interface de usuário e utilizando a referência de Silva (2019), foi adaptado um catálogo de padrões para contemplar requisitos de acessibilidade. Utilizando esse catálogo como guia para soluções de implementação, foram desenvolvidos protótipos de interface de alta fidelidade do jogo digital Programmer de forma que estes validem o catálogo como ferramenta importante para a evolução de um jogo digital que contemple requisitos de acessibilidade.

Outro artefato relevante deste trabalho é a atualização do GDD para uma versão que contém os novos requisitos de acessibilidade do jogo digital Programmer, apresentandos no Backlog do Produto.

6. CONSIDERAÇÕES FINAIS

Este trabalho teve como proposta principal a evolução do jogo digital Programmer contemplando requisitos de acessibilidade utilizando padrões de projeto de interface. Para isso, foi necessária uma verificação geral do jogo para entender suas principais funcionalidades e avaliar as características de acessibilidade que apresentava.

Após a realização dessa análise, foi constatada a necessidade da criação de uma documentação mais robusta para tornar possível a evolução do jogo. A partir disso, então, foram criados o GDD, uma nova versão dos padrões de interface de usuário utilizados por Silva (2019) para desenvolver o jogo original, dessa vez, contemplando requisitos de acessibilidade. O *checklist* desenvolvido por Cheiran (2013) também foi aplicado e pôde esclarecer quais eram os principais pontos que tornavam o jogo não acessível para pessoas com alguma deficiência. O catálogo de padrões de interface de usuário utilizado por Silva (2019) foi adaptado para uma versão que contemplasse requisitos de acessibilidade em suas recomendações. Essa adaptação foi baseada nos estudos que definiram diretrizes sobre acessibilidade em jogos digitais de Cheiran (2013).

A especificação de processos claros facilitou o desenvolvimento e a criação dessa documentação, bem como a utilização de referências que dialogavam com o tema do trabalho. Assim sendo, esses resultados foram satisfatórios e contribuíram na criação do Backlog do Produto. Este documento foi construído após a análise do catálogo de padrões de interface de usuário, onde o desenvolvedor pode identificar quais eram as principais lacunas de acessibilidade que o jogo apresentava para propor alterações que mantinham o design e as funcionalidades do jogo dentro dos mesmos padrões utilizados por Silva (2019). O catálogo de padrões de interface de usuários também foi utilizado durante o desenvolvimento como guia para a aplicação das soluções de acessibilidade.

No processo de implementação de mudança, contudo, foi identificada a necessidade da alteração de tecnologia do jogo digital Programmer para a conclusão das tarefas definidas previamente. Principalmente porque a *engine* utilizada pelo desenvolvedor não possibilita o tratamento da acessibilidade durante o desenvolvimento das funcionalidades.

Essa *engine* foi escolhida, pois foi a mesma utilizada por Silva (2019) para desenvolver o jogo digital Programmer original. Foi possível, assim, compreender a importância de avaliar as tecnologias utilizadas para desenvolver o jogo, focando nas alterações que se propõe realizar, como um dos principais pilares do processo de evolução de software. Isso deve ser realizado de forma aprofundada, além da aplicação do checklist utilizado por este trabalho, que ajudou a entender, inicialmente, qual era o tratamento de acessibilidade no jogo mas não solucionou os problemas com a tecnologia utilizada.

É importante salientar também que os problemas encontrados no processo de evolução de software apresentam os desafios de contemplar a acessibilidade após o desenvolvimento do sistema ter sido concluído. Portanto, é necessário adotar as recomendações de acessibilidade para jogos digitais como referência a partir das atividades de análise e projeto.

Dessa forma, foram propostos novos protótipos de tela que permitissem realizar a evolução do jogo a partir de uma perspectiva acessível. As tecnologias HTML5 e CSS foram escolhidas pelo amplo conteúdo a cerca do assunto, possibilitando então ao desenvolvedor as ferramentas necessárias para colaborar ao objetivo principal do trabalho. Além disso, essas tecnologias são desenvolvidas para plataforma web, dessa forma, garantindo a manutenção da plataforma original do jogo.

Tendo em vista os artefatos gerados como resultados deste trabalho, como o catálogo de padrões de interface de usuário, validado com a implementação dos protótipos de interface de alta fidelidade e o GDD nas suas duas versões, é possível afirmar que os objetivos deste trabalho foram alcançados. Propõe-se como trabalhos futuros, então, a implementação do Backlog de Produto desenvolvido neste trabalho, adaptando o jogo a nova tecnologia proposta para o desenvolvimento das funcionalidades referidas no Backlog do Produto além das funcionalidades já existentes do Programmer, contemplando os requisitos de acessibilidade.

REFERÊNCIAS

ARAÚJO, Maria *et al.* **Mobile audio games accessibility evaluation for users who are blind.** International Conference on Universal Access in Human-Computer Interaction. p. 242–259. 2017.

BRASIL. Decreto-lei nº 6.949, de 25 de agosto de 2009. **Diário Oficial [da] República Federativa do Brasil**, Brasília, DF, 31 ago. 2009.

BRASIL. Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. **Comitê de Ajudas Técnicas.** Tecnologia Assistiva. – Brasília: CORDE. 138 p. 2009.

CHEIRAN, Jean Felipe Patikowski. **Jogos inclusivos: diretrizes de acessibilidade para jogos digitais.** 2013. Dissertação (Mestrado em Ciência da Computação). Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, 2013.

FOLMER, Eelke. **The Glossary of Human Computer Interaction.** 2015. Disponível em < <https://www.interaction-design.org/literature/book/the-glossary-of-human-computer-interaction> > Acesso em: out. 2019.

FOLMER, Eelke. **Designing Usable and Accessible Games with Interaction Design Patterns.** 2007. Disponível em <https://www.gamasutra.com/view/feature/129843/designing_usable_and_accessible_.php> Acesso em: out. 2019.

GRAMMENOS, Dimitris.; SAVIDIS, Anthony.; STEPHANIDIS, Constantine. **Designing universally accessible games.** Revista Computers in Entertainment (CIE) - SPECIAL ISSUE: Media Arts and Games. Nova Iorque. v. 7, p. 17–1 – 17. fev. 2009.

HAUGE, Jannicke Baalsrud *et al.* **Perspectives on accessibility in digital games.** International Conference on Entertainment Computing. v. 11112. p. 402–406. 2018.

HERSH, Marion; LEPORINI, Barbara. **Accessibility and Usability of Educational Gaming Environments for Disabled Students.** 12th IEEE International Conference on Advanced Learning Technologies, ICALT 2012. jul. 2012.

HERSH, Marion; LEPORINI, Barbara. **An overview of accessibility and usability of educational games.** p. 63–101. jan. 2013.

HIRA, Willian *et al.* **Criação de um modelo conceitual para Documentação de Game Design.** Simpósio Brasileiro de Games e Entretenimento Digital. 2016.

IGDA – International Game Developers Association. **Accessibility in Games: Motivations and Approaches.** 2004. Disponível em <

http://archives.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf > Acesso em: out. 2019.

IWARSSON, Susanne; STÅHL, Agnetha. Accessibility, usability and universal design—positioning and definition of concepts describing person-environment relationships. **Disability and rehabilitation**, v. 25, n. 2, p. 57-66, 2003.

KITCHENHAM, Barbara. **Procedures for performing systematic reviews**. Keele, Reino Unido, Keele University. v. 33. ago. 2004.

KOBAYASHI, Aline.; LETIZIO, Caroline.; TANAKA, Eduardo. **Relationship between accessibility and software evolution**. Pernambuco, Brasil, 5th Latin American Conference on Human-Computer Interaction, 2011. p. 298–302. out. 2011.

KULTSOVA, Marina *et al.* **A two-phase method of user interface adaptation for people with special needs**. Creativity in Intelligent Technologies and Data Science. Cham: Springer International Publishing, v. 754. p. 805–821. Ago. 2017.

LESSE. **Laboratory of Empirical Studies in Software Engineering**. Disponível em: < <http://lesse.com.br/tools/slr/>>, Acesso em 01 set. 2019.

LETIZIO, Caroline; TANAKA, Eduardo. **Evaluating Accessibility When Software Evolves**. Iadis International Conference Interfaces And Human Computer Interaction, v. , n. , p. 303 – 306. 2012.

NAÇÕES UNIDAS (ONU). **Standard Rules on the Equalization of Opportunities for Persons with Disabilities**. New York: UN, 1993.

OLIVEIRA, Romeu; MOURA, Ana; LEITE, Julio. **Reengineering for accessibility: A strategy based on software awareness**. Curitiba, Brasil, 17th Brazilian Symposium on Software Quality, 2018. p. 180–189. out. 2018.

PETRILLO, Fábio dos Santos. **Práticas Ágeis no Processo de Desenvolvimento de Jogos Eletrônicos**. 2008. Dissertação (Mestrado em Ciência da Computação). Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, 2008.

PRYOR, Michael. **Going Beyond The Board: A Whole New Trello Is Here**. 2021. Disponível em < <https://blog.trello.com/a-whole-new-trello> >. Acesso em 22 de abril de 2021.

SCHELL, Jesse. **The Art of Game Design: A Book of Lenses**. 2nd. ed. Natick, MA, USA: A. K. Peters, Ltd., 2014.

SILVA, Fernando; BRINCHER, Sandro. **Jogos digitais como ferramenta de ensino: reflexões iniciais**. outra travessia; Dossiê Especial V. I: Literaturas Digitais, Florianópolis, Brasil. v. 1, n. 2, p. 42–69, 2011.

SILVA, Pedro Henrique França. **Desenvolvendo um jogo com padrões de interface para ajudar no aprendizado de programação**. Orientador: Jean Felipe Patikowski Cheiran. 2019. 77 p. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Software) - Universidade Federal do Pampa, Curso de Engenharia de Software, Alegrete, 2019.

SOMMERVILLE, Ian. **Engenharia de software**. 9ª ed. São Paulo. PEARSON BRASIL, 2011.

SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum – O Guia Definitivo para o Scrum: As Regras do Jogo**. 2020. Disponível em < <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf> >. Acesso em: maio de 2021.

SZYKMAN, Alexandre; GOIS, João; BRANDÃO, André. **A perspective of games for people with physical disabilities**. Annual Meeting of the Australian Special Interest Group for Computer Human Interaction. Parkville, VIC, Australia, 2015. p. 274–283. dez. 2015.

TALARICO NETO, Americo *et al.* **Padrões de Interação – O Contexto WEB**. Departamento de Computação, Universidade Federal de São Carlos, São Carlos, Brasil. 2004.

TORRENTE, Javier *et al.* **Designing serious games for adult students with cognitive disabilities**. International Conference on Neural Information Processing. Doha, Qatar, 2012. p. 603–610. nov. 2012.

TORRENTE, Javier *et al.* **Evaluation of three accessible interfaces for educational point-and-click computer games**. Journal of Research and Practice in Information Technology, v. 45. ago. 2013.

WHITTON, Nicola. **Game engagement theory and adult learning**. Simulation & Gaming, v. 41. ago. 2010.

YUAN, Bei; FOLMER, Eelke. **Blind hero: Enabling guitar hero for the visually impaired**. 10th International ACM SIGACCESS Conference on Computers and Accessibility. Halifax, Nova Scotia, Canada, 2008. p. 169–176. jan. 2008.

APÊNDICES

APÊNDICE A – Game Design Document do jogo digital Programmer

PROGRAMMER

Game Design Document

Versão: 1.0

Autores:

Bruno Oliveira de Araújo

Alegrete,

Novembro de 2019

Índice

1. VISÃO GERAL	50
2. REGRAS DO JOGO	50
3. PERSONAGENS	51
4. CONTROLES	51
5. INTERFACE	51
6. BACKLOG DO PRODUTO	51

1. Visão Geral

Programmer é um jogo digital educacional para plataforma *web*, do gênero *puzzle*. O jogo é dividido em desafios, sendo que os desafios representam algoritmos que o usuário deve solucionar para resolver um problema proposto. O objetivo do jogador no desafio é montar um código que resolve um problema. Para isso, deverá escolher entre os blocos de códigos e montá-los na ordem correta. Isso deve ser realizado dentro de um tempo determinado.

1.1 Formas de pontuação

Respondendo corretamente os desafios o jogador receberá uma pontuação. Essa pontuação varia de acordo com o número de vezes que o jogador erra o desafio e com o tempo que demora para responder o enunciado.

1.2 Itens

O jogador receberá *feedback* de suas ações, além de receber dinheiro quando completar os desafios. Esse dinheiro poderá ser usado para desbloquear novos desafios com sua dificuldade elevada. O dinheiro é informado através de uma tela de pontuação que, além disso, dá uma nota ao desempenho do jogador.

1.3 Design de Níveis

Os níveis do jogo são definidos em fácil, médio e difícil. O jogador poderá, inicialmente, escolher entre os desafios definidos como fácil. Será possível avançar para o nível médio após o jogador resolver um número suficiente de desafios que o conceda dinheiro para avançar ao próximo nível.

2. Regras do Jogo

Existem apenas duas regras que delimitam as ações do jogador:

2.1 Tempo limite para resolver os desafios: O jogador tem um número delimitado de minutos para responder as questões propostas. Esse tempo é contado de forma crescente começando em zero segundo até dez minutos.

2.2 Soluções pré-definidas: O jogador já terá as soluções construídas previamente, logo só poderá organizar o código da forma correta, sem poder realizar alterações nas linhas de código.

3. Personagens

Os personagens existentes no jogo são o “Programador”, personagem controlado pelo jogador, e o “Chefe”, personagem que apresenta os enunciados dos desafios a serem realizados.

4. Controles

O jogador utiliza apenas o cursor para realizar as ações do jogo, utilizando como ferramenta o *mouse* ou o *touchpad* para isso. A navegação por teclado ainda não está habilitada. Os movimentos realizados são de selecionar os desafios antes de iniciá-los e arrastar as soluções corretamente, organizando as respostas.

5. Interface

As Figuras 1, 2, 3 e 4, a seguir, apresentam as principais telas do jogo. Na Figura 1, é apresentada a tela de *loading*, essa tela é apresentada durante o carregamento do jogo durante o início e após a seleção de um desafio para iniciar. A Figura 2 apresenta a tela de menu principal, onde é possível escolher entre os desafios do tipo fácil e selecionar a opção “começar” para iniciar a resolução do problema. Na Figura 3, o enunciado do desafio é apresentado. Na Figura 4, é apresentado a tela onde o jogador poderá solucionar o desafio.

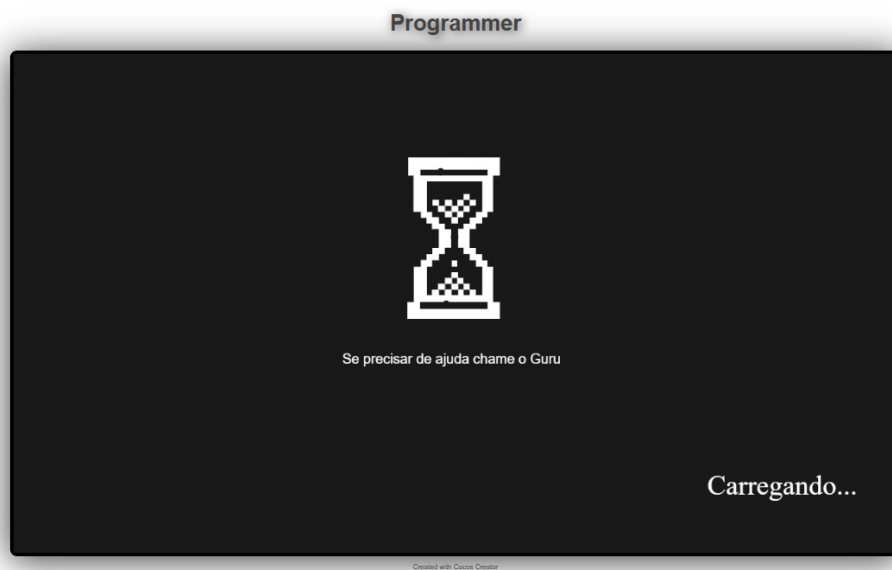


Figura 1 – Tela de *loading*
Fonte: O próprio autor.

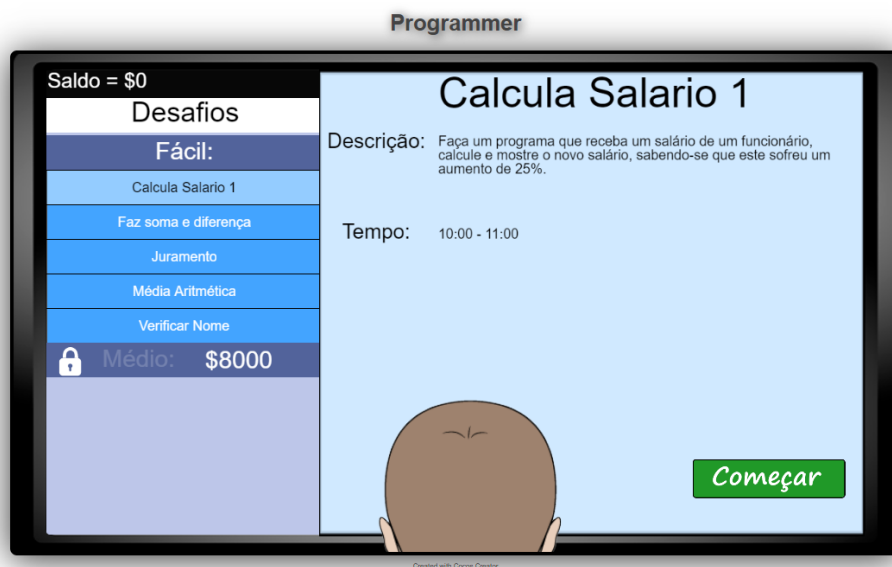


Figura 2 – Tela de menu principal
Fonte: O próprio autor.

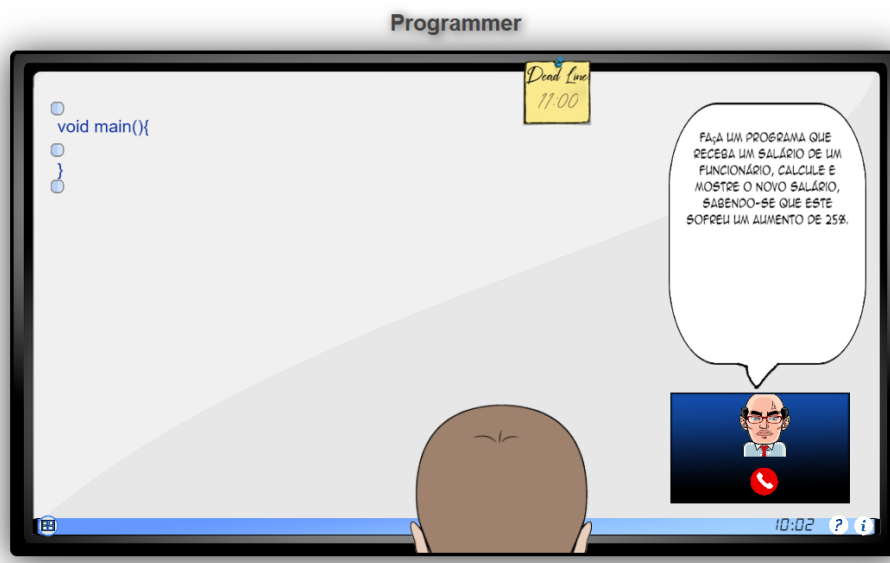


Figura 3 – Tela de apresentação do desafio
Fonte: O próprio autor.

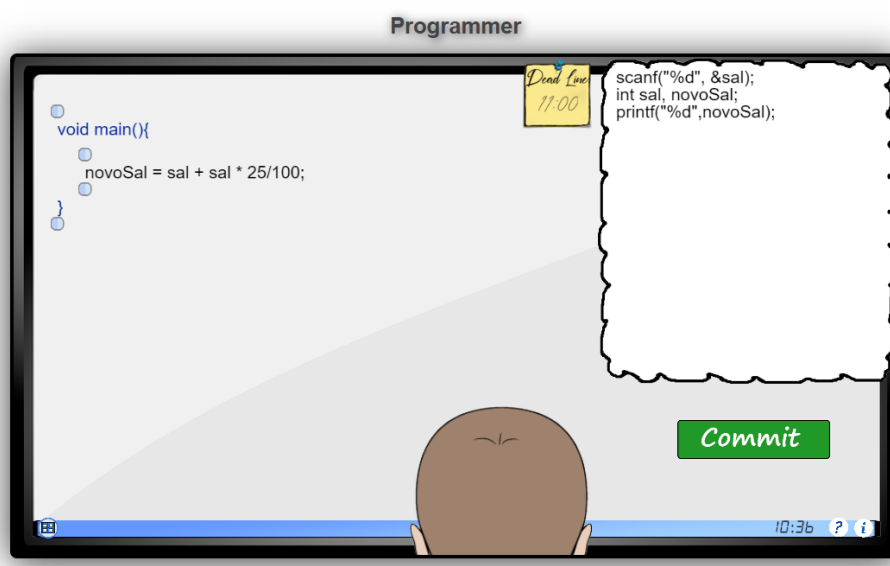


Figura 4 – Tela de solução do desafio
Fonte: O próprio autor.

**APÊNDICE B – Ficha de Avaliação Manual de Acessibilidade em Jogos (Cheiran
2013) preenchida**

Nome: Bruno Oliveira de Araújo

Jogo avaliado: Programmer

Recomendações de nível A

Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
1.2.1. Legendas (pré-gravadas)			x	5
1.3.1. Gráficos (básico)		x		1
1.3.2. Feedback (visual)		x		1
1.3.3. Som (mono/estéreo)		x		1
1.3.4. Feedback (sonoro)		x		1
1.4.1. Fonte e formatação legíveis	x			1
1.4.2. Uso de cor		x		1
Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
2.2.1. Pausar e repetir	x			1
2.4.1. Início rápido	x			1
2.5.1. Salvamente e recuperação de opções (global)			x	5
2.5.2. Homogeneidade	x			1
2.5.3. Personalização de controles (básico)		x		1
2.5.4. Controles simplificados	x			1
2.5.5. Sensibilidade		x		1
2.5.6. Interação por voz (básico)			x	1
Critérios	O jogo atende ao			Dificuldade de

	critério?			testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
3.1.1. Linguagem simples		x		1
3.2.1 Progressão de dificuldade (básico)	x			1
3.2.2. Foco		x		1
3.3.1. Ajuda (manual)	x			1
3.4.1. Características de acessibilidade		x		1
3.4.2. Manuais (online)		x		1
3.5.1 Ajude de dificuldade (básico)	x			1
3.5.2. Modo de treinamento (guiado)		x		1
Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível	
4.1.1. Suporte coerente		x		1

APÊNDICE C – Catálogo de Padrões de Interface de Usuário

Catálogo de Padrões de Projeto de Interface

Versão 1.0

Bruno Oliveira de Araújo

Os padrões de projeto de interface de usuário apresentados neste catálogo, selecionados a partir do trabalho de Silva (2019), foram adaptados para contemplar acessibilidade e servir como guia para o desenvolvimento da evolução do jogo digital Programmer.

1. Skeuomorphic

Nome do Padrão: Skeuomorphic;

Intenção: Mostrar interfaces ao jogador sem que elas atrapalhem na sua suspensão de descrença;

Motivação: Em jogos mais complexos, é necessário se criar interfaces extras para auxiliar o jogador ou para que ele possa realizar tarefas. Um dos grandes problemas quando se está projetando esse tipo de interface para um jogo é a necessidade de não afetar a suspensão de descrença do jogador ao acessar essas interfaces;

Aplicabilidade: Esse padrão deve ser usado quando o desenvolvedor precisar mostrar a um jogador uma interface muitas vezes durante o jogo e não quiser que isso interfira em sua suspensão de descrença;

Consequências: O design da interface pode ficar limitado por causa da intenção de manter a fidelidade ao mundo real; Aumenta a suspensão de descrença do jogador;

Acessibilidade: Utilizando as recomendações de acessibilidade, é possível realizar alguns ajustes para tornar o jogo mais acessível quando desenvolver esse padrão. Adicionar textos alternativos, como apresentado na diretriz 1.1.1 de Cheiran (2013) na representação dos objetos do jogo que passam informações importantes ao jogador, além de colocar nomes para suas funções. Isso também é resolvido a partir da diretriz 1.4.2 de não utilizar apenas cores para apresentar informações. Outra solução possível de aplicar a partir desse padrão nas interfaces construídas é definir os elementos listados de forma clara, como visto na diretriz 2.4.5 facilitando o entendimento do usuário do que é apresentado na tela e em ordem lógica para a identificação das opções de escolha do usuário.

2. Dicas de Loading

Nome do Padrão: Dicas de Loading;

Intenção: Mostrar dicas ao jogador enquanto ele espera o carregamento do jogo terminar;

Motivação: Em vários jogos é necessário que seja feito o carregamento do conteúdo do jogo para que o jogador possa começar a jogar. No entanto, nesse meio tempo o jogador não interage com o jogo de nenhuma forma. Para aliviar isso, durante as telas de carregamento o jogo pode dar dicas que irão ajudar o jogador quando ele for realmente jogar;

Aplicabilidade: Esse padrão deve ser usado quando o jogador precisar esperar o carregamento de algum conteúdo sem que ele interaja com o jogo de nenhuma forma;

Consequências:

- O jogador pode ganhar conhecimento das mecânicas do jogo enquanto espera;
- O jogador pode ficar entretido dependendo do conteúdo das dicas.

Acessibilidade: Na tela de dicas é possível utilizar a recomendação da diretriz 1.4.4 de apresentação visual de textos possuindo um contraste mínimo de 4:5:1, exceto textos com tamanho “muito grande”. Também é importante verificar se os textos apresentados não são imagens, facilitando assim a utilização de leitores de tela, e a aplicação de texto alternativo para repassar informações.

3. Opções

Nome do Padrão: Opções;

Intenção: Mostrar opções para o jogador ajustar o jogo da forma que mais o agradar;

Motivação: Um jogador pode ser qualquer pessoa, dessa forma jogadores tem preferências diferentes de como querem utilizar um jogo. Nesse sentido, é recomendado que jogos disponibilizem uma maneira de jogadores customizarem o jogo da forma que quiserem. Para isso, existe o padrão de interface de opções que disponibiliza um menu onde o jogador pode fazer todas as alterações possíveis de se realizar em um determinado jogo;

Aplicabilidade: Esse padrão deve ser usado quando o jogo apresentar algum tipo de configuração que o jogador possa mudar;

Consequências:

- O jogador poderá customizar o jogo para se adequar a suas preferências;
- O jogador saberá onde procurar quando quiser mudar alguma configuração do jogo.

Acessibilidade: Nesse padrão é possível utilizar a recomendação de utilizar textos alternativos, como visto na diretriz de 1.1.1 de Cheiran (2012) para elementos não-textuais, ou seja, se há ícones na tela é importante dar ao jogador uma outra forma de entender o seu significado. Também é necessário utilizar a boa prática de usar só uma forma de passar informação, seja ela visual, auditiva ou que contenha formas. Uma combinação desses elementos é o ideal, dentro do possível.

4. Tutorial

Nome do Padrão: Tutorial;

Intenção: Ajudar o jogador a entender as mecânicas do jogo;

Motivação: Quando um jogo tem muitas mecânicas, o jogador pode se sentir sobrecarregado pela quantidade de informação. Dessa forma, é recomendado que o jogo disponibilize uma forma do jogador se familiarizar com essas mecânicas;

Aplicabilidade: Aplicar quando houver muitas mecânicas em um jogo ou se alguma mecânica for complexa;

Consequências:

- O jogador terá acesso a explicação das mecânicas.
- Jogadores mais impacientes podem se frustrar com o tutorial.

Acessibilidade: Para trabalhar a acessibilidade nesse padrão é necessário saber como esse tutorial foi realizado. Caso o tutorial seja realizado em forma de vídeo demonstrando como o jogo funciona, no menu do tutorial seria possível implementar a recomendação de acessibilidade de não utilizar apenas cores ou formas para apresentar informações, baseando-se na diretriz 1.4.2 de Cheiran (2013). Além disso a utilização de legendas é imprescindível para deixar o vídeo mais acessível, como apresenta a diretriz 1.2.1. A aplicação de legendas nesse tipo de vídeo também é fundamental para melhorar a acessibilidade à essas informações. Caso o tutorial seja realizado de forma apenas textual, é importante permitir ao usuário redimensionar as fontes do texto (1.4.7), além da opção de poder alterar a cor da fonte do texto e também do fundo do texto (1.4.5).

5. Seleção de Fases

Nome do Padrão: Seleção de Fases;

Intenção: Disponibilizar para o jogador a possibilidade de selecionar o nível que quer jogar;

Motivação: Muitos jogos se organizam através de etapas que o jogador tem de cumprir, ou seja, através de níveis. Com isso, é necessário que o jogador possa escolher qual nível ele quer realizar. Para isso, é recomendado que o jogo exiba esses níveis de forma organizada para o jogador;

Aplicabilidade: Quando o jogo é dividido em níveis que o jogador tem de completar;

Consequências:

- O jogador poderá escolher qual nível irá fazer.
- Dependendo da forma com que os níveis foram organizados, o jogador poderá saber sobre a dificuldade dos níveis, temas e etc.

Acessibilidade: A opção de navegação por teclado é um requisito de acessibilidade que pode ser aplicado nesse padrão, como prevê a diretriz 2.1.1. Além disso, tornar a seleção de fases o mais simples possível, com as opções de dificuldade claras tanto em áudio como visualmente é necessário.

6. Tela de Pontuação

Nome do Padrão: Tela de Pontuação;

Intenção: Mostrar para o jogador seu desempenho ao jogar o jogo;

Motivação: Uma boa forma de se incentivar o jogador a continuar jogando o jogo, é dar pontuação ao seu desempenho, dessa forma ele se sente motivado a melhorar. Nesse sentido, surge a necessidade de se criar uma tela que mostre esse resultado para o jogador;

Aplicabilidade: O jogo tem algum tipo de medição de desempenho do jogador e é necessário mostrar esse isso a ele;

Consequências:

- O jogador poderá se sentir motivado a melhorar no jogo.
- O jogador pode se sentir frustrado ao não receber uma boa pontuação.

Acessibilidade: Na tela de apresentação da pontuação do jogador é possível utilizar a recomendação de acessibilidade da diretriz 1.4.4 de apresentação visual de textos possuindo um contraste mínimo de 4:5:1, exceto textos com tamanho

“muito grande”. Também é importante verificar se os textos apresentados não são imagens, facilitando assim a utilização de leitores de tela, e a aplicação de texto alternativo para repassar essas informações.

APÊNDICE D – Documento de Tipos de Alteração

Documento de Tipos de Alterações a serem feitas no jogo digital Programmer

Autor: Bruno Oliveira de Araújo

Versão: 1.0

Sendo as possíveis alterações identificadas para aplicar requisitos de acessibilidade ao jogo digital Programmer, alterações de feedback e de apresentação de informações utilizando diversos meios, uma modificação na estrutura do projeto não se faz necessária. Não há também a necessidade de utilização de um banco de dados para essas possíveis alterações. Logo, a interface será a parte mais afetada durante a evolução do sistema. Será necessário também adicionar comandos dentro do código-fonte para o Programmer ser acessado via teclado.

APÊNDICE E – Documento de Mudanças Propostas

Documento de Mudanças Propostas do jogo digital Programmer

Autor: Bruno Oliveira de Araújo

Versão: 2.0

Após a criação do Game Design Document do jogo digital Programmer e a aplicação de um checklist de acessibilidade (Cheiran, 2013) foi possível compreender melhor suas funcionalidades e necessidades de aprimoramento. Dessa forma, algumas mudanças são propostas neste documento.

O jogo digital Programmer não apresenta feedback sonoro das principais ações dentro do jogo. Essa questão foi identificada após a aplicação do checklist onde um de seus critérios é que um jogo possa realizar essa ação. Sendo assim, esse aspecto do jogo necessita de uma alteração para cumprir com esse requisito.

O feedback visual também apresenta erros, não definindo uma resposta visualmente boa para o estado atual do jogo, como a seleção de opções de resposta para os desafios da partida. Além disso, elementos de interação não apresentam todas as informações de forma clara, dificultando o entendimento do usuário. A configuração do tamanho dos textos e das imagens do jogo também não está disponível. Ainda sobre o feedback visual, não há a possibilidade de desativar as animações e efeitos de física.

A opção de configurar um tema de alto contraste dentro do jogo também não está disponível. Apesar de ser um problema não identificado pelo checklist, isso consta nas diretrizes de Cheiran (2013). Essa opção apresentaria elementos interativos relevantes e texto do jogo num contraste de 7:1.

Outro ponto identificado pelo checklist de acessibilidade é a possibilidade de mapeamento e personalização dos controles do jogo. O Programmer não apresenta essa opção. Logo, isso se torna uma funcionalidade a ser alterada.

Para aumentar o alcance do jogo, é importante para os jogadores que as informações possam ser apresentadas por mais de um meio. O que atualmente é representado apenas por texto, pode ser apresentado utilizando meios sonoros.

Como forma de melhorar o entendimento das funcionalidades, apresentar uma opção de menu de forma clara e robusta, ou seja, não utilizando apenas cores ou símbolos é essencial para pessoas com deficiência visual.

O checklist também identificou a falta de uma manual online para a utilização do jogo, como também a falta de um modo de treinamento guiado, no qual o jogador poderá praticar livremente as mecânicas do jogo com suas devidas instruções.

APÊNDICE F – Backlog do Produto

Backlog do Produto				
ID	Requisito	Prioridade	Status	Sprint
1	Como jogador quero usá-lo pelo teclado e/ou pelo mouse.	Alta	a fazer	1
2	Como jogador quero configurar a apresentação de feedback sonoro do jogo.	Alta	a fazer	n/a
3	Como jogador quero configurar a opção de alto contraste do jogo.	Média	a fazer	n/a
4	Como jogador quero configurar o tamanho das fontes e imagens do jogo.	Média	a fazer	n/a
5	Como jogador quero ativar ou desativar as animações do jogo.	Alta	a fazer	2
6	Como jogador quero configurar o áudio para mono ou estéreo.	Média	a fazer	n/a
7	Como jogador quero ajustar a velocidade e aceleração do cursor dentro do jogo.	Média	a fazer	n/a
8	O sistema deve apresentar informações utilizando texto e cor.	Alta	a fazer	1
9	Como jogador quero acessar um manual online do jogo.	Baixa	a fazer	n/a
10	Como jogador quero acessar um modo de treinamento guiado.	Baixa	a fazer	n/a
11	Como jogador quero configurar o cronometro do jogo.	Alta	a fazer	2
12	Como jogador quero saber a localização do cursor.	Média	a fazer	3
13	A estrutura e os elementos da interface devem estar perceptíveis, compreensíveis e operáveis para os usuários de leitores de tela.	Alta	a fazer	3
14	O código-fonte deverá estar de acordo com os padrões web de acessibilidade.	Alta	a fazer	3
15	Desafios, menus e funcionalidades devem funcionar de forma previsível.	Alta	a fazer	3
16	Forneça aos jogadores tempo suficiente para ler, entender e usar conteúdos e funcionalidades.	Alta	a fazer	1

Quadro 1 – Backlog do Produto

Fonte: O próprio autor.

APÊNDICE G – Game Design Document do jogo digital Programmer

PROGRAMMER

Game Design Document

Versão: 2.0

Autores:

Bruno Oliveira de Araújo

Alegrete,
Abril de 2021

Índice

1. VISÃO GERAL	50
2. REGRAS DO JOGO	50
3. PERSONAGENS	51
4. CONTROLES	51
5. INTERFACE	51
6. ACESSIBILIDADE	52

1. Visão Geral

Programmer é um jogo digital educacional para plataforma *web*, do gênero *puzzle*. O jogo é dividido em desafios, sendo que os desafios representam algoritmos que o usuário deve solucionar para resolver um problema proposto. O objetivo do jogador no desafio é montar um código que resolve um problema. Para isso, deverá escolher entre os blocos de códigos e montá-los na ordem correta. Isso deve ser realizado dentro de um tempo determinado.

1.1 Formas de pontuação

Respondendo corretamente os desafios o jogador receberá uma pontuação. Essa pontuação varia de acordo com o número de vezes que o jogador erra o desafio e com o tempo que demora para responder o enunciado.

1.2 Itens

O jogador receberá *feedback* de suas ações, além de receber dinheiro quando completar os desafios. Esse dinheiro poderá ser usado para desbloquear novos desafios com sua dificuldade elevada. O dinheiro é informado através de uma tela de pontuação que, além disso, dá uma nota ao desempenho do jogador.

1.3 Design de Níveis

Os níveis do jogo são definidos em fácil, médio e difícil. O jogador poderá, inicialmente, escolher entre os desafios definidos como fácil. Será possível avançar para o nível médio após o jogador resolver um número suficiente de desafios que o conceda dinheiro para avançar ao próximo nível.

2. Regras do Jogo

Existem apenas duas regras que delimitam as ações do jogador:

2.1 Tempo limite para resolver os desafios: O jogador tem um número delimitado de minutos para responder as questões propostas. Esse tempo é contado de forma crescente começando em zero segundo até dez minutos.

2.2 Soluções pré-definidas: O jogador já terá as soluções construídas previamente, logo só poderá organizar o código da forma correta, sem poder realizar alterações nas linhas de código.

3. Personagens

Os personagens existentes no jogo são o “Programador”, personagem controlado pelo jogador, e o “Chefe”, personagem que apresenta os enunciados dos desafios a serem realizados.

4. Controles

O jogador utiliza apenas o cursor para realizar as ações do jogo, utilizando como ferramenta o *mouse* ou o *touchpad* para isso. A navegação por teclado ainda não está habilitada. Os movimentos realizados são de selecionar os desafios antes de iniciá-los e arrastar as soluções corretamente, organizando as respostas.

5. Interface

As Figuras 1, 2, 3 e 4, a seguir, apresentam as principais telas do jogo. Na Figura 1, é apresentada a tela de *loading*, essa tela é apresentada durante o carregamento do jogo durante o início e após a seleção de um desafio para iniciar. A Figura 2 apresenta a tela de menu principal, onde é possível escolher entre os desafios do tipo fácil e selecionar a opção “começar” para iniciar a resolução do problema. Na Figura 3, o enunciado do desafio é apresentado. Na Figura 4, é apresentado a tela onde o jogador poderá solucionar o desafio.

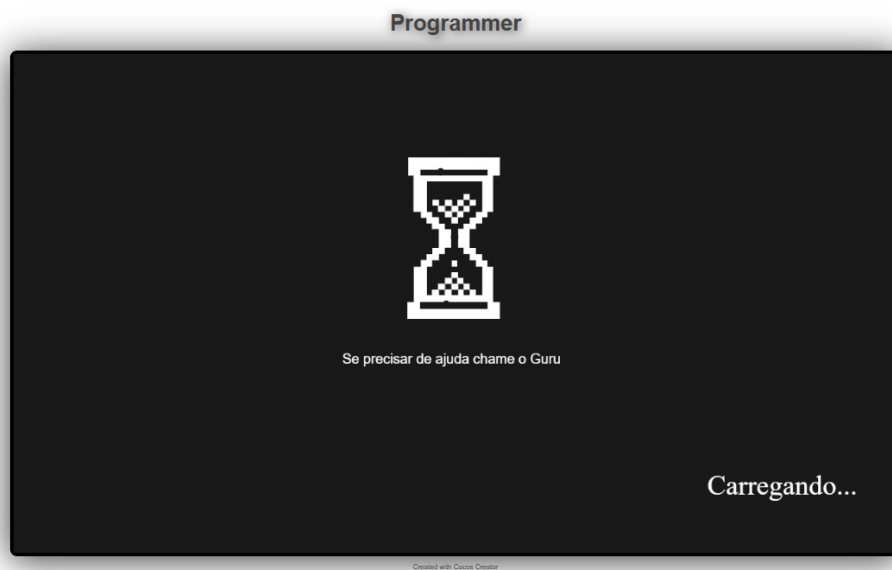


Figura 1 – Tela de *loading*
Fonte: O próprio autor.

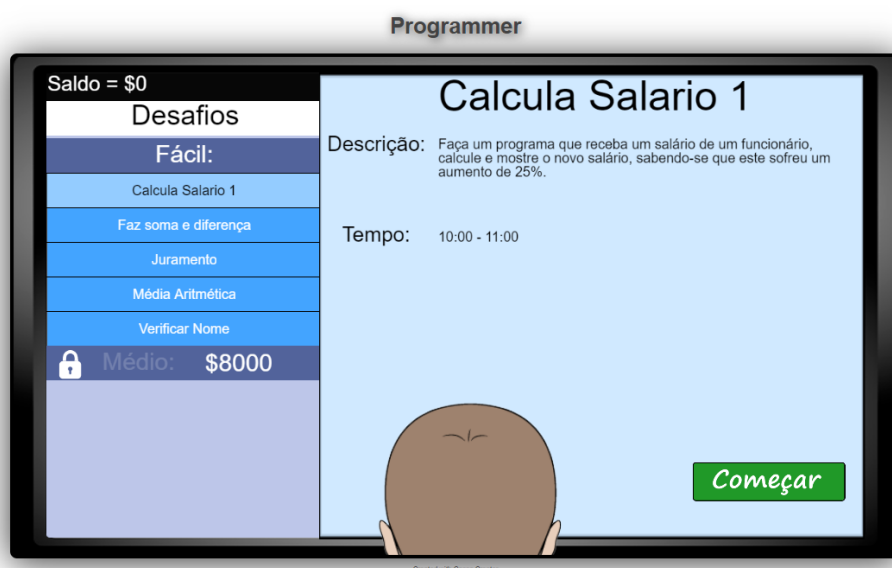


Figura 2 – Tela de menu principal
Fonte: O próprio autor.

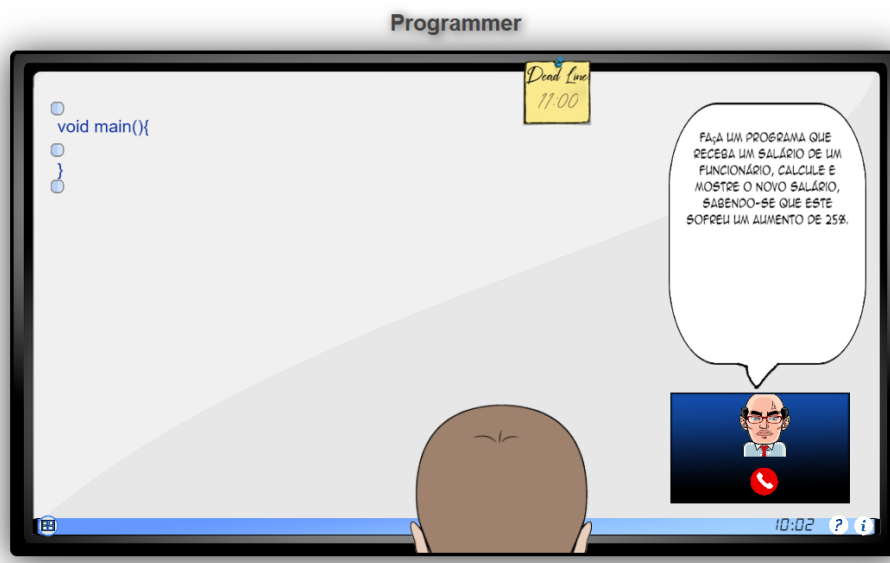


Figura 3 – Tela de apresentação do desafio
Fonte: O próprio autor.

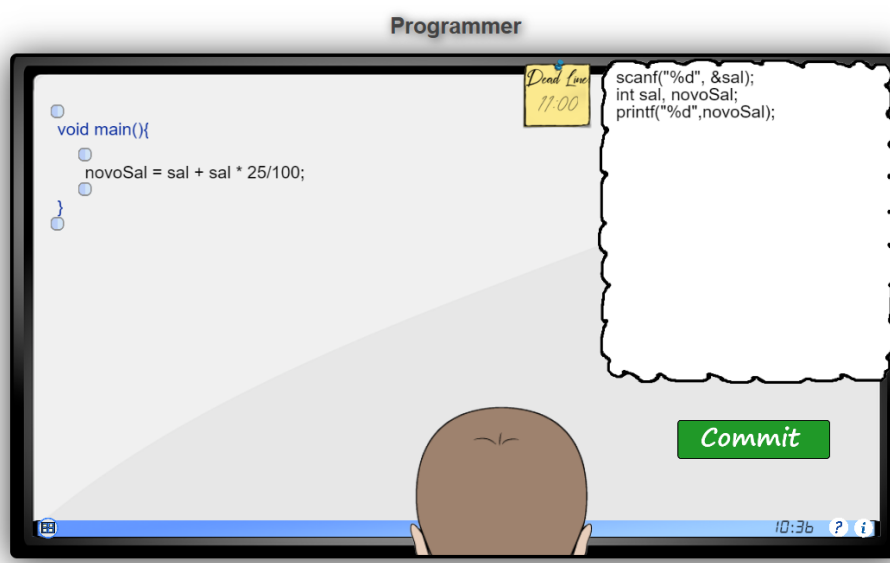


Figura 4 – Tela de solução do desafio
Fonte: O próprio autor.

6. Acessibilidade

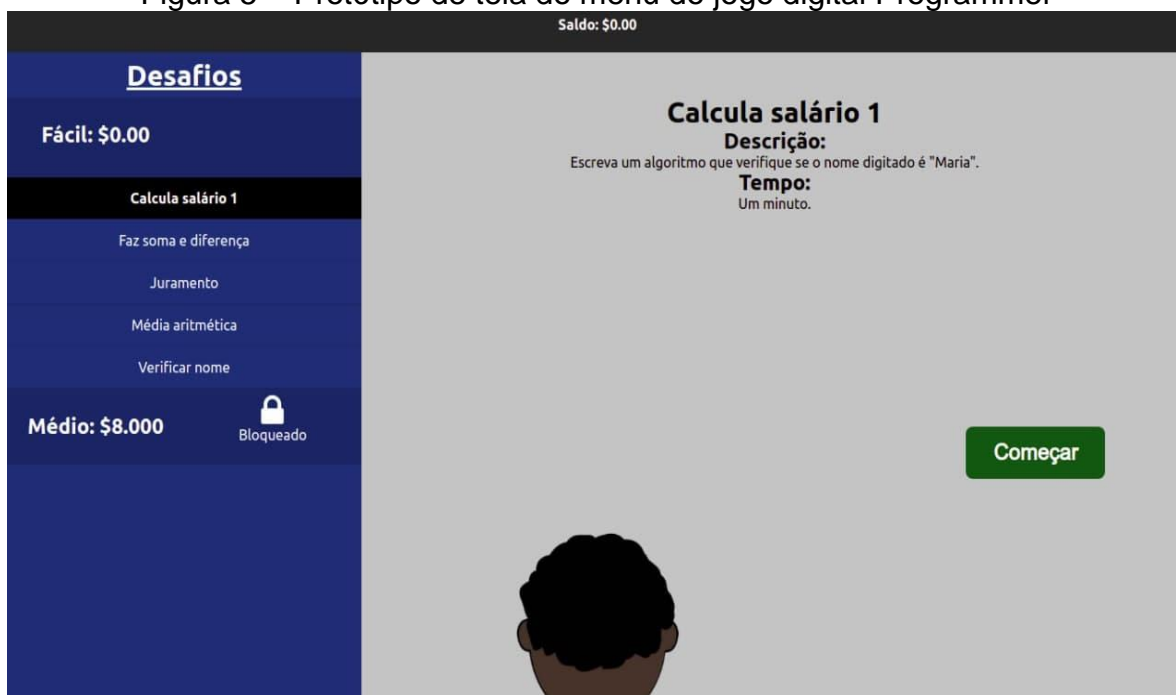
Para contemplar requisitos de acessibilidade, é necessário adicionar novas funcionalidades ao jogo digital Programmer. Essas funcionalidades são apresentadas a seguir:

- Como jogador quero usar o jogo pelo teclado e/ou pelo mouse.

- Como jogador quero ativar ou desativar as animações do jogo.
- Como jogador quero configurar o cronômetro do jogo.
- Como jogador, ao navegar com o teclado, quero saber minha localização na interface.
- Como jogador quero acessar um manual online do jogo.
- Como jogador quero acessar um modo de treinamento guiado.
- Como jogador quero configurar a opção de alto contraste do jogo.
- Como jogador quero configurar o tamanho das fontes e imagens do jogo.
- Como jogador quero ajustar a velocidade e aceleração do cursor dentro do jogo.
- Como jogador quero configurar a apresentação de feedback sonoro do jogo.

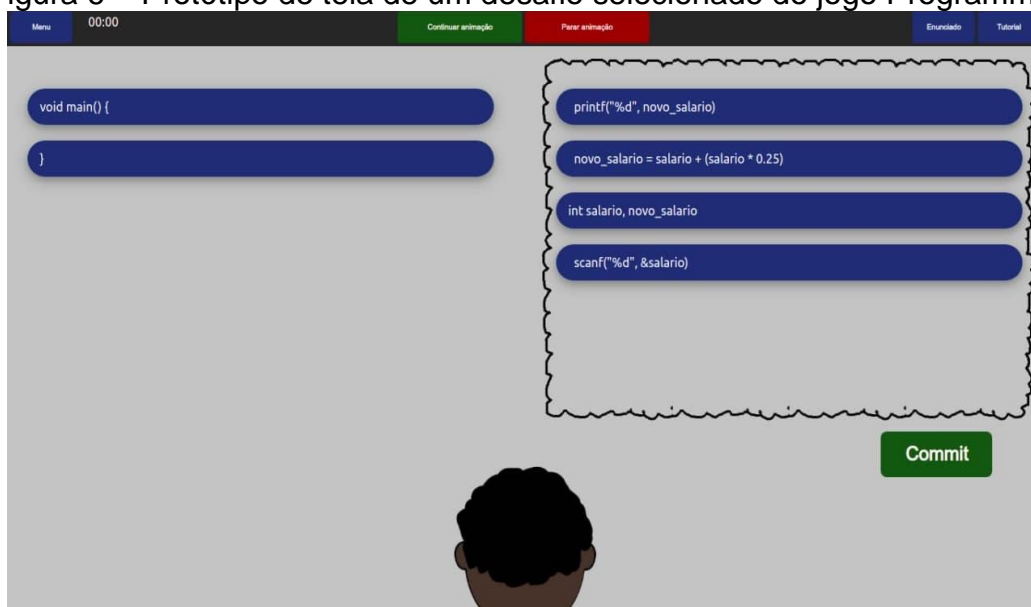
Como exemplo de uma possível implementação dessas novas funcionalidades foram desenvolvidos protótipos de tela que contemplem os requisitos de acessibilidade, utilizando um catálogo de padrões de interface de usuário que contemple requisitos de acessibilidade. Esses protótipos podem ser vistos nas Figuras 5 e 6 a seguir:

Figura 5 – Protótipo de tela do menu do jogo digital Programmer



Fonte: O próprio autor.

Figura 6 – Protótipo de tela de um desafio selecionado do jogo Programmer



Fonte: O próprio autor.

ANEXOS

ANEXO A <FICHA DE AVALIAÇÃO MANUAL DE ACESSIBILIDADE EM JOGOS – CHEIRAN 2013>

Ficha de Avaliação Manual de Acessibilidade em Jogos – Cheiran 2013

Nome:

Jogo avaliado:

Recomendações de nível A

Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
1.2.1. Legendas (pré-gravadas)				
1.3.1. Gráficos (básico)				
1.3.2. Feedback (visual)				
1.3.3. Som (mono/estéreo)				
1.3.4. Feedback (sonoro)				
1.4.1. Fonte e formatação legíveis				
1.4.2. Uso de cor				
Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
2.2.1. Pausar e repetir				
2.4.1. Início rápido				
2.5.1. Salvamente e recuperação de opções (global)				
2.5.2. Homogeneidade				
2.5.3. Personalização de controles (básico)				
2.5.4. Controles simplificados				
2.5.5. Sensibilidade				

2.5.6. Interação por voz (básico)				
Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível avaliar	
3.1.1. Linguagem simples				
3.2.1 Progressão de dificuldade (básico)				
3.2.2. Foco				
3.3.1. Ajuda (manual)				
3.4.1. Características de acessibilidade				
3.4.2. Manuais (online)				
3.5.1 Ajude de dificuldade (básico)				
3.5.2. Modo de treinamento (guiado)				
Critérios	O jogo atende ao critério?			Dificuldade de testar (valor de 1 a 5)
	Sim	Não	Não é possível	
4.1.1. Suporte coerente				

ANEXO B <DIRETRIZES DE ACESSIBILIDADE PARA JOGOS CHEIRAN 2013 – VERSÃO FINAL>

Resumo

As Diretrizes de Acessibilidade para Jogos cobrem um conjunto de recomendações para tornar jogos digitais mais acessíveis. Essas diretrizes se concentram em beneficiar pessoas com deficiências (incluindo pessoas com deficiências visuais, auditivas, motoras e mentais) e foram compiladas a partir de outros conjuntos de diretrizes similares e disponíveis na área.

Esse documento busca atender às necessidades de desenvolvedores e avaliadores.

Introdução

As Diretrizes de Acessibilidade para Jogos cobrem um conjunto de recomendações para tornar jogos digitais mais acessíveis e beneficiar pessoas com deficiências (incluindo pessoas com deficiências visuais, auditivas, motoras e mentais). Adicionalmente, o cumprimento dessas recomendações amplia a qualidade de uso para pessoas com limitações temporárias (como lesões em membros), pessoas com transtornos de desenvolvimento (como síndromes do espectro autista), pessoas com limitações tecnológicas (como problemas em dispositivos de interação) e pessoas idosas.

As DAJ foram compiladas na estrutura WCAG 2.0 a partir dos seguintes trabalhos (além dos conteúdos da própria WCAG 2.0):

- Accessibility in Games: Motivations and Approaches de IGDA
- Guidelines for the development of entertaining software for people with multiple learning disabilities de UPS Project
- Guidelines for developing accessible games de Roland Ossmann
- Game Accessibility Guidelines
- Game Accessibility Top Ten de IGDA GASIG

- Blind Computer Games: guidelines for building blind-accessible computer games de J. Bannick

Camadas de Orientação nas Diretrizes de Acessibilidade para Jogos

De forma a orientar diferentes públicos em relação à acessibilidade em jogos, diversas camadas de recomendações foram adotadas: princípios globais, diretrizes gerais e critérios de sucesso testáveis.

- **Princípios:** mantendo a mesma estrutura das diretrizes de acessibilidade web WCAG 2.0, os quatro princípios fundamentais de acessibilidade são mantidos: perceptível, operável, compreensível e robusto.
- **Diretrizes:** a coleção de 12 diretrizes originais na estrutura do documento WCAG 2.0 foram estendidas para contemplar aspectos diferenciados em relação a interação e plataformas dos jogos digitais. Essas diretrizes ainda apresentam metas básicas pra implantação de acessibilidade, mas não englobam testes já que formam apenas uma estrutura geral para compreensão dos critérios testáveis.
- **CrITÉrios de sucesso:** para cada diretriz, foram mantidos critérios testáveis que indicam como verificar se uma diretriz é atendida e qual é o nível de conformidade com essa diretriz. Há três níveis de conformidades que foram elencados segundo o esforço e a dificuldade técnica de implantar aquela diretriz: A (mais baixo), AA e AAA (mais alto). Detalhes adicionais são apresentados na seção Conformidade.

Diretrizes

Princípio 1: Perceptível - A informação e os componentes de interface devem ser apresentados aos jogadores de forma que possam percebê-los.

Diretriz 1.1. Textos alternativos: forneça alternativas textuais para todo o conteúdo não textual de forma que possa ser mudado para as diferentes formas que as pessoas necessitam como fala ou símbolos.

1.1.1. Conteúdo não textual: todo o conteúdo não textual com informação essencial do jogo deve possuir texto alternativo que fornece significado

equivalente. Esse critério inclui modelos e imagens de personagem e objetos interativos, mas exclui controles seletores do jogador (como botões e barras deslizantes que são cobertos pela diretriz 4.1) e elementos que sejam puramente decorativos ou relacionados à formatação. (Nível AA)

Diretriz 1.2. Mídias temporais: forneça alternativas para mídias temporais.

1.2.1. Legendas (pré-gravada): legendas para todo o conteúdo falado (incluindo falas e narração em vídeos, reprodução de gravações e falas de personagens dentro do jogo) devem poder ser habilitadas/desabilitadas. (Nível A)

1.2.2. Legendas descritivas (pré-gravada): legendas para todo o conteúdo de áudio (incluindo falas, narrações, efeitos sonoros e descrição de tipo de música em animações de vídeos e dentro do jogo) devem poder ser habilitadas/desabilitadas. (Nível AA)

1.2.3. Legendas (ao vivo): legendas para todo o conteúdo falado ao vivo (incluindo narração de partidas e comentários de partidas ao vivo que sejam transmitidas diretamente no jogo) devem poder ser habilitadas/desabilitadas. (Nível AAA)

1.2.4. Audiodescrição (pré-gravada): audiodescrição para todas as animações (incluindo vídeos e sequências de animação de eventos dentro do jogo) deve poder ser habilitada/desabilitada (audiodescrição dos demais elementos de dentro do jogo é coberta pela diretriz 1.3). (Nível AAA)

1.2.5. Língua de sinais (pré-gravada): interpretação em língua de sinais para todo o conteúdo de áudio falado (incluindo narração em vídeos e diálogos dentro do jogo) deve poder ser habilitada/desabilitada. (Nível AAA)

Diretriz 1.3. Adaptável: crie conteúdos e interfaces com o jogador que possam ser apresentados de formas diferentes (por exemplo, em resolução menor ou apenas em áudio) sem perder informação essencial.

1.3.1. Gráficos (básico): a estrutura visual do jogo deve poder ser ajustada em todos os seguintes elementos: (Nível A)

- Ponteiros e marcas: o tamanho do indicador do dispositivo apontador (como seta ou mira) deve poder ser alterado, mesmo que seja consequência da alteração de resolução;
- Resolução: a resolução do jogo (como um todo) deve poder ser alterada;
- Redução de detalhes: a quantidade ou o nível de detalhes para os principais elementos do jogo deve poder ser alterada individualmente ou como um todo (podendo incluir qualidade de texturas, partículas e sombras, quantidade de efeitos de física, e outros) (iluminação e efeitos especiais são cobertos pela diretriz 2.3);
- Campo de visão: o campo de visão em jogos tridimensionais deve ser automaticamente ajustado conforme o dispositivo de visualização do jogador ou deve poder ser escolhido;
- Segundo plano: elementos de segundo plano e elementos não interativos devem poder ter seu movimento e sua animação reduzidos ou desabilitados.

1.3.2. Feedback (visual): indicativos ou respostas visuais imediatos devem existir ou poder ser habilitados/desabilitados para todas as informações essenciais, incluindo: (Nível A)

- Entrada de dados: indicação de acionamento dos principais comandos no dispositivo de entrada de dados configurado pelo jogador (como animação de recarga de arma após pressionar tecla no teclado, mudança de cursor para um alvo após selecionar uma habilidade com um clique de mouse, mudança de inclinação do personagem após mudar orientação de dispositivo com acelerômetro ou giroscópio);
- Eventos: indicação de eventos essenciais para o jogador (como recebimento de dano, aparecimento novas missões, ativação de uma habilidade, e outros), incluindo todas as informações essenciais adicionais (direção da fonte de dano, posição ou direção das novas missões, identificação e duração da habilidade ativada);

- Estado: indicação de mudanças essenciais de estado para o jogador (como evolução de nível, morte do personagem, seleção de uma unidade, e outros), incluindo todas as informações essenciais adicionais (quantidade de níveis obtidos, causa/origem da morte do personagem e posição ou direção do corpo do personagem morto, posição ou direção da unidade selecionada);
- Interação: indicação dos elementos com os quais o jogador está interagindo diretamente em um dado momento (como mercador ou loja selecionado, personagem com o qual está se comunicando, alvo de um ataque, e outros);
- Progresso: indicação da evolução de um processo ou uma tarefa no jogo (tempo de conjuração de uma magia, tempo para armar uma bomba, tempo restante para fugir de um local, e outros) ou do carregamento de um novo cenário.

1.3.3. Som (mono/estéreo): som deve poder ser escolhido entre monoaural e estereofônico. (Nível A)

1.3.4. Feedback (sonoro): indicativos sonoros imediatos devem existir ou poder ser habilitados/desabilitados para todas as informações essenciais, incluindo: (Nível A)

- Entrada de dados: indicação de acionamento dos principais comandos no dispositivo de entrada de dados configurado pelo jogador;
- Eventos: indicação de eventos essenciais para o jogador, incluindo todas as informações essenciais adicionais;
- Estado: indicação de mudanças essenciais de estado para o jogador, incluindo todas as informações essenciais adicionais;
- Interação: indicação dos elementos com os quais o jogador está interagindo diretamente em um dado momento;
- Progresso: indicação da evolução de um processo ou uma tarefa no jogo ou do carregamento de um novo cenário.

1.3.5. Feedback (outros): indicativos imediatos devem existir ou poder ser habilitados/desabilitados para todas as informações essenciais e para todos os dispositivos de saída configurados pelo jogador (como controle com vibração), incluindo: (Nível AA)

- Entrada de dados: indicação de acionamento dos principais comandos nos dispositivos de entrada de dados configurados pelo jogador;
- Eventos: indicação de eventos essenciais para o jogador, incluindo todas as informações essenciais adicionais;
- Estado: indicação de mudanças essenciais de estado para o jogador, incluindo todas as informações essenciais adicionais;
- Interação: indicação dos elementos com os quais o jogador está interagindo diretamente em um dado momento;
- Progresso: indicação da evolução de um processo ou uma tarefa no jogo ou do carregamento de um novo cenário.

1.3.6. Orientação e localização alternativas (básico): alternativas sonoras para orientação e localização devem existir ou poder ser habilitadas/desabilitadas, incluindo ao menos uma das alternativas a seguir: (Nível AA)

- Bússola: um sistema de narração ou som tridimensional indica a orientação do jogador em relação a objetivos ou pontos cardeais no mundo virtual;
- Sonar: um sonar pode ser acionado para informar por meio de som tridimensional a posição de objetivos e obstáculos.

1.3.7. Som (surround): som surround ou tridimensional deve poder ser habilitado/desabilitado. (Nível AA)

1.3.8. Opções simplificadas: a interface de menus e opções deve possuir uma versão simplificada que pode ser habilitada/desabilitada e que apresenta apenas os controles essenciais ou os controles mais comuns. (Nível AA)

1.3.9. Comunicação multijogador: se o jogo permitir interação multijogador sobre uma rede de computadores, todos os seguintes critérios devem ser atendidos: (Nível AA)

- Bate-papo textual: deve ser possível comunicar-se com outros jogadores por meio de texto;
- Bate-papo falado: deve ser possível comunicar-se com outros jogadores por meio de canais de voz;
- Escolher bate-papo: deve ser possível escolher iniciar um jogo no qual os demais jogadores usem apenas bate-papo textual ou apenas bate-papo falado.

1.3.10. Comunicação multijogador rápida: se o jogo permitir interação multijogador sobre uma rede de computadores, todos os seguintes critérios devem ser atendidos: (Nível AA)

- Alertas rápidos: deve ser possível disparar alertas rápidos (pings) em um minimapa ou no cenário;
- Comunicação rápida: deve ser possível expressar informações simples por meio de comunicação não verbal e comunicação pictográfica como emissão de sons, emoticons ou animações.

1.3.11. Visão periférica: para informações essenciais temporárias ou com limite de tempo (como indicação de missões opcionais, objetivos com limite de tempo ou eventos rápidos) que são visualmente apresentadas durante o jogo, um dos critérios abaixo deve ser atendido: (Nível AA)

- Posição central: todas as informações relevantes são mostradas na área central onde se concentra a interação com o jogo e onde o jogador mais mantém a atenção;
- Alerta na posição central: um alerta é mostrado na área central onde se concentra a interação com o jogo e onde o jogador mais mantém a atenção indicando a presença das informações essenciais e onde podem ser encontradas na tela.

1.3.12. Gráficos (aprimorado): a estrutura visual do jogo deve poder ser ajustada em todos os seguintes elementos: (Nível AAA)

- Tamanho de elementos: grupos de elementos essenciais na interação devem poder ter o tamanho significativamente aumentado (mais que 150% do original) em conjunto ou individualmente;

- Remoção de detalhes: elementos visuais do jogo sem informação essencial devem poder ser removidos completamente (incluindo partículas específicas, sombras, efeitos de física e outros) (iluminação e efeitos especiais são cobertos pela diretriz 2.3);
- Remoção de segundo plano: elementos de segundo plano ou elementos não interativos devem poder ser completamente removidos.

1.3.13. Sem gráficos tridimensionais: a renderização de elementos tridimensionais deve poder ser completamente desabilitada. (Nível AAA)

1.3.14. Som (binaural): som binaural deve poder ser habilitado/desabilitado ou deve substituir som estereofônico. (Nível AAA)

1.3.15. Som (graves ampliados): som com graves ampliados deve poder ser habilitado/desabilitado. (Nível AAA)

1.3.16. Narração de comandos: narração de comandos e controles seletores deve poder ser habilitada/desabilitada para todos os elementos com informação essencial, incluindo: (Nível AAA)

- Comandos: sempre que o jogador aciona um comando em um dispositivo, esse comando deve ser narrado;
- Controles seletores: sempre que o jogador seleciona ou ajusta uma opção (como botões, barras deslizantes ou opções de menus), o nome, o papel e o valor desse controle devem ser narrados;
- Tabelas: para cada célula selecionada, o cabeçalho e o conteúdo devem ser narrados;
- Modelos e imagens de elementos interativos: sempre que um personagem, objeto ou outro elemento interativo for selecionado, o nome, o estado e a descrição desse elemento devem ser narrados;
- Menus e opções de instalação: todo o processo de instalação deve ser narrado segundo os critérios anteriores.

1.3.17. Orientação e localização alternativas (aprimorado): alternativas sonoras para orientação, localização e descrição do ambiente devem poder ser habilitadas/desabilitadas, incluindo: (Nível AAA)

- GPS falado: um sistema de narração deve poder ser acionado para informar a posição e a orientação do jogador, fornecer as direções para seu objetivo e descrever o ambiente e os objetos nas proximidades.

1.3.18. Audiodescrição (interativa): um sistema de narração deve poder ser habilitado/desabilitado para informar as características gerais de um elemento selecionado ou resumir as características básicas de um ambiente visualizado pelo jogador. (Nível AAA)

Diretriz 1.4. Discernível: torne mais fácil para os jogadores ver e ouvir conteúdos, incluindo separar primeiro plano de plano de fundo.

1.4.1. Fonte e formatação legíveis: para os textos no jogo, todos os seguintes critérios são atendidos: (Nível A)

- Fonte: fontes padrão (incluindo menus, legendas e balões de dicas) devem possuir formas claras e não devem ter tamanho pequeno (preferencialmente maior que 10 pontos; sob nenhuma circunstância menor que 8 pontos), exceto quando houver uma opção alternativa e claramente marcada para visualizá-los de forma mais clara;
- Parágrafos: parágrafos não devem ser justificados para textos com três ou mais linhas, não devem possuir frases com todas as letras maiúsculas e não devem possuir mais que 70 caracteres por linha.

1.4.2. Uso de cor: cores nunca são usadas como único meio de indicar uma ação ou resposta, apresentar uma informação essencial ou distinguir informações essenciais no jogo. (Nível A)

1.4.3. Controle de áudio: o volume de cada importante fonte de sons deve poder ser ajustado ou desabilitado separadamente especialmente para os sons não

essenciais na progressão do jogo (incluindo sons de fundo, música de fundo, efeitos sonoros e diálogos de personagens). (Nível AA)

1.4.4. Contraste (mínimo): a apresentação visual de textos e elementos interativos essenciais (como personagens, elementos do menu e cursores) deve possuir um contraste mínimo de 4.5:1, exceto para textos com tamanho muito grande (cujo contraste mínimo pode ser 3:1) e elementos não interativos ou irrelevantes para progressão no jogo. (Nível AA)

1.4.5. Cores alternativas: em relação a cores que representam informações essenciais, ao menos uma das seguintes alternativas é implementada: (Nível AA)

- Modos para acromatopsia: modos alternativos de cores devem poder ser habilitados/desabilitados para que um jogador daltônico escolha aquele que se adapte melhor;
- Personalização de cores: para elementos com informações essenciais (como diferenciação entre unidades aliadas e inimigas), as cores devem poder ser selecionadas para que um jogador escolha aquele que se adapte melhor.

1.4.6. Espaçamento e tamanho: todos os elementos interativos devem possuir espaçamento padrão e tamanho inicial padrão que tornem mínima a precisão necessária para operá-los, especialmente em telas pequenas (como telas de dispositivos móveis). (Nível AA)

1.4.7. Redimensionar fontes: todo o texto deve poder ser redimensionado em até 200% sem perda de conteúdo ou funcionalidade e sem a necessidade de tecnologias assistivas. (Nível AA)

1.4.8. Distinção de sons: deve ser garantido que todos os efeitos sonoros e músicas com informação essencial ou relevantes para entender o estado do jogo (como sons diferentes de disparo para armas diferentes ou músicas distintas para ambientes seguros e hostis) são facilmente diferenciáveis. (Nível AA)

1.4.9. Baixo ou nenhum áudio de fundo: o volume dos sons de fundo deve poder ser desabilitado ou deve estar 20dB abaixo do som principal de uma fala (como narração ou diálogo) ou informação sonora essencial, exceto para sons ocasionais que duram menos que dois segundos. (Nível AA)

1.4.10. Vozes (básico): a velocidade e o tom das vozes devem poder ser ajustados (incluindo diálogos, vídeos e narração de comandos). (Nível AA)

1.4.11. Contraste (aprimorado): para a apresentação visual do jogo, ao menos uma das alternativas a seguir é verdadeira: (Nível AAA)

- Alto contraste: a apresentação visual de textos e elementos interativos relevantes (como personagens, elementos do menu e cursores) deve possuir contraste com mínimo de 7:1, exceto para textos com tamanho muito grande (cujo contraste mínimo pode ser 4.5:1) e elementos não interativos ou irrelevantes para progressão no jogo;
- Preto e branco: a apresentação visual do jogo apenas em preto e branco deve poder ser habilitada/desabilitada.

1.4.12. Personalizar fontes: todo o texto deve poder ter o tipo e a cor da fonte alterados. (Nível AAA)

1.4.13. Vozes (aprimorado): para as vozes de narração mais importantes e presentes no jogo (como narração de história, audiodescrição ou narração de comandos), uma das alternativas está disponível: (Nível AAA)

- Escolha de narrador: ao menos duas vozes distintas (uma masculina e outra feminina) devem estar disponíveis para escolha;
- Alteração de timbre: o timbre das vozes deve poder ser ajustado.

Princípio 2: Operável - Os componentes de interface, interação e navegação do jogador devem ser operáveis.

Diretriz 2.1. Acessível por teclado: torne todas as funcionalidades disponíveis por um teclado.

2.1.1. Teclado: todos os conteúdos e funcionalidades do jogo devem ser acessíveis e operáveis por meio do teclado, exceto quando depende do caminho percorrido pelo movimento do jogador e não apenas dos pontos de origem e destino. Por exemplo, escrita à mão depende do caminho percorrido pelo movimento do dispositivo e não apenas da posição de clique inicial e final. (Nível AA)

2.1.2. Teclado (sem exceção): todos os conteúdos e todas as funcionalidades do jogo devem ser acessíveis e operáveis por meio do teclado. (Nível AAA)

Diretriz 2.2. Tempo suficiente: forneça aos jogadores tempo suficiente para ler, entender e usar conteúdos e funcionalidades.

2.2.1. Pausar e repetir: para animações ou eventos que tenham informações essenciais para progressão do jogo (como mensagens em tutoriais, diálogos entre personagens ou vídeos), todos os seguintes critérios são satisfeitos: (Nível A)

- Pausar: animações, mensagens e eventos podem ser pausados por tempo indeterminado e retomados com um comando simples;
- Repetir: animações, mensagens e instruções podem ser repetidas a qualquer momento e mesmo depois de terem sido exibidas.

2.2.2. Salvamento automático: o estado do jogo é automaticamente salvo em pontos estratégicos (incluindo mudanças de fase e pontos de controle que antecedem um desafio). (Nível AA)

2.2.3. Ajustar tempos: para animações, vídeos ou eventos que tenham uma duração ou um limite de tempo (como duração de uma animação, limite de tempo para derrotar um inimigo poderoso ou limite de tempo para escolher um personagem para iniciar o jogo), ao menos uma das alternativas a seguir é verdadeira: (Nível AA)

- Ajustar: é possível ajustar a velocidade de execução de animações ou vídeos;

- Estender: é possível estender o tempo disponível em eventos com um comando simples;
- Desabilitar: é possível desativar os limites de tempo em eventos;
- Pular: é possível pular eventos com limites de tempo sem prejuízos para o jogador no progresso no jogo;
- 20 horas: o tempo limite em eventos é maior que 20 horas.

2.2.4. Ajustar velocidade do jogo (básico): um controle da velocidade do jogo deve estar disponível, permitindo selecionar entre velocidade normal de ação do jogo ou velocidade reduzida (câmera lenta). (Nível AA)

2.2.5. Sem limites de tempo: nenhum evento possui um limite de tempo pré-determinado. (Nível AAA)

2.2.6. Ajustar velocidade do jogo (aprimorado): um controle da velocidade do jogo (ou outro mecanismo equivalente) deve estar disponível, permitindo selecionar velocidades entre velocidade normal de ação do jogo ou ação por turnos. (Nível AAA)

Diretriz 2.3. Convulsões e desorganização: permita a redução de efeitos que possam causar convulsões e evite eventos repentinos.

2.3.1. Iluminação e efeitos especiais (básico): a quantidade de efeitos especiais e fontes de iluminação deve poder ser reduzida de forma que quaisquer elementos que pisquem (como explosões, relâmpagos e luzes intermitentes) fiquem abaixo de três flashes por segundo. Elementos diferentes que estejam em uma mesma área que representa 10% do tamanho da tela do jogador e que pisquem simultaneamente devem, nesse caso, ser considerados um único elemento. (Nível AA)

2.3.2. Iluminação e efeitos especiais (avançado): efeitos especiais e múltiplas fontes de iluminação devem poder ser habilitados/desabilitados. (Nível AAA)

2.3.3. Eventos repentinos: movimentos ou eventos inesperados não devem ocorrer em jogos cuja mecânica não exige esse comportamento (incluindo jogos de estratégia em turnos e quebra-cabeças). (Nível AAA)

Diretriz 2.4. Navegável: forneça meios que auxiliem os jogadores a navegar pelo ambiente, acessar conteúdos e elementos interativos, e determinar onde estão.

2.4.1. Início rápido: opções de início rápido devem estar disponíveis de forma que o jogador não tenha que percorrer múltiplos níveis de menus para iniciar um novo jogo. (Nível A)

2.4.2. Salvamento e recuperação de estado: opções de salvar o estado atual e retornar a um estado anterior do jogo devem estar sempre disponíveis para o jogador (salvamento automático é coberto pela diretriz 2.2). (Nível AA)

2.4.3. Pular eventos: eventos que não fazem parte da mecânica principal do jogo devem poder ser pulados sem comprometer a progressão natural do jogo. (Nível AA)

2.4.4. Orientação: um mapa deve estar disponível para auxiliar na orientação e localização em ambientes complexos, indicando a posição do personagem e demais informações essenciais. (Nível AA)

2.4.5. Ordem de foco: se os elementos interativos (incluindo opções de menus, personagens e objetos interativos) podem ser navegados sequencialmente, a ordem correta na qual os componentes recebem foco deve poder ser programaticamente determinada. (Nível AA)

2.4.6. Acesso direto: partes e conteúdos especiais do jogo devem poder ser diretamente acessados pelo jogador sem que ele tenha que seguir uma estrutura linear ou superar um desafio (como encontrar um portal secreto em uma fase ou uma arma especial escondida) para obtê-los. (Nível AAA)

2.4.7. Orientação direta: a orientação direta usando pontos cardeais (e, opcionalmente, colaterais) deve poder ser configurada pelo jogador conforme o dispositivo de entrada de dados que esteja utilizando. (Nível AAA)

Diretriz 2.5. Configurável: permita que o jogador ajuste, simplifique e salve os controles para o jogo.

2.5.1. Salvamento e recuperação de opções (global): todas as alterações realizadas no comportamento padrão do jogo (alteração de controles, resolução, recursos de fala, volumes e outros) devem poder ser salvas pelo jogador e automaticamente recuperadas quando o jogo for reaberto. (Nível A)

2.5.2. Homogeneidade: o dispositivo de entrada de dados escolhido e configurado pelo jogador deve poder ser utilizado para interagir em todo o jogo sem a necessidade de outros dispositivos. Se mais de um dispositivo for configurado, ambos devem poder ser usados para interação. (Nível A)

2.5.3. Personalização de controles (básico): todos os controles essenciais do jogo devem poder ser remapeados ou reconfigurados para outros controles em qualquer dispositivo suportado pelo jogo. (Nível A)

2.5.4. Controles simplificados: para os controles do jogo, ao menos uma das seguintes alternativas é verdadeira: (Nível A)

- Simplicidade padrão: os controles do jogo são naturalmente simples e utilizam a menor quantidade de acionamentos possível nos dispositivos de entrada (geralmente, um comando para mover-se e um comando para todas as demais ações é a configuração mais simples possível);
- Modo simplificado: uma versão simplificada dos controles (que segue o princípio acima) pode ser habilitada/desabilitada.

2.5.5. Sensibilidade (básico): propriedades de sensibilidade básicas dos dispositivos de entrada devem poder ser ajustadas pelo jogador (como velocidade e aceleração de ponteiros ou sensibilidade dos direcionais de um joystick). (Nível A)

2.5.6. Comandos de voz (básico): quando houver entrada por comandos de voz, ela nunca deve ser usada como único meio de executar um comando e deve ser apenas um método alternativo. (Nível A)

2.5.7. Personalização de controles (aprimorado): todos os controles do jogo devem poder ser remapeados para outros controles em qualquer dispositivo de entrada de dados do jogador. (Nível AA)

2.5.8. Simultaneidade de controles: em relação à necessidade de acionamento simultâneo de controles (como pressionar um direcional e um botão ao mesmo tempo), um dos critérios a seguir é atendido: (Nível AA)

- Controles automáticos: os controles que são essencialmente simultâneos devem possuir uma opção de acionamento automático pelo jogo como, por exemplo, aceleração e freio automáticos, pulo automático ao atingir um obstáculo, disparo automático ao ter um inimigo na mira, recarga automática ao ficar sem munição, varredura automática de elementos em um menu, e outros.
- Sem controles simultâneos: o acionamento de controles simultâneos não é necessário e existe apenas como método complementar de interação.

2.5.9. Tempos específicos: não existem tempos específicos para o acionamento de controles ou de elementos interativos (incluindo ativar rapidamente uma sequência de controles, pressionar múltiplas vezes um botão e eventos de reação rápida). (Nível AA)

2.5.10. Movimentação e orientação: se houver controle simultâneo e assíncrono de movimento e orientação (como andar e apontar uma arma com dois controles distintos), uma das alternativas está disponível: (Nível AA)

- Desabilitar: deve ser possível desabilitar a mudança de orientação sem interferir na mecânica do jogo;
- Chavear: deve ser possível trocar entre movimento e mudança de orientação pelo acionamento de um comando;

- Automatizar: deve ser possível habilitar/desabilitar movimento ou orientação automático.

2.5.11. Comandos de voz (aprimorado): para a entrada de dados realizada pelo microfone (incluindo comandos de voz), uma das seguintes alternativas é verdadeira: (Nível AA)

- Vocabulário restrito: nenhum comando exige articulação de fala maior que uma pequena palavra (como "sim" ou "abrir");
- Limiar sonoro: o comando por voz é binário e acionado quando o volume captado pelo microfone ultrapassa um limiar sonoro (geralmente 50%), podendo ser acionado com sopros ou batidas.

2.5.12. Teclas, botões e comandos especiais: teclas, botões e os comandos especiais da plataforma do jogo (como teclas de atalho do sistema operacional ou botões de atalho do console) devem poder ser habilitadas/desabilitadas enquanto o jogo estiver executando, exceto para teclas e comandos emergenciais (como atalho padrão para fechar aplicativo ou botão de menu global do console). (Nível AA)

2.5.13. Sensibilidade (aprimorado): propriedades de sensibilidade avançadas dos dispositivos de entrada devem poder ser ajustadas pelo jogador (como velocidade de repetição de um controle ao mantê-lo pressionado, velocidade para reconhecer múltiplos acionamentos de botões em mouse ou gamepads, e mecanismos para ignorar acionamento acidental de novos comandos por 0.5 segundos após um primeiro comando ser recebido). (Nível AA)

2.5.14. Salvamento e recuperação de opções (perfis): todas as alterações realizadas no comportamento padrão do jogo (alteração de controles, resolução, recursos de fala, volumes e outros) devem poder ser salvas pelo jogador e recuperadas por meio da seleção de um perfil de jogador antes do jogo ser iniciado. (Nível AAA)

Diretriz 2.6. Compatível com dispositivos: torne o jogo compatível com a maior quantidade possível de dispositivos de entrada de dados.

2.6.1. Simultaneidade de dispositivos: todos os dispositivos de entrada de dados suportados podem ser simultaneamente utilizados. (Nível AA)

2.6.2. Dispositivos clássicos: deve haver compatibilidade (ou seja, controles podem ser remapeados para o dispositivo em questão) com ao menos um dispositivo específico de cada uma das classes a seguir: (Nível AA)

- Teclado: incluindo teclados físicos padrão e reduzidos;
- Mouse: incluindo mouses padrão, touchpads, trackpoints, trackballs e superfícies sensíveis a toque;
- Joystick ou gamepad.

2.6.3. Dispositivos alternativos (básico): deve haver compatibilidade (ou seja, controles podem ser remapeados para o dispositivo em questão) com ao menos um dispositivo específico de uma das classes a seguir: (Nível AA)

- Teclados virtuais: essencialmente teclados virtuais;
- Teclados alternativos: incluindo teclados simplificados ou reduzidos;
- Apontadores alternativos: incluindo pistolas e mesas digitalizadoras;
- Sensores de movimento: incluindo device tracking, body tracking e eye tracking;
- Acionadores: incluindo chaves, pedais, microfone e tapetes para jogos de ritmo.

2.6.4. Dispositivos alternativos (aprimorado): deve haver compatibilidade (ou seja, controles podem ser remapeados para o dispositivo em questão) com ao menos um dispositivo específico de cada uma das classes a seguir: (Nível AAA)

- Teclados virtuais: essencialmente teclados virtuais;
- Teclados alternativos: incluindo teclados simplificados ou reduzidos;
- Apontadores alternativos: incluindo pistolas e mesas digitalizadoras;
- Sensores de movimento: incluindo device tracking, body tracking e eye tracking;

- Acionadores: incluindo chaves, pedais, microfone e tapetes para jogos de ritmo.

Princípio 3: Compreensível - Informação e interface com o jogador devem ser compreensíveis.

Diretriz 3.1. Legível: faça o conteúdo textual legível e compreensível.

3.1.1. Linguagem simples (básico): todo conteúdo textual (incluindo opções de menus) deve ser escrito em linguagem simples, objetiva e sucinta (todas as frases devem possuir 50 palavras ou menos). (Nível A)

3.1.2. Idioma do jogo: o idioma do jogo deve poder ser programaticamente determinado. (Nível AA)

3.1.3. Linguagem simples (avançado): todo conteúdo textual (incluindo opções de menus) deve respeitar todos os seguintes critérios: (Nível AAA)

- Palavras não usuais: o jogo deve possuir algum mecanismo para explicar o significado de todas as palavras não usuais (como termos técnicos ou jargão do jogo);
- Abreviações: o jogo deve possuir algum mecanismo para expandir quaisquer siglas e abreviações apresentadas, apresentando seu texto completo.

3.1.4. Idioma das partes: o idioma de algum elemento do jogo (conteúdo de uma carta, pichação em uma parede, capa de um livro, e outros) deve poder ser programaticamente determinado. (Nível AAA)

Diretriz 3.2. Previsível: faça com que os desafios, os conteúdos e as funcionalidades do jogo apareçam e funcionem de formas previsíveis e esperadas.

3.2.1. Progressão de dificuldade (básico): o nível de desafio do jogo deve ser gradualmente aumentado conforme a progressão natural do jogo (conforme o jogo

evolui e os personagens ficam mais poderosos, os próximos desafios tornam-se maiores). (Nível A)

3.2.2. Foco: quando um elemento interativo ganha ou perde foco, ele não deve iniciar uma alteração de contexto que interfira com uma informação essencial que está sendo exibida (como fazer uma dica importante desaparecer ao passarmos o mouse sobre outro personagem ou pular para a próxima fase automaticamente após concluir um diálogo), exceto se for a funcionalidade específica daquele elemento. (Nível A)

3.2.3. Progressão de dificuldade (aprimorado): o nível de desafio do jogo deve ser gradualmente aumentado conforme a progressão das habilidades do jogador (conforme o jogador adquire experiência, superando desafios com maior velocidade ou consumindo uma quantidade menor de recursos, os próximos desafios tornam-se maiores). (Nível AA)

3.2.4. Interação estacionária: todos os elementos interativos que necessitam de precisão para ativação não devem se movimentar ou mudar de estado (como opções que se movimentam ou menu drop-down), exceto quando a necessidade de precisão fizer parte da mecânica de jogo (disparar contra um inimigo em movimento). (Nível AA)

3.2.5. Reconhecimento de estado salvo: ao salvar um estado do jogo, um nome (que pode ser escolhido pelo jogador no caso de salvamento manual) e uma miniatura devem ser associados ao estado gravado para facilitar o reconhecimento do contexto de salvamento. (Nível AA)

3.2.6. Narrativa: em relação à estrutura narrativa do jogo, ao menos uma das características abaixo deve ser atendida: (Nível AA)

- Narrativa simples: a estrutura narrativa do jogo é naturalmente simples e pode ser facilmente compreendida, lembrada e prevista pelo jogador;
- Resumos de narrativa: o jogo fornece algum mecanismo (presente no menu de opções, no início de cada jogo, nas telas de transição de cenários, ou

em outros locais) para que o jogador tenha uma perspectiva geral dos eventos passados, um panorama do estado atual e uma ideia dos próximos desafios ou missões.

Diretriz 3.3. Assistência: ajude jogadores a evitar e corrigir enganos.

3.3.1. Ajuda (manual): um sistema de ajuda deve poder ser acionado a qualquer momento para auxiliar o jogador com instruções, lembretes ou dicas de como superar um desafio. (Nível A)

3.3.2. Ajuda (automática): um mecanismo de ajuda que identifica dificuldades na interação ou no progresso no jogo deve poder ser habilitado/desabilitado para auxiliar o jogador com instruções, lembretes ou dicas de como superar um desafio ou progredir no jogo. (Nível AA)

3.3.3. Modos assistivos: um mecanismo de auxílio nas mecânicas do jogo (como mira automática, desaceleração automática em curvas e uso automático de cura) deve poder ser habilitado/desabilitado. (Nível AA)

3.3.4. Ajuste de dificuldade (automático): a dificuldade dos desafios deve ser automaticamente ajustada conforme a habilidades e dificuldades do jogador. (Nível AA)

3.3.5. Passo a passo para configurações: um mecanismo de passo a passo deve estar disponível para auxiliar o jogador escolher as configurações mais adequadas para sua experiência de jogo. (Nível AAA)

Diretriz 3.4. Documentação: disponibilize documentação acessível e em múltiplos formatos.

3.4.1. Características de acessibilidade: detalhes dos recursos de acessibilidade do jogo e de seus requisitos devem estar disponíveis e claramente

indicados na embalagem ou no local de disponibilização do jogo (como página web ou software de compra de jogos). (Nível A)

3.4.2. Manuais (online): as versões web dos manuais devem estar em formatos acessíveis. (Nível A)

3.4.3. Manuais (offline): as versões digitais locais dos manuais devem estar em formatos acessíveis (preferencialmente em HTML). (Nível AA)

3.4.4. Multimeios: devem ser disponibilizados e promovidos múltiplos recursos complementares para auxiliar no entendimento das mecânicas do jogo (incluindo brinquedos, adesivos, livros, vídeos, wikis e páginas dedicadas na web). (Nível AA)

3.4.5. Exportação de ajuda: elementos de ajuda encontrados dentro do jogo devem poder ser exportados para fora do jogo em documentos em formatos acessíveis (preferencialmente HTML). (Nível AAA)

Diretriz 3.5. Aprendizado e desafio: disponibilize modos de treinamento e ajuste manual do nível de desafio.

3.5.1. Ajuste de dificuldade (geral): o nível de desafio deve poder ser selecionado a partir de uma lista (contendo, por exemplo, níveis de dificuldade fácil, normal e difícil) e deve garantir mudanças significativas no desafio do jogo. (Nível A)

3.5.2. Modo de treinamento (guiado): um modo de treino guiado deve ser disponibilizado no qual o jogador pode treinar as mecânicas do jogo seguindo instruções em um passo a passo. (Nível A)

3.5.3. Ajuste de dificuldade (individual): os níveis de desafio relacionados a diferentes elementos do jogo (como inteligência dos inimigos, dificuldade dos quebra-cabeças, e outros) devem poder ser individualmente ajustados e devem

garantir mudanças significativas no desafio do jogo. Todos os ajustes de dificuldade devem estar disponíveis em qualquer ponto do jogo. (Nível AA)

3.5.4. Tutoriais dentro do jogo: ao iniciar um novo jogo, deve ser disponibilizado um modo tutorial que apresenta e treina com o jogador os controles básicos. (Nível AA)

3.5.5. Modo de treinamento (livre): um modo de treino livre deve ser disponibilizado no qual o jogador pode praticar livremente as mecânicas do jogo e permanecer em um ambiente de experimentação seguro para ajustar as configurações. (Nível AA)

3.5.6. Equilíbrio multijogador: se o jogo permitir interação multijogador sobre uma rede de computadores, todas as seguintes opções devem estar disponíveis ao iniciar um jogo: (Nível AA)

- Jogar somente com outros jogadores que usam recursos de acessibilidade do jogo;
- Jogar somente com outros jogadores que não usam recursos de acessibilidade do jogo;
- Sem restrições, ou seja, jogar com outros jogadores com quaisquer configurações.

Princípio 4: Robusto - O conteúdo deve ser interpretado de forma confiável por uma ampla variedade de dispositivos de saída, incluindo tecnologias assistivas.

Diretriz 4.1. Compatível: maximize a compatibilidade com os recursos de acessibilidade do jogador, incluindo tecnologias assistivas.

4.1.1. Suporte coerente: todos os elementos (incluindo menus e processo de instalação) devem possuir os mesmos recursos de acessibilidade disponíveis durante o jogo. Assim, se há compatibilidade com leitores de tela ou com teclado

virtual durante o jogo, essa compatibilidade também existirá nos menus e no processo de instalação. (Nível A)

4.1.2. Modo janela: a exibição em modo janela deve poder ser habilitada/desabilitada para utilização de tecnologias assistivas em software, incluindo teclados virtuais. (Nível AA)

4.1.3. Nome, papel e valor: para todos os componentes de interação (incluindo opções de menu, controles seletores, balões de dicas e links), todos os seguintes critérios são atendidos: (Nível AA)

- O nome e o papel podem ser programaticamente determinados;
- O estado, as propriedades e os valores podem ser alterados pelo jogadores e programaticamente determinados;
- A notificação da mudança desses itens deve estar disponível para os recursos do jogador, incluindo tecnologias assistivas.

4.1.4. Conflitos: as combinações de teclas usadas como atalhos no jogo não devem estar em conflito com as combinações de teclas usadas como atalhos pelos principais leitores de tela e outras tecnologias assistivas disponíveis para a plataforma alvo. (Nível AAA)

4.1.5. Efeitos colaterais: a instalação ou a execução de um jogo não deve alterar os arquivos de sistema ou as configurações de quaisquer outros softwares instalados (como drivers e bibliotecas de recursos), evitando assim comprometer recursos utilizados por tecnologias assistivas. (Nível AAA)