UNIVERSIDADE DE CAXIAS DO SUL
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
BACHARELADO EM TECNOLOGIAS DIGITAIS

LUÍS FILIPE SEVERGNINI

Serious game como ferramenta de ensino de lógica de programação para crianças

Marcelo Luís Fardo
Orientador

Caxias do Sul, Novembro de 2016

Serious game como ferramenta de ensino de lógica de programação para crianças

por

Luís Filipe Severgnini

Projeto de Diplomação submetido ao curso de Bacharelado em Tecnologias Digitais, do Centro de Ciências Exatas e Tecnologia, da Universidade de Caxias do Sul, como requisito obrigatório para graduação.

Trabalho de Conclusão de Curso

Orientador: Marcelo Luís Fardo
Banca examinadora:

Carlos Eduardo Nery
CCET/UCS
Elisa Boff
CCET/UCS

SUMÁRIO

LISTA DE ACRÔNIMOS 4

LISTA DE FIGURAS 5

LISTA DE TABELAS 6

RESUMO 7

ABSTRACT 8

INTRODUÇÃO 9
Questão de pesquisa 10
Objetivos 10

1 REFERENCIAL TEÓRICO 11
1.1 *Serious Games* 11
1.1.1 Jogos sérios e *edutainment* 12
1.1.2 Por que utilizar *serious games*? 13
1.2 Aprendizagem 14
1.3 Jogos e diversão 17
1.3.1 A importância da diversão nos *serious games* 17
1.3.2 O que faz um jogo ser seriamente divertido? 19
1.3.3 Modelo de três níveis de diversão em games 22

2 TRABALHOS RELACIONADOS 24
2.1 *CodeCombat* 24
2.2 Code.org 27

3 METODOLOGIA 30
3.1 Camada de aprendizagem 31
3.2 Camada de narrativa 32
3.3 Camada de *gameplay* 32
3.4 Camada de experiência do usuário 33
3.5 Camada de tecnologia 34

4 DESENVOLVIMENTO 35
4.1 Camada de aprendizagem 35
4.1.1 Por que *puzzles*? 35
4.1.2 *Puzzles* no jogo 36
4.1.3 Formato e linguagem 37
4.1.4 Conteúdo e nível de dificuldade 38
4.1.5 Lapidando o conceito inicial 41
4.2 Camada de narrativa 42
4.3 Camada de *gameplay* 43
4.3.1 O gênero do jogo 43
4.3.2 Mecânicas, modos e objetivos do jogo 44
4.4 Camada de experiência do usuário 47
4.5 Camada de tecnologia 48

5 PLAYTESTING 51
5.1 Resultados do *playtesting* 53

6 CONCLUSÃO 56

REFERÊNCIAS 59

APÊNDICE A - QUESTIONÁRIO PÓS-PLAYTESTING 63

LISTA DE ACRÔNIMOS

API *Application Programming Interface*
DPE *Design, play and experience*
ESRB *Entertainment Software Rating Board*
MDA *Mechanics, dynamics and aesthetics*
RPG *Role-playing game*
ZDP Zona de Desenvolvimento Proximal

LISTA DE FIGURAS

Figura 1.1: Esquema de funcionamento da Zona de Desenvolvimento Proximal proposto por Huguet (adaptado por FARDO, 2013) 15
Figura 1.2: O canal de fluxo (CSIKSZENTMIHALYI, 1990) 17

Figura 2.1: Componentes da interface gráfica do jogo *CodeCombat* 25
Figura 2.2: Componentes da interface gráfica do jogo *Code.org* 27

Figura 3.1: Estrutura expandida do framework DPE (WINN, 2008, tradução nossa) 31

Figura 4.1: Plataformas quebradas bloqueando o caminho do personagem . . 36
Figura 4.2: Interface do usuário dos *puzzles* 37
Figura 4.3: *Feedback* positivo após a resolução de um *puzzle* 38
Figura 4.4: *Puzzles* do jogo contém apenas instruções sequenciais 39
Figura 4.5: Solução para um *puzzle* de repetição 4x4 40
Figura 4.6: Solução para um *puzzle* 4x4 com estruturas de decisão 40
Figura 4.7: Evolução: ideia original, esboço intermediário e versão final 41
Figura 4.8: Mecânicas de movimentação básica 45
Figura 4.9: Obstáculos: plataformas, espinhos e penhascos 45
Figura 4.10: Itens escondidos em árvores e áreas de difícil acesso 46
Figura 4.11: Área secreta contendo moedas e itens raros 46
Figura 4.12: Indicadores de status e direção 47
Figura 4.13: Instruções referentes aos controles do jogo 48
Figura 4.14: Arquitetura física do jogo 49

LISTA DE TABELAS

Tabela 1.1: Frequência geral das categorias de conteúdo de fatores de diversão. Adaptado de Wang, Shen e Ritterfield (2009, pg. 31) . . . 20

Tabela 2.1: Lista de conceitos abordados no jogo *CodeCombat* 27
Tabela 2.2: Lista de conceitos abordados no jogo *Code.org* 29

Tabela 5.1: Resultados do Teste de Jogabilidade 54

RESUMO

Os *serious games* são ferramentas de aprendizagem eficazes, pois são motivadores e porque podem comunicar de forma eficiente conceitos e fatos sobre diversos assuntos. O objetivo deste trabalho era projetar e desenvolver um jogo sério para o ensino de lógica de programação. Para atingir este objetivo, foram investigadas as relações entre os *serious games* e a aprendizagem, do ponto de vista da teoria da Zona de Desenvolvimento Proximal de Vygotsky. Além disso, foram discutidas a importância da diversão nos jogos sérios e jogos digitais com uma proposta semelhante. O *serious game* foi desenvolvido na *engine* Unity3D, seguindo a metodologia proposta pelo *framework* DPE. O jogo foi avaliado a partir de sessões de *playtesting*, cujos resultados demonstraram que o *serious game* desenvolvido é divertido.

Palavras-chave: Jogos eletrônicos, *serious games*, aprendizagem, lógica de programação.

ABSTRACT

Serious games are effective learning tools because they are motivating and can communicate concepts and facts about many subjects in an eficient way. This paper's aim was to design and develop a serious game for teaching programming logic. In order to achieve this goal, we investigated the relations between serious games and learning from the perspective of Vygotsky's Zone of Proximal Development's theory. Moreover, we discussed the importance of fun in serious gaming as well as digital games with similar purposes. The game was built in Unity3D engine, based on the methodology proposed by DPE framework. The game was evaluated through playtesting sessions. Overall, playtesting results demonstrate that the developed serious game is fun.

Keywords: electronic games, serious games, learning, programming logic.

INTRODUÇÃO

Entre os jogadores de *videogames* não é incomum ouvir relatos de que um jogo ensinou mais sobre um determinado assunto do que as aulas convencionais de uma escola. Uma das possíveis explicações para esse fenômeno é que, por possuírem caráter de entretenimento, os jogos podem servir como motivação intrínseca para os alunos, fazendo com que eles aprendam inconscientemente, uma vez que estão focados em jogar, não em aprender (SHREVE, 2005).

Apesar disso, os jogos eletrônicos não têm a intenção de desmerecer os livros ou as aulas tradicionais da escola; muito pelo contrário: são projetados para entreter e podem servir como complemento para um determinado conteúdo escolar, além de facilitar a compreensão de situações complexas que não costumam ser representadas de forma eficiente nos livros. Consequentemente, podemos inferir que os jogos eletrônicos são uma forma de ensinar, o que os caracteriza como ferramenta de aprendizagem.

No entanto, é necessário ter cautela ao utilizar jogos comerciais como forma de ensino, pois eles podem conter elementos considerados impróprios ou negativos (RITTERFELD; WEBER, 2006). Afinal, seu objetivo principal não é ensinar; é entreter. Desta forma, como poderiam os jogos eletrônicos ganhar popularidade como ferramentas de aprendizagem?

Para esse fim, é necessário que o conteúdo dos jogos seja projetado para conter apenas elementos positivos e educacionais. Essa é a razão da existência dos *serious games*, os jogos "nos quais a educação (em suas várias formas) é o objetivo principal, ao invés do entretenimento" (MICHAEL; CHEN, 2005, tradução nossa), tema de pesquisa deste trabalho de conclusão de curso.

Além da motivação pessoal de trabalhar com design de jogos eletrônicos, a justificativa para a realização deste trabalho é a necessidade de implementação de games para um projeto de pesquisa da Universidade de Caxias do Sul, o projeto Gamification (UCS, 2013). Este projeto investiga as principais tecnologias disponíveis para o desenvolvimento de uma rede social gamificada, ou seja, uma rede social que faz "uso de mecânicas, estética e pensamentos dos games para envolver pessoas, motivar a ação, promover a aprendizagem e resolver problemas" (KAPP, 2012 apud FARDO, 2013, p. 63).

Desta forma, levando em conta o contexto no qual o *serious game* será integrado, qual seria uma boa temática de ensino para o jogo? Considerando que o público-alvo da rede social consiste em crianças e adolescentes, acredita-se que lógica de programação seria um bom tema de ensino. Afinal, além de auxiliar no desen-

volvimento do raciocínio lógico, o aprendizado desta disciplina poderia facilitar o processo de iniciação do aluno em um curso de computação, principalmente porque forneceria uma base para a disciplina de algoritmos - conhecida por ser difícil para pessoas sem conhecimento prévio de programação de computadores.

O primeiro capítulo deste trabalho descreve os conceitos básicos necessários para o entendimento (e a elaboração) de uma proposta de jogo sério. Obviamente, por se tratar do desenvolvimento um *serious game*, houve a necessidade de pesquisar os conceitos de *serious games* e de aprendizagem. Além disso, considerando que uma das premissas do jogo sério proposto é a diversão, será discutida a importância da diversão nos *serious games*, bem como os fatores que tornam os jogos eletrônicos divertidos.

Em seguida, no segundo capítulo, serão apresentados dois jogos eletrônicos que também abordam o ensino de lógica de programação, o CodeCombat e o Code.org. Ambos possuem programas de ensino muito interessantes e diversos elementos que influenciaram o desenvolvimento deste trabalho.

No terceiro capítulo, será apresentada a metodologia na qual o jogo proposto será desenvolvido, o *framework* DPE (*design, play, experience*), ao passo em que o quarto capítulo discutirá as decisões tomadas durante o processo de desenvolvimento deste trabalho. Finalmente, nos dois últimos capítulos serão apresentados os resultados dos testes de jogabilidade e as considerações finais, onde serão discutidos os resultados e as reflexões geradas por esta pesquisa.

Questão de pesquisa

Um jogo educativo para ensino de lógica de programação pode funcionar como produto de entretenimento e ferramenta de aprendizagem ao mesmo tempo?

Objetivos

O objetivo principal deste trabalho é desenvolver um jogo educativo para ensinar lógica de programação para crianças. Além disso, este trabalho possui os seguintes objetivos específicos:

- Investigar as relações entre serious games e a aprendizagem;
- Pesquisar jogos digitais semelhantes;
- Projetar e implementar um game educativo para ensino de lógica de programação;
- Planejar, aplicar e avaliar um teste de jogabilidade do game educativo.

1 REFERENCIAL TEÓRICO

Por maior que seja a ânsia de resolver um problema de pesquisa, antes de apresentar uma proposta de solução é necessário embasamento teórico. No caso deste trabalho em particular, o referencial teórico foi construído a partir de pesquisa bibliográfica e do estudo de trabalhos relacionados.

Na primeira seção deste capítulo, haverá uma introdução ao conceito de *serious games*, bem como a discussão acerca de sua utilização como ferramenta de aprendizagem. Em seguida, a teoria da Zona de Desenvolvimento Proximal será brevemente discutida, a fim de estabelecer uma base para a elaboração do conteúdo do *serious game*. Por fim, será discutida a importância da diversão nos *serious games* e quais os principais fatores que tornam um jogo divertido.

1.1 *Serious Games*

Os jogos sérios, como também são conhecidos os *serious games*, são aplicações de *hardware* ou *software* desenvolvidas a partir dos princípios do *game design*, cujo objetivo principal não é o entretenimento (RICHVOLDSEN, 2009). São jogos nos "quais a educação (em suas várias formas) é o objetivo principal, ao invés do entretenimento" (MICHAEL; CHEN, 2005, tradução nossa). Ritterfield, Cody e Vorderer (2009) argumentam que *serious games* são jogos desenvolvidos com conteúdo e propósito além do puro entretenimento, caracterizando-se normalmente como jogos para ensinar, treinar e promover hábitos saudáveis e mudança social.

Apesar de ser um campo de pesquisa relativamente novo, o termo *serious game* foi utilizado antes mesmo da popularização dos jogos digitais, há mais de 40 anos. Utilizando jogos de tabuleiro e de cartas como referência, em seu livro intitulado "Serious Games", de 1968, Clark Abt argumentava que jogos

> [...] podem ser jogados a sério ou casualmente. Nos importamos com os *serious games* no sentido de que estes jogos têm uma finalidade educativa explícita e cuidadosamente pensada, não se destinando a serem jogados principalmente pelo entretenimento. Isso não significa que os *serious games* não são, ou não deveriam ser, divertidos. (ABT, 1987, pg. 9, tradução nossa[1])

[1] "Games may be played seriously or casually. We are concerned with serious games in the sense that these games have an explicit and carefully thought-out educational purpose and are not intended to be played primarily for amusement. This does not mean that serious games are not,

Levando isso em consideração, Michael e Chen (2005) defendem que a definição mais simples para um *serious game* é um jogo que não tem entretenimento, prazer ou diversão como propósito primário. Não obstante, isso não significa que jogos sérios não são interessantes, prazerosos ou divertidos; a diferença é que eles possuem outro propósito, uma motivação ulterior. Este argumento será retomado adiante, onde será discutida a importância da diversão nos *serious games*.

Ainda no que diz respeito à conceituação de *serious games*, Ritterfield e Weber (2006) acrescentam que qualquer jogo digital pode apresentar formas de aprendizagem, independentemente do jogo ser considerado sério ou não. Entretanto, neste aspecto, o que diferencia um jogo digital usual de um jogo sério é que os *serious games* estão associados com características positivas, como seriedade, educação e aprendizado (RITTERFIELD; WEBER, 2006). Ben Sawyer (apud MICHAEL; CHEN, 2005), cofundador da *Serious Game Initiative*, por sua vez, explica que a palavra sério em *serious games* tem a intenção de refletir o propósito do jogo, o porquê de sua criação, não necessariamente se referindo ao conteúdo do jogo propriamente dito.

1.1.1 Jogos sérios e *edutainment*

Com o advento dos computadores pessoais multimídia nos anos 90, diversas ferramentas educacionais baseadas em *software* começaram a ser desenvolvidas para ensinar alunos da pré-escola e novos leitores. Esta forma de ensinar na qual o entretenimento é utilizado como meio para a educação ficou conhecida como *edutainment* (MICHAEL; CHEN, 2005, p. 24). A *Entertainment Software Rating Board* (*ESRB* (*Entertainment Software Rating Board*) ou Junta de Qualificação de Software de Entretenimento) define *edutainment* como "conteúdo que provém o desenvolvimento de habilidades específicas ou o reforço do aprendizado aos usuários, em um cenário de entretenimento" (ESRB, 2015, tradução nossa[2]).

Frequentemente, *edutainment games* são referidos como sinônimo de *serious games*. No entanto, apesar das semelhanças, Michael e Chen (2005) afirmam que os *serious games* ultrapassam o foco limitado das aplicações de *edutainment*, na medida em que abrangem todos os tipos de educação e todas as idades. Por isso, o *edutainment* pode ser considerado subconjunto do tópico global *serious games* (MICHAEL; CHEN, 2005, p. 24).

Para os propósitos deste trabalho, não é necessário estender a discussão acerca do conceito de *edutainment*. No entanto, é importante realçar essa diferença entre os dois tipos de jogos porque um dos trabalhos relacionados - a ser apresentado no

---

or should not be, entertaining."

[2] "Content of product provides user with specific skills development or reinforcement learning within an entertainment setting."

próximo capítulo - é um exemplo de *edutainment*.

1.1.2 Por que utilizar *serious games*?

Na época em que os *serious games* começaram a se firmar no ambiente acadêmico, Henry Jenkins, estudioso dos meios de comunicação, fez uma interessante análise a seu respeito. Em um de seus argumentos, Jenkins destacou que a utilização dos videogames apenas como meio de entretenimento parece limitar seu potencial como mídia (JENKINS, 2006). Em outras palavras, se os jogos são tão bem sucedidos como mídia de entretenimento, por que não utilizá-los para outros fins, como ensino e simulação?

O linguista, educador e entusiasta dos jogos, James Paul Gee (2003, 2005 apud WINN, 2008) acredita no potencial de ensino dos jogos, principalmente por causa da aprendizagem, pois eles apresentam a quantidade exata de desafio, ajuda e *feedback*, recompensando o domínio por meio de novos desafios. Lieberman (2006) aponta que, além da motivação, jogos podem ser utilizados para o ensino de percepção e coordenação, resolução de problemas, conhecimentos gerais, habilidades e comportamentos, auto-regulação e terapia, autoconhecimento, relações sociais e atitudes e valores. Além disso, os jogos sérios permitem que esse aprendizado ocorra sem as consequências do mundo real (MICHAEL; CHEN 2005), o que pode reduzir o medo de errar durante o processo de aprendizagem.

Gee argumenta ainda que

> Muitos dos bons jogos de computador e videogame são longos, complexos e difíceis, especialmente para os iniciantes. Não é sempre que as pessoas desejam fazer coisas difíceis. Perante o desafio de levá-las a fazer isso, normalmente há duas alternativas. Nós os forçamos, que é a solução que as escolas usam. Ou, uma tentação quando dinheiro está em jogo, nós podemos idiotizar o produto. Nenhuma dessas opções é viável para a indústria, ao menos neste momento. Eles não podem forçar as pessoas a jogar e a maioria dos jogadores ávidos não quer que seus jogos sejam idiotizados (GEE, 2004, tradução nossa[3])

Isso demonstra que a tarefa de ensinar por meio de um jogo não é trivial. Sendo assim, como os *game designers* conseguem fazer jogadores novos e sem experiência aprender seus jogos, independentemente da dificuldade? James Gee (2004) responde que os *game designers* têm encontrado métodos muito eficientes de levar pessoas

[3] "Many good computer and video games [...] are long, complex, and difficult, especially for beginners. People are not always eager to do difficult things. Faced with the challenge of getting them to do so, two choices are often available. We force them, which is the solution schools use. Or, a temptation when profit is at stake, we can dumb down the product. Neither option is open to the game industry, at least for the moment. They can't force people to play, and most avid players don't want their games dumbed down".

a aprender e a desfrutar a aprendizagem, isto é, jogos podem ensinar, mas cabe à equipe de desenvolvimento elaborar uma forma adequada de tornar isso viável. Complementando a linha de pensamento de Gee, Clark Abt defende que:

> *Games* são dispositivos eficazes de formação e ensino para estudantes de todas as idades e em muitas situações, pois são altamente motivadores e porque comunicam de forma muito eficiente os conceitos e os fatos de vários assuntos (ABT apud MICHAEL; CHEN, 2005, pg. 25, tradução nossa[4]).

Desta forma, pode-se concluir que um bom motivo para usar *serious games* é o grande potencial de ensino que eles possuem. Outro motivo é o fato de que a geração atual de estudantes e *trainees* cresceu em meio aos videogames. Portanto, por serem nativos digitais[5], ou seja, falantes nativos das linguagens digitais dos computadores, videogames e da Internet (PRENSKY, 2001), é muito provável que eles joguem e aprendam por meio dos *serious games*.

Obviamente, desenvolver um *serious game* não é tarefa fácil; esse processo envolve o esforço de uma equipe mista de desenvolvimento - composta por *game designers*, programadores, artistas, especialistas e educadores. Apesar disso, acredita-se que os benefícios proporcionados por seu uso são maiores do que os desafios de desenvolvimento. Antes de pensar em uma forma de desenvolver um *serious game*, todavia, é importante entender como funciona o processo de aprendizagem. Por isso, esse será o tópico da próxima seção[6].

1.2 Aprendizagem

Para desenvolver um *serious game*, um jogo que tem como foco principal ensinar ao invés de entreter, são necessárias habilidades além do *game design* propriamente dito. Uma dessas habilidades é saber ensinar. Por isso, para que um bom jogo sério possa ser elaborado, acredita-se ser necessário entender como funciona o processo de aprendizagem - preferencialmente, no âmbito dos games. Há evidências de que a teoria da Zona de Desenvolvimento Proximal de Vygotsky pode ser relacionada ao uso de jogos como ferramenta de ensino (HUGUET, 2012; FARDO, 2013). Porém, antes de estabelecer relações entre essa teoria e os *serious games*, é importante construir um breve entendimento do que é a Zona de Desenvolvimento Proximal.

[4] "Games are effective teaching and training devices for students of all ages and in many situations because they are highly motivating, and because they communicate very efficiently the concepts and facts of many subjects".

[5] Termo cunhado por Marc Prensky para se referir à geração que nasceu em meio às tecnologias digitais.

[6] O conteúdo apresentado na seção de Aprendizagem foi pensado com o intuito de introduzir brevemente um dos inúmeros instrumentos existentes para apoiar o planejamento pedagógico de um *serious game*. Obviamente, um trabalho cujo foco é a educação propriamente dita exigiria uma pesquisa mais aprofundada nessa área.

Ao tentar estabelecer uma relação entre aprendizado e desenvolvimento, Vygotsky (1998) observou que a capacidade de aprendizado assistido de crianças com o mesmo nível de desenvolvimento mental variava enormemente. Desta forma, verificou-se que elas não tinham a mesma idade mental e que o curso subsequente de seu aprendizado seria diferente. A essa diferença, Vygotsky deu o nome de Zona de Desenvolvimento Proximal (ZDP). A ZDP é

> [...] a distância entre o nível de desenvolvimento real, que se costuma determinar através da solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes (VYGOTSKY, 1998, pg. 112).

Conforme Fardo (2013), Vygotsky defendia que desta forma era possível melhor estimar o nível de desenvolvimento de uma criança, pois ela levava em conta as funções que ainda não haviam sido completamente desenvolvidas - o que é importante, considerando que os dois níveis de desenvolvimento são dinâmicos (FARDO, 2013). Isso significa que o que hoje faz parte do desenvolvimento potencial, sendo realizado apenas por meio de auxílio, amanhã será desenvolvimento real, sem a necessidade de auxílio, formando um ciclo recursivo. Para facilitar a visualização do funcionamento da ZDP, Huguet (2012) propôs a divisão do processo em quatro etapas recursivas, conforme a Figura 1.1:

Figura 1.1: Esquema de funcionamento da Zona de Desenvolvimento Proximal proposto por Huguet (adaptado por FARDO, 2013)

Para ilustrar o funcionamento deste ciclo recursivo, vejamos o seguinte exemplo:

- Um aluno sem conhecimento prévio em programação de computadores ingressa

em um curso de lógica de programação. A grade curricular deste curso inclui o ensino de algoritmos sequenciais, condicionais e iterativos. Neste estágio, o desenvolvimento real sobre lógica de programação deste aluno é nulo.

- A partir do momento em que o conteúdo (algoritmos sequenciais) começa a ser ensinado, o aluno ainda não consegue resolver os problemas individualmente. Por isso, necessita da ajuda do professor - o alguém mais experiente - para resolver problemas e construir conhecimento acerca deste assunto (etapa 1); a ZDP do aluno começa a ser perturbada.
- Depois de algumas aulas e vários exercícios sobre algoritmos sequenciais, o aluno avança em seus conhecimentos e começa a apresentar uma postura auto-suficiente em relação ao conteúdo que antes não conhecia (etapa 2). O conhecimento de algoritmos sequenciais passa a ser parte do desenvolvimento potencial do aluno.
- Ao internalizar os conteúdos sobre algoritmos sequenciais, o aluno deixa temporariamente a ZDP; o novo conhecimento obtido, que fazia parte do desenvolvimento potencial, passa a integrar seu desenvolvimento real (etapa 3).
- A quarta etapa consiste na recursão para as etapas anteriores. Desta forma, ao iniciar os estudos no segundo tópico do curso (algoritmos condicionais), o aluno utilizará o que foi internalizado para continuar expandindo seu conhecimento.

Idealmente, o processo de aprendizagem em um *serious game* deve acontecer da mesma forma como ilustrado no exemplo acima. A relação entre as duas situações fica evidente se substituirmos o "alguém mais experiente" (no caso do exemplo acima, o professor) pelo *serious game*. Isso significa que, ao interagir com um *serious game*, o aluno recebe auxílio até o momento em que ele consegue resolver os problemas sem assistência.

No entanto, não são todas as atividades ou problemas que conseguem perturbar a ZDP. Conforme Vygotsky (apud FARDO, 2013), as atividades propostas devem se adiantar ao desenvolvimento, interferindo no nível de desenvolvimento potencial de um indivíduo, isto é, devem estar ligeiramente acima do seu nível de habilidade. Por conseguinte, os problemas não podem nem ser tão fáceis a ponto de não perturbar a ZDP, nem tão difíceis a ponto de fugir completamente do nível de desenvolvimento potencial (ex: explicar conceitos de física quântica para alguém que está começando a estudar as leis de Newton).

Levando isso em consideração, Fardo (2013) observou que há uma relação direta entre o funcionamento do esquema da ZDP e o conceito de Fluxo (*flow*) de Mihaly Csikszentmihalyi (1990), frequentemente associado à diversão em jogos. "Fluxo é um estado mental de imersão total" (RABIN, 2011, pg. 77) no qual o indivíduo está tão concentrado em uma tarefa específica, que perde a noção de tempo e dos

problemas emocionais.

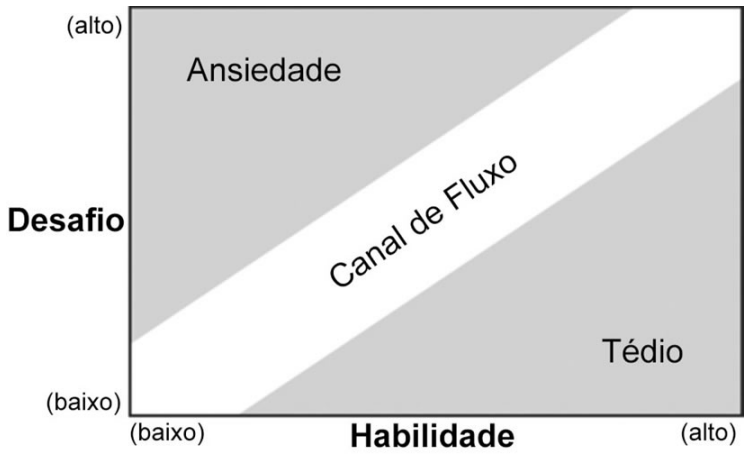

Figura 1.2: O canal de fluxo (CSIKSZENTMIHALYI, 1990)

A relação entre o Fluxo e o funcionamento da ZDP se torna evidente se considerarmos que para alcançar o estado de Fluxo é necessário um equilíbrio entre o desafio de uma tarefa e as habilidades necessárias para realizá-la - nem tão fácil, nem tão difícil (na Figura 1.2, a área branca em ascensão diagonal é o canal de Fluxo). Desta forma, o estado de Fluxo pode ser considerado um poderoso potencializador da ZDP (FARDO, 2013).

1.3 Jogos e diversão

A questão de pesquisa deste trabalho de conclusão envolve verificar se um jogo educativo para o ensino de lógica de programação é capaz de atuar como produto de entretenimento e como ferramenta de aprendizagem ao mesmo tempo. Previamente, neste capítulo, vimos que um jogo pode funcionar como ferramenta de aprendizagem. Desta forma, para responder à pergunta, é necessário ainda determinar uma forma de verificar se o jogo funciona como produto de entretenimento. Porém, antes de chegar a esse tópico, é importante entender a relação entre o ato de jogar e a diversão (entretenimento) - e qual a importância da diversão (se alguma) nos jogos sérios.

1.3.1 A importância da diversão nos *serious games*

Em 1938, em seu livro *Homo Ludens*, Johan Huizinga dissertou sobre o ato de jogar do ponto de vista filosófico e epistemológico. No que diz respeito à natureza e ao significado do jogo como fenômeno natural, Huizinga (1999) descreve seis características dos jogos:

1. Voluntariedade, uma forma de liberdade: nunca é imposto pela necessidade física ou pelo dever moral, tornando-se necessário apenas na medida em que o "prazer por ele provocado se o torna uma necessidade";

2. Fazer de conta: jogar não é "vida 'corrente' nem vida 'real"';
3. Imersão: jogos são capazes de absorver inteiramente o jogador;
4. Limitação, isolamento: são jogados "'até o fim' dentro de certos limites de tempo e de espaço";
5. Regras: jogos criam ordem e são ordem;
6. Socialização: criam grupos sociais de jogadores e tendem a causar pessoas envolvidas em um tipo específico de atividade a se identificar como um grupo.

Embora tenham sido cunhadas ainda na primeira metade do século XX, as definições de Huizinga parecem ser igualmente aplicáveis aos jogos eletrônicos que conhecemos hoje[7]. Mais recentemente, Steve Rabin (2011) aponta que o ato de jogar é voluntário, sem propósito aparente, diferente de comportamentos sérios, improvisado e, acima de tudo, divertido - "jogar é agradável"(RABIN, 2011, pg. 59). Rabin aponta ainda que é seguro afirmar que "o jogar faz o jogador se sentir bem. Jogadores estão se divertindo quando jogam"(Ibid.).

Apesar disso, um jogo é divertido apenas se os jogadores gostam de jogá-lo, pois

> [...] considerando que jogos são uma atividade voluntária, algo que o jogador escolhe fazer, há uma implicação de prazer, seja por causa da expectativa ou por uma experiência passada. Na ausência do prazer esperado ou por causa de uma experiência desagradável no passado, o jogador pode optar por não participar ou encontrar algo diferente para fazer. Em outras palavras, se o jogador não achar que o jogo é divertido, é improvável que ele o escolha para jogar novamente. (MICHAEL; CHEN, 2005, pg. 20, tradução nossa[8]).

Levando isso em conta, o quão importante é a diversão nos *serious games*? Para Michael e Chen (2005, pg. 41), o principal argumento em favor da diversão nos *serious games* é que ela ajuda a motivar os jogadores a interagir e a aprender por conta própria. No entanto, ainda há um debate acerca da importância da diversão nos jogos sérios. Afinal, os serious games são, por definição, jogos cujo propósito é ensinar - e ensinar nem sempre é divertido.

Em seu livro *Serious Games: Games That Educate, Train, and Inform*, Michael e Chen (Ibid.) demonstram os resultados de uma pesquisa realizada com desenvol-

[7] Obviamente, há algumas exceções. Afinal, Huizinga não poderia prever fenômenos como os *e-Sports* (competições de jogos eletrônicos com premiações em dinheiro, tal qual esportes tradicionais como futebol e basquete) e os *serious games*, categorias de jogos nas quais o jogador toma a posição de atleta ou aluno, onde a característica da voluntariedade/despretensão não se aplica.

[8] "Since games are a voluntary activity, something the player chooses to do, there is an implication of enjoyment, either in anticipation or based on past experience. In the absence of anticipated enjoyment, or because of an unpleasant earlier experience, the player may choose to not participate or find something else to do. In other words, if a player does not find a game fun, he is unlikely to choose to play it again".

vedores de *serious games*, educadores e pesquisadores. Segundo a pesquisa, 80% dos entrevistados responderam que o "elemento diversão" é importante ou muito importante. Como os próprios autores comentam, essa pesquisa está longe de retratar a indústria dos *serious games* por completo, mas o fato de nenhum entrevistado ter apontado o fator diversão como "não importante" não deve ser desconsiderado.

Apesar dessa tendência, é preciso ter cuidado para não desviar o foco do *serious game*, porque por mais que o jogador se divirta ao interagir com um jogo sério, se ele não conseguir aprender o conteúdo pretendido, o *serious* game terá fracassado. Em outras palavras, se os jogadores não conseguirem aplicar de uma forma útil o que aprenderam no mundo real, significa que o *serious game* falhou em sua missão (MICHAEL; CHEN, 2005, pg. 43).

1.3.2 O que faz um jogo ser seriamente divertido?

A diversão é um fator muito importante nos jogos de entretenimento e pode servir como motivação extra no caso dos *serious games*. Mas o que é diversão? Como podemos verificar se um jogo é divertido ou não? De acordo com o dicionário da língua portuguesa Aurélio (FERREIRA, 2004), diversão é "divertimento, entretenimento, distração" - ou seja, é algo que entretém ou ocupa. Não obstante, a diversão é subjetiva; todos têm suas próprias opiniões sobre o que é diversão. Por conseguinte, determinar se algo é divertido pode ser um grande desafio.

Considerando a ausência de um consenso acerca do que é diversão nos jogos eletrônicos, Wang, Shen e Ritterfield (2009) realizaram um estudo intitulado *"Enjoyment of Digital Games: What Makes Them 'Seriously' Fun?"*, no qual elaboram uma extensa análise de conteúdo de 60 *reviews*[9] profissionais de jogos eletrônicos, com o objetivo de determinar quais são os principais fatores de diversão no contexto dos jogos de entretenimento e quando um jogo digital pode ser considerado divertido. Evidentemente, os três reconhecem que os fatores que tornam um jogo de entretenimento agradável podem não ter o mesmo efeito em um jogo sério. Apesar disso, apontam que ao identificar os elementos dos jogos que contribuem para uma melhor experiência, é possível

> [...] estabelecer um referencial útil para a compreensão da diversão em ambos os jogos, sérios e de entretenimento, além de explorar novas estratégias para melhorar a qualidade da diversão em jogos sérios (WANG; SHEN; RITTERFIELD, 2009, pg. 25, tradução nossa[10]).

[9] *Reviews* de jogos eletrônicos são análises, críticas ou resenhas feitas por jogadores bastante experientes.

[10] "[...] we can establish a useful frame of reference for understanding media enjoyment in both entertainment and serious games, and for exploring new strategies to improve the fun quality of serious games".

A partir da análise dos resultados desse estudo, os autores encontraram os 27 fatores de diversão descritos na Tabela 1.1, onde estão listados em ordem decrescente de frequência geral. Os fatores mais predominantes nas *reviews* dos especialistas foram: *game design* geral (17,7% das menções em comentários), apresentação visual (13,1%), controle (9,6%), apresentação de áudio (6,9%) e complexidade e diversidade (6,6%). Em contrapartida, os menos citados foram: fantasia, presença e interatividade.

Tabela 1.1: Frequência geral das categorias de conteúdo de fatores de diversão. Adaptado de Wang, Shen e Ritterfield (2009, pg. 31)

Categoria de Conteúdo	**Definição**	**% Frequência**
Game Design geral	Comentários gerais sobre o *design* do jogo	17,7
Apresentação visual	A qualidade dos gráficos no jogo	13,1
Controle	A facilidade, intuitividade e eficácia dos controles	9,6
Apresentação de áudio	A qualidade da música, efeitos sonoros e dublagem	6,9
Complexidade e diversidade	A quantidade e qualidade de opções significativas apresentadas ao jogador e o quão bem essas opções colaboram entre si para possibilitar uma experiência de jogo profunda e intrigante	6,6
Experiência de jogo geral	Comentários gerais acerca da experiência de entretenimento durante o ato de jogar	4,6
Usabilidade	As funcionalidades e a estabilidade de um jogo, como o tempo de carregamento, taxa de *frames*, *bugs* ou navegabilidade dos menus	4,1
Mecânicas	O grau no qual as regras básicas e atividades principais do jogo estão bem-estabelecidas e agradáveis	4,1
Novidade	A originalidade ou inovação de um jogo, como a incorporação de novas ideias de uma maneira persuasiva versus a repetição de velhos conceitos	4,0
Trama	A existência e a qualidade das tramas e enredos em um jogo	3,8
Personagens	Se um personagem de um jogo é atrativo, reconhecível, customizável e se ele possui profundidade	3,6

Continua na próxima página

Tabela 1.1: Continuação

Categoria de Conteúdo	**Definição**	**% Frequência**
Integração social	A possibilidade, exigência e qualidade das interações humanas durante o jogo, especialmente relacionadas às características multijogador	3,6
Desafio	A dificuldade de um jogo e se ele é projetado para fornecer uma experiência equilibrada que não é nem frustrante, nem sem esforço para o jogador	2,8
Inteligência artificial	O design da e a interação com a inteligência artificial em um jogo	1,8
Duração	Se o jogo apresenta uma duração suficiente de interação antes de ser finalizado	1,7
Humor	O uso e a eficácia do humor em um jogo	1,6
Capacidade tecnológica geral	Comentários gerais sobre o aspecto tecnológico de um jogo	1,5
Níveis	A capacidade do design de níveis de fornecer estruturas eficientes para melhorar a experiência de jogo	1,5
Apresentação estética geral	Comentários gerais sobre a apresentação estética, como o aspecto visual, efeitos sonoros e estilo	1,4
Empolgação	O ritmo de um jogo e o prazer sensorial experienciado pelo jogador	1,3
Liberdade	O grau no qual a estrutura de um jogo permite aos jogadores seguir diferentes caminhos de ação à vontade	1,3
Fator *replay*	Se os jogadores querem jogar o jogo múltiplas vezes	0,9
Semelhança à realidade	Como um jogo se assemelha a ambientes, situações e interações sociais no mundo real	0,8
Gratificação	Quando elementos do jogo fornecem ao jogador uma noção de recompensa ao completar tarefas	0,8
Interatividade	O loop contínuo de ação-reação entre o(s) jogador(es) e o mundo do jogo	0,3
Presença	O nível no qual jogadores experienciam os objetos físicos virtuais, os atores sociais virtuais, e o seu "eu virtual" gerado por tecnologias de mídia como se fossem reais	0,3

Continua na próxima página

Tabela 1.1: Continuação

Categoria de Conteúdo	Definição	% Frequência
Fantasia	Se o jogo entrega aos jogadores uma experiência fantástica e imaginativa que normalmente é impossível na vida real	0,2

Dentre todas as *reviews* relacionadas aos fatores de diversão, 55.4% dos comentários foram positivos, 35.6% foram negativos e 9% foram comentários neutros. Respectivamente, as categorias mais e menos mencionadas nos comentários positivos foram as mesmas que apareceram no quadro geral: *game design* geral, apresentação visual, apresentação de áudio, complexidade e diversidade e controle, sendo os mais mencionados; fantasia, interatividade e presença, os menos mencionados. O conjunto de comentários negativos seguiu o mesmo padrão, com exceção da categoria de usabilidade, que apareceu com maior frequência do que a apresentação de áudio.

Ao analisar os dados coletados, os autores do estudo perceberam relações entre os 27 fatores, agrupando-os e os classificando em 5 categorias distintas, denominadas "as Cinco Grandes da diversão em jogos eletrônicos":[11] capacidade tecnológica, *game design*, apresentação estética, experiência de jogo de entretenimento e narrativa (Ibid. pg 42). Essa classificação não tem a intenção de reduzir os aspectos da diversão em games em apenas cinco grupos; ao invés disso, visa fornecer uma possível estrutura de taxonomia genérica para o entendimento do assunto (Ibid.).

1.3.3 Modelo de três níveis de diversão em games

Ainda em seu estudo, por meio de análises mais detalhadas do *ranking* de categorias de conteúdo, Wang, Shen e Ritterfield (2009, pg. 43) perceberam que alguns fatores como humor, mecânicas e gratificação, tendem a aparecer em comentários positivos mais frequentemente, enquanto outros, como controle, usabilidade, desafio e inteligência artificial eram mais comuns nos comentários negativos. Outra relação estabelecida foi a de que alguns fatores, como personagens, interação social, novidade e gratificação, contribuíram positivamente nos jogos divertidos, ao passo em que outros fatores, como experiência geral de jogo, trama, interação social e duração, parecem ter diminuído o valor do entretenimento dos jogos "não-divertidos" (Ibid.).

A partir dessas relações, os três autores entenderam que há certos limiares que um jogo deve passar para ser considerado jogável, divertido ou até mesmo super divertido (Ibid.). Por conseguinte, propuseram um modelo capaz de refletir essa ideia: o modelo de três níveis de diversão em games. Os níveis propostos são: limiar

[11] Os autores utilizam o termo "*Big Five of digital game enjoyment*" para se referir às seguintes categorias: "(1) *technological capacity*, (2) *game design*, (3) *aesthetic presentation*, (4) *entertainment game play experience*, (5) *narrativity*". O termo *Big Five* é uma alusão à expressão homônima utilizada na psicologia.

de jogabilidade, limiar de diversão e fatores de aumento da diversão[12].

O limiar de jogabilidade consiste no que é necessário para que o jogo seja jogável. Mais especificamente, refere-se aos elementos básicos dos jogos, como usabilidade, controle, desafios e apresentação visual (Ibid. pg. 43). Este nível do modelo de três níveis está diretamente relacionado com a capacidade tecnológica e serve como pré-requisito para o limiar de diversão. Afinal, é difícil de imaginar alguém se divertindo ao jogar um jogo feio, que demora para carregar e possui *bugs*.

O limiar de diversão, por sua vez, engloba os fatores de diversão relacionados às categorias de apresentação estética e ao *game design*. Este nível do modelo compreende elementos de jogos como qualidade audiovisual, complexidade e diversidade, mecânicas, liberdade, níveis, grau de dificuldade balanceado e gratificação (Ibid.). Wang et al. (2009) também explicam que um jogo divertido deve conter gráficos e efeitos sonoros agradáveis ao público, possuir diversas opções de exploração em diferentes níveis, apresentar tomadas de decisão e ação, além de ações interligadas às consequências que sucedem, possibilitando que os jogadores construam suas próprias trajetórias e experiências dentro do jogo (Ibid.). "Essas características satisfazem nosso desejo humano inato pela descoberta e resolução de problemas, criando sentimentos verdadeiros de prazer" (GEE, 2005, 2007 apud WANG; SHEN; RITTERFIELD, 2009, pg 44, tradução nossa[13]). Consequentemente, pode-se inferir que um jogo que alcança o limiar de diversão é um jogo divertido. Esse será o critério adotado para avaliar se o jogo desenvolvido neste trabalho é um produto de entretenimento.

Finalmente, os fatores de aumento da diversão, último nível do modelo de diversão, são os elementos diferenciais dos jogos extremamente divertidos (jogos com as *reviews* mais positivas do estudo). Para Wang, Shen e Ritterfield (2009), estes fatores estão relacionados a elementos de *game design* superiores (complexidade e diversidade, trama, mecânicas e gratificação), qualidade superior na apresentação estética (ambientes audiovisuais altamente sofisticados e imersivos), e principalmente à narrativa e à interação social (durante e após o jogo) - responsáveis por separar os jogos divertidos dos superdivertidos.

A partir desta classificação, será possível avaliar com maior propriedade se o jogo desenvolvido pode ser considerado um produto de entretenimento - ou seja, se ele atinge o limiar de diversão. No próximo capítulo, serão apresentados os trabalhos relacionados.

[12] "Playability threshold, enjoyability threshold and super fun-boosting factors".

[13] "These things satisfy our innate human desires for discovery and problem solving and create genuine feelings of pleasure"

2 TRABALHOS RELACIONADOS

Além de apresentar um breve referencial teórico sobre os *serious games* e a aprendizagem em jogos sérios, este trabalho de conclusão de curso tem como objetivo específico pesquisar por trabalhos que tenham abordado algum aspecto do ensino de lógica de programação através dos jogos eletrônicos. Inicialmente, acreditava-se que não havia nenhum jogo direcionado exclusivamente ao ensino de lógica de programação - mais especificamente, ao ensino de programação para crianças.

Todavia, verificou-se que há pelo menos dois jogos eletrônicos com esse intuito: *CodeCombat*, um RPG que coloca o aluno no papel de um heroi em uma jornada pelo aprendizado; e *Code.org*, um jogo com dezenas de exercícios e muito material relacionado ao ensino de lógica de programação. Apesar de possuírem características muito diferentes um do outro, ambos serviram como inspiração para a elaboração do jogo proposto neste trabalho. No decorrer deste capítulo, serão apresentadas as características principais destes jogos.

2.1 *CodeCombat*

O *CodeCombat* é um jogo eletrônico do gênero *RPG* (*Role-playing game*) de multijogador (*multiplayer role-playing game*) projetado para ensinar programação para estudantes com idade superior a 9 anos, sem necessidade de experiência de programação prévia. Uma maneira simples de explicar do que se trata o *CodeCombat* é utilizar as palavras da própria equipe de desenvolvimento:

> Se queres aprender a programar, não precisas de aulas. Precisas é de escrever muito código e divertires-te enquanto o fazes. [..] É sobre isso que é a programação. Tem de ser divertida. Não divertida do género 'yay, uma medalha' mas divertida do género 'NÃO MÃE, TENHO DE ACABAR O NÍVEL!' É por isso que o CodeCombat é um jogo multijogador e não um jogo que não passa de um curso com lições. Nós não vamos parar enquanto não puderes parar–mas desta vez, isso é uma coisa boa. [..] Se vais ficar viciado em algum jogo, vicia-te neste e torna-te num dos feiticeiros da idade da tecnologia. (CODECOMBAT, 2015)

Por meio deste manifesto, os desenvolvedores de *CodeCombat* expressam tudo o que o jogo tem a oferecer: uma maneira extremamente divertida de aprender a programar. Ao mesmo tempo, ao afirmar que o *CodeCombat* não é um "jogo que não

passa de um curso com lições", percebe-se que há uma tentativa de se distanciar do rótulo estereotípico dos *serious games*, principalmente do subgênero *edutainment*.

Por isso, para cumprir a premissa de ser diferente dos demais jogos que ensinam a programar, *CodeCombat* conta com vários elementos de jogos comerciais, que vão desde personagens com poderes especiais e mundos para explorar até recompensas e itens colecionáveis. Desta forma, o jogador vivencia a jornada de um heroi, coletando tesouros, derrotando inimigos e resolvendo problemas ao longo do jogo, sendo que todas as ações executadas pelo personagem são comandadas através de uma linguagem de programação - a sua escolha entre *Python*, *Javascript*, *CoffeeScript*, *Clojure* e *Lua*.

Durante o jogo, o jogador enfrenta diversos desafios de lógica, dispondo apenas de comandos específicos para cumprir seus objetivos. Em um dos níveis, por exemplo, o personagem deve coletar todas as gemas espalhadas pelo cenário, além de derrotar o inimigo que as vigia. Para conseguir realizar esta tarefa, o jogador deve mover seu personagem pelo mapa de modo a coletar as gemas e comandar um ataque contra o inimigo. Traduzindo para a linguagem do jogo, isso significa que o jogador deve ordenar seu personagem com comandos de movimentação (*move up*, *move right*, *move left* e *move down*) e de ataque (*attack*), descrevendo um algoritmo que resolve o problema.

Como podemos ver na Figura 2.1, uma fase do jogo *CodeCombat* possui uma interface do usuário com os seguintes elementos:

Figura 2.1: Componentes da interface gráfica do jogo *CodeCombat*

1. Cenário: exibe o cenário (salas, cavernas, florestas, etc.), o personagem, as co-

ordenadas (pontos em vermelho) nas quais o personagem pode se movimentar, os inimigos e os objetos;
2. Objetivos: exibe a lista de objetivos da fase atual, indicando quais objetivos ainda não foram concluídos;
3. Status do personagem: por se tratar de um jogo RPG, o personagem possui características como força, velocidade, resistência e pontos de vida. Este indicador aponta quantos pontos de vida restam ao personagem.
4. Lista de comandos: exibe os comandos disponíveis ao personagem durante o nível. Comandos novos são liberados juntamente com equipamentos para o personagem (p. ex: o item "botas" dá acesso aos métodos andar para frente, para trás, à esquerda e à direita);
5. Área de trabalho: painel no qual o jogador escreve as resoluções (algoritmos) para os problemas de cada nível.

Ainda no que diz respeito aos elementos de jogos, percebeu-se que não há narrativa por trás da jornada do personagem, ou seja, a motivação para que o personagem inicie sua missão é desconhecida. Considerando que o jogo é um RPG - onde o mundo no qual a história acontece é muito importante - essa ausência pode ser interpretada como um ponto negativo no fator diversão.

Apesar disso, há muito mais fatores positivos do que negativos no jogo *CodeCombat*. Além de centenas de níveis diferentes para explorar, o jogo oferece material exclusivo aos professores, possibilitando a criação de turmas, a avaliação e o monitoramento do progresso de cada um de seus alunos. Outro aspecto positivo do *CodeCombat* é o suporte a vários idiomas, incluindo português de Portugal. Entretanto, muitas de suas funcionalidades não são gratuitas, exigindo cadastro por parte do aluno ou professor. Dentre as funcionalidades pagas destacam-se os níveis extras, os clãs privados, novos personagens e as partidas *multiplayer*. No que tange à aprendizagem, o conteúdo ensinado em *CodeCombat* se divide em quatro núcleos (mundos do jogo): *Kithgard Dungeon*, no qual são estudados os conceitos iniciais, como sintaxe básica, utilização de variáveis e laços de repetição; *Backwoods Forest*, no qual são introduzidos os algoritmos condicionais e operadores relacionais lógicos; *Sarven Desert*, onde o conceito de vetor (array) é introduzido; e *Cloudrip Mountain*, no qual são apresentadas técnicas avançadas, como funções e desenho de formas. A Tabela 2.1 descreve todos os conceitos abordados no decorrer do jogo:

Finalmente, é importante mencionar que diversos elementos de *CodeCombat* serviram como inspiração para a idealização do jogo proposto neste trabalho de conclusão, principalmente no que diz respeito ao fator diversão. Afinal, assim como no *CodeCombat*, o objetivo deste trabalho é desenvolver um produto de entretenimento, não um "jogo que não passa de um curso com lições".

Tabela 2.1: Lista de conceitos abordados no jogo *CodeCombat*
Fonte: http://br.codecombat.com/teachers

Kithgard Dungeon	**Backwoods Forest**	**Sarven Desert**	**Cloudrip Mountain**
Sintaxe básica	Comando SE	Aritmética	Objetos Literais
Métodos	Operadores lógicos	Laços WHILE	Laços FOR
Parâmetros	Propriedades de Objetos	Comando BREAK	Funções
Strings	Leitura de dados	Vetores	Desenhos
Laços de repetição		Comparação de strings	Módulo
Variáveis		Encontrar Min./Max.	

2.2 Code.org

Com uma abordagem bastante diferente do *CodeCombat*, o *Code.org*[1], uma iniciativa da *Google*, *Microsoft*, *Facebook* e *Twitter*, oferece aos usuários uma plataforma de aprendizagem de algoritmos interativa, o *CodeStudio*. Em um primeiro contato com o software, pessoas familiarizadas com o *Scratch*[2] vão notar várias semelhanças com o *Code.org*, uma vez que ambas plataformas permitem criar algoritmos por meio de blocos visuais de código. Porém, apesar de se assemelhar muito à ferramenta do MIT, o *CodeStudio* traz funcionalidades que o *Scratch* não oferece, como cursos, listas de exercícios agrupados por conteúdo, nível de dificuldade e faixa etária, vídeos e até mesmo ementas dedicadas aos professores.

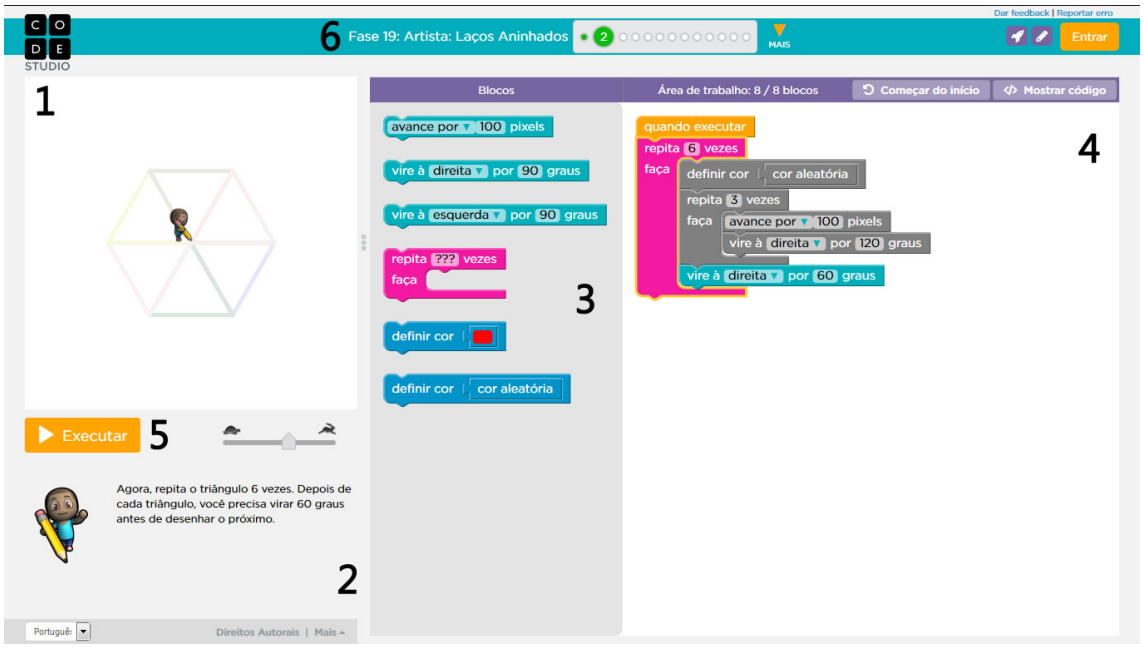

Figura 2.2: Componentes da interface gráfica do jogo *Code.org*

Como podemos ver na Figura 2.2, a interface gráfica da plataforma *CodeStudio*

[1]https://code.org/
[2]Ferramenta de ensino de programação desenvolvida pelo MIT.

possui suporte à língua portuguesa e é composta pelos seguintes elementos:

1. Quadro de visualização: lugar no qual os resultados da execução dos programas desenvolvidos são exibidos;
2. Descrição da atividade: exibe dicas e informações sobre a atividade que deve ser realizada;
3. Blocos: lista de blocos de código que podem ser utilizados para resolver a atividade proposta;
4. Área de trabalho/código: área na qual o aluno deve montar o algoritmo com os blocos de código disponíveis;
5. Controles da cena: elementos da interface gráfica do usuário que permitem a execução do código, bem como o controle da velocidade em que a resolução do problema é apresentada;
6. Barra de progresso: indica em qual fase do curso o aluno se encontra e quanto ainda resta para a sua conclusão.

No que diz respeito ao ensino, um dos cursos mais completos oferecidos pela plataforma é um curso de lógica de programação, o *Course2*. De acordo com a descrição disponível no site da plataforma, "neste curso, os alunos vão criar programas para resolver problemas e desenvolver jogos ou histórias interativas que eles podem compartilhar" (CODE.ORG, 2015).

O currículo do *Course2*, demonstrado na Tabela 2.2, foi elaborado para ensinar atividades introdutórias de lógica de programação, compreendendo desde a definição do que é um algoritmo até o estudo de algoritmos sequenciais, iterativos e condicionais. O curso é recomendado para alunos do 2° ao 5° ano, sem experiência prévia em programação.

Além disso, para ganhar a atenção das crianças e adolescentes, o Code.org utiliza personagens de jogos e desenhos animados famosos como *Angry Birds*, *Plants vs. Zombies*, *Minecraft* e *Frozen* durante os tutoriais e exercícios. No exemplo acima, os termos "abelha", "artista" e "labirinto" indicam quais personagens serão utilizados no decorrer de cada atividade. Por meio destas atividades, adultos, adolescentes e crianças aprendem desde um comando simples como desenhar uma linha reta na tela, até algoritmos um pouco mais elaborados, como desenhar figuras geométricas, nos quais são utilizados *loops* de repetição e várias instruções condicionais.

Por último, é importante ressaltar que, com base nos conceitos discutidos previamente no referencial teórico deste trabalho, o *Code.org* se caracteriza mais como um objeto de aprendizagem hipermidiático do que como um game propriamente dito, ou seja, pode-se dizer que o *Code.org* é uma plataforma de *edutainment*. Apesar

[3]A letra U ao lado da descrição é oriunda do termo em inglês *unplugged*, indicando que a atividade deve ser realizada sem a utilização do computador, ou seja, *offline*.

Tabela 2.2: Lista de conceitos abordados no jogo *Code.org*

Lição	**Descrição**
1	Programação em papel quadriculado [U][3]
2	Algoritmos da vida real: aviões de papel [U]
3	Labirinto: sequência
4	Artista: sequência
5	Repetindo
6	Labirinto: laços
7	Artista: laços
8	Abelha: laços
9	Programação por revezamento [U]
10	Abelha: depuração
11	Artista: depuração
12	Condições [U]
13	Abelha: condicionais
14	Pulseiras binárias
15	O grande evento
16	*Flappy bird*
17	Laboratório: crie uma história
18	Seu rastro digital [U]
19	Artista: laços aninhados

disso, *Code.org* é uma excelente ferramenta gratuita de ensino e possui diversos elementos interessantes, como as dicas e sugestões, os blocos de código e os exercícios de lógica de programação, que serviram de inspiração para o desenvolvimento do game proposto neste trabalho.

No próximo capítulo, será apresentado o *framework* DPE, cuja metodologia foi seguida durante o desenvolvimento do jogo proposto neste trabalho de conclusão.

3 METODOLOGIA

O jogo desenvolvido neste trabalho de conclusão é um *serious game*. Desta forma, considerando o fato de que o design de um *serious game* envolve desafios diferentes do *game design* de um jogo de entretenimento (WINN; HEETER, 2007), foi necessário procurar e estudar uma metodologia capaz de suprir as necessidades específicas de desenvolvimento do jogo proposto.

Conforme Hunick, LeBlanc e Zubek (2004), o *framework MDA* (*Mechanics, dynamics and aesthetics*)[1] mostrou-se uma abordagem útil no design e análise de jogos. Apesar disso, esse método não considera especificamente nenhum aspecto além da jogabilidade, não englobando as características únicas dos *serious games*. Por conseguinte, Winn (2008) sugere a utilização do *framework* DPE, uma expansão do MDA voltada para os *serious games*.

O *framework* DPE (*design*, *play* and *experience* ou projetar, jogar e experienciar), proposto por Brian Winn, é uma expansão do *framework* MDA que fornece linguagem, metodologia e um processo para o design de *serious games* (WINN, 2008). Em outras palavras, o *framework* fornece uma estrutura organizacional e um processo formal para o desenvolvimento de jogos sérios. Para Winn (Ibid.), a utilização deste *framework* pode diminuir muitos dos problemas encontrados em metodologias improvisadas no desenvolvimento de *serious games*.

O nome DPE deriva das ações realizadas durante o ciclo de desenvolvimento: o *game designer* projeta o jogo (*design*); os jogadores jogam o jogo (*play*), resultando na experiência do jogador (*experience*). Desta forma, para um game design eficaz, o game designer deve estabelecer os objetivos da experiência resultante (Ibid.). Após definir a experiência que o jogo deve proporcionar aos jogadores, é necessário elaborar mecânicas capazes de proporcioná-la - processo que envolve design, prototipação e muitos testes.

Como podemos ver na Figura 3.1, o *framework* DPE expandido ilustra os subcomponentes do design de serious games, incluindo as camadas de aprendizagem, narrativa, *gameplay* e experiência do usuário.

[1] O framework MDA (mecânicas, dinâmicas e estética ou mechanics, dynamics and aesthetics) foi projetado e ensinado por Marc LeBlanc para esclarecer e fortalecer os processos iterativos dos desenvolvedores, estudiosos e pesquisadores, tornando mais fácil para todas as partes envolvidas decompor, estudar e projetar jogos e artefatos de jogos (HUNICKE; LEBLANC; ZUBEK, 2004).

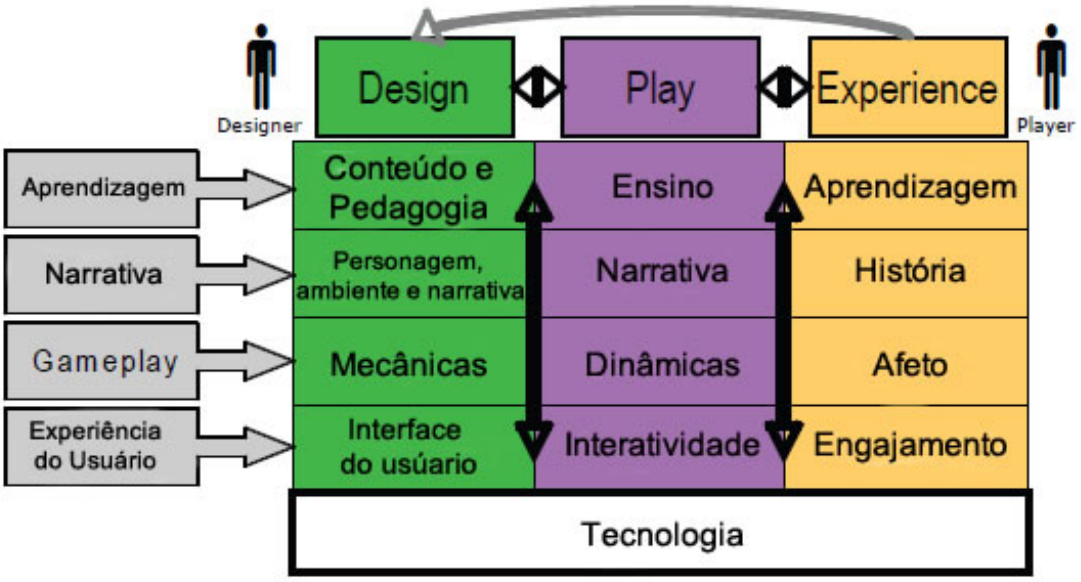

Figura 3.1: Estrutura expandida do framework DPE (WINN, 2008, tradução nossa)

Nas próximas sub-seções, cada uma destas camadas será discutida individualmente. É importante ressaltar que o uso do termo designer poderá se referir tanto ao game designer quanto aos especialistas de conteúdo, artistas ou pedagogos.

3.1 Camada de aprendizagem

Na camada de aprendizagem, o game designer "elabora o conteúdo e a pedagogia, os quais resultam em ensino quando o jogador interage com o jogo" (WINN, 2008, pg. 1015, tradução nossa[2]). Consequentemente, isto leva a um conjunto de resultados de aprendizagem derivados da experiência geral. Desta forma, o designer do jogo deve definir os objetivos da aprendizagem para que, posteriormente, possa elaborar os conteúdos para atender a esses objetivos. Este processo se assemelha muito com a construção do currículo de um curso, por exemplo.

Winn sugere ainda que o estudo da Taxonomia de Bloom (BLOOM, 1971) é útil para a elaboração dos resultados de aprendizagem no design de *serious games*. Para esta finalidade, todavia, este trabalho utiliza o conceito de Zona de Desenvolvimento Proximal de Vygostky. Apesar disso, independentemente do teórico escolhido, é importante definir os objetivos de aprendizagem desde o início do projeto, pois eles formarão a base para a elaboração do conteúdo e da pedagogia, bem como a base para a avaliação da eficácia do aprendizado do jogo (Ibid.).

[2] "[...] designs the content and pedagogy, which results in teaching when the player plays the game."

3.2 Camada de narrativa

Conforme Rouse (2001 apud WINN, 2008), há duas perspectivas na narrativa de jogos: a história do designer e a história do jogador. A partir deste ponto de vista, a história do game designer é a narrativa elaborada para ambientar o mundo do jogo, atribuir propósito à jornada dos personagens, fornecer motivação e transmitir conteúdo ao jogador. A experiência resultante da história criada pelo game designer combinada às escolhas e interações do jogador durante o jogo constroem a história do jogador.

Geralmente, alguns gêneros de jogos - como aventura e RPG - costumam possuir narrativas bem elaboradas, ao passo que outros, como puzzles ou esportes, quase não possuem história. Entretanto, todos os jogos têm uma história do jogador, que no mínimo reflete a história dos desafios encontrados pelo jogador e como eles foram resolvidos (Ibid.).

Apesar disso, o game designer deve ser cuidadoso ao criar a história para um *serious game*, pois a narrativa deve ser coerente com os resultados de aprendizagem esperados. Winn (2008) argumenta que os resultados de aprendizagem (*learning outcomes*) costumam complicar a criação de histórias para serious game. Por exemplo, em uma situação hipotética, na um game designer está desenvolvendo um *serious game* para ensinar História, qual porcentagem da narrativa pode diferir dos fatos históricos sem prejudicar o cumprimento de seus objetivos? Por isso, cada decisão na elaboração da narrativa deve ser influenciada pelos resultados de aprendizagem desejados.

3.3 Camada de *gameplay*

É a camada que define o que o jogador consegue fazer em um jogo: quais escolhas ele poderá fazer e que ramificações estas escolhas terão durante o resto do jogo (ADAMS; ROLLINGS; 2007). A camada de *gameplay* se divide em: mecânicas (*mechanics*), dinâmicas (*dynamics*) e afetos (*affects*).

Conforme Schell (2008), mecânicas são "procedimentos e regras do seu jogo. Mecânicas descrevem o objetivo do seu jogo, como os jogadores podem e como não podem tentar alcançá-lo e o que acontece quando eles tentam". Ou seja, são as regras que definem o funcionamento do mundo do jogo, o que os jogadores podem fazer, quais desafios enfrentarão e quais são seus objetivos.

As dinâmicas, por sua vez, são o "comportamento resultante quando as regras são instanciadas ao longo do tempo a partir da influência das interações do jogador" (WINN, 2008, pg. 16, tradução nossa[3]). Por último, as experiências resultantes,

[3] "The dynamics are the resulting behavior when the rules are instantiated over time with the influence of the player's interactions."

ou emoções derivadas do jogador, são os afetos. Nesta camada, o papel do game designer é definir os objetivos afetivos de seu jogo e de que forma deseja alcançá-los (mecânicas e, consequentemente, dinâmicas). A única forma de determinar se as mecânicas realmente atingem os objetivos afetivos é o *playtesting* (teste de jogabilidade). O game designer pode utilizar as informações obtidas no *playtesting* para modificar as mecânicas de acordo com os objetivos afetivos. Este processo é conhecido como balanceamento do jogo (Ibid.). Normalmente, várias iterações de projeto, prototipagem, *playtesting* e revisão são necessárias para balancear um jogo.

Conforme Winn (2008), uma forma comum de equilibrar o jogo é balancear o nível de dificuldade, de modo a alcançar o estado de Fluxo. Outra forma de equilibrar o *gameplay* é alterar a frequência na qual recompensas são dadas ao jogador. Winn (2008) aponta que é comum balancear a frequência das recompensas conforme o nível de dificuldade dos desafios, oferecendo recompensas melhores ao jogador nas partes mais desafiadoras do jogo. De acordo com Rabin (2011), é uma boa prática manter o jogador em um *loop* de tarefas e recompensas no qual "ao fazerem alguma coisa, conseguem algo", mas deve-se ter cuidado para não oferecer recompensas demais, nem em menor quantidade do que se espera.

Outro aspecto que demanda equilíbrio é a progressão do jogo (WINN, 2008). No início do jogo, o jogador pode se sentir sobrecarregado com a grande quantidade de opções enquanto ainda está aprendendo a jogar. Por isso, novos objetivos e mecânicas devem ser introduzidas aos poucos, para que o jogador consiga se acostumar aos novos elementos do jogo - o que vêm ao encontro do conceito de ZDP. Desta forma, o jogador terá tempo para adquirir, praticar e dominar novas habilidades para completar novos objetivos. O processo então se repete, com base nas competências anteriormente introduzidas (Ibid.).

Finalmente, Rolling e Adams (2003 apud WINN, 2008) afirmam que um jogo (de entretenimento) balanceado é um jogo no qual o principal fator determinante para o sucesso do jogador é o seu nível de habilidade. Winn (2008), por sua vez, acrescenta que, embora isso seja verdade para os *serious games* na maioria das vezes, balancear um *serious game* normalmente envolve fatores extras, principalmente relacionados aos resultados de aprendizagem.

3.4 Camada de experiência do usuário

A camada de experiência do usúario é a mais visível a partir da perspectiva do jogador e está relacionada principalmente com a interface, a interatividade e o engajamento do jogador. Afinal, é por meio dela que o game design se manifesta, pois ela engloba todos os aspectos visíveis do jogo, desde o que o usuário vê, ouve e interage até a forma como essa interação acontece (controles do jogo). No que diz

respeito aos jogos sérios, o propósito da interface do usuário é "criar um veículo para a realização dos resultados desejados" (WINN, 2008, pg. 1018, tradução nossa[4]).

No que tange a camada de experiência do usuário, o objetivo do game designer é desenvolver um jogo capaz de submergir o jogador em seu mundo e envolvê-lo na experiência de jogo. Portanto, para alcançar tal objetivo, é necessário desenvolver uma interface de usuário transparente, ou seja, criar uma experiência na qual o jogador não precisa focar a sua atenção em como jogar o jogo (quais controles usar), dedicando-se exclusivamente em vivenciar a história e a experiência de aprendizagem.

3.5 Camada de tecnologia

A tecnologia escolhida para o desenvolvimento do jogo pode ter um grande impacto no design de um *serious game* (WINN, 2008). Dentre as camadas descritas previamente, a que mais depende da tecnologia é a camada de experiência do usuário. Afinal, embora um game designer possa elaborar um jogo e desenvolver um protótipo de baixa fidelidade com o intuito de fazer um *playtesting* rápido, por mais que essa versão simplificada ajude a avaliar a eficácia do projeto, sua interface será muito diferente da versão final baseada em computador. Logo, a experiência do usuário será diferente.

Além disso, certos elementos dos jogos só podem ser realizados dependendo da tecnologia na qual o jogo é construído. Por exemplo, mecânicas de jogo que exigem óculos de RV (realidade virtual) não funcionariam como esperado em um protótipo de papel. Este tipo de mecanismo requer um nível muito mais elevado de sofisticação na tecnologia do jogo e, consequentemente, tende a exigir mais recursos para implementar.

Finalmente, a tecnologia pode tanto facilitar quanto limitar o projeto de um *serious game*. Logo, as capacidades e limitações da tecnologia e os recursos necessários para implementá-la podem influenciar consideravelmente o resultado final do jogo, e devem ser levadas em conta durante o processo de design (Ibid.).

No próximo capítulo, será apresentada a proposta de solução deste trabalho, na qual a metodologia do *framework* DPE foi aplicada.

[4] "[...] the purpose of the user interface is also to create a vehicle to realize the desired serious outcomes."

4 DESENVOLVIMENTO

Considerando a necessidade de desenvolvimento de um *game* para uma rede social gamificada, mencionada na introdução deste trabalho, foi decidido que o jogo desenvolvido seria um *serious game* para ensinar lógica de programação para crianças. O processo de desenvolvimento deste jogo baseou-se na metodologia proposta pelo *framework DPE* (*Design, play and experience*), apresentado no capítulo anterior. Por isso, ao longo deste capítulo serão discutidas as definições do *serious game* do ponto de vista de cada camada deste *framework*.

4.1 Camada de aprendizagem

De acordo com o *framework* DPE, é muito importante definir os objetivos da aprendizagem no início do projeto. Do ponto de vista educacional, o objetivo de *"Alice e o Mistério dos Algoritmos"* é ensinar os princípios básicos de lógica de programação aos jogadores, principalmente no que diz respeito à disciplina de algoritmos. Mais especificamente, é fazer com que, por meio do jogo, os jogadores desenvolvam um entendimento básico sobre um algoritmo, aprendendo a construir um conjunto de instruções para realizar tarefas simples, bem como a utilizar os princípios básicos dos algoritmos para resolver problemas.

Levando isso em consideração, inicialmente, definiu-se que os seguintes conteúdos seriam abordados ao longo do jogo: algoritmos sequenciais, iterativos e condicionais; e operadores lógicos E e OU. No entanto, devido à complexidade do tema e a relativamente curta duração do jogo, optou-se por introduzir apenas o conteúdo relacionado aos algoritmos sequenciais. De qualquer forma, foi necessário definir o formato e a linguagem dos problemas que seriam elaborados para cada conteúdo escolhido, ou seja, a forma de apresentar o conteúdo que o jogo iria se propor a ensinar.

4.1.1 Por que *puzzles*?

Nas primeiras dinâmicas do processo criativo, surgiu a ideia de abordar o elemento de lógica de programação por meio de um personagem robô ajudante. Nesse cenário, o personagem controlado pelo jogador não conseguiria avançar sozinho em determinados pontos do trajeto e teria que solicitar ajuda de outro personagem, o robô ajudante. Porém, por definição, o robô não executaria nenhuma ação sem a intervenção do jogador. Para ajudar o personagem, o jogador precisaria dar-lhe

instruções, construindo algoritmos para resolver problemas. Consequentemente, o ensino de lógica de programação ocorreria por meio dessa interação entre o personagem ajudante e o jogador.

No entanto, depois de algumas discussões sobre o assunto, chegou-se ao consenso de que essa mecânica poderia ser muito maçante e complexa para o jogador, pois ele teria que analisar e identificar os desafios de cada cenário para poder programar seu aliado - o que não necessariamente é ruim, mas não parece promissor em um jogo de plataforma. Além disso, a mecânica dificultaria muito o processo de *level design*, pois todos os níveis teriam que ser pensados em torno do robô ajudante.

Sendo assim, foi necessário elaborar uma nova mecânica para ensinar lógica de programação, e a ideia que surgiu foram os *puzzles* (quebra-cabeças). "*Puzzles* são problemas" (RABIN, 2011). Scott Kim (2008) acrescenta que *puzzles* são problemas que possuem uma resposta correta e são divertidos de se resolver. Logo, justamente por serem divertidos e desafiadores, costumam ser utilizados com frequência em *video games*.

Em *"Alice e o Mistério dos Algoritmos"*, os quebra-cabeças têm um papel muito importante, pois o jogador deve resolvê-los para avançar de nível. Os p*uzzles* desempenham um papel ainda mais importante se considerarmos seu aspecto avaliativo: se o jogador conseguir resolvê-los, significa que conseguiu aprender o conteúdo que o jogo se propõe a ensinar.

4.1.2 *Puzzles* no jogo

No universo do jogo, os *puzzles* são chamados de enigmas. No decorrer de cada nível do jogo, o jogador encontrará passagens bloqueadas e baús do tesouro que só podem ser liberados por meio da resolução de um enigma. A Figura 4.1 é uma captura de tela *in-game* que ilustra uma dessas situações:

Figura 4.1: Plataformas quebradas bloqueando o caminho do personagem

Para avançar, o jogador deve interagir com as máquinas de desenho (pontos de interrogação) existentes no cenário e resolver os enigmas que aparecerem. Essencialmente, os *puzzles* do jogo consistem na reprodução de desenhos por meio de comandos recebidos via interação com os botões da máquina de desenho. Em outras palavras, desvendar um enigma significa descrever os traços de uma figura por meio de um algoritmo.

4.1.3 Formato e linguagem

Inicialmente, é importante ressaltar que o formato no qual os *puzzles* são apresentados ao jogador foi inspirado na atividade *Graph Paper Programming* (programação em papel quadriculado), utilizada originalmente no Code.org, um dos trabalhos estudados abordados no capítulo 2. Além disso, faz-se necessário esclarecer que a disposição dos componentes da interface do usuário dos *puzzles* também foi influenciada pelas interfaces dos jogos CodeCombat e Code.org.

De qualquer forma, como mencionado anteriormente, para iniciar a resolução de um enigma, o jogador deve fazer o personagem interagir com um mecanismo. A interface desse mecanismo é composta por botões (comandos), uma tela de desenho em branco, uma tela de desenho preenchida (modelo a ser reproduzido) e por um painel que exibe a lista de comandos executados (histórico), como representado na Figura 4.2.

Figura 4.2: Interface do usuário dos *puzzles*

Para executar os comandos necessários para resolver os *puzzles* do jogo, o jogador tem à disposição um conjunto de botões que representam os seguintes comandos: mover para cima, para baixo, à direita e à esquerda e desenhar (item 1 da Figura 4.2). Além dos comandos usados na construção dos algoritmos, existem botões de ação para desfazer o último comando, para cancelar a resolução do enigma e para

executar a sequência de comandos formulada, realizando uma tentativa de solucionar o *puzzle*.

Desta forma, para reproduzir o desenho ilustrado no painel modelo (item 3 na Figura 4.2), o jogador poderia construir o seguinte algoritmo: seta para baixo, à direita e desenhar. É importante destacar que o jogador pode resolver os enigmas de maneiras diferentes. Por isso, a resposta atribuída neste exemplo não é a única correta.

Outra funcionalidade presente no mecanismo é o histórico de comandos (item 4 da Figura 4.2), onde são exibidos os comandos enviados pelo jogador. O *feedback* proporcionado pelo histórico é essencial para que o jogador não se confunda durante a elaboração do algoritmo. Quando o jogador considerar que o algoritmo está completo, ele deve pressionar o botão *"Confirmar"* (botão verde). Quando o botão *"Confirmar"* é clicado, a execução passo-a-passo do algoritmo é exibida e o jogador é informado sobre o resultado, conforme a Figura 4.3

Figura 4.3: *Feedback* positivo após a resolução de um *puzzle*

Ao deparar-se pela primeira vez com um enigma, é muito provável que o jogador não entenda o que deve ser feito. Por isso, no primeiro *puzzle* do jogo, o jogador precisa realizar um tutorial prático (*hands-on*) que explica o que são enigmas e como proceder para resolvê-los (item 5 da Figura 4.2).

4.1.4 Conteúdo e nível de dificuldade

Por serem a porta de entrada no universo da computação, os *puzzles* sobre algoritmos sequenciais são os primeiros enigmas a aparecer no jogo. Em outras palavras, inicialmente o jogador pode utilizar apenas instruções sequenciais para reproduzir um determinado desenho (ver Figura 4.4).

Figura 4.4: *Puzzles* do jogo contém apenas instruções sequenciais

Devido à curta duração do jogo (apenas quatro níveis), acredita-se que qualquer conteúdo posterior aos algoritmos sequenciais não perturbaria a ZDP (Zona de Desenvolvimento Proximal) do jogador, uma vez que não haveria tempo suficiente para internalizar o conhecimento adquirido de um nível para o outro. Desta forma, nesta versão inicial do jogo, o escopo dos objetivos de aprendizagem foi reduzido para contemplar apenas os algoritmos sequenciais.

No entanto, apesar de não estarem presentes nesta versão do jogo, é importante mencionar que os conteúdo relacionados aos algoritmos condicionais e iterativos foram elaborados. Em uma versão futura do jogo, se a sequência que costuma ser adotada nos cursos de algoritmos do âmbito acadêmico fosse seguida, o próximo tópico de ensino seria algoritmos condicionais. Entretanto, devido ao formato dos enigmas, assim como se percebe nos jogos *Code.org* e *CodeCombat*, não faria sentido apresentar condições em problemas de caráter sequencial. Por isso, faria mais sentido abordar o conteúdo de laços de repetição antes dos algoritmos condicionais.

Os algoritmos iterativos que seriam trabalhados em *"Alice e o Mistério dos Algoritmos"* possuiriam um número definido de iterações, pois entende-se que o jogador deve perceber quantas vezes uma instrução deve ser repetida para resolver um dado problema - ou no caso do *serious game*, um *puzzle*. Os quebra-cabeças de repetição implicariam na adição de um novo símbolo, indicando a estrutura de repetição e quantidade de vezes que a instrução deveria se repetir.

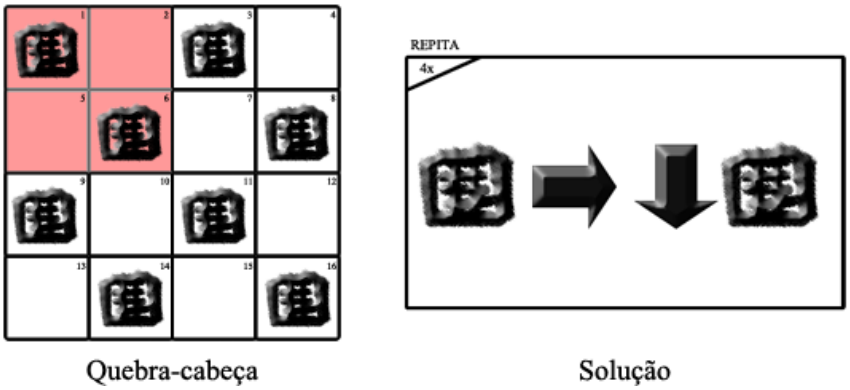

Figura 4.5: Solução para um *puzzle* de repetição 4x4

No exemplo da Figura 4.5, o desenho poderia ser completado repetindo as instruções indicadas por 4 vezes, sendo que nos *puzzles* que envolvem *loops*, a repetição da instrução ocorreria da esquerda para a direita. Utilizando esta mesma estrutura, foi possível elaborar *puzzles* para os algoritmos condicionais e, consequentemente, para as estruturas condicionais lógicas E e OU.

Em outro exemplo, representado na Figura 4.6, nota-se que o painel está preenchido com quadrados na metade superior e com círculos na metade inferior. Para reproduzir este desenho, além de utilizar um laço de repetição, o jogador precisaria usar um bloco condicional, cuja função seria executar uma determinada instrução dependendo da condição que estaria sendo avaliada. No exemplo acima, sempre que a condição $N^{o}\ QUADRADO <= 8$ fosse verdadeira, o programa desenharia um quadrado. Caso contrário, desenharia um círculo.

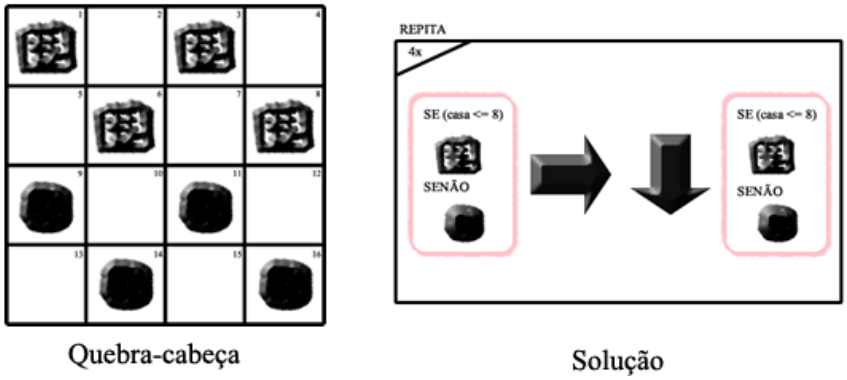

Figura 4.6: Solução para um *puzzle* 4x4 com estruturas de decisão

Evidentemente, sempre que desafiado a resolver um *puzzle*, o jogador seria informado sobre o tipo de estruturas que deveria utilizar e como utilizá-las. Para tanto, sempre que um novo conteúdo fosse introduzido no jogo, um novo tutorial *hands-on* que explicasse seu funcionamento deveria ser elaborado.

No entanto, independentemente de seu esforço, um dos maiores desafios para o *game designer* é fazer com que os jogadores permaneçam no Fluxo pelo maior tempo

possível. Paralelamente, para um *designer* de um *serious game*, o desafio é projetar os *puzzles* de seu jogo de modo a operar no nível de desenvolvimento potencial de seus jogadores, conseguindo perturbar sua ZDP.

Isso se deve principalmente ao fato de que não há fórmula que defina um ponto de equilíbrio entre a dificuldade de uma tarefa e as habilidades necessárias para realizá-la. Sendo assim, *a priori*, não há como estimar quantos *puzzles* sobre algoritmos sequenciais devem ser resolvidos para que o jogador tenha fundamentação para entender os algoritmos de iteração. Por isso, a proporção na qual a dificuldade aumenta deve ser testada e modificada por meio de *playtesting*, até que um valor considerado ideal seja encontrado.

4.1.5 Lapidando o conceito inicial

Desde que a primeira ideia do jogo foi concebida, ela tem sido modificada para entregar a melhor experiência possível de entretenimento e aprendizagem aos jogadores. Como se pode observar na Figura 4.7, a forma na qual os puzzles seriam representados na interface gráfica do jogo, desde a ideia inicial até o produto final, sofreu diversas alterações:

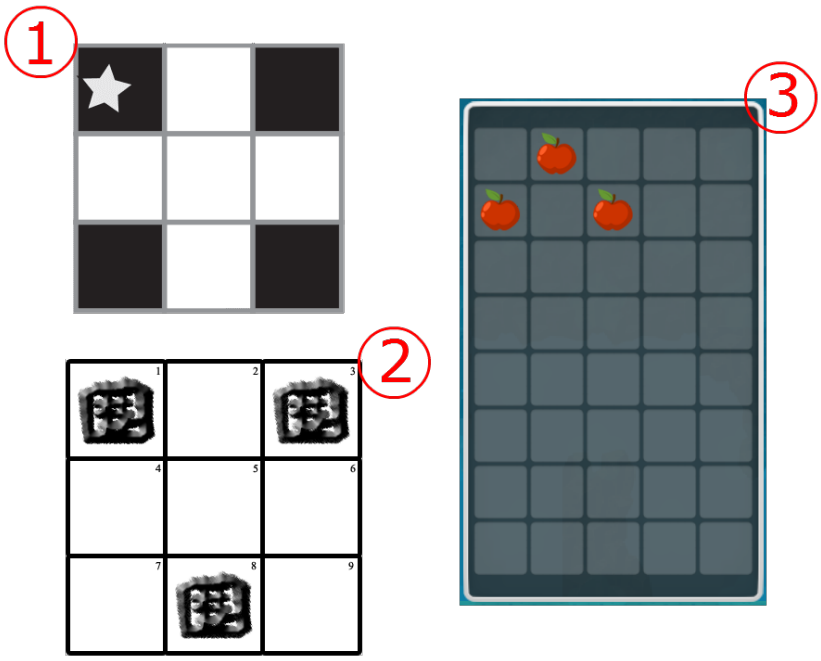

Figura 4.7: Evolução: ideia original, esboço intermediário e versão final

A Figura 4.7 descreve três matrizes: duas quadradas de 3 x 3 (n°1 e n°2) e uma retangular de 8 x 5 (matriz n°3). A matriz n°1 é a mesma proposta na atividade *"Graph Paper Programming"*, onde os elementos da matriz são representados por quadrados que devem ser preenchidos de acordo com o desenho proposto. A partir desse rascunho, também foi definido que, no início de cada desafio, o cursor estaria

posicionado no primeiro quadrado da primeira linha da matriz.

A matriz n°2 é uma evolução direta da n°1, sendo que a principal diferença entre os projetos das duas matrizes é que a segunda possui uma identificação numérica sequencial que varia entre 1 e N, onde N é o número de elementos da matriz. Essa numeração seria utilizada na construção de algoritmos iterativos e condicionais, mas acabou sendo retirada na versão final do produto, pois o escopo foi reduzido para contemplar apenas os algoritmos sequenciais.

Desta forma, com o intuito de possibilitar diferentes níveis de complexidade e diferentes formas para desenhar, um novo projeto foi elaborado: a matriz n°3. Além das diferenças de tamanho e da ausência da numeração, o novo modelo utiliza a figura de uma maçã para representar o preenchimento de um quadrado da matriz. Essa alteração foi realizada para que houvesse uma relação entre os desenhos dos enigmas e os itens que o jogador deve coletar em cada nível do jogo.

Outra alteração que afetou bastante as características do jogo está relacionada à forma de resolver enigmas. Inicialmente, a interação com os puzzles havia sido pensada de modo a exibir imediatamente o resultado de cada comando emitido pelo jogador, ou seja, todos os comandos enviados pelo jogador à máquina de desenhos (direções, desenhar ou desfazer) seriam exibidos em seguida na tela de desenho. A responsividade imediatista da interface dos *puzzles* fazia com que a ideia parecesse mais intuitiva do que a atual, assemelhando-se bastante com o ato de desenhar propriamente dito.

Entretanto, a premissa fundamental deste trabalho é ensinar lógica de programação, e não técnicas de desenho. Logo, a forma de interagir com os enigmas teve que ser alterada para emular com maior precisão o raciocínio a elaboração de um algoritmo: descrever uma sequência de passos que resolvem um determinado problema. Na próxima seção, serão apresentadas as definições da camada de narrativa do jogo.

4.2 Camada de narrativa

No que diz respeito à camada de narrativa do jogo, foram definidas a ambientação, a história e o *background* do personagem principal. Considerando que *"Alice e o Mistério dos Algoritmos"* é um *serious game*, qual seria uma boa temática para o jogo? Esta questão é muito subjetiva, pois se um determinado tema consegue atrair a atenção de alguns jogadores, não significa que despertará interesse em todos. Por isso, não havia como ter certeza que a escolha realizada foi correta antes de testar o jogo.

De qualquer forma, durante o processo criativo, decidiu-se que o jogo seria ambientado em um mundo mágico retratado a partir do ponto de vista de Alice, uma

jovem curiosa e aventureira. A história se inicia quando esta jovem recebe uma missão perigosa de seu avô: investigar a cidade proibida de Turren, um vilarejo distante e escondido nas florestas, em busca de uma cura para a doença que assola o condado onde vivem. De acordo com as lendas locais, Turren teria sido visitada por seres superiores de outro planeta. Até então, Alice acreditava que as lendas eram apenas histórias inventadas para assustar as crianças e mantê-las afastadas da floresta. Mas a partir deste momento, a jovem sentiu-se determinada a desvendar os mistérios por trás do vilarejo.

No jogo, a trama é contada por meio de uma breve introdução textual no primeiro nível, intitulado *"Minha primeira jornada"*. Além disso, há menções ao nome da personagem e ao seu avô na tela inicial (título) e em algumas instruções no decorrer dos primeiros dois níveis do jogo.

Evidentemente, a história do *serious game* ainda não foi terminada. Porém, considerando que esta é apenas uma versão de demonstração do game, não pareceu correto finalizar a narrativa da jornada da heroína. Além disso, apesar de um pouco abstratas, essas definições forneceram detalhes o suficiente para a criação das mecânicas e a elaboração do conteúdo gráfico e sonoro do jogo.

4.3 Camada de *gameplay*

De acordo com o *framework* DPE, a camada de *gameplay* define o que o jogador consegue fazer em um jogo. Para esse fim, foram definidos o gênero, as mecânicas, os modos e os objetivos do jogo. Primeiramente, serão discutidos os motivos da escolha do gênero do jogo. Por último, serão apresentadas as mecânicas, os modos e os objetivos.

4.3.1 O gênero do jogo

Na maioria das vezes, quando um *game designer* concebe a ideia de um jogo, ele procura arquitetar em sua mente a narrativa (história e ambientação), as mecânicas e o gênero do jogo. Desta forma, a definição do escopo se torna uma tarefa menos complicada, pois alguns limites são estabelecidos (um jogo simulador de futebol não possuirá carros de corrida, por exemplo). Por isso, antes de começar a expandir as ideias iniciais de *"Alice e o Mistério dos Algoritmos"*, definiu-se que o seu gênero será o de plataforma em duas dimensões.

Originalmente, os jogos de plataforma "envolvem a personagem correndo e pulando em um campo de jogo com visão lateral" (RABIN, 2011), ou seja, com posicionamento bidimensional da câmera. Dentre os diversos gêneros existentes, como aventura, ação, luta, *survival-horror*, corrida e esportes - apenas para mencionar alguns - o gênero de plataforma 2D foi escolhido pelos seguintes motivos:

1. Primeiramente, apesar de se tratar de um *serious game*, comprometendo-se a ensinar os princípios de lógica de programação, a premissa de desenvolvimento de *"Alice e o Mistério dos Algoritmos"* é a diversão. Em outras palavras, é fazer com que o jogador se sinta da mesma forma como se sentiria jogando *Super Mario World* ou *Sonic: the Hedgehog*, uma premissa parecida com a do jogo *CodeCombat*.
2. Em segundo lugar, por definição, jogos de plataforma demandam de seus jogadores habilidades como coordenação motora e raciocínio lógico, sem tornar os controles muito complexos - o que, de acordo com a percepção deste trabalho, é ideal em um jogo que tem como público-alvo crianças e adolescentes. Além disso, o nível de dificuldade das habilidades exigidas no decorrer do jogo pode ser projetado para aumentar progressivamente, fazendo com que o jogador esteja sempre na Zona de Desenvolvimento Proximal.
3. Por último, apesar de não ser uma constante, jogos de plataforma costumam ser relativamente mais simples de desenvolver do que jogos de outros gêneros, o que possibilitou que esse trabalho pudesse ser desenvolvido no prazo e com os recursos estabelecidos.

4.3.2 Mecânicas, modos e objetivos do jogo

Com o gênero e a temática de *"Alice e o Mistério dos Algoritmos"* definidos, foi possível elaborar as mecânicas, os modos e os objetivos do jogo. Essas características definem seu funcionamento geral, delimitando as ações que podem ser realizadas pelo jogador, bem como o que é necessário fazer para completar os objetivos.

Inicialmente, *"Alice e o Mistério dos Algoritmos"* conta com apenas um modo de jogador único (*singleplayer*), no qual a personagem enfrenta muitos perigos, além de resolver enigmas de lógica de programação para completar cada nível do jogo. Logo, o objetivo principal do jogo é finalizar todos os níveis para obter mais informações sobre o mistério que deu início à jornada da personagem.

Por ser um jogo de plataforma, *"Alice e o Mistério dos Algoritmos"* apresenta mecânicas de movimentação básica como locomoção e pulo. Essas ações são indispensáveis para os jogos do gênero, uma vez que o personagem controlado pelo jogador deve superar obstáculos e coletar itens pelo cenário a todo o momento (Figura 4.8).

Figura 4.8: Mecânicas de movimentação básica

Além da movimentação, outras mecânicas muito comuns em jogos de plataforma estão presentes em *"Alice e o Mistério dos Algoritmos"*:

- Obstáculos: servem o propósito de criar desafios interessantes para o jogador. Em outras palavras, são elementos responsáveis por impedir que os objetivos do jogo sejam realizados com facilidade. Para avançar nas fases do jogo, o jogador deve desviar de obstáculos como espinhos e penhascos e utilizar plataformas para alcançar áreas de difícil acesso (Figura 4.9).

Figura 4.9: Obstáculos: plataformas, espinhos e penhascos

- Eliminação da fase: o jogador tem um tempo limite pré-definido para realizar os percursos de cada nível. Caso ele não consiga terminar a tempo, o progresso da fase é perdido e o jogador perde uma vida. Outras situações que caracteri-

zam derrota/eliminação da fase são: cair em um penhasco (buraco sem fundo) e encostar-se em algum objeto mortífero (espinhos, entre outros).

- Pontuação e moedas: certas ações desempenhadas pelos jogadores - como coletar moedas, encontrar itens e resolver *puzzles* - geram pontos de bonificação. Estes elementos servem como estímulo de competição, motivando os jogadores a conseguir uma pontuação maior do que a de seus amigos.
- Itens escondidos e áreas secretas: alguns itens e moedas são mais difíceis de serem obtidos, pois estão escondidos em áreas secretas ou de difícil acesso (acessíveis apenas por meio do uso de técnicas mais avançadas de movimentação, como pular na parede). Desta forma, caso queira coletar tais itens, o jogador deverá perceber as sinalizações diferentes no cenário e descobrir a entrada das áreas secretas - normalmente, estão em árvores ou áreas de difícil acesso (Figura 4.10 e Figura 4.11).

Figura 4.10: Itens escondidos em árvores e áreas de difícil acesso

Figura 4.11: Área secreta contendo moedas e itens raros

Pelo que se pode evidenciar acima, as mecânicas do jogo *"Alice e o Mistério dos Algoritmos"* possuem grandes influências de jogos clássicos de plataforma 2D. Afinal, tão importante quanto criar é buscar inspiração em outros jogos bem sucedidos que apresentam características semelhantes.

Obviamente, por se tratar de um *serious game*, além das mecânicas apresentadas acima, *"Alice e o Mistério dos Algoritmos"* precisa ensinar o conteúdo propriamente dito. Por isso, o jogo conta com uma mecânica que o diferencia dos demais jogos de plataforma: *puzzles* de lógica de programação, explicados detalhadamente na seção da camada de aprendizagem. Na próxima seção, serão apresentadas as definições da camada de experiência do usuário do jogo.

4.4 Camada de experiência do usuário

Em se tratando da camada de experiência do usuário, os principais fatores de um jogo são a apresentação audiovisual, os controles e a interface gráfica. No que tange à apresentação visual, foi definido que o jogo utilizaria *sprites* 2D[1] de alta definição que combinam com o ambiente no qual a narrativa do jogo está inserida. Levando isso em consideração, os *sprites* do cenário e da personagem foram selecionados e adquiridos de um site de arte 2D independente, o *GameArt2D*[2].

Assim como no caso dos *sprites*, os efeitos sonoros e as trilhas da vitória e da tela inicial foram selecionados e adquiridos em repositórios da *internet*. A única exceção, no caso da apresentação sonora, foi a trilha principal, presente em todos os níveis do jogo. A composição dessa trilha foi solicitada a um artista com experiência na criação de áudio para jogos, pois nenhuma das trilhas encontradas parecia proporcionar a sensação que o game designer havia imaginado.

Figura 4.12: Indicadores de status e direção

[1] *Sprites* são imagens ou animações bidimensionais utilizados para compor uma cena maior.
[2] *Sprites* disponíveis em: *http://www.gameart2d.com*

Na Figura 4.12, ainda no que se refere à experiência de jogo planejada pelo *game designer*, é possível perceber que a temática de floresta - que permeia a história do jogo - está presente nos *sprites* de fundo, das rochas e do solo. Na mesma ilustração, nota-se também que há outro elemento importante para a experiência do usuário nos jogos: a interface gráfica. Ao longo de cada fase, o status do jogador pode ser verificado na parte superior da tela, onde são exibidos a quantidade de vidas e tempo restantes, além da pontuação. Indicadores de direção, perigo e suspeita também estão espalhados pelo cenário para dar senso de direção e objetivo ao jogador.

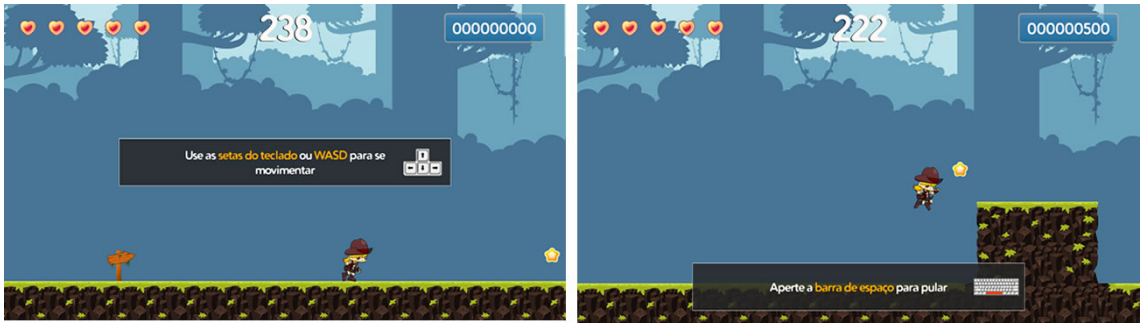

Figura 4.13: Instruções referentes aos controles do jogo

Além dos indicadores de status e direção, a interface gráfica é percebida nas instruções dadas aos jogadores. Nas duas capturas de tela ilustradas na Figura 4.13, é possível observar que o jogador está sendo instruído a utilizar as setas do teclado para controlar seu personagem: setas direcionais para a locomoção do personagem e a barra de espaços para o pulo. Na próxima seção, serão apresentados detalhes acerca da camada de tecnologia.

4.5 Camada de tecnologia

No que tange à tecnologia de desenvolvimento de jogos, um dos aspectos mais importantes a definir é a *engine* ou motor de jogos. *Engines* são plataformas de desenvolvimento para simplificar a produção de jogos eletrônicos (EBERLY, 2006). Geralmente, as *engines* incluem motores gráficos para renderização de gráficos 2D e 3D, motores de física para detecção de colisão e simulação de partículas, suporte a animação, sons, inteligência artificial, redes, suporte a *scripts* em diversas linguagens de programação e outros componentes (Ibid.).

Desta forma, a escolha da *engine* de jogos pode ser determinante no sucesso de um jogo, pois é a partir das funcionalidades - ou limitações - que ela possui que o jogo vai ser desenvolvido. Uma *engine* 2D não permite o desenvolvimento de um jogo de tiro 3D em primeira pessoa, por exemplo. Para o desenvolvimento do jogo proposto neste trabalho, a *engine* escolhida foi a Unity3D (UNITY3D, 2015).

Dentre os fatores que levaram à sua escolha, a Unity3D se destaca por ser uma fer-

ramenta gratuita e completa, suportando diversas plataformas (Ibid.). Além disso, a Unity fornece ampla documentação e é relativamente fácil de aprender (possui uma baixa curva de aprendizado), além de possuir uma comunidade de desenvolvimento extensa e ativa. E, apesar do nome, a Unity3D possui suporte completo para o desenvolvimento de jogos 2D - o que é perfeito para o caso do *"Alice e o Mistério dos Algoritmos"*, principalmente por ser compatível com o *software* Tiled (TILED, 2016), utilizado para elaboração e organização dos *sprites* 2D dos mapas do jogo.

No que diz respeito à programação, a Unity3D possui suporte às linguagens Javascript, C# e Boo (Python). Considerando que as funcionalidades disponíveis para cada linguagem são as mesmas, os fatores determinantes para a escolha da linguagem foram a familiaridade e o suporte nativo à programação orientada a objetos. Por isso, a linguagem escolhida foi C# (C Sharp). Neste aspecto, é importante destacar que, com exceção dos *scripts* de movimentação do personagem e de animação da interface gráfica, cujas bibliotecas foram adquiridas gratuitamente em repositórios de recursos para jogos, os *scripts* foram inteiramente desenvolvidos na linguagem C#.

Considerando a plataforma na qual a rede social gamificada funciona, o jogo *"Alice e o Mistério dos Algoritmos"* foi desenvolvido para computadores pessoais, além de contar com a possibilidade de ser exportado para *web* e dispositivos móveis. Por conseguinte, a arquitetura física do jogo pode ser descrita conforme a Figura 4.14:

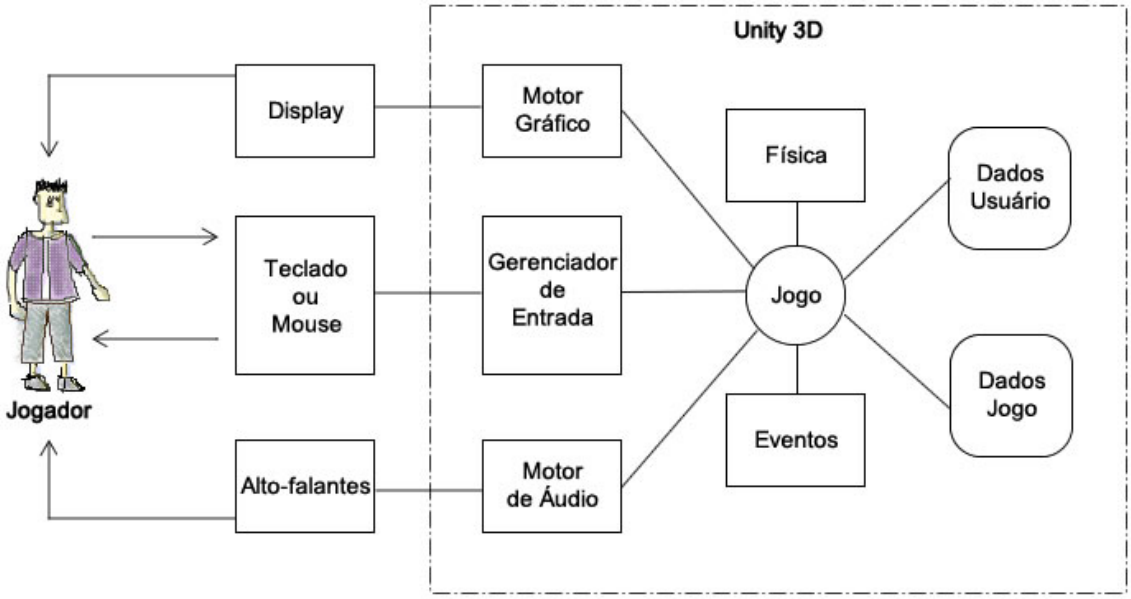

Figura 4.14: Arquitetura física do jogo

Como mostra a Figura 4.14, o jogador interage diretamente com os dispositivos de entrada (teclado e mouse), que enviam sinais ao gerenciador de entrada (gerenciamento de input nativo da Unity3D). Este, por sua vez, é consultado pelo

gerenciador do jogo (*scripts* que gerenciam o input) para que os comandos executados sejam reconhecidos. Por conseguinte, o gerenciador do jogo calcula a saída (*feedback*) utilizando os motores de física e os eventos do jogo e a retorna por meio dos dispositivos de saída (*display* e alto-falantes). Além disso, o progresso do jogador é persistido em um arquivo serializado no formato binário, o que possibilita que suas estatísticas sejam carregadas sempre que o jogo for iniciado.

Obviamente, em uma possível integração com a rede social gamificada, fatores como conexão cliente-servidor e armazenamento na nuvem devem ser levados em conta. Porém, como a rede social mencionada ainda não possui uma interface de integração (*API* (*Application Programming Interface*)), estes fatores foram desconsiderados durante a elaboração da estrutura de software proposta para este trabalho.

Nos próximos capítulos, serão apresentados os resultados do teste de jogabilidade realizado e as considerações finais deste trabalho de conclusão.

5 PLAYTESTING

Após a conclusão da etapa de implementação, realizou-se a última parte do ciclo de desenvolvimento do jogo *"Alice e o Mistério dos Algoritmos"*: planejar, aplicar e avaliar um teste de jogabilidade, mais conhecido como *playtesting* - um dos objetivos específicos deste trabalho de conclusão. O autor do livro *"The Art of Game Design: A Book of Lenses"*, Jesse Schell, define que o *playtesting* é um tipo de teste no qual o objetivo é reunir pessoas para jogar seu jogo e verificar se ele proporciona a experiência para a qual foi projetado (SCHELL, 2008, pg. 390).

Schell escreve ainda que o *playtesting* é uma oportunidade para o *game designer* ver seu jogo em ação e que é possível maximizar a qualidade de um teste de jogabilidade ao responder às seguintes perguntas: "Por quê estamos fazendo um teste de jogabilidade? Quem deveria participar? Onde deveríamos realizá-lo? O quê devemos procurar? Como conseguiremos a informação que precisamos?" (SCHELL, 2008, pg. 401, tradução nossa[1]).

Levando isso em consideração, o primeiro passo para planejar um bom teste de jogabilidade é responder o *porquê* de sua realização. Obviamente, pelo fato deste projeto possuir uma questão de pesquisa definida desde sua proposta inicial, o objetivo do *playtesting* é fornecer uma possível resposta para essa pergunta. Consequentemente, pode-se dizer que o teste de jogabilidade foi realizado para determinar se o *serious game* desenvolvido pode ser considerado um produto de entretenimento (divertido) ou não. Em outras palavras, o intuito era verificar se o jogo atinge o limiar de diversão estabelecido no modelo de três níveis proposto por Wang, Shen e Ritterfield (2009).

No entanto, Schell aponta que as perguntas que o *playtesting* deseja responder devem ser tão específicas quanto possível; perguntar "O meu jogo é divertido?" não é o suficiente (SCHELL, 2008, pg. 392). Em virtude disso, foi realizada uma pesquisa detalhada acerca da diversão em jogos digitais, a qual está disponível no referencial teórico deste trabalho. Nessa pesquisa, foram apresentados os 5 fatores principais da diversão em games, bem como um modelo de três níveis de diversão.

A partir desse modelo, é possível definir se um jogo é jogável (alcança o limiar de jogabilidade), divertido (alcança o limiar de diversão) ou excepcional (possui fatores de aumento da diversão). Deste modo, elaborou-se um questionário (APÊNDICE

[1] "Why are we doing a playtest? Who should be there? Where should we hold it? What will we look for? How will we get the information we need?

A) para identificar se o jogo possui os requisitos mínimos para ser considerável jogável e divertido. As questões foram direcionadas especificamente a cada elemento compreendido em cada nível do modelo de três níveis.

Após definir o *porquê*, foi necessário definir *quem* deveria participar do *playtesting*. Para evitar um *feedback* tendencioso, definiu-se que apenas pessoas que nunca tiveram contato com o jogo participariam como testadores. De acordo com Schell (Ibid.), a indústria refere-se a este tipo de testadores como *"tissue testers"*. A vantagem dessa prática é que as pessoas que nunca tiveram contato com o jogo percebem as coisas de um jeito diferente do que o *game designer* está acostumado, o que as torna muito importantes nos testes que buscam determinar questões de usabilidade, comunicação e apelo inicial (Ibid.).

Segundo Schell (Ibid.), a desvantagem em utilizar *tissue testers* é que

> "Jogos são geralmente jogados múltiplas vezes, durante várias sessões. Se você testar seu jogo apenas com *'tissue testers'*, você corre o risco de produzir um jogo que tem forte apelo à primeira-vista, mas que se torna maçante após múltiplas sessões" (SCHELL, 2008, pg. 394, tradução nossa[2]).

No entanto, esse risco não é tão relevante para o jogo *"Alice e o Mistério dos Algoritmos"*, uma vez que o gênero do jogo costuma dispor de uma vasta gama de níveis - onde novas mecânicas podem ser introduzidas conforme a dificuldade progride.

No que tange ao local de realização dos testes de jogabilidade, os possíveis locais podem ser o próprio estúdio de desenvolvimento, um laboratório específico para *playtesting*, um local público, a residência dos jogadores que testam o jogo e até mesmo a *internet*(Ibid.). O teste de jogabilidade do *serious game* foi realizado em um laboratório de informática da Universidade de Caxias do Sul, uma mescla entre local público e estúdio de desenvolvimento. O local foi escolhido por ser gratuito e porque acreditava-se que haveria muitos estudantes da área da computação disponíveis para testar o jogo.

Após definir *porque*, por *quem* e *onde* o teste de jogabilidade seria realizado, era necessário definir *o quê* se estava procurando. Neste aspecto, há dois tipos de coisas para observar: o que o *game designer* sabe que está procurando, que são as perguntas do questionário (por quê?); e o que o *game designer* não sabe que está procurando, como críticas, sugestões e reações dos jogadores durante o teste de jogabilidade.

Por último, foi necessário definir *como* seria obtida a informação necessária.

[2] "Games are generally played multiple times, over many sessions. If you only test your game with 'tissue testers', you run the risk of making a game that has strong first-time appeal, but gets boring after multiple plays."

Nesta etapa, decidiu-se que não seria dada nenhuma explicação prévia aos jogadores que participariam dos testes. A intenção era justamente deixar o jogo explicar-se por si só, principalmente por sua característica de ferramenta de apoio à aprendizagem. Desta forma, seria possível verificar se os jogadores conseguiriam aprender a jogar por conta própria e, caso não conseguissem, a informação passada verbalmente ao testador poderia ser adicionada ao tutorial do jogo.

Outro aspecto importante relacionado ao modo como as informações seriam obtidas no teste de jogabilidade são os dados coletados após a sessão de testes. Como mencionado anteriormente, um questionário (*survey*) foi elaborado justamente para essa finalidade. A *survey* foi aplicada individualmente a cada jogador assim que sua respectiva sessão de jogo terminava. Na próxima seção, serão apresentados os resultados do teste de jogabilidade.

5.1 Resultados do *playtesting*

Para facilitar a extração do resultado do *playtesting*, o questionário foi elaborado com perguntas direcionadas aos elementos específicos de cada nível do modelo de três níveis de diversão. Desta forma, definiu-se que as questões relacionadas à usabilidade, controle, desafio e apresentação visual seriam utilizadas para verificar se o jogo atingiria o limiar de jogabilidade, ao passo que as perguntas relacionadas à qualidade visual, apresentação de áudio, complexidade e diversidade, mecânicas, liberdade, níveis, equilíbrio do grau de dificuldade e gratificação seriam úteis para verificar se o jogo atingiria o limiar de diversão.

Além disso, considerou-se que as perguntas do questionário eram fáceis de quantificar, pois possuíam o formato de escala de cinco-pontos, onde 1 era a nota mais baixa, e 5 a nota mais alta. Para Schell (2008, pg. 400), uma *survey* costuma ser mais eficaz quando as pontuações são rotuladas, por isso as pontuações foram rotuladas conforme a natureza de cada questão (*ex: 1. Péssimo - 5. Ótimo; 1. Não gostei - 5. Adorei, etc.*). Levando isso em consideração, designou-se que 3,5 seria a nota mínima para atingir cada limiar do modelo de três níveis. A única exceção era a pergunta relacionada ao equilíbrio do grau de dificuldade, onde a melhor nota possível era 3, pois significaria que a dificuldade aumenta no ritmo ideal (nem lentamente, nem rapidamente).

A partir do momento em que o desenvolvimento do jogo foi finalizado, o questionário elaborado e os critérios definidos, foi possível colocar em prática o teste de jogabilidade. Conforme mencionado anteriormente, o local escolhido para o teste foi um laboratório de informática da Universidade de Caxias do Sul (sala 406 do bloco 71). A sessão de *playtesting* ocorreu no dia 23 de novembro de 2016, nos períodos vespertino e noturno, e contou com a participação de 19 *tissue testers*, todos alunos

dos cursos da área computação.

Os participantes foram convidados a jogar e receberam instruções para preencher o questionário após o término da sessão de jogo. Todos os *testers* jogaram *"Alice e o Mistério dos Algoritmos"* até o fim, ou seja, finalizaram todos os quatro níveis do jogo. Para contabilizar os resultados, as notas médias das perguntas foram agrupadas por fator de diversão. Posteriormente, as médias dos fatores de diversão foram agrupadas de acordo com sua finalidade: limiar de jogabilidade e limiar de diversão. Na Tabela 5.1, pode-se visualizar que a média dos fatores referentes ao limiar de jogabilidade é 4,15, enquanto a média referente ao limiar de diversão é 4,07. Por conseguinte, é correto afirmar que, tanto o limiar de jogabilidade, quanto o de diversão foram alcançados. Em outras palavras, os *testers* consideram que o jogo é divertido.

Tabela 5.1: Resultados do Teste de Jogabilidade

Nível	**Fator de Diversão**	**Média (escala 1 a 5)**
Limiar de Jogabilidade	Usabilidade	4,59
	Controle	3,58
	Desafio	4,05
	Apresentação Visual	4,37
		4,15
Limiar de Diversão	Qualidade Visual	4,32
	Apresentação de Áudio	4,11
	Complexidade e Diversidade	3,95
	Mecânicas	4,47
	Liberdade	3,58
	Níveis	4,16
	Equilíbrio do Grau de Dificuldade[3]	3,53
	Gratificação	3,89
		4,07

Além de indicar o resultado do *playtesting*, as médias permitem verificar quais fatores se sobressaíram e quais devem ser melhorados. Os fatores que receberam o melhor *feedback* foram: usabilidade, mecânicas e apresentação. O equilíbrio do grau de dificuldade também provou estar próximo do ideal (tendendo um pouco a ser rápido demais), uma vez que a média 3,53 está próxima de 3, considerada ideal. Por outro lado, os fatores que receberam o pior *feedback* foram: gratificação, liberdade e controle.

Ao analisar as questões relacionadas aos fatores com as piores médias, pode-se inferir que os jogadores acham que o jogo não proporciona a liberdade de escolha e o sentimento de gratificação que poderia proporcionar, e tampouco fornece a melhor

[3]A questão referente a este fator possui caráter de exceção, por isso não foi considerada para o cálculo da média.

opção de controle durante a realização dos *puzzles*. Outrossim, os participantes do *playtesting* foram encorajados a dar críticas e sugestões de melhoria. Dentre as observações feitas, as mais recorrentes foram:

- O teclado poderia ser uma opção melhor do que o *mouse* para controlar a interface dos *puzzles*;
- *Checkpoints* (marcadores na metade dos níveis) poderiam ser implementados para diminuir a frustração do jogador ao morrer ao final de uma fase;
- O jogador deveria ter a opção de ignorar o tutorial dos *puzzles* caso soubesse como proceder;
- O tutorial dos *puzzles* poderia utilizar-se de mais recursos audiovisuais e reduzir a quantidade de texto;
- As instruções referentes à movimentação deveriam indicar somente as setas para frente e para trás do teclado, uma vez que as demais não interferem no controle da personagem;
- O efeito de suavização do movimento da câmera que segue a personagem deveria ser removido ou ajustado, pois faz com que a personagem se perca de vista em determinadas partes do cenário.

No geral, pode-se dizer que alguns aspectos do jogo precisam ser revistos, principalmente no que diz respeito ao polimento (acabamentos, detalhes, efeitos, etc.). Apesar disso, muitos dos comentários realizados pelos *testers* foram positivos, consistindo em elogios às mecânicas, aos *puzzles* e à estética do jogo, o que corrobora com os resultados positivos do *playtesting*.

6 CONCLUSÃO

Este projeto foi dividido em duas etapas: concepção e planejamento, onde o jogo foi definido; e desenvolvimento e testes, etapa na qual o jogo foi desenvolvido de fato. A primeira etapa deste trabalho envolveu a investigação das relações entre os *serious games* e a aprendizagem, a pesquisa de jogos digitais semelhantes e o projeto de um *serious game* para o ensino de lógica de programação.

Ao pesquisar os conceitos relacionados ao desenvolvimento de jogos sérios, foi possível aprender o que são *serious games* e que benefícios a sua utilização pode trazer. Além disso, foi investigado o funcionamento do processo de aprendizagem do ponto de vista da teoria da Zona de Desenvolvimento Proximal de Vygotsky, o que permitiu estabelecer relações entre os *serious games* e a aprendizagem. A partir deste estudo, percebeu-se que um *serious game* atua no papel do "alguém mais experiente", buscando sempre perturbar a ZDP do jogador. Durante a pesquisa, verificou-se também que a diversão pode ser um fator muito importante em um jogo sério, principalmente no que diz respeito à motivação.

Para complementar o referencial teórico, foram pesquisados jogos com propostas similares à deste trabalho. Essa parte da pesquisa foi muitíssimo importante, pois permitiu estabelecer parâmetros de comparação e apresentou vários elementos que puderam ser utilizados em *"Alice e o Mistério dos Algoritmos"*. Além disso, foi necessário escolher uma metodologia para o desenvolvimento do jogo proposto. O *framework* DPE mostrou-se adequado para a modelagem de todas as partes de um *serious game*, pois sua estrutura abrange desde a elaboração do conteúdo, até a criação de mecânicas e a escolha da tecnologia. A elaboração da proposta de solução se baseou nas camadas deste *framework*.

A segunda etapa do projeto, por sua vez, envolveu atividades relacionadas ao desenvolvimento do jogo, como a elaboração de conteúdo, criação e seleção de materiais audiovisuais, *level design* e programação. O jogo foi desenvolvido na *engine* Unity3D, sendo que os *scripts* foram criados inteiramente para o propósito deste trabalho (exceto os *scripts* de movimentação do personagem e animação da interface gráfica, que foram adquiridos com o intuito de facilitar o desenvolvimento).

Todas as atividades deste projeto foram realizadas com o intuito de responder a questão que motivou e, consequentemente, direcionou este trabalho de conclusão de curso: *"um jogo educativo para ensino de lógica de programação pode funcionar como produto de entretenimento e ferramenta de aprendizagem ao mesmo tempo?"*. Pode

não parecer, mas responder a essa pergunta foi um trabalho bastante desafiador. Afinal, respondê-la implicou em pesquisar e relacionar dois temas consideravelmente distintos: diversão e aprendizagem. Desta forma, foi necessário responder um ponto de cada vez.

O primeiro ponto é: *o jogo é divertido?* Conforme discutido no referencial teórico, a diversão é subjetiva. Logo, foi necessário procurar por um método que, de alguma forma, pudesse estimar e mensurar a diversão proporcionada pelo jogo. A pesquisa resultou na escolha do modelo de três níveis proposto por Wang, Shen e Ritterfield. De acordo com esse modelo, se um jogo consegue atingir o limiar de diversão (segundo nível do modelo), ele pode ser considerado divertido.

Para verificar a resposta para essa pergunta, foi realizada uma sessão de *playtesting*. O teste consistia em jogar os quatro níveis do jogo e, em seguida, responder a um questionário. As perguntas do questionário eram diretamente ligadas aos fatores de diversão que compõem os limiares do modelo de três níveis de diversão. Os resultados do *playtesting* indicaram que o jogo conseguiu atingir os limiares de jogabilidade e diversão. Desta forma, pode-se dizer que a questão de pesquisa foi parcialmente respondida, uma vez que foi possível verificar que o jogo é divertido.

A questão de pesquisa deste trabalho de conclusão também menciona a avaliação do *serious game* como ferramenta de aprendizagem. Para responder a essa parte da questão íntegra e cientificamente, todavia, seria necessário realizar uma extensa série de testes de jogabilidade e provas, além de dispor de, no mínimo, dois grupos de testadores (um com e o outro sem o apoio do *serious game*) sem conhecimentos prévios na disciplina de algoritmos.

Um ambiente de testes com tamanha complexidade demandaria mais tempo para ser realizado, não cabendo no escopo deste trabalho. Por conseguinte, optou-se por não avaliar o *serious game* como ferramenta de aprendizagem. Entretanto, isso não significa que a resposta da questão de pesquisa foi negligenciada; afinal, foi em virtude dessa questão que o conteúdo do jogo foi elaborado com base na teoria da Zona de Desenvolvimento Proximal de Vygtosky.

Além disso, com a experiência prática e teórica obtida ao longo deste projeto, sabe-se quais são os procedimentos necessários para verificar se um jogo é um produto de entretenimento e uma ferramenta de aprendizagem ao mesmo tempo. Desta forma, este trabalho limita-se em afirmar que, mediante ao teste de jogabilidade realizado e as premissas de desenvolvimento de jogos seguidas, o *serious game* desenvolvido pode ser considerado um produto de entretenimento.

Com os objetivos específicos cumpridos e a questão de pesquisa parcialmente respondida, é importante ressaltar que as expectativas para o desenvolvimento - que eram muito altas - foram atingidas. Criar os níveis, desenvolver os *puzzles* e elaborar o conteúdo do jogo foram atividades bastante desafiadoras. Afinal, desenvolver um

jogo não é uma tarefa fácil. Contudo, pode-se dizer que, apesar do *game* desenvolvido possuir diversos aspectos a melhorar, o objetivo principal - e pessoal - deste trabalho de conclusão de curso foi cumprido.

REFERÊNCIAS

ABT, Clark. **Serious Games**. Lanham: University Press Of America, 1987. Disponível em: <https://books.google.com.br/books?id=axUs9HA-hF8C>. Acesso em: 28 nov. 2015.

ADAMS, E.; ROLLINGS, A.. **Fundamentals of Game Design**. New Jersey: Prentice Hall, 2007.

BLOOM, Benjamin S. **Taxonomia de los objetivos de la educacion:** la clasificación de las metas educacionales. Buenos Aires: AID, 1971. 364 p. (Biblioteca nuevas orientaciones de la educacion)

CODE.ORG. **Code.org**. 2015. Disponível em: <https://code.org/>. Acesso em: 16 nov. 2015.

CODECOMBAT. **About:** CodeCombat - Learn how to code by playing a game. 2015. Disponível em: <http://br.codecombat.com/about>. Acesso em: 16 nov. 2015.

CSIKSZENTMIHALYI, Mihaly. **Flow:** The Psychology of Optimal Experience. New York: Harpercollins, 1990.

EBERLY, D.h.. **3D Game Engine Design:** A Practical Approach to Real-Time Computer Graphics. 2. ed. Florida: Crc Press, 2006. Disponível em: <https://books.google.com.br/books?id=TnwZBwAAQBAJ>. Acesso em: 28 nov. 2015.

ESRB. **Game Ratings & Content Descriptors**. 2015. Disponível em: <http://www.esrb.org/ratings/ratings_guide_gamecenter.aspx >. Acesso em: 28 nov. 2015.

FARDO, Marcelo Luís. **A gamificação como estratégia pedagógica:** estudo de elementos dos games aplicados em processos de ensino e aprendizagem. 2013. 104 f. Dissertação (Mestrado em Educação) - Programa de Pós-Graduação em Educação, Universidade de Caxias do Sul, Caxias do Sul.

FERREIRA, Aurélio Buarque de Holanda. **Novo dicionário Aurélio da língua portuguesa.** 3. ed. Curitiba: Positivo, 2004.

GEE, James Paul. **Learning by design:** Games as learning machines. 2004. Dis-

ponível em: <http://www.raco.cat/index.php/IEM/article/download/204239/272773>. Acesso em: 28 nov. 2015.

HUGUET, Marie-Pierre. **Observations from Marie-Pierre Huguet:** Beyond the game. In: SHELDON, Lee. The Multiplayer Classroom: Designing Coursework as a Game. Boston, MA: Cengage Learning, 2012.

HUIZINGA, Johan. Homo ludens: Versuch einer bestimmung des spielelements der kultur.1938. Publicado originalmente em 1944. Tradução para língua portuguesa: **Homo Ludens: O Jogo Como Elemento da Cultura**. São Paulo, SP. Perspectiva, 1999.

HUNICKE, Robin; LEBLANC, Marc; ZUBEK, Robert. **MDA:** A Formal Approach to Game Design and Game Research. 2004. Disponível em: <http://www.aaai.org/Papers/Workshops/2004/WS-04-04/WS04-04-001.pdf>. Acesso em: 28 nov. 2015.

JENKINS, Henry. **Getting serious about games.** 2006. Disponível em: <http://www.henryjenkins.org/2006/07/getting_ serious_about_games.html>. Acesso em: 28 nov. 2015.

KIM, Scott. **The art of puzzles.** 2008. Disponível em: <https://www.ted.com/talks/scott_kim_takes_apart_the_art_of_puzzles>. Acesso em: 13 nov. 2015.

LIEBERMAN, D. A. **What can we learn from playing interactive games?.** In: VORDERER, Peter; BRYANT, Jennings (Ed.). Playing Video Games: Motives, Responses and Consequences. Mahwah, N.j: Lawrence Erlbaum Associates, 2006. Cap. 25. p. 379-397.

MICHAEL, David; CHEN, Sande. **Serious Games:** Games That Educate, Train, and Inform. 2. ed. Connecticut: Cengage Learning Ptr, 2005.

PRENSKY, Marc. **Digital Natives, Digital Immigrants.** In: PRENSKY, Marc. On the Horizon. NCB University Press, Vol. 9 No. 5, Outubro 2001. Disponível em: <http://www.marcprensky.com/writing/Prensky

RABIN, Steven (Ed.). **Introdução ao desenvolvimento de games.** São Paulo: Cengage Learning, 2011.

RICHVOLDSEN, Havard. **Serious Gaming:** Serious content in an entertaining

framework. 2009. 50 f. Dissertação (Mestrado) - Curso de Master Of Science In Electronics, Department Of Electronics And Telecommunications, Norwegian University Of Science And Technology, Trondheim, 2009.

RITTERFIELD, U; WEBER, R. **Video Games for Entertainment and Education.** In: VORDERER, Peter; BRYANT, Jennings (Ed.). Playing Video Games: Motives, Responses and Consequences. Mahwah, N.j: Lawrence Erlbaum Associates, 2006. Cap. 27. p. 399-413.

RITTERFIELD, Ute; CODY, Michael; VORDERER, Peter (Ed.). **Serious Games:** Mechanisms and Effects. Nova Iorque: Routledge, 2009.

SCHELL, Jesse. **The Art of Game Design:** A Book of Lenses. Florida: Crc Press, 2008. 520 p.

SHREVE, Jenn. **Let the Games Begin:** Entertainment Meets Education. 2005. Disponível em: <http://www.edutopia.org/video-games-classroom>. Acesso em: 28 nov. 2015.

TILED. **Tiled Map Editor.** 2016. Disponível em: <http://www.mapeditor.org>. Acesso em: 27 nov. 2016.

UCS (Caxias do Sul). **Projeto Gamification.** 2013. Disponível em: <https://www.ucs.br/portais/ccet/pesquisa/projetos/10529/>. Acesso em: 28 nov. 2015.

UNITY3D. **Unity - Game Engine.** 2015. Disponível em: <https://unity3d.com/pt>. Acesso em: 28 nov. 2015.

VYGOTSKY, Lev S. **A Formação Social da Mente:** O Desenvolvimento dos Processos Psicológicos Superiores. Org. por Michel Cole et al. Tradução José Cipolla Neto, Luís Silveira Menna Barreto, Solange Castro Afeche. 6ª Ed. São Paulo: Martins Fontes, 1998.

WANG, Hua; SHEN, Cuihua; RITTERFELD, Ute. **Enjoyment of Digital Games:** What Makes Them "Seriously" Fun?. In: RITTERFELD, Ute; CODY, Michael; VORDERER, Peter (Ed.). Serious Games: Mechanisms and Effects. Nova Iorque: Routledge, 2009. Cap. 3. p. 25-47.

WINN, Brian. **The design, play, and experience framework.** In: FERDIG, Richard E.. Handbook of Research on Effective Electronic Gaming in Education.

Hershey, Pa: Igi Global, 2008. p. 1010-1024. Disponível em: <http://gel.msu.edu/winn/Winn_DPE_chapter_final.pdf>. Acesso em: 28 nov. 2015.

WINN, Brian; HEETER, Carrie. **Resolving conflicts in educational game design through playtesting.** Innovate: Journal of Online Education, v. 3, n. 2, 2007.

APÊNDICE A - Questionário pós-*playtesting*

Alice e o Mistério dos Algoritmos

Este formulário tem o objetivo de avaliar a experiência de jogo proporcionada por Alice e o Mistério dos Algoritmos, jogo que desenvolvi como parte de meu trabalho de conclusão de curso.

O preenchimento demora menos de 10 minutos e é MUITÍSSIMO importante para a conclusão do meu trabalho! Agradeço muito, desde já!

*Obrigatório

Usabilidade

1. **O objetivo de cada nível sempre esteve claro para você?** *
   Ex: direção para seguir, o que fazer e onde chegar (desconsidere áreas secretas e itens escondidos)
   Marcar apenas uma oval.

	1	2	3	4	5	
Nunca esteve claro	◯	◯	◯	◯	◯	Sempre esteve claro

2. **Avalie a qualidade das instruções fornecidas durante o jogo** *
   Ex: informações sobre como se movimentar, pular, coletar moedas, interagir com enigmas, etc
   Marcar apenas uma oval.

	1	2	3	4	5	
Péssima	◯	◯	◯	◯	◯	Ótima

3. **O quão intuitivos são os ícones utilizados no jogo?** *
   Avalie sua compreensão do significado dos ícones do jogo (Ex: itens a coletar, moedas, ponto de interrogação e placas de aviso)
   Marcar apenas uma oval.

	1	2	3	4	5	
Nada intuitivos	◯	◯	◯	◯	◯	Muito intuitivos

4. **Como você avalia o tutorial disponível no primeiro nível do jogo?** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Péssimo	◯	◯	◯	◯	◯	Ótimo

5. **Avalie a qualidade da interação com a interface gráfica dos enigmas do jogo** *
   Ex: utilização do mouse, posicionamento, resposta aos cliques, etc.
   Marcar apenas uma oval.

	1	2	3	4	5	
Péssima	◯	◯	◯	◯	◯	Ótima

Controle

6. **O quanto você concorda que o mouse é melhor do que o teclado para resolver um enigma do jogo?** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Discordo plenamente	◯	◯	◯	◯	◯	Concordo plenamente

Desafio

7. **Em se tratando de desafio, como foi a sua experiência de jogo?** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Entediante	◯	◯	◯	◯	◯	Desafiadora

Apresentação Visual

8. **Avalie o posicionamento dos menus, indicadores de status e instruções** *
   Ex: escolha dos níveis do jogo, indicadores de pontuação e vidas restantes, instruções sobre como jogar
   Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

9. **Avalie os ícones de cada botão, esquema de cores e o posicionamento dos componentes (matriz de desenho, histórico e comandos) na interface gráfica dos enigmas do jogo** *
   Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

Qualidade visual

10. **O quanto você gostou dos gráficos do jogo?** *
    Ex: personagem, plataformas, árvores, plano de fundo, itens
    Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

Apresentação de áudio

11. **Avalie a qualidade das músicas e efeitos sonoros do jogo** *
    Ex: músicas de fundo, menu principal, tema da vitória e demais efeitos sonoros.
    Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

Complexidade e Diversidade

12. **Considerando a curta duração do jogo, indique o quão diverso e complexo o jogo lhe pareceu** *
    Considere os objetivos primários, itens escondidos, áreas secretas, número de ações possíveis dentro do mundo do jogo
    Marcar apenas uma oval.

	1	2	3	4	5	
Nada complexo ou diverso	◯	◯	◯	◯	◯	Muito complexo e diverso

Mecânicas

13. **Indique o quanto você gostou da ideia de coletar moedas e itens escondidos nos níveis do jogo** *
    Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

14. **Indique o quanto você gostou da ideia de abrir caminhos bloqueados no cenário por meio de enigmas** *
    Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

Liberdade

15. **Avalie a liberdade de escolha que o jogo proporciona** *
    Considere os caminhos obrigatórios, alternativos (maneiras diferentes de avançar) e áreas secretas
    Marcar apenas uma oval.

	1	2	3	4	5	
Não há liberdade	◯	◯	◯	◯	◯	Liberdade total

Níveis

16. **Avalie a qualidade dos níveis (fases) do jogo** *
Considere todos os seguintes aspectos: desafio, complexidade, liberdade, mecânicas, diversidade, posicionamento dos itens e moedas, caminhos bloqueados, posicionamento dos enigmas, etc
Marcar apenas uma oval.

	1	2	3	4	5	
Não gostei	◯	◯	◯	◯	◯	Adorei

Equilíbrio do Grau de dificuldade

17. **Considerando a curta duração do jogo, indique o quão rapidamente a dificuldade do jogo aumenta** *
Exemplo de análise: o jogo exige muito conhecimento do jogador a partir do nível 3 e os níveis 1 e 2 não fornecem o que é necessário para ser bem-sucedido nos níveis seguintes
Marcar apenas uma oval.

	1	2	3	4	5	
Muito lentamente	◯	◯	◯	◯	◯	Muito rapidamente

Gratificação

18. **O jogo proporciona uma sensação de recompensa na medida em que os desafios são concluídos?** *
Ex: você se sente recompensado ao ganhar três estrelas após coletar todas as moedas e itens escondidos de um nível
Marcar apenas uma oval.

	1	2	3	4	5	
Não me sinto recompensado	◯	◯	◯	◯	◯	Sinto-me muito recompensado