

**CENTRO UNIVERSITÁRIO DE ANÁPOLIS - UNIEVANGÉLICA CURSO DE
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

LEANDRO DA SILVA CÂMARA

**DESENVOLVIMENTO DE UM JOGO ELETRÔNICO PARA A EDUCAÇÃO
INCLUSIVA**

Anápolis - GO

2017

LEANDRO DA SILVA CÂMARA

**DESENVOLVIMENTO DE UM JOGO ELETRÔNICO PARA A EDUCAÇÃO
INCLUSIVA**

Trabalho de Conclusão de Curso
apresentado ao Curso de Bacharelado em
Engenharia de Computação do Centro
Universitário de Anápolis –
UniEVANGÉLICA, sob orientação do Prof.
Ms. Millys Fabrielle Araujo Carvalhaes.

Anápolis - GO

2017

LEANDRO DA SILVA CÂMARA

**DESENVOLVIMENTO DE UM JOGO ELETRÔNICO PARA A EDUCAÇÃO
INCLUSIVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Computação do Centro Universitário de Anápolis – UniEVANGÉLICA, sob orientação do Prof. Ms. Millys Fabrielle Araujo Carvalhaes.

Banca Examinadora

.....
Prof. Ms. Millys Fabrielle Araujo Carvalhaes
Orientador

.....
Prof. (.....)
Convidado

.....
Prof. (.....)
Convidado

Nota:

Anápolis, de de 2017.

DESENVOLVIMENTO DE UM JOGO ELETRÔNICO PARA A EDUCAÇÃO INCLUSIVA

RESUMO

Em muitos aspectos, os jogos eletrônicos possibilitam um melhor ambiente de aprendizado. Os jogos permitem um ajuste de nível de dificuldade de acordo com as habilidades do jogador, além de prover ao mesmo um *feedback* claro e imediato, e possibilita escolhas e controle sobre suas ações. Este trabalho propõe o desenvolvimento de um projeto de jogo eletrônico, utilizando as técnicas de *game design* e a ferramenta *Unity* (para implementação), assim como a elaboração de um *game design document* (GDD). A pesquisa busca demonstrar o desenvolvimento de um jogo eletrônico educativo e por meio das classificações da Taxonomia de Bloom, a equipe responsável pelo projeto do jogo consiga planejar e projetar a aprendizagem proposta pelo jogo, além de servir de apoio para o professor na avaliação da efetividade da aprendizagem. O resultado deste trabalho visa descrever o desenvolvimento de um jogo eletrônico que apoie o professor em sala de aula, facilitando no processo de ensino e na inclusão de alunos com necessidades especiais educacionais.

Palavras-chave: Educação Inclusiva. Jogos educacionais. Jogo sérios. Taxonomia de Bloom. Tecnologias educacionais.

DESENVOLVIMENTO DE UM JOGO ELETRÔNICO PARA A EDUCAÇÃO INCLUSIVA

ABSTRACT

In many ways, electronic games enable a better learning environment. The games allow the adjustment of difficulty according to the player's abilities, besides providing the same a clear and immediate comment, and allows choices and control over their actions. This paper proposes the development of an electronic gaming project, using the game design techniques and the Unity tool (for implementation), as an elaboration of a game design document (GDD). The research seeks to demonstrate the development of an electronic educational game and through the distribution of the Bloom Taxonomy, a team responsible for the game project is able to plan and design a learning proposed by the game, as well as to support the teacher in the evaluation of effectiveness learning. The result is a learning work in the development of an electronic game that supports the teacher in the classroom, facilitating the teaching process and the inclusion of students with special educational needs.

Keywords: Educational games. Educational Technologies. Inclusive education. Serious game. Taxonomy of Bloom.

LISTA DE ILUSTRAÇÕES

Figura 1 - Categorização atual da Taxonomia de Bloom proposta por Anderson, Krathwohl e Airasian, no ano de 2001.....	16
Figura 2 - Módulos de uma <i>Game Engine</i>	22
Figura 3 - Ambiente de desenvolvimento do <i>Construct 2</i>	24
Figura 4 - Ambiente de desenvolvimento do <i>Game Maker</i>	24
Figura 5 - Ambiente de desenvolvimento do <i>RPG Maker</i>	25
Figura 6 - Ambiente de desenvolvimento do <i>Source 2</i>	26
Figura 7 - Ambiente de desenvolvimento do <i>Unreal Engine 4</i>	27
Figura 8 - Ambiente de desenvolvimento do <i>Unity 3D</i>	27
Figura 9 – Logo do jogo <i>BaseKids</i>	31
Figura 10 – Personagem do jogo <i>BaseKids</i>	34
Figura 11 – Fluxograma do jogo <i>BaseKids</i>	37
Figura 12 – Estrutura do projeto.....	41
Figura 13 – Organização dos componentes das fases	42
Figura 14 – Organização dos componentes gerais e de interfaces de menu.....	43
Figura 15 – Componente <i>Box Collider 2D</i>	44
Figura 16 – Tela inicial	46
Figura 17 – Mapa de Fases.....	47
Figura 18 – Fase “Massas e Outros”	47
Figura 19 – Formação de palavra	48

LISTA DE TABELAS

Tabela 1 – Demonstração do Fluxo Básico.....	38
Tabela 2 – Mapa de Fases.....	46

LISTA DE ABREVIATURA E SIGLAS

2D	Duas dimensões (bidimensional)
3D	Três dimensões (tridimensional)
APAE	Associação de Pais e Amigos dos Excepcionais
GDD	<i>Game Design Document</i>
HTML5	<i>HyperText Markup Language</i> (versão 5)
IA	Inteligência Artificial
NEE	Necessidades Especiais Educacionais
PBL	<i>Project Based Learning</i>
RPG	<i>Role Playing Game</i> (Jogo de Interpretação)
TBL	<i>Team-based Learning</i>
WAC	<i>Writing Across the Curriculum</i>

SUMÁRIO

1 INTRODUÇÃO	11
2 REFERENCIAL TEÓRICO	14
2.1 Metodologias de ensino.....	14
2.1.1 Educação inclusiva.....	14
2.1.2 Taxonomia de Bloom	16
2.2 <i>Game Design</i>.....	18
2.2.1 Elementos do <i>Game Design</i>.....	18
2.2.2 <i>Game Design Document</i>.....	19
2.3 <i>Game Engine</i>.....	20
2.3.1 Arquitetura	21
2.3.2 <i>Game engines</i>	23
2.3.2.1 <i>Unity</i>	28
3 METODOLOGIA DE PESQUISA	30
4 PROJETO DO JOGO.....	31
4.1 <i>BaseKids</i>.....	31
4.1.1 Classificação de aprendizagem.....	32
4.1.2 GDD.....	33
4.1.2.1 Conceito	33
4.1.2.2 <i>Gameplay</i>	34
4.1.2.2.1 Personagem	34
4.1.2.2.2 Voz do personagem	34
4.1.2.2.3 Enredo	34
4.1.2.2.4 Fluxo do jogo.....	35
4.1.2.2.5 Progressões / regras do jogo.....	39
4.1.3 Implementação.....	40
4.1.3.1 Estrutura do projeto	40
4.1.3.2 <i>Box Collider 2D</i>.....	44

4.1.3.3 <i>IEventHandler</i>	44
5. RESULTADOS	45
6 CONSIDERAÇÕES FINAIS	49
7 REFERÊNCIAS BIBLIOGRÁFICAS	50
APÊNDICE 1 – Imagens do jogo <i>BaseKids</i>	53

1 INTRODUÇÃO

O contexto escolar é um ambiente que deve ser estimulador para várias aprendizagens, apresentando elementos que influenciam o desenvolvimento dos alunos. Dentre esses elementos destaca-se a importância do professor e dos alunos no processo de ensino/aprendizagem, enquanto participantes ativos nos processos de construção do conhecimento.

Entretanto, devido familiarização dos alunos com ferramentas tecnológicas, como *smartphones*, computadores e *videogames*, muitos professores têm dificuldade em manter os alunos atentos, motivados e engajados dentro do processo de ensino/aprendizagem. Além disso, quando há alunos com necessidades educacionais especiais nas salas de aula, a dificuldade se torna ainda maior, pois afirmam não estar preparados para educação inclusiva¹ (TOLEDO; MARTINS, 2009). Em relação aos professores de turmas inclusivas, Toledo e Martins (2009) afirmam que:

“É fundamental que eles tenham proatividade com relação às situações que vivenciam no contexto escolar, especialmente quando são responsáveis por atuar junto a crianças que apresentam necessidades especiais.” (TOLEDO; MARTINS, 2009, p. 4127).

Dados do Censo Escolar da Educação Básica de 2016 revelam que houve um crescimento significativo nas matrículas de pessoas com deficiência na educação básica regular, e que, ainda no ano de 2016, eram quase 800 mil matrículas em turma comuns. Das escolas brasileiras, 57,8% possuíam alunos com deficiência, transtornos globais do desenvolvimento ou altas habilidades incluídas em classes comuns. Em 2008, esse percentual era de apenas 31% (INEP, 2017).

De acordo com Lins (2014), a educação inclusiva enfrentou diversos problemas no decorrer de sua implantação, entre eles: a falta de formação dos profissionais da educação; a discriminação vinda dos pais, dos colegas e da sociedade; a falta de materiais didáticos adequados e instalações propícias para a inclusão.

Sabe-se que a boa vontade dos professores e sua preparação são condições necessárias, no entanto, nem todos os professores estão preparados para lidar com turmas onde há alunos com necessidades educacionais especiais (NEE). Marchesi

¹ Educação inclusiva é uma abordagem humanística, democrática que percebe o sujeito e suas singularidades tendo como objetivos o crescimento, a satisfação pessoal e a inserção social de todos (NORONHA; PINTO, 2012).

(2004) apresenta uma visão semelhante ao comentar que criar escolas inclusivas requer muito mais que boas intenções, declarações e documentos oficiais, requer que a sociedade, escolas e professores tomem consciência das tensões e organizem condições para criação de escolas inclusivas de qualidade. Além disso, para este autor, a preparação do professor também se constitui condição necessária para o processo de inclusão dos alunos com NEE. Para Marchesi (2004):

“É muito difícil avançar no sentido das escolas inclusivas se os professores em seu conjunto, e não apenas professores especialistas em educação especial, não adquirirem uma competência suficiente para ensinar todos os alunos.” (MARCHESI, 2004, p. 44).

Neste sentido, na sociedade atual, a eficácia do ensino tem sido questionada por não atender ao perfil dos estudantes, que possuem um padrão de raciocínio diferente das gerações anteriores (PRENSKY, 2010). Tem-se o aluno, que possui habilidade com *games* e outras mídias digitais, e o professor, que censura e condena este tipo de prática (PRENSKY, 2010). Portanto, aumenta a dificuldade do professor em prender a atenção do aluno e mantê-lo engajado e motivado dentro do processo de aprendizagem.

Estudos recentes apontam a eficácia de novas tecnologias para o processo de ensino. Uma das que vem ganhando destaque é a utilização de jogos eletrônicos em sala de aula que contribui positivamente para o processo de ensino (WANG, 2007). De acordo com Wang (2007), os jogos eletrônicos possibilitam um melhor ambiente de aprendizado, além de permitir um ajuste de nível de dificuldade conforme as habilidades do jogador, provendo um *feedback* claro e imediato, escolhas e controle sobre suas ações.

Diante deste contexto, este trabalho propõe desenvolver um jogo eletrônico, baseando-se em estudos bibliográficos, utilizando a Taxonomia de Bloom para definição dos objetivos pedagógicos, as técnicas de *game design*, assim como o uso de uma *game engine* para implementação do mesmo. Este jogo tem o objetivo de servir como instrumento de apoio no processo de ensino/aprendizagem, no engajamento e motivação dos alunos no âmbito da educação inclusiva.

Para o desenvolvimento do jogo, faz-se necessário a realização de um estudo bibliográfico acerca da Taxonomia de Bloom, a identificação dos métodos e técnicas do *game design*, a elaboração do *Game Design Document* (GDD) e o uso da *game engine Unity* para implementação do jogo.

Neste contexto, com o advento das tecnologias em salas de aula e a familiarização dos alunos com os jogos eletrônicos, muitas escolas estão adicionando-os ao processo de ensino os benefícios que os jogos proporcionam às crianças e adolescentes. Segundo Alves e Bianchin (2010), o jogo (como instrumento de aprendizagem) é um recurso de extremo interesse aos educadores, uma vez que sua importância está diretamente ligada ao desenvolvimento do ser humano em uma perspectiva social, criativa, afetiva, histórica e cultural. Os jogos podem servir de apoio para o professor e pela sua capacidade de envolver o jogador ao assunto abordado, eles podem apoiar o trabalho do professor e da escola no processo de ensino e inclusão.

Kishimoto (1998) afirma que o jogo possui duas funções na educação: a **função lúdica**, uma vez que propicia diversão, e a **função educativa**, pois ensina qualquer coisa que complete o aluno em seu saber, seus conhecimentos e sua apreensão do mundo. Levando em conta Huizinga (2000), não há como negar a dimensão educativa dos jogos, uma vez que propiciam o desenvolvimento de várias habilidades no aluno como a atenção, a concentração, o raciocínio lógico, dentre tantas outras.

Por meio deste trabalho, pretende-se contribuir para o processo de ensino e de inclusão de alunos com NEE, através do desenvolvimento de um jogo eletrônico, relatando as etapas de sua construção, abordando desde sua idealização até a implementação do jogo.

O capítulo 2 aborda os conceitos e referenciais teóricos que apoiam a execução do trabalho, como a Taxonomia de Bloom, o *game design* e *game engine*. No capítulo 3 encontra-se a metodologia utilizada para elaboração e construção do jogo eletrônico.

No capítulo 4 contempla o projeto do jogo, abordando o *Game Design Document* (GDD) e a implementação do mesmo, enquanto no capítulo 5 são apresentados os resultados do trabalho.

2 REFERENCIAL TEÓRICO

Neste referencial teórico, serão abordados temas relevantes para o desenvolvimento do trabalho, como: metodologias de ensino, educação inclusiva, Taxonomia de Bloom, *game design*, *Game Design Document* (GDD), *game engine* e a ferramenta de desenvolvimento *Unity*.

2.1 Metodologias de ensino

No processo de ensino-aprendizagem, o professor deve levar em consideração que o conhecimento do aluno está em processo de construção e, por esse motivo, deve mobilizar o aluno e utilizar metodologias adequadas para repassar seu conhecimento e preparar o estudante na busca constante pelo conhecimento (MIRANDA; NOVA; JÚNIOR, 2012).

Segundo Traversini e Buaes (2009, p. 145), as metodologias de ensino são “práticas pedagógicas operacionalizadas por meio de conjuntos de atividades escolares propostas pelos professores com vistas a alcançar a aprendizagem de determinados conhecimentos, valores e comportamentos”.

As metodologias servem como um auxílio para o professor desempenhar o seu trabalho (KODJAOGLANIAN et al., 2003) e visam contribuir para o aprendizado do aluno. Os professores devem, entre as metodologias, identificar qual irá contribuir no processo de ensino-aprendizagem, sendo adequada à disciplina.

2.1.1 Educação inclusiva

O termo Incluir significa “compreender; abranger; inserir; introduzir; fazer parte” (FERREIRA, 2001, p. 380, apud ALMEIDA, 2015). Portanto, a proposta filosófica e legalista da inclusão é o respeito à diversidade (BRASIL, 1996, apud ALMEIDA, 2015). Isto implica afirmar que a inclusão não admite formas de segregação nem a partir do ponto de vista social e econômico, nem do ponto de vista político e cultural.

Na escola, a inclusão tem a meta principal de não permitir nenhum aluno fora do ensino regular (ALMEIDA, 2015). Suas características fundamentais, na opinião de Ballard (1997, apud Sánchez, 2005), são: a não discriminação das deficiências, da cultura e do gênero. Referindo-se a todos os alunos de uma comunidade escolar sem nenhum tipo de exceção. Para Ballard (1997, apud Sánchez, 2005), todos os alunos

têm o mesmo direito a ter acesso a um currículo culturalmente valioso e em tempo completo, como membros de uma classe escolar e de acordo com sua idade. A educação inclusiva enfatiza a diversidade mais que a semelhança.

Na década de 1990, mais exatamente no ano de 1994, aconteceu a Conferência Mundial sobre Necessidades Educativas Especiais em Salamanca, foi assinada a Declaração de Salamanca, que foi um marco histórico altamente significativo a favor da inclusão, pois contribuiu para impulsionar a Educação Inclusiva em todo o mundo (SÁNCHEZ, 2005).

O princípio fundamental que orienta a Declaração de Salamanca é o de que as escolas devem acomodar todas as crianças, possibilitando que elas aprendam juntas, independentemente de quaisquer dificuldades ou diferenças que possam ter, quer sejam de origens física, intelectual, social, emocional, linguística ou outras (AGUIAR, 2004).

Segundo Sánchez (2005), esta Declaração proclama que:

- Todas as crianças têm direito à educação e deve-se dar a elas a oportunidade de alcançar e manter um nível aceitável de conhecimentos;
- Cada criança tem características, interesses, capacidades e necessidades de aprendizagem que lhe são próprias;
- Os sistemas de ensino devem ser organizados e os programas aplicados de modo que tenham em conta todas as diferentes características e necessidades;
- As pessoas com necessidades educacionais especiais devem ter acesso às escolas comuns; e
- As escolas comuns devem representar um meio mais eficaz para combater as atitudes discriminatórias, criar comunidades acolhedoras, construir uma sociedade integradora e alcançar a educação para todos.

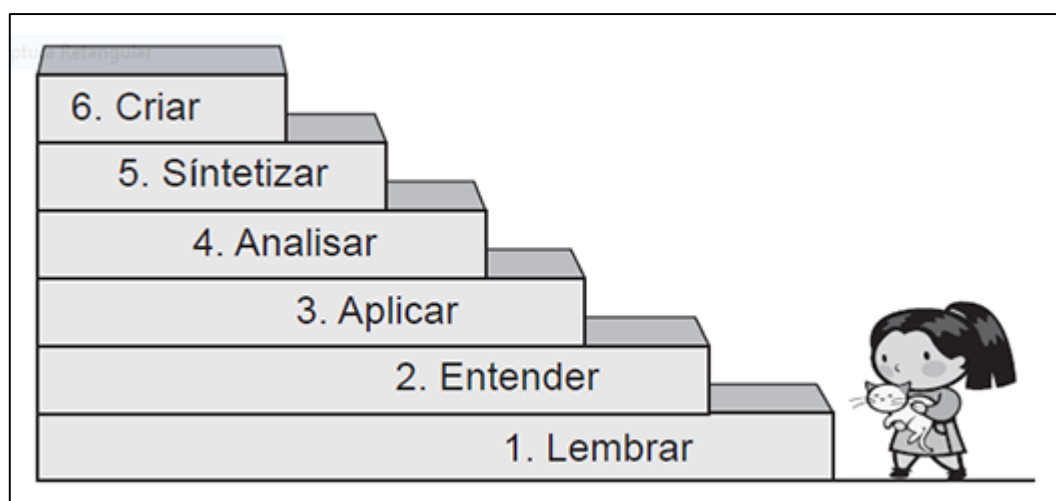
Assim, as escolas devem reconhecer e responder às diversas necessidades de seus discentes, respeitando tanto estilos como ritmos diferentes de aprendizagem e assegurando uma educação de qualidade a todos, por meio de currículo apropriado, modificações organizacionais, estratégias de ensino, uso de recursos e parcerias com a comunidade (AGUIAR, 2004).

2.1.2 Taxonomia de Bloom

A Taxonomia de Bloom é uma classificação de níveis de raciocínio, dividida em seis níveis, numa sequência que vai do mais simples (conhecimento) ao mais complexo (avaliação) (BARATO, 2009) - isso significa que, para adquirir uma nova habilidade pertencente ao próximo nível, o aluno deve ter dominado e adquirido a habilidade do nível anterior (FERRAZ; BELHOT, 2010). De acordo com Chapman (2009), é uma estrutura que pode ser aplicada para planejar, projetar e avaliar a efetividade da aprendizagem e de treinamentos.

Segundo Conklin (2005), a Taxonomia de Bloom, e sua classificação hierárquica dos objetivos de aprendizagem, tem sido uma das maiores contribuições acadêmicas para educadores que, conscientemente, procuram meios de estimular, nos seus discentes, raciocínio e abstrações de alto nível (*higher order thinking*).

Figura 1 - Categorização atual da Taxonomia de Bloom proposta por Anderson, Krathwohl e Airasian, no ano de 2001.



Fonte: Ferraz e Belhot (2010).

Os processos categorizados pela Taxonomia dos Objetivos Cognitivos de Bloom, além de representarem resultados de aprendizagem esperados, são cumulativos, o que caracteriza uma relação de dependência entre os níveis e são organizados em termos de complexidades dos processos mentais (FERRAZ; BELHOT, 2010), como mostra a Figura 1. A Figura 1 também apresenta as categorias (revisadas) criadas pelo grupo de Bloom e as formas de descrever os conhecimentos que se deseja que os alunos desenvolvam (WALL; TELLES, 2004).

Segundo Wall e Telles (2004), a estrutura do processo cognitivo na taxonomia de Bloom são definidas através dos seguintes conceitos:

- *Lembrar* – Reconhecer e reproduzir ideias e conteúdo. Reconhecer requer distinguir e selecionar uma determinada informação e reproduzir ou recordar. Está mais relacionado à busca por uma informação relevante memorizada. Representado pelos seguintes verbos no gerúndio: Reconhecendo e Reproduzindo.
- *Entender* – Estabelecer uma conexão entre o novo e o conhecimento previamente adquirido. A informação é entendida quando o aprendiz consegue reproduzi-la com suas “próprias palavras”. Representado pelos seguintes verbos no gerúndio: Interpretando, Exemplificando, Classificando, Resumindo, Inferindo, Comparando e Explicando;
- *Aplicar* – Executar ou usar um procedimento numa situação específica e pode também abordar a aplicação de um conhecimento numa situação nova. Representado pelos seguintes verbos no gerúndio: Executando e Implementando;
- *Analisar* – Dividir a informação em partes relevantes e irrelevantes, importantes e menos importantes e entender a inter-relação existente entre as partes. Representado pelos seguintes verbos no gerúndio: Diferenciando, Organizando, Atribuindo e Concluindo;
- *Sintetizar* – Realizar julgamentos baseados em critérios e padrões qualitativos e quantitativos ou de eficiência e eficácia. Representado pelos seguintes verbos no gerúndio: Checando e Criticando;
- *Criar* – Colocar elementos junto com o objetivo de criar uma nova visão, uma nova solução, estrutura ou modelo utilizando conhecimentos e habilidades previamente adquiridos. Envolve o desenvolvimento de ideias novas e originais, produtos e métodos por meio da percepção da interdisciplinaridade e da interdependência de conceitos. Representado pelos seguintes verbos no gerúndio: Generalizando, Planejando e Produzindo.

Segundo Ferraz e Belhot (2010), a Taxonomia de Bloom é um instrumento cuja finalidade é auxiliar a identificação e a declaração dos objetivos ligados ao

desenvolvimento cognitivo que engloba a aquisição do conhecimento, competência e atitudes, visando facilitar o planejamento do processo de ensino e aprendizagem.

2.2 Game Design

A projeção de um jogo dá-se de forma multidisciplinar e multifuncional, envolvendo as equipes responsáveis pelo *Game Design*, Programação, Artes, e outras áreas necessárias. Esse trabalho conjunto permite que o processo de desenvolvimento do jogo passe por todas as fases e testes necessários, resultando em um jogo eficiente e bem aceito pelo público (BREYER, 2006, apud AZOUBEL et al., 2016).

O “*game designer*” tem um papel de extrema relevância para o processo de criação e desenvolvimento de um jogo. Suas funções na equipe são diversas: pode atuar na escrita de documentos sobre o *Design*, na determinação dos objetivos e da lógica do jogo, projeção da interface a que o usuário terá acesso, concepção da história do jogo e seus diálogos, além de realizar testes com o *software* para garantir seu bom funcionamento (JUNIOR et al, 2002).

Em suma, é responsável por toda a mecânica do jogo. Desde seu funcionamento até as questões de *layout* e estética, dando início ao *game design* em si, ao planejar a forma como o jogo irá funcionar e comportar-se diante de um usuário específico. A escolha do público-alvo é um fator ao qual esse profissional deve atentar-se desde o início do processo, já que é a escolha do perfil que permitirá a criação das propriedades do jogo em desenvolvimento, grau de dificuldade, controle, contexto, entre outras características (SATO, 2010, apud AZOUBEL et al., 2016).

Brathwaite (2009, apud SATO, 2010) explica que o “*game design* é o processo de criar a disputa e as regras de um jogo”, além de frisar a importância de buscar promover a motivação para o jogador, por meio de objetivos.

2.2.1 Elementos do Game Design

O *game design* pode ser dividido em três áreas específicas: mecânica do núcleo, história/narrativa e interatividade, afirmam Rollings e Adams (2003). É a tradução da visão do *game designer* transcrita em um conjunto de regras que podem ser interpretadas pela equipe que escreve o código. A mecânica do núcleo é implementada através de um modelo (computacional) matemático do mundo do jogo.

Esta área descreve a forma como o jogo funciona, não a forma como o *software* opera (ROLLINGS; ADAMS, 2003).

Todos os jogos contam uma narrativa e sua complexidade depende do jogo. A narrativa refere-se à apresentação da história, interatividade, ao gráfico, som e as interfaces do usuário. A interatividade é o que o jogador vê, ouve e age dentro do jogo (ROLLINGS; ADAMS, 2003).

2.2.2 Game Design Document

O documento de *design* de jogo, ou GDD (*Game Design Document*) é uma ferramenta textual produzida pelo *game designer* que descreve todas as características de um jogo (PEDERSEN, 2009). Segundo Kreimeier (2003), o GDD é um documento de texto, em geral bem ilustrado, que descreve diversos elementos de um jogo, como jogabilidade (*gameplay*), controles, interfaces, personagens, armas, golpes, inimigos, fases e todos os aspectos gerais do jogo, tendo a função de comunicar e guiar os diversos envolvidos no processo de desenvolvimento do jogo. Muitas vezes, esse documento é chamado de "bíblia" do jogo, sendo realmente usado como uma Bíblia, uma referência para todos os envolvidos no desenvolvimento do projeto, mantendo todos ligados aos mesmos objetivos (PEDERSEN, 2009).

O GDD é inspirado na metodologia da produção cinematográfica, tendo como exemplo um roteiro de cinema. Com base nas informações definidas no documento, os artistas criam o visual do *game* e os programadores desenvolvem os recursos necessários para a interatividade do produto. Além de ser baseado em um roteiro técnico de cinema, o GDD também é o documento de projeto de um jogo (LEMES, 2009).

De acordo com Barbaro (1983, apud Lemes, 2009):

“Pode-se definir o roteiro técnico (de um filme) como uma sistemática e ordenada tentativa de previsão do futuro filme em todos os detalhes; previsão que, praticamente, se concretiza num manuscrito que contém a descrição, sequência por sequência e plano por plano, das ações, dos diálogos, a indicação dos sons e da música de acompanhamento, e a solução de todos os problemas técnicos e artísticos que se apresentarão durante a realização do filme.” (LEMES, 2009)

É no GDD que tudo se concentra antes da produção do jogo começar. Shuytema (2008, apud Lemes, 2009) afirma que este documento usa palavras, tabelas e diagramas para explicar o funcionamento de um *game*, a partir da história

do mundo ficcional do *game* até a organização de botões em interface e ao modo como um arqueiro luta contra um espadachim.

Perucia et al. (2007) listam alguns itens que podem ser abordados em GDD:

- **Conceito:** nome do jogo; apresentação resumida do jogo; público-alvo; estilo de jogo; história; principais regras do jogo;
- **Especificações técnicas:** *hardware*; sistema operacional; *hardware* mínimo; requerimentos de *software*; gráficos;
- **Especificações do jogo:** número de fases; níveis de dificuldade; vidas; descrição dos tipos ou modos de jogo; sistema de pontuação; sistema de *ranking* (ou *high scores*); opções de configuração; número de jogadores; recursos de carga e gravação (*load* e *save*); sistema de câmera; personagens; itens de jogo; itens de cenário; tabela de itens; evolução de fases; tabela de mensagens;
- **Dispositivos de entrada:** suporte para *mouse*; dispositivos de entrada para os menus; dispositivos de entrada do jogo; definição de teclas e botões;
- **Design gráfico e arte:** abertura; descrição de *layout* de menus e telas; descrição de *layout* do jogo na fase; definição de fases; definição do jogo;
- **Sonorização:** definição das músicas nos menus; definição das músicas nas fases; definição dos efeitos sonoros de menu e outros; definição dos efeitos sonoros de jogo (nas fases);
- **Desenvolvimento:** tempo de desenvolvimento; alocação de pessoal; metas.

2.3 Game Engine

A *game engine* (motor de jogo, traduzido pelo autor) é o elemento que controla todos os componentes e recursos de um jogo. Tudo gira em torno do motor, que é o núcleo de toda tecnologia. Em essência, é nele que estará o enredo transformado em regras, que podem ser implementadas pelo programador, além de disponibilizar recursos audiovisuais para a interface. No campo do desenvolvimento de jogos, o motor é o elemento dominado pela Ciência da Computação, concentrando todos os elementos técnicos de lógica de programação (FILHO, 2011).

Segundo Lewis e Jacobson (2002, apud Filho, 2011), os motores de jogos apresentam uma coleção de recursos que capturam eventos de entrada, geram saídas gráficas e de áudio e gerenciam a dinâmica presente no jogo. É comum que um motor comercial apresente, integradas dentro de um único produto, todas as ferramentas necessárias ao desenvolvimento completo de um jogo, deixando para o programador a tarefa de fazer a lógica do jogo.

2.3.1 Arquitetura

A arquitetura de uma *game engine* corresponde exatamente à definição dos seus módulos e a maneira como é realizada a interação entre eles. Tais módulos são responsáveis por identificar e gerenciar todos os eventos que ocorrem ao longo do jogo, desde eventos gerados pelo usuário até aqueles causados por interação entre os objetos do jogo. Por isso, existem normalmente módulos para manipulação de objetos, renderização, sonorização, inteligência artificial (IA), rede e outros (BATTAIOLA et al., 2002).

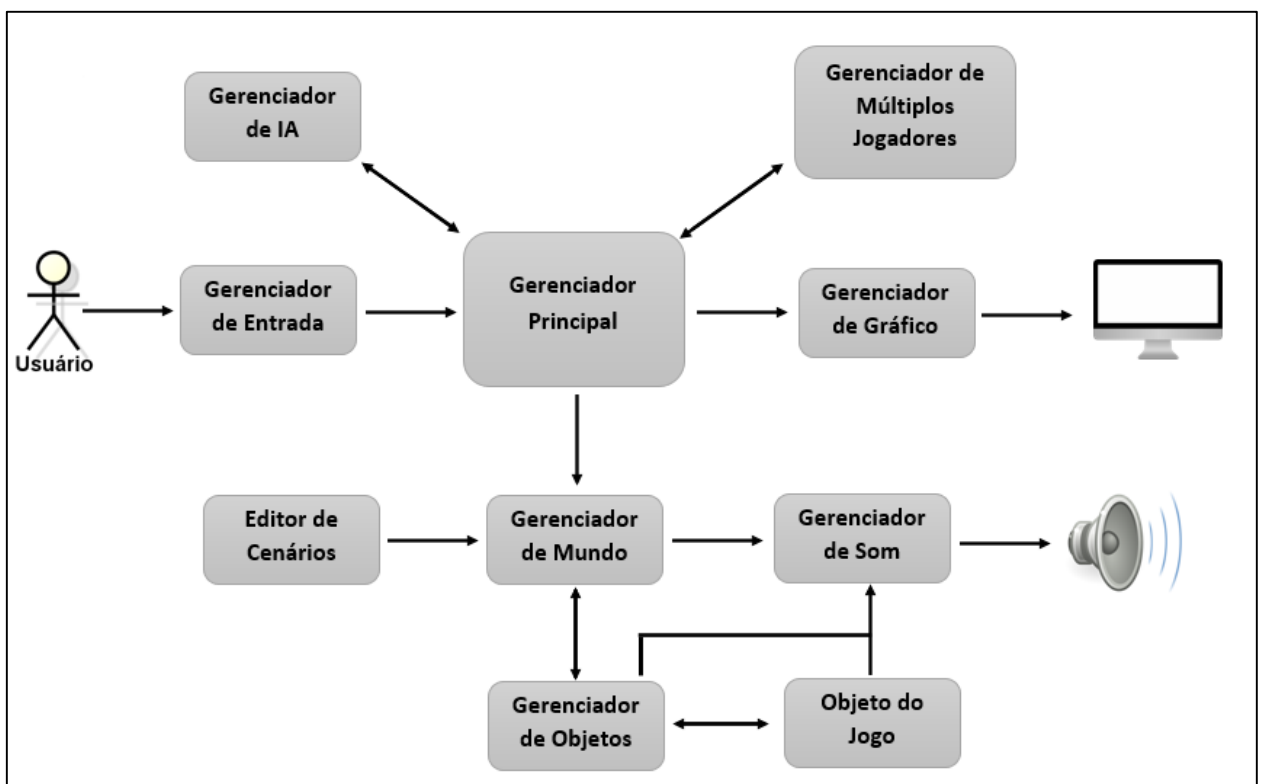
Segundo Battaiola et al. (2002) a arquitetura de uma *game engine* é composta pelos seguintes componentes:

- **Gerenciador de Entrada:** Módulo responsável por identificar eventos em dispositivos de entrada, por exemplo, o teclado, e encaminhar o ocorrido para o módulo principal que irá processar o dado e realizar a ação correspondente;
- **Gerenciador Gráfico:** Realizar todo o processamento necessário para transformar a cena criada pelos objetos do jogo em dados que sejam suportados pelas rotinas do sistema para desenho na tela;
- **Gerenciador de Som:** Responsável pela execução e processamento de todo o áudio do jogo;
- **Gerenciador de Inteligência Artificial:** Gerencia o comportamento de objetos controlados pelo computador. Para isso, realiza certas ações de acordo com o estado atual do jogo e algumas regras;
- **Gerenciador de Múltiplos Jogadores:** Permitir que jogadores do mesmo jogo se comuniquem entre si, gerencia a conexão e a troca de informações entre os diversos computadores que estão conectados via *Internet* ou *Intranet*;

- **Gerenciador de Objetos:** Módulo responsável por gerenciar os grupos de objetos do jogo, carregando os mesmos e controlando o seu ciclo de vida;
- **Objeto do Jogo:** Representa uma entidade que faz parte do jogo e as informações necessárias para que ela seja gerenciada, como por exemplo, posição, ângulo e dimensão, além disto, também é responsável pelo controle de colisões;
- **Gerenciador do Mundo:** Responsável por armazenar o estado atual do jogo, conta com os diversos gerenciadores de objetos para realizar tal tarefa;
- **Editor de Cenários:** Ferramenta para a criação de mundos que posteriormente serão carregados pelo gerenciador de mundos;
- **Gerenciador Principal:** Parte principal do jogo, servindo como ligação para a troca de informações entre os outros módulos do jogo;

A Figura 2 mostra, segundo Battaiola et al. (2002), como os módulos de uma *game engine* estão interligados.

Figura 2 - Módulos de uma *Game Engine*.



Fonte: (LIMA, 2008).

Segundo Pessoa e Ramalho (2001), uma *game engine* completa deve atender aos seguintes requisitos:

- Arquitetura modular, para que possa ser utilizada em cada novo jogo;
- Bom nível de abstração, ao prover objetos com funcionalidades já implementadas;
- Definição de objetos básicos, como objetos do jogo e o mapa (cenário) do jogo;
- Detecção e gerenciamento de eventos gerados por teclado, *mouse*, *joysticks*, etc.;
- Algoritmos para desenho dos objetos do jogo, podendo ter duas dimensões (2D) e três dimensões (3D);
- Funcionalidades úteis a jogos 3D, como iluminação e transformações geométricas;
- Execução dos sons do jogo em resposta a eventos ocorridos;
- Implementação de algoritmos de IA para os objetos inteligentes dos jogos;
- Implementação de algoritmos para troca remota de dados, gerenciamento de sessões, sincronização das informações compartilhadas do jogo, etc.;
- Implementação de algoritmos para modelagem física de objetos.

2.3.2 Game engines

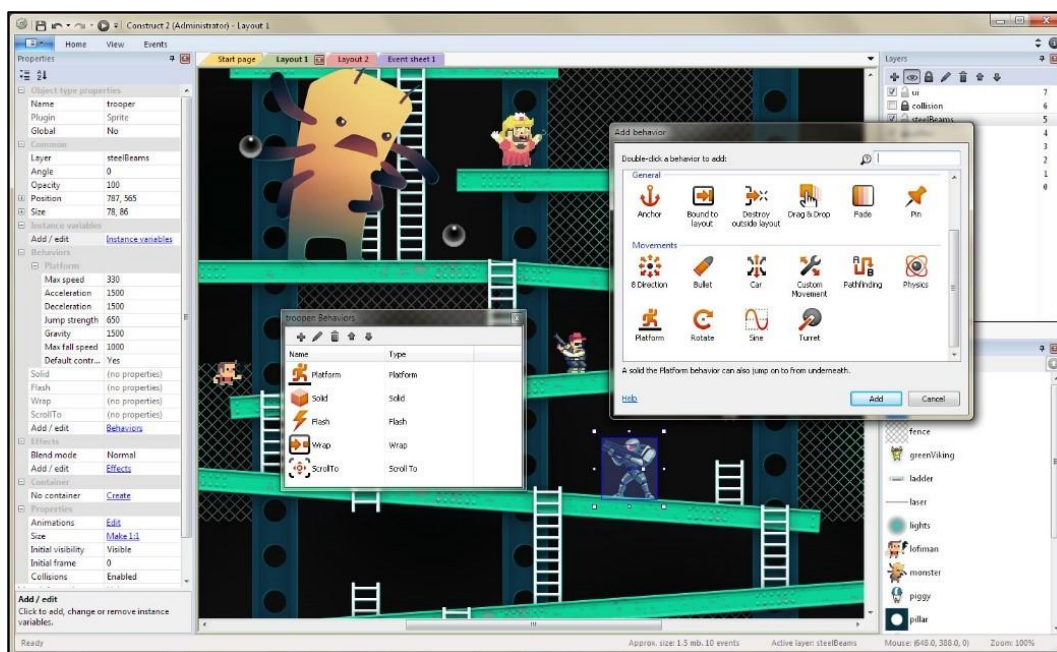
Atualmente existem diversas *game engines* completas, algumas delas são gratuitas, *open source*² ou comercializadas. Abaixo são apresentadas algumas destas *game engines*:

- **Construct 2:** criada pela empresa Scirra³ e lançada para o público em 2007, a *Construct 2* (Figura 3) permite a criação de jogos digitais multiplataforma em 2D baseados em *HTML5* (*HyperText Markup Language* – versão 5). Os *games* feitos nela podem funcionar em *smartphones*, *tablets*, computadores, navegadores e o *console* Wii U (DIAS, 2016);

² *Open source* refere-se a algo que as pessoas podem modificar e compartilhar porque seu design é acessível ao público (OPEN SOURCE, 2017).

³ <https://www.scirra.com>

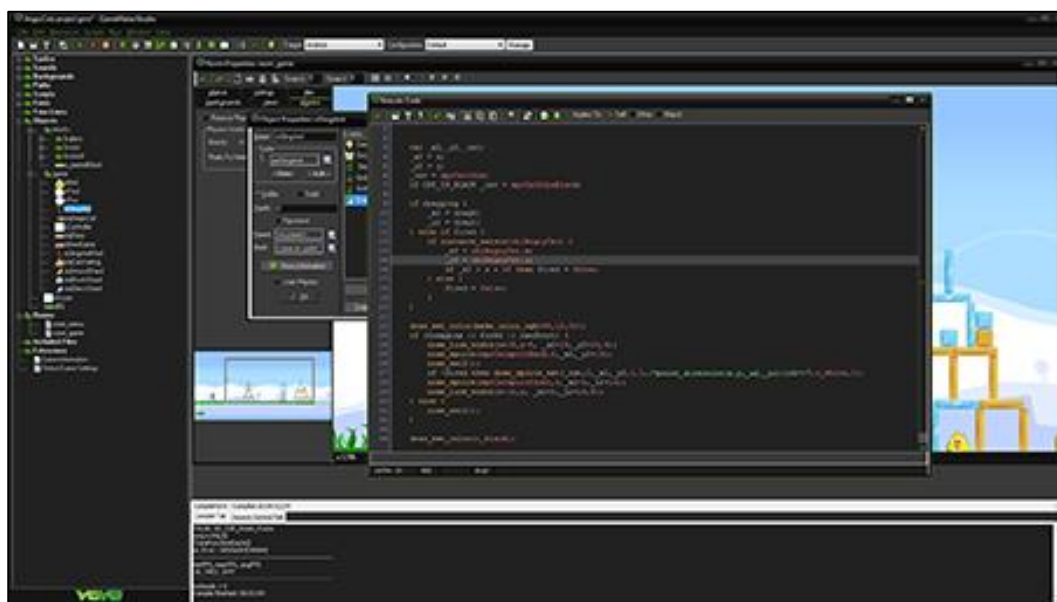
Figura 3 – Ambiente de desenvolvimento do Construct 2.



Fonte: (SCIRRA, 2017).

- **Game Maker**: desenvolvida pela empresa YoYo Games⁴, a *Game Maker* (Figura 4) indicada para iniciantes no desenvolvimento de jogos, onde não é preciso saber programar para iniciar um projeto, devido ao sistema “arrastar e soltar”, disponibilizado pela *game engine* (DIAS, 2016);

Figura 4 – Ambiente de desenvolvimento do GameMaker.

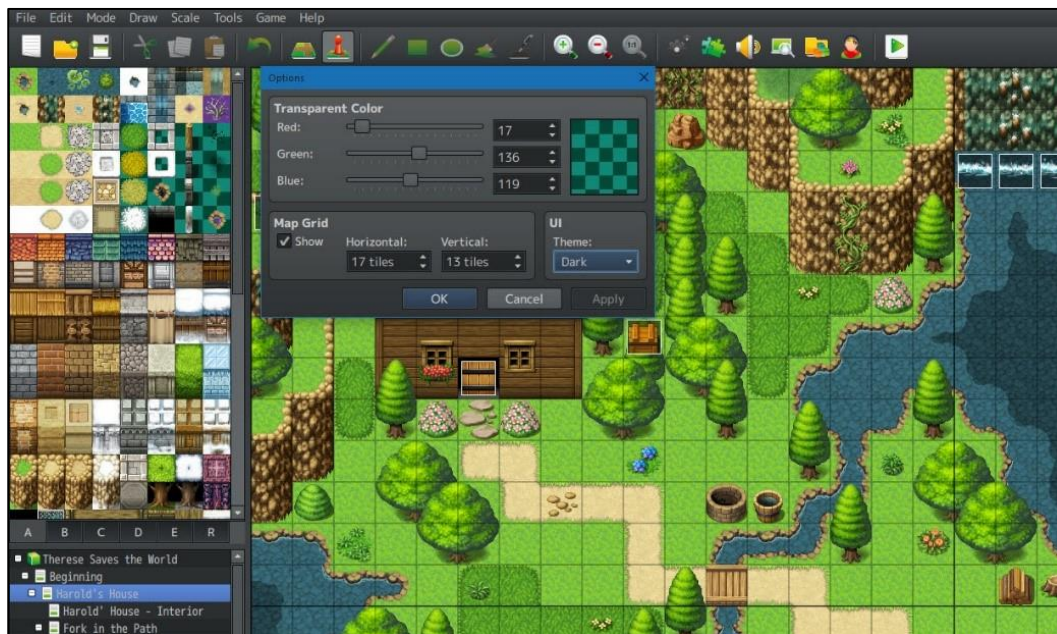


Fonte: (GAMES, 2017).

⁴ <https://www.yoyogames.com>

- **RPG Maker**: consiste em uma série de recursos (Figura 5) para a criação de jogos no estilo *RPG (Role Playing Game)*. O *RPG Maker*⁵ foi criado pela empresa ASCII e hoje pertence à associada Enterbrain (DIAS, 2016);

Figura 5 – Ambiente de desenvolvimento do RPG Maker.



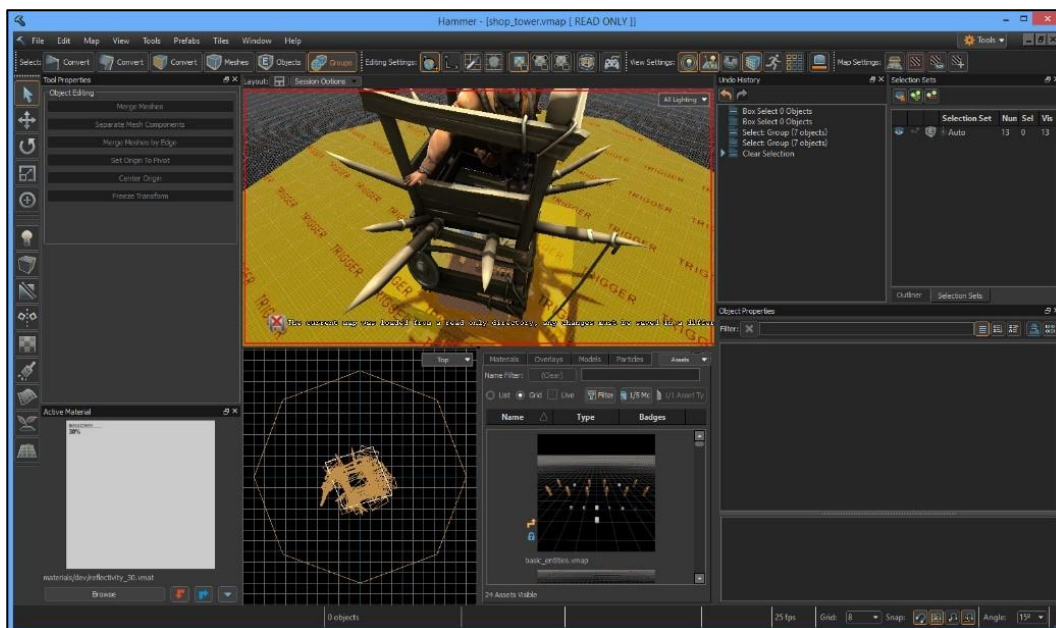
Fonte: (MAKER, 2017).

- **Source 2**: desenvolvida pela Valve Corporation⁶ como sucessor do motor *Source Engine*. *Source 2* (Figura 6), distribuído gratuitamente para desenvolvedores, foi anunciado em 2015, utilizando um novo motor de física, chamado Rubikon. O primeiro jogo lançado utilizando o motor *Source 2* foi o jogo *Dota 2*, seguido do jogo *Counter Strike: Global Offensive* (VALVE, 2017);

⁵ <http://www.rpgmakerweb.com>

⁶ <http://www.valvesoftware.com>

Figura 6 – Ambiente de desenvolvimento do Source 2.

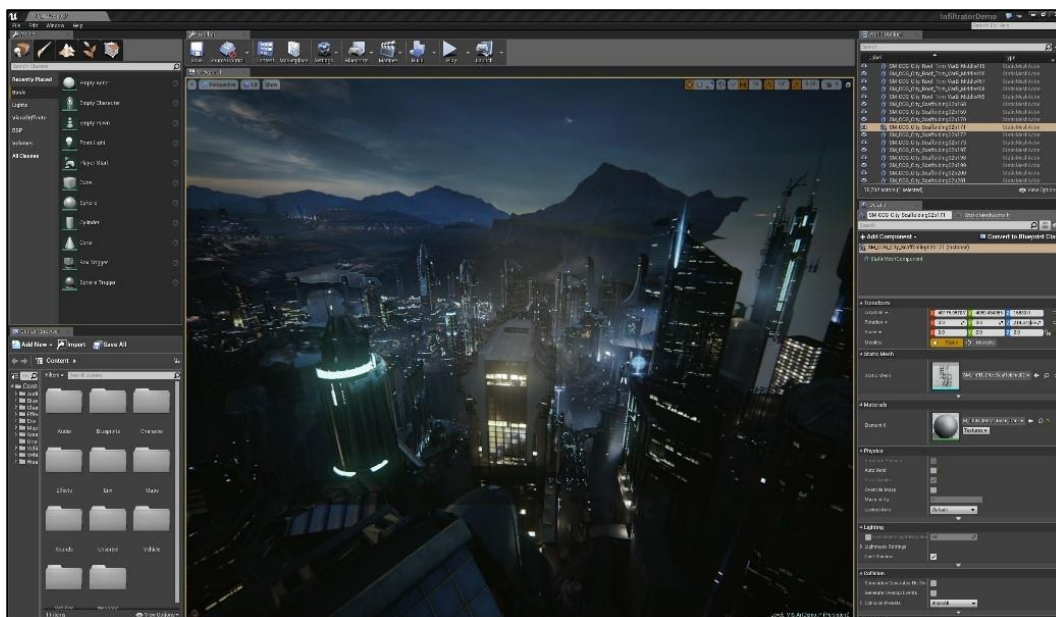


Fonte: (VALVE, 2017).

- Unreal Engine 4:** desenvolvida pela Epic Games⁷, escrita em C++, oferecendo um alto grau de portabilidade. Grande parte da *engine* é escrita usando o *UnrealScript*, uma linguagem de *script* proprietária. Uma das principais características visuais da *Unreal Engine* (Figura 7) é o uso do método de renderização para calcular a iluminação de cenas computadorizadas com grande precisão e realismo, conhecido *High Dynamic Range Rendering* ou *High Dynamic Range Lighting* (LIMA, 2008);

⁷ <https://epicgames.com>

Figura 7 – Ambiente de desenvolvimento do Unreal Engine 4.



Fonte: (ENGINE, 2017).

- **Unity 3D:** desenvolvida pela *Unity Technologies*⁸, é uma *game engine* (Figura 8) voltada para o mercado comercial, possui gráficos de última geração e permite tanto o desenvolvimento de jogos para computadores, quanto para *consoles* e dispositivos móveis (LIMA, 2008).

Figura 8 – Ambiente de desenvolvimento do Unity 3D.



Fonte: (UNITY3D, 2017).

⁸ <https://unity3d.com>

2.3.2.1 Unity

A *Unity* é um *game engine* proprietária e uma IDE criada pela *Unity Technologies*. Assim como toda *game engine*, ela facilita o desenvolvimento de jogos pelo fato do desenvolvedor não precisar programar diretamente para *DirectX* ou *OpenGL*, pois ela já faz isso automaticamente (KUCERA, 2013).

A *Unity* se tornou uma das *engines* mais populares entre desenvolvedores independentes após seus criadores terem liberado uma edição gratuita (com algumas limitações) em 2009. *Games* de franquias famosas, como *Angry Birds*, foram feitos usando essa *game engine* (DIAS, 2017).

A *Unity* pode fazer jogos para produtos da *Apple* (*Mac*, *iPhone*, *iPod*, *iPad*), da *Microsoft* (*Xbox*, *Windows*), da *Google* (dispositivos com *Android*), da *Sony* (*Playstation 3*), da *Nintendo* (*Wii*) e para navegadores *Web* (*Internet Explorer*, *Mozilla Firefox*, *Google Chrome*, *Opera* e *Safari*) (KUCERA, 2013).

Ela possui uma ferramenta de *scripts* baseada no Mono⁹, possibilitando a programação em *C#* e *UnityScript* (conhecido como *JavaScript* pelo *software*). A *Unity* importa em vários formatos, tanto 2D quanto 3D, o que elimina o trabalho de exportar alguma imagem digital antes de importar na *Unity*. Ela aceita os arquivos dos seguintes programas (KUCERA, 2013):

- *Blender*;
- *Autodesk 3DS Max*;
- *Autodesk Maya*;
- *Maxon CINEMA 4D*;
- *Cheetah3D*;
- *Luxology Modo 3D*;
- *NewTek LightWave*;

A *Unity* também lê arquivos com formato *FBX*, *DAE*, *3DS*, *DXF*, *OBJ* e também aceita o *COLLADA*. Para arquivos de textura (arquivos 2D) ela aceita os formatos: *PSD*, *TIFF*, *JPG*, *TGA*, *PNG*, *GIF*, *BMP*, *IFF*, *PICT*. Esses *assets* (recursos)

⁹ Mono é uma ferramenta para desenvolver e executar aplicações .NET em diferentes plataformas (KUCERA, 2013).

podem ser adicionados ao projeto de jogo, e utilizados através da *interface* gráfica da *Unity* (KUCERA, 2013).

Outra grande vantagem do *Unity* é a disponibilização de ferramentas de aprendizado para o desenvolvedor. No *site* (<https://unity3d.com>) da ferramenta estão disponíveis vários tutoriais, além de toda a documentação necessária para o desenvolvedor utilizar as classes do *Unity* em seus *scripts* (MACHADO, 2014).

3 METODOLOGIA DE PESQUISA

A pesquisa realizada foi conduzida para se verificar quais métodos e técnicas do *game design* podem ser aplicadas ao jogo eletrônico, de forma que o mesmo auxilie o professor no processo de ensino/aprendizagem, servindo de apoio (para o professor) em salas de aula inclusivas. Por fim, elaborar e desenvolver o projeto de um jogo educativo.

Para levantamento das metodologias de ensino ativas, utilizou-se a pesquisa bibliográfica. Posteriormente, foram especificados requisitos, por meio do *game design documento* (GDD), para a elaboração do projeto de jogo eletrônico.

Utilizando pesquisa bibliográfica, será levantado métodos e técnicas do *game design*, a fim de elaborar o projeto do jogo com base nos requisitos definidos anteriormente e desenvolver o jogo utilizando a *game engine Unity*.

Ao final da pesquisa obteve-se informações suficientes acerca das técnicas existentes no âmbito de *game design*, que permitiram o desenvolvimento de um jogo eletrônico, no qual tem como objetivo servir de instrumento de apoio no processo de ensino, aumentar o engajamento e a motivação dos alunos dentro no âmbito da educação inclusiva. Desta forma, a metodologia permitiu atingir os objetivos (geral e específicos) da pesquisa. É necessário que seja realizado testes em relação à eficiência do jogo em salas de aula inclusivas, contudo, tais experimentos não são contemplados neste trabalho, tornando-se um trabalho futuro.

4 PROJETO DO JOGO

Com base nos estudos apresentados no Referencial Teórico, neste capítulo é apresentado o projeto de jogo eletrônico. Para melhor organização, é apresentado o *game design document* (GDD), contendo as informações do jogo construído, e os detalhes do desenvolvimento do jogo por meio da *game engine Unity*.

4.1 *BaseKids*

Para este estudo foi desenvolvido o projeto de um jogo eletrônico, denominado “*BaseKids*”. O jogo *BaseKids* foi desenvolvido por meio do Concurso INOVApps¹⁰ (edital de 2015), realizado pelo Ministério das Comunicações e pelo Ministério da Ciência, Tecnologia, Inovações e Comunicações, sendo um dos jogos aprovados pelo concurso.

Figura 9 – Logo do jogo *BaseKids*.



Fonte: Equipe *BaseKids*.

O jogo *BaseKids* foi desenvolvido por uma equipe, formada pelos integrantes:

- Giovanni Batista Moreira Pocivi (Narração do Personagem);
- João Pedro Lima Ribeiro (Designer/Idealização);
- Karina Hellen Neves da Silva (Desenvolvimento/Elaboração);
- Leandro da Silva Câmara (Desenvolvimento/Elaboração);
- Lucas Antônio dos Reis Braga (Desenvolvimento/Elaboração);
- Matheus G. Bronca Arantes (Designer/Idealização);

Embora tenha sido elaborado um GDD para desenvolvimento do jogo (para o INOVApps), este trabalho utilizou referências de diversos autores, a fim de

¹⁰ <http://www.brasil.gov.br/infraestrutura/2015/09/governo-seleciona-505-projetos-para-concurso-de-aplicativos>

fundamentar o desenvolvimento do jogo e relacionar com métodos e técnicas do *game design* e a Taxonomia de Bloom.

4.1.1 Classificação de aprendizagem

O jogo *BaseKids* explora o nível “Aplicar”, descrito na Taxonomia de Bloom (item 2.1.2), sendo útil para prover uma avaliação da aprendizagem do jogador. Relacionando com as classificações propostas por Bloom, o jogo tem como objetivo pedagógico:

- **Lembrar:** instruir o jogador, por meio de tutoriais ou sugestões (por intermédio do personagem), como é o funcionamento do jogo (mecânicas, interfaces, objetivos das fases). Após as instruções, o jogador deve, reconhecendo a necessidade, reproduzir ações demonstradas nos tutoriais;
- **Entender:** quando o jogador solicita que o personagem lhe dê uma dica, para que o ajude a reconhecer o objeto solicitado, o jogo disponibiliza ao jogador uma dica, por meio de texto e áudio, de forma que o mesmo deva interpretar e comparar as características dos objetos disponíveis, afim de identificar o objeto solicitado;
- **Aplicar:** por meio das dicas e das curiosidades presentes no jogo, o jogador pode utilizar as mesmas informações em seu cotidiano. Tais informações podem auxiliá-lo no reconhecimento de objetos reais.

Durante a elaboração do jogo, foram realizadas algumas entrevistas com professores da Associação de Pais e Amigos dos Excepcionais (APAE) de Anápolis, onde os mesmos fizeram sugestões, de modo que, o jogo pudesse ser útil às crianças atendidas pela APAE. Com o objetivo de aumentar a aplicabilidade do jogo na educação inclusiva, foram atendidas as sugestões feitas pelos professores:

- Utilizar nos textos, apenas **letras maiúsculas**, pois algumas crianças tinham dificuldades em diferenciar letras minúsculas de letras maiúsculas;
- **Utilizar áudios** (por meio do personagem) de forma que, caso uma criança não soubesse ler, ou que tivesse dificuldades na leitura, pudesse compreender os objetivos e conhecimentos proporcionados no jogo;
- O personagem interagir **constantemente** (por meio de áudios) com o jogador, pois algumas crianças tinham dificuldades de se manterem focadas nas atividades propostas pelos professores. Desta forma, caso o

jogador tivesse dificuldades para identificar os objetos na prateleira, o personagem iria auxiliar o jogador por meio da dica;

- Utilizar imagens que se aproximem da **realidade** (objetos do supermercado) para que a criança pudesse associar o nome do objeto à imagem;
- Jogo possua comandos que estimule a **coordenação motora**. Para tal sugestão, foi definido que, em algumas fases, o jogador teria que arrastar os objetos ao invés de simplesmente selecioná-los.
- Apresente um **tutorial** indicando o que se deve fazer (na verdade era a dificuldade dos professores, em entender o que deve ser feito no jogo, para depois explicar para a criança).

4.1.2 GDD

Para descrever as características do jogo foi construído a especificação do mesmo, por meio do *Game Design Document* (GDD). Nesta seção são apresentados o conceito e o *gameplay* (e os elementos que o contemplam).

Os elementos abordados no GDD foram selecionados (a partir do item 2.2) com base nas características do jogo, mesmo que autores sugeriram a utilização de vários elementos. Por exemplo, a especificação de itens como “armas”, “golpes” e “inimigos” não são necessários para o GDD do jogo *BaseKids*.

4.1.2.1 Conceito

O jogo ***BaseKids*** se passará na parte interna de um supermercado, onde existem variedades de alimentos e objetos. É composto por nove fases, ou seja, o supermercado possui nove setores diferentes, e a cada três fases seguintes o nível de dificuldade aumenta.

O personagem do jogo irá interagir constantemente com o jogador, solicitando que selecione ou arraste (variando de acordo com os níveis de fases) os itens presentes na prateleira, instruindo o jogador a alocá-los dentro do carrinho de compras. Para cada item selecionado corretamente, são apresentadas informações e curiosidades sobre o mesmo, proporcionando ao jogador aprendizado e interatividade. Será disponibilizada uma dica em relação ao item para auxiliar na escolha do jogador.

4.1.2.2 *Gameplay*

Nesta seção são abordados o enredo, a descrição sobre o personagem, as mecânicas do jogo, as possíveis interações entre o jogo e o jogador, assim como os desafios encontrados pelo jogador e quais os métodos usados para superá-los.

4.1.2.2.1 Personagem

Há apenas um personagem presente no jogo (Figura 10), ele é representado por um garoto. Este personagem irá interagir constantemente com o jogador, orientando-o em suas escolhas.

Figura 10 – Personagem do jogo BaseKids.



Fonte: Equipe *BaseKids*.

4.1.2.2.2 Voz do personagem

A voz do personagem é uma voz infantil, no intuito de promover uma interatividade maior com o jogador. A criança responsável por gravar os áudios das falas do personagem foi Giovanny Batista Moreira Pocivi (gravadas no ano de 2016).

4.1.2.2.3 Enredo

O jogador poderá escolher uma das fases que estarão dispostas em ordem sequencial no “Mapa de Fases”. Cada fase refere-se à um dos setores do supermercado, havendo uma prateleira com os itens distribuídos, de acordo com o setor do supermercado.

As três primeiras fases (1 a 3) possui como dificuldade o nível “fácil”, enquanto as 3 fases seguintes (4 a 6) são do nível “médio” e as demais fases (7 a 9) são do nível “difícil”. Nas fases fáceis, é necessário apenas selecionar (tocar) os objetos solicitados. A partir das fases de nível médio, é necessário arrastar os objetos, enquanto no nível difícil, o personagem solicita os objetos por meio de características (por exemplo, a cor) do mesmo, sem dizer o nome, pois o nome do objeto está disponível apenas como “dica”.

O personagem cumprimenta e apresenta à fase ao jogador. Em seguida, o personagem solicita que o jogador identifique os objetos na prateleira e os coloque no carrinho de compras, sendo necessário selecionar ou arrastar (dependendo da dificuldade da fase).

Caso o jogador tenha dificuldade na identificação dos objetos, o personagem ajuda-o, indicando o botão de “Dica”, a fim de que jogador tenha acesso à alguma característica do objeto, facilitando sua identificação.

Identificado o objeto correto, o jogador poderá formar o nome do objeto, juntando as sílabas, podendo selecionar a sílaba ou arrastá-la (dependendo da dificuldade da fase). Ao final, formada a palavra corretamente, o personagem dirá ao jogador uma curiosidade sobre o objeto, de forma que o jogador possa aprender mais sobre os itens presentes no supermercado.

4.1.2.2.4 Fluxo do jogo

Em relação ao fluxo (Figura 11) do jogo BaseKids, inicialmente, o jogador visualizará a tela principal, com as opções “Jogar”, “Créditos” e “Sair”, além da opção de “ativar/desativar” a música do jogo.

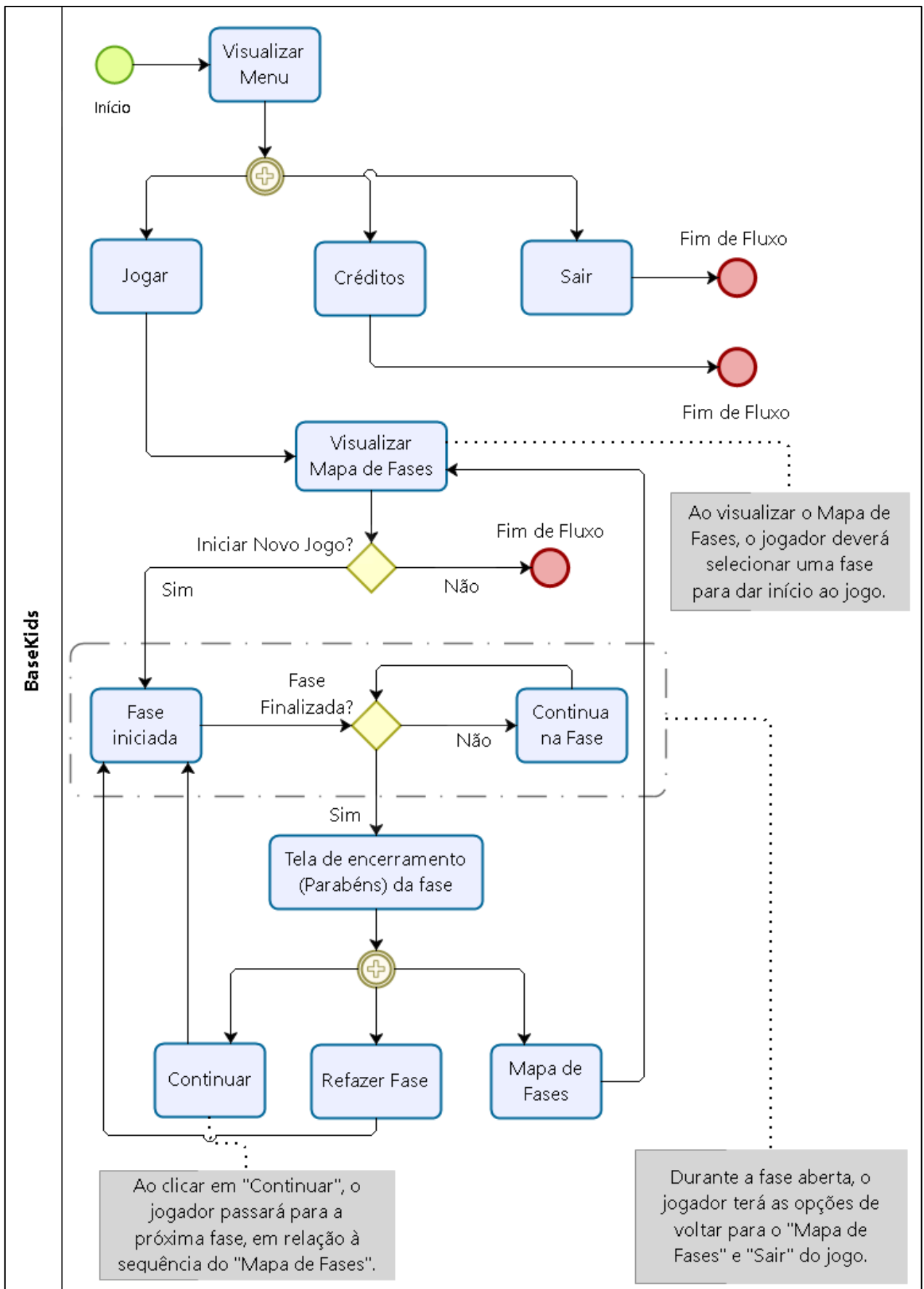
Selecionando a opção “Jogar”, o jogador é redirecionado. Selecionando a opção “Créditos”, o jogador é redirecionado para a tela com informações da equipe e com a lista de ferramentas utilizadas para desenvolvimento do jogo. Selecionando a opção “Sair”, é apresentado uma interface, para confirmar se o jogador deseja sair do jogo.

Durante a fase, o jogador deve alocar no carrinho o objeto solicitado pelo personagem, além de possuir as opções de “Sair” do jogo, retornar ao “Mapa de Fases” e “Continuar” na fase.

Ao final da fase, o jogador é redirecionado para a tela de “Parabéns” com a possibilidade de realizar três escolhas:

- Retornar ao “Mapa de Fases”;
- Refazer a fase;
- Continuar (O jogador dá sequência as fases de acordo com a ordem presente no “Mapa de Fases”).

Figura 11 – Fluxograma do jogo BaseKids.



Fonte: Autor da pesquisa.

Como demonstração (Tabela 1) do fluxo básico do jogo, foi descrito as ações do personagem, do jogador e do sistema em relação à situação em que o jogador escolhe corretamente um dos objetos da Fase 1 (Massas e Outros).

Tabela 1 – Demonstração do Fluxo Básico.

Fluxo Básico – Fase 1		
Personagem	Jogador	Sistema
- Olá, seja bem-vindo ao BaseKids! Quero muito brincar com você. Selecione uma fase e vamos nos divertir.	Seleciona a fase desejada.	Redireciona para a fase selecionada.
- Bem-vindo a Fase “Massas e Outros”! - Vamos brincar? - Selecione os ovos para o carrinho.	Seleciona o item correto para o carrinho.	Verifica que o item está correto e redireciona para a tela de formação da palavra.
- Muito bem! Agora, forme a palavra “Ovos”.	Forma a palavra colocando as sílabas na ordem correta.	Valida individualmente as sílabas e verifica que a palavra está correta.
- Parabéns, a palavra está correta!	N/A	N/A
[Curiosidade] - Você sabia que o ovo contém proteínas de alto valor biológico e é fonte de vitamina B12, ferro, fósforo, manganês, potássio e sódio?	N/A	Verifica se o item atual é o último item da fase (a ser selecionado). Se sim, redireciona para a tela “Parabéns”.
- Agora vamos para o próximo item.	N/A	Redireciona para a fase atual.

- Selecione o macarrão para o carrinho.	Seleciona o item correto para o carrinho.	Verifica que o item está correto e redireciona para a tela de formação da palavra.
---	---	--

Fonte: Autor da pesquisa.

4.1.2.2.5 Progressões / regras do jogo

Nesta seção são especificadas as regras e as condições para que o jogador tenha progresso no jogo. As regras presentes no jogo *BaseKids* são descritas nos seguintes itens:

- Caso o jogador não escolha o objeto correto, ele terá a opção de tentar novamente. Não terá um limite de tentativas.
- Não haverá sistema de pontuação.
- O jogador poderá jogar qualquer fase em uma quantidade de vezes que desejar, sem restrições.
- Durante a “Fase Aberta”, todas as fases conterão como indicação seu respectivo nome.
- Caso o jogador arraste o objeto para o local incorreto, ou seja, não arraste para dentro do carrinho, o objeto voltará para sua posição inicial.
- Caso o jogador arraste as sílabas da palavra para um local incorreto, ou seja, não arraste para o campo em branco, a sílaba voltará para sua posição inicial.
- Se o jogador não selecionar nenhum objeto em um prazo de 20 segundos, o personagem irá solicitar que o jogador acesse o botão “Dica” para auxiliar a sua escolha;
- No momento em que o jogador estiver formando a palavra, arrastando as suas sílabas até o campo em branco, haverá a verificação das sílabas, validando se a resposta está correta ou errada.
- Caso a sílaba alocada esteja correta, o seu contorno ficará com a cor verde. Caso esteja errada, o seu contorno ficará com a cor vermelha e em seguida retornará à sua posição original.

4.1.3 Implementação

Para o desenvolvimento do jogo, foi utilizado a *game engine Unity*. Esta possui uma ferramenta interna chamada de Editor. Com essa ferramenta é possível navegar pelo mundo do jogo em questão, definir a posição de cada objeto do jogo, inspecionar cada um destes objetos para verificar seus componentes, ter uma pré-visualização do jogo em questão, visualizar *logs*, entre outros.

A *Unity* segue uma arquitetura padronizada denominada “programação orientada a composição de componentes” (HASSELBRING, 2006). Esse formato apresenta o conceito de que o comportamento de determinado objeto deve ser composto por diversos componentes. Dessa forma, se for necessário mudar o comportamento de determinado objeto, basta apenas retirar o componente que causa o comportamento que se deseja eliminar ou substituir o componente por outro com o comportamento desejado.

A composição de componentes pode oferecer grandes vantagens para o desenvolvimento de jogos. Algumas dessas vantagens são: reusabilidade de comportamentos, fácil customização, adaptação e fácil manutenção de um objeto de jogo.

4.1.3.1 Estrutura do projeto

O projeto foi estruturado (Figura 12) de forma a organizar as animações, áudios, cenas, *design* (desenhos), fontes, *materials* (materiais) e *scripts* (código-fonte).

Figura 12 – Estrutura do projeto.

Fonte: Equipe *BaseKids*.

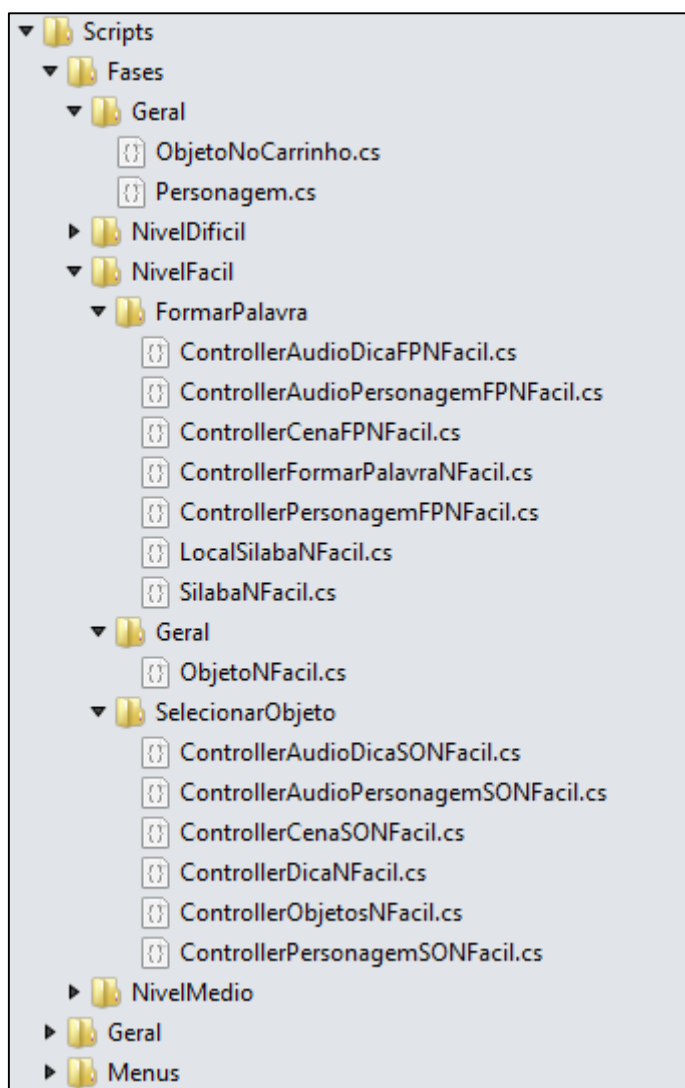
A pasta “*Animations*” contém as animações do personagem, sendo as animações da boca, dos olhos e dos braços. Os áudios utilizados nas fases e cenas do jogo foram organizados dentro da pasta “*Audios*”, incluindo os áudios do personagem, dicas e curiosidades.

As cenas do jogo (incluindo os menus e fases) estão organizadas na pasta “*Cenas*”. Na pasta “*Design*”, contém as imagens utilizadas no jogo, como o personagem, objetos, botões, entre outros.

Na pasta “*Fontes*” contém a fonte utilizada nos textos do jogo. Planos de fundo estão organizados na pasta “*Materials*”. Enquanto o código-fonte dos componentes implementados (utilizados nos elementos do jogo), na linguagem de programação C#, estão na pasta “*Scripts*”.

Os componentes presentes nas fases foram separados por níveis. Na Figura 13, está visível os componentes das fases de nível fácil, entretanto, os componentes dos demais níveis seguem a mesma estrutura, mudando apenas o nome do arquivo.

Figura 13 – Organização dos componentes das fases.



Fonte: Equipe *BaseKids*.

Em todas as fases, estão sendo utilizados os componentes (Figura 13) “ObjetoNoCarrinho” e “Personagem”. O componente “ObjetoNoCarrinho” é responsável por manter o objeto no carrinho, caso o usuário tenha acertado a escolha do objeto, enquanto o componente “Personagem” é responsável por manter todos os áudios que podem ser ditos pelo personagem durante a cena da fase. Os áudios variam de acordo com a fase e cena do jogo.

Os componentes (Figura 13) utilizados nas cenas de “formação de palavra” das fases de nível fácil são responsáveis por manter e reproduzir os áudios da dica e do personagem, gerenciar a cena (incluindo a tela de opções), gerenciar as sílabas em relação aos locais onde o jogador aloca as mesmas, verificando se a palavra formada está com as sílabas ordenadas corretamente, gerenciar as animações, o

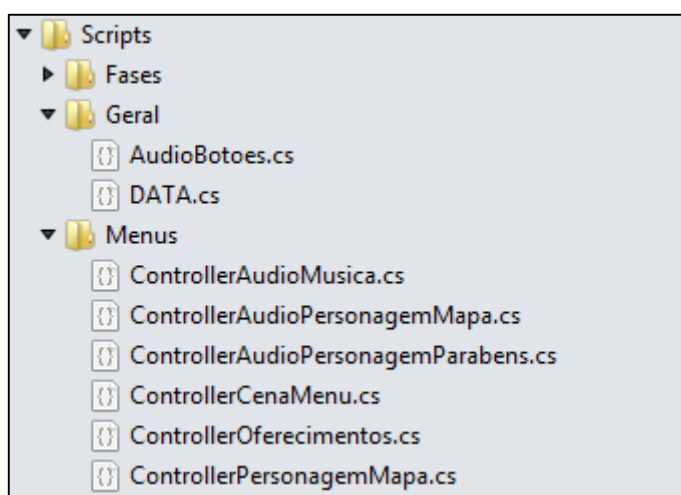
espaço (bloco) para o onde o jogador pode alocar a sílaba e gerenciar os comportamentos das sílabas (por meio de eventos acionados pelo jogador).

O componente (Figura 13) utilizado em todas as cenas das fases de nível fácil, responsável pelo gerenciamento dos comportamentos e informações dos objetos, como textos e áudios (referente ao nome do mesmo), dicas e curiosidades, é o componente “ObjetoNFacil”.

Os componentes (Figura 13) utilizados nas cenas de “seleção do objeto” das fases de nível fácil são responsáveis por manter e reproduzir os áudios da dica (inclusive os textos) e do personagem, gerenciar a cena (incluindo a tela de opções), gerenciar os objetos da prateleira, verificando se o jogador acertou o objeto solicitado pelo personagem, e gerenciar as animações do personagem.

Os componentes (Figura 14) gerais e das interfaces de menu (inclusive da tela de oferecimento) são responsáveis por gerenciar o botão que “habilita/desabilita” a música do jogo, armazenar informações compartilhadas por todas as cenas (por exemplo, se os tutoriais foram realizados), gerenciar as músicas reproduzidas no jogo, manter e reproduzir os áudios (nas telas “Mapa de Fases” e “Parabéns”) da dica e do personagem, gerenciar as cenas (“Oferecimentos”, “Tela Inicial”, “Mapa de Fases”, “Créditos” e “Parabéns”) e as animações do personagem.

Figura 14 – Organização dos componentes gerais e de interfaces de menu.

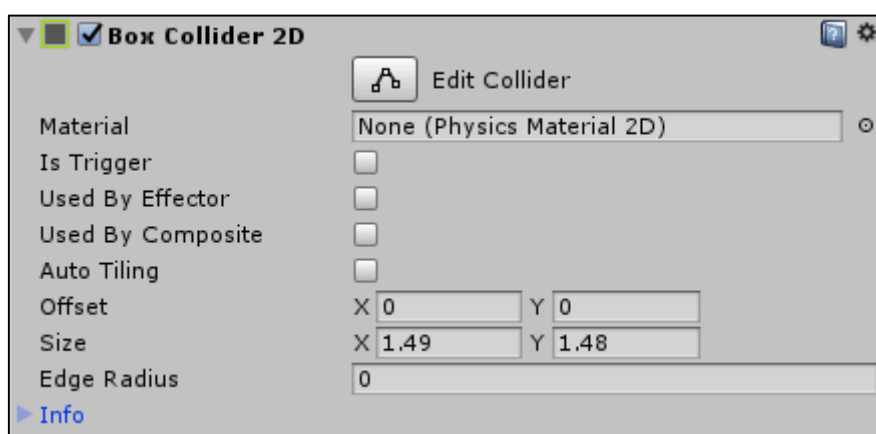


Fonte: Equipe *BaseKids*.

4.1.3.2 *Box Collider 2D*

O componente *Box Collider 2D* (Figura 15) é um componente usado com física 2D. Tal componente é utilizado quando se deseja detectar colisões sobre um objeto. Sua forma é um retângulo com uma posição, largura e altura, definidas no espaço de coordenadas do objeto gráfico (também denominado como *Sprite*). Este componente foi utilizado para detectar os cliques ou toques do jogador sobre os objetos da prateleira.

Figura 15 – Componente *Box Collider 2D*.



Fonte: Autor da pesquisa.

4.1.3.3 *IEventSystemHandler*

O sistema de eventos da *Unity* suporta uma série de eventos, que podem ser, inclusive, customizados pelo programador. Os eventos utilizados implementam as interfaces que herdam da interface principal (abstrata) - *IEventSystemHandler*.

Para identificar quando o jogador clica/toca em um objeto, é utilizado o método “*OnPointerDown*”, implementado da interface “*IPointerDownHandler*”. O método “*OnPointerUp*” é acionado quando o ponteiro é liberado (do objeto) após o evento de clique/toque, sendo este implementado da interface “*IPointerUpHandler*”.

Após o clique/toque sobre o objeto e antes da ação de “arrastar” ser iniciada, o método “*OnBeginDrag*” é acionado (por meio da implementação da interface “*IBeginDragHandler*”). O método “*OnDrag*” (implementado da interface “*IDragHandler*”) é executado enquanto o objeto está sendo arrastado, enquanto o método “*OnEndDrag*” (implementado da interface “*IEndDragHandler*”) é acionado quando a ação “arrastar o objeto” é concluída.

Quando um outro objeto é solto sobre o objeto que o implementa a interface “*IDropHandler*”, o método “*OnDrop*” é acionado.

5. RESULTADOS

Inicialmente a *game engine Construct2* estava sendo utilizada para construção do jogo *BaseKids*, por motivos relacionados à experiência da equipe (responsável pelo desenvolvimento do jogo) com a tecnologia e pela simplicidade que a ferramenta proporcionava no desenvolvimento de jogos 2D, por abstrair a programação.

Entretanto, durante o desenvolvimento, na realização dos testes, foi identificado que, dependendo da forma como o jogo era compilado pela *Construct2*, a funcionalidade “Sair do Jogo” não funcionava ou o dispositivo (móvel) era bloqueado (suspensão) quando o jogador passava alguns segundos sem interagir com o mesmo, o que comprometeria diretamente a qualidade do *software*. Dependendo da dificuldade do jogador, especificamente em relação ao tempo necessário para concluir uma ação, quando o dispositivo fosse suspenso por inatividade o jogador teria que desbloquear o dispositivo.

Diante dos problemas apresentados e da dificuldade em encontrar soluções, a equipe decidiu substituir a *Construct2* pela *Unity*, como *game engine* utilizada para o desenvolvimento do jogo *BaseKids*, pelos seguintes fatores:

- A ferramenta *Unity* possui uma documentação organizada e com muitos tutoriais;
- A equipe tinha conhecimento da linguagem de programação JAVA. Na ferramenta *Unity* há suporte para uso da linguagem de programação C#. Na visão da equipe, isso facilitaria em relação à curva de aprendizagem, devido às familiaridades entre JAVA e C#, e principalmente em relação ao prazo para entrega do projeto;

Constatou-se que, a *game engine Unity* possui uma comunidade muito grande e ativa, o que contribuiria nos casos de sugestões e ajuda por motivos de erros e dificuldade na implementação de funcionalidades;

O jogo BaseKids foi desenvolvido por meio da *game engine Unity*, baseando-se no *Game Design Document (GDD)*, elaborado conforme os métodos e técnicas do *game design*.

A tela inicial do jogo (Figura 16) apresenta o personagem, a logo do jogo, as opções de menu e o botão de “habilitar/desabilitar” a música.

Figura 16 – Tela inicial.



Fonte: Equipe *BaseKids*.

No mapa de fases (Figura 17) foi especificado, conforme a Tabela 2, os níveis de dificuldades para as fases, sendo definido o nível fácil para as fases 1, 2 e 3, o nível médio para as fases 4, 5 e 6, e nível difícil para as fases 7, 8 e 9.

Tabela 2 – Mapa de Fases.

Mapa de Fases		
Fácil	Médio	Difícil
[Fase 1] Massas e Outros	[Fase 4] Papeleria	[Fase 7] Moda
[Fase 2] Verduras e Frutas	[Fase 5] Eletrodomésticos	[Fase 8] Utensílios Domésticos
[Fase 3] Bebidas	[Fase 6] Brinquedos	[Fase 9] Eletrônicos

Fonte: Autor da pesquisa.

Nesta tela (Figura 17), as fases são organizadas pelo nível de dificuldade – fácil (verde), médio (amarelo) e difícil (vermelho). O personagem cumprimenta o jogador e convida-o a jogar, selecionando uma das fases.

Figura 17 – Mapa de Fases.



Fonte: Equipe BaseKids.

Na tela de “seleção dos objetos” (Figura 18), o jogador deve alocar os objetos solicitados (pelo personagem) para o carrinho de compras. O nome da fase (setor do supermercado) é apresentado acima da prateleira, para melhor identificação da fase. Nesta tela, são disponíveis os botões da “Dica”, de “habilitar/desabilitar” a música do jogo, menu de opções (*Pause*) e o botão (*Play*) responsável por fazer com que o personagem repita no nome do objeto solicitado. Os objetos são disponíveis na prateleira.

Figura 18 – Fase “Massas e Outros”.



Fonte: Equipe BaseKids.

Na tela de “formação de palavra” (Figura 19), o jogador deve selecionar ou arrastar (dependendo do nível de dificuldade da fase) as sílabas, de forma que sejam organizadas em sequência, a fim de formar o nome do objeto. Caso o jogador esteja arrastando as sílabas, as mesmas devem ser alocadas até o espaço reservado para as sílabas, tais locais estão disponíveis na parte inferior central da cena, com o fundo na cor branca. Os locais com cor marrom e com transparência estão desabilitados, de forma que seja indicado ao jogador em qual local o mesmo deve alocar a sílaba escolhida. Nesta cena, são disponíveis os botões de “habilitar/desabilitar” a música do jogo, menu de opções (*Pause*) e o botão (*Play*) responsável por fazer com que o personagem repita no nome do objeto solicitado.

Figura 19 – Formação de palavra.



Fonte: Equipe *BaseKids*.

O jogo *BaseKids* foi desenvolvido visando a disponibilidade em dispositivos móveis e navegadores. A *game engine Unity* permitiu que o jogo fosse compilado para as plataformas *Android* e *WebGL*. Em relação à plataforma *Android*, para instalação do jogo no dispositivo, é necessário que a versão do sistema operacional seja 4.1 ou superior, enquanto para plataforma *Web*, o jogo é compatível com navegadores, conforme definido pelas especificações¹¹ no manual da *Unity*:

- Mozilla Firefox (versão 52 ou superior);
- Google Chrome (versão 57 ou superior);
- Apple Safari (versão 11 ou superior);
- Microsoft Edge (versão 16 ou superior);

¹¹ <https://docs.unity3d.com/Manual/webgl-browsercompatibility.html>

6 CONSIDERAÇÕES FINAIS

Este estudo teve como objetivo a construção de um jogo eletrônico educativo, que pudesse servir como instrumento de apoio no processo de ensino/aprendizagem, no engajamento e motivação dos alunos no âmbito da educação inclusiva.

Por meio das pesquisas bibliográficas relacionadas às metodologias de ensino, foi possível compreender os conceitos que englobam o processo de ensino/aprendizagem no que se refere à educação inclusiva e a Taxonomia de Bloom, permitindo assim perceber as dificuldades enfrentadas no contexto escolar.

Realizando um levantamento dos métodos e técnicas de *game design*, permitiu-se compreender o objetivo e os elementos do *game design*, assim como a estrutura que contemplam o *Game Design Document* (GDD), e a *game engine Unity*.

Foi elaborado o GDD do jogo *BaseKids*, baseando-se nos estudos acerca do *game design* e seus elementos, definindo o conceito, a classificação de aprendizagem e o *gameplay* do jogo. Utilizando a *game engine Unity*, foi feita a implementação do jogo seguindo a arquitetura “orientada a composição de componentes”.

O jogo *BaseKids* foi elaborado e desenvolvido com o propósito de servir de instrumento por professores e alunos, de forma a contribuir positivamente para o processo de ensino/aprendizagem e a educação inclusiva. Por meio de entrevistas com professores da APAE – Anápolis, obteve-se sugestões para que o jogo atendesse crianças atendidas pela instituição, permitindo que o jogo estivesse melhor adequado para a educação inclusiva. Espera-se que o presente estudo possa também servir de apoio para aqueles que tenham interesse em desenvolver jogos eletrônicos educativos, independentemente da escolha da *game engine* para desenvolvimento do jogo.

Percebe-se que é necessário a realização de testes em relação ao jogo *BaseKids*, a fim de se verificar a sua eficiência quanto à aplicabilidade em salas de aula, principalmente no âmbito da usabilidade e acessibilidade, já que o jogo tem o propósito de ser um instrumento de apoio na educação inclusiva, sendo assim uma possibilidade de trabalho futuro, de forma que possa melhorar/aprimorar os aspectos de usabilidade e acessibilidade do jogo *BaseKids*.

7 REFERÊNCIAS BIBLIOGRÁFICAS

BARBARO, U. **Argumento e Roteiro**. São Paulo, Global Editora, 1983.

BARATO, J. **Taxonomia de Bloom**. 2009. Disponível em: <<http://jarbas.wordpress.com/2009/11/21/taxonomia-de-bloom/>>. Acesso em: 7 mai. 2017.

BATTAIOLA, A. L. et al. **Desenvolvimento de jogos em computadores e celulares**. Revista de Informática Teórica e Aplicada, 2001.

BRASIL. MINISTÉRIO DA EDUCAÇÃO. **Dia da Síndrome de Down revela evolução da inclusão no Brasil**. Brasília, DF: 2015. Disponível em: <<http://portal.mec.gov.br/ultimas-noticias/202-264937351/21167-dia-da-sindrome-de-down-revela-evolucao-da-inclusao-no-brasil>>. Acesso em 26 abr. 2017.

CHAPMAN, A. **Bloom's taxonomy - learning domains**. Businessballs, 2009. Disponível em: <<http://www.businessballs.com/bloomstaxonomyoflearningdomains.htm>>. Acesso em: mai. 2017.

CONKLIN, J. A taxonomy for learning, teaching and assessing: a revision of Blooms's taxonomy of educational objectives. **Educational Horizons**, v. 83, n. 3, p. 153-159, 2005.

DIAS, R. **Game Engine: o que é, para que serve e como escolher a sua**. 2016. Disponível em: <<http://producaodejogos.com/game-engine/>>. Acesso em: 19 mai. 2017.

DIAS, R. **Unity – Guia Completo sobre a Game Engine**. 2017. Disponível em: <<http://producaodejogos.com/unity/>>. Acesso em: 19 mai. 2017.

ENGINE, U. **Unreal Engine**. Disponível em: <<https://www.unrealengine.com/>>. Acessado em: mai. 2017.

FERRAZ, A. P. C. M; BELHOT, R. V. **Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais**. 2010. Disponível em: <<http://www.scielo.br/pdf/gp/v17n2/a15v17n2>>. Acesso em set. 2016.

FILHO, I. J. S. **Profissionais do game design**. 2011. Disponível em: <<https://dcc.catalao.ufg.br/up/498/o/ltamar2011.pdf>>. Acesso em: mai. 2017.

GAMES, Y. **Game Maker**. Disponível em: <<http://www.yoyogames.com/gamemaker>>. Acessado em: mai. 2017.

HUIZINGA, J. **Homo Ludens: o jogo como elemento da cultura**. 4. ed. São Paulo: Perspectiva, 2000.

JUNIOR, A. S. R; NASSU, B. T; JONACK, M. A. **Um estudo sobre os processos de desenvolvimento de Jogos Eletrônicos (Games)**, 2002. Disponível em:

<http://www.ademar.org/textos/proc_desenv_games/proc_desenv_games.pdf>. Acesso em: dez. 2017.

KISHIMOTO, T. M. (org). **O jogo e a Educação Infantil**. São Paulo: Pioneira, 1998.

KODJAOGLANIAN, V. L.; BENITES, C. C. A.; MACÁRIO, I.; LACOSKI, M. C. E. K.; ANDRADE, S. M. O.; NASCIMENTO, V. N. A.; MACHADO, J. L. Inovando métodos de ensino- -aprendizagem na formação do psicólogo. **Psicologia: Ciência e Profissão**. Brasília, v. 23, n. 1, p. 2-11, mar. 2003.

KUCERA, J. **Desenvolva jogos com a Unity 3D**. 2013. Disponível em: <<http://www.devmedia.com.br/desenvolva-jogos-com-a-unity-3d/29125>>. Acesso em: 19 mai. 2017.

KREIMEIER, B. **Game Design Methods: A 2003 Survey**. Disponível em: <http://www.gamasutra.com/view/feature/2892/game_design_methods_a_2003_survey.php> Acesso em: 8 de mai. 2017.

LEMES, D. O. **Games independentes: fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais**. 2009. Disponível em: <<https://www.misp.pucsp.br/handle/handle/18241>>. Acesso em: 19 mai. 2017.

LIMA, E. E. S. **3D Game Builder: Uma Game Engine para a criação de jogos 3D**. 2008. Disponível em: <http://edirlei.3dgb.com.br/artigos/3d_game_builder_tcc.pdf>. Acesso em: mai. 2017.

LINS, E. K. S. M. **A inclusão da criança com síndrome de down e seu processo de alfabetização**. Pedagogia, São Paulo, p. 4, 2014.

MACHADO, H. **Unity 3D: Introdução ao desenvolvimento de games**. 2014. Disponível em: <<http://www.devmedia.com.br/unity-3d-introducao-ao-desenvolvimento-de-games/30653>>. Acesso em: 18 mai. 2017.

MAKER, R. **RPG Maker**. Disponível em: <<http://www.rpgmakerweb.com/>>. Acessado em: mai. 2017.

MARCHESI, A. **Da linguagem da deficiência às escolas inclusivas**. In: COLL, César; MARCHESI, Álvaro; PALACIOS, Jesús; (Orgs.). Desenvolvimento psicológico e educação. Trad. Fátima Murad, Porto Alegre: Artmed, 2004.

MIGUEL, G. C.; MONTEIRO, L. R, J; CAVALHIERI, A. M. **Componentes de uma Game Engine**. 2006. Disponível em: http://cognitio.incubadora.fapesp.br/portal/atividades/cursos/posgrad/jogos_eletronicos/2006/trabalhos/tf2/TF2-M_GameEngine_Cesar-Marcos-Julio.pdf. Acesso em: mai. 2017.

MIRANDA, G. J; NOVA, S. P. C. C; CORNACCHIONE J. E. B. Os saberes dos professores-referência no ensino de Contabilidade. **Revista Contabilidade & Finanças**. São Paulo, v. 23, n. 59, p. 142-153, mai./ago. 2012.

- NORONHA, E. G; PINTO, C. L. **Educação Especial e Educação Inclusiva: aproximações e convergências**, 2011. Disponível em: <http://www.catolicaonline.com.br/semanapedagogia/trabalhos_completos>. Acesso em: mai. 2017.
- PEDERSEN, R. **Game design foundations**. 1. ED. Sudbury: Wordware publishing, INC. 2003.
- PERUCIA, A, S; BERTHEM, A. C; BERTSCHINGER, G. L; MENEZES, R. R. C. **Desenvolvimento de Jogos Eletrônicos – Teoria e Prática**. São Paulo, Novatec, 2005.
- PESSOA, C. **wGEM: Um Framework de Desenvolvimento de Jogos para Dispositivos Móveis**. Dissertação de Mestrado, Pernambuco, Novembro, 2001.
- PESSOA, A. C. C; RAMALHO, L. G; BATTAIOLA, L. A. **wGEM: um Framework de Desenvolvimento de Jogos para Dispositivos Móveis**. 2002.
- PRENSKY, M. **Teaching Digital Natives: partnering for real learning**, California, USA: Corwin Press, 2010.
- ROLLINGS, Andrew; ADAMS, Ernest. **Andrew Rollings and Ernest Adams on Game Design**. 1ª edição. New Riders, 2003
- SCIRRA. **Construct 2**. Disponível em: <<https://www.scirra.com/construct2>>. Acessado em: mai. 2017.
- SCHUYTEMA, P. **Design de Games: uma abordagem prática**. São Paulo, Cengage Learning, 2008.
- TOLEDO, E. H.; MARTINS, J. B. **A atuação do professor diante do processo de inclusão e as contribuições de Vygotsky**. Psicopedagogia, Paraná, 2009.
- TRAVERSINI, C. S; BUAES, C. S. Como discursos dominantes nos espaços da educação atravessam práticas docentes? **Revista Portuguesa de Educação**. Braga, v. 22, n. 2, p. 141-158, 2009.
- UNITY3D. **UNITY: Game Development Tool**. Disponível em: <<http://unity3d.com/>>. Acessado em: mai. 2017.
- VALVE. **Valve Developer Community**. Disponível em: <<https://developer.valvesoftware.com/>>. Acessado em: mai. 2017.
- WALL, P. TELLES, M. **A Taxonomia de Bloom**. Dynamiclab, 2004. Disponível em: <<http://www.dynamiclab.com/moodle/mod/forum/discuss.php?d=436>>. Acesso em: mai. 2017.
- WANG, W. S. **O aprendizado através de jogos para computador: por uma escola mais divertida e mais eficiente**. Pedagogia, São Paulo, nov. 2005.

APÊNDICE 1 – Imagens do jogo *BaseKids*

As imagens do *BaseKids* referem-se ao jogo após o desenvolvimento, contemplando desde os menus, como a tela inicial, o mapa de fases e créditos, até as fases do jogo (seleção dos objetos, formação da palavra e a tela de parabéns).

Imagem 1 – Tela inicial.



Fonte: Equipe *BaseKids*.

Imagem 2 – Mapa de Fases.



Fonte: Equipe *BaseKids*.

Imagem 3 – Créditos.



Fonte: Equipe *BaseKids*.

Imagem 4 – Sair.



Fonte: Equipe *BaseKids*.

Imagem 5 – Fase “Massas e Outros”.



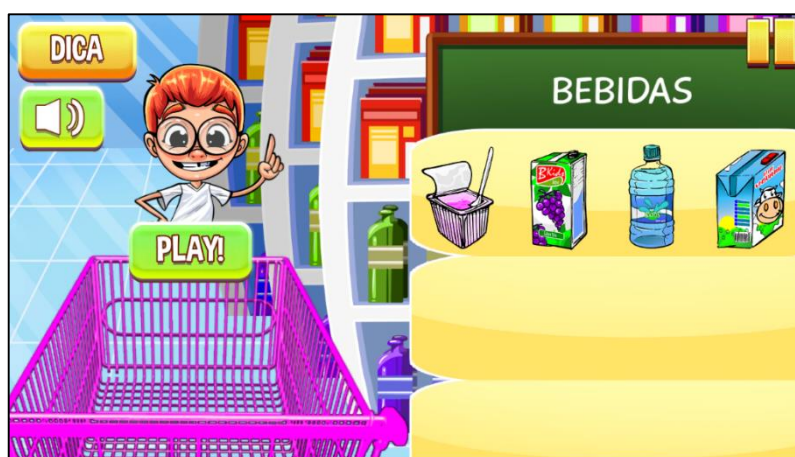
Fonte: Equipe *BaseKids*.

Imagem 6 – Fase “Verduras e Frutas”.



Fonte: Equipe *BaseKids*.

Imagem 7 – Fase “Bebidas”.



Fonte: Equipe *BaseKids*.

Imagem 8 – Fase “Papelaria”.



Fonte: Equipe *BaseKids*.

Imagem 9 – Fase “Eletrodomésticos”.



Fonte: Equipe *BaseKids*.

Imagem 10 – Fase “Brinquedos”.



Fonte: Equipe *BaseKids*.

Imagem 11 – Fase “Moda”.



Fonte: Equipe *BaseKids*.

Imagem 12 – Fase “Utensílios Domésticos”.



Fonte: Equipe BaseKids.

Imagem 13 – Fase “Eletrônicos”.



Fonte: Equipe BaseKids.

Imagem 14 – Dica.



Fonte: Equipe BaseKids.

Imagem 15 – Opções.

Fonte: Equipe *BaseKids*.

Imagem 16 – Formação de palavra.

Fonte: Equipe *BaseKids*.