



FACULDADE BILAC
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

DANIEL VARJÃO CHAVES
RÔMULO LIMA DE OLIVEIRA
TALITA ANDRESSA MARTINS PINTO

CAAPORA RPG

SÃO JOSÉ DO CAMPOS – SÃO PAULO
2015

DANIEL VARJÃO CHAVES
RÔMULO LIMA DE OLIVEIRA
TALITA ANDRESSA MARTINS PINTO

CAAPORA RPG

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Faculdade Bilac, como requisito à obtenção
do grau de bacharel em Ciência da Computação.

Orientador: Prof. Msc Marcos Flávio de
Souza Reis

SÃO JOSÉ DO CAMPOS – SÃO PAULO

2015

Pinto, Talita Andressa Martins.

Caapora RPG / Talita Andressa Martins Pinto
Daniel Varjão Chaves, Rômulo Lima de Oliveira. - 2015

69f.

Monografia – Faculdade Bilac – São José dos Campos

Referencias: f. 60-63

1. Jogo. 2. Unity3D. 3. Caapora. 4. Desenvolvimento
Mobile. I. Chaves, Daniel Varjão. II. Oliveira, Rômulo Lima
de.

CDU -

DANIEL VARJÃO CHAVES
RÔMULO LIMA DE OLIVEIRA
TALITA ANDRESSA MARTINS PINTO

CAAPORA RPG

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Faculdade Bilac, como requisito à obtenção
do grau de bacharel em Ciência da Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Msc Marcos Flávio de Souza Reis (Orientador)
Faculdade Bilac

Prof. Me. Gerson Penha Neto
Faculdade Bilac

Prof. Valentino D'Ambrosi Junior
Faculdade Bilac

À nossa família, por acreditarem que somos capazes e por sempre nos incentivarem e não deixar-nos abater pelo desanimo. Ao professor Marcos Reis, que apesar dos contra tempos nunca desistiu do nosso projeto.

AGRADECIMENTOS

Primeiramente a Deus que permitiu que este sonho fosse possível, e não somente nestes anos como universitários, mas que em todos os momentos é o maior mestre que alguém pode ter. Aos nossos familiares, pelo amor, incentivo e apoio incondicional.

A esta faculdade, seu corpo docente, direção e administração que viabilizaram a janela que hoje visamos um horizonte superior.

Agradecemos a todos os professores por nos proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, que se dedicaram a nós, e por não somente terem nos ensinado, mas por terem nos feito aprender. Aos professores Gérson Neto e Marcos Reis, pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia.

Aos amigos e colegas universitários, no qual sempre sugeriram melhorias tanto na vida acadêmica quanto para a vida pessoal, a eles depois da conclusão do curso, saudades irá resumir nossa caminhada juntos.

“É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitória nem derrota.”

(Franklin Roosevelt)

RESUMO

O objetivo desse projeto é conscientizar o público infantil sobre o desmatamento, que está ocorrendo dentro do Brasil, principalmente dentro da Floresta Amazônica.

O jogo Caapora tem como cenário a Floresta Amazônica, lugar onde tem como protetor o personagem principal, o Caipora, uma figura folclórica que tenta ajudar a mãe natureza recuperando ou evitando danos causados pela destruição humana.

No desenvolvimento foram utilizadas tecnologias atuais como o Unity3D, que possui uma série de recursos gráficos, uma linguagem de alto nível como o C# e que também permite disponibilizar o jogo para várias plataformas.

Palavras-chave: Jogo. Unity 3D. Caapora. Desenvolvimento Mobile

ABSTRACT

This project objective is to educate the children public on deforestation that is occurring within Brazil, mostly in the Amazon rainforest.

Caapora game has as scenario the Amazon Rainforest, where it has a protector the main character, Caipora, which it is trying to help the Mother Nature recovering or avoiding damage made by human beings.

To build the game was chosen current technologies such as Engine Unity3D, this tool works with high-level programming language C# which makes easier to export the game to multiple platforms.

LISTA DE ILUSTRAÇÕES

Figura 1 – Jogo Pong criado pela Atari em 1972	17
Figura 2 – Diversidade de formas da Caipora	19
Figura 3 – Jogo Half Life 2, desenvolvido pela Valve Corporation (2004)	20
Figura 4 – Jogo Flight Simulator desenvolvido pela Microsoft (2015)	21
Figura 5 – Jogo Final Fantasy XIII desenvolvido pela Square Enix (2009)	22
Figura 6 – Fifa 15 desenvolvido pela EA Sports (2014)	22
Figura 7 – Representação gráfica de um plano bidimensional	23
Figura 8 – Representação gráfica de um espaço isométrico (2.5D)	24
Figura 9 – Representação gráfica de um espaço tridimensional	25
Figura 10 – Venda Mundial de Smartphones por Sistemas Operacionais	27
Figura 11 – Comparação do formato .class da JVM com o .dex usado pela Dalvik VM. .	29
Figura 12 – Prática do Scrum	31
Figura 13 – Interação das atividades	32
Figura 14 – Funcionamento do Controle de Versão	33
Figura 15 – Uso da engine Unity 3D para desenvolvimento do jogo Caapora	39
Figura 16 – Gameobject do Player	40
Figura 17 – Utilização do Adobe Illustrator CC no jogo Caapora	41
Figura 18 – Jogo Pora: Free Cockatoos desenvolvido para conscientização dos mal tratos as aves da Indonésia	43
Figura 19 – Caso de uso, ações do jogador	45
Figura 20 – Menu Principal	48
Figura 21 – Cenário do jogo Caapora RPG	49
Figura 22 – Tileset utilizado na criação do cenário	50
Figura 23 – Sprite do caipora correndo	51
Figura 24 – Componente Animation	51
Figura 25 – Componente Animator	52
Figura 26 – Elemento de colisão em volta de um tile	52
Figura 27 – Componente Rigidbody	53
Figura 28 – Caipora com o balde	54
Figura 29 – Caipora lançando água nas chamas	54
Figura 30 – Fogo se alastrando	55

Figura 31 – Exemplo de NPC	56
Figura 32 – Estrutura de Classes de Personagens	57
Figura 33 – Estrutura de Classes de NPC’s	58
Figura 34 – Controles de movimentação e ação do personagem.	69
Figura 35 – Ilustração do Cenário	70

LISTA DE TABELAS

Tabela 1 – Tabela contendo nome, versão e ano de lançamento dos sistemas Androids. . .	30
Tabela 2 – Requisitos Funcionais	44
Tabela 3 – Requisitos Não Funcionais	45
Tabela 4 – Documentação do caso de uso Jogar	46
Tabela 5 – Documentação do caso de uso Personagem	46
Tabela 6 – Documentação do caso de uso Opções	47
Tabela 7 – Documentação do caso de uso Sair	47

LISTA DE SIGLAS E ABREVIATURAS

<i>ARPG</i>	Action Role-playing Game
<i>DLL</i>	Dynamic-link library
<i>FPS</i>	Frames Per Second
<i>GC</i>	Garbage Collector
<i>GUI</i>	Graphical User Interface
<i>IA</i>	Inteligência Artificial
<i>JVM</i>	Java Virtual Machine
<i>MCTI</i>	Ministério da Ciência, Tecnologia e Inovação
<i>MIT</i>	Massachusetts Institute of Technology
<i>NPC</i>	Non-player Character
<i>OHA</i>	Open Handset Alliance
<i>ONG</i>	Organização Não Governamental
<i>PDA</i>	Personal Digital Assistant
<i>PC</i>	Personal Computer
<i>PHP</i>	Hypertext Preprocessor
<i>RPG</i>	Role-playing Game
<i>SO</i>	Sistema Operacional
<i>UI</i>	User Interface
<i>URL</i>	Uniform Resource Locator
<i>USP</i>	Universidade de São Paulo
<i>XML</i>	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	17
1.1	MOTIVAÇÃO	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	OS GÊNEROS DOS JOGOS	20
2.2	CATEGORIA DE JOGOS	23
2.3	ENGENHARIA DE SOFTWARE	25
2.4	DESENVOLVIMENTO MOBILE	26
2.4.1	Aplicativos Móveis	27
2.5	ANDROID	28
2.5.1	Google Play	29
2.5.2	Versões Disponíveis	30
2.6	SCRUM	31
2.7	CONTROLE DE VERSÃO	32
2.7.1	Bitbucket	33
2.8	INTELIGÊNCIA ARTIFICIAL	34
2.9	BUSCA HEURÍSTICA	35
2.9.1	Algoritmo A* (A estrela)	35
2.10	ORIENTAÇÃO OBJETO	36
2.10.1	Classes e Objetos	36
2.10.2	Classes Abstratas	37
2.10.3	Interface	37
2.10.4	Padrões de Projeto	37
2.10.5	Singleton	37
2.10.6	Object Pool	38
3	TECNOLOGIAS E FERRAMENTAS UTILIZADAS	39
3.1	UNITY 3D – AMBIENTE DE DESENVOLVIMENTO	39
3.1.1	Scene	40
3.2	LINGUAGEM DE PROGRAMAÇÃO C#	40
3.3	ADOBE ILLUSTRATOR CC	41
3.4	VISUAL STUDIO 2015	41

4	ANÁLISE E PROJETO DO SISTEMA	43
4.1	ANÁLISE DE PRODUTOS SEMELHANTES	43
4.2	REQUISITOS DO SISTEMA	44
4.2.1	Requisitos Funcionais	44
4.2.2	Requisitos Não Funcionais	44
4.3	MODELAGEM DO SISTEMA	45
4.3.1	Caso de uso Jogar	45
4.3.2	Caso de uso Personagem	46
4.3.3	Caso de uso Opções	46
4.3.4	Caso de uso Sair	47
5	DESENVOLVIMENTO	48
5.1	CENAS DO JOGO	48
5.1.1	Menu Principal	48
5.1.1.1	Botões	48
5.1.1.2	Título	49
5.1.1.3	Imagen de fondo	49
5.1.2	Jogo	49
5.1.2.1	Tilesets	49
5.1.2.2	Sprites	50
5.1.2.3	Animação	50
5.1.2.4	Collider	52
5.1.2.5	Rigidbody	53
5.2	MECANISMOS	53
5.2.1	Pegar Balde	53
5.2.2	Encher o balde e Jogar Água	54
5.2.3	Chamas	55
5.2.4	Pathfinding	55
5.3	ESTRUTURA DE CLASSES	56
5.3.1	GameManager	57
5.3.2	Enemy	57
5.3.3	NPC	58
5.3.4	Inventory	58

5.3.5	Tree	59
5.3.6	Water	59
5.3.7	Fire	59
5.3.8	SpreadFire	59
5.4	FUNCIONAMENTO DO JOGO	59
6	CONCLUSÃO	60
6.1	TRABALHOS FUTUROS	60
	REFERÊNCIAS	61
	GLOSSÁRIO	64
	APÊNDICES	65
	APÊNDICE A – Game Design Document	66
A.1	INTRODUCÃO	66
A.1.1	Conceito	66
A.1.2	Gênero	66
A.1.3	Público Alvo	66
A.1.4	Plataforma	66
A.2	HISTÓRIA E NARRATIVA	66
A.2.1	Enredo	66
A.2.2	Personagens	67
A.2.2.1	Caipora	67
A.2.2.2	Curupira	67
A.2.2.3	Iara	67
A.2.2.4	Boto Cor de Rosa	67
A.3	GAMEPLAY	68
A.3.1	Estruturas de Missões e desafios	68
A.3.1.1	Primeira Missão	68
A.3.1.2	Segunda Missão	68
A.3.1.3	Terceira Missão	68
A.3.1.4	Quarta Missão	68
A.4	ENTRADAS	69
A.4.1	Controle	69
A.5	UNIVERSO DO JOGO	70

A.5.1 Cenário 70

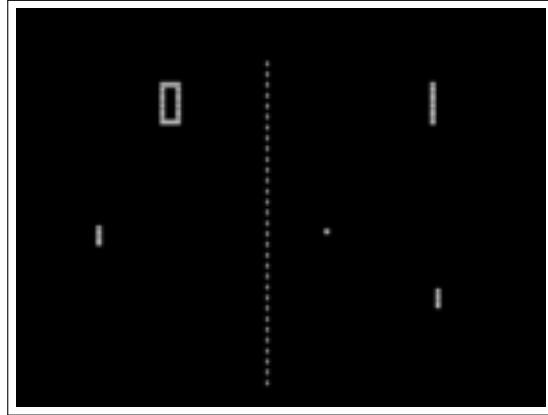
1 INTRODUÇÃO

Em 1958 surgia o primeiro jogo eletrônico com o nome de "Tênis para dois", criado pelo físico William Higinbotham. O jogo era processado por um computador analógico e exibido em um osciloscópio. (MASSARANI, 2013)

Em 1961 um grupo de estudantes do MIT desenvolveram o jogo Spacewar! No jogo o objetivo era que os jogadores controlassem suas naves num ambiente escuro, tentando abater o adversário. O jogo era desenvolvido num grande computador, que ocupava o espaço de uma sala inteira, o DEC PDP-1, este computador tinha como objetivo executar cálculos em geral, porém, após o jogo Spacewar! o computador ficou famoso pela possibilidade de entreter outros. (RAMOS, 2007)

Foi então em 1972, que o primeiro jogo comercial foi lançado pela Atari, o Pong, ilustrado na figura 1. Era um jogo simples e intuitivo, características que o fez ser tão popular para a época. (TRACCO, 2006)

Figura 1 – Jogo Pong criado pela Atari em 1972



Fonte: (WIKIPEDIA, 2005)

Os jogos citados foram os pioneiros para o desenvolvimento de jogos digitais. Desde então devido a exigência do público para se obter conteúdo de maior qualidade e interface gráfica nos jogos digitais, foram colocados esforços para se aprimorar essa categoria. Um dos fatores que contribuiu para este grande progresso foi a popularização da internet e dos dispositivos móveis.

Atualmente, nos dispositivos móveis, a categoria jogos tornou-se um aplicativo de passatempo global, dados comprovados na pesquisa realizada pela empresa Flurry, informando que os jogadores globais (plataforma Android) gastam em média 37 minutos por dia jogando.

(FLURRY, 2014).

Um dos dispositivos moveis em destaque é o *smartphones*, no qual evidenciado na pesquisa realizada pela consultoria Nielsen, 68 milhões de brasileiros utilizam *smartphones* (NIELSEN, 2015)

Tendo em base a estatística da Nielsen, as palavras do diretor geral da Pontomobi, Renato Virgilli, tornam-se ainda mais expressivas, diz ele: "Com o número cada vez maior de celulares, de *smartphones* e de *tablets* nas mãos dos usuários, cresce o interesse por aplicativos, seja para diversão ou entretenimento". (PIRES; VIRGILI, 2015)

Uma das vias para qual o mercado de aplicativos de jogos tem mostrado desenvolvimento eficiente é como ferramenta pedagógica. "O aluno é totalmente ativo ao usar um aplicativo, diferente de uma TV, que ele tem uma postura mais passiva. Os envolvidos no processo passam de consumidores a produtores de conteúdo, e a ter mais autonomia e criatividade, habilidades que serão demandadas no seu futuro profissional".

Considerações essa de Tori, coordenador do Laboratório de Tecnologias Interativas da USP (Universidade de São Paulo) de que o aplicativo tem a capacidade de auxiliar o desenvolvimento de uma tarefa, e que a escola não pode viver numa realidade desconectada do aluno.

A professora e diretora-executiva do Instituto Educadigital, Priscila Gonsales vem com o conceito de que recurso digitais e audiovisuais auxiliam a interação e visualização, aponta também que o aplicativo não substitui a metodologia tradicional e a utilização da ferramenta deva ser contextual ao conteúdo abordado. (LUISE, 2013)

Baseando-se nos itens descritos acima: popularidade no uso de aplicativos e que este tem eficiência no uso didático, foi desenvolvido e será apresentado neste documento o jogo digital Caapora RPG, que tem como objetivo ensinar crianças sobre a importância da preservação da mata e ressaltar a cultura folclórica brasileira. Aplicativo este desenvolvido para uso em dispositivos móveis com sistema operacional Android.

1.1 MOTIVAÇÃO

A floresta amazônica faz parte do ecossistema do mundo, é fundamental para o equilíbrio climático e é conhecida por ser o pulmão do mundo. Faz parte de 9 países dentre eles Brasil, Bolívia, Peru, Colômbia, Equador e Venezuela. E contém metade dos animais terrestres do planeta.

Apesar da extensão e de sua importância, a floresta Amazônica sofre uma grande degradação causada pelo desmatamento predatório e ilegal, poluição de rios, desequilíbrio em sua fauna e flora e em vezes ocorrendo a expulsão a força de populações nativas de seus territórios.

Em uma recente pesquisa da ONG Instituto do Homem e do Meio Ambiente da Amazônia (Imazon) do Pará, o aumento do desmatamento cresceu mais de 400% em comparação a novembro de 2013. (MESSIAS, 2015)

Outra grande riqueza que a Amazônia possui, é a sua grande diversidade de cultura, na parte que engloba a Amazônia brasileira (Acre, Amapá, Amazonas, Pará, Rondônia, Roraima e parte dos estados do Mato Grosso, Tocantins e Maranhão) tem-se os contos folclóricos no qual é conjunto de criações culturais de uma comunidade, baseado nas suas tradições expressas individual ou coletivamente, representativo de sua identidade social, segundo a Carta do Folclore Brasileiro.(FOLCLORE, 1995)

Em grande parte dos contos folclóricos os personagens são criaturas místicas tendo alguns deles o objetivo de proteger a floresta e as criaturas nela habitadas.

Um exemplo deste personagem é o Caipora (Caapora, em tupi, significa habitante do mato (TERRA, 2014)), vive montado em um porco do mato e é conhecido por espantar os madeireiros com seu assobio, ajudar animais presos em armadilhas e aprontar travessuras, com o objetivo de espantar os que estão fazendo mal a mãe natureza.

Segundo Franchini, o caipora é conhecido por várias regiões do Brasil, mas a sua forma física e seu método de ação varia de acordo com a região. Na figura 2 é possível ver dois tipos diferentes em que se acreditam que o Caipora possa existir, em um ele é uma figura feminina e em outro é um menino-índio montado em um porco do mato. (FRANCHINI, 2011)

Figura 2 – Diversidade de formas da Caipora



Fonte: (ART, 2015)

2 FUNDAMENTAÇÃO TEÓRICA

2.1 OS GÊNEROS DOS JOGOS

Gênero de jogo pode ser definido como uma modalidade ou tipo que permite o agrupamento de diversos jogos de acordo com suas características de jogabilidade. Segundo Eucidio, pensar no jogo é pensar no estilo de produção que o jogo seguirá, pensar nos desafios, dificuldades e possibilidades. (ARRUDA, 2014)

A plataforma dos jogos também influenciam para determinar o gênero do jogo. Em smartphones, por exemplo, os jogos tendem a ser mais casuais e rápidos. Existe uma infinidade de categorizações. (COTA et al., 2014)

- Jogos de Ação

Significa atividade, agir de acordo com situação. O jogador tem que tomar alguma ação em tempo real, agindo rapidamente. Jogos deste gênero é o mais popular para a plataforma PC. Um exemplo de jogo de ação é o Half-Life 2, onde possui as características de ser um jogo sagaz, envolvente e com enredo, na figura 3 encontra-se a capa da Mídia digital deste jogo. (LUCCHESE; RIBEIRO, 2014)

Figura 3 – Jogo Half Life 2, desenvolvido pela Valve Corporation (2004)



Fonte: (IGN, 2004)

- Jogos de Simulação

É a representação da realidade, são jogos que simulam situações vividas por

seres humanos. A principal utilização por este gênero é para simulação de voos, tendo como objetivo o treinamento de pilotos, como por exemplo, o jogo Flight Simulator da Microsoft. Na figura 4 é possível ver a simulação que ocorre no jogo. (LUCCHESE; RIBEIRO, 2014)

Figura 4 – Jogo Flight Simulator desenvolvido pela Microsoft (2015)



Fonte: (MICROSOFT, 2014)

- **Jogos de RPG**

Tem como característica o desenvolvimento gradativo do personagem. O jogador assume o papel do personagem aonde pode haver narrativas para auxiliar no decorrer do jogo. Um jogo muito conhecido que utiliza deste gênero é o Final Fantasy XIII, na figura 5 encontra-se a capa da Mídia digital deste jogo. (LUCCHESE; RIBEIRO, 2014)

Figura 5 – Jogo Final Fantasy XIII desenvolvido pela Square Enix (2009)



Fonte: (OTAKU, 2015)

- **Jogos de Esportes**

A principal característica de jogos deste gênero é que o controle ocorre sobre o personagem não mecânico. Normalmente o jogo tem um esforço físico do jogador no mundo virtual, em alguns jogos o personagem cansa e diminui sua velocidade. Um exemplo de jogo de esporte é o FIFA, conforme demonstrado na figura 6. (LUCCHESE; RIBEIRO, 2014)

Figura 6 – Fifa 15 desenvolvido pela EA Sports (2014)



Fonte: (WALLPAPERS, 2014)

- **Jogos Educacionais**

São jogos que possuem o intuito de ensinar as pessoas sobre determinado assunto, reforçando assim seu desenvolvimento e entendimento sobre um evento histórico ou cultural, ou ainda auxiliar na aprendizagem de alguma habilidade enquanto

de joga. (WIKIPEDIA, 2015)

2.2 CATEGORIA DE JOGOS

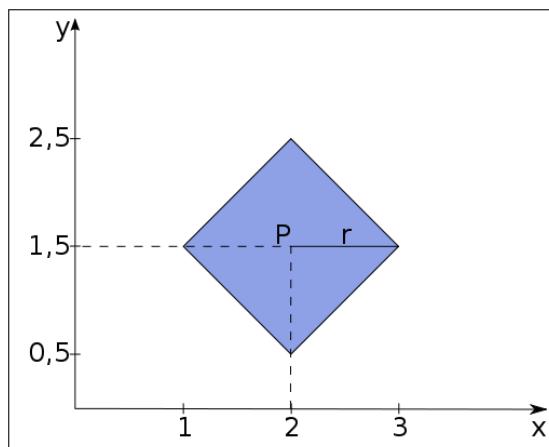
É possível classificar os jogos em 3 categorias, sendo estas:

- Jogos 2D

Implementados a partir de gráficos bidimensionais somente com uma camada.

Perspectiva *Top-down* - tem a perspectiva sobre a cabeça do personagem e este pode se movimentar em qualquer ângulo. *Side Scrolling* - tem a perspectiva do lado do personagem onde este se move para direita ou esquerda, comum em jogos de plataforma. Na figura 7 é possível verificar a representação gráfica de um plano 2D (bidimensional). (SILVA et al., 2009)

Figura 7 – Representação gráfica de um plano bidimensional



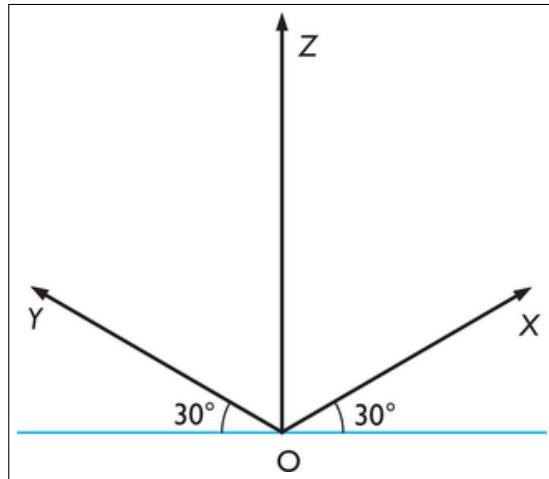
Fonte: (WIKIPEDIA, 2009)

- Jogos 2.5D

Atribuído em jogos onde o 2D e o 3D são combinados para reproduzir um cenário mais realista. Nesta categoria os personagens são modelados em imagens 2D e se movimentam em um cenário 3D ou um personagem em 3D em um fundo 2D, porém sendo mais comum um cenário que apresenta características de dimensão de profundidade a partir da sobreposição de imagens 2D.

Se divide na categorias: Isométrico, projeção oblíqua, *billboarding* e escalamento do eixo Z. Na figura 8 é possível verificar a representação gráfica de um plano 2.5D (isométrico). (SILVA et al., 2009)

Figura 8 – Representação gráfica de um espaço isométrico (2.5D)



Fonte: (GRAFICA, 2012)

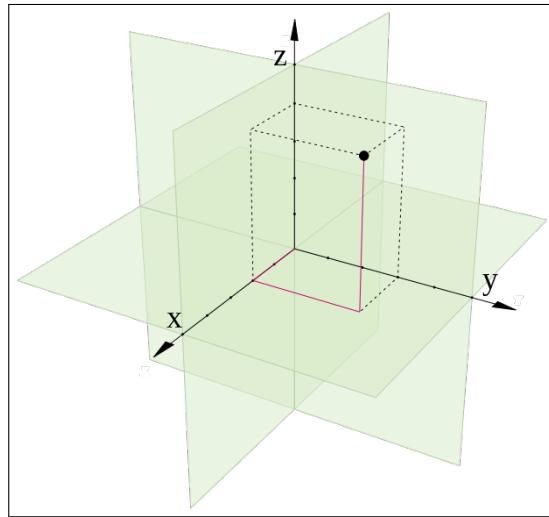
- Jogos 3D

São jogos onde o cenário é modelado tridimensionalmente sendo possível o personagem mover-se em qualquer direção. Se dividem nas categorias

- 3D Fixo - Jogo onde a câmera é fixa.
- Primeira pessoa - Jogo onde a câmera está na posição na altura dos olhos do personagem.
- Terceira pessoa - Jogo aonde câmera está próxima ao personagem, sempre o acompanhando.

Na figura 9 é possível verificar a representação gráfica de um plano 3D (tridimensional). (SILVA et al., 2009)

Figura 9 – Representação gráfica de um espaço tridimensional



Fonte: (WIKIPEDIA, 2011)

2.3 ENGENHARIA DE SOFTWARE

Sommerville definiu engenharia de software como uma disciplina da engenharia que cuida de todos os aspectos da produção de software, desde a especificação do sistema até a sua manutenção, depois que ele entra em operação.

A engenharia é constituída de processos de software, que são conjuntos de atividades e resultados para se produzir um software. Existem quatro atividades fundamentais, são elas:

- Especificação de Software: o software a ser desenvolvido e as restrições para sua operação são definidos. Desenvolvimento de Software: o software deve ser feito atendendo suas especificações.
- Validação de Software: o software tem de ser validado para garantir que está em cima do que o cliente deseja.
- Evolução do Software: o software sofre modificações para atender as necessidades do cliente. (SOMMERVILLE, 2007)

No desenvolvimento de software, o principal objetivo é a criação de sistemas que atendam às necessidades dos clientes e usuários, ou seja, uma correta especificação dos requisitos se torna essencial para que o desenvolvimento tenha sucesso. Uma forma de entender melhor esses requisitos é dividindo-os em: Requisitos Funcionais e Requisitos Não-Funcionais.

Os Requisitos Funcionais definem as funções que componentes do sistema ou sistemas devem executar. Os Requisitos Não-Funcionais incluem limitações no produto, como por

exemplo: desempenho, confiabilidade, segurança e limitações no desenvolvimento, como custos e tempo, componentes a serem reutilizados, entre outros. (VASCONCELOS et al., 2006)

2.4 DESENVOLVIMENTO MOBILE

Um dispositivo móvel pode ser definido como um computador de bolso, normalmente composto de uma tela e um teclado em miniatura podendo estes serem combinados em um só dispositivo conhecido como *touchscreen*.

Características dos dispositivos móveis:

- Pequeno em tamanho
- Baixo consumo de energia
- Curto tempo de inicialização
- Armazenamento de dados local e/ou remoto

Os dispositivos móveis mais populares são:

- *Smartphone*
- *Tablet*
- Console portátil
- *Notebook*

(SANTOS, 2011)

A tecnologia que torna viável a existência dos dispositivos móveis começou a partir do primeiro celular inventado, em 3 de abril de 1983, o Motorola DynaTAC 8000x. Foi então que em 1992 a empresa Apple lançou no mercado o primeiro PDA que continha memória de 1MB e tela sensível ao toque, desde então esta tecnologia tem se desenvolvido e mostrado como vantagem a possibilidade dos dados serem acessados em qualquer lugar a qualquer hora. (SANCHES, 2011)

Dentre os dispositivos móveis citados, o que mais se destacam são os *smartphones*. Um *smartphone* (telefone inteligente) poder ser definido como um celular que possui muitas funções. Essas funções são realizadas de maneira mais eficientes do que em um celular normal, em geral possuem internet WiFi, navegadores, assim como também é possível realizar a instalação de aplicativos e possui um Sistema Operacional (SO). (TORRES, 2013)

Uma das grandes vantagens dos *smartphones* é a capacidade de qualquer pessoa desenvolver um aplicativo para o aparelho pois, em geral, o SO é um software aberto, onde existe a flexibilidade da criação de aplicativos.

Um Sistema Operacional é um conjunto de programas responsável por alocar recursos do hardware fornecendo também uma interface para o usuário. Os sistemas operacionais para smartphones mais conhecidos atualmente são: iOS, Android, Windows Mobile e BlackBerry (BENFICA, 2011)

Segundo uma pesquisa realizada pela Gartner, o sistema operacional mais vendido em 2015 (para *smartphones*) foi o Android e em segundo o iOS, conforme pode ser visto na figura 10. (FREIRE, 2015)

Figura 10 – Venda Mundial de Smartphones por Sistemas Operacionais

Venda Mundial de Smartphones por Sistemas Operacionais (Unidades por milhares)				
Sistema Operacional	2015	2015 Mercado em %	2014	2014 Mercado em %
Android	271,01	82,2	243,844	83,8
iOS	48,086	14,6	35,345	12,2
Windows	8,198	2,5	8,095	2,8
BlackBerry	1,153	0,3	2,044	0,7
Outros	1,229	0,4	1,416	0,5
Total	329,676	100	290,384	100

Fonte: Gartner, 2015

2.4.1 Aplicativos Móveis

Um aplicativo móvel, conhecido também como App é um software desenvolvido para dispositivos móveis, por exemplo, *smartphones*, *Tablets*, *notebooks*, esses aplicativos podem ser nativos, Web ou híbridos. (AMBROS, 2013)

- Aplicativos Nativos

São os aplicativos que podem ser acessados na tela principal através de seu ícone. Podem ser instalados através de um aplicativo de loja específico para sua plataforma, como por exemplo o sistema iOS utiliza a App Store baixar seus aplicativos, enquanto o Android utiliza o Google Play. Esses aplicativos residem no dispositivo.

- *Mobile Web Apps*

São aplicativos que na verdade são sites e não aplicativos raiz, como os nativos. São executados a partir de um navegador e normalmente criado um ícone na tela principal do dispositivo para que quando clicando ele acesse a URL determinada.

- Aplicativos Híbridos

São os aplicativos que são parcialmente web e nativos. Estes devem ser baixados

pela loja de aplicativos, ficam armazenados no dispositivo porém podem também serem utilizado como web App.

Segundo dados divulgados pelo Ministério de Ciência Tecnologia e Inovação (MCTI), o mercado de aplicativos móveis está em alta, movimentando mais de US\$ 25 bilhões por ano no Brasil, com expectativas de alcançar US\$ 70 milhões até 2017. (PINHEIRO, 2015)

2.5 ANDROID

Android é um sistema operacional para dispositivos móveis, seu código é aberto (*open-source*). Inicialmente desenvolvido pela Android Inc, em 2003 comprada pela empresa Google e desde 2007 mantida pela *Open Handset Alliance* (OHA).

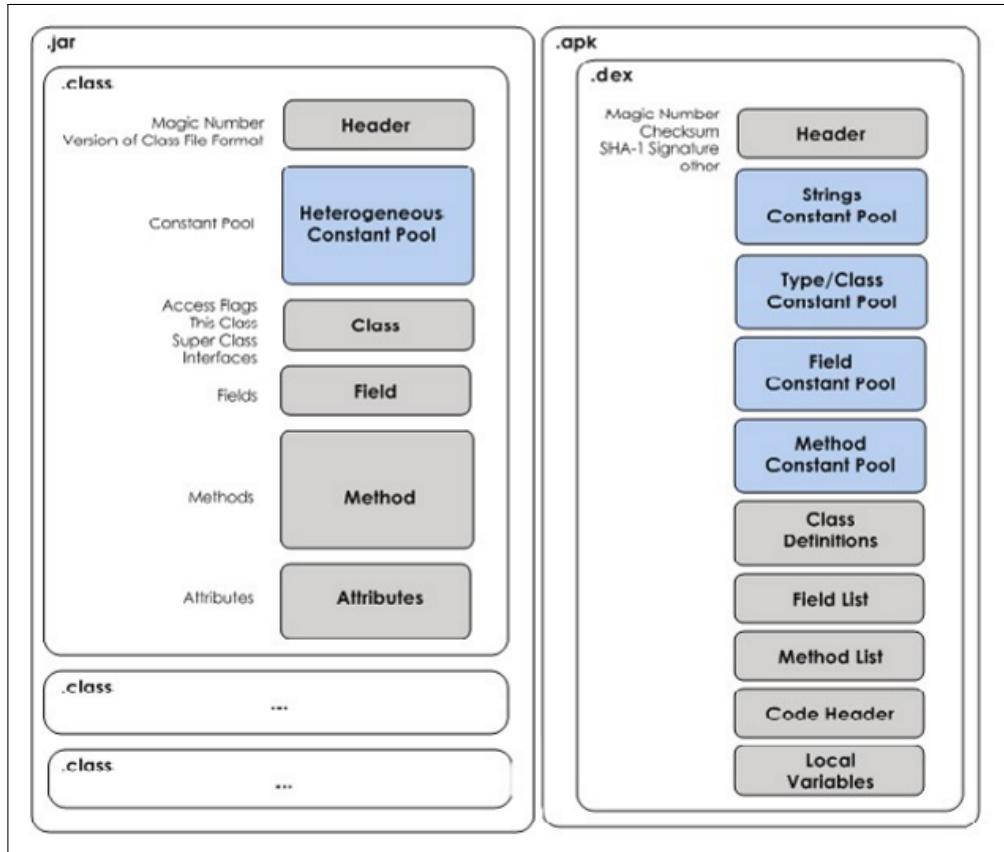
O sistema operacional Android possui seu núcleo (*Kernel*) em Linux e utiliza Java como linguagem de programação padrão. Também utiliza de sua própria máquina virtual chamada de Dalvik.

A máquina virtual Dalvik possui grandes diferenças entre a Java Virtual Machine (JVM) tornando-a superior; a Dalvik é baseada em registradores e não em pilhas como a JVM.

A Dalvik executa seus arquivos na extensão ".dex" (Dalvik Executable) que nada mais são do que arquivos Java já previamente projetados e compilados realizando a otimização de memória e compartilhamento de dados.

Na figura 11 é possível fazer um comparativo entre a JVM (.class) e a Dalvik VM (.dex). (KORJENIOSKI, 2011)

Figura 11 – Comparação do formato .class da JVM com o .dex usado pela Dalvik VM.



Fonte: Marcelo Korjenioski, 2011

2.5.1 Google Play

Google Play também conhecida como Google Play Store é a loja virtual da Google para venda de aplicativos. Antigamente esta loja virtual recebia o nome de Android Market mas em 2002 seu nome mudou para haver a unificação de produtos. No Google Play esta disponível para usuário baixar aplicativos gratuito e/ou pagos. (ALEXANDRE, 2012)

Para um desenvolvedor distribuir seus aplicativos na Google Play é necessário que o mesmo faça seu cadastro e pague uma quantia de U\$25,00, feito isso será possível fornecer seu aplicativo na loja virtual. Uma vez que o desenvolvedor tem a conta criada é possível ter acesso a comentários, estatísticas e erros de seus aplicativos. Caso o aplicativo seja pago, o lucro para o desenvolvedor é de 70% do valor cobrado pelo aplicativo. (CARVALHO, 2012)

2.5.2 Versões Disponíveis

Desde de seu inicio, o Android nomeia suas versões com nomes de doces e sobre-mesas e segue uma ordem alfabética. Ainda não foi revelada pela empresa o real motivo deste padrão de nomenclatura.

Atualmente existem 13 versões do sistema operacional Android, estes apresentados na tabela 1. (STUDIO, 2015)

Tabela 1 – Tabela contendo nome, versão e ano de lançamento dos sistemas Androids.

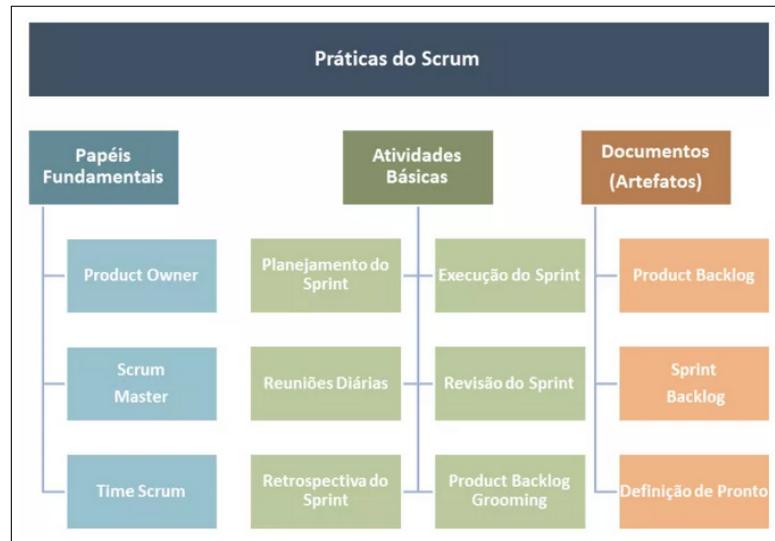
Nome	Versões	Ano de Lançamento
Alpha (Apple Pie)	1.0	2008
Beta (Banana Bread)	1.1	2009
Cupcake	1.5	2009
Donut	1.6	2009
Eclair	2.0 e 2.1	2009
Froyo	2.2	2010
Gingerbread	2.3	2010
Honeycomb	3.0; 3.1; 3.2	2011
Ice Cream Sandwich	4.0	2011
Jelly Bean	4.1; 4.2; 4.3	2012 e 2013
KitKat	4.4	2013 e 2014
Lollipop	5.0	2014 e 2015
Marshmallow	6.0	2015

Fonte: (HAMMERSCHMIDT, 2015)

2.6 SCRUM

Scrum é uma metodologia ágil para gestão e planejamento de projetos de softwares e sua base fundamental pode ser bem descrita na figura 12.

Figura 12 – Prática do Scrum



Fonte: (VIEIRA, 2014)

Cada equipe de Scrum, geralmente, possuem 3 papéis:

- *Product Owner*

É responsável por decidir os recursos e funcionalidades utilizadas no projeto e também sua responsabilidade manter clareza no objetivo do projeto, para que haja uma melhor interação o ProductOwner geralmente colabora com o ScrumMaster.

- *Scrum Master*

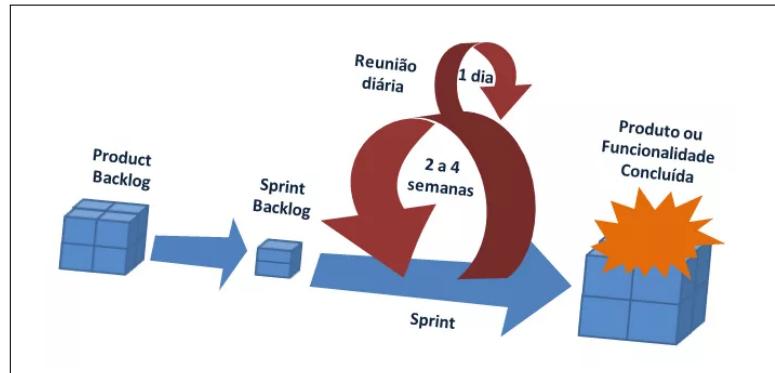
É de sua responsabilidade ajudar todos envolvidos a entender os valores, princípios e práticas do Scrum. Também fica responsável por melhorias no uso do Scrum e está sempre orientando para que não haja a perda de foco.

- *Development Team*

São todos da equipe responsáveis pelo desenvolvimento em si do software, possuem papéis como: arquiteto, testador, programador, entre outros. Para esta equipe é recomendável que se organizem para determinar a melhor maneira de realizar o trabalho.

A figura 13 representa o processo de interação das atividades.

Figura 13 – Interação das atividades



Fonte: (VIEIRA, 2014)

O *Product Owner* tem a visão final do produto, representado na figura 14 como grande cubo. Este cubo é "divido" em vários outros cubos pequenos, este chamado de *Product Backlog*.

O *Product Backlog* pode ser descrito, em uma forma simplificada, como várias etapas/objetivos para se chegar ao produto final. Para planejar a prioridade dos *Backlog* também quando e quanto tempo durará seu desenvolvimento, é utilizado o *Sprint*.

O *Sprint* tem duração mediana de 1 a 4 semanas, mas é também flexível dependendo do tempo desenvolvimento final estimado no começo do planejamento.

Também existe a técnica do *Daily Scrum*. Nele é feita as três perguntas consideradas básicas para haver um melhor desenvolvimento de todos da equipe. Estas são:

1. O que fiz ontem que ajudou o time a atingir a meta do *Sprint*?
2. O que vou fazer hoje para ajudar o time a atingir a meta do *Sprint*?
3. Existe algum impedimento que não permita a mim ou ao time atingir a meta do *Sprint*?

(VIEIRA, 2014)

2.7 CONTROLE DE VERSÃO

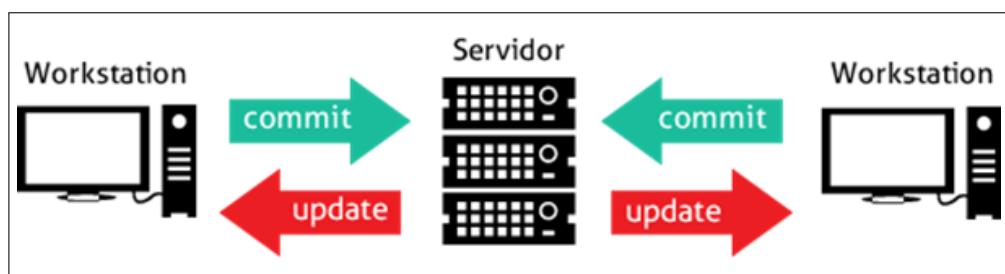
O controle de versão tem como objetivo a gerencia de versões de um mesmo projeto. Vários programadores podem trabalhar em um mesmo projeto sendo possível manter um histórico de atualizações.

Dentre as infinidades de vantagens para se utilizar um "controlador de versões" as três principais que se destacam são:

- Possibilidade de salvar o histórico
- Possibilidade de desenvolver versões diferentes
- Possibilidade de se programar em paralelo

O controle de versão funciona da seguinte forma: os históricos contendo as modificações de cada versão ficam armazenados em um repositório (servidor). Este processo dá à liberdade para que o desenvolvedor possa baixar a última versão, trabalhar em seu código e posteriormente atualizar a versão já previamente armazenada no servidor, este processo pode ser visto na figura 14.

Figura 14 – Funcionamento do Controle de Versão



Fonte: (RAMOS, 2015)

Para que seja realizada a sincronização entre a *workstation* e o servidor é necessário utilizar os comandos:

- *Commit* - realiza o envio das alterações realizadas para o servidor gerando um novo histórico de atualizações.
 - *Update* - realiza o envio da última versão contida no servidor para a *workstation*.
- (RAMOS, 2015)

2.7.1 Bitbucket

No mercado atualmente existem dois grandes serviços que disponibilizam o controle de versionamento e estes são: Github e o Bitbucket.

Bitbucket é uma ferramenta de domínio da empresa Atlassian e implementa dois padrões de versionamentos, o GIT e o Mercurial. (MAUDA, 2015)

2.8 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial (IA) é um ramo de pesquisa da ciência da computação que procura através de meio computacionais desenvolver mecanismos e/ou dispositivos que simulem a capacidade do ser humano de raciocinar e resolver problemas, ou seja, ser inteligente.

O campo de IA tem como objetivo, o contínuo aumento da "inteligência" do computador, pesquisando, para isto, os fenômenos da inteligência natural. Para este fim, IA é definida como sendo uma coleção de técnicas suportadas por computador emulando algumas capacidades dos seres humanos. Esta coleção inclui:

- Resolução de problemas
- Compreensão de Linguagem Natural
- Visão e Robótica
- Sistemas Especialistas e Aquisição de Conhecimento
- Metodologias de Representação de Conhecimento

Hoje em dia a Inteligência Artificial é utilizada em vários fins, não só exclusivamente para informática, no entanto, um dos pontos em que se destaca é para o desenvolvimento de jogos.

Primeiramente utilizada em jogos clássicos como xadrez ou jogo da velha e atualmente utilizada para jogos digitais.

Os algoritmos de IA desenvolvidos para jogos digitais podem ser divididos em três blocos:

- Movimento:

De como um personagem é movimentado, do local inicial até o destino final, também determina o cálculo do percurso, detectando objetos e desviando de obstáculos.

- Tomada de Decisão:

Cada personagem possui um conjunto de atividades e estados possíveis, estes sendo: estar parado, fugir, atacar adversário, entre outros. Para que cada ação seja possível, a personagem precisará fazer uma tomada de decisão analisando o contexto. Esta decisão pode ser de ativar um algoritmo de movimento, alterar um estado interno ou a animação. Não necessariamente essas alterações serão visuais.

- Estratégia:

A maioria dos jogos digitais desenvolvidos com IA ocorrem nos dois tipos já citados estes também podem conter personagens que trabalham em grupo ou equipe contando com uma tomada de decisão individual.

As pesquisas iniciaram-se na Segunda Guerra Mundial, com os cientistas Hebert Simon, Allen Newell, Jonh McCarthy com o objetivo em comum de reproduzir uma máquina que simulasse o cérebro humano. (SIMAO, 2013)

2.9 BUSCA HEURÍSTICA

A busca heurística pode ser definida como uma estratégia que faz uso do conhecimento específico do problema, sendo possível obter soluções mais eficientes do que uma estratégia sem informação. (RUSSELL; NORVIG, 2003)

De acordo com (RICH; KNIGHT, 1993), heurística é uma técnica eficiente para um processo de busca. A heurística, mesmo que as vezes possa passar por pontos de interesse despercebidos, pode levar a direcionamentos interessantes. A heurística aprimora os caminhos a serem percorridos, entretanto pode desconsiderar o melhorar caminho a percorrer.

2.9.1 Algoritmo A* (A estrela)

O A* (lê-se A estrela) é um algoritmo de busca de caminho (*pathfinding*). Segundo Russel (2003) A* é a forma mais eficiente pela busca da melhor escolha, na qual avalia os nós combinando o custo para alcançar cada nó $g(n)$ e o custo do nó ate o objetivo $h(n)$.

$$f(n) = g(n) + h(n)$$

A função $f(n)$ representa o custo estimado da solução menos custosa passando por n e chegando ao estado-objetivo. O A* utiliza também o método de lista aberta e fechada aonde listas abertas armazenam todos os nós que foram gerados porém ainda não examinados e a fechada armazena os nós que já analisados.

O algoritmo A* é completo e eficiente contanto que a função heurística $f(n)$ não superestime o custo ate o estado-objetivo. (SILVA, 2005)

2.10 ORIENTAÇÃO OBJETO

O termo "Orientação a Objeto" conota o processo de modelar um sistema como "um grupo de agentes autônomos que colaboram para realizar algum procedimento de alto nível". Este paradigma possibilita o desenvolvimento de software baseado em objetos, assim como os do mundo real, que interagem a fim de resolver um determinado problema.

Segundo Daniel Clark, programação orientada a objeto pode ser definida como uma abordagem em desenvolvimento de software em que a estrutura do software é baseada na interação entre objetos para realizar uma tarefa.

Ainda segundo Daniel Clark, os benefícios em utilizar este paradigma seriam:

- Uma transição mais intuitiva dos modelos de análise de negócio para implementação de modelos de software.
- A Possibilidade de manter e implementar mudanças no programa de forma mais eficiente e rápida.
- A possibilidade de reuso de componentes de código em outros programas dentre outros.

(CLARK, 2013)

2.10.1 Classes e Objetos

Como dito na definição de orientação a objetos, objetos e suas interações são recursos indispensáveis para a existência deste paradigma, neste contexto pode-se definir objeto como uma estrutura que incorpora dados e os procedimentos para trabalhar com estes dados.

Para a construção dos objetos faz-se necessário um outro recurso, conhecido como Classe que segundo Daniel Clark se entende por uma definição complexa de tipo de dado para um objeto e que ela contém informação sobre como um objeto deveria se comportar, incluindo seu nome, métodos, propriedades e eventos.

Uma implementação de um objeto é definida por sua classe. A classe especifica os dados internos de um objeto, sua representação e define as operações que um ele pode realizar.

(CLARK, 2013)

2.10.2 Classes Abstratas

Segundo Danny Poo, em uma hierarquia de classe há classes que são tão genéricas que não há intensão em criar objetos a partir delas. Tais classes são destinadas a conter apenas atributos ou métodos comuns a suas subclasses para propósito de reuso. Essas classes são conhecidas como abstratas. (POO DEREK KIONG, 2007)

Já Booch diz que Classe abstrata é aquela que possui como propósito principal definir uma interface comum entre suas subclasses definindo parte ou toda a implementação de suas operações. (BOOCH, 1991)

2.10.3 Interface

Interface é a visão externa de, por exemplo, uma classe, objeto, componente, ou estrutura composta, que enfatiza sua abstração enquanto esconde sua estrutura e os segredos de seu comportamento. (BOOCH, 1991)

Segundo Klark a utilização de interface é útil quando quisermos garantir que o código cliente que usa as classes saiba quais métodos estão disponíveis, como eles deveriam ser chamados, e os valores de retorno esperados. (CLARK, 2013)

2.10.4 Padrões de Projeto

Segundo Gamma, Padrões de projetos resolvem muitos dos problemas de do dia-a-dia que arquitetos de software encontram, e de diversas formas diferentes. (GAMMA RICHARD HELM, 1994)

Christopher Alexandre disse, "Cada Padrão descreve um problema que ocorre repetidamente no nosso ambiente, e então descreve a solução chave para o problema, de tal forma que você pode usar essa solução milhares de vezes, sem nunca vir a repeti-lo." (ALEXANDER et al., 1997)

2.10.5 Singleton

Este Padrão de projeto garante que uma classe tenha apenas uma instância, e forneça um ponto global de acesso a ela. (BOOCH, 1991)

2.10.6 Object Pool

Este padrão de projeto segundo Alan, ele gerencia o reuso de objetos quando eles consomem muito recurso para criar ou quando há um limite no numero de um tipo específico de objetos que podem ser criados. (SHALLOWAY; Trott, 2004)

3 TECNOLOGIAS E FERRAMENTAS UTILIZADAS

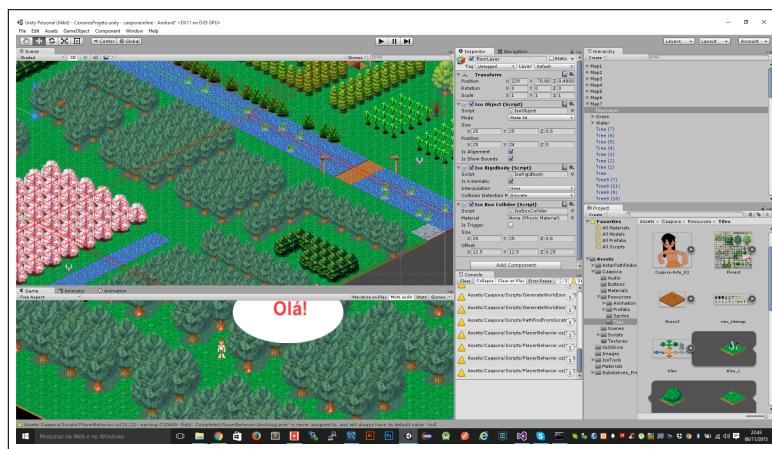
3.1 UNITY 3D – AMBIENTE DE DESENVOLVIMENTO

Unity é uma ferramenta com múltiplas utilidades na qual permite o usuário a desenvolver desde um jogo simples até de um de última geração.

Segundo o próprio site, onde é fornecida a ferramenta, informa que o Unity é um motor de desenvolvimento integrado que fornece uma funcionalidade pioneira para criação de jogos e outros conteúdos interativos. Poderá utilizar o Unity para montar sua arte e recursos em cenas e ambientes; adicionar física, editar e testar simultaneamente seu jogo e, quando preparado, publicar em suas plataformas escolhidas, tais como computadores fixos, a rede, iOS, Android, Wii, PS3 e Xbox 360.

O Unity suporta três linguagens de programação que são Boo, JavaScript e o C#. O Unity também possui módulos que podem ser instalados possibilitando o desenvolvimento sem linha de código. (UNITY, 2015) Na figura 15 é demonstrado o uso da Engine Unity 3D para o desenvolvimento do jogo.

Figura 15 – Uso da engine Unity 3D para desenvolvimento do jogo Caapora



Fonte: Elaborado pelo autor

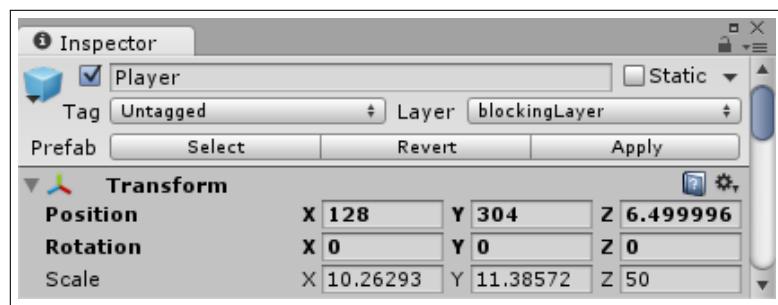
Umas das principais vantagens de se utilizar o Unity é a possibilidade de desenvolver o jogo uma única vez e poder utilizá-lo em mais de 10 plataformas diferentes sem as necessidades de alterar o produto inicial. Algumas destas plataformas são: iPads, PC e iPhone. (UNITY, 2015)

3.1.1 Scene

Scene (traduzido do inglês livre: cena) é a parte mais importante dentro do jogo, pois cada uma representa uma fase dentro do jogo. Essa cena é constituída de vários objetos chamados de *GameObject*, que são recipientes onde pode ser agregados componentes.

Um *gameObject* possui por padrão um componente chamado *transform*, que tem como objetivo definir a posição, a rotação e a escala do objeto, ou seja sem ele o objeto não existiria dentro do jogo. Na figura 16 temos o *gameObject* do *Player* acompanhado do componente *transform*. (UNITY, 2015)

Figura 16 – GameObject do Player



Fonte: Elaborado pelo autor

3.2 LINGUAGEM DE PROGRAMAÇÃO C#

C# é uma linguagem elegante e de tipos protegidos, orientada a objeto e que permite aos desenvolvedores construir uma variedade de aplicações seguras e robustas, compatíveis com o .NET Framework. É possível usar C# para criar muitos aplicativos de cliente do Windows, serviços Web XML, componentes distribuídos, aplicativos de cliente-servidor, aplicativos de banco de dados, e muito mais.

O Visual C# fornece um editor de códigos avançado, designers de interface de usuário convenientes, depurador integrado, e muitas outras ferramentas para facilitar o desenvolvimento de aplicativos baseados na linguagem C# e no .NET Framework.

A linguagem C# foi desenvolvida pela Microsoft para ser um framework .NET. O C# é similar ao C++ e Java, sendo também uma linguagem fortemente tipada, orientada a objeto, portanto suporta heranças, polimorfismo e encapsulamento. Algumas características desta linguagem são:

- Controle de versão: cada assembly gerado, seja como EXE ou DLL, tem informação sobre a versão do código.
- Orientada a objetos: em C#, qualquer variável tem de fazer parte de uma classe.
- Linguagem gerenciada: todo gerenciamento de memória é feito pelo *runtime* via GC (GarbageCollector), e não diretamente pelo programador, reduzindo assim as chances de cometer erros comuns.

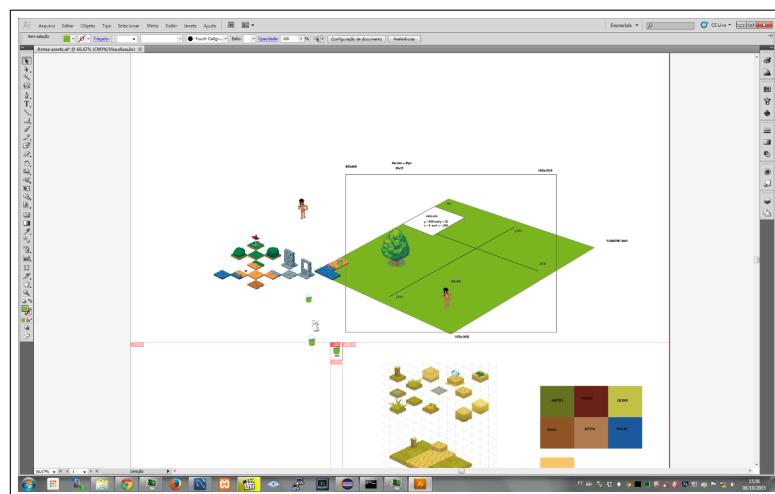
(MICROSOFT, 2015)

3.3 ADOBE ILLUSTRATOR CC

A Adobe Illustrator CC é um software de gráficos vetoriais padrão do setor, usado em todo o mundo por designers de todos os tipos que querem criar gráficos digitais, ilustrações e tipografia para todos os tipos de mídia: impressão, web, interativa, vídeo e móvel. (ADOBE, 2015)

Na figura 17 temos a representação do cenário dentro do Adobe Illustrator.

Figura 17 – Utilização do Adobe Illustrator CC no jogo Caapora



Fonte: Elaborado pelo autor

3.4 VISUAL STUDIO 2015

Visual Studio é um ambiente de desenvolvimento da Microsoft na qual permite desenvolver aplicativo para plataformas WEB, Windows, Mac e Linux. É uma ferramenta direcionada para desenvolver em linguagem de programação C# e o *framework* .NET.

Segundo o próprio site da Visual Studio, a empresa define a ferramenta como: "Um ambiente de desenvolvimento integrado e sofisticado para criação de aplicativos impressionantes para Windows, Android e iOS, de aplicativos Web modernos e serviços de nuvem. "

Uma grande vantagem em se utilizar o Visual Studio 2015 é permitir o desenvolvimento em multiplataforma e também a disponibilidade de variadas extensões, de PHP a serviços de jogos. (STUDIO, 2015)

4 ANÁLISE E PROJETO DO SISTEMA

4.1 ANÁLISE DE PRODUTOS SEMELHANTES

Um dos jogos que possui objetivos semelhantes ao do Caapora RPG e que coincidiu com a motivação do desenvolvimento é o jogo *Pora: Free Cockatoos* representado na figura 18.

A *Indonesian Society for Animal Welfare* (traduzido do inglês livre Sociedade Indonésio para o Bem Estar dos Animais)junto com outros desenvolvedores criaram um aplicativo para conscientizar e educar em massa. “Muitas pessoas agora usam smartphones desde muito jovens”, diz Kinanti Kusumawardani, diretor executivo da *Indonesian Society for Animal Welfare*. “Jogos para celular são um meio para chamar a atenção de maneira lúdica, uma vez que as pessoas parecem estar menos interessadas em ouvir palestras sobre o assunto”.

Para o desenvolvimento do jogo foi realizada uma competição em que os desenvolvedores deveriam criar um jogo que estimulasse a preservação da vida selvagem.

O Jogo *Pora: Free Cockatoos* tem como personagem principal um peixe chamado Pora que possui um canhão onde ele lança torpedos aos caçadores que aprisionam as cacatuas. Os vilões prendem as aves dentro de garrafas plásticas, uma prática maldosa que realmente acontece.

“Os jogos podem fazer com que usuários se sintam envolvidos por uma causa sem que pareça algo muito sério”, afirmou Andi Surja Boediman, um parceiro da empresa que trabalha com os desenvolvedores. “É eficaz para promover a conscientização” (ANDA, 2015)

Figura 18 – Jogo Pora: Free Cockatoos desenvolvido para conscientização dos mal tratos as aves da Indonésia



Fonte: (ANDA, 2015)

4.2 REQUISITOS DO SISTEMA

4.2.1 Requisitos Funcionais

Foram definidos os seguintes requisitos funcionais, conforme exibido na Tabela 2:

Tabela 2 – Requisitos Funcionais

Requisito	Função
RF1	O jogo deve ter um menu inicial contendo as seguintes opções: Iniciar Jogo e Opções.
RF2	Ao iniciar o jogo, o jogo deve apresentar um resumo sobre o cenário ao qual o jogador estará durante o jogo.
RF3	O jogo deve ter a opção de Pular o resumo do inicio do jogo.
RF4	A movimentação do personagem será controlada através do <i>touchscreen</i> com controle direcional.
RF5	Usuário terá a capacidade de pegar o balde de aguá quando o ícone da mão for tocado.
RF6	Usuário terá a capacidade jogar água no fogo apertando na tela o botão de ataque.
RF7	O jogo apresentará uma barra de vida do personagem Caapora que irá diminuir toda vez que ele sofrer dano
RF8	O balde deverá ter uma barra de informação que informará a quantidade de água que já foi enchido
RF9	Caso o usuário consiga apagar todo o fogo no tempo determinado, será mostrada uma tela dizendo que o jogador conseguiu
RF10	Caso o usuário não consiga apagar todo o fogo em tempo determinado, será mostrada uma tela dizendo que o jogador falhou
RF11	Caso o jogador deixe o personagem sofrer dados que o faça perder todos os pontos de vida será mostrada uma tela dizendo que o jogador falhou
RF12	Caso o jogador deixe o personagem amigo sofrer danos que o faça perder todos os pontos de vida será mostrada uma tela dizendo que o jogador falhou
RF13	Os personagens amigos do personagem principal devem segui-lo em sua jornada
RF14	Os pontos de vida, experiência e nível devem ser exibidos em tempo-real
RF15	Os jogo deve exibir dicas por um período de tempo de como proceder em cada situação na primeira fase
RF16	A câmera principal do jogo deve perseguir o personagem principal em sua jornada

Fonte: Elaborado pelo autor

4.2.2 Requisitos Não Funcionais

Foram definidos os seguintes requisitos não funcionais, conforme exibido na Tabela

3:

Tabela 3 – Requisitos Não Funcionais

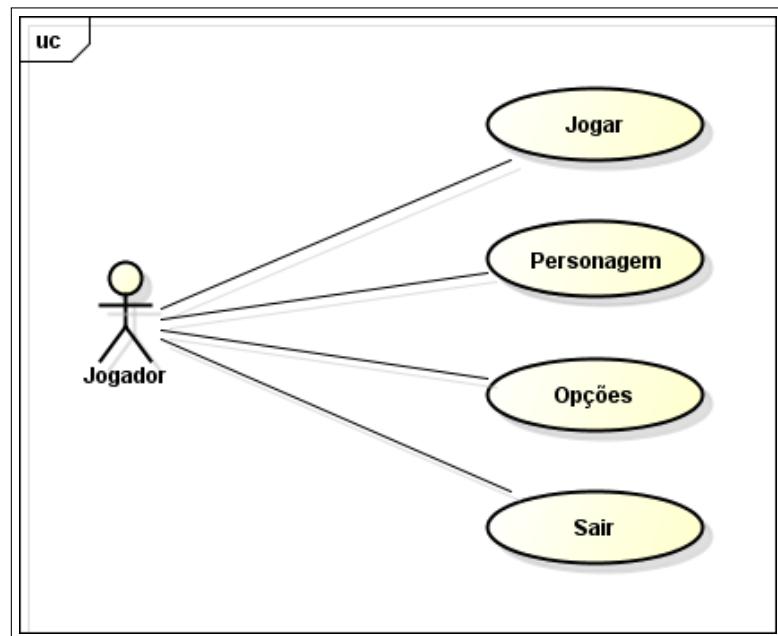
Requisito	Função
RNF1	O aplicativo deve funcionar na versão 3.0 do Android ou superior.
RFN2	O carregamento do jogo deve durar no máximo 20 segundos.
RFN3	A tela do dispositivo móvel deve ser <i>touchscreen</i> e com resolução mínima de 480 por 800 pixels
RFN4	A interface gráfica deve ser de fácil utilização
RFN5	O jogo deve possuir portabilidade para múltiplas plataformas
RFN6	O jogo deve rodar com no mínimo com 30 FPS

Fonte: Elaborado pelo autor

4.3 MODELAGEM DO SISTEMA

A figura 19 apresenta o caso de uso do jogo, apresentando as ações que o jogador pode executar na aplicação.

Figura 19 – Caso de uso, ações do jogador



Fonte: Elaborado pelo autor

4.3.1 Caso de uso Jogar

A tabela 4 representa o caso de uso "Jogar" que detalha a interação do jogador com a aplicação sempre que a opção Jogar for selecionada.

Tabela 4 – Documentação do caso de uso Jogar

Nome do Caso de uso	Jogar
Ator Principal	Jogador
Resumo	Esse caso de uso descreve as etapas percorridas por um jogador após ele selecionar a opção jogar.
Pré - condições	Não se aplica
Ações do Ator	Ações do Sistema
1- O jogador seleciona a opção "Jogar" no menu principal. 3 - Jogador seleciona a opção "Pular" após ler as mensagens. 5 - Jogador aguarda o carregamento do cenário	2 - Sistema exibe uma tela contendo uma mensagem informativa sobre o jogo. 4 - Sistema carrega o cenário do jogo. 6 - Sistema inicia o jogo.

Fonte: Elaborado pelo autor

4.3.2 Caso de uso Personagem

A tabela 5 representa o caso de uso "Personagem" que detalha a interação do jogador com a aplicação sempre que o jogador desejar trocar o personagem dentro do jogo.

Tabela 5 – Documentação do caso de uso Personagem

Nome do Caso de uso	Personagem
Ator Principal	Jogador
Resumo	Esse caso de uso descreve as etapas percorridas por um jogador durante a seleção de personagem
Pré - condições	Não se aplica
Ações do Ator	Ações do Sistema
1- O jogador seleciona a opção "Personagem" na interface 3 - Jogador seleciona o personagem desejado 5 - Jogador aguarda a validação do sistema	2 - Sistema exibe uma tela contendo personagens 4 - Sistema valida a escolha do jogador 6 - Sistema retorna ao menu principal.

Fonte: Elaborado pelo autor

4.3.3 Caso de uso Opções

A tabela 6 representa o caso de uso "Opções" que detalha a interação do jogador com a aplicação sempre que o jogador quiser mudar alguma configuração.

Tabela 6 – Documentação do caso de uso Opções

Nome do Caso de uso	Opções
Ator Principal	Jogador.
Resumo	Esse caso de uso descreve as etapas percorridas por um jogador após ele selecionar "Opções".
Pré - condições	Não se aplica.
Ações do Ator	Ações do Sistema
1- O jogador seleciona a opção "Opções" na interface	2 - Sistema exibe uma tela contendo informações sobre áudio e vídeo.
3 - Jogador seleciona o item que deseja alterar	4 - Sistema exibe o item escolhido
5 - Jogador altera as configurações do item escolhido	6 - Sistema valida as alterações
7 - Jogador seleciona a opção voltar ao menu principal	8 - Sistema retorna ao menu principal.

Fonte: Elaborado pelo autor

4.3.4 Caso de uso Sair

A tabela 7 representa o caso de uso "Sair" que detalha a interação do jogador com a aplicação sempre que o jogador desejar sair do jogo.

Tabela 7 – Documentação do caso de uso Sair

Nome do Caso de uso	sair
Ator Principal	Jogador.
Resumo	Esse caso de uso descreve as etapas percorridas por um jogador após ele selecionar "Sair".
Pré - condições	Não se aplica.
Ações do Ator	Ações do Sistema
1- O jogador seleciona a opção "Sair" na interface	2 - Sistema fecha o jogo.

Fonte: Elaborado pelo autor

5 DESENVOLVIMENTO

Neste capítulo, serão apresentados os conceitos envolvidos no desenvolvimento do jogo.

5.1 CENAS DO JOGO

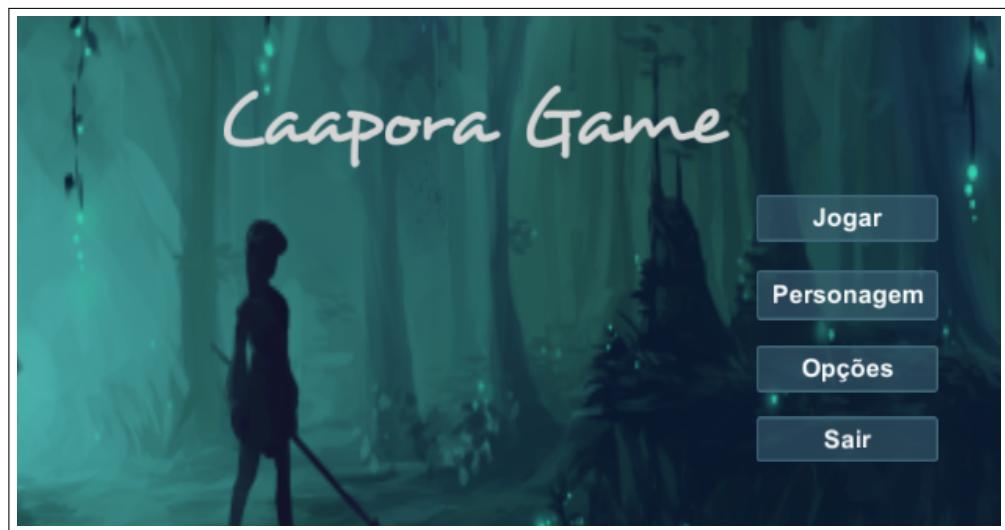
O jogo está dividido em três cenas, que são: Menu Principal, Tutorial e Jogo.

5.1.1 Menu Principal

O menu principal é a primeira tela que o usuário terá contato ao abrir a aplicação. Nela o usuário poderá começar o jogo, selecionar personagem, alterar configuração de som e sair do jogo.

A figura 20 apresenta a tela inicial de jogo que será apresentada ao usuário.

Figura 20 – Menu Principal



Fonte: Elaborado pelo autor

5.1.1.1 Botões

Para criar os botões no menu foi utilizado um componente chamado UI, ele é um sistema que permite criar interfaces rápida e intuitivas.

5.1.1.2 Título

Para criar o título foi usado um componente do sistema UI, chamado text que permite inserir um texto dentro de uma determinada área. Ele também permite editar alguns detalhes como tipo de fonte, tamanho e cor.

5.1.1.3 Imagem de fundo

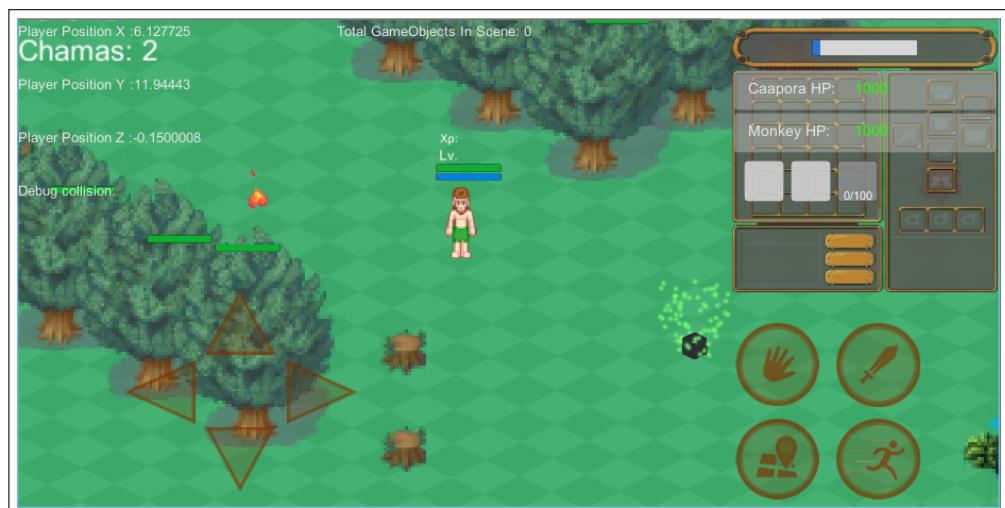
Para inserir a imagem de fundo foi preciso adicionar um componente do UI chamado Panel. Ele delimita a área da do jogo, facilitando assim a inserção da imagem de fundo.

5.1.2 Jogo

A cena do jogo foi construída tendo como base o mapa isométrico, que é um cenário mais próximo do 3D. Foi utilizado alguns recursos como: *Tilesets*, *Sprites*, Animação, *Collider* e *Rigidbody*.

A figura 21 mostra o cenário completo.

Figura 21 – Cenário do jogo Caapora RPG



Fonte: Elaborado pelo autor

5.1.2.1 Tilesets

Os *tilesets* são elementos gráficos dentro de uma imagem. Eles possuem tudo o que é preciso para a construção de mapas e são classificados de acordo ao tipo de mapa que será

criado, no caso do jogo caipora, foi usado um *tileset* da floresta.

Na figura 22 temos o *tileset* usado no jogo.

Figura 22 – Tileset utilizado na criação do cenário



Fonte: (MAKER, 2015)

5.1.2.2 Sprites

Os *sprites* são elementos gráficos dentro de uma única imagem que tem como objetivo criar animações. Eles foram usados na criação dos personagens e dos animais.

A figura 23 mostra um dos *sprites* do caipora.

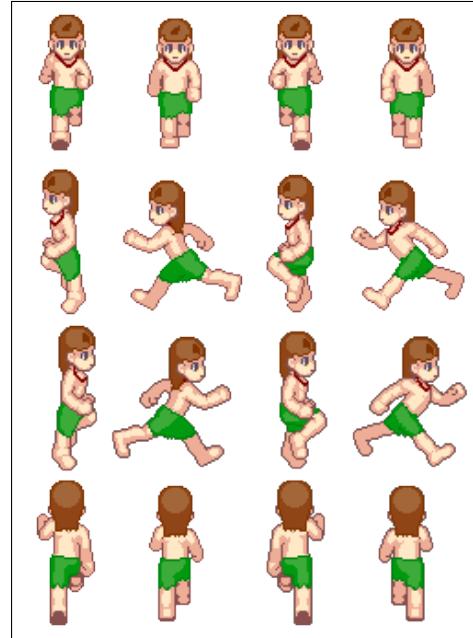
5.1.2.3 Animação

A animação, tanto dos personagens quanto dos animais é criado dentro de um componente do Unity chamado Animation. Nele é criado um clipe de animação para cada ação do personagem, onde os *sprites* são inseridos individualmente em uma linha do tempo e é determinada a velocidade que terá a animação.

A figura 24 mostra a criação do clipe de animação “Caapora-left”, onde temos uma sequencia de *sprites* inseridos na linha do tempo.

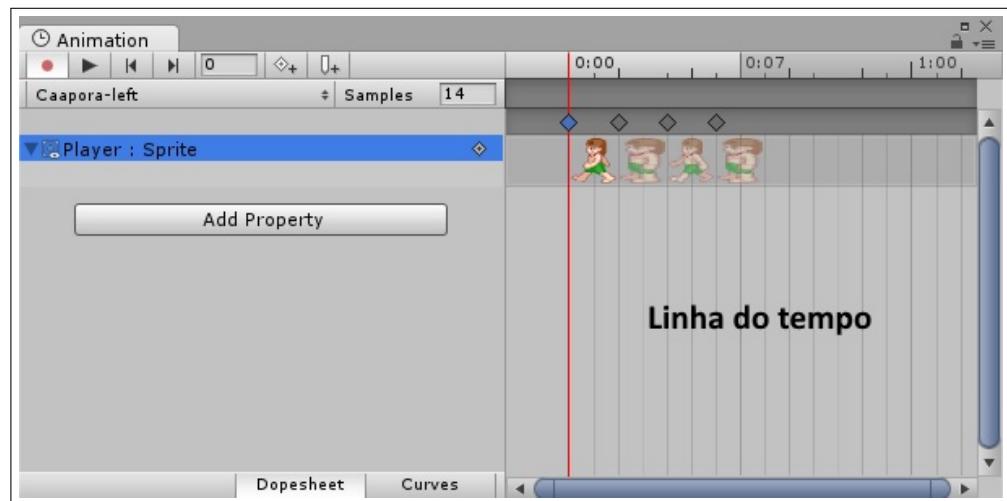
Após a criação dos clipes de animação, é necessário criar as transições entre eles. Para isso o Unity fornece um componente chamado *Animator*. No *animator*, conforme vemos na

Figura 23 – Sprite do caipora correndo



Fonte: Elaborado pelo autor

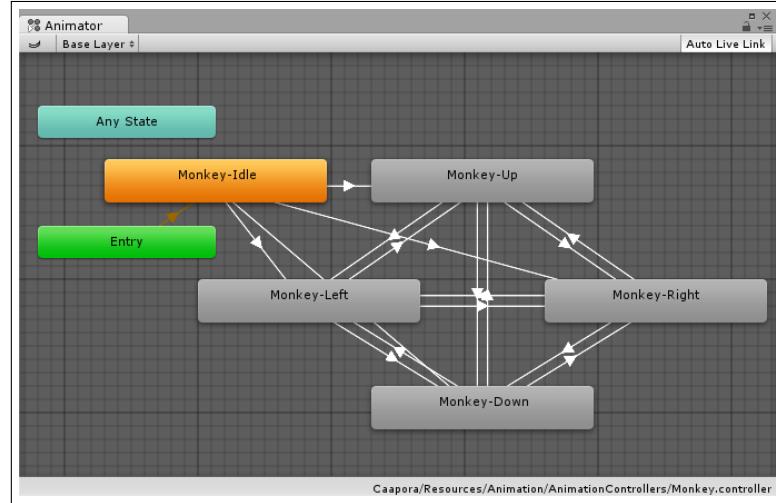
Figura 24 – Componente Animation



Fonte: Elaborado pelo autor

figura 25, os clipes de animação são ligados através das transições, que são as setas indo de um clipe para outro, e é criado condições para que haja a transição, ou seja, sempre que houver uma condição dentro do jogo, haverá uma mudança entre os clipes de animação.

Figura 25 – Componente Animator



Fonte: Elaborado pelo autor

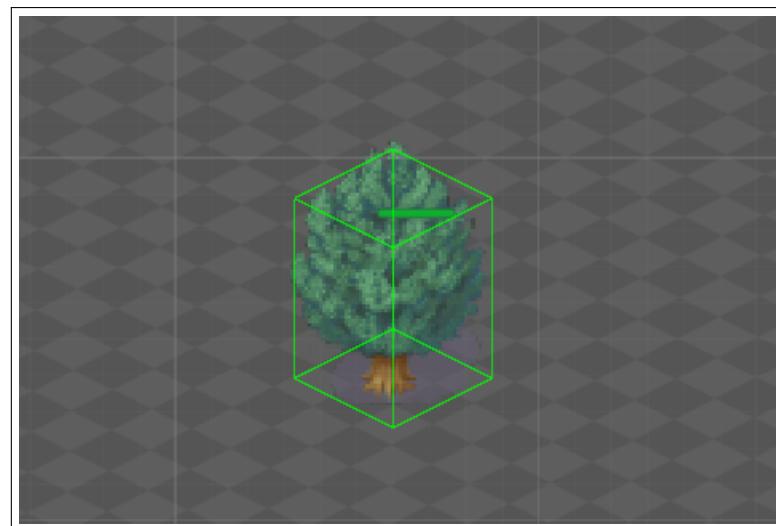
5.1.2.4 Collider

O colisor é um componente que impede que um objeto transpasse o outro, ele define a forma de um objeto para fins de colisões físicas.

Ele foi utilizado tanto nos personagens quantos nos objetos do cenário afim de evitar que esses objetos não atravessassem os demais.

Na figura 26 temos um tile e o elemento de colisão em volta. As linhas que traçam um retângulo representam o colisor.

Figura 26 – Elemento de colisão em volta de um tile



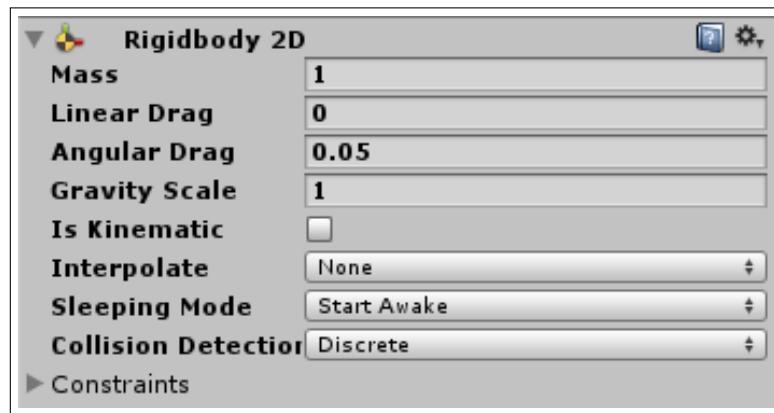
Fonte: Elaborado pelo autor

5.1.2.5 Rigidbody

O *Rigidbody* é um componente que permite um comportamento físico em um determinado objeto. Quando esse componente é anexado ao objeto, automaticamente esse objeto vai responder a gravidade.

A figura 27 mostra o componente *Rigidbody*.

Figura 27 – Componente Rigidbody



Fonte: Elaborado pelo autor

5.2 MECANISMOS

5.2.1 Pegar Balde

Para implementar este mecanismo foram utilizados os componentes *BoxCollider* do personagem e do elemento balde que ao identificar a colisão um com o outro, permite que o caipora pegue o balde inserindo-o em seu inventário retirando o objeto balde do cenário dando a impressão do objeto ter sido coletado.

O estado do personagem então muda para um estado que identifica ele com o balde e da mesma forma, habilita a animação do caipora para a animação com o balde permitindo que ele colete a água. Na figura 28 é possível visualizar o personagem Caipora com o balde capturado.

Figura 28 – Caipora com o balde



Fonte: Elaborado pelo autor

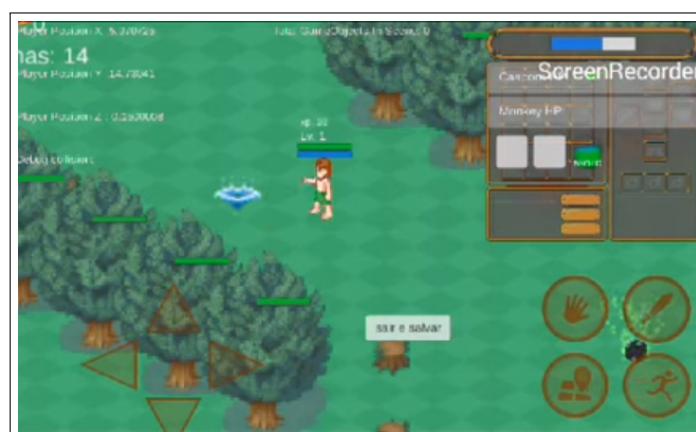
5.2.2 Encher o balde e Jogar Água

Após o Caipora ter pegado o balde ele deverá enche-lo. Para encher o balde o personagem deve ter em seu inventário o balde e ao colidir com o objeto que representa a água um contador é incrementando e o painel de inventário da tela indica visualmente o nível de água que o balde encheu.

De acordo com a quantidade de água disponível no balde o personagem então poderá lançar no cenário objetos do tipo borrifos de água, que se auto destrói, usando o método `Destroy()` e o recurso de *Coroutines* do motor de jogos Unity3D, logo após ser lançado.

A figura 29 apresenta o Caipora jogando água nas chamas.

Figura 29 – Caipora lançando água nas chamas



Fonte: Elaborado pelo autor

5.2.3 Chamas

São objetos que representam as chamas no mundo real e que causam destruição por onde passam, as chamas se multiplicam pelo cenário automaticamente, com o uso de *Coroutines* do motor de jogos Unity3D, em linha ou em círculos e conforme a dificuldade do jogo eles se espalham mais rapidamente.

As chamas possuem pontos de danos na classe que as representa que são usados para reduzir os pontos de vida de qualquer personagens vivo no jogo, incluindo árvores. Este comportamento se encontra em todos os seres vivos do jogo que herdam a classe *Creature*.

Para destruir este tipo de objetos é necessário que ele utilize o objeto do tipo borrifão de água, que é lançado pelo caipora, causando o efeito de ter apagado as chamas com a água borrifada.

A figura 30 apresenta as chamas se espalhando.

Figura 30 – Fogo se alastrando



Fonte: Elaborado pelo autor

5.2.4 Pathfinding

O algoritmo de busca de melhor caminho (*Pathfinding*) A* foi usado em duas situações no jogo; para personagens NPC (*Non-Player Character*) amigos do personagem principal e para os inimigos que seguem o caipora e seus amigos com o objetivo de ataca-los. Este recurso faz com que os personagens do jogo se movimentem sozinhos no mapa, melhorando consideravelmente a jogabilidade.

A figura 31 apresenta o personagem NPC *Monkey* seguindo o Caipora com um

caminho formado pelo algoritmo A*.

Figura 31 – Exemplo de NPC

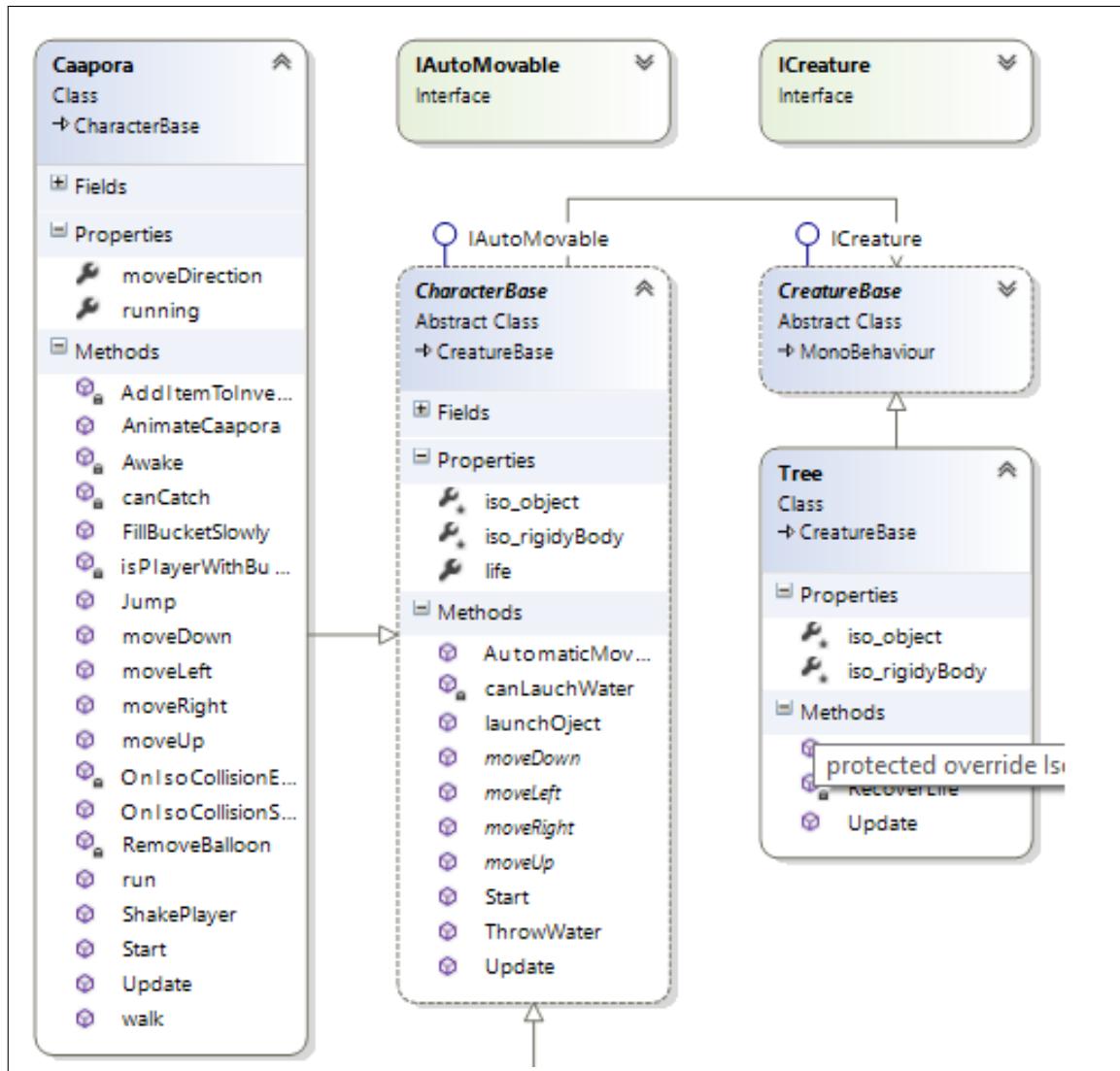


Fonte: Elaborado pelo autor

5.3 ESTRUTURA DE CLASSE

A figura 32 apresenta a estrutura de Classes dos Personagens utilizada para o desenvolvimento do jogo.

Figura 32 – Estrutura de Classes de Personagens



Fonte: Elaborado pelo autor

5.3.1 GameManager

Esta classe é responsável por gerenciador todo o ciclo de vida do jogo para isso ela utiliza o padrão de projeto *Singleton* que garante apenas uma instância para um objeto durante toda a execução do jogo. Ela que implementa a maior parte da regra de negócio do jogo incluindo condições de vitórias e o fim de jogo.

5.3.2 Enemy

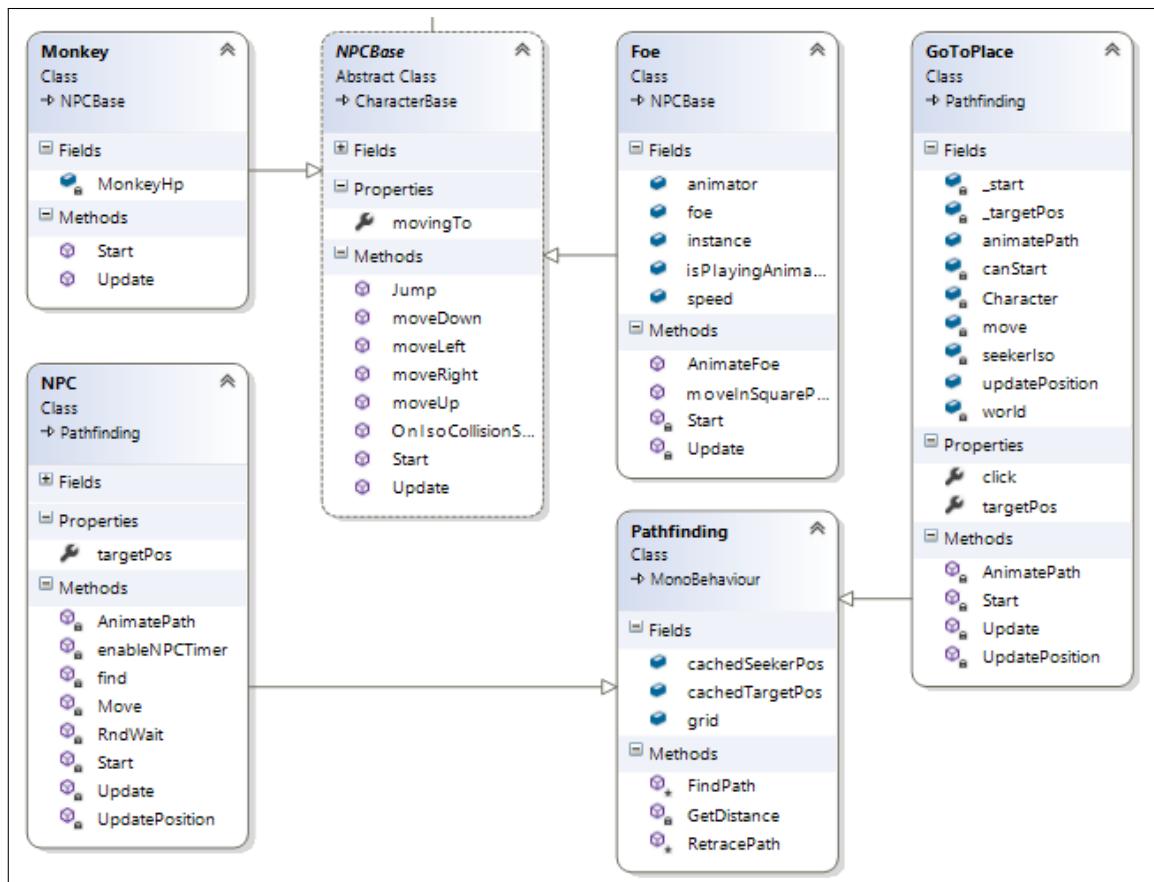
Classe que representa os estados e comportamentos dos inimigos no jogo.

5.3.3 NPC

Personagens deste tipo implementam o algoritmo de *pathfinding* permitindo que movimentem-se sozinhos no jogo, possuindo uma inteligência artificial. Utiliza os métodos da classe *Pathfinding* para personagens que perseguem o Caipora, é utilizando tanto amigos quanto em inimigos.

A figura 33 apresenta a estrutura de Classes dos NPCs utilizada para o desenvolvimento do jogo.

Figura 33 – Estrutura de Classes de NPC's



Fonte: Elaborado pelo autor

5.3.4 Inventory

Esta classe tem por finalidade armazenar os objetos coletados pelo personagem principal e exibir suas características. Como exemplo temos o balde que no início do jogo o caipora tem que pegá-lo e enche-lo para apagar as chamas que estão queimando as árvores.

5.3.5 Tree

Esta classe representa as árvores no cenário e possui características herdadas da classe abstrata CreatureBase, possuindo assim pontos de vida e ataque podendo morrer ao sofrer danos.

5.3.6 Water

Classe que representa a água no cenário, ao colidir com o fogo apaga-o. Ao colidir com o Caipora com o balde aumenta a quantidade de água no balde.

5.3.7 Fire

Classe que representa um fogo no *game*, possui pontos de danos que são utilizados quando objetos do tipo seres vivos colidem com ele reduzindo seus pontos de vida.

5.3.8 SpreadFire

Classe que automatiza a expansão do fogo pelo cenário, extraíndo as instâncias do *Object Pool* e inserindo no mapa tanto em linha quanto em círculos.

5.4 FUNCIONAMENTO DO JOGO

A aplicação do Caapora RPG se inicia no menu principal, onde o jogador pode iniciar o jogo, escolher um personagem, alterar configurações e sair do jogo.

Ao selecionar a opção jogar, é apresentado ao jogador a tela do tutorial, onde terá uma breve descrição do que o jogador deverá fazer durante o jogo.

Dentro da tela do jogo, a missão do jogador é apagar o fogo que está destruindo a mata, para isso ele deverá pegar um balde que está próximo ao rio e encher, com isso o jogador se dirige até o local onde está pegando o fogo e lança a aguá. A missão termina quando todo o fogo na floresta for apagado. Os detalhes da missão estão descritos no *Game Design Document* que está no Apêndice.

6 CONCLUSÃO

Este projeto tem o objetivo de conscientizar as crianças sobre o desmatamento que ocorre em todo Brasil, principalmente na Floresta Amazônica. Para atingir esse objetivo, foi desenvolvido um jogo que é uma atividade rica em aprendizagem com o intuito de mostrar o que o país está enfrentando e o quanto isso é grave.

O jogo possui características importantes, já que o gênero (RPG com elementos de ação), faz com que o jogador assuma o personagem dentro do jogo, dando a ele uma perspectiva do que a natureza está enfrentando.

Todo o desenvolvimento do jogo foi criado dentro do ambiente da Unity, isso devido à facilidade que o software oferece, como uma grande variedade de componentes que auxiliam na criação do jogo e também devido a ele ser multiplataforma, ou seja, apesar de ele ser para a plataforma Android, o jogo pode ser migrado para uma outra plataforma.

Com relação a performance, o jogo atingiu os requisitos de rodar à 30 quadros por segundo, visto que em jogos 2D onde são usados clipes de animação mais simples a taxa de 12 FPS é a mínima aceitável.

Por fim, o projeto atendeu as expectativas visto que a principal mensagem que o jogo deveria transmitir foi alcançada, já que ele ilustra bem o que está ocorrendo na floresta, assim como também foi utilizado as tecnologias e ferramentas atuais.

6.1 TRABALHOS FUTUROS

Como trabalhos futuros a serem melhorados e implementados dentro do jogo, estimase as seguintes funcionalidades:

- Criação de mais 3 cenários;
- Inclusão de mais animais;
- Criação do Curupira, que será o segundo personagem que o jogador poderá escolher;
- Melhoria do sistema de inteligência artificial utilizado no jogo, visando a realidade dos movimentos executados pelo IA;

REFERÊNCIAS

- ADOBE. **Illustrator CC Perguntas frequentes**. 2015. Disponível em: <<https://helpx.adobe.com/br/illustrator/faq.html>>.
- ALEXANDER, C.; ISHIKAWA, S.; SILVERSTEIN, M. **A Pattern Language: Towns, Buildings, Construction**. [S.l.]: Oxford University Press, 1997.
- ALEXANDRE. **O que é Google Play?** 2012. Disponível em: <<http://smartmundo.com.br/o-que-e-play-store/>>.
- AMBROS, L. **Diferença entre Aplicativos Nativos, Híbridos e Mobile Web Apps**. 2013. Disponível em: <<http://www.luisaambros.com/blog/diferenca-entre-aplicativos-nativos-hibridos-e-mobile-web-apps/>>.
- ANDA. **ONG cria jogos para celular como forma de educar contra a caça**. 2015. Disponível em: <<http://www.anda.jor.br/07/09/2015/ong-cria-jogos-celular-forma-educar-caca%E2%80%8F>>.
- ARRUDA, E. P. **Fundamentos para o Desenvolvimento de Jogos Digitais**. [S.l.]: Bookman, 2014.
- ART, D. **Caipora**. 2015. Disponível em: <<http://light-schizophrenia.deviantart.com/art/Caipora-172961182> and <http://www.deviantart.com/art/Caipora-24074862>>.
- BENFICA, A. **O que é Smartphone?** 2011. Disponível em: <<http://www.telefonescelulares.com.br/o-que-e-smartphone/>>.
- BOOCH, G. **Object Oriented Design: With Applications**. [S.l.]: Addison-Wesley Professional, 1991.
- CARVALHO, G. **Como publicar no Google Play**. 2012. Disponível em: <<http://www.tutoriandroid.com/2012/05/como-publicar-no-google-play.html>>.
- CLARK, D. **Beginning C Object-Oriented Programming**. [S.l.]: Apress, 2013.
- COTA, T. T.; JUNIOR, N. V.; ISHITANI, L. **Impacto do gênero de jogo digital na motivação dos idosos para jogar**. 2014. Disponível em: <<http://www.anda.jor.br/07/09/2015/ong-cria-jogos-celular-forma-educar-caca%E2%80%8F>>.
- FLURRY. **Estudo aponta que usuários Android jogam 37 minutos por dia**. 2014. Disponível em: <<http://www.tudocelular.com/android/noticias/n42509/usuarios-android-jogam-37-minutos-todos-os-dias.html>>.
- FOLCLORE, C. N. de. **Carta do folclore barsileiro**. 1995. Disponível em: <<http://www.fundaj.gov.br/geral/folclore/carta.pdf>>.
- FRANCHINI, A. S. **As 100 melhores lendas do folclore brasileiro**. [S.l.]: L & PM Editores, 2011.
- FREIRE, R. **96,8% de todos os smartphones vendidos no planeta são Android ou iPhone**. 2015. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/08/968-de-todos-os-smartphones-vendidos-no-planeta-sao-android-ou-iphone.html>>.

GAMMA RICHARD HELM, R. J. J. V. E. **Design Patterns: Elements of Reusable Object-Oriented Software.** [S.l.]: Addison-Wesley, 1994.

GRAFICA, E. **DIBUJO ISOMETRICO Y PERSPECTIVA ISOMETRICA.** 2012. Disponível em: <<http://josegrass27.blogspot.com.br/2012/10/dibujo-isometrico-y-perspectiva.html>>.

HAMMERSCHMIDT, R. **Linha do Tempo: por dentro da evolução do Android.** 2015. Disponível em: <<http://www.tecmundo.com.br/android/82344-linha-tempo-dentro-evolucao-do-sistema-android.htm>>.

IGN. **Half-Life 2 Review.** 2004. Disponível em: <<http://www.ign.com/games/half-life-2/pc-492830,urldate={10/12/2011}>>.

KORJENIOSKI, M. **Desenvolvimento de Jogos 2D com Android.** Dissertação (Mestrado) — UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ, 2011.

LUCCHESE, F.; RIBEIRO, B. **Conceituação de Jogos Digitais.** 2014. Disponível em: <<http://www.dca.unicamp.br/~martino/disciplinas/ia369/trabalhos/t1g3.pdf>>.

LUISE, D. **Aplicativos inovam aprendizado e incentivam autonomia do aluno.** 2013. Disponível em: <<http://educacao.uol.com.br/noticias/2013/03/06/aplicativos-inovam-aprendizado-incentivando-postura-ativa-e-autonomia-do-aluno-conheca-jogos-e-programas.htm>>.

MAKER, M. R. **Tilesets de florestas (arvores, plantas, etc...).** 2015. Disponível em: <<http://www.mundorpgmaker.com.br/topic/110435-tilesets-de-florestas-arvores-plantas-etc>>.

MASSARANI, S. **Tennis for two: O primeiro Videogame.** 2013. Disponível em: <http://www.massarani.com.br/FGHQ_Tennisfortwo.html>.

MAUDA, E. **Controle de Versão de Códigos – BitBucket – Parte 1.** 2015. Disponível em: <<http://www.mauda.com.br/?p=297>>.

MESSIAS, J. **Desmatamento na Floresta Amazônica aumenta mais de 400 %.** 2015. Disponível em: <vhttp://g1.globo.com/jornal-hoje/noticia/2015/01/desmatamento-na-floresta-amazonica-aumenta-mais-de-400.html,urldate={14/12/2011}>.

MICROSOFT. **Introdução à linguagem C # e ao .NET Framework.** 2015. Disponível em: <<https://helpx.adobe.com/br/illustrator/faq.html>>.

MICROSOFT, N. **Microsoft Licenses Flight Simulator Game Franchise To Dovetail Games, New Game Coming In 2015.** 2014. Disponível em: <<http://microsoft-news.com/microsoft-licenses-flight-simulator-game-franchise-to-dovetail-games-new-game-coming-in-2015/>>.

NIELSEN. **68 milhões usam a internet pelo Smartphone no Brasil.** 2015. Disponível em: <<http://www.nielsen.com/br/pt/press-room/2015/68-milhoes-usam-a-internet-pelo-smartphone-no-Brasil.html>>.

OTAKU, F. F. **Final Fantasy XIII - Wallpapers.** 2015. Disponível em: <<http://www.ffotaku.com/final-fantasy-xiii/wallpapers.php>>.

- PINHEIRO, M. Brasil decola na indústria de apps e mercado acumula lucros nesta década.** 2015. Disponível em: <http://www.correio braziliense.com.br/app/noticia/tecnologia/2015/05/11/interna_tecnologia,482694/brasil-decola-na-industria-de-apps-e-mercado-acumula-lucros-nesta-deca.shtml>.
- PIRES, R.; VIRGILI, R. Mercado de aplicativos é um oásis de crescimento econômico.** 2015. Disponível em: <<http://fatoonline.com.br/conteudo/1311/mercado-de-aplicativos-e-um-oasis-de-crescimento-economico?or=topico&p=to&i=6&v=1>>.
- POO DEREK KIONG, S. A. D. Object-Oriented Programming and Java.** [S.l.]: Springer Science Business Media, 2007.
- RAMOS, A. Controle de Versão, Git, Github e Bitbucket – Afinal, o que é tudo isso ?** 2015. Disponível em: <<https://aprendendowww.wordpress.com/2015/02/20/controle-de-versao-git-github-e-bitbucket-afinal-o-que-e-tudo-isso/>>.
- RAMOS, H. M. A historia do jogos de computadores.** Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2007.
- RICH, E.; KNIGHT, K. Inteligência artificial.** [S.l.]: Makron Books, 1993.
- RUSSELL, S.; NORVIG, P. Artificial intelligence: a modern approach.** [S.l.]: Prentice Hall, 2003.
- SANCHES, R. O primeiro celular da história.** 2011. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2011/07/o-primeiro-celular-da-historia.html>>.
- SANTOS, R. M. Analise do contexto móvel nos teste de usabilidade de aplica coes moveis.** 2011. Disponível em: <<http://www.repositoriobib.ufc.br/000012/00001234.pdf>>.
- SHALLOWAY, A.; Trott, J. R. Design Patterns Explained: A New Perspective on Object-Oriented Design.** [S.l.]: Addison-Wesley Professional, 2004.
- SILVA, J. B. da. Estudo entre algoritmo A* e busca em largura para planejamento de caminho de personagens em jogos do tipo Pacman.** Dissertação (Mestrado) — Universidade Regional de Blumenau, 2005.
- SILVA, M. P. R.; COSTA, P. D. P.; PRAMPERO, P. S.; FIGUEIREDO, V. A. de. Elementos de Computacao Grafica utilizados em Jogos Digitais 2D.** 2009. Disponível em: <<http://www.dca.unicamp.br/~martino/disciplinas/ia369/trabalhos/t1g3.pdf>>.
- SIMAO, A. Inteligencia artficial: Aplicada a jogos digitais.** 2013. Disponível em: <<http://ambs.perl-hackers.net/sebentas/iaj/>>.
- SOMMERVILLE. Engenharia de software.** [S.l.]: Pearson Education, 2007.
- STUDIO, V. Visão geral dos produtos do Visual Studio 2015.** 2015. Disponível em: <<https://helpx.adobe.com/br/illustrator/faq.html>>.
- TERRA, M. por. Quem é a caipora?** 2014. Disponível em: <<http://www.mundoporterra.com.br/educacional/quem-e-a-caipora/>>.
- TORRES, C. E. Palestra Mobilidade - Computação móvel, Dispositivos e Aplicativos 2013.** 2013. Disponível em: <<http://pt.slideshare.net/cetorres/palestra-mobilidade-computao-mvel-dispositivos-e-aplicativos-2013>>.

TRACCO, M. **Nolan Bushnell, o jogo nao acabou...** 2006. Disponível em: <<http://super.abril.com.br/tecnologia/nolan-bushnell-o-jogo-nao-acabou>>.

UNITY. **Engine Unity 3D.** 2015. Disponível em: <<https://unity3d.com/pt>>.

VASCONCELOS, A. M. L. de; ROUILLER, A. C.; MACHADO, C. Ângela F.; MEDEIROS, T. M. M. de. **Introdução à engenharia de software e à qualidade de software.** 2006. Disponível em: <<http://mtc-m16.sid.inpe.br/col/sid.inpe.br/jeferson/2004/12.10.09.14/doc/publicacao.pdf>>

VIEIRA, D. **Scrum: A Metodologia Ágil Explicada de forma Definitiva.** 2014. Disponível em: <<http://www.mindmaster.com.br/scrum/>>.

WALLPAPERS, H. **FIFA 15 GAME.** 2014. Disponível em: <http://www.hdwallpapers.in/fifa_15_game-wallpapers.html>.

WIKIPEDIA. **The two paddles return the ball back and forth. The score is kept by the numbers (0 and 1) at the top of the screen.** 2005. Disponível em: <<https://en.wikipedia.org/wiki/Pong#/media/File:Pong.png>>.

WIKIPEDIA. **Os eixos x e y simbolizam o espaço de duas dimensões.** 2009. Disponível em: <https://pt.wikipedia.org/wiki/Espa%C3%A7o_bidimensional#/media/File:Ball_taxi_metric.svg>.

WIKIPEDIA. **O tri-eixo cartesiano (geometria analítica tridimensional) é um ícone do espaço tridimensional.** 2011. Disponível em: <https://pt.wikipedia.org/wiki/Espa%C3%A7o_tridimensional#/media/File:Cartesien-system.svg>.

WIKIPEDIA. **Jogos Educacionais.** 2015. Disponível em: <https://pt.wikipedia.org/wiki/Jogos_educacionais>.

APÊNDICES

APÊNDICE A – Game Design Document

A.1 INTRODUCÃO

Este documento visa informar aspectos conceituais e artísticos do jogo Caapora RPG. O intuito do documento é estabelecer um objetivo comum para o desenvolvimento do jogo, auxiliando em duvidas gerais que possam aparecer nas fases de implementação do jogo em questão.

A.1.1 Conceito

Caapora RPG é um jogo 2.5D, monojogador, de *action/RPG* habituada no cenário que será a floresta Amazônica, onde o jogador incorpora o Caipora e junto com a Iara e o Boto cor-de-rosa precisam defender a floresta dos caçadores, lenhadores e mineradores.

A.1.2 Gênero

É um jogo no estilo *Action/RPG*, ou seja, um RPG com elementos de ação.

A.1.3 PÚBLICO ALVO

O jogo destina-se a jogadores com faixa etária dos 5 aos 12 anos, pois se trata de um jogo infantil e educativo.

A.1.4 Plataforma

O jogo será desenvolvido para dispositivos móveis, em especial para *smartphones*

A.2 HISTÓRIA E NARRATIVA

Descreve a história do jogo, buscando informar onde se passa o jogo e os principais personagens envolvidos.

A.2.1 Enredo

Após ter uma visão onde a floresta é destruída, o Caipora, Guardião da Floresta, decide deixar de ser uma lenda e passa a agir. Desde então ele tem como missão evitar tudo

aquilo que prejudique a Floresta Amazônica, como queimadas, desmatamento, poluição, entre outros. Para isso, ele contará com a ajuda de alguns membros como a Iara e o Boto.

A.2.2 Personagens

Descreve em profundidade os protagonistas principais do jogo, assim como os secundários que serão aqueles que ajudaram o personagem principal de algum modo.

A.2.2.1 Caipora

Caipora é o protagonista do jogo. É um índio conhecido como o “guardião da floresta” devido ao seu lado protetor. Sua missão é proteger a floresta Amazônica e os animais que nela habitam de inimigos, como os caçadores, mineradores, lenhadores e empresários que buscam de alguma forma se aproveitar da riqueza que há nela.

Ele possui um arco com flechas e devido a sua convivência com os animais adquiriu a habilidade de imita-los, com isso utiliza desses requisitos para atacar e amedrontar seus inimigos.

A.2.2.2 Curupira

Curupira é um personagem do jogo que poderá ser selecionado no lugar do Caipora. Ele é um habitante da mata de estatura pequena, possui cabelos avermelhados e seus pés são voltados para trás. Sua função é proteger árvores, plantas e animais. Ele possui uma lança e pode emitir sons e assobios agudos que amedrontam seus inimigos.

A.2.2.3 Iara

Iara é uma personagem do jogo que ajudará o Caipora em alguns dos seus objetivos. Ela é uma sereia que possui uma voz irresistível capaz de atrair quem se aproxima dela para o fundo do rio.

A.2.2.4 Boto Cor de Rosa

Boto cor-de-rosa é um personagem do jogo que ajudará o Caipora a combater os incêndios. Ele é um peixe que habita nos rios da floresta Amazônica. Possui uma habilidade que lhe permite lançar jatos de água

A.3 GAMEPLAY

Descreve os desafios encontrados pelo jogador e os métodos usados para superá-los.

A.3.1 Estruturas de Missões e desafios

O jogo será dividido em 4 missões que serão divididas entre os cenários do jogo.

A.3.1.1 Primeira Missão

Alguns pontos da floresta estão pegando fogo. Com isso o jogador deverá percorrer um caminho até o rio e pegar baldes de água para apagar o fogo. O jogador poderá recorrer ao especial do boto cor-de-rosa que é um jato de água, entretanto esse especial após seu uso terá um tempo de resfriamento. A missão termina quando não houver mais árvores pegando fogo na floresta.

A.3.1.2 Segunda Missão

Os lenhadores estão cortando árvores em certo ponto da floresta. Nesta Missão o jogador deverá percorrer o caminho até o local e utilizar o poder do caipora de imitar os animais para amedrontar os lenhadores e expulsar eles do local, entretanto eles irão para um outro local da floresta e novamente o jogador terá que ir até eles e amedrontar até que eles desistam.

A.3.1.3 Terceira Missão

Alguns mineradores estão poluindo o rio. Com isso, o jogador deverá atacar os mineradores utilizando a lança ou o arco com flechas. Cada ataque terá um tempo de resfriamento. Nesta missão o jogador poderá recorrer a Iara para atrair os mineradores para dentro do rio.

A.3.1.4 Quarta Missão

Empresários querem derrubar algumas árvores e destruir o habitat dos índios com o intuito de construir um parque de diversões. Nesta missão o jogador terá a ajuda dos índios, eles percorrerão a floresta criando armadilhas para destruir qualquer tipo de maquinário.

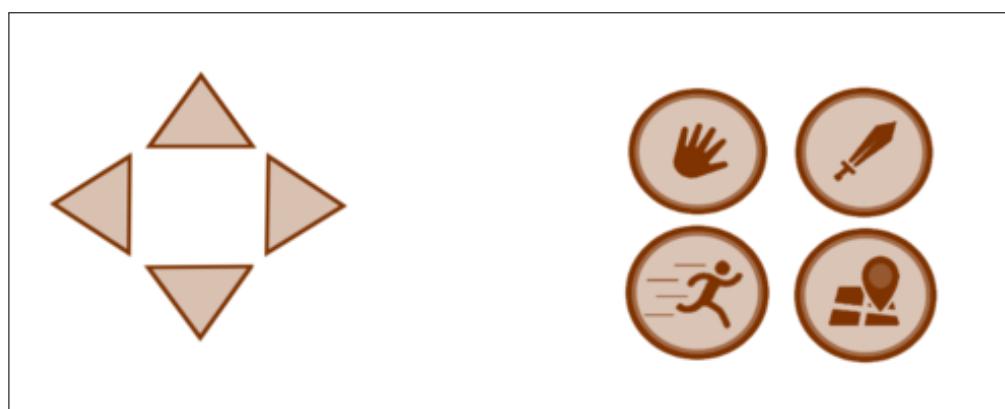
A.4 ENTRADAS

O jogador utilizará os controles de toque. Através dele o jogador poderá escolher as opções de tela inicial, como Jogar, Personagem, opções e até mesmo jogar.

A.4.1 Controle

Com os comandos de toque é possível movimentar o personagem em seu percurso e também executar ações, conforme ilustrado na figura 34.

Figura 34 – Controles de movimentação e ação do personagem.



Fonte: Elaborado pelo autor

- O botão move o personagem para cima.
- O botão move o personagem para baixo.
- O botão move o personagem para a esquerda.
- O botão move o personagem para a direita.
- O botão com ícone pega objetos no chão.
- O botão com ícone lança objetos.
- O botão com ícone faz com que todo o cenário fique visível.

- O botão com ícone  faz com que o personagem corra.

A.5 UNIVERSO DO JOGO

A.5.1 Cenário

O jogo está dividido em quatro cenários que serão construídos tendo como base o mapa isométrico, conforme ilustra a figura 35.

Figura 35 – Ilustração do Cenário



Fonte: Elaborado pelo autor

Os cenários serão interligados através de portais, que poderão ser utilizados sempre que o personagem concluir uma missão.