



MARLON JONAS DE OLIVEIRA LIMA

**PESQUISA E DESENVOLVIMENTO DA
INTERFACE DE UM JOGO PARA A
PLATAFORMA ANDROID**

LAVRAS – MG

2013

MARLON JONAS DE OLIVEIRA LIMA

**PESQUISA E DESENVOLVIMENTO DA INTERFACE DE UM JOGO
PARA A PLATAFORMA ANDROID**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Dr. Joaquim Quinteiro Uchôa

LAVRAS – MG

2013

MARLON JONAS DE OLIVEIRA LIMA

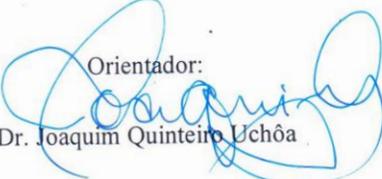
**PESQUISA E DESENVOLVIMENTO DA INTERFACE DE UM JOGO PARA A
PLATAFORMA ANDROID**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 30 de agosto de 2013

Me. André Pimenta Freire UFLA

Me. Bruno de Oliveira Schneider UFLA

Orientador:

Dr. Joaquim Quinteiro Uchôa

LAVRAS – MG

2013

RESUMO

O trabalho apresenta todo o processo de desenvolvimento da interface de um jogo de estratégia baseado em turnos para um dispositivo móvel com o sistema operacional Android, desde a concepção até os testes e otimização do mesmo. Foram desenvolvidos um documento simples de *Game Design* e um projeto de interface detalhado para a produção dos elementos gráficos da interface e implementação do seu código. Ao final de sua implementação foram realizados testes de usabilidade e de experiência de usuário, visando avaliar a qualidade da interface produzida. Como resultado foi produzido o documento de *Game Design* de um jogo com temática inspirada nas tribos americanas. Foram produzidas todas as telas necessárias para a versão final do jogo. Algumas funcionalidades de algumas telas não puderam ser implementadas devido a falta da implementação da mecânica do jogo.

Palavras-chave: Jogos para Dispositivos Móveis, Interface Natural de Usuário; Jogo de estratégia baseado em turnos.

LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do sistema operacional Android, fonte: http://elinux.org/Android_Architecture	12
Figura 2 - Interface diegética de Doom 3 - fonte: http://www.gamegen.com.br/wp-content/uploads/2012/08/DOOM-3-BFG-Edition-01-625x352.jpeg	14
Figura 3 - Grafo de Navegação	19
Figura 4 - Diagrama de Classes.....	23
Figura 5 - Tela Inicial	26
Figura 6 - Tela de Configuração	27
Figura 7 - Tela de Configuração de Partida.....	28
Figura 8 - Tela de Exploração	28
Figura 9 - Tela de Unidades	29
Figura 10 - Tela de Melhorias	30
Figura 11 - Tela de Descrição	30
Figura 12 - Tela de Combate.....	31
Figura 13 - Tela de Configuração	32
Figura 14 - Tela de Créditos.....	32
Figura 15 - Folha de <i>assets</i>	36

LISTA DE SIGLAS

NUI	<i>Natural User Interface</i>
GUI	<i>Graphical User Interface</i>
CLI	<i>Command Line Interface</i>
SDK	<i>Software Development Kit</i>
ADT	<i>Android Development Tools</i>
HUD	<i>Heads-up Display</i>
UI	<i>User Interface</i>

SUMÁRIO

1	INTRODUÇÃO.....	6
1.1	Contextualização	6
1.2	Objetivos	7
1.3	Metodologia	7
1.4	Estrutura do texto	8
2	REVISÃO DE LITERATURA.....	9
2.1	Jogos Eletrônicos.....	9
2.1.1	Desenvolvimento de Jogos	9
2.2	Android.....	10
2.3	Interface de Usuário.....	12
2.3.1	Tipos de Interface para Jogos	13
2.3.2	Interface Natural de Usuário.....	14
3	METODOLOGIA	16
3.1	Material e Métodos	16
3.2	Desenvolvimento.....	17
3.2.1	Pré-produção	17
3.2.2	Produção	20
3.2.3	Pós-produção	24
4	RESULTADOS E DISCUSSÃO	26
4.1	Interface Desenvolvida.....	26
4.2	Discussão.....	33
5	CONCLUSÃO	37
6	REFERÊNCIAS	39
7	APENDICE A.....	40

1 INTRODUÇÃO

1.1 Contextualização

Este trabalho faz parte de um projeto maior onde será desenvolvido um jogo de estratégia baseado em turnos para a plataforma Android.

O gênero de jogo selecionado para o desenvolvimento deste trabalho foi o de estratégia, definido por Novak (2011) como o gênero originado dos jogos clássicos de tabuleiro, como o xadrez, em que jogadores devem administrar um conjunto limitado de recursos para atingir uma meta específica. E como no xadrez, cada jogador possui um turno para fazer suas decisões e realizar a sua jogada. O jogo então é classificado como baseado em turnos.

A temática do jogo desenvolvido foi inspirada nas sociedades tribais americanas. Um tema ainda pouco explorado no desenvolvimento de jogos. Adicionando também um pouco da cultura nacional na temática do jogo, utilizando as tribos brasileiras como personagens do jogo.

A plataforma selecionada para o desenvolvimento do jogo foi a plataforma Android. O Android é um sistema operacional para dispositivos móveis, desenvolvido pela empresa Google. Ele possui código livre, segundo uma licença *Apache*. O desenvolvimento do código é realizado através da linguagem de programação Java, que depois de compilado é interpretado por uma máquina virtual no sistema operacional Android. Devido a sua atual popularidade, a plataforma Android foi selecionada para a implementação do jogo.

Este trabalho aborda a pesquisa e o desenvolvimento da interface do jogo. A interface é composta por todos os elementos visuais e sonoros apresentados ao jogador pelo jogo. Ela é responsável por administrar toda a interação entre o jogador e o jogo. Fornecendo uma resposta visual e sonora para cada ação do jogador com o jogo. O modo de interação primário dos sistemas Android são as telas sensíveis ao toque, portanto o tipo de interface selecionado para o desenvolvimento do jogo é a interface natural.

1.2 Objetivos

O objetivo geral deste trabalho foi o desenvolvimento da interface para um jogo de estratégia baseado em turnos para a plataforma Android. Para isso um conjunto de objetivos específicos foi estabelecido. São eles:

- a) Elaboração de um documento com o projeto da interface, contendo a descrição de cada tela do jogo, com todos os seus elementos e formas possíveis de interação.
- b) Elaboração e produção dos elementos apresentados na tela do jogo.
- c) Implementação do código da interface na linguagem Java seguindo as especificações do projeto da interface.
- d) Realização de testes da interface em um dispositivo móvel, avaliando se a disposição dos elementos do jogo, os meios de interação e as transições de tela estão de acordo com as descritas no documento de projeto de interface

1.3 Metodologia

Este projeto foi desenvolvido de maneira conjunta com os também estudantes da Universidade Federal de Lavras (UFLA) Jeferson Barrile Tomazella e Renan Faraco Teixeira. Cada membro do projeto foi responsável pelo desenvolvimento de pelo menos um módulo do jogo. O jogo possui seis módulos: interface, inteligência artificial, desenvolvimento de níveis, arte, música e programação. Os módulos de arte, música e programação foram desenvolvidos de forma colaborativa entre todos os membros do projeto. Este trabalho irá abordar apenas a metodologia do desenvolvimento da interface do jogo e é a primeira parte do projeto a ser concluída.

O desenvolvimento da interface foi realizado em três etapas. Na primeira etapa o projeto da interface foi elaborado, descrevendo todas as telas do jogo e como o usuário pode interagir com cada uma delas. E na segunda etapa foi realizada a implementação da interface. Para isso foi utilizada a linguagem de programação Java, no ambiente de desenvolvimento Eclipse, utilizando a SDK (*Software Development Kit*) do Android. E todos os elementos gráficos foram produzidos no editor de imagens Adobe Photoshop

CC. E na terceira etapa foram realizados os testes, para isto foram utilizados a maquina virtual disponibilizada pelo SDK do Android e os dispositivos móveis Samsung GT-I9505 e Asus Eee Pad Transformer TF101.

1.4 Estrutura do texto

Este trabalho está estruturado da seguinte maneira:

- O Capítulo 2 apresenta a Revisão de Literatura, mostrando os principais conceitos utilizados no trabalho.
- O Capítulo 3 mostra a Metodologia do trabalho, detalhando a sequência de atividades, a maneira de condução do trabalho e o modo como os conceitos foram aplicados.
- O Capítulo 4 contem os Resultados deste trabalho, mostrando quais foram os resultados finais.
- O Capítulo 5 finaliza o trabalho apresentando as conclusões e propondo trabalhos futuros.

2 REVISÃO DE LITERATURA

2.1 Jogos Eletrônicos

Os jogos eletrônicos têm sua origem nos anos 50, quando pesquisadores e estudantes buscando relaxar de suas tarefas tradicionais de programação de complexas funções matemáticas começaram programar jogos para executar nos grandes *mainframes* das universidades e institutos de pesquisa. No ano de 1961 foi desenvolvido o jogo interativo para computador “*Spacewar!*”, pelo estudante do MIT (Instituto de Tecnologia de Massachusetts), Steve Russel. Posteriormente adaptado para o fliperama como “*Space Wars*” (NOVAK, 2011).

No início, o acesso aos jogos eletrônicos era restrito a pesquisadores e estudantes das universidades, que possuíam computadores aptos a executar os jogos. Porém, com o advento dos fliperamas, os jogos eletrônicos se tornaram acessíveis ao público geral. Com a grande popularidade conquistada pelos jogos eletrônicos, um novo ramo na indústria foi estabelecido, dedicado exclusivamente para a produção dos mesmos.

Atualmente a indústria de jogos é dividida em três ramos distintos de produção, são eles, computador, consoles e portáteis. Cada um dos ramos já possui seus gêneros usuais de jogos já estabelecidos, devido a limitações dos dispositivos de entrada ou até mesmo por motivos culturais. Jogos de ação, aventura e lutas são encontrados com mais frequência em jogos de console que nos outros dois ramos. Já os jogos de RPG (*Role-Playing Game*), FPS (*First-Person Shooter*) e estratégia são gêneros tradicionais do computador. Por final, os jogos casuais e do tipo quebra-cabeças são mais comuns nos portáteis, devido a curta duração de suas partidas e a facilidade de seus controles.

2.1.1 Desenvolvimento de Jogos

O desenvolvimento de jogos eletrônicos é similar ao desenvolvimento de *software*. Porém os jogos são diferentes quanto à sua finalidade: enquanto um *software* é desenvolvido como uma ferramenta para a execução de uma tarefa específica, um jogo é

desenvolvido como uma ferramenta de entretenimento. Portanto, um jogo, além de ser funcional e eficiente como um *software*, deve ser divertido.

Jesse Schell (2011) descreve a estrutura de um jogo através de uma tétrede elementar: mecânica, narrativa, estética e tecnologia. A mecânica é definida pelas regras do jogo, ou seja, o que se pode ou não fazer, como se vence ou perde. A narrativa é definida pela história do jogo, com seus personagens e sua ambientação. A estética é definida pela arte do jogo, como os seus elementos serão apresentados ao jogador, se possui gráficos “cartunizados” ou realistas. A tecnologia é definida pela parte técnica do jogo, como dispositivos de entrada e os recursos de *hardware* necessários para executar o jogo. Utilizando os conceitos da tétrede elementar apresentado por Schell é criado um documento com o *Game Design* do jogo, onde são descritas todas as suas informações.

Após a etapa inicial de desenvolvimento o jogo entra em sua fase *alfa*. “A fase *alfa* de desenvolvimento é o ponto em que um game pode ser jogado do começo ao fim” (NOVAK, 2011). O jogo não precisa estar completo nesta fase. E pode ser publicado para teste para um grupo seletivo de jogadores. Esta prática é adotada para o aperfeiçoamento do Game Design, onde podem ocorrer alterações na mecânica. “Depois de passar pela fase *alfa* o jogo entra na fase *beta*. Nesta fase, a ênfase está na correção de problemas” (NOVAK, 2011). O jogo também é publicado em sua fase *beta*, porém para um grupo ligeiramente maior que o da fase *alfa*. Nesta fase o mesmo já está completo, com todo o seu conteúdo implementado.

O lançamento do jogo é conhecido como a fase Ouro. Após ela se inicia a pós-produção, na qual são produzidas correções para os problemas encontrados após o lançamento e novos conteúdos são produzidos para incrementar o jogo.

2.2 Android

O Android¹ é um sistema operacional de código aberto desenvolvido pela Google Inc. para dispositivos móveis. Ele foi desenvolvido tendo como base o *kernel* 2.6 do Linux.

¹ <http://www.android.com/>

Desde o seu lançamento no ano de 2008 ele já recebeu oito novas versões, todas com o nome de sobremesas.

Devido ao grande número de versões do Android, o desenvolvimento para a plataforma sofre de um problema chamado de fragmentação. Aplicativos desenvolvidos para versões mais recentes do Android não possuem retro compatibilidade com versões mais antigas. Este problema faz com que uma versão mínima do sistema operacional seja selecionada no momento de desenvolvimento da aplicação. “No ano de 2011, muitos proeminentes desenvolvedores de dispositivos Android concordaram em oferecer suporte para a última versão do sistema operacional Android para um dispositivo por um período de 18 meses. [...] oferecendo uma grande ajuda na redução da fragmentação” (ZECHNER, 2011).

A arquitetura do Android é composta por vários grupos de componentes organizados em forma de camadas, onde cada camada é construída com elementos da camada inferior. Ela pode ser separada em cinco camadas distintas: *Kernel* do Linux, *Android Runtime*, Bibliotecas, *Framework* de aplicação e Aplicações. O *Kernel* do Linux é responsável pelas funções básicas do sistema operacional, como o gerenciamento de processos, memória e rede. Ele também fornece os *drivers* para o *hardware* do dispositivo. A camada *Android Runtime* é responsável pela criação e execução das aplicações do sistema. A camada de Bibliotecas possui um conjunto básico de bibliotecas do sistema operacional, responsáveis por tarefas como renderização do vídeo, execução de áudio e escrita na tela. A camada de *Framework* de aplicação reúne as bibliotecas da camada de Bibliotecas e fornece ferramentas para o desenvolvimento das aplicações, como administração de periféricos ou de serviços em segundo plano. E a camada de Aplicações reúne o conjunto de aplicativos disponíveis para a utilização do usuário do sistema. A Figura 1 apresenta a arquitetura do sistema operacional Android.

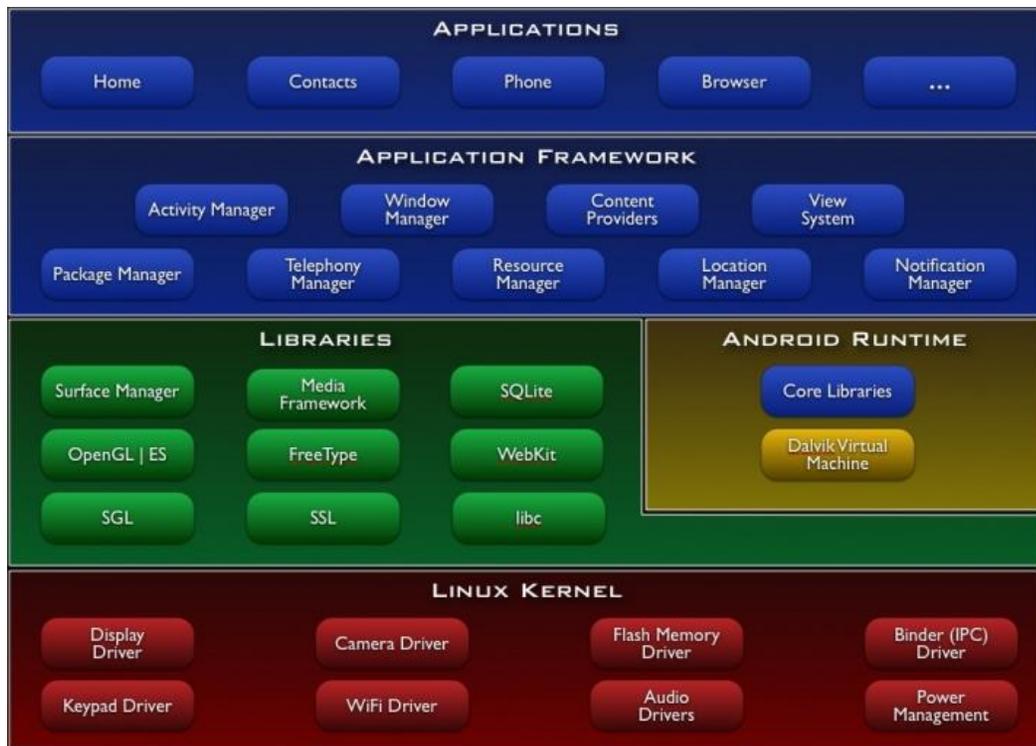


Figura 1 - Arquitetura do sistema operacional Android, fonte: http://linux.org/Android_Architecture

2.3 Interface de Usuário

A interface de usuário do jogo é um elemento chave para o entretenimento do jogador. Ela é o elemento mais próximo do jogador, é a partir dela que ele interage com o jogo. “[...] interfaces de usuário eficazes, para os profissionais, significa que os médicos podem fazer diagnósticos mais precisos, pilotos podem voar aviões com uma maior segurança, crianças aprendem de forma mais eficaz e artistas gráficos podem explorar possibilidades criativas de forma mais fluida” (SCHNEIDERMAN, 2005). No caso dos jogos, ela permite uma experiência mais divertida ao jogador.

Diferentemente das interfaces de usuário de *software* e páginas da *internet*, as interfaces de usuário dos jogos devem levar o usuário a um estado de imersão. “O

jogador deve ser capaz de concentrar-se exclusivamente na experiência do jogo, sem distrações externas (ambientais) ou internas (mentais)” (NOVAK, 2011). Ao mesmo tempo deve auxiliar o jogador a executar as tarefas dentro do jogo, sem interferir em sua experiência. “O objetivo de uma interface é fazer os jogadores sentirem que tem o controle de suas experiências” (SCHELL, 2011).

2.3.1 Tipos de Interface para Jogos

A interface pode ser separada em duas categorias distintas: física e virtual. A interface física está vinculada aos dispositivos que o jogador manipula enquanto joga. Ela pode ser separada em interface física de entrada e de saída. Os elementos de entrada são os controles de um console, o *mouse* e teclado de um computador ou a tela sensível ao toque de um dispositivo móvel. Já os elementos de saída são o monitor do console ou do computador, o visor do dispositivo móvel, os alto-falantes e os motores de vibração dos controles de um console e de um dispositivo móvel. A interface virtual é representada pela camada entre a interface física e o jogo propriamente dito.

A interface virtual pode ser dividida em duas categorias: ativa e passiva. A interface virtual ativa permite a interação do jogador com a interface, normalmente através de botões e menus. Ela permite que o jogador configure o jogo, salve sua partida ou personalize o seu personagem. “Outra interface ativa comum é o sistema de ação, que contém escolhas do jogador relacionadas ao modo de jogar. [...] As opções disponíveis nessa interface incluem palavras de ação, permitindo que o jogador entre em combate, explore, comunique-se ou acesse outros elementos relacionados ao modo de jogar.” (NOVAK, 2011). Já a interface virtual passiva é responsável pela exibição de informações importantes a respeito do jogo para o jogador, como pontuação, tempo da partida ou o número de balas restantes em sua arma. Normalmente ela é apresentada em forma de HUD (*Heads Up Display*), “uma tela transparente que apresenta informação ao jogador sem que seja necessário sair de seu ponto de vista.” (FAGERHOLT, 2009). Existem modelos de interface em que a camada virtual passiva é tão fina que se torna praticamente inexistente. Neste modelo as informações são apresentadas no próprio jogo, como elementos do mundo virtual. Este tipo de interface é chamada de interface

diegética. Ela é definida como uma “interface que está inclusa no mundo do jogo, ela pode ser vista e escutada pelo personagem do jogo.” (ANDREWS, 2010). O jogo Doom 3 lançado pela id Software² no ano de 2007 apresenta um tipo de interface diegética na qual o numero de balas da arma é apresentado na própria arma, como apresentado na Figura 2.



Figura 2 - Interface diegética de Doom 3 - fonte: <http://www.gamegen.com.br/wp-content/uploads/2012/08/DOOM-3-BFG-Edition-01-625x352.jpeg>

2.3.2 Interface Natural de Usuário

A Interface Natural de Usuário ou NUI (*Natural User Interface*) é o tipo de interface utilizado em sistemas com suporte a entrada de dados via toque, gestos ou voz. O termo “natural” refere-se à forma em que o usuário interage com a interface e não a origem da interface. Mark Weiser (1991) chama este tipo de tecnologia Computação Ubíqua. Uma tecnologia na qual uma pessoa pode comandar uma máquina da mesma forma que se comunica com outra pessoa.

O desenvolvimento de uma interface para um sistema que tem como método de entrada uma tela sensível ao toque é diferente de um sistema que tem como entrada

² <http://www.idsoftware.com/games/doom/doom3>

um teclado e um mouse. Um novo paradigma de interação é necessário, pois as possibilidades e limitações de cada método de entrada são diferentes. Para entender o porquê das interfaces serem diferentes, é necessário entender como os métodos de entrada diferem entre si.

Um computador tradicional possui um monitor como dispositivo de saída e o conjunto de mouse e teclado como dispositivos de entrada. Toda a comunicação entre o usuário e o computador é realizada através desses dispositivos. Uma Interface Gráfica de Usuário (GUI – *Graphical User Interface*) do tipo WIMP (*Windows Icons Menus Pointer*) é utilizada para exibir as informações no monitor. Este tipo de interface utiliza a metáfora de uma área de trabalho para a exibição da informação na tela, associando elementos de mundo real a elementos virtuais. Para selecionar um ícone apresentado pela interface basta usuário mover o *mouse* para deslocar um cursor virtual pelo monitor até o ícone desejado e pressionar o botão do mouse para selecioná-lo.

Por sua vez, um sistema com uma tela sensível ao toque possui o mesmo dispositivo para realizar a função de saída e entrada. Para executar a mesma tarefa de selecionar um ícone apresentado na tela basta o usuário tocar o ícone apresentado na tela do dispositivo. Aparentemente um modelo muito mais simples de interação. Porém esta simplicidade vem ao custo da funcionalidade exercida pelo ponteiro do *mouse*. Consequentemente, todas as metáforas que utilizam o posicionamento do cursor presentes nas WIMP GUI não podem ser utilizadas em sistemas com telas sensíveis ao toque. E diferentemente de sistemas com *mouse* os sistemas de tela sensível ao toque são capazes de registrar mais de um toque ou clique em pontos distintos da tela simultaneamente.

As GUIs são desenvolvidas tendo como base um *mouse* como dispositivo de entrada, portanto suas metáforas não podem ser aplicadas de forma eficiente em sistemas com uma tela sensível ao toque como entrada. Com a popularização das telas sensíveis ao toque surgiu a necessidade da criação de um tipo de interface específico para os sistemas com telas sensíveis ao toque. A NUI é desenvolvida tendo como base as telas sensíveis ao toque.

3 METODOLOGIA

Este trabalho, como citado na Introdução, faz parte de um projeto maior com o objetivo de produzir um jogo de estratégia baseado em turnos para a plataforma Android. A produção da interface aconteceu de forma isolada dos outros elementos do jogo, portanto nenhuma mecânica do jogo foi implementada neste trabalho.

3.1 Material e Métodos

Para a execução deste trabalho foram utilizados os aplicativos Adobe Photoshop³, para a produção dos elementos apresentados na tela do jogo, e o ambiente de desenvolvimento Eclipse⁴ para a implementação do código da interface.

A versão do editor de imagens utilizado foi o Adobe Photoshop CC, disponibilizado como uma versão de testes por um período de 30 dias no site do desenvolvedor⁵, também disponibilizado da forma de assinatura mensal. Porém o aplicativo Adobe Photoshop pode ser substituído pelo editor de imagens GIMP, disponibilizado de forma gratuita no site do desenvolvedor⁶.

A versão utilizada do ambiente de desenvolvimento foi o Eclipse Standard 4.3, disponibilizado de forma gratuita no site de seu desenvolvedor⁷. Para a utilização do Eclipse para o desenvolvimento de aplicações Android é necessária a instalação do SDK do Android e do *plugin* do Eclipse ADT, ambos fornecidos de forma gratuita no site do desenvolvedor⁸. Existe também um pacote contendo o Eclipse com o ADT instalado e o SDK do Android juntos, também disponibilizado no mesmo site que o ADT e o SDK são disponibilizados⁹.

³ <http://www.adobe.com/br/products/photoshop.html>

⁴ <http://www.eclipse.org/>

⁵ <https://creative.adobe.com/>

⁶ <http://www.gimp.org/>

⁷ <http://www.eclipse.org/>

⁸ <http://developer.android.com/>

⁹ <http://developer.android.com/>

3.2 Desenvolvimento

O desenvolvimento deste trabalho foi realizado em três etapas. Na etapa de pré-produção foi desenvolvida a parte intelectual e de planejamento do trabalho, com a escrita do *Game Design* e do projeto da interface. Na etapa de produção foi realizado a maior parte do trabalho técnico, com a produção dos elementos visuais utilizados no jogo e a implementação do código da interface. E na etapa de pós-produção foi realizado o trabalho de revisão e otimização do código e dos elementos da interface.

3.2.1 Pré-produção

A primeira tarefa a ser realizada na produção de um novo jogo é a criação do documento de *Game Design*. Como descrito no referencial teórico existem vários aspectos do jogo que devem ser documentados antes mesmo do início da produção de protótipos e artes. Como este trabalho trata apenas da produção da interface, não foi necessário um documento de *Game Design* muito complexo. O modelo de Jesse Schell da téttrade do Game Design, descrito no referencial teórico, foi utilizado para a produção do documento.

A Tecnologia selecionada para a produção do jogo foi a Android, portanto ele foi produzido para dispositivos móveis equipados com telas sensíveis a toque. Para a mecânica o jogo *Heroes of Might and Magic III*¹⁰ produzido pela companhia 3DO no ano de 1999 foi utilizado como base. Foi criado um conjunto de regras mais simples para facilitar a jogabilidade em um dispositivo móvel e reduzir o processamento necessário ser executado de forma fluida em um dispositivo com poder de processamento e memória limitado.

No jogo *Heroes of Might and Magic III* o jogador controla uma das raças do jogo e é lhe dado o objetivo de eliminar as outras raças. Ele tem o controle de um castelo e tem vários heróis e um exército sob seu controle. No castelo o jogador pode administrar suas construções e unidades. O jogador deve guiar seus heróis acompanhados de seu exercito através de um mapa e eliminar os outros jogadores.

¹⁰ <http://www.heroesofmightandmagic.com/heroes3/>

Quando dois heróis se encontram no mapa eles batalham e apenas um é vitorioso. Acontece o mesmo quando um herói encontra um castelo inimigo. O primeiro jogador a conseguir eliminar todos os outros jogadores do mapa ganha a partida. Para a simplificação da mecânica, foi reduzida a quantidade de raças, unidades e construções disponíveis no jogo. O sistema de múltiplos heróis foi removido, assim como a utilização de mapas durante as batalhas.

A narrativa foi reduzida para apenas cenário e personagens, pois o jogo não apresenta um modo de história, apenas o modo escaramuça¹¹. O cenário selecionado foi o continente americano. E os personagens são os índios americanos, que a mérito de simplificação foram separados em três grupos: norte, centro e sul, representados respectivamente pelas tribos norte-americanas, mesoamericanas e sul-americanas.

A estética selecionada para a produção do jogo foi a cartum de duas dimensões, devido a facilidade na produção dos materiais gráficos e a na implementação do código para elementos em duas dimensões.

Para a produção do documento foram realizados os seguintes procedimentos: análise da mecânica do jogo, análise da narrativa, pesquisa de imagens para representação dos elementos do jogo, produção de protótipos e escrita do documento.

Analisando a mecânica simplificada do jogo podemos perceber que o jogo precisa de no mínimo três telas distintas: uma para a exploração do mapa, uma para administrar as construções e unidades, e outra para as batalhas. Porém, como o jogo foi sendo desenvolvido para um dispositivo móvel, em que a maioria das telas têm um tamanho reduzido, não é recomendado colocar muitas informações em uma única tela. Portanto a tela para administrar as construções e unidades foi separada em três telas distintas: uma para as unidades, uma para as construções e outra tela para descrever o que cada construção faz. Além disso, em geral, todo jogo precisa de uma tela de

¹¹ Modo de jogo comum aos jogos de estratégia, nele o jogador enfrenta outros jogadores controlados pelo computador. Este modo é caracterizado por não apresentar elementos da história do jogo.

abertura, uma tela de configurações e uma tela de créditos. Analisando a narrativa podemos perceber que existe mais de uma opção de tribo para o jogador escolher. Portanto foi necessária uma nova tela para o jogador configurar a sua partida. Com isso, são necessárias nove telas distintas no jogo, permitindo a criação do grafo de navegação entre as telas. O grafo de navegação é apresentado na Figura 3.

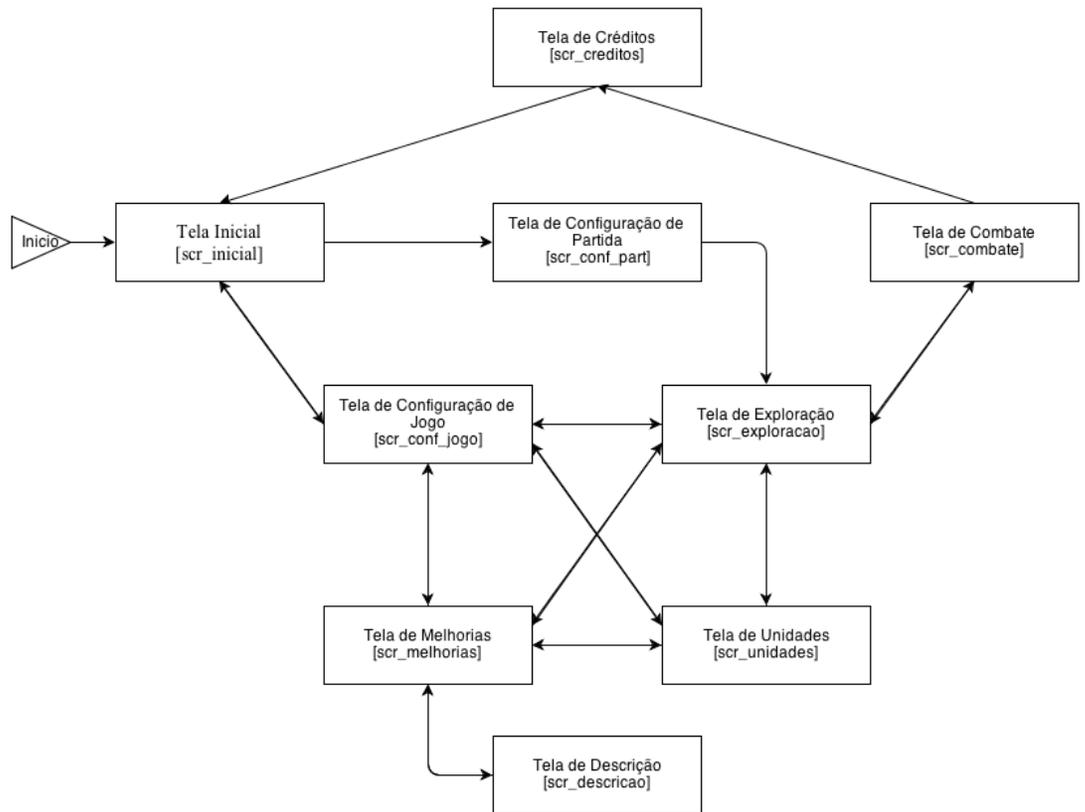


Figura 3 - Grafo de Navegação

Analisando a narrativa do jogo, é possível construir uma relação de elementos que necessitam de uma representação visual no jogo. Assim é estruturada uma lista de personagens, cenários, construções e recursos, presentes no jogo. A partir desta lista são

feitas pesquisas por imagens que representem estes elementos, para que uma descrição possa ser escrita no projeto e posteriormente reproduzida em forma de um desenho.

Nas etapas iniciais, os protótipos foram feitos em papel utilizando apenas representações do tamanho e posição dos elementos na tela. Já nas etapas finais os protótipos foram realizados no editor de imagens, utilizando as imagens finais que seriam utilizadas na interface. Nas etapas finais a escrita do projeto da interface foi realizada em paralelo com a criação e produção dos *assets* e da programação da interface. A escrita do projeto de uma tela era realizada assim que um protótipo apresentava uma estética agradável e um nível de usabilidade mínimo. Ambas as avaliações foram realizadas de forma empírica.

3.2.2 Produção

A etapa de produção foi separada em duas partes. Na primeira os elementos da tela foram produzidos e na segunda foi realizada a implementação do código da interface.

Para a criação e produção dos elementos da tela um arquivo no formato PSD¹², formato padrão do editor de imagens Adobe Photoshop, com as dimensões da maior resolução suportada pelo jogo foi criado. Este formato foi selecionado por suportar edição em várias camadas e ter suporte ao canal alfa, que possibilita imagens com transparência. Utilizando o projeto da interface como referência, todos os elementos necessários na tela do jogo foram desenhados, respeitando tamanhos e descrições do projeto. Cada elemento foi transformado em um objeto inteligente, o que transforma as camadas selecionadas em um novo arquivo de imagem no formato PSB¹³, formato padrão para objetos inteligentes do editor de imagens Adobe Photoshop. Esta ação auxiliou no processo de gerar as imagens utilizadas no código da interface e permitiu a replicação para criar elementos semelhantes. Pois cada objeto inteligente pode ser editado como uma nova imagem, permitindo que cada elemento da tela seja editado de

¹² Arquivo do tipo Adobe Photoshop Document

¹³ Arquivo do tipo Adobe Photoshop Large Document

forma isolada. Imagens com uma camada alfa foram salvas no formato PNG¹⁴ e imagens que não necessitavam da camada alfa foram salvas no formato JPG¹⁵. Estes formatos de imagem foram selecionados por possuírem suporte nativo do sistema operacional Android. Para cada elemento isolado da tela foi gerado uma nova imagem. Elas foram salvas em uma única resolução e organizados em um sistema de pastas que os separavam segundo tela, tipo e tribo. Sendo os tipos: imagem, botão e texto.

Existem duas formas de armazenar arquivos em uma aplicação Android. Utilizando a pasta “/res” ou a pasta “/assets”. A grande diferença entre as duas está na declaração dos elementos a serem utilizados pela aplicação. Na pasta “/res” é necessário declarar todos os elementos que podem ser exibidos na aplicação, enquanto na pasta “/assets” eles podem ser declarados conforme a necessidade de uso. Portanto todos os elementos visuais e sonoros foram armazenados como *assets*. Visando a flexibilidade fornecida pela declaração por demanda.

A produção dos *assets* aconteceu em duas etapas distintas do trabalho, na produção e na pós-produção. A produção dos *assets* na etapa de pós-produção será discutida no tópico de Resultados e Discussão. Na etapa de produção, os desenhos foram produzidos seguindo as descrições presentes no projeto da interface. Os desenhos pertinentes ao mundo do jogo foram produzidos por outro membro do projeto e foram devidamente creditados na tela de créditos do jogo.

Para a implementação do código foi utilizado o *framework* apresentado por Zechner (2011) no livro “*Beginning Android 4 Games Development*”. Ele pode ser encontrado para *download* no site da editora¹⁶. O *framework* possui um conjunto de funções básicas para o funcionamento do jogo, como: carregamento e descarregamento de imagens e sons na memória, desenho no *canvas*, execução de áudio e atualização da tela do jogo. Ele utiliza dois pacotes, um para a sua interface e outro para a sua

¹⁴ Arquivo do tipo Portable Network Graphics

¹⁵ Arquivo do tipo Joint Photographic Experts Group

¹⁶ <http://www.apress.com/9781430239871>

implementação. Além dos pacotes do framework foi criado um pacote para as classes do jogo. As principais classes do jogo são apresentadas no diagrama da Figura 4.

No total foram criadas quinze classes no pacote do jogo. Nove delas representam as telas do jogo, enquanto as outras realizam as tarefas de declaração de *assets*, carregamento e descarregamento de *assets* na memória, e armazenamento dos dados do jogo. Todas as classes de tela de jogo possuem um conjunto de variáveis responsáveis pelo armazenamento dos valores das coordenadas e tamanhos dos *assets* utilizados naquela tela. Esta prática foi adotada para centralizar os dados referentes às coordenadas dos assets e evitar múltiplos valores de coordenadas para um mesmo *asset*. No caso da classe “TelaDescricao” existe um conjunto de variáveis responsáveis pelo armazenamento dos valores das coordenadas do mapeamento dos *assets* da tela, visto que todos foram armazenados em uma única imagem.

Além das variáveis com as coordenadas, as classes de tela de jogo possuem dois procedimentos semelhantes em todas elas. Eles são responsáveis pela atualização do jogo (*update*) e pelo desenho dos *assets* no *canvas* (*present*). No procedimento *update* são registrados os eventos de toque, portanto toda a implementação dos botões e dos elementos ativos da interface é realizada neste procedimento. A cada atualização do jogo o procedimento *present* redesenha todos os *assets* na tela. Todas as animações e movimentos da tela do jogo são realizadas modificando os valores das coordenadas e dos mapeamentos dos *assets* desenhados pelo procedimento *present*.

Como a atualização da tela do jogo acontece muito rapidamente foi criada uma variável para reduzir a velocidade em que as imagens da tela são. Nela é estabelecido um valor no qual a mudança de imagens ocorre, assim a cada ciclo do jogo o valor da variável a crescido em um. Quando o valor determinado é alcançado, as imagens são alteradas e a variável é zerada para a repetição do processo. Esta prática evita que uma animação ou um movimento aconteça muito rápido na tela, diminuindo a taxa de atualização das coordenadas e mapeamentos dos *assets*.

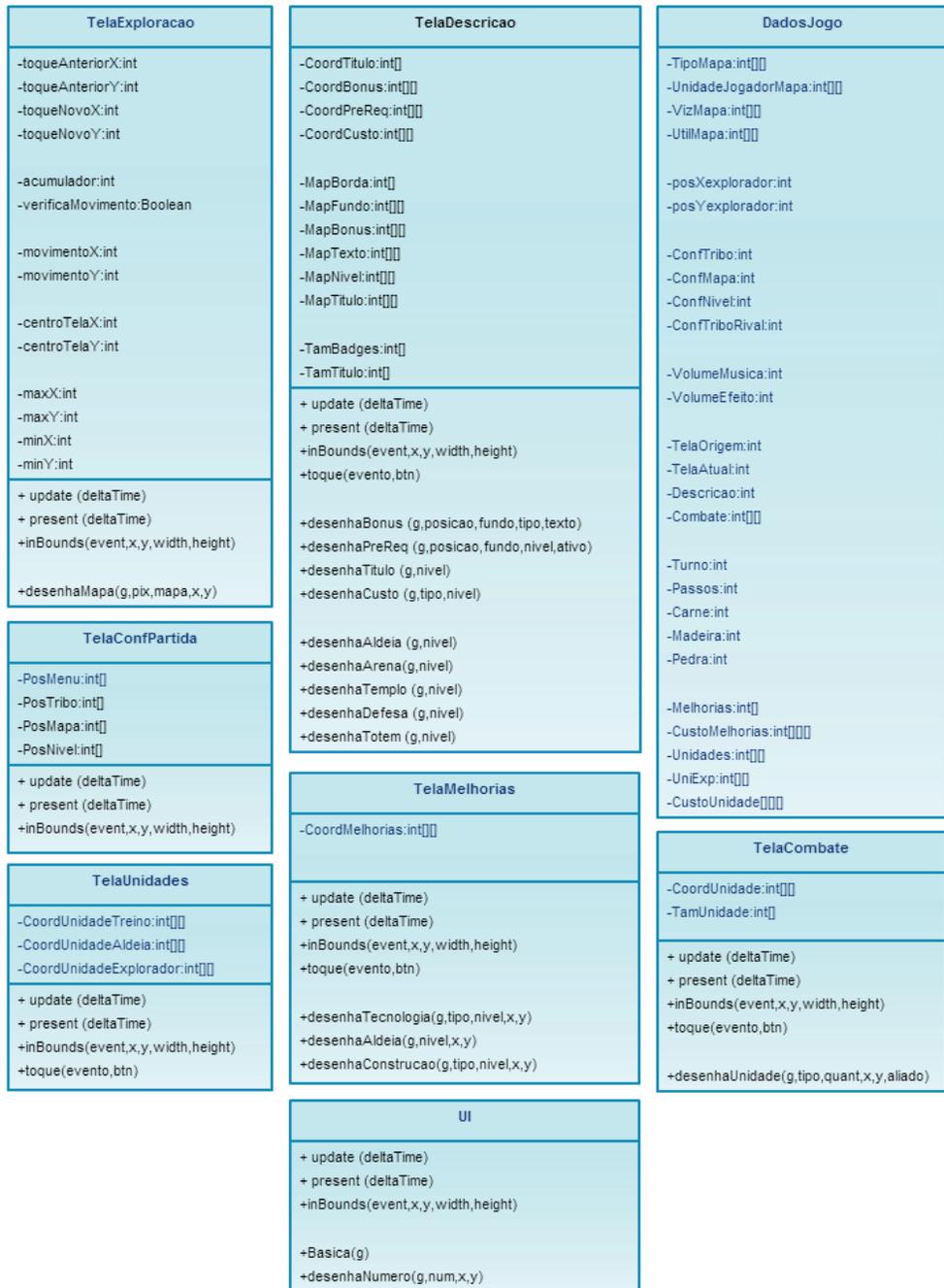


Figura 4 - Diagrama de Classes

Uma classe chamada “UI” foi criada para abrigar os procedimentos responsáveis pelo desenho dos elementos comuns a mais de uma tela. Nela estão presentes os procedimentos de desenho da interface básica do jogo e do desenho de números na tela. Já os procedimentos de desenho dos elementos específicos de cada tela estão localizados em suas próprias classes. Eles foram criados devido a grande quantidade de elementos semelhantes nas telas, assim um único procedimento poderia ser utilizado para desenhar mais de um elemento da tela. O procedimento mais complexo criado está presente na classe “TelaExploracao”, chamado “desenhaMapa”. Ele é responsável pelo desenho do mapa do jogo no *canvas*. Como o jogador pode navegar pelo mapa arrastando o dedo em qualquer direção, o mapa deve ser redesenhado a cada pixel que o jogador move. Portanto o procedimento desenha o mapa na tela possuindo como referencia as coordenadas do centro da tela. Estas coordenadas não podem ser menores que metade da resolução da tela ou maiores que o tamanho do mapa menos metade da resolução da tela. Isso impossibilita o jogador de ultrapassar os limites do mapa. Por padrão, as coordenadas iniciais são as mesmas do explorador do jogador.

A classe “DadosJogo” foi criada com o propósito de armazenar as informações do jogo. Nela estão centralizadas informações como pontuação, configurações e valores dos recursos de cada jogador. Ela também é utilizada para armazenar os dados do mapa jogo, como tipo, tamanho e posição dos *assets* a serem desenhados na tela.

3.2.3 Pós-produção

Na pós-produção foram realizados testes do jogo. Ele era executado na maquina virtual disponibilizada pelo SDK do Android e nos dispositivos móveis. Os testes de usabilidade e experiência de usuário foram realizados de forma empírica. Nele um grupo de dez pessoas foi questionado quanto ao modo de uso do jogo. Avaliando como bom, médio ou ruim. Através do teste de usabilidade, foi avaliado se a interface permitia que o usuário realizasse as operações disponíveis no jogo de maneira eficaz. Nos testes de experiência o mesmo grupo era questionado quanto a aparência do jogo, avaliando a qualidade das imagens, a ergonomia para a execução das ações do jogo e a experiência durante a utilização do jogo. Quando alguma falha era encontrada em qualquer um

desses dois elementos o projeto da interface era adaptado para corrigir o problema e o código reescrito.

4 RESULTADOS E DISCUSSÃO

4.1 Interface Desenvolvida

Na tela inicial, apresentada na Figura 5, são apresentados ao jogador o título do jogo são fornecidas as opções que iniciar o jogo tocando em jogar ou de ir para a tela de configurações tocando em configurar.

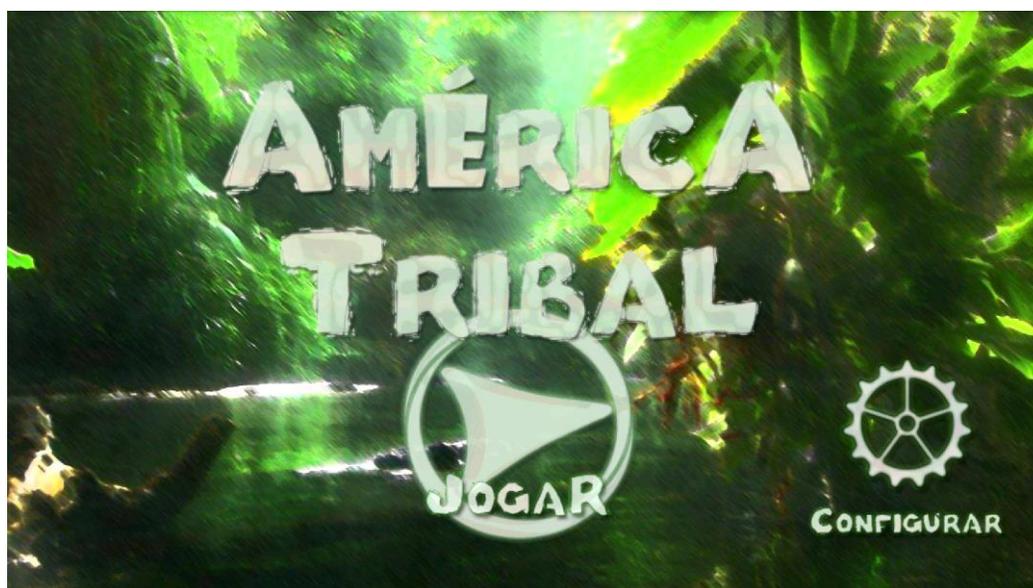


Figura 5 - Tela Inicial

Na tela de configuração, apresentada na Figura 6, o jogador pode alterar o valor do volume da musica de fundo do jogo e dos seus efeitos sonoros.

Na tela de configuração de partida, apresentada na Figura 7, o jogador realiza a configuração da partida a ser iniciada. São apresentados as tribos, mapas e níveis disponíveis para escolha do jogador. O jogador pode iniciar a partida tocando em iniciar ou retornar a tela inicial tocando na seta indicando retornar.

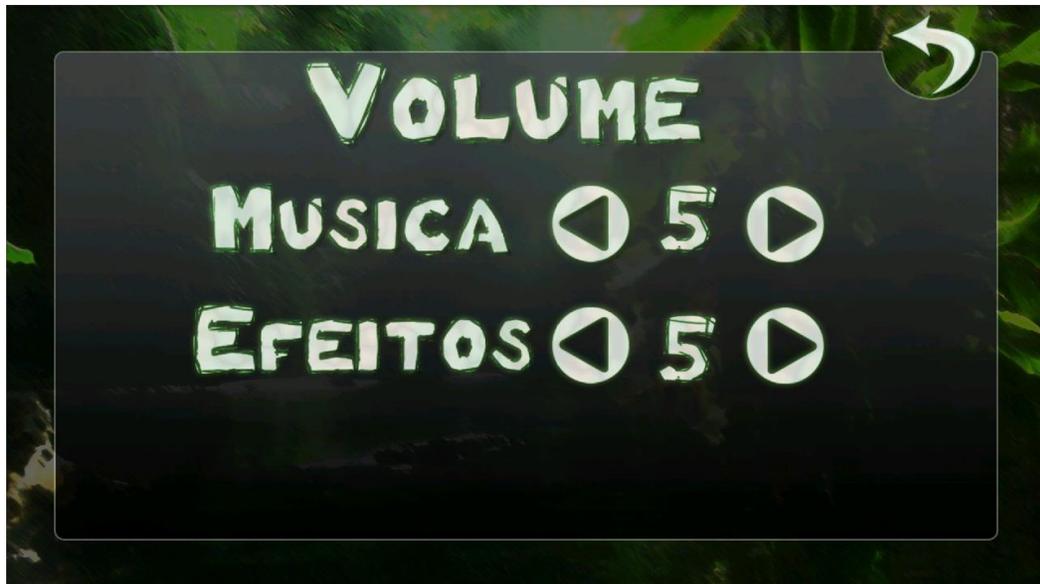


Figura 6 - Tela de Configuração

A tela de exploração, apresentada na Figura 8, é a tela em que o jogador pode explorar o mapa do jogo, encontrar inimigos, recursos e construções para interagir. Ele pode navegar pelo mapa arrastando o dedo pela tela arrastando o mapa na direção do seu movimento, e seleciona um quadrado específico apenas tocando nele. Tocando em uma unidade inimiga é realizada a transição para a tela de combate.

Na tela de unidades, apresentada na Figura 9, o jogador pode recrutar novas unidades e organizar as unidades presentes na aldeia e as que acompanham o explorador. Elas foram organizadas em três linhas, na superior são apresentadas as unidades disponíveis para o recrutamento. Na linha do meio as unidades presentes na aldeia. Já na linha inferior estão as unidades que acompanham o explorador. Quando as unidades para recrutamento são tocadas os recursos necessários para o seu recrutamento são consumidos e a unidade é adicionada às unidades da aldeia. Quando as unidades da aldeia são tocadas elas são transferidas para o primeiro espaço vago das unidades que

acompanham o explorador. Por fim, quando as unidades do explorador são tocadas elas são transferidas para as unidades da aldeia.



Figura 7 - Tela de Configuração de Partida



Figura 8 - Tela de Exploração



Figura 9 - Tela de Unidades

Na tela de melhorias, apresentada na Figura 10, o jogador pode administrar as suas construções e tecnologias. Os níveis de cada melhoria são apresentados na forma de indicadores luminosos no canto inferior direito de cada uma delas. Quando o jogador toca em alguma das melhorias é realizada uma transição para a tela de descrição daquela melhoria específica.

O jogador pode alternar entre as telas de exploração, melhorias e unidades através dos botões localizados nos cantos inferiores da tela. Respectivamente representados pelas silhuetas do explorador e da aldeia da tribo selecionada e pela silhueta de dois machados cruzados. No canto superior esquerdo é apresentado o mostrador de recursos e na borda inferior é apresentado o contador de passos, exibindo a quantidade de passos restantes no turno.



Figura 10 - Tela de Melhorias



Figura 11 - Tela de Descrição

Na tela de descrição, apresentadas na Figura 11 são exibidos os bônus, pré-requisitos e custo de uma melhoria específica. O jogador pode tocar no botão de subir o nível para comprar a melhoria ou pode tocar no botão de retornar para voltar à tela de melhorias.



Figura 12 - Tela de Combate

Na tela de combate, apresentada na Figura 12, é realizado o confronto entre as unidades do jogador e as unidades inimigas. Do lado direito são apresentadas as unidades do jogador, e do lado esquerdo as unidades inimigas. Cada lado abriga no máximo três unidades. A quantidade de unidades é apresentada na parte inferior das unidades. O jogador pode atribuir os ataques manualmente o tocar na imagem do cérebro para que essa atribuição seja feita de modo automático. Para acelerar o processo do combate o jogador pode tocar na imagem do relógio.

As principais telas possuem uma imagem de uma engrenagem no canto superior direito da tela. Tocando nela acontece uma transição para a tela de configuração. Porém se a tela de origem for alguma das telas de jogo o botão de retorno a tela inicial é exibido, como é apresentado na Figura 13.

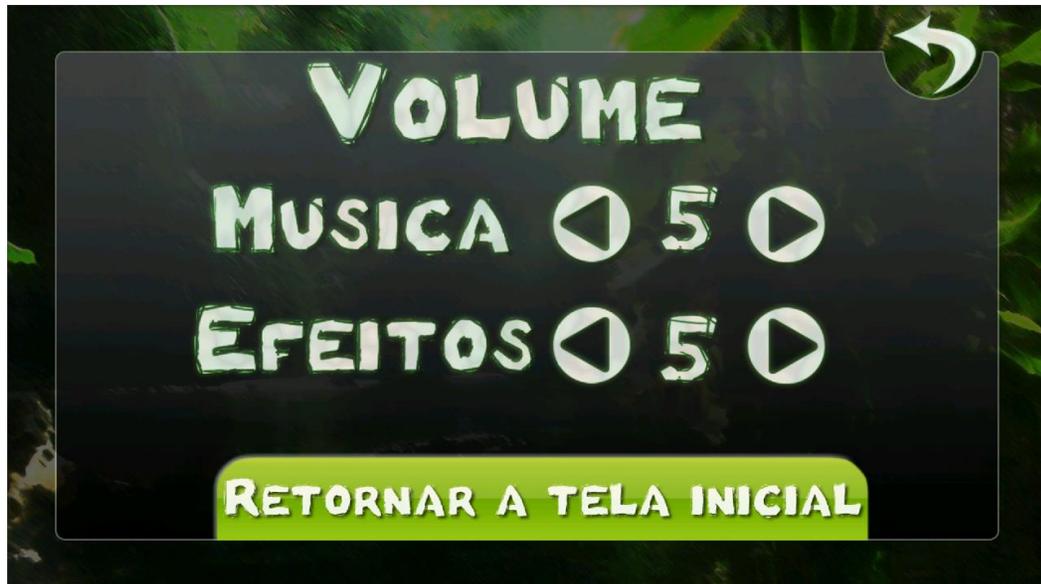


Figura 13 - Tela de Configuração



Figura 14 - Tela de Créditos

Na tela de créditos, apresentada na Figura 14 é exibido os nomes da equipe de desenvolvimento. Tocando na seta de retornar é realizada uma transição para a tela inicial.

4.2 Discussão

O desenvolvimento do documento de *Game Design* se mostrou a etapa mais ineficaz, pois não foi adotado um processo estruturado para a sua execução. Portanto, infelizmente foi adotada uma metodologia de tentativa e erro. A técnica mais eficaz adotada nesta etapa foi a prototipação, onde modelos de papel foram utilizados para simular o jogo. Através do uso de protótipos, puderam ser observados erros na mecânica do jogo, como regras demasiadamente complexas, sistema de batalha falho e elementos do jogo que poderiam ser eliminados sem afetar a mecânica básica do jogo.

A etapa de desenvolvimento do projeto da interface foi a mais longa de todo o trabalho, representando mais da metade do todo tempo gasto para a execução do trabalho. Como na escrita do documento de *Game Design*, a técnica mais eficiente utilizada na etapa de escrita do projeto de interface foi a prototipação. A taxa de retrabalho nesta etapa foi extremamente alta, pois muitas das telas necessitaram a criação de até dez protótipos até que uma versão atingisse um nível de qualidade aceitável. A cada nova versão do protótipo eram realizadas alterações como a correção posição de algum elemento, a adição de elementos ausentes e necessários para a composição da tela ou até a remoção de elementos avaliados como desnecessários para a composição da tela. Para a escrita da descrição das representações visuais das unidades e melhorias foi necessária a realização de uma vasta pesquisa por imagens que retratam as tribos utilizadas no jogo. Os resultados se mostraram escassos e limitados, o que dificultou a escrita da descrição de algumas unidades. Por este motivo foi necessário realizar alterações no documento de *Game Design*, aumentando a abrangência de tribos de cada região da América, possibilitando a utilização de um maior acervo de imagens para utilizar como base para a descrição das representações visuais das unidades.

O processo de criação e produção dos *assets* ocorreu de forma paralela ao de escrita do projeto de interface. Nele foi realizado apenas um polimento¹⁷ das imagens utilizadas dos protótipos. Devido ao curto espaço de tempo disponível para a execução do trabalho foi dada prioridade para a produção dos *assets* visuais, para que todos os elementos do jogo possuísem uma representação final. Por esse motivo apenas os sons necessários para um *feedback* positivo e negativo foram utilizados.

Como o desenvolvimento da interface aconteceu de forma isolada do projeto principal, muitos elementos do jogo não estavam disponíveis no momento da implementação da interface. Devido à falta da mecânica do jogo implementada, muitas características pertinentes ao comportamento do jogo devido a sua mecânica foram implementadas de forma provisória. Esta falta também impediu que alguns elementos da interface pudessem ser devidamente implementados, como: animação, sistema de batalhas completo, sistema de passagem de turnos, e sistema de recrutamento de novas unidades.

Durante a realização dos testes nos dispositivos móveis ocorreu o erro “OutOfMemoryError” que impedia a execução da aplicação. Este erro ocorre quando uma requisição de memória não pode ser atendida devido a falta de recursos do dispositivo. Toda aplicação Android é executada em uma máquina virtual dentro do sistema operacional, e cada instancia desta máquina possui uma quantidade limitada de memória. Para contornar este problema foram utilizadas duas abordagens distintas para reduzir o custo de memória do jogo. Foram elas: otimização dos *assets* dos elementos do jogo e otimização código da interface.

Após a realização de alguns testes foi observado que é mais eficiente carregar um único *asset* contendo todos os elementos de uma tela e realizar o mapeamento deles via código que carregá-los de forma isolada. Foi observado que uma grande parcela da memória é utilizada apenas para o endereçamento dos *assets* do jogo. Portanto

¹⁷ Jargão utilizado no meio de desenvolvimento de jogos. É um processo de aperfeiçoamento ou otimização de algum elemento do jogo. Pode ser realizado na arte, musica ou código do jogo.

reduzindo a quantidade de itens a serem carregados na memória, há uma boa redução no uso de memória pelo jogo. Dessa forma, uma única imagem contendo todos os *assets* necessários para uma tela foi criado, como apresentado na Figura 15. Inicialmente todos os *assets* do jogo estavam sendo carregados na memória. E muitos deles não estavam sendo utilizados durante o jogo. Para reduzir a quantidade de memória, foi implementado um sistema para carregar apenas os *assets* necessários para a exibição de uma tela, realizando o despejo da memória assim que um conjunto de *assets* não fossem mais utilizados. Com essas duas adaptações o jogo passou a consumir menos memória, permitindo que ele fosse executado em dispositivos mais simples, com uma quantidade mínima de 256MB de memória.

Após a realização dos testes de usabilidade e experiência de usuário nos dispositivos móveis, foi observada uma grande necessidade de tratamento da entrada de toque, pois existem muitas formas erradas de realizar um mesmo comando dentro do jogo. Como exemplo, podemos citar a tela de exploração, nela o jogador pode navegar pelo mapa arrastando o dedo pela tela. Para selecionar um quadrado do mapa basta tocar nele. A única avaliação realizada para validade desta ação era verificar se o toque inicial e o ponto em que o dedo era retirado da tela estavam dentro da área do botão. Assim, se o usuário tocar em um quadrado, arrastar o mapa e retornar o dedo para o ponto inicial e retirar o dedo, a ação era validada e o toque era registrado. Dessa forma além dos pontos iniciais e final é necessário avaliar se o jogador realiza algum movimento depois do toque inicial, invalidando a ação quando é comprovado um movimento superior a um limiar.



Figura 15 - Folha de assets

Outro problema observado foi o do movimento altamente preciso do mapa. Quando ocorre um toque na tela a área de contato do dedo com a tela varia até ele ser retirado da tela. O sistema calcula o ponto médio da área em contato com a tela fornecendo um valor exato das coordenadas do toque. Assim quando o jogador toca um quadrado específico para selecioná-lo ocorre uma pequena alteração no ponto central da tela, causando um pequeno tremor no mapa exibido na tela. Este tremor causa certo nível de desconforto no jogador. Assim foi necessário implementar um sistema que impede o mapa de se mover até que o deslocamento do ponto médio seja superior a um limiar.

5 CONCLUSÃO

Mesmo após a conclusão deste trabalho, realizando todos os passos necessários para o desenvolvimento de uma interface para um jogo, é difícil definir em que ponto a função de um *designer* de interfaces inicia e finaliza no desenvolvimento de um jogo. Pois o seu trabalho está altamente atrelado aos trabalhos do *game designer*, artista, músico e programador. Em grandes equipes de desenvolvimento ele pode ser responsável unicamente pela produção do projeto de interface, dividindo o seu trabalho com o *game designer* e programador. Já em equipes pequenas todo o trabalho do *designer* de interfaces pode ser realizado pelo programador sozinho.

Para o desenvolvimento deste trabalho foi utilizado um *framework* simples, possuindo apenas as funções e procedimentos essenciais para o funcionamento do jogo. Esta escolha foi realizada para que as peculiaridades do desenvolvimento da interface pudessem ser observadas durante a sua produção. Porém após a sua conclusão foi observado que o processo de projeto e desenvolvimento de uma interface para jogos é um trabalho mais intelectual do que técnico, pois cerca de dois terços do tempo total gasto para a execução deste trabalho foram gastos na etapa de escrita do projeto da interface e na produção dos elementos gráficos do jogo. Já na etapa de implementação do código da interface, foi gasto mais tempo lidando com as questões técnicas de administração de memória do que com o tratamento da entrada, que tem relação direta com a interface do jogo. Portanto é altamente recomendada a utilização de um *framework* mais robusto ou até mesmo uma *engine* para a produção do jogo. Estas ferramentas oferecem todo o suporte técnico necessário para a produção do jogo, permitindo que o designer de interface foque na produção de uma interface divertida e com um nível de usabilidade aceitável.

Para trabalhos futuros é interessante a realização da otimização da interface visando uma melhor experiência de usuário. Para isso são necessárias a implementação das animações dos elementos do jogo, como o movimento das unidades paradas e atacando e o movimento dos elementos do cenário do jogo, como árvores e água. Além

da animação, é necessária a inclusão de efeitos sonoros mais complexos e de uma trilha sonora a ser executada como som de fundo.

Após a conclusão da implementação da mecânica do jogo é necessária a criação de elementos para a interface para realizar a troca de turnos, o combate entre as unidades e a confirmação do recrutamento de novas unidades.

Quando concluído o projeto, o código jogo será publicado sob a licença GPL, permitindo futuras alterações no código.

6 REFERÊNCIAS

ANDREWS, M. Game UI Discoveries: What Players Want. **Gamasutra**, 23 fev. 2010. Disponível em: <http://www.gamasutra.com/view/feature/4286/game_ui_discoveries_what_players_.php>. Acesso em: 17 dez. 2012.

FAGERHOLT E., LORENTZON M. **Beyond the HUD: User Interfaces for Increased Player Immersion in FPS Games**. Chalmers University of Technology, 2009.

NOVAK, J. **Desenvolvimento de games**. Delmar Cengage Learning, 2011.

SCHELL, J. **A Arte do Game Design: O livro original**. Elsevier, 2011.

SCHNEIDERMAN, B., PLAISANT, C. **Designing The User Interface**. Pearson Education, 2005.

WEISER, M. **The Computer for the 21st Century**. 1991.

WIGDOR, D., WIXON, D. **Brave NUI World: Designing Natural User Interfaces for Touch and Gesture**. Elsevier, 2011.

ZECHNER, M., GREEN, R. **Beginning Android 4 Games Development**. Apress, 2011.

7 APENDICE A

GAME DESIGN

1 CONCEITO DO JOGO

1.1 Premissa

No jogo o jogador comanda seu exercito tribal através dos mapas no intuito de dominar os seus inimigos. Deverá fazer uma administração dos seus recursos e criar uma estratégia para conquistar o mapa e derrotar todos os seus inimigos.

1.2 Motivação do Jogador

O jogador deve explorar os mapas em busca de recursos para evoluir sua tribo e construir um exercito para dominar o seu inimigo. O jogo termina quando um dos jogadores tem sua aldeia tomada pela tribo inimiga. Sendo vitorioso aquele que conquistar todas as aldeias do mapa. O jogador pode perder o jogo quando tem sua aldeia dominada por uma tribo inimiga e tem seu exercito eliminado, ou senão através da desistência.

1.3 Diferencial

O jogo tem como protagonistas os povos indígenas da América, um tema pouco abordado no mercado de jogos. O tema possui um valor educacional, podendo assim ser utilizado como uma ferramenta educacional.

1.4 Público-alvo

O público-alvo do jogo são os jovens adultos, nascidos nas décadas de 1980 e 1990. Em especifico os jovens que tiveram um grande contato com os primeiros jogos de estratégia.

1.5 Gênero

O gênero escolhido para o desenvolvimento do jogo foi o de estratégia baseada em turnos, apresentando alguns elementos do RPG.

2 PROJETO DO JOGO

2.1 Tribos

Devido ao grande número de tribos indígenas na América, as tribos serão agrupadas pela sua semelhança cultural. Assim as tribos foram divididas em três grupos, as tribos do norte, centro e sul da América. Mesmo nesta divisão existe uma grande diferença cultural dentre as tribos dentro de cada grupo, porém, a mérito de simplificação estas diferenças serão ignoradas.

- **Norte:** as tribos do norte serão representadas pelas culturas norte-americanas. Dando ênfase nas tribos Cherokee, Navarro e Apache.
- **Centro:** as tribos do centro serão representadas pelas culturas mesoamericanas. Dando ênfase nas tribos Astecas, Incas e Maias.
- **Sul:** as tribos do sul serão representadas pelas culturas sul-americanas. Dando ênfase nas tribos Guarani, Ianomâmis e Pataxós.

2.2 Mapas

Os mapas serão formados por um conjunto de quadrados. Ordenados de forma de grade, de altura e largura iguais. Os quadrados são classificados em quatro grupos distintos: terreno, bloqueio, interação primária e interação secundária. Cada jogador pode ocupar apenas um quadrado por vez, e apenas um jogador pode ocupar um quadrado por vez.

2.2.1 Terreno

Os quadrados de terreno serão constituídos por terra, grama e areia. Os jogadores utilizaram este tipo de quadrado para a exploração do mapa. O quadrado de terreno é acessível se pelo menos um quadrado adjacente ortogonalmente também é acessível. Quando um quadrado de terreno apresenta uma entidade interativa ele é classificado como um quadrado de interação primária ou secundária.

2.2.2 Bloqueio

Os quadrados de bloqueio tem a finalidade de limitar a área de exploração do jogador. Eles serão representados por montanha, árvore e água. E eles não serão acessíveis de nenhuma direção.

a) Interação Primária

Os quadrados de interação primária são uma variação dos quadrados de terreno, porém possuem uma entidade interativa. Estas entidades são: construções e recursos ilimitados. Quando o jogador caminha para um quadrado de interação primária ele ativa o gatilho da entidade interativa, executando uma ação. Os quadrados de interação primária podem ser ativados mais de uma vez.

b) Interação Secundária

Os quadrados de interação secundária são como os de interação primária, porém permitem uma única utilização. Suas entidades interativas são: unidades, recursos finitos e bônus temporários.

c) Recursos

Os recursos são a moeda do jogo. Eles são necessários para erguer novas construções, recrutar novas unidades, fazer melhorias nas construções. Eles podem ser obtidos através da dominação dos quadrados de interação primária, garantindo uma quantidade daquele recurso por turno enquanto dominar aquele quadrado. Ou senão através dos quadrados de interação secundária, recebendo pequenas quantidades de recurso.

- **Ilimitados:** os recursos ilimitados são fornecidos pelos quadrados de interação primária. A pedra será fornecido pelas minas, a madeira será fornecida pelas florestas e os alimentos pelas fazendas. Uma vez ativado o quadrado interativo por um jogador, ele passará a fornecer uma quantidade de seu recurso a cada turno àquele jogador, até que outro jogador faça uma nova ativação naquele

quadrado. Os valores e tipos dos recursos fornecidos serão gerados aleatoriamente na criação do mapa.

- **Finitos:** os recursos finitos serão fornecidos pelos quadrados de interação secundária. Quando ativados pela primeira vez os quadrados fornecem uma pequena quantidade de um recurso e se tornam um quadrado de terreno normal. Os valores e tipos de recursos serão gerados aleatoriamente na criação do mapa.

d) Construções

As construções serão encontradas nos quadrados de interação primária. Existem três tipos de construções que podem ser encontradas pelo mapa, as bases dos jogadores, os observatórios, e as aldeias.

Cada jogador possui uma base no mapa, e não existem bases vazias pelo mapa. O jogador pode interagir quantas vezes forem necessárias com sua base. E quando ele interage com a base do inimigo ele irá entrar em modo de combate com aquele jogador. Se vitorioso o jogador passa a controlar a base de seu inimigo.

Os observatórios tem a função de revelar mapa quando ativados, podendo ser utilizado por qualquer jogador. Ele revela um numero aleatório de quadrados a sua volta, determinado na criação do mapa.

As aldeias são pequenas construções onde o jogador pode recrutar novas unidades, sendo renovada a cada sete turnos. Qualquer jogador pode recrutar unidades, sendo limitado apenas pela quantidade de unidades disponíveis na aldeia.

e) Unidades

As unidades serão encontradas nos quadrados de interação secundária. Existindo duas variações no tipo de unidade, ela pode ser aliada ou agressiva. Quando o jogador interagir com uma unidade aliada ela irá dar ao jogador a oportunidade de entrar para o

grupo do jogador. E quando o jogador interage com uma unidade agressiva ele irá entrar em modo de combate com aquela unidade.

2.3 Aldeia e Melhorias

Cada jogador inicia o jogo com uma base sobre o seu controle. Quando o jogador interage com a sua base ele irá sair do modo de exploração e irá para o modo de administração de sua base. Lá ele poderá criar novas construções para a sua aldeia, fazer melhorias para as construções existentes, e recrutar novas unidades.

a) Aldeia

A aldeia é a principal construção da base. Ela é um pré-requisito para quase todas as outras construções da base, e a cada nível fornece uma quantidade maior de recursos por turno. E também possibilita a pesquisa de novas tecnologias para o aumento da produção de recursos.

- **Aldeia I** → **Custo:** 0C 0M 0P
Bônus: +2C +2M +2P /dia
Pré-requisito: Nenhum
- **Aldeia II** → **Custo:** 0C 10M 10P
Bônus: +4C +3M +3P /dia
Pré-requisito: Aldeia I
- **Aldeia III** → **Custo:** 0C 30M 30P
Bônus: +6C +4M +4P /dia
Pré-requisito: Aldeia II
- **Aldeia IV** → **Custo:** 0C 60M 60P
Bônus: +8C +5M +5P /dia
+1 Unidade Lendária /7dias
Pré-requisito: Aldeia III

b) Totem

O totem é a construção que fornece proteção mística (DefM) para as unidades presentes na aldeia. Sendo também um pré-requisito para a construção do templo.

- **Totem I**→ **Custo:** 0C 10M 0P
Bônus: +5% DefM
Pré-requisito: Arena II
- **Totem II**→**Custo:** 0C 10M 5P
Bônus: +15% DefM
Pré-requisito: Totem I, Templo
- **Totem III**→**Custo:** 40C 10M 0P
Bônus: +25% DefM
Pré-requisito: Totem II, Aldeia III

c) Defesa

As construções de defesa tem o intuito de proteger a base. A cada nível ela é capaz de absorver certa porcentagem do dano físico (Def) causado às unidades presentes na aldeia.

- **Defesa I**→ **Custo:** 0C 10M 0P
Bônus: +10% Def
Pré-requisito: Arena I
- **Defesa II**→**Custo:** 0C 0M 15P
Bônus: +15% Def
Pré-requisito: Defesa I, Aldeia II, Tec. Madeira I
- **Defesa III**→**Custo:** 15C 10M 0P
Bônus: +25% Def
Pré-requisito: Defesa II, Totem III

d) Arena

A arena irá produzir a unidade básica, a unidade de longo alcance e a unidade animal. No seu primeiro nível ela irá produzir apenas unidades básicas, sendo que cada tribo possui uma unidade básica diferente. No seu segundo nível ela irá produzir as unidades de longo alcance e terá um aumento na produção de unidades básicas por ciclo.

- **Arena I** → **Custo:** 5C 5M 5P
Bônus: +5 Unidades Básicas /7dias
Pré-requisito: Arena I
- **Arena II** → **Custo:** 15C 15M 15P
Bônus: +7 Unidades Básicas /7dias
+5 Unidades Longo Alcance /7dias
Pré-requisito: Arena I, Aldeia II
- **Arena III** → **Custo:** 15C 10M 0P
Bônus: +9 Unidades Básicas /7dias
+7 Unidades Longo Alcance /7dias
+4 Unidades Animal /7dias
Pré-requisito: Arena II

e) **Templo**

O templo irá produzir as unidades místicas. E quando for construído irá fornecer a um bônus de proteção mística para a base.

- **Templo** → **Custo:** 20C 20M 20P
Bônus: +3 Unidades Místicas /7dias
Pré-requisito: Totem I

f) **Tecnologia em Carne**

Quando pesquisada a tecnologia de carne irá permitir que mais aldeões trabalhem na extração de comida da aldeia. Alocando mais aldeões nas fazendas dominadas pela tribo.

- **Tec. Carne I** → **Custo:** 5C 0M 0P
Bônus: +10% C /dia
Pré-requisito: Aldeia I
- **Tec. Carne II** → **Custo:** 10C 0M 0P

Bônus: +20% C /dia

Pré-requisito: Tec. Carne I, Aldeia II

- **Tec. Carne III → Custo:** 15C 0M 0P

Bônus: +30% C /dia

Pré-requisito: Tec. Carne II, Aldeia III

g) Tecnologia em Madeira

Quando pesquisada a tecnologia de madeira irá permitir que mais aldeões trabalhem na extração de madeira da aldeia. Alocando mais aldeões nas florestas dominadas pela tribo.

- **Tec. Madeira I → Custo:** 0C 5M 0P

Bônus: +10% M /dia

Pré-requisito: Aldeia I

- **Tec. Madeira II → Custo:** 0C 10M 0P

Bônus: +20% M /dia

Pré-requisito: Tec. Madeira I, Aldeia II

- **Tec. Madeira III → Custo:** 0C 15M 0P

Bônus: +30% M /dia

Pré-requisito: Tec. Madeira II, Aldeia III

h) Tecnologia em Pedra

Quando pesquisada a tecnologia de pedra irá permitir que mais aldeões trabalhem na extração de pedra da aldeia. Alocando mais aldeões nas minas dominadas pela tribo.

- **Tec. Pedra I → Custo:** 0C 0M 5P

Bônus: +10% P /dia

Pré-requisito: Aldeia I

- **Tec. Pedra II → Custo:** 0C 0M 10P

Bônus: +20% P /dia

Pré-requisito: Tec. Pedra I, Aldeia II,

- **Tec. Pedra III** → **Custo:** 0C 0M 15P

Bônus: +30% P /dia

Pré-requisito: Tec. Pedra II, Aldeia III

2.4 Personagens

O jogo iniciará com cada jogador comandando um explorador e um pequeno grupo de unidades, variando o conjunto de unidades para cada tribo. Cada jogador possui apenas um explorador, e quando uma batalha é perdida e o desbravador do jogador é eliminado ele reaparece na aldeia do jogador. Novas unidades só poderão ser adicionadas ao grupo do desbravador quando o mesmo estiver presente no quadrado de uma de suas aldeias.

Cada unidade possui sete atributos básicos: vitalidade, ataque, defesa física, defesa mística, iniciativa, destreza e crítico.

- **Vitalidade (Vit):** determina a quantidade de pontos de vida da unidade. Quando sua vitalidade chega a zero a unidade morre e é retirada do grupo do desbravador.
- **Ataque (Atq):** composto de dois valores, um mínimo e um máximo. Quando um ataque é realizado, um valor aleatório é gerado entre o valor mínimo e o máximo do ataque para determinar a quantidade de pontos de vida será subtraído da unidade atacada.
- **Defesa física (Def):** determina a taxa de absorção do dano físico. Reduz o dano em uma porcentagem pré-definida.
- **Defesa mística (DefM):** determina a taxa de absorção do dano místico. Reduz o dano em uma porcentagem pré-definida.
- **Iniciativa (Ini):** determina a ordem em que as unidades irão fazer suas ações na batalha. Cada unidade possui um valor único entre 0 e 1 como iniciativa. No início de cada turno um número aleatório entre 0 e o número

de unidades em batalha é atribuído a cada unidade. O valor da iniciativa é utilizado como critério de desempate.

- **Destreza (Des):** probabilidade de acerto do ataque. Definido por uma porcentagem. Toda vez que um ataque é realizado um numero aleatório entre 0 e 100 é gerado para avaliar se o ataque atingiu o alvo. Em caso negativo nenhum dano é causado ao alvo, e nenhum outro teste é realizado.
- **Crítico (Crit):** verifica se o ataque realizado irá realizar um dano critico ou um dano comum. Ela é definida por uma porcentagem. Toda vez que é confirmado um ataque é gerado um numero aleatório entre 0 e 100 para verificar o critico. Se confirmado o dano é aumentado em 50%.

a) Unidade Básica

As unidades básicas são os guerreiros mais comuns de cada tribo. Eles são a linha de frente em qualquer batalha. São as unidades mais baratas e as que são produzidas em maior quantidade a cada ciclo.

Nome	Vit	Atq	Def	DefM	Ini	Des	Crit
Tribo do Norte: Guerreiro com Machado	20	5	15	0	0,21	90	5
Tribo do Centro: Guerreiro com Honda	20	5	15	0	0,23	90	5
Tribo do Sul: Guerreiro com Lança	20	5	15	0	0,22	90	5

b) Unidade de Longo Alcance

As unidades de longo alcance são os caçadores de cada tribo. Eles possuem armas para atacar os seus inimigos a distancia. É a segunda unidade mais barata e com mais disponibilidade de recrutamento de cada tribo.

Nome	Vit	Atq	Def	DefM	Ini	Des	Crit
Tribo do Norte: Guerreiro com Tomahawk	25	15	15	25	0,63	60	50
Tribo do Centro: Guerreiro com Boleadeira	25	15	15	25	0,61	60	50
Tribo do Sul: Arqueiro	25	15	15	25	0,62	60	50

c) Unidade Animal

A unidade animal é uma fera domesticada pela tribo para auxiliar nas batalhas. É uma unidade com um custo relativamente baixo e de grande poder em batalha, porém poucas são produzidas por ciclo para o recrutamento.

Nome	Vit	Atq	Def	DefM	Ini	Des	Crit
Tribo do Norte: Búfalo	50	15	25	10	0,31	80	10
Tribo do Centro: Jaguar	50	15	25	10	0,30	80	10
Tribo do Sul: Lobo Guará	50	15	25	10	0,32	80	10

d) Unidade Mística

As unidades místicas são representadas pelos feiticeiros de cada tribo. São as unidades mais próximas dos deuses que as tribos veneram. E possuem poderes místicos para atacar os seus inimigos. As unidades místicas possuem um alto dano, porém são muito frágeis e não resistem a muitos ataques.

Nome	Vit	Atq	Def	DefM	Ini	Des	Crit
Tribo do Norte: Xamã	10	25	5	50	0,13	70	40
Tribo do Centro: Sacerdote	10	25	5	50	0,11	70	40
Tribo do Sul: Pajé	10	25	5	50	0,12	70	40

a) Unidade Mítica

A unidade mítica é a última unidade possível de ser recrutada, e é a mais forte de cada tribo. Elas são as figuras veneradas por cada uma das tribos, consideradas deuses. Elas são as unidades mais caras para se recrutar e apenas uma é disponibilizada por turno. Porém cada uma possui um grande poder, capaz de destruir grandes grupos inimigos.

Nome	Vit	Atq	Def	DefM	Ini	Des	Crit
Tribo do Norte: <i>Skin Walker</i> (Troca-peles)	90	35	55	85	0,93	75	25
Tribo do Centro: Quetzalcatl	90	35	55	85	0,91	75	25
Tribo do Sul: Tupã/Garaci	90	35	55	85	0,92	75	25