

Projeto Wargroove

Especificação de Requisitos de Software

Versão 1.1

15/09/2020

Versão	Autores	Data	Ação
0.5	Gustavo Dirschnabel e João Jandre Paraquetti	13/09/2020	Esboço inicial
1.0	Gustavo Dirschnabel e João Jandre Paraquetti	15/09/2020	Primeiro Draft
1.1	Gustavo Dirschnabel	19/09/2020	Adicionados RF14-RF16

Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos de software.

1 Introdução

1.1 Objetivo: desenvolvimento de um programa que suporte a disputa de partidas de wargroove na modalidade usuários contra usuário.

1.2 Definições: Wargroove é um jogo de estratégia em turnos onde os jogadores controlam um exército, liderado por um comandante, com o objetivo de destruir a base do oponente ou nocautear o comandante inimigo.

1.2.1 Turnos: Na sua vez de jogar, um jogador pode dar um comando para cada uma das peças sob seu controle (referido como “**ação**”). Quando um jogador fica sem ações disponíveis ou fica satisfeito com o seu turno, ele termina o seu turno e começa o de seu oponente. Alguns efeitos no jogo

só são ativados na chegada da vez de um jogador, esse são efeitos aplicados “no começo de seu turno”.

1.2.2 Peças: Wargroove possui dois tipos de peças, **Estruturas** e **Unidades**, que possuem as seguintes características:

- **Estruturas** são peças estáticas divididas em dois grupos: **Estruturas de recrutamento** e **Vilas**. Usando a ação de estruturas de recrutamento, um jogador pode pagar um custo em ouro para comprar uma Unidade, posicionando-a adjacente a estrutura. Vilas são estruturas que dão ao jogador uma quantidade de ouro no começo de seu turno.
- **Unidades** são peças móveis que representam os soldados que cada jogador possui. Unidades possuem as ações **Atacar**, **Capturar**, **Esperar** e **Reforçar**. Wargroove possui alguns **tipos de Unidades**, que possuem vantagens e desvantagens. Comandantes são Unidades insubstituíveis que possuem uma ação especial, o **Groove**.
- Pontos de vida (abreviado para **PV**): Toda peça tem um valor de 0% a 100% que representa a integridade da peça. Se os PVs de uma peça ficam menores ou iguais a 0%, ela é removida (unidades) ou neutralizada (estruturas).
- Todas as peças de um jogador são consideradas **controladas** por ele. Peças de seu oponente são consideradas **inimigas**. Estruturas não controladas por nenhum dos jogadores são consideradas **neutras**.
- Comandantes e Estruturas controladas recebem uma pequena porcentagem de PVs a custo nenhum no começo do turno.

1.2.3 Tabuleiro: As peças de cada jogador ocupam e se deslocam em “**Quadrados**”, subdivisões do mapa de batalha em forma de grade. Duas peças não podem ocupar a mesma posição no tabuleiro. Adicionalmente, cada quadrado possui um tipo de terreno, que influencia no movimento de unidades e no ataque entre peças. O estado inicial da partida é:

- Cada jogador possui seu comandante já posicionado no tabuleiro.
- Cada jogador controla as estruturas de um lado do mapa.
- O centro do mapa possui estruturas neutras.

1.2.4 Movimentação: Uma opção exclusiva a Unidades que permite que ela saia do quadrado atual e vá a uma posição destino selecionada pelo jogador, passando pelos quadrados entre os dois. Algumas regras são associadas à movimentação:

- Cada tipo de Unidade possui um número máximo de quadrados que pode se locomover usando seu movimento. A posição de destino de um movimento deve estar dentro desse alcance.
- Unidades podem passar por quadrados ocupados por peças controladas, mas são impedidas de passar por quadrados que possuem peças inimigas.
- Movimentação deve ser combinada com uma ação da Unidade.
- Uma Unidade que já teve sua ação esgotada não poderá se mover até o próximo turno do jogador que a controla.
- É possível selecionar o quadrado atual de uma Unidade como destino de seu movimento, ela apenas fica onde está.

1.2.5 Ações:

1.2.5.1 Recrutamento: Cada estrutura de recrutamento controlada pelo jogador pode consumir uma quantidade de ouro para posicionar uma Unidade em um quadrado adjacente à estrutura, definido pelo jogador. Cada tipo de Unidade terá um custo diferente associado a ela.

1.2.5.2 Atacar: Permite com que os jogadores eliminem Unidades inimigas e capturem estruturas deles. Necessária para se atingir qualquer uma das condições de vitória. Possui as seguintes regras e mecânicas:

- Pode ser iniciado apenas por Unidades. A unidade que utiliza essa ação contra outra peça é chamada **atacante**, seu alvo é chamado **defensor**.
- Todas as Unidades possuem um alcance de ataque. A opção de realizar um ataque aparece apenas se, a partir do quadrado destino de seu movimento, um inimigo estiver dentro do alcance de ataque.

- Se vários inimigos são alvos válidos de uma ação de ataque, apenas um deles pode ser escolhido como defensor.
- A quantidade de PVs reduzida com um ataque (**Dano**), é dependente de dois fatores: O dano base de um tipo de unidade contra outro e a quantidade de PVs do atacante no momento do ataque.
- Em um combate, primeiro a unidade atacante aplica seu dano na defensora e depois, levando em conta a sua nova quantidade de PVs, a defensora contra-ataca. Isso significa que atacantes possuem vantagem.
- Todas as Unidades possuem uma condição específica ao seu tipo que, se cumprida, aumenta o dano de um ataque ou contra-ataque multiplicativamente. Essas condições são geralmente associadas ao posicionamento da Unidade em relação as outras peças controladas pelo jogador ou a algum tipo de terreno de um quadrado.

1.2.5.3 Capturar: Uma interação que ocorre entre uma Unidade e uma estrutura neutra e que dá o controle da estrutura ao jogador que a capturou. Possui as seguintes regras:

- Uma estrutura neutra só pode ser capturada por alguns tipos de Unidades.
- A opção de capturar só ficará disponível se o destino final do movimento da Unidade for adjacente ao quadrado em que a estrutura está.
- Quando capturada, a estrutura receberá metade do valor de PVs da unidade que a capturou.

1.2.5.4 Esperar: É a opção padrão que apenas resulta em movimento da Unidade.

1.2.5.5 Reforçar: Uma opção para transferir PVs de Estruturas a Unidades, pagando um custo, tem as seguintes regras:

- Apenas fica disponível se o destino final do movimento da Unidade for adjacente a uma estrutura controlada.

- Transfere o máximo de PV's possível entre os dois, com esse limite definido pela quantidade de ouro do jogador e dos PVs da estrutura selecionada.

1.2.5.6 Groove: Habilidades especiais exclusivas a cada comandante, que podem ser utilizadas para conseguir uma vantagem no jogo.

Possui as seguintes mecânicas e regras:

- Precisa ser carregada para poder ser utilizada. Seu nível de carregamento começa em 0% e alcance o máximo de 100%, quando fica disponível para uso.
- O comandante controlado por um jogador recebe uma pequena carga de Groove a nenhum custo, no começo de seu turno.
- Envolvimento do comandante em batalhas e capturas carrega uma quantidade adicional de Groove.
- Quando a habilidade é utilizada, sua quantidade de carga torna-se 0%.

1.2.6 Abreviações:

CM1: *Click* com o botão esquerdo do mouse

CM2: *Click* com o direito do mouse

1.3 Referências

<https://wargroovewiki.com/Unit_Types> (Imagens de unidades)

<<https://wargroovewiki.com/Buildings>> (Imagens de estruturas)

2 Visão geral

2.1 Arquitetura do programa: programa orientado a objetos, standalone.

2.2 Premissas de desenvolvimento:

- O programa deve apresentar uma interface gráfica bidimensional;
- O programa deve ser implementado em Java, devendo executar em qualquer plataforma que disponha da máquina virtual Java.

3 Requisitos de software

3.1 Requisitos funcionais

RF1-Estabelecimento das facções: o programa deve apresentar a opção no menu inicial “*Escolha de facção*”, que permite ao usuário escolher entre 4 facções diferentes, escolha que deve ser feita antes do início do jogo.

RF2-Iniciar: o programa deve apresentar a opção “iniciar” na interface de jogo, para o início de uma nova partida, onde o jogador insere sua identificação e aperta o botão para iniciar a partida e onde o programa define a ordem de turnos dos jogadores. O tabuleiro carregará em um sucesso dessa funcionalidade, em uma falha, nada acontece.

RF3-Movimentação de unidades: o programa deve suportar o movimento de unidades, durante seu turno, seguindo as regras definidas em 1.2.4, o jogador deve poder, por meio de um CM1 para seleção da unidade, seguido de um segundo CM1 para confirmar o movimento ou um M2 para cancelar o movimento, além disso o programa deve atualizar o tabuleiro para refletir o novo estado do mesmo.

RF4-Realce de quadrados: o programa deve apresentar o elemento gráfico de destaque de quadrados do tabuleiro quando o jogador selecionar uma unidade no tabuleiro por meio de um CM1, para mostrar o alcance de movimento da unidade, assim como o alcance de ataque da mesma.

RF5-Recrutamento de unidades: o programa deve implementar o recrutamento de novas unidades, que ocorre durante o turno do jogador, como definido em 1.2.5.1, abrindo uma janela específica para esse propósito após um CM1 na estrutura de recrutamento, caso o jogador não cumpra os requisitos para o recrutamento, o botão de recrutamento deve ser desabilitado. O programa deve atualizar o tabuleiro para refletir esta ação.

RF6-Realizar Ataques: durante seu turno e seguindo as regras definidas em 1.2.5.2, um jogador deve poder atacar unidades inimigas. Esta ação é feita

através de um CM1 na unidade atacante, um CM1 na opção de ataque e por último um CM1 na unidade que será atacada. Quando um jogador escolhe uma unidade e ela não pode atacar, a opção de atacar não deve aparecer. Se o ataque ocorrer, as peças envolvidas devem ser atualizadas para refletir as regras definidas em 1.2.5.2.

RF7-Passar o turno: o programa deve permitir que, a qualquer momento do seu turno, o jogador possa terminá-lo utilizando um botão na interface de batalha, fazendo com que a vez passe para o oponente.

RF8-Matriz de dano: a qualquer momento durante a partida, um jogador deve poder utilizar um botão na interface de batalha para abrir uma matriz que descreve a porcentagem de dano que cada unidade dá em todos os tipos de peças.

RF9-Inspeccionar terreno: a qualquer momento, um jogador deve poder inspecionar o terreno de um quadrado do tabuleiro, com um CM2, para abrir uma interface que mostra as características do terreno.

RF10-Capturar estruturas neutras: durante seu turno, respeitando as regras definidas em 1.2.5.3, o jogador deve poder transformar uma estrutura neutra em uma aliada, ganhando os benefícios da mesma. Esta ação é feita através de um CM1 na unidade desejada, e um CM1 na opção de captura e por último um CM1 na estrutura desejada. O programa deve atualizar o tabuleiro para refletir esta ação.

RF11-Reforçar Unidades: durante o seu turno, respeitando as regras definidas em 1.2.5.5, o jogador deve poder, através de um CM1 na unidade desejada e um CM1 na opção de reforçar, reforçar uma unidade aliada. O programa deve atualizar a peça para refletir esta ação.

RF12-Usar o Groove: durante seu turno, respeitando as regras definidas em 1.2.5.6, o jogador deve poder, através de um CM1 em seu comandante, seguido de um CM1 na opção de Groove, e por último um CM1 no alvo, utilizar o

groove do comandante, quando esta ação é feita, o jogo deve atualizar o tabuleiro ou as peças afetadas, para refletir o groove específico.

RF13-Render-se: um jogador deve poder render-se a qualquer momento utilizando um botão na interface de batalha, para desistir da partida e conceder vitória ao seu oponente. Após isso, a partida não está mais em andamento.

RF14-Conectar: Um jogador deve, no menu principal antes de começar uma partida, conectar-se ao servidor NetGames usando um botão. Quando um jogador se conecta, ele poderá iniciar a partida (RF12).

RF15-Desconectar: Um jogador que se conectou ao servidor poderá se desconectar se a partida não estiver em andamento usando um botão na interface de jogo. Isso o levará ao menu principal.

RF16-Receber Jogada: A interface NetGames deve, com o jogador que não possui a vez, receber as jogadas do jogador da vez pela rede, reproduzindo-as no mesmo momento que acontecem e na tela de quem as recebeu.

3.2 Requisitos não funcionais

RNF1-Especificação de projeto: além do código Java, deve ser produzida especificação de projeto baseada em UML, segunda versão;

RNF2-Interface gráfica para usuário: o programa deverá ter interface gráfica dinâmica, que alterna entre menu principal e batalha, sendo a segunda sincronizada entre os usuários durante a partida;

RNF3-Peças dos jogadores: as peças dos jogadores devem ser representadas no programa através de imagens retiradas da wiki do jogo (referência 1 e 2);

RNF4-Tecnologia de interface gráfica para usuário: a interface gráfica deve ser baseada em JavaFX.

RNF5-Plataforma de desenvolvimento de código: deve-se utilizar a IDE IntelliJ para implementação.

RNF6-Plataforma de modelagem de diagramas: deve-se utilizar a plataforma Visual Paradigm para modelar os diagramas do projeto.