Especificações

Alunos:

Bárbara Aparecida de Castro Silva - Nº USP: 6431012
Caio Salgado - NºUSP: 6920219
Luciana dos Santos Kayo - Nº USP: 6430992
Vinícius Garcia de Rezende - Nº USP: 6466279

1. Informações gerais do jogo

a. **Proposta**:

O jogo será feito a partir de um mapa fechado, visto por cima. O mapa será representado por uma matriz, onde os personagens poderão ser movimentados para as quatro direções principais: direita, esquerda, cima e baixo.

Cada jogador controlará um herói principal, com alguma habilidade especial, e um exército de outros guerreiros, com habilidades básicas de ataque e defesa. O jogador começará o jogo em uma base próxima a um canto do mapa, com uma quantidade limitada de recursos para criar seu exército inicial, e deverá administrar seus recursos ao longo da partida para que possa adquirir mais guerreiros.

O jogo será inspirado na série A Song of Ice and Fire, de George R. R. Martin, ambientado na época medieval. Dessa forma, após colocar seu nome na tela inicial, o jogador deverá escolher uma Casa pela qual irá lutar. Cada Casa tem "*stats"* específicas para os guerreiros básicos (pontos de vida *(HP)*, dano de ataque, defesa, velocidade, etc) e um herói principal que a representará. O herói tem pontos de magia (*MP*) que poderão ser usados para "*castar spells*" (usar habilidade) ou para chamar unidades extras ("*summon*"). As unidades "*summonáveis*" também serão diferenciadas pela Casa escolhida.

Haverá templos/lojas espalhadas pelo mapa onde é possível recuperar HP ou MP. Também haverá sub-bases onde o jogador ganhará recursos (coins) por cada turno que dominar aquela sub-base.

Em cada turno, o jogador move as unidades sob seu controle de acordo com o número de posições determinado por suas velocidades, ou seja, pelo tanto de "quadradinhos" que ela pode percorrer no mapa.

O modo exato de ataque ainda não foi determinado, mas se uma unidade estiver no caminho de uma unidade inimiga, elas não poderão se mover sem lutar.

Haverá um modo único de jogo, o **Last Man Standing**. O objetivo do jogador será derrotar o herói inimigo, tendo obrigatoriamente que ter o seu herói vivo ao final do jogo.

b. **O que foi implementado até agora:**

Temos uma tela inicial simples, que pede ao jogador para apertar espaço pra jogar, e assim abre-se o mapa do jogo. O Player 1 está escolhendo automaticamente a Casa Stark e o Player 2 a Casa Lannister, por enquanto. As bases dos Players ficam nos cantos superior esquerdo e inferior direito, e cada herói já começa em sua base.

Implementamos o personagem selecionado com as posições para onde ele pode se mover destacadas. Quando o personagem está selecionado, um segundo clique numa área destacada o move até lá, e então chega o turno do outro jogador.

Temos algumas coisas já implementadas mas que não estão sendo usadas, porque precisam de menus, que ainda não criamos.

c. **Pretensões**:

O jogo final contará com uma mapa com sub-bases. Também apresentará uma tela inicial melhorada, onde os jogadores podem digitar seus nomes para guardar suas pontuações (o score será implementado na versão 1.0 do jogo).

As imagens próprias do jogo serão acrescentadas e seus papéis, descritos (onde são usadas e o que representam) no manual do usuário. As regras finais do jogo também farão parte do manual.

Por fim, não teremos apenas um personagem na tela, obviamente, mas pretendemos implementar a seleção de um personagem específico, dentre todos os personagens do exército controlado pelo jogador. E, como é estritamente necessário para uma boa prática de programação, padronizaremos o código.

d. Como rodar o jogo?

O jogo não necessita de instalação. Foi gerado em um arquivo .jar que já inclui as bibliotecas do LWJGL necessárias para sua execução.

O .jar foi gerado com especificações para Windows, dado que programamos nosso projeto nesse mesmo sistema operacional.

e. Definição das classes e relacionamentos:

Juntamente com o código do projeto e o .jar, enviamos um diagrama de classes. As classes especificadas no diagrama são:

i. **BasePosition**: herda de Position. Apesar de não ter atributos definidos ainda, sabemos que uma base não poderá deixar que um jogador inimigo use-a para criar seu exército (uma vez que um jogador só pode adquirir novos guerreiros dentro de sua base) e também que ela não gera "coins" (recursos) para o jogador.
ii. **Character**: personagem genérico, tem vida, velocidade, casa a qual pertence, um tipo de ataque.
iii. **Effect**: tem uma relação de composição com Skill, pois só pode ocorrer quando uma habilidade é usada, e se ela for "destruída", ele é destruído junto. É o efeito que causa um tipo de dano diferente em cada "stat" do personagem inimigo, como por exemplo, diminuir vida, mana ou velocidade.
iv. **FullScreen**: implementa o modo Full Screen.
v. **Game**: composição de Map, Player e outras coisas que ainda serão implementadas. Inicializa e finaliza o jogo. É no game que um novo mapa é gerado e novos jogadores serão criados. Quando Game é finalizado, tudo é destruído com ele.
vi. **GameController**: aqui são implementados os métodos que controlam os turnos do jogo, além de atributos para os jogadores. É também a classe que gera o mapa. Os turnos ainda estão pseudo implementados.
vii. **Hero**: herda de Character, sendo um personagem especial no jogo. Um herói é o guerreiro principal do exército de um jogador. Como diferencial, pode usar habilidades especiais, "summonando" outras unidades, e com isso, gastar mana.
viii. **Map**: composição de posições. Posições não existem se um mapa não for declarado no jogo. Representado por uma matriz de Positions.
ix. **Menu**: tela inicial do jogo, bem simples, apenas para testes.
x. **Player**: representa o jogador. Player está associado à classe Character: um player controla vários personagens, de diferentes tipos.
xi. **Position**: representa uma posição do mapa e contém o tamanho de uma célula, em pixels, que forma a matriz positionMatrix, que tem atributos para saber se aquela posição está ocupada por um personagem, de qual jogador etc. Uma Position é uma posição no grid, ou seja, numa tela de 800x600 pixels, uma posição de tamanho 4x4 pixels faz com que o grid tenha 200x150 posições na matriz. Uma Position é um quadradinho da matriz que nos auxilia a imprimir o mapa na tela e não um pixel!
xii. **Skill**: tem uma relação de dependência com Hero, pois cada herói tem uma habilidade diferente que lhe permite chamar um Summon diferente. Uma habilidade tem efeitos.
xiii **SubBasePosition**: herda de Position e gera novos recursos, por turno, quando conquistada por um jogador.
xiv. **Tempo**: controla o fps.

xv. **TestMain**: o main do projeto.
xvi. **TypeOfSkill**: tipos de habilidades e seus atributos e métodos.
xvii. **Warrior**: guerreiro genérico, também herda de Character, mas como diferencial em relação aos outros personagens, tem um custo em “coins” (recurso) para que possa ser criado, aumentando assim o exército.