

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA – INE
Engenharia de Software I
Especificação de Requisitos de Software

Projeto : Fallen Angels

Versão: 4.0

Data: 28/11/2018

Versão Autores Data Ação

Versão	Autores	Data	Ação
1.0	Franck KUMAKO, Gabriel Campos, Fierre	11/09/2018	Estabelecimento dos requisitos
2.0	Franck KUMAKO, Gabriel Campos,	09/10/2018	Modelagem de classes, casos de uso, visão geral de iteração e refinamento de casos de uso com diagrama de atividade
3.0	Franck KUMAKO, Gabriel Campos	13/11/2018	Refinamento de todo o conteúdo do primeiro estágio e mais refinamento de sequência e modelagem de estado
4.0	Franck KUMAKO, Gabriel Campos	28/11/2018	Modelagem de classes, casos de uso, visão geral de iteração, refinamento de casos de uso e modelagem de estados e algoritmos de métodos

Conteúdo:

1. Introdução;
2. Visão geral;
3. Requisitos de software.
4. Esboço da parte gráfica.

1 Introdução

1.1-Objetivo: Desenvolvimento de um jogo que suporte a disputa de partidas de combate em turnos na modalidade usuário *versus* usuário.

-Como o jogo funciona: trata-se de um jogo de combate em turnos, onde primeiramente é disponibilizada ao usuário uma coleção de personagens da qual ele deve escolher apenas um. Feito isso, o programa leva os dois jogadores à tela principal, onde cada um deve, na sua vez, escolher qual ação será realizada naquele turno. Cada personagem possui um *set* de ações disponíveis, sendo elas: três tipos diferentes de golpes que infligem dano ao adversário (o dano dos golpes será calculado de acordo com uma constante “dano base” e um fator de aleatoriedade) e três tipos diferentes de ações defensivas (que recuperam uma porcentagem de vida do usuário atual e aplica à ele um efeito que reduz o dano do oponente na próxima rodada). Um jogador ganhará o jogo quando reduzir a vida de seu oponente a zero.

1.2-Regras do jogo

R1: O jogo é realizado entre dois jogadores em uma arena virtual.

R2: Cada jogador possui um número de *hitpoints*, determinado de acordo com o personagem que escolheu. Este número nunca poderá exceder seu valor inicial (mesmo que uma ação de cura adicione mais pontos de vida que o personagem possuía no começo da partida).

R3: Toda ação, seja ofensiva ou defensiva, possui seu dano base e *range*. O dano final é calculado multiplicando-se o dano base por um número aleatório gerado de acordo com o *range*. Se o *range* for 30, a ação poderá infringir 30% de dano a mais, ou a menos. Se o *range* de uma ação ofensiva for maior que 100, o dano mínimo que ela irá infringir será zero (fazendo, assim, que uma ação ofensiva nunca cure o oponente).

2 Visão Geral

2.1 Arquitetura do programa: programa orientado a objetos, plataforma distribuída.

Premissas de desenvolvimento:

- O Jogo deve ser interativo;
- O Jogo deve executar de forma distribuída, na qual dois jogadores se comunicam por meio do servidor NetGames;
- 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:*

Requisito funcional 1 – conectar: o programa deve apresentar a opção de menu “*conectar*” para requisitar uma nova conexão com o servidor, considerando as seguintes restrições:

- Caso haja falha na conexão, o programa deve mostrar isso por meio de uma mensagem de erro, com a sua descrição.
- Caso a conexão seja estabelecida com sucesso, o programa deve mostrar isso por meio de uma mensagem de sucesso.

Requisito funcional 2 - Escolher o nome: Deve ser disponibilizado ao usuário uma interface para informar o nome do jogador

Requisito funcional 3 -Escolher Personagem: Deve ser disponibilizado ao usuário uma interface para selecionar o lutador pretendido e os seus respectivos golpes.

Requisito funcional 4 -Jogar: Após clicar no botão “Jogar” a partida vai ser iniciada, o software deve apresentar a tela principal com as configurações previamente definidas pelos jogadores, isso inclui o carregamento dos elementos gráficos do jogo (Arena e informações dos jogadores), a definição aleatória das casas iniciais dos jogadores, e definir, também aleatoriamente, qual jogador iniciará a partida. Esta opção deve ser acessível por menu.

Requisito funcional 5 Reiniciar partida: O sistema deve permitir que uma partida seja reiniciada, permitindo que os jogadores que estavam jogando comecem uma nova partida exatamente com a mesma configuração. O sistema deve validar, caso a partida estiver em curso, se o usuário realmente deseja reiniciar a partida

Requisito funcional 6 Proceder uma jogada: O sistema deve permitir que uma jogada se desenvolva da seguinte forma:

- Destacar qual é o jogador da vez.
- Permitir que o jogador selecione o golpe ou a ação pretendida .
- Validar a jogada, verificando se o oponente ainda tem vida disponível.

Requisito funcional 7 Decidir vencedor: A aplicação deve ser capaz de verificar se há um vencedor a cada turno finalizado conforme regras do jogo. Se houver, é decretado o fim da partida e a interface tabuleiro torna se inoperante para qualquer um dos jogadores

Requisito funcional 9 – Sair: O programa deve apresentar a opção de menu “*sair*”, fechando assim a janela e encerrando a partida. O sistema deve validar, caso a partida estiver em curso, se o usuário realmente deseja finalizar a partida.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 – Especificação de projeto: além do código Java, deve ser produzida especificação de projeto baseada em UML;

Requisito não funcional 2 – Interface gráfica com usuário: O software deverá apresentar uma interface gráfica bidimensional, que mostrará o conteúdo da Arena para todos os jogadores.

Requisito não funcional 4 - Atualização da interface gráfica:

■ A cada tentativa de jogada do usuário, o estado da partida deve mudar e ser exibido na interface a vida restante dos lutadores,

Requisito não funcional 3 – Tecnologia de interface gráfica para usuário: A interface gráfica deve ser baseada em *Java-Swing*.

4 Esboço da interface gráfica



