

Projeto Dofus

Especificação de Requisitos de Software

Versão 1.3

06/10/2020, Criação: 18/09/2020

Versão	Autor(es)	Data	Ação
1.0	Henrique da Cunha Buss	18/09/2020	Estabelecimento do documento inicial (Introdução, Visão Geral e Requisitos de Software)
1.1	Henrique da Cunha Buss	28/09/2020	Adicionado requisito funcional Selecionar Personagem
1.2	Henrique da Cunha Buss	01/10/2020	Adicionados requisitos funcionais referentes ao Netgames Server e melhoria da descrição do jogo
1.3	Henrique da Cunha Buss	06/10/2020	Adaptado conforme discutido na primeira apresentação

Conteúdo:

1. Introdução;
2. Visão Geral;
3. Requisitos de Software.

1. Introdução

1.1 Objetivo do Desenvolvimento

Este projeto visa desenvolver um programa inspirado nos jogos Dofus e Wakfu da empresa francesa Ankama, implementando seus sistemas de combate entre jogadores, de maneira mais simplificada, on-line.

1.2 Abreviaturas e Definições

Abreviatura	Significado
PA	Pontos de Ação
PM	Pontos de Movimento
PV	Pontos de Vida

1.2.1 Visão geral do jogo

O sistema de combates admite dois times, que irão batalhar um contra o outro, onde cada time é composto por 1 a 6 jogadores, sendo que cada jogador controla um personagem, e cada personagem é de uma classe. Cada classe possui um conjunto único de feitiços.

Os personagens se situam em um tabuleiro de 100 posições (dispostas em formato quadrangular, formando um tabuleiro 10x10), que pode ou não possuir obstáculo(s). Obstáculos representam células onde os jogadores não podem ocupar ou usar para se movimentar.

O combate se dá em formato de turnos, onde um jogador tem a sua vez, executa suas ações, termina seu turno, e o próximo jogador tem a sua vez. Os jogadores são postos em uma fila circular, de modo que a ordem dos turnos se mantém a cada ciclo que se completa (do primeiro jogador ao último). Durante seu turno, o jogador pode andar pelo tabuleiro e lançar feitiços, em qualquer ordem e combinação que desejar, ou seja, pode andar uma posição, lançar um feitiço, andar mais uma posição, lançar outro feitiço, etc. Ao terminar suas ações, o jogador deve clicar no botão de término de turno para passar a vez ao próximo jogador. O objetivo do jogo é matar todos os personagens do time inimigo. Mais detalhes são dados abaixo.

1.2.2 Personagens e PVs

Cada personagem tem um número base de PVs e um número de PVs atual, e cada personagem é de uma classe: Guerreiro ou Mago. O Guerreiro possui um número base de PVs maior do que o Mago. O número atual de PVs não pode ser maior do que o número base de PVs.

Um personagem é dito morto caso seu número atual de PVs seja menor ou igual a zero. Personagens mortos são retirados do tabuleiro (não ocupam uma posição, não têm mais um turno, e não podem interagir com o jogo ou com o tabuleiro) até o final da partida. O jogador com seu personagem morto pode assistir a partida, mas não interagir com ela.

1.2.3 Feitiços e PAs

Cada personagem tem um número base de PAs e um número de PAs atual. O número de PAs atual do personagem é restabelecido como o número base de PAs do personagem no início de cada turno.

Feitiços são lançados em posições do tabuleiro, estejam elas ocupadas por um jogador ou não, desde que a posição alvo não seja um obstáculo. Cada feitiço possui um alcance máximo, ou seja, a distância (em células do tabuleiro) do jogador ao alvo deve ser menor ou igual ao alcance do feitiço para que este possa ser lançado.

Cada feitiço possui um custo de PAs, que é gasto quando o feitiço é lançado. Um feitiço só pode ser lançado caso o número atual de PAs do jogador seja maior ou igual ao custo do feitiço, se a posição alvo está dentro do alcance do feitiço e a posição alvo não é um obstáculo.

Cada feitiço possui também um efeito e um valor, sendo que o efeito pode ser de dano ou de cura. Um feitiço de dano irá subtrair o número atual de PVs do jogador atingido pelo seu valor. Um feitiço de cura irá somar ao número atual de PVs do jogador atingido o seu valor.

1.2.4 Movimentação e PMs

Cada personagem tem um número base de PMs e um número de PMs atual. O número de PMs atual do personagem é restabelecido como o número base de PMs do personagem no início de cada turno.

Durante seu turno, o jogador pode andar pelo tabuleiro, desde que o número de posições andadas seja menor ou igual ao número atual de PMs. A cada posição andada (na vertical ou horizontal - diagonais custam dois PMs, já que são uma movimentação horizontal e uma vertical), o personagem perde um PM. Personagens não podem andar ou terminar seu movimento sobre uma posição que seja obstáculo.

1.2.5 Término de turno

Ao terminar suas ações, o jogador deve clicar no botão de término de turno, para que o próximo jogador possa ter seu turno. Se o jogador que terminou o turno é o último da fila, o próximo da vez é o primeiro que jogou, e depois o segundo, o terceiro, e assim por diante.

1.3 Referências

- <<https://www.dofus.com/pt>> (Site Dofus - não possui muita informação adicional, apenas por curiosidade)

2. Visão Geral

2.1 Arquitetura do Software

Programa orientado a objetos, em modelo cliente-servidor distribuído.

2.2 Premissas de Desenvolvimento

- O programa deve ser implementado em Java;
- O programa deve executar em qualquer plataforma que possua máquina virtual Java;
- O programa deve usar Netgames como suporte a execução distribuída;
- O projeto deve ser produzido com Visual Paradigm.

3. Requisitos de Software

3.1 Requisitos Funcionais

Requisito funcional 1 - Iniciar partida:

O programa deve apresentar a opção de menu ***“iniciar”***, para o início de uma nova partida, operação em que é definido a identificação dos jogadores e a ordem de turnos;

Requisito funcional 2 - Reiniciar partida:

O programa deve apresentar a opção de menu ***“reiniciar”***, para o início de uma nova partida, considerando as seguintes restrições:

- A operação mantém os jogadores já definidos;
- A operação mantém a ordem de turnos já definida, mas começa do início;

- Caso haja partida em andamento, o programa deve solicitar autorização para sua interrupção - neste caso, apenas com autorização dos usuários a operação pode prosseguir.

Requisito funcional 3 - Selecionar personagem:

O programa deve permitir que o usuário selecione um dos personagens disponíveis antes do início da partida, e o usuário só deve poder começar uma partida se tiver escolhido seu personagem.

Requisito funcional 4 - Fazer movimento:

O programa deve permitir que os jogadores possam se movimentar no tabuleiro, por meio de *click* de botão esquerdo do *mouse* sobre a posição em que o jogador deve ir, considerando as seguintes restrições:

- O jogador deve ser o jogador da vez;
- O número de células do tabuleiro percorridas não deve ser maior do que o número atual de PMs do jogador;
- A célula não pode estar ocupada por outro jogador;
- A célula não pode estar ocupada por um obstáculo.

Requisito funcional 5 - Selecionar feitiço:

O programa deve permitir que os jogadores selecionem um de seus feitiços, por meio de *click* de botão esquerdo do *mouse* sobre o feitiço na barra de feitiços. Ao selecionar o feitiço, deve ser mostrado no tabuleiro as posições onde ele pode ser usado.

Requisito funcional 6 - Lançar feitiço:

O programa deve permitir que os jogadores lancem seus feitiços, por meio de *click* de botão esquerdo do *mouse* na célula alvo quando o feitiço está selecionado, considerando as seguintes restrições:

- O jogador deve ser o jogador da vez;
- A célula alvo deve estar dentro do alcance do feitiço;
- A célula alvo não pode estar ocupada por um obstáculo;
- O jogador deve ter ao menos o número de PAs que o feitiço exige.

Requisito funcional 7 - Terminar turno:

O programa deve permitir que os jogadores possam terminar seu turno, por meio de *click* de botão esquerdo do *mouse* em um botão intitulado “**terminar turno**”, que só fica ativo caso o jogador seja o jogador da vez.

Requisito funcional 8 - Morte de personagem:

Ao ter seus PVs zerados, o personagem deve ser considerado morto, ou seja, deve sair da fila de turnos e não pode mais influenciar no tabuleiro. O jogador que tem seu personagem morto pode assistir o jogo, mas não pode interagir com ele.

Requisito funcional 9 - Comunicação do estado da partida:

A cada tentativa de movimento, lançamento de feitiço ou término de turno de um usuário (bem sucedida ou não), o novo estado da partida deve ser exibido na interface de todos os usuários, enviando as informações através do Netgames Server, considerando as seguintes restrições:

- Na tentativa bem sucedida de movimento, o tabuleiro deve ser atualizado com a nova posição do jogador, bem como o número restante de PMs do jogador;
- Na tentativa mal sucedida de movimento, o jogador deve ser comunicado o tipo de irregularidade: tentativa de movimento em célula ocupada (por outro jogador ou por obstáculo), PMs insuficientes, fora da sua vez, ou personagem morto;
- Na tentativa bem sucedida de lançamento de um feitiço, as informações dos personagens devem ser atualizadas, para refletir os efeitos do feitiço, bem como o número restante de PAs do jogador da vez;
- Na tentativa mal sucedida de lançamento de um feitiço, o jogador deve ser comunicado o tipo de irregularidade: tentativa de lançamento de feitiço em célula ocupada por obstáculo, PAs insuficientes, fora da sua vez ou personagem morto;
- Na tentativa bem sucedida de término de turno, os jogadores devem ser comunicados do novo jogador da vez;
- Na tentativa mal sucedida de término de turno, o jogador deve ser comunicado o tipo de irregularidade: jogador fora do turno ou personagem morto;

Requisito funcional 10 - Conectar ao servidor:

Os jogadores devem poder se conectar ao Netgames Server para transmitir e receber dados do jogo.

Requisito funcional 11 - Desconectar do servidor:

Os jogadores devem poder se desconectar do Netgames Server, encerrando a comunicação de dados do jogo.

Requisito funcional 12 - Receber determinação de início:

Cada jogador deve poder receber a determinação de início de partida, dada quando um jogador inicia uma partida e o servidor permite que a partida se inicie. Para que o jogador possa participar de uma partida, ele já deve ter escolhido seu personagem.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 - Especificação de projeto:

Além do código Java, deve ser produzida a especificação de projeto baseada em UML, segunda versão.

Requisito não funcional 2 - Interface gráfica para usuário:

O programa deverá ter interface gráfica única, compartilhada pelos usuários.

Requisito não funcional 3 - Representação das posições:

Uma posição do tabuleiro pode ser ocupada por um jogador, um obstáculo, ou estar vazia, e será representada da seguinte maneira:

- Fundo transparente, sem símbolos, no caso de estar vazia;
- Fundo vermelho, sem símbolos, no caso de estar ocupada por um obstáculo;
- Fundo transparente, com símbolo correspondente ao jogador, no caso de estar ocupada por um jogador.

Requisito não funcional 4 - Representação dos jogadores:

Cada jogador deve ter uma representação única, e para isso, cada jogador é representado por um número, equivalente à sua posição na fila de turnos. O jogador da vez deve ser representado usando outra cor, para ser destacado. Jogadores mortos não devem ser representados.