

Especificação de Requisitos

Projeto Yu-Gi-Oh

Especificação de Requisitos de Software

Versão 1.2
23/07/2014

Versão	Autoras	Data	Ação
1.0	Bruna Sudoski Stéphanie Leal	28/04/2014	Estabelecimento dos requisitos
1.1	Bruna Sudoski Stéphanie Leal	12/05/2014	Correção: Requisitos de Software
1.2	Stéphanie Leal	23/07/2014	Correção: Regras do Jogo

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 do jogo Yu-Gi-Oh na modalidade usuário contra usuário em execução distribuída.

1.2 Regras do jogo

1.2.1 O objetivo:

Cada jogador inicia com uma quantidade de 8000 pontos de vida, onde o objetivo é que cada um tente zerar os pontos de vida do adversário e defender os seus próprios, utilizando um baralho próprio pra cada um. O jogador perde o jogo quando seus pontos de vida são zerados ou quando acabam suas cartas.

1.2.2 As cartas:

1.2.2.1 Tipo Monstros

As cartas de monstro são utilizadas para defender os pontos de vida do jogador, bem como para atacar os pontos de vida do jogador adversário.

Cada monstro possui dois valores: de ataque (utilizado para atacar os monstros do adversário, bem como seus pontos de vida) e de defesa (utilizado para defender os pontos de vida do jogador). Durante o jogo, os monstros podem ser colocados em campo em dois modos, o modo de ataque (carta colocada na vertical), onde o monstro poderá atacar e defender utilizando seus pontos de ataque, e o modo de defesa (carta colocada na horizontal), onde o monstro poderá apenas defender utilizando seus pontos de defesa. O jogador pode escolher colocar sua carta monstro virada para baixo ou para cima tanto em modo de defesa como de ataque (estratégia para esconder a carta).

1.2.3 O campo

Cada jogador possui um campo próprio para utilizar suas cartas, onde cada um contém cinco espaços a serem utilizados exclusivamente por monstros, um espaço para o baralho próprio e um espaço para uma pilha de cartas descartadas chamada de cemitério. Sendo assim, como um todo, o campo dispõe de vinte e quatro espaços, doze para cada jogador.

1.2.4 O jogo

Cada jogador possui um baralho próprio, chamado de “deck”, a ser escolhido no início da partida, onde cada um contém 40 cartas.

Cada rodada é composta por fases específicas, e cada rodada trará sempre o mesmo leque de fases a serem executadas, com exceção da primeira rodada, que disponibilizará a execução de apenas algumas fases específicas e diferenciadas.

O jogador que realizar a primeira jogada da partida terá que executar as seguintes fases:

- Fase de Invocação: Colocar um monstro em campo, em qualquer modo escolhido pelo jogador;
- Fase de Finalização: O jogador terminará sua partida realizando esta fase, e passando a vez para o oponente.

As demais rodadas serão compostas pelas seguintes fases:

- Fase de Invocação: Colocar um monstro em campo escolhendo em que modo o monstro ficará durante aquela rodada.
- Fase Principal: O jogador pode alterar o modo em que o monstro se encontra, de ataque para defesa e vice-versa, o jogador pode também escolher um** monstro próprio, que esteja em modo de ataque, para atacar um monstro específico do oponente, ou seus pontos de dano, caso o adversário não possua monstros em campo. (** O procedimento só pode ser executado uma vez por monstro naquela rodada).
- Fase de Finalização: fase que finaliza o lance e passará a vez para o jogador oponente.

A compra de cartas do deck é obrigatória em uma rodada, e será sempre a primeira ação a ser executada automaticamente pelo jogo. Além desta ação a Fase de Finalização, que termina a rodada do jogador, precisa obrigatoriamente ser executada independente de o jogador ter realizado a Fase Principal ou não.

1.2.5 Determinando o dano

O cálculo de danos é realizado diferentemente, dependendo do tipo de situação em que os monstros podem se encontrar.

Situação 1. Quando o Monstro do Oponente está em Modo de Ataque

Ao atacar um monstro que está em Modo de Ataque (na vertical), comparam-se os pontos de ataque de ambos os monstros.

Situação 1.1: Pontos de ataque do Atacante > Pontos de ataque do Oponente

Quando os pontos de ataque do monstro atacante são superiores aos pontos de ataque do monstro do oponente, o monstro do oponente é destruído e esta carta é colocada no cemitério.

Subtrai-se os pontos de ataque do monstro do oponente do total de pontos de ataque do monstro atacante. O resultado é subtraído dos Pontos de Vida do oponente.

Situação 1.2: Pontos de ataque do Atacante = Pontos de ataque do Oponente

Quando os pontos de ataque do monstro atacante são iguais aos pontos de ataque do monstro do oponente, o resultado é considerado um empate e ambos os monstros são destruídos e ambas as cartas são colocadas no cemitério.

Nenhum jogador sofre qualquer dano. Os seus Pontos de Vida permanecem os mesmos.

Situação 1.3: Pontos de ataque do Atacante < Pontos de ataque do Oponente

Quando os pontos de ataque do monstro atacante são inferiores aos pontos de ataque do monstro do oponente, o monstro atacante é destruído e a carta enviada para ao cemitério.

Subtrai-se os pontos de ataque do monstro atacante do total de pontos de ataque do monstro do oponente. O resultado é subtraído dos Pontos de Vida do jogador atacante.

Situação 2: Quando o Monstro do Oponente está em Posição de Defesa

Ao atacar um monstro que está em Posição de Defesa (na horizontal), compara-se o ataque do monstro atacante com a defesa do monstro do oponente.

Situação 2.1: Pontos de ataque do Atacante > Pontos de defesa do Oponente

Quando os pontos de ataque do monstro atacante são superiores aos pontos de defesa do monstro do oponente, o monstro do oponente é destruído e sua carta enviada para o cemitério.

Nenhum jogador sofre qualquer dano. Os seus Pontos de Vida permanecem os mesmos.

Situação 2.2: Pontos de ataque do Atacante = Pontos de defesa do Oponente

Quando os pontos de ataque do monstro atacante são iguais aos pontos de defesa do monstro do oponente, nenhum dos monstros é destruído.

Nenhum jogador sofre qualquer dano. Os seus Pontos de Vida permanecem os mesmos.

Situação 2.3: Pontos de ataque do Atacante < Pontos de defesa do Oponente

Quando os pontos de ataque do monstro atacante são inferiores aos pontos de defesa do monstro do oponente, nenhum dos monstros é destruído.

Subtrai-se os pontos de ataque do monstro atacante do total de pontos de defesa do monstro do oponente. O resultado é subtraído dos Pontos de Vida do jogador atacante.

Situação 3: Dano Direto: Quando o oponente não tem monstros

Se o oponente não tiver monstros na arena, ele recebe Dano Direto. O total dos pontos de ataque do monstro atacante é subtraído dos Pontos de Vida do oponente.

Notas:

- Nenhum monstro colocado em campo, ou que tenha sido descartado poderá voltar pras mãos do jogador.

Referências:

<http://pt.wikipedia.org/wiki/Yu-Gi-Oh!_Trading_Card_Game > (história).

<<http://regras.net/jogo-yu-gi-oh/>> (regras originais do jogo)

2 Visão Geral

Arquitetura do programa: programa orientado a objetos escrito na linguagem Java, utiliza o framework NetGamesNRT para conexão com servidor.

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.
- O programa deve possuir uma execução distribuída com suporte para a aplicação distribuída, NetGamesNRT, para a interação entre os jogadores.

3 Requisitos de Software

3.1 Requisitos Funcionais:

Requisito funcional 1 – Conectar: o programa deve possuir a opção “conectar” para obter uma conexão com o servidor efetuando a interação entre os jogadores.

Requisito funcional 2 – Definição de jogadores: o programa deve permitir a identificação dos jogadores antes do início da partida.

Requisito funcional 3 – Iniciar: o programa deve apresentar a opção “iniciar” para o início de uma nova partida.

Requisito funcional 4 – Campo de batalha: programa deve possuir uma tela do campo de batalha bem como permitir a visualização das cartas que estão em campo do jogador 1 e do jogador 2.

Requisito funcional 5 – Definição do baralho: o programa deve escolher automaticamente o baralho do jogador.

Requisito funcional 6 – Procedimento de lance: o programa deve permitir que o jogador da rodada escolha uma carta que está em sua mão para colocá-la no campo, considerando as seguintes restrições:

- O programa deve permitir que o jogador escolha o modo em que sua carta irá para o campo (Ataque ou Defesa).
- O programa deve permitir que o jogador escolha uma carta que está no seu campo para atacar uma carta do adversário.
- O programa deve colocar a(s) carta(s) destruída(s) no cemitério.

Requisito funcional 7 – Comunicação do estado da partida: Para cada procedimento de lance dos jogadores o novo estado da partida deve ser exibido na tela, considerando as seguintes restrições:

- Ao término de cada lance o programa deve colocar a(s) carta(s) destruída(s) no cemitério.
- O programa deve atualizar o placar a cada jogada.
- O programa deve mostrar a fase em que o oponente se encontra.
- O programa deve identificar o jogador vencedor.

Requisito funcional 8 – Desconectar: o programa deve possuir a opção “desconectar”, caso o jogador queira desistir da batalha ou fechar o programa, se desconectando do servidor.

3.2 Requisitos Não Funcionais

Requisito não funcional 1 – Linguagem do programa: o código deve ser escrito na linguagem Java.

Requisito não funcional 2 – Interface gráfica para usuário: o programa deve possuir interface gráfica única, partilhada pelos usuários.

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

Requisito não funcional 4 – Estabelecimento de quem inicia: o primeiro jogador que clicar na opção “iniciar” recebe a identificação de primeiro jogador e iniciará a partida.

4 Esboço da Interface Gráfica

Tela inicial:



Tela Principal:

