



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

DIGIMON CARD BATTLE

Heithor Simões Marques
Higor Nocetti
Sadi Júnior Domingos Jacinto

Professor orientador: Ricardo Pereira e Silva

Florianópolis

2018

Heithor Simões Marques
Higor Nocetti
Sadi Júnior Domingos Jacinto

DIGIMON CARD BATTLE

Análise e modelagem iniciais requeridos pelo professor da disciplina Análise e Projeto de Sistemas, Ricardo Pereira e Silva, necessário para obtenção de nota.

Professor orientador: Ricardo Pereira e Silva

Florianópolis

2018

ESPECIFICAÇÕES DE REQUISITOS DE SOFTWARE

Versão	Autor(es)	Data	Ação
1.0	Heithor Simões Marques Higor Nocetti Sadi Júnior Domingos Jacinto	25/08/2018	Análise e estabelecimento iniciais dos requisitos

Sumário

1	INTRODUÇÃO	3
1.1	OBJETIVOS	3
1.2	O JOGO	3
1.2.1	REGRAS	3
1.2.2	FUNCIONAMENTO DO JOGO	4
2	VISÃO GERAL	5
2.1	ARQUITETURA DO PROGRAMA	5
2.2	PREMISSAS DE DESENVOLVIMENTO	5
3	REQUISITOS DE SOFTWARE	6
3.1	REQUISITOS FUNCIONAIS	6
3.2	REQUISITOS NÃO FUNCIONAIS	6

1 INTRODUÇÃO

1.1 OBJETIVOS

Desenvolvimento de um programa que suporta emulação virtual distribuída em rede, permitindo a dois usuários disputarem uma partida do jogo Digimon Card Battle...

1.2 O JOGO

Trata-se de um jogo de cartas, onde o objetivo principal é ser o primeiro a derrotar 03(três) Digimon Cards do adversário.

1.2.1 REGRAS

- Cada jogador deve ter um deck (baralho) com 30 cartas.
- As cartas podem ser de “Option Cards” ou de “Digimon Cards”.
- As Digimon Cards possuem 10 atributos, sendo eles:
 1. HP
É a vida da carta;
 2. Ataque 1;
 3. Ataque 2;
 4. Ataque 3;
 5. Efeito do terceiro ataque;
 6. DP (Digivolve Points)
É a quantidade de pontos que você precisa para poder usá-la no campo de batalha com 100% do potencial;
 7. +P
A quantidade de pontos que a carta adiciona aos DP;
 8. Especialidade da carta
Pode ser água, fogo ou grama;
 9. Nível
R, C ou U
 - Cartas do nível R utilizam 100% de seu potencial em batalha e não precisam de DP;
 - Cartas do nível C utilizam 50% de seu potencial em batalha se não forem usadas via Evolução;
 - Cartas do nível U utilizam 25% de seu potencial em batalha se não forem usadas via Evolução;
 10. Efeito da carta
Caso seja usada como suporte.
- Nenhum jogador pode ter mais que 4 cartas na mão.
- Option Cards tem apenas o atributo Efeito da Carta.

1.2.2 FUNCIONAMENTO DO JOGO

A partida é composta de 3 fases:

1. Fase de compra:

Nessa fase o jogador compra cartas do deck até que sua mão tenha 4 cartas.

Se você não gostar das cartas, o jogador pode descartar toda a sua mão para o cemitério e comprar até ter 4 cartas novamente em sua mão. Não é permitido o descarte de um número de cartas menor que 4, sendo necessário o descarte da mão inteira, mesmo que apenas para trocar uma única carta.

Após a compra, se o jogador estiver satisfeito com suas cartas e não possuir nenhum Digimon Card no campo de batalha, o jogador escolhe um Digimon Card para pôr no campo de batalha,

Caso já haja um Digimon Card inicia-se a fase 2.

2. Fase de evolução:

Nessa fase o jogador pode sacrificar um Digimon Card para acumular DP.

Se o jogador já tiver acumulado DP suficiente, o mesmo pode evoluir o Digimon no campo de batalha para um que esteja na sua mão.

Todos os pontos de DP são consumidos, mesmo que o jogador tenha mais que o suficiente.

A ordem de evolução é: de R para C e de C para U, não sendo transitivo.

Também é necessário que a Especialidade das cartas sejam as mesmas.

3. Fase de Batalha:

Nessa fase é onde o combate ocorre.

Se o adversário não possuir algum Digimon no campo de batalha, esse turno é passado.

Se ambos os jogadores possuírem Digimons no campo de batalha, cada um escolherá um ataque.

Após escolherem os ataques, ambos tem a opção de usar uma carta como suporte podendo ser uma Option Card ou um Digimon Card. O adversário escolhe primeiro a carta de suporte e depois o jogador atual do turno.

A seguir, as seguintes ações são realizadas, na ordem apresentada:

- (a) A carta de suporte do jogador atual do turno tem seu efeito;
- (b) A carta de suporte do adversário tem seu efeito;
- (c) Efeito do ataque 3 do jogador atual do turno, se ele tiver selecionado esse ataque;
- (d) Efeito do ataque 3 do adversário, se ele tiver selecionado esse ataque;
- (e) Ataque do jogador atual do turno;
- (f) Ataque do adversário.

O Digimon adversário é derrotado quando seu HP chega a 0.

Ao fim da fase de batalha, é o turno do adversário e tudo se repete.

2 VISÃO GERAL

2.1 ARQUITETURA DO PROGRAMA

Programa escrito em linguagem que segue o paradigma de Orientação à Objetos.

2.2 PREMISSAS DE DESENVOLVIMENTO

O programa deve:

- Ser implementado em linguagem *Java*, devendo executar em qualquer plataforma que disponha da máquina virtual *Java*;
- A aplicação deverá suportar rede, através de arquitetura cliente/servidor, fazendo uso da ferramenta *NetGamesNRT*, permitindo, dessa forma, uma aplicação distribuída.
- A aplicação deve apresentar uma interface gráfica, única e distribuída para os usuários.

3 REQUISITOS DE SOFTWARE

3.1 REQUISITOS FUNCIONAIS

1. **Conectar:**

O software deve apresentar em seu menu a opção “Conectar”, para, assim, estabelecer conexão com o servidor *NetGames*.

2. **Desconectar:**

O jogo deve apresentar a opção de menu “Desconectar”, permitindo se desconectar do servidor, encerrando assim uma possível partida em andamento.

3. **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 do(s) jogador(es).

4. **Abandonar partida:**

O programa deve apresentar a opção de menu “Fugir” para encerrar a partida atual e voltar para o menu principal.

5. **Realizar jogada/lance:**

O programa deve permitir aos jogadores, através do uso do mouse, realizar suas jogadas, em seus respectivos turnos, desde que obedecidas as regras do jogo.

6. **Informações do estado da partida:**

O programa deve apresentar aos jogadores, através de uma interface gráfica, informações sobre a partida atual, incluindo:

- Se existe um vencedor.
- O nome do jogador vencedor, caso o mesmo exista./
- Alertas de jogadas inválidas (fora do turno, tentar comprar do baralho quando a mão já está completa, ...)
- O nome do jogador cujo turno é o atual.

3.2 REQUISITOS NÃO FUNCIONAIS

1. **Especificação de projeto:**

Código desenvolvido em linguagem *Java*, além de ser produzida especificação de projeto baseada na segunda versão da linguagem de modelagem *UML*.

2. **Interface gráfica para usuário:**

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

3. **Execução Distribuída:**

A aplicação deverá suportar rede, através de arquitetura cliente/servidor, fazendo uso da ferramenta *NetGamesNRT*, permitindo, dessa forma, uma aplicação distribuída.