Especificação de Requisitos

Yaniv

09/04/2024

|  |  |  |  |
| --- | --- | --- | --- |
| Versão | Autores | Data | Ação |
| 1.0 | Eduardo Gwoszdz, Micael Angelo, Brendo Alves | 13/04/2024 | Estabelecimento dos requisitos |

Conteúdo

1. Introdução

1.1. Objetivo

1.2. Definições e abreviações

1.3. Referência

2. Visão Geral

2.1 Arquitetura do Jogo

2.2 Premissas de desenvolvimento

3. Regras do Jogo

3.1 Definições

3.2 Pontuação

3.3 Pre-Round

3.4 In-Round

3.5 Pos-Round

4. Requisitos de software

4.1. Requisitos funcionais

4.2. Requisitos não-funcionais

1 Introdução

1.1 Objetivo

Desenvolvimento de um jogo que comporta 4 jogadores, cada um em sua máquina, se enfrentarem em uma partida de Yaniv, na modalidade JxJ (Jogador contra Jogador).

1.2 Definições e Abreviações

* Abreviações

MC – Monte de Compras

MD – Monte de Descarte

RF – Requisitos funcionais

RNF – Requisitos não funcionais

* Definições:

Descrito no tópico 3.1, Regras do Jogo

1.3 Referencia

Yaniv (jogo) < <https://gamerules.com/rules/jhyap-card-game/> >

2 Visão Geral

2.1 Arquitetura do Jogo

Se baseia em um sistema distribuído onde há uma divisão clara de funções entre o cliente, que solicita serviços ou recursos, e o servidor, que fornece esses serviços ou recursos, permitindo a comunicação e interação entre eles através de uma rede.

2.2 Premissas de desenvolvimento

* Implementação em Python é requerida para o programa.
* Utilização do DOG para facilitar a execução distribuída do programa.
* Além da codificação, é necessário criar uma segunda versão da especificação do projeto, baseada em UML.

3 Regras do Jogo

3.1 Definições

O jogo é realizado com um baralho com 54 cartas (conjuntos de A, 2, 3 ..., Q, K. com 4 naipes (Copas, Ouros, Paus e Espadas) mais 2 coringas.

O jogo possui rotação entre turno para cada jogador. Sendo possíveis ações de compras, descartes e Yaniv. Possuindo 2 Montes de cartas na mesa, 1 MC e 1 MD, sendo o de compras realizado com o restante das cartas após serem distribuídas aos jogadores e o MD será acessível somente em relação ao descarte do último jogador.

3.2 Pontuação

As cartas possuem valores, sendo esses, distribuídos com os seguintes valores:

Ases = 1

Números = Seus próprios valores (2 de Copas = 2 | 3 de Espadas = 3 ...)

Família Real = 10 (J de Espadas = 10, Q de Copas = 10, K de Ouros = 10 ...)

Coringas = 0

* 1. Pre-round

Antes do início da partida, é aguardado os jogadores se conectarem para se iniciar a partia, sendo essa, possível de iniciar somente com a quantidade de 4 jogadores, após todos conectados, é possível iniciar a partida.

* 1. In-round

Ao iniciar o jogo, todos os jogadores recebem cinco cartas aleatórias do baralho, em seguida, as cartas restantes do baralho ficam disponível para a compra no MC (cartas ficam ocultas até algum jogador realizar a compra) e em seguida uma carta do MC será aberta e adicionada ao MD para iniciar a partida com a possibilidade de escolher onde comprar, no MC ou MD.

O primeiro jogador irá fazer uma compra em um dos montes e em seguida realizar o descarte de uma carta qualquer ou uma combinação de cartas, sendo essas combinações descritas abaixo:

* **Combinação 1 – Sequencia:**

Essa combinação é possível tendo em sua mão uma sequência de pelo menos 3 cartas com valores sucessivos, contínuos, não rotativo e de mesmo naipe (Espadas, Copas, Paus ou Ouros). Exemplos:

Exemplo 1: 2 de Copas, 3 de Copas e 4 de Copas.

Exemplo 2: 10 de Espadas, J de Espadas, Q de Espadas e K de Espadas.

Exemplo 3: Q de Copas, Coringa, K de Copas.

Exemplo 4 (sequencia invalida, pois é rotativa): Q de Ouros, K de Ouros, Ás de Ouros.

* **Combinação 2 – Conjunto:**

Essa combinação é possível tendo em sua mão um conjunto de pelo menos 2 cartas de mesmo valor e consequentemente de naipes diferentes. Exemplos.

Exemplo 1: 5 de Paus e 5 de Copas.

Exemplo 2: 7 de Copas, 7 de Espadas e 7 de Ouros.

Exemplo 3: Coringa e K de Ouros.

Exemplo 4 (Conjunto invalido, pois é de naipes diferentes): 4 de Ouros e 4 de Copas.

Após o primeiro jogador realizar sua compra e seu descarte, o próximo jogador ficara disponível para realizar compras do MC ou MD e em seguida descartar uma carta qualquer ou uma das combinações acima descritas, e assim por diante para cada jogador.

Cada jogador, quando estiver na sua vez de jogar, é obrigado realizar uma compra de um dos montes de cartas, MC ou MD.

* **Compra no MC:**

Uma carta é retirada do monte de compras e será adicionada a mão de cartas do jogador.

* **Compra no MD:**

Somente a última carta do descarte realizado pelo último jogador ficará disponível no monte de descarte para ser comprado e será adicionada a mão de cartas do jogador.

Após um jogador alcançar em sua mão de cartas um valor igual ou inferior a 6 pontos para chamar **YANIV** (disponível após obter a pontuação mínima necessária). Sendo esse o vencedor dessa partida, sendo assim, é adicionado os pontos (pontuação somada dos valores das cartas atuais em sua mão) de cada jogador em seus respectivos placares.

Porém, caso outro jogador possuir uma pontuação inferior ao jogador que chamou Yaniv, esse mesmo jogador que chamou Yaniv receberá uma penalidade de 30 pontos e os demais será adicionado o valor atual de sua mão de cartas ao placar.

* 1. Pos-round

Ao final da partida, após os valores das pontuações serem adicionados aos placares de cada jogador, a partida é reiniciada, porém, com a pontuação dos placares atualizadas e repetirá as mesmas ações até que um jogador alcance a pontuação igual ou superior a 100 pontos.

A pontuação máxima é o que determina o final da partida, no instante que algum jogador alcançar tal feito, será verificado qual jogador possui a menor pontuação no placar e esse será considerado o vencedor da partida.

Caso ocorra empate, as posições no ranque poderão se repetir, podendo haver múltiplas colocações.

1. Requisitos de Software

4.1 Requisitos funcionais

RF 1. **Iniciar Programa:** Ao ser executado o programa deve aparecer um menu aguardando os quatro jogadores através do Dog Server (utilizando recursos de Dog). Apenas em caso de conexão bem sucedida as demais funcionalidades estarão habilitadas. No caso de conexão mal sucedida, a alternativa é encerrar o jogo.

RF 2. **Iniciar Jogo:** O programa deve apresentar a opção de menu “start game” para o início de uma nova partida. O procedimento de início de partida consiste em enviar uma solicitação de início a Dog Server, que retornará o resultado, que será a notificação, em caso de êxito, ou a razão da impossibilidade de início de partida, caso contrário. A interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de lance. Esta funcionalidade só deve estar habilitada se o programa estiver em seu estado inicial, isto é, sem partida em andamento e com a mesa em seu estado inicial.

RF 3. **Restaurar Estado Inicial do jogo**: O programa deve apresentar a opção “restaurar estado inicial” para levar o programa ao seu estado inicial, isto é, sem partida em andamento e com a interface na fase inicial. Esta funcionalidade só deve estar habilitada se o programa estiver com o jogo finalizado.

RF 4. **Selecionar Monte de Compras**: O jogador deve selecionar o MC se desejar comprar sua carta, através do “Buy mount”, a carta selecionada é destacada visualmente na interface do programa e em seguida é adicionada a mão do jogador. O próximo passo é aguardar o jogador a descartar as cartas (RF 6. Selecionar Cartas).

RF 5. **Selecionar Monte de Descarte:** Similar ao “RF 4. Selecionar Monte de Compras. Se o jogador optar por adquirir a carta do monte de descarte, a carta é destacada visualmente e poderá ser adquirida através do “Discard mount” e em seguida é adicionada a mão do jogador. O próximo passo é aguardar o jogador a descartar cartas (RF 6. Selecionar Cartas)

RF 6. **Selecionar Cartas:** O jogador deve selecionar uma ou mais cartas desejadas para descartar uma combinação de cartas a serem despejadas no MD, as cartas serão destacadas visualmente e poderá ser descartada através de um botão “Discard”. Em seguida é necessário esperar os demais jogadores em seus respectivos turnos.

RF 7. **Notificação de início de partida:** O programa notificará assim que houver uma solicitação de início de partida, realizada pelo Dog Server, em função da solicitação de início de partida por parte de outro jogador conectado ao servidor. A interface irá atualizar para receber ao receber tais informações, porém, somente o jogador local poderá iniciar a partida.

RF 8. **Receber Jogada:** O programa receberá uma jogada do adversário, enviada por Dog Server, quando for a vez do adversário do jogador local. A jogada recebida deve ser um lance regular e conter as informações especificadas para o envio de jogadas, como por exemplo, RF 4, RF 5 e RF 6.

RF 9: **Receber notificação de abandono:** O programa notificará assim que um jogador abandonar a partida, por parte de um adversário remoto, enviada por Dog Server. Nesse caso os jogadores serão notificados do abandono e consequentemente a execução se encerra para todos os jogadores restantes.

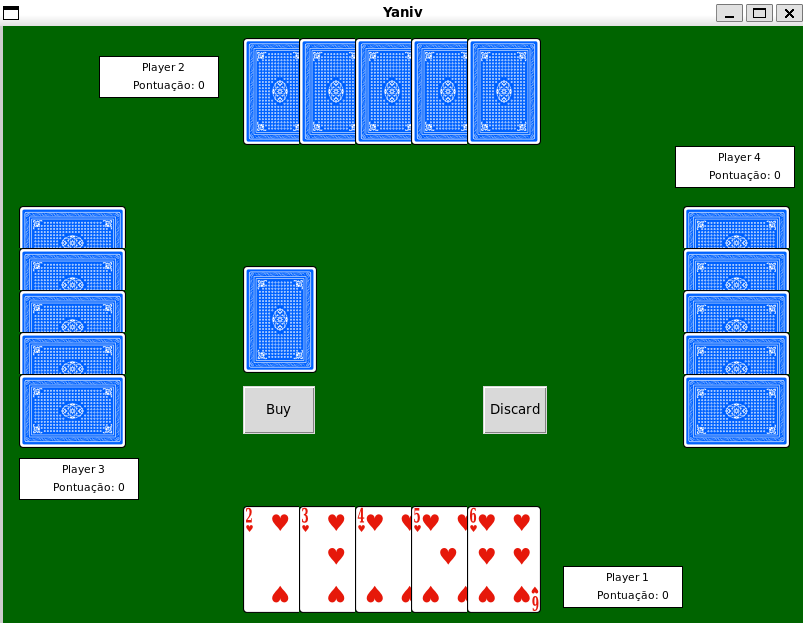
RF 10. **Receber End Game**: O programa deve interagir com os demais usuários quando um dos jogadores pressionar o botão Yaniv, o que determina o final do Round. Após a chamada o programa verifica os valores de pontuação de cada jogador, determinando se está dentro das regras. Em seguida os placares são atualizados e caso termine o jogo é informado o campeão, ranqueado os jogadores.

4.2 Requisitos não funcionais

RNF 1. **Especificação de Projeto**: O código de desenvolvimento será Python na versão 3.10 e juntamente com a especificação de projeto baseada em UML com a ferramenta Visual Paradigm.

RNF 2. **Especificação de Projeto**: O funcionamento distribuído é realizado pelo framework DOG.

RNF 3. **Especificação de Projeto:** A interface gráfica do programa é baseada na biblioteca em TKinter.

RNF 4. **A Interface do programa:** A interface será produzida conforme o esboço da imagem abaixo: