

Faculdade de Engenharia da Universidade do Porto



Laboratório de Programação
Orientada por Objetos

Aplicação Distribuída
POKER

Turma: 2MIEIC01

Grupo de Projeto: 5

Autores:

José Luís Pacheco Martins (up201404189@fe.up.pt)

Marcelo Diocleciano RodriguesFerreira(up201405323@fe.up.pt)

Índice de figuras

- Figura 1 - Imagem inicial do servidor
- Figura 2 - waiting players
- Figura 3 - Imagem de jogo, onde se assinala o jogador que vai jogar
- Figura 4 - Imagem de jogo, onde um jogador fez fold
- Figura 5 - imagem de jogo, onde foi encontrado o vencedor
- Figura 6 - Imagem inicial da aplicação para android
- Figura 7 - Menu About
- Figura 8 - Menu Stats
- Figura 9 - Não tem conta, para se conectar
- Figura 10 - Não tem dinheiro suficiente na sua conta, para se conectar
- Figura 11 - Utiliza a sua conta, para se conectar
- Figura 12 - Erro ao conectar a ronda já está em progresso
- Figura 13 - A conectar
- Figura 14 - Imagem de jogo, ainda não entrou na ronda
- Figura 15 - Imagem de jogo no android
- Figura 16 - Imagem no dispositivo android depois de se saber o vencedor
- Figura 17 - Diagrama de packages do Client
- Figura 18 - Diagrama de packages do Server
- Figura 19 - Diagrama de classes do package Network do Client
- Figura 20 - Diagrama de classes do package GUI do Client
- Figura 21 - Diagrama de classes do package Logic do Client
- Figura 22 - Diagrama de classes do package GUI do Server
- Figura 23 - Diagrama de classes do package Network do Server
- Figura 24 - Diagrama de classes do package Test do Server
- Figura 25 - Diagrama de classes do package Logic do Server
- Figura 26 - Diagrama de classes do package Common

Índice

- [1.Introdução](#)
- [2.Manual de Utilização](#)
 - [2.1.Resumo das funcionalidades suportadas](#)
 - [2.2.Instalação e arranque do programa](#)
 - [2.3.Modos de Utilização](#)
 - [2.3.1 Server-side](#)
 - [2.3.2 Client-side](#)
 - [2.4.Ficheiros de entrada e saída](#)
- [3.Conceção e Implementação](#)
 - [3.1 Estrutura de Packages](#)
 - [3.1.1 Estrutura de packages do Client](#)
 - [3.1.2 Estrutura de Packages do Server](#)
 - [3.2 Diagramas de Classes](#)
 - [3.2.1 Package Network do Client](#)
 - [3.2.2 Package GUI do Cliente](#)
 - [3.2.3 Package Logic do Client](#)
 - [3.2.4 Package GUI do Server](#)
 - [3.2.5 Package Network do server](#)
 - [3.2.6 Package Test do server](#)
 - [3.2.7 Package Logic do server](#)
 - [3.2.8 Package Common \(Server e Client\)](#)
 - [3.3 Padrões de desenho utilizados](#)
 - [3.4 Dificuldades encontradas](#)
 - [3.5 Bibliotecas e ferramentas utilizadas](#)
- [4.Testes Realizados](#)
- [Conclusão](#)

1.Introdução

No âmbito do segundo projeto da unidade curricular Laboratório de Programação Orientada por Objetos, foi desenvolvida uma aplicação distribuída que permite jogar Poker.

A aplicação baseia-se numa arquitetura cliente-servidor, em que a aplicação cliente foi desenvolvido para android e a aplicação servidora foi desenvolvida para desktop.

Para a comunicação foi usada a biblioteca LipeRMI.

Numa pequena introdução, o servidor é responsável por mostrar a mesa com todos os jogadores, dar as cartas, avaliar os vencedores, receber/tratar todas as notificações dos clientes assim como notificar os clientes aquando da sua vez, no outro lado o cliente disponibiliza uma interface amigável e intuitiva que permite enviar as suas ações para o servidor.

Este relatório tem como principal objetivo mostrar o funcionamento da aplicação desenvolvida e os detalhes da sua implementação..

Inicialmente será feita uma descrição detalhada de como utilizar a aplicação e do seu respetivo funcionamento, depois serão abordados os detalhes da implementação recorrendo a diagramas de packages/classes, bibliotecas utilizadas, testes realizados e padrões de desenho utilizados e por fim serão dadas algumas considerações finais em relação ao desenvolvimento do projeto.

2.Manual de Utilização

2.1.Resumo das funcionalidades suportadas

O projeto permite ligar até seis dispositivos android á aplicação servidora no desktop.

O servidor espera até que pelo menos 2 jogadores estejam conectados ao servidor e tenham notificado o servidor que estão prontos para jogar, a partir deste momento o servidor mostra a mesa de jogo assim como as informações de cada utilizador e dá as cartas a cada jogador, depois os jogadores vão sendo notificados pela devida ordem seguindo as regras do Texas Hold'em Poker.

No fim de cada etapa da ronda o servidor mostra as cartas do respetivo (Flop, Turn e River) na mesa pela respetiva ordem e repete o processo de notificação dos jogadores, no fim da ronda o servidor calcula os vencedores mostrando-os na mesa e notificando também todos os clientes com a informação da sua vitória ou derrota.

A aplicação android depois de se conectar ao servidor e ter notificado o servidor que quer jogar espera que as condições para o inicio do jogo estejam reunidas depois quando chega a sua vez de jogar é mostrado o respetivo tempo que o utilizador ainda tem para jogar e o utilizador pode escolher a ação desejada(Raise,Fold,All-in,Call ou Check) e envia-la ao servidor.

A aplicação também mostra as informações relativas ao jogador como o seu dinheiro, nome e a aposta atual.

Quando a ronda termina são mostradas do total das 7 cartas (5 da mesa e as duas privadas do jogador) as 5 cartas que constituem a melhor mão do jogador e a informação que indica se este ganhou ou perdeu, bem como o tipo de mão que conseguiu.

Depois é possível notificar novamente o servidor da nossa prontidão para iniciar uma nova ronda e o processo repete-se.

A aplicação android atualiza e guarda automaticamente estatísticas e informações do jogador durante a execução do jogo num ficheiro, estas informações são por exemplo o nome, dinheiro, o valor máximo do pot ganho até ao momento numa ronda entre outras, desta forma se o jogador já se tiver conectado alguma vez ao servidor não precisa de escolher o nome pode usar as informações da sua conta existente (nome e dinheiro) para se conectar ao servidor.

De notar que a aplicação também guarda o seu estado, quando é bloqueada,para desta forma se o utilizador bloquear a aplicação o tempos continuar a correr

2.2.Instalação e arranque do programa

A aplicação do servidor pode ser arrancada com o batch file fornecido na pasta do projeto com o nome “server.bat”.

A aplicação do cliente pode ser instalada através da apk fornecida em qualquer dispositivo android.

2.3. Modo de Utilização

2.3.1 Server-side

Em primeira instancia, quando se inicializa o servidor é mostrado o seguinte ecrã.



Figura 1 - Imagem inicial do servidor

A partir deste momento os jogadores conseguem conectar-se ao servidor. Quando alguém se conecta ao servidor o ecrã muda aparecendo a informação do jogador conectado e se o jogador estiver a verde significa que o jogador está pronto para jogar se estiver a vermelho ainda não está pronto para jogar.

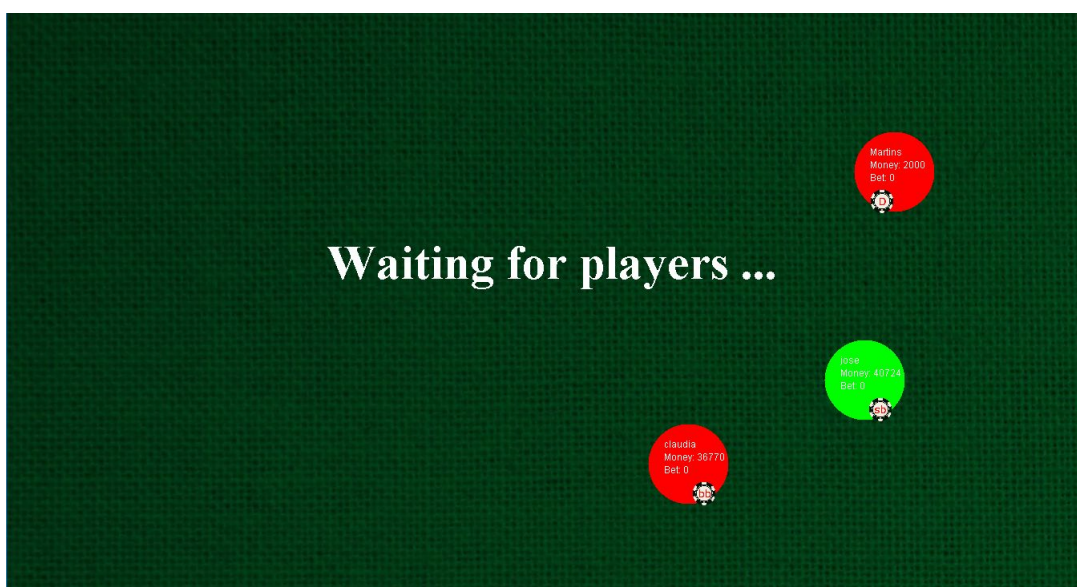


Figura 2 - waiting players

Quando todos os jogadores estão prontos o jogo começa, mudando-se para o ecrã de jogo. As moedas nos jogadores indicam que é o dealer(D), small Blind (sb) e big blind(d). No centro da mesa encontra-se o pot atual com o respetivo valor.



Figura 3 - Imagem de jogo, onde se assinala o jogador que vai jogar

Neste ecrã quando jogador está com o highlight a verde significa que é a vez de ele jogar. Quando um jogador escolhe a ação Fold(desistir do jogo) a sua imagem fica a cinza e este deixa de ser notificado pelo servidor para jogar.



Figura 4 - Imagem de jogo, onde um jogador fez fold

Quando a ronda chega ao fim são mostrados os vencedores a azul e os derrotados a preto, aparecendo no centro da mesa o tipo de mão que o(s) vencedor(s) possui(em) assim como o(s) seu(s) nome(s).



Figura 5 - imagem de jogo, onde foi encontrado o vencedor

Neste momento os jogadores podem notificar novamente o servidor da sua prontidão para iniciar uma nova ronda e quando todos os jogadores estiverem prontos a nova ronda começa repetindo-se o processo.

2.3.2 Client-side

No lado do cliente depois de abrir a aplicação é mostrado o seguinte menu.

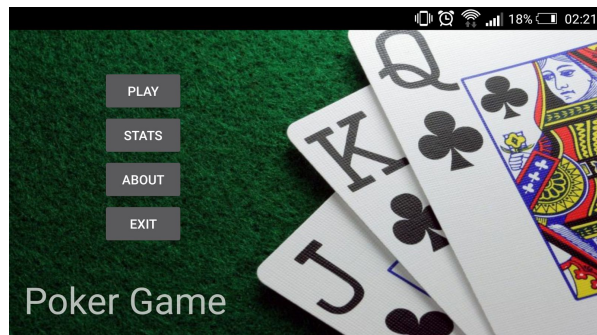


Figura 6 - Imagem inicial da aplicação para android

Na menu about existe apenas uma pequena descrição da aplicação.



Figura 7 - Menu About

No menu das estatísticas é possível consultar as mesmas assim como fazer o reset destas, de notar que o reset não altera o nome e o dinheiro do jogador esses dados só são alterados aquando da criação de uma nova conta(caso do nome) e no final de cada ronda conforme o jogador tenha ganho ou perdido(caso do dinheiro).

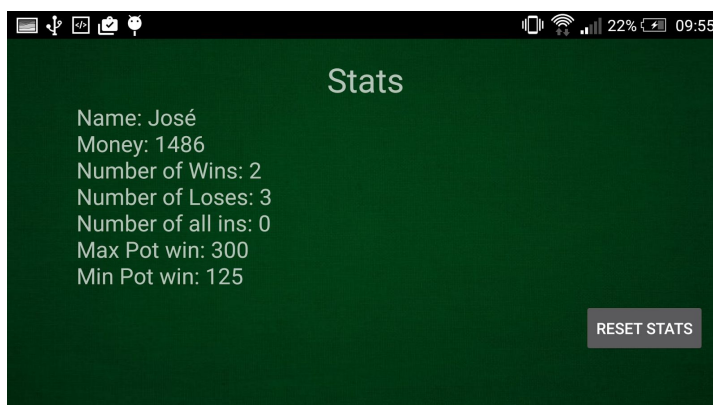


Figura 8 - Menu Stats

Quando o utilizador pretende jogar carregando no menu play é deparado com um dos seguintes ecrãs.

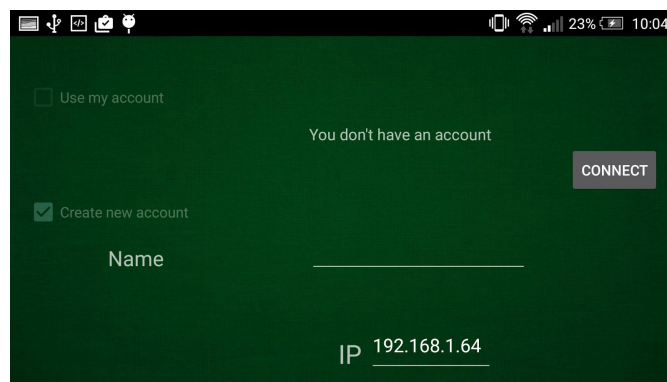


Figura 9 - Não tem conta, para se conectar

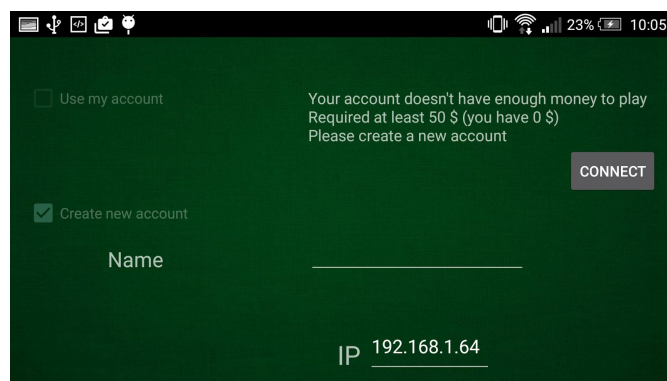


Figura 10 - Não tem dinheiro suficiente na sua conta, para se conectar

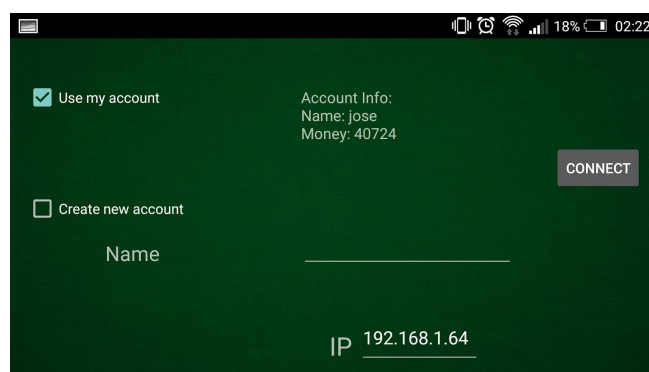


Figura 11 - Utiliza a sua conta, para se conectar

Caso o jogador nunca tenha jogado o jogo é obrigado a criar uma nova conta, se o utilizador já possuir uma conta mas não tiver o dinheiro da aposta mínima também terá que criar uma nova conta, se o utilizador possuir uma conta e dinheiro suficiente pode optar entre a conta existente ou criar uma nova.

O utilizador deve preencher o campo ip com o ip da máquina onde foi arrancado o servidor.

(Sugestão: usar o command prompt do windows escrever “ipconfig” e preencher este campo com o IPv4 Address)

(Nota:ao criar uma conta nova as estatísticas sofrem um reset)

Depois de carregar no botão connect a aplicação tenta conectar-se ao servidor, se a conexão não for possível porque:

- Servidor está fechado/ip errado;
- O dispositivo não possui a internet ligada;
- A mesa está cheia;
- Já existe um jogo a decorrer;
- Nome escolhido não é válido.

O utilizador é retornado para o ecrã inicial com um toast que indica o porque de não ter sido possível estabelecer a ligação, por exemplo:

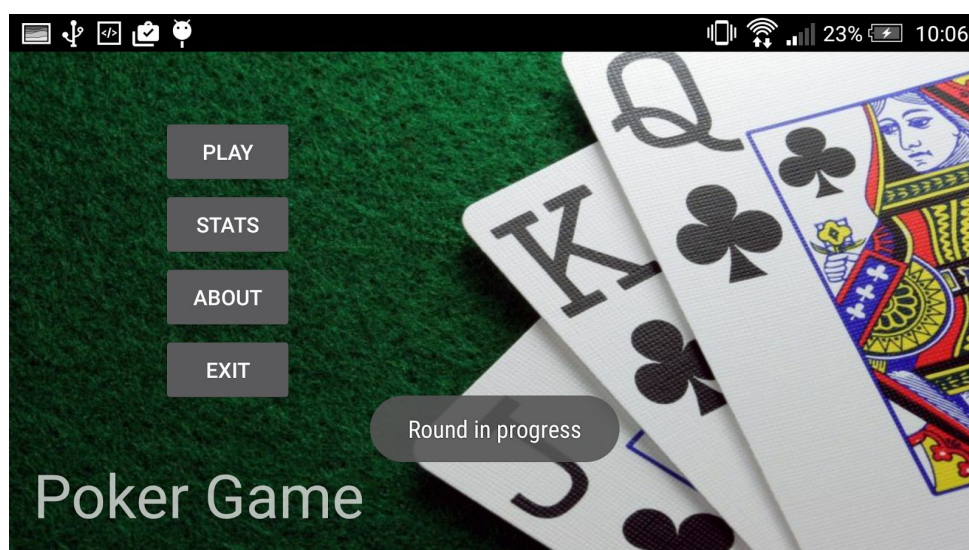


Figura 12 - Erro ao conectar a ronda já está em progresso

Enquanto a conexão é estabelecida é mostrado o seguinte ecrã.

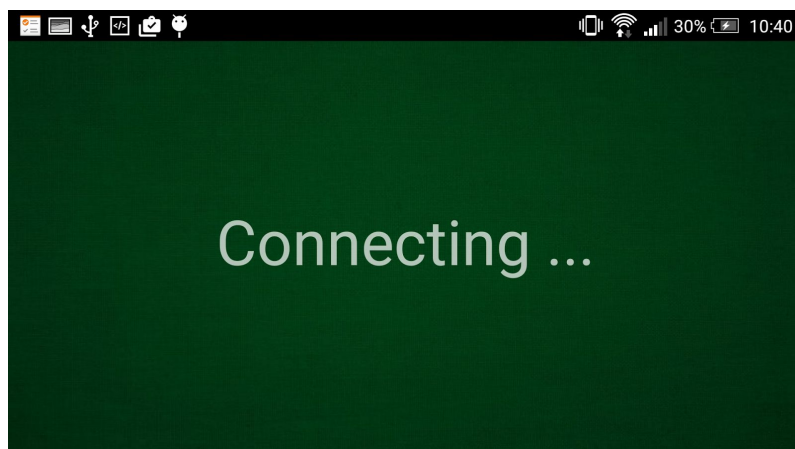


Figura 13 - A conectar

Depois do utilizador ter estabelecido conexão com o servidor é mostrado o seguinte ecrã e ao carregar em “START” o jogador notifica o servidor dizendo que está pronto para jogar
(Nota:depois de carregar em“START” o botão desaparece).

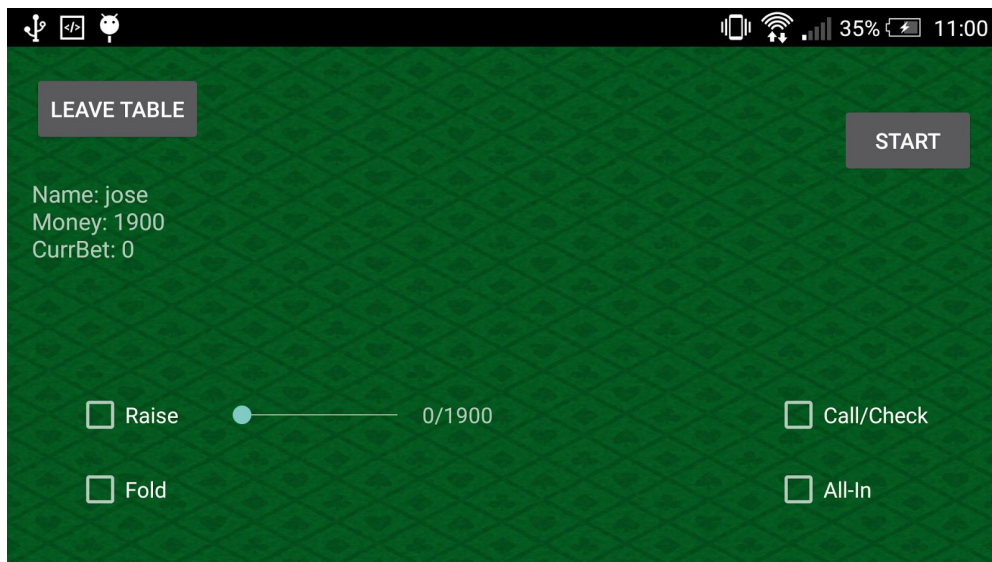


Figura 14 - Imagem de jogo, ainda não entrou na ronda

Quando todos os jogadores estiverem prontos o servidor irá dar as cartas a cada jogador e caso seja a vez do utilizador em questão é mostrado um botão play que permite notificar o servidor da ação que pretendemos tomar, é também mostrado um contador decrescente a vermelho que indica o tempo restante para jogar.

Se o utilizador não jogar nesse tempo o servidor assume que o jogador deu Fold(desistiu do jogo).

Do lado esquerdo poder ver-se o estado em que é o utilizador a jogar e do lado direito o estado em que outro utilizador está a jogar

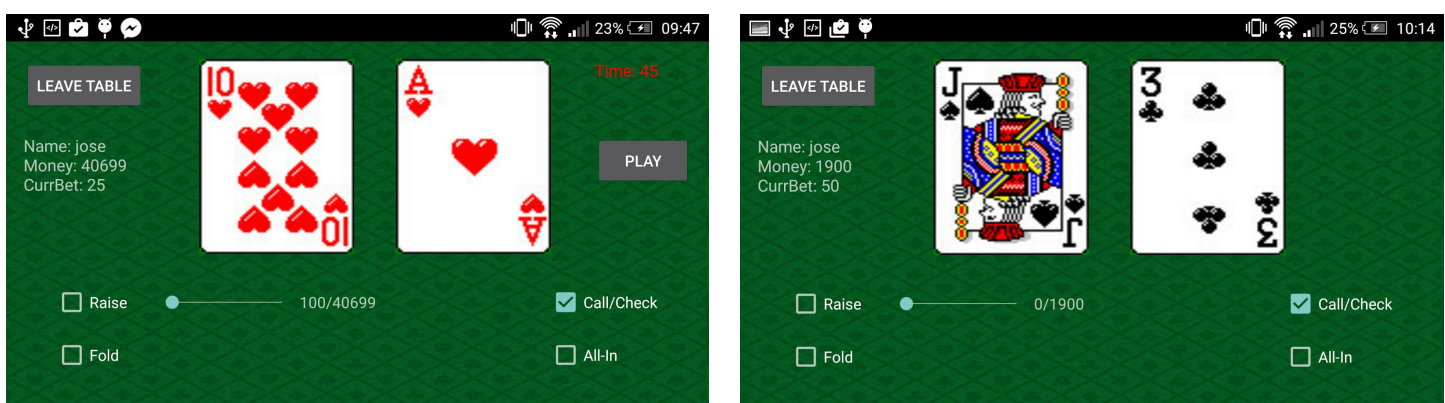


Figura 15 - Imagem de jogo no android

De notar que o utilizador pode sair da mesa durante o jogo através do botão “LEAVE TABLE” ou pressionando a tecla para voltar para trás que todos os dispositivos android possuem.

Se o utilizador sair do jogo, este continua normalmente com os utilizadores que ainda permanecem na mesa.

No final da ronda é mostrada uma indicação a indicar a vitória ou derrota do jogador assim como as cartas que constituem a melhor mão do jogador e o seu tipo.

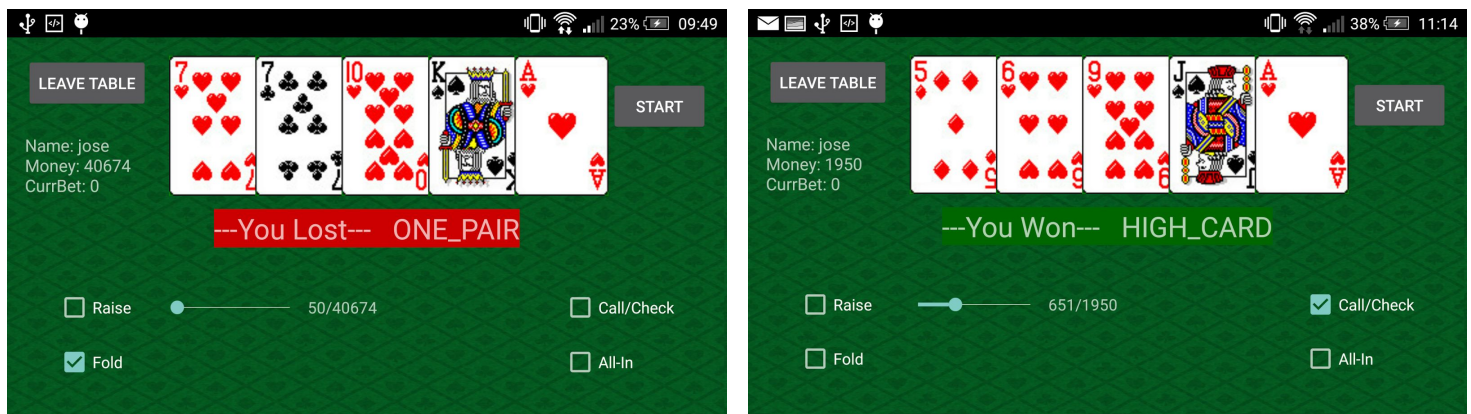


Figura 16 - Imagem no dispositivo android depois de se saber o vencedor

Neste momento o botão “START” volta a aparecer para o utilizador poder notificar o servidor de que está pronto para uma nova ronda e quando todos os jogadores estiverem prontos a ronda começa e o processo repete-se.

(Nota: se o jogador tiver perdido e não tiver o dinheiro necessário para realizar uma a posta minima neste caso 50, ao carregar em start a aplicação volta ao ecrã inicial mostrando um toast a dizer “Out Of Money” e o jogador terá que criar uma nova conta para poder jogar novamente)

2.4.Ficheiros de entrada e saída

A aplicação android cria um ficheiro onde guarda estatísticas e informações da conta do jogador que são utilizadas ao conectar-se ao servidor, de modo a que depois da primeira conexão ao servidor o jogador não precise de criar uma conta nova e possa utilizar a conta já criada que já contem o seu nome e o dinheiro do jogador.

3. Conceção e Implementação

3.1 Estrutura de Packages

3.1.1 Estrutura de packages do Client

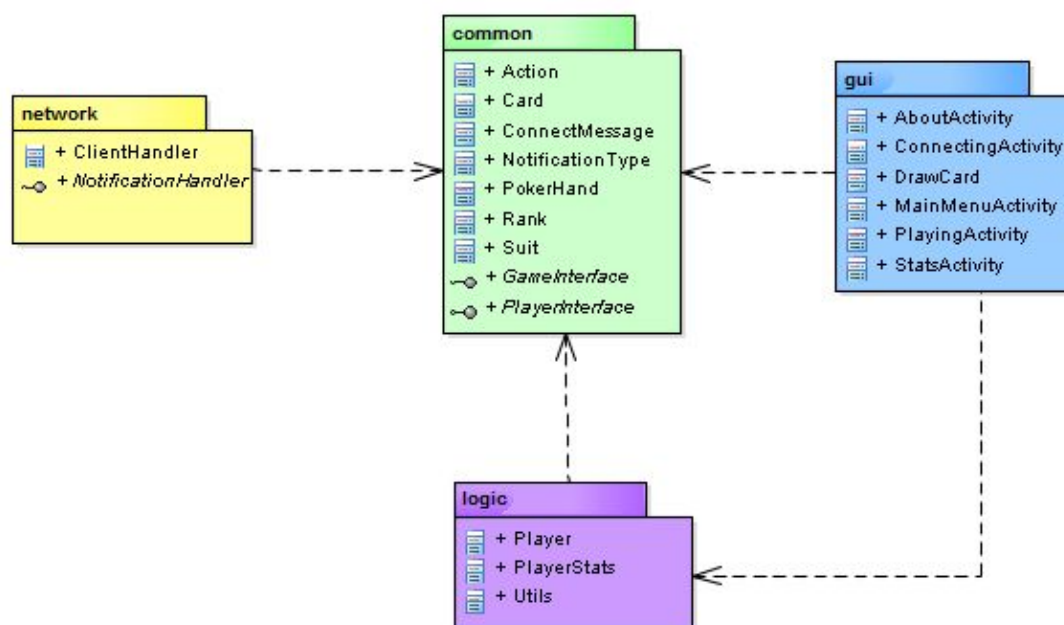


Figura 17 - Diagrama de packages do Client

Package	Descrição
Network	Contém as operações para configurar a rede
Logic	Contém a lógica do client
GUI	Interface gráfica do client
Common	Contém informação comum ao client e ao server

3.1.2 Estrutura de Packages do Server

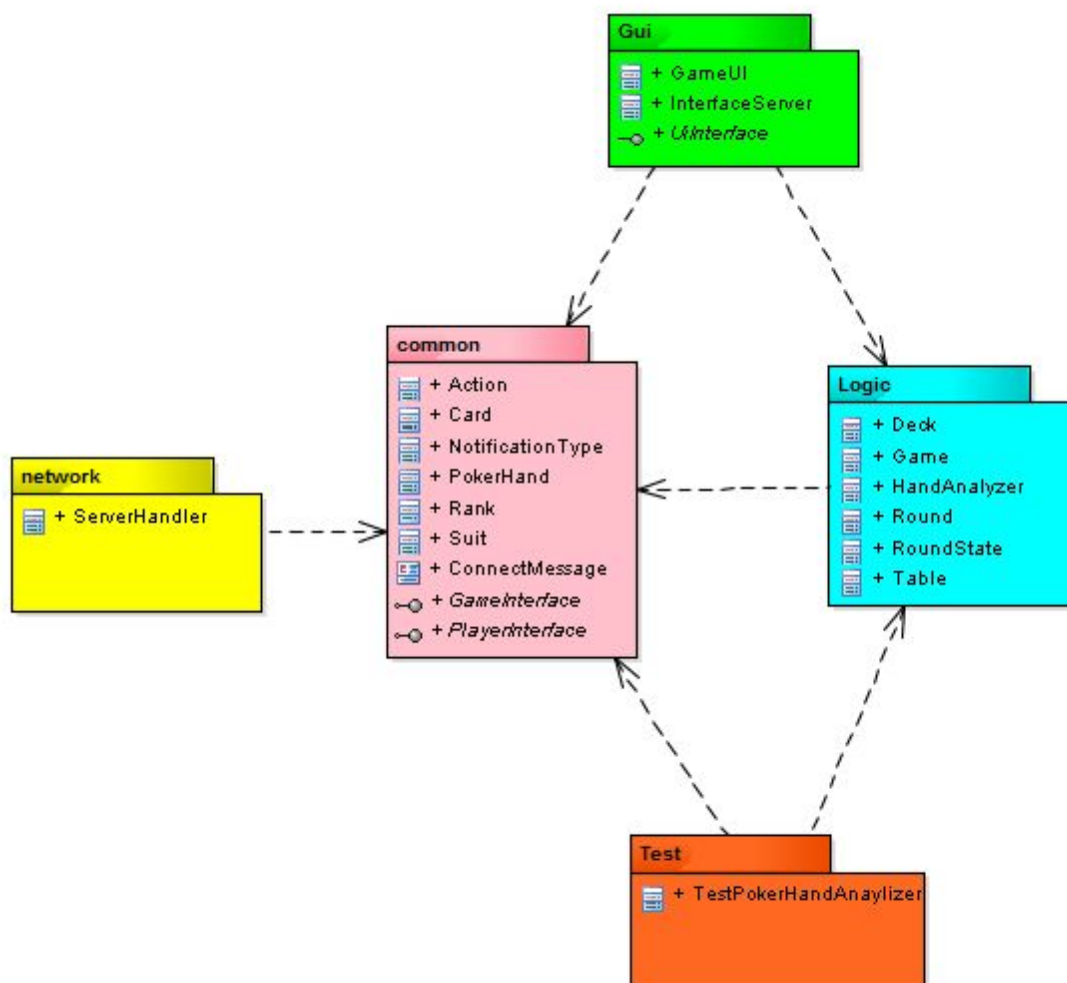


Figura 18 - Diagrama de packages do Server

Package	Descrição
GUI	Contém a interface do server
Logic	Contém toda a lógica do jogo
Network	Contém as operações para configurar a rede
Test	Contém os testes realizados à lógica
Common	Contém a informação que é partilhada pelo server e pelo client

3.2 Diagramas de Classes

3.2.1 Package Network do Client

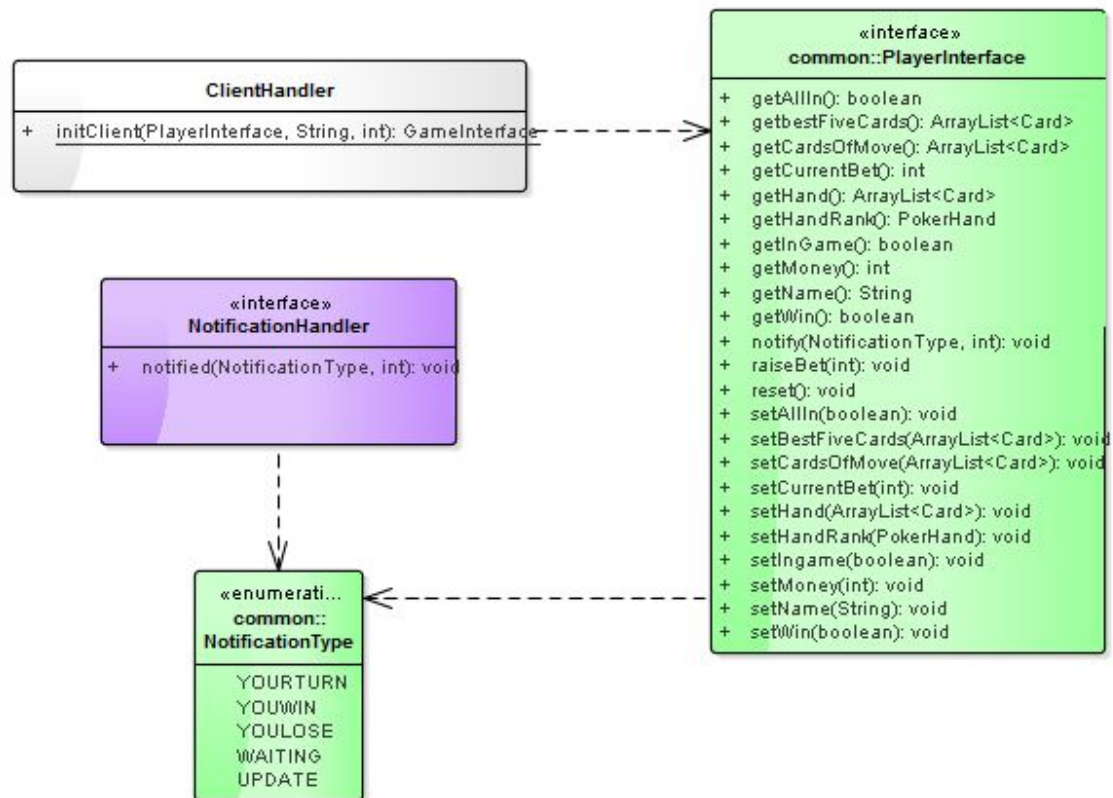


Figura 19 - Diagrama de classes do package Network do Client

Classe	Descrição
clientHandler	Responsável por obter o objeto remoto do jogo
NotificationHandler	Interface implementada pela PlayingActivity para controlar as notificações do servidor

3.2.2 Package GUI do Cliente

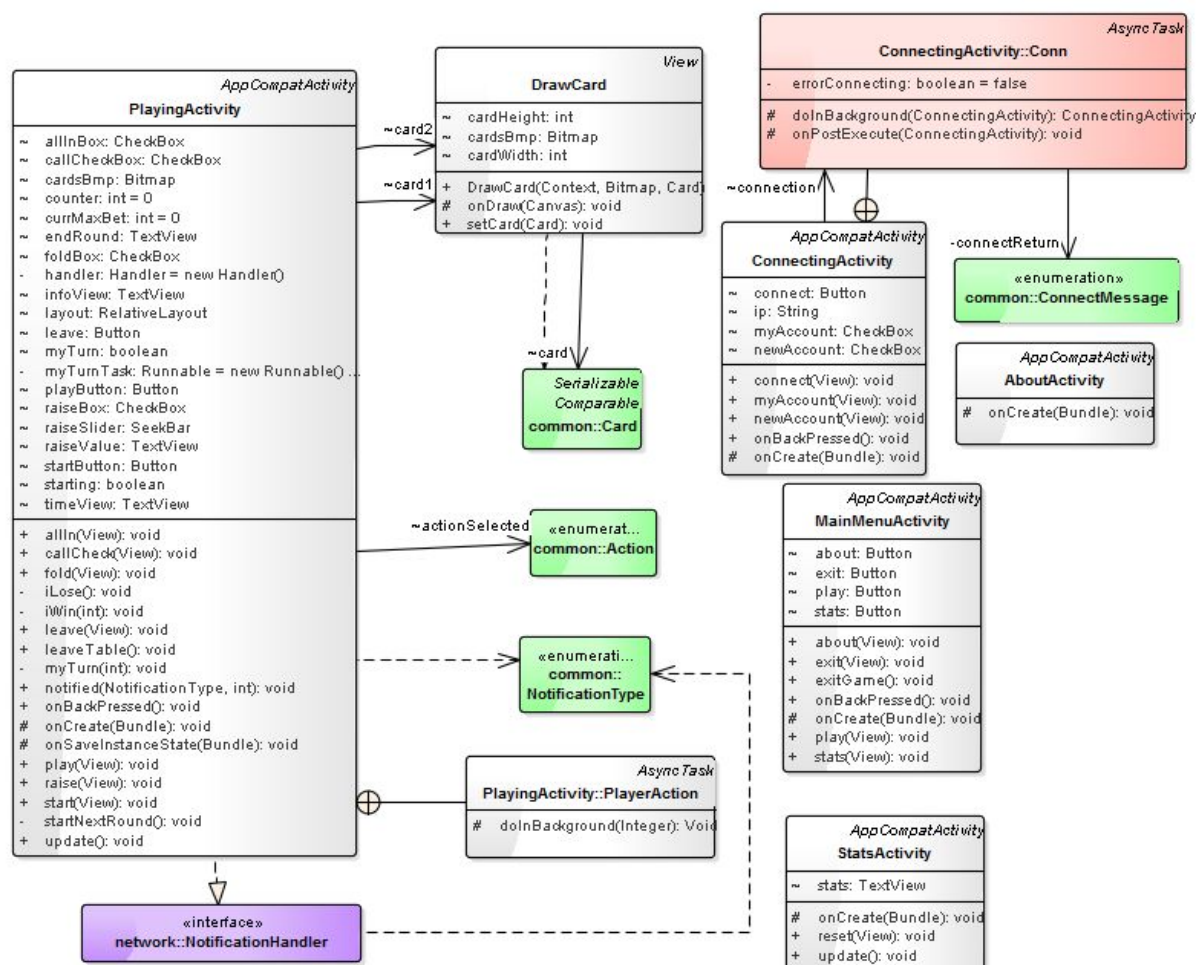


Figura 20 - Diagrama de classes do package GUI do Client

Classe	Descrição
PlayingActivity	Responsável pela interface de jogo
ConnectingActivity	Responsável pela interface de conexão ao servidor
StatsActivity	Atividade que mostra as stats do jogador
MainMenuItem	Atividade inicial com o menu principal
AboutActivity	Atividade responsável por mostrar o menu about
DrawCard	Extende View e desenha uma carta usando o canvas
Conn	Responsável pela conexão ao servidor em background
PlayerAction	Responsável pelo envio da ação do jogador ao servidor em Background

3.2.3 Package Logic do Client

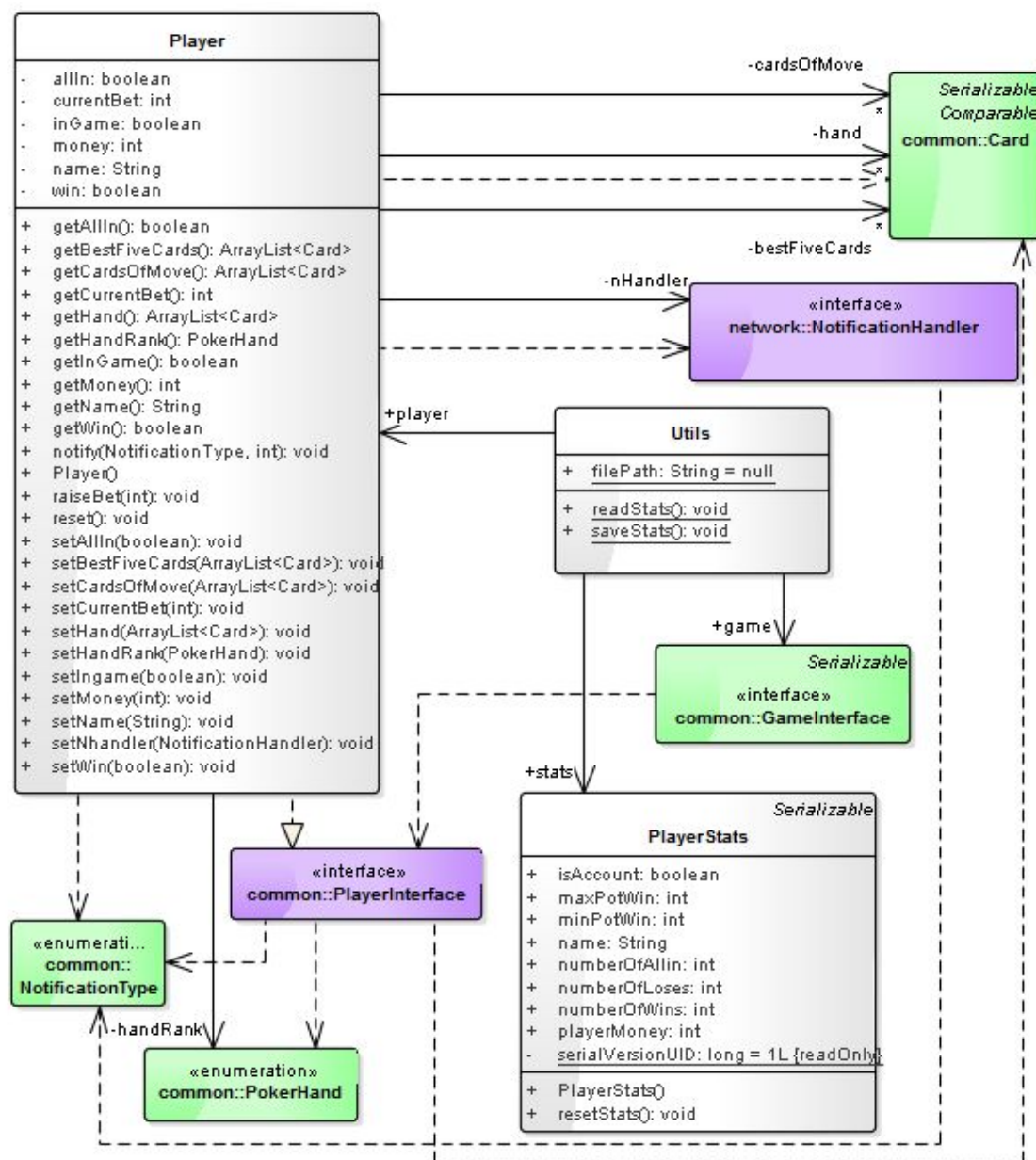


Figura 21 - Diagrama de classes do package Logic do Client

Classe	Descrição
Player	Representa a informação do jogador e é o objeto de Call Back passado ao servidor.
Utils	Contem Utilidade como métodos para guardas e ler ficheiros
PlayerStats	Classe que armazena todas as estatísticas do jogador

3.2.4 Package GUI do Server

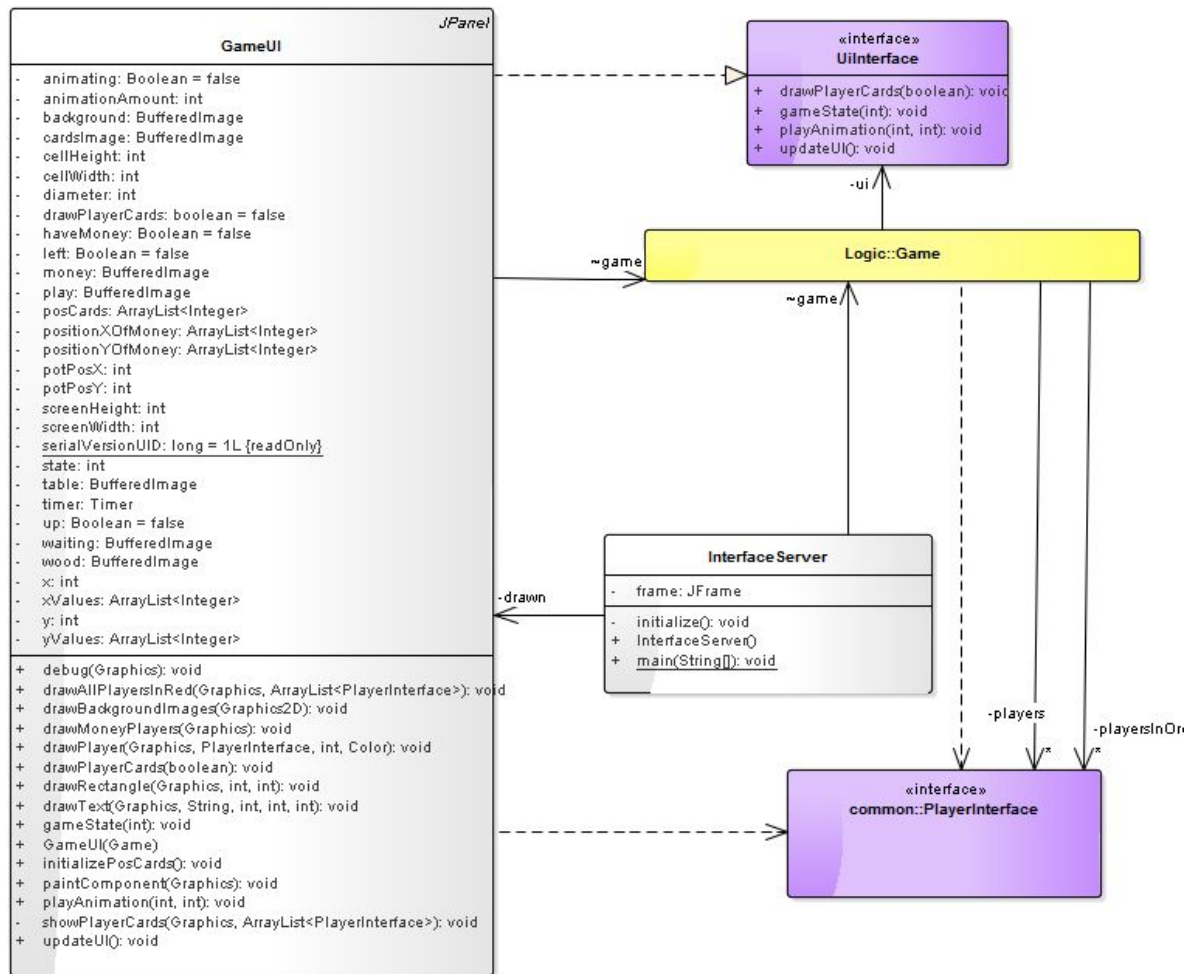


Figura 22 - Diagrama de classes do package GUI do Server

Classe	Descrição
GameUI	JPanel responsável pela parte gráfica do jogo
InterfaceServer	Classe que contém o main e registra o objeto remoto do jogo
UiInterface	Interface que permite que o jogo notifique a parte gráfica para fazer animações ou mudar o seu estado

3.2.5 Package Network do server

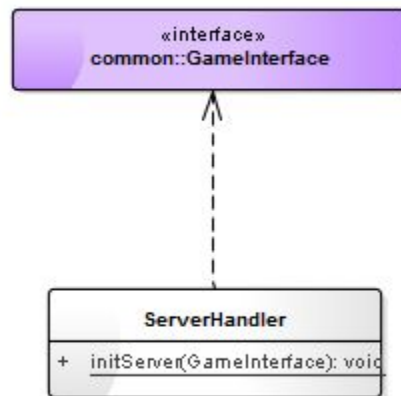


Figura 23 - Diagrama de classes do package Network do Server

Classe	Descrição
ServerHandler	Responsável pelo bind da interface do jogo

3.2.6 Pakage Test do server

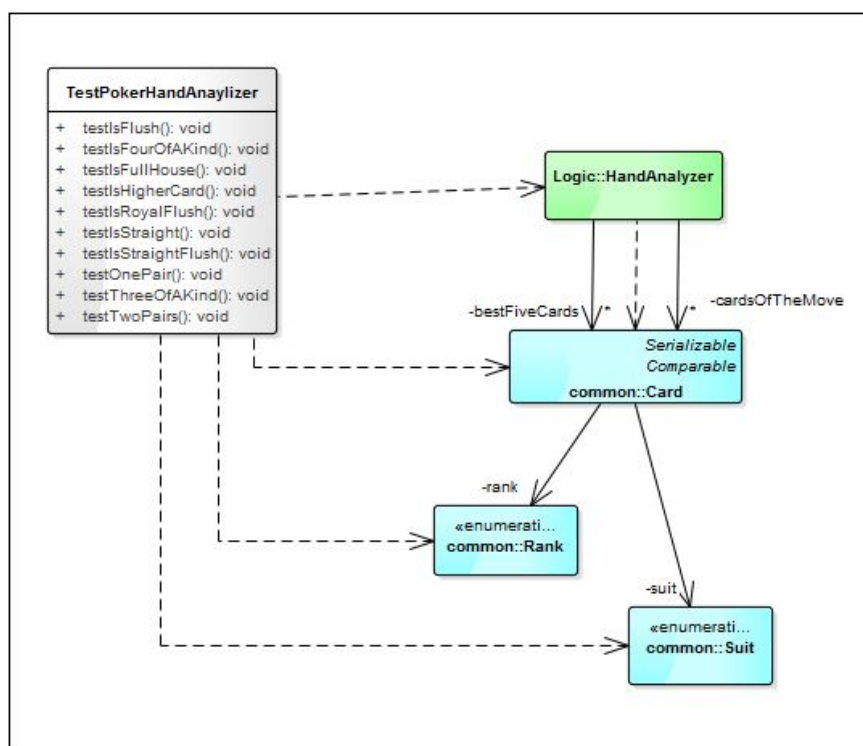


Figura 24 - Diagrama de classes do package Test do Server

3.2.7 Package Logic do server

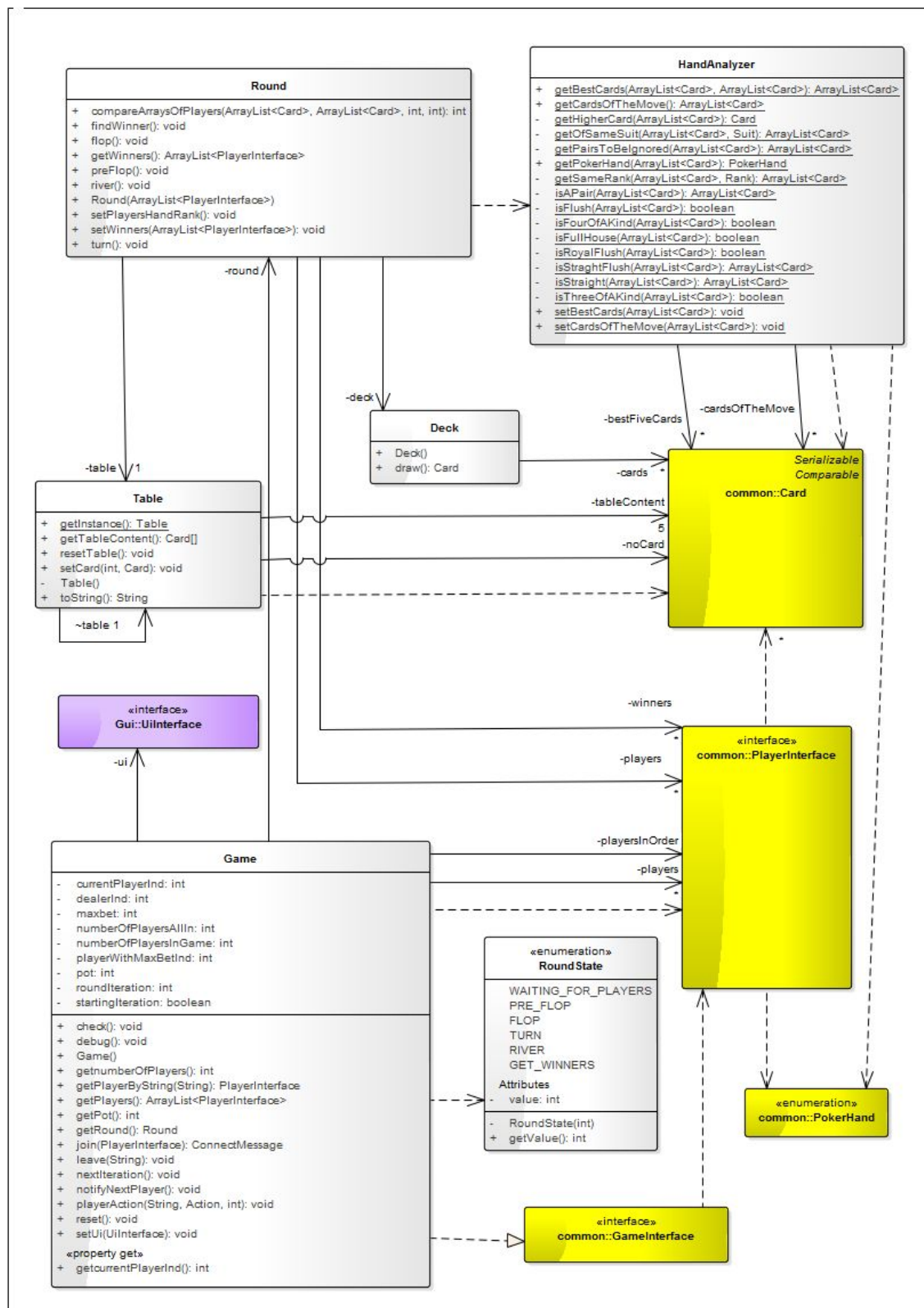


Figura 25 - Diagrama de classes do package Logic do Server

Game	Implementa a interface remota, é responsável pela notificação dos jogadores e
Deck	Representa um baralho
Table	Representa a mesa (Singleton)
Round	Representa a ronda, responsável por dar as cartas aos jogadores e a mesa e obter os vencedores da ronda.
HandAnalyzer	Responsável pela análise do tipo de mão que o jogador tem

3.2.8 Package Common (Server e Client)

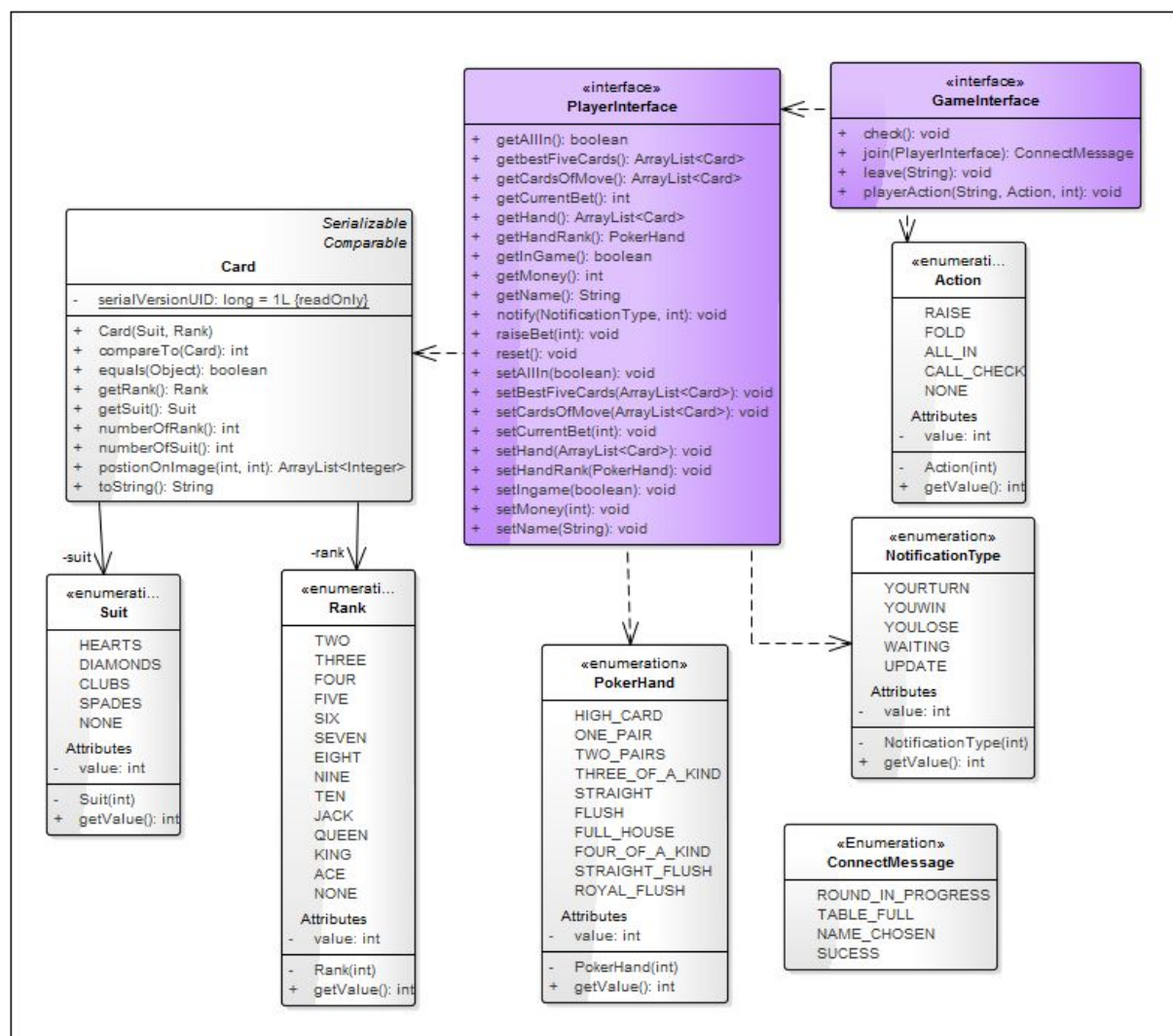


Figura 26 - Diagrama de classes do package Common

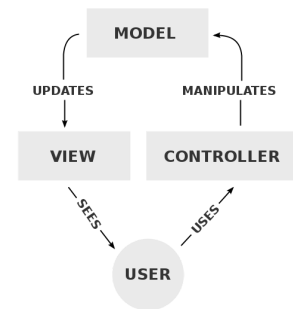
Classe	Descrição
Card	Representa uma carta
PlayerInterface	Interface de Call Back
GameInterface	Interface implementada pelo servidor

3.3 Padrões de desenho utilizados

Os padrões de desenho utilizados foram os seguintes:

Singleton- Usado na Classe Table no Package da Logic do servidor;

MVC(Model View Controller) - A aplicação android funciona como CONTROLLER uma vez que manipula o objeto remoto do jogo(MODEL) que por sua vez manipula a interface gráfica (VIEW).



3.4 Dificuldades encontradas

A principal dificuldade encontrada foi sem dúvida a comunicação entre o servidor e o cliente onde se destaca o uso da biblioteca LipeRMI no android.

A biblioteca LipeRMI tem várias limitações como a latência por vezes elevada, e a falta de informação sobre a biblioteca online, dificultou bastante a resolução de erros.

Outra das dificuldades foi a falta de experiência com o desenvolvimento para android, que tornou algumas tarefas normalmente simples mais complicadas pela falta de alguns conceitos sobre as particularidades do desenvolvimento em android.

No entanto com algum esforço todas as dificuldades acabaram por ser superadas, tendo no entanto atrasado bastante o desenvolvimento do projeto

3.5 Bibliotecas e ferramentas utilizadas

Durante o projeto a aplicação do servidor foi totalmente desenvolvida no IDE eclipse, enquanto que a parte do cliente foi desenvolvida no Android Studio.

Foi utilizada a biblioteca LipeRMI, para a implementação da comunicação entre o servidor e o cliente.

Não foi usado LibGdx.

4. Testes Realizados

Devido a não ser possível testar tudo com testes automáticos, para além dos testes automáticos fizemos ainda vários testes manuais.

Testes automáticos:

- Confirmação de que dado um arrayList que contem as cinco cartas da mesa mais duas do jogador em questão, a avaliação da poker hand é bem sucedida, em todos os casos (Royal Flush, Straight, Four of a Kind, Two Pairs, One Pair, Straight Flush, Flush, Three of a kind e High card);
- Confirmação de que o arrayList que tem as cinco melhores cartas e o arrayList que tem as melhores cartas de um determinado jogador tem os elementos esperados.

Testes manuais:

- Verificação das variações do dinheiro do Client;
- Confirmação de que as cartas ilustradas correspondem às esperadas;
- Verificação de que as cartas são viradas no momento certo;
- Confirmação do estabelecimento de conexão entre o server e o client;
- Confirmação de que as ações escolhidas pelo client são as que são realizadas no server;
- Verificação de que foi o jogador correto a vencer a ronda;
- Verificação do correto movimento das moedas jogadas pelo player até ao pot;
- Confirmação de que o jogador só joga quando for a vez dele;
- Confirmação da correta alternância do big blind, small blind e do dealer;
- Confirmação de que o jogador apenas é notificado caso seja a vez dele e este ainda estiver em jogo;
- Confirmação de que se o jogador não fizer a sua jogada ao fim de um determinado tempo, uma ação default é realizada;
- Confirmação de que quando o player se desconecta do servidor no decorrer da ronda, as notificações aos outros jogadores se mantêm corretas independentemente de a saída ocorrer durante o seu turno ou não;
- Confirmação da correção dos dados guardados e lidos dos ficheiros;
- Confirmação do correto tratamento de todas as exceções originadas pela conexão ao servidor e pelas notificações ao servidor;
- Verificação de que o cliente guarda o estado da atividade quando o ecrã é bloqueado e depois desbloqueado.

Conclusão

Com a realização deste projeto no âmbito da Unidade Curricular de Laboratório de Programação Orientada por Objetos, foi possível utilizar várias competências adquiridas ao longo da cadeira assim como aprender algumas novas ligadas ao desenvolvimento em android.

Foram atingidos todos os objetivos, propostos pelo grupo na entrega intermédia.

Uma possível melhoria ao nosso jogo, seria a implementação de animações nas cartas e o uso de outro tipo de comunicação mais baixo nível por forma a diminuir a latência.

As tarefas foram de igual modo distribuídas, pelos dois membros do grupo.