U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Laboratório de Progamação Orientada por Objetos
2015/2016

Champions Field

Turma 5
José Carlos Alves Vieira
up201404446@fe.up.pt

Renato Sampaio Abreu
up201403377@fe.up.pt

7 de Junho de 2016

Índice

1. Introdução

No âmbito da unidade curricular Laboratório de Programação Orientada a Objetos do Mestrado Integrado em Engenharia Informática e Computação, e com o objetivo de criar uma aplicação Android, propusemo-nos, então, a desenvolver o jogo Champions Field. Este, de uma forma geral, consiste num simulador de futebol, em que o utilizador, em modo singleplayer ou multiplayer, controla um jogador, tendo como objetivo derrotar a equipa adversária.

A aplicação foi desenvolvida pensada tanto para desktop como para Android, sendo possível de ser jogada em ambas os dispositivos.

Assim, o presente relatório tem como objetivo explicar de forma detalhada as funcionalidades, modos de utilização e também os detalhes de implementação da respetiva aplicação.

Posto isto, o relatório está estruturado em várias secções que incidem minunciosamente nos pontos referidos anteriormente.

O projeto em questão encontra-se no repositório: ChampionsField.

2. Manual de Utilização

2.1. Funcionalidades

A aplicação Champions Field permite ao utilizador escolher entre um modo Singleplayer e Multiplayer.

Assim, relativamente ao modo Singleplayer, o utilizador pode correr a aplicação no desktop ou no Android. Neste modo, é apresentado um campo de futebol com vários jogadores, em que um é controlado pelo jogador, podendo no entanto, caso o utilizador assim o pretenda, ocorrer uma troca do jogador controlado. O objetivo é, então, derrotar a equipa adversária, marcando o maior número de golos possíveis. Devido à dificuldade de implementação de uma AI para o controlo dos jogadores não controlados pelo utilizador, este modo poderá ser considerado como um "treino", possuindo apenas uma AI básica, mas jogável.

Em relação ao modo Multiplayer, primeiramente é necessário que haja uma aplicação Servidor a correr num Desktop e cuja função é ser host de no máximo dois jogos (rooms). A aplicação cliente, inicializada a partir do Andoid, caso haja um host conecta-se ao Servidor e a um de dois rooms e fica em espera até que haja jogadores suficientes ligados para que a partida inicie, ou seja, espera que exista um cliente da equipa "Azul" e outro da equipa "Vermelha", não podendo os dois clientes terem o mesmo nome (alterável nas "Settings"), voltando ao menu principal caso um cliente se tente ligar e não siga as condições descritas acima. No momento em que ocorre a inicialização do jogo, cada aplicação Cliente permite ao utilizador controlar o respetivo jogador. As alterações ocorridas em cada aplicação Cliente são enviadas ao Servidor e consequentemente aos restantes jogadores, havendo uma sincronização constante de de todas as aplicações Cliente.

2.2. Instalação e arranque do programa

Caso se pretenda correr a aplicação a partir do código do projeto, é necessário importá-lo para o Android Studio.

No modo Singleplayer, basta selecionar a opção "Run 'Desktop'" ou "Run 'Android'" para correr a aplicação, visto que não há restrições quanto ao dispositivo utilizado ou à rede ao qual se esteja conectado No último caso, é necessário ligar o dispositivo Android ao computador, com o modo de depuração ativado.

Posto isto, é necessário definir o número de jogadores por equipa, caso o utilizador pretenda (por predefinição, o número é 3), e escolher a opção "Singleplay".

No modo Multiplayer, primeiro, é necessário inicializar o Servidor ("Run 'MPServer.main()'". A aplicação Cliente pode ser inicializada no próprio desktop ou então no Android. Uma vez que haja jogadores suficientes conetados ao servidor dá-se início ao jogo.

Por outro lado, no modo Multiplayer para que possa ocorrer conexão entre o Cliente e o Servidor é necessário que os respetivos dispositivos estejam conectados a uma rede cujo IPV4 seja igual ao definido na classe Network do package Server. É também pertinente voltar a frisar que a aplicação Servidor tem que ser, obrigatoriamente, executada em primeiro lugar.[1]

[1] Pode ser necessário configurar o firewall do Windows para permitir a ligação com as aplicações Java.

2.3. Modo de utilização

Executando o apk no Android ou correndo o programa em desktop, é mostrado o ecrã incial que inclui “TextButtons”, os quais quando selecionados permitem aceder aos dois modos de jogo (Multi ou Single play), ao menu de opções do jogo e ao menu dos melhores resultados.

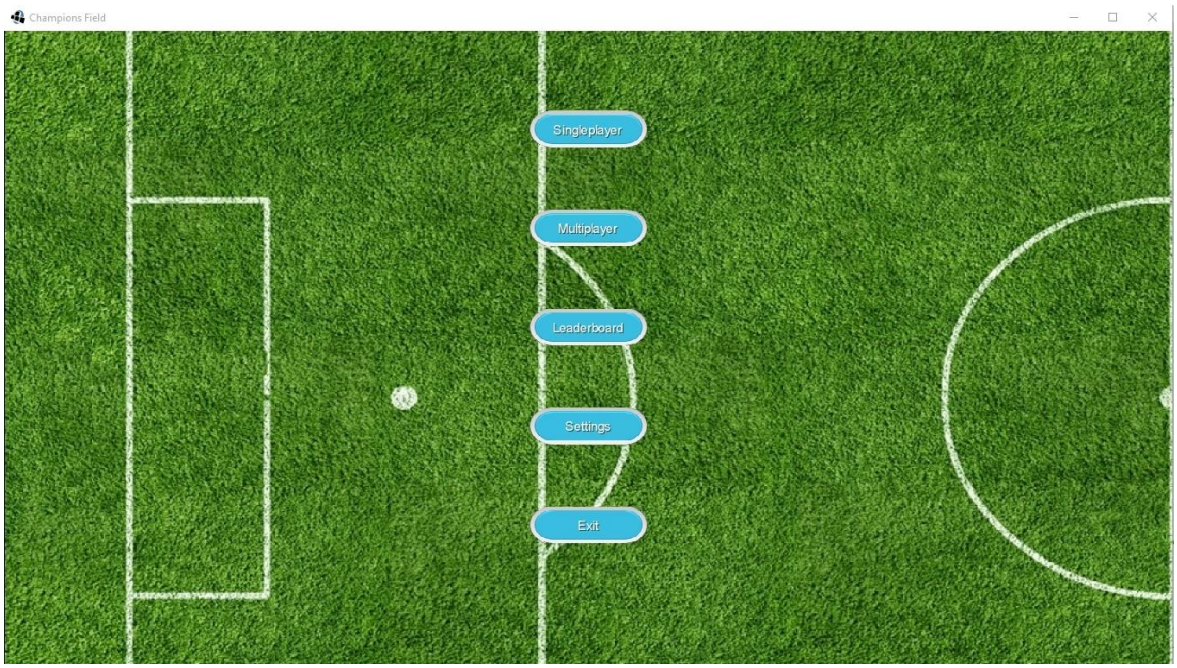


Figura 1 - Menu principal

No menu “Settings”, o utilizador pode escolher o seu username e a equipa inicial, opções estas necessárias apenas no modo Multiplayer. A opção “Players per Team” define o número de jogadores de cada equipa no modo Singleplayer.

Figura 2 – Menu “Settings”

No menu “Leaderboard”, o utilizador verifica a tabela dos melhores jogadores, ordenada segundo o número de golos marcados.

Figura 3 – Menu “Leaderboard”

Acedendo ao modo de jogo “Singleplay”, é mostrado o display de uma animação de loading enquanto a instância do “SinglePlayMatch” é devidamente inicializada.

Figura 4 – Loading Singleplay

No ecrã de jogo Singleplay, o utilizador controla um jogador, recorrendo ao “Touchpad” localizado no canto inferior esquerdo. No canto inferior direito encontra-se um botão de “switch”, permitindo ao utilizador trocar entre os jogadores da sua equipa.

Figura 5 – Singleplay match

No modo Singleplay existem “power ups” que podem ser capturados por qualquer elemento das equipas. Os poderes são: Aumento de velocidade de todos os jogadores da própria equipa, aumento da velocidade do jogador que apanhou o power up, e diminuição da velocidade dos jogadores da equipa contrária. As suas imagens, de acordo com a ordem referida anteriormente, são:

Figura 6 – Power ups

Uma vez inicializado o Servidor, o utilizador, na sua aplicação Cliente, pode aceder à opção “Multiplayer”. Posto isto, será direcionado para o respetivo Lobby do Servidor, podendo então entrar em qualquer “Room”, desde que esta não se encontre cheio e que o cliente tenha um nome diferente dos jogadores que já estão conectados.

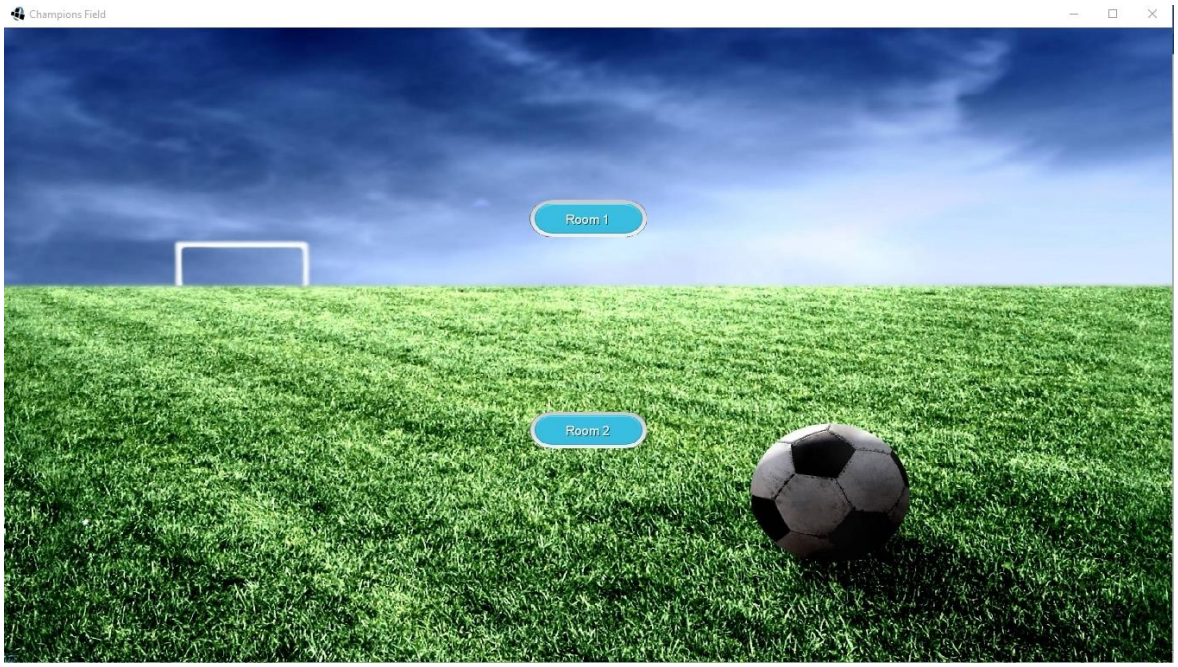


Figura 7 – Multiplayer lobby

Após entrar na respetiva “Room”, é mostrado um ecrã similar ao exibido no loading do Singleplay, até que ambas as equipas que vão participar no jogo estejam completas, ou seja, o jogo multiplayer apenas é inicializado quando cada equipa tem o número de jogadores (Clientes) previamente definido.

Uma vez que os requisitos anteriores sejam cumpridos é então dado o ínicio do jogo. O controlo do jogador é, tal como no Singleplayer, realizado atráves do Touchpad existente no canto inferior esquerdo.

Através da comunicação entre o Servidor e os Clientes existentes na Room em questão, cada alteração de posição a que o jogador controlado ou a bola são sujeitos, a informação dessa alteração é enviada ao Servidor, fazendo depois o update de todos os Clientes. Desta forma, todos os Clientes estão constantemente a sofrerem updates e por conseguinte a fazer o display sincronizado do jogo.

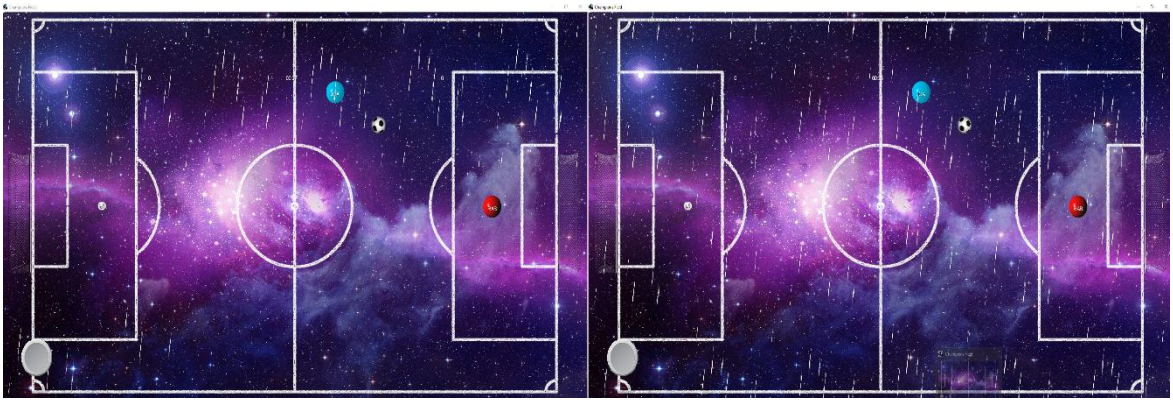


Figura 8 – Multiplayer match entre duas aplicações Cliente

2.4. Ficheiros de entrada e de saída

A aplicação inclui um ficheiro “Statistics.txt”, o qual é acedido para leitura na “Leaderboard” e para leitura e escrita no final de um Multiplayer match.

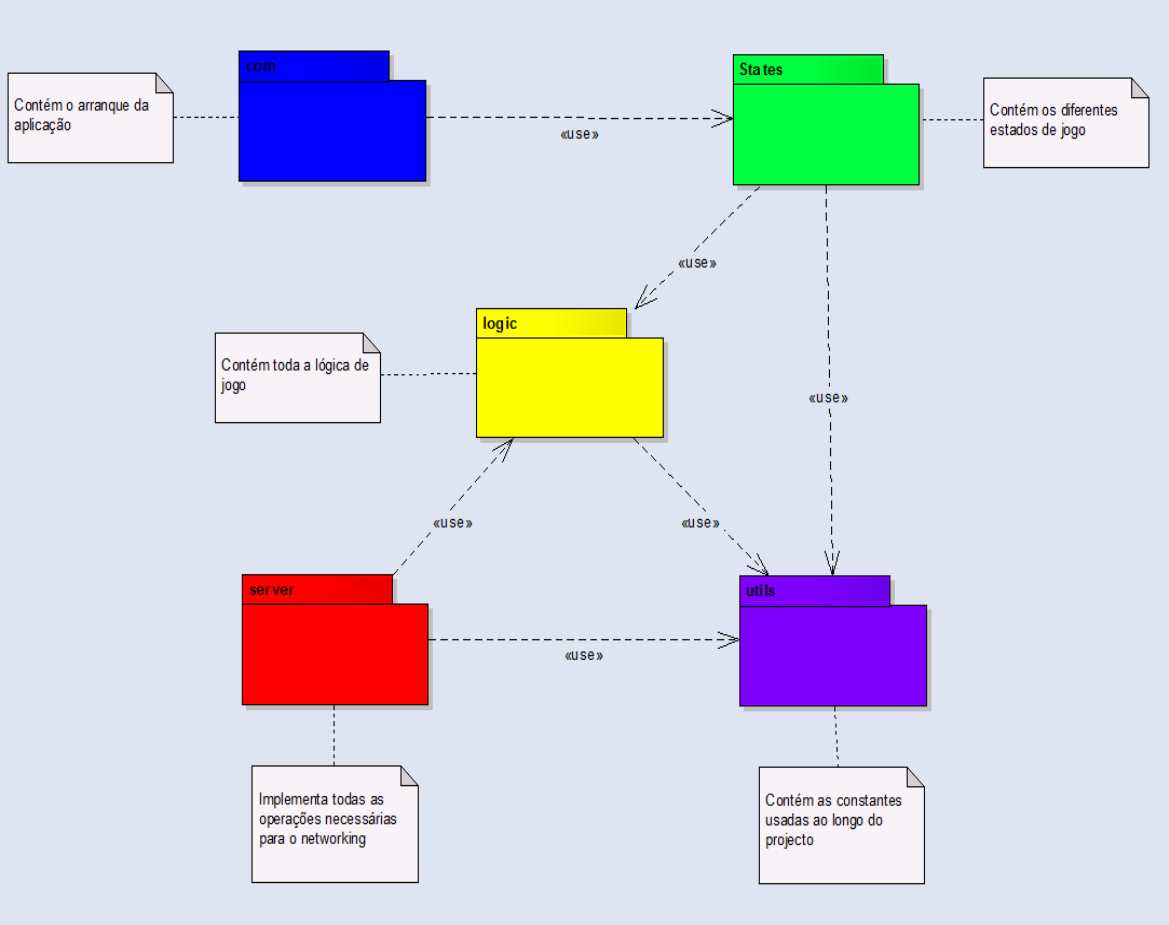
Desta forma, no menu “Leaderboard”, o acesso é realizado de forma a recolher informação sobre os jogos Multiplayer realizados e sobre as estatísticas dos jogadores presentes nesses jogos.

No final do Multiplayer match, o ficheiro é utilizado para atualizar as estatísticas dos jogadores participantes no respetivo jogo, de acordo com a sua prestação. Para isto, é primeiro feito um acesso para leitura e posteriormente um update, ou seja escrita no ficheiro com as estatísticas atualizadas.

3. Conceção, Implementação e Teste

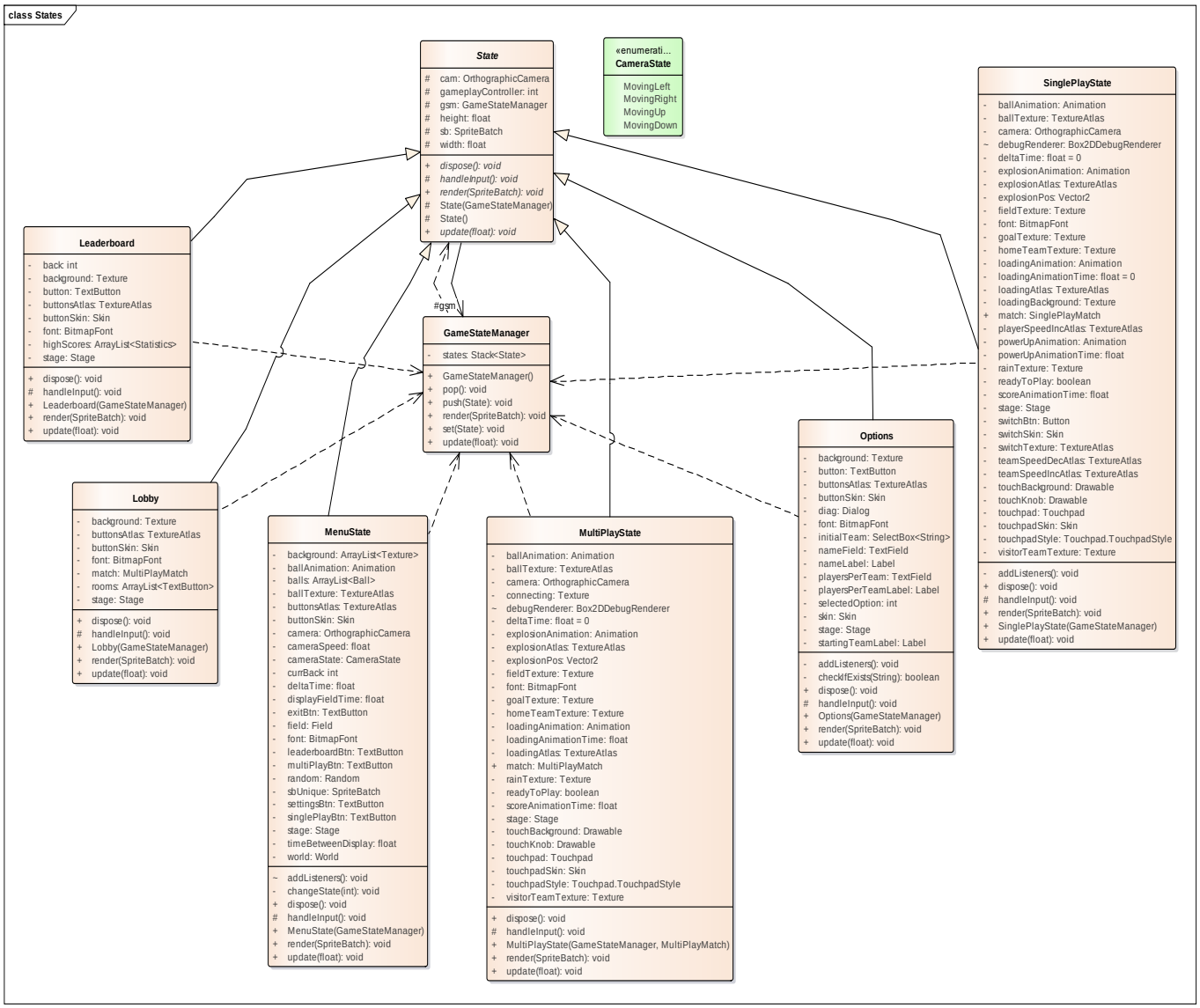
3.1. Estrutura de packages

Diagrama UML com os packages utilizados na aplicação, evidenciando as suas dependências.



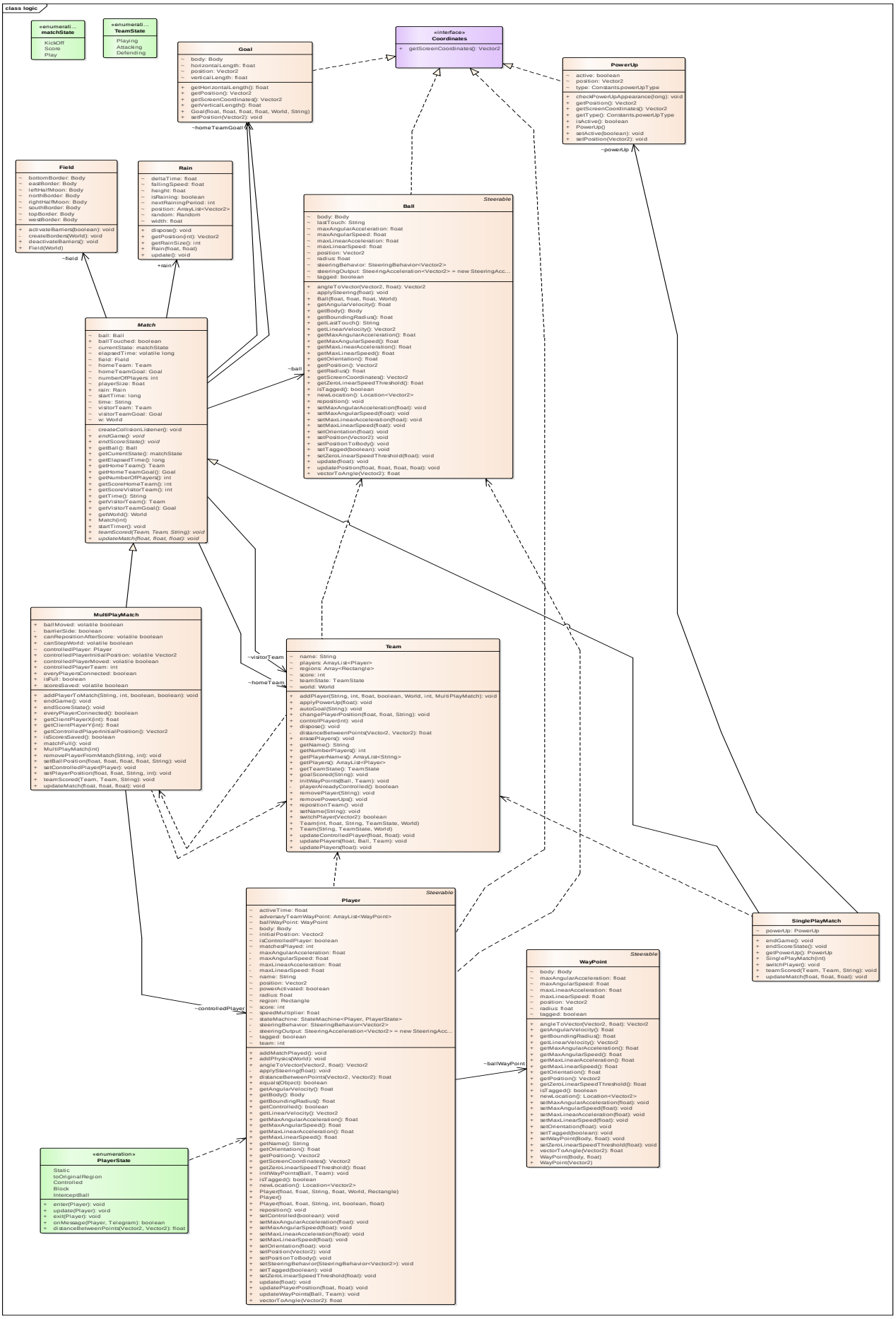
3.2. Estrutura de classes

Package **States** – Responsável pela implementação de toda a interface gráfica da aplicação, bem como a transição entre os respetivos menus.



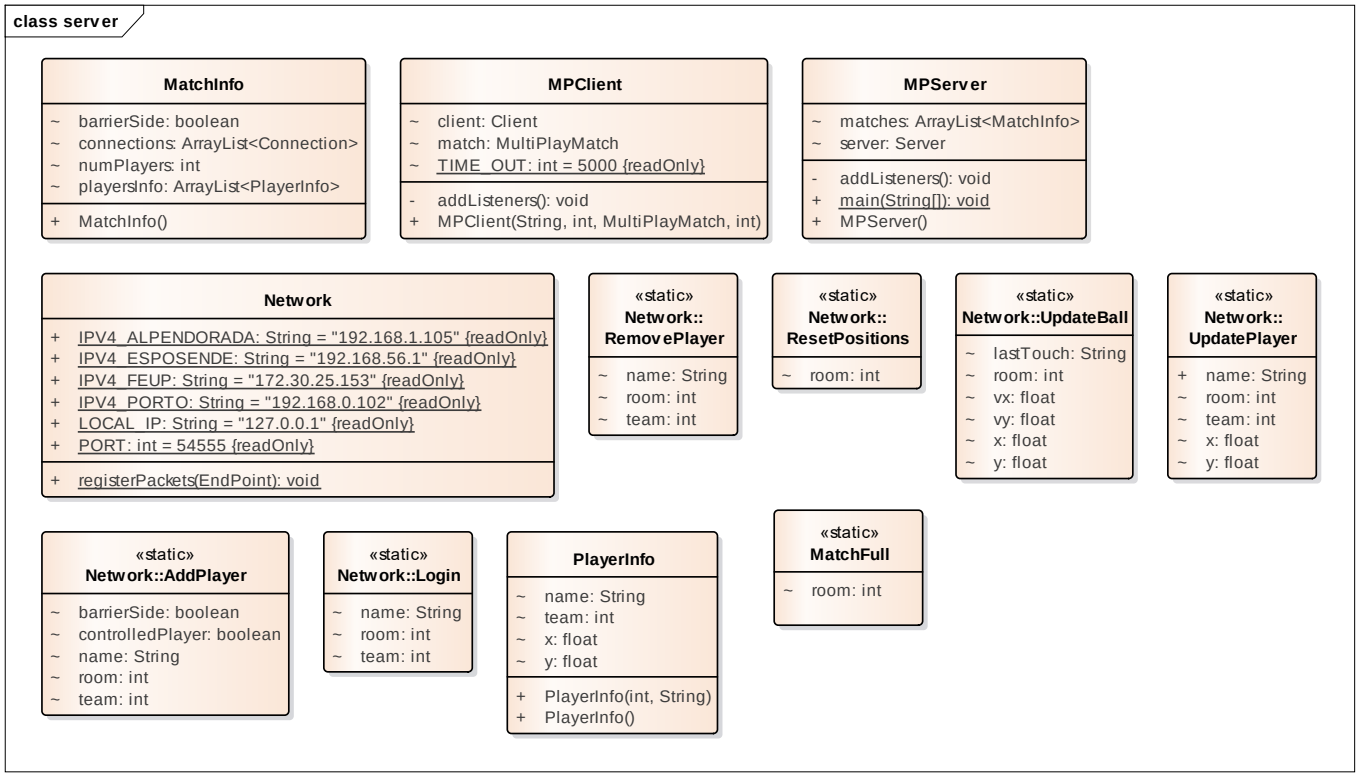
Classe	Descrição
Leaderboard	Implementa o ecrã de highscores.
Options	Implementa o ecrã de opções e a respetiva interface responsiva.
SinglePlayState	Implementa o ecrã de jogo do modo Singleplayer.
MultiPlayState	Implementa o ecrã de jogo do modo Multiplayer.
Lobby	Implementa o ecrã de conexão do modo Multiplayer.
MenuState	Implementa o ecrã principal da aplicação.
State	Classe abstrata de um estado.
GameStateManager	O controlador de estados. Possui uma stack, e o estado que está no topo é o estado que está a ser tratado.

Package **Logic** – Responsável pela lógica de jogo, bem como pela física e as suas relações com os objetos de jogo.



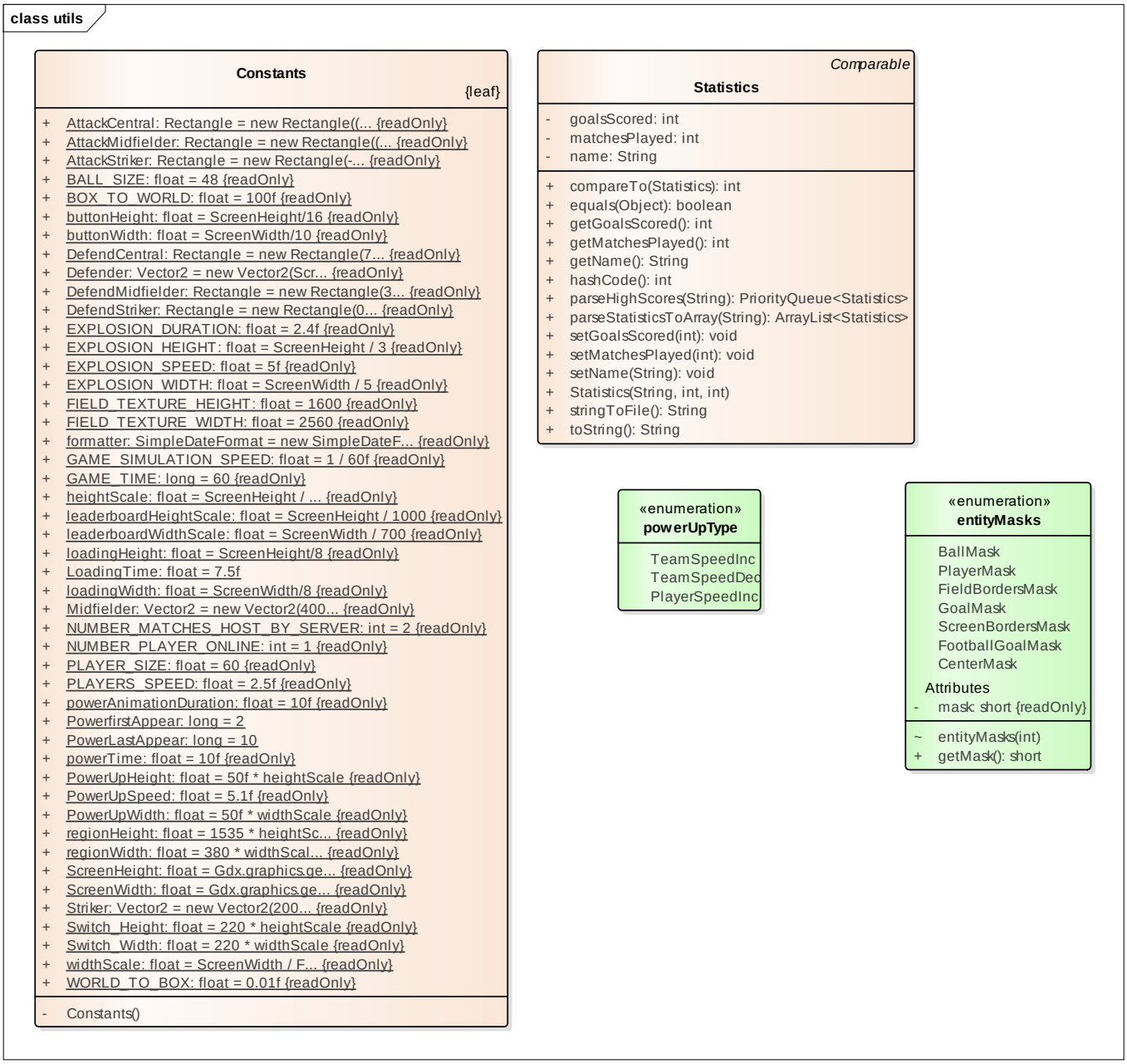
Classe	Descrição
Ball	Representa a bola do jogo.
Coordinates	Interface cuja função transforma as coordenadas de objetos do “World” em coordenadas do ecrã.
Field	Representa o campo de futebol e os respetivos limites do campo e do ecrã.
Goal	Representa a baliza e a linha de golo.
PowerUp	Classe que determina o aparecimento de powerups, bem como as suas características.
Rain	Representa a chuva aplicada.
Team	Contém a informação de uma equipa, nomeadamente todos os jogadores.
WayPoint	Representa um ponto e as suas características físicas.
Player	Classe que abstrai o jogador.
PlayerState	Classe que representa o estado atual do “player”, e atualiza-o conforme.
Match	Classe abstrata que a informação comum entre as partidas singleplayer e multiplayer.
SinglePlayMatch	Subclasse de Match, possui as particularidades de uma partida offline e da AI.
MultiPlayMatch	Subclasse de Match, possui as particularidades de uma partida online.

Package **Server** – Responsável pela implementação da aplicação Servidor e Cliente, tal como os mecanismos de sincronização e envio de informação entre ambos.



Classe	Descrição
Network	Possui diferentes IP's e porta, e regista as classes a trocar entre o cliente e servidor.
MPClient	Cliente que permite a comunicação com o servidor (envio e receção de mensagens).
MPServer	O servidor de jogo que comunica com todos os clientes.
MatchInfo	Contém informações necessárias para distinguir entre as partidas que ocorrem no servidor.
PlayerInfo	Contém informações necessárias para distinguir entre os diferentes jogadores no servidor.

Package **Utils** – Funções auxiliares e constantes utilizadas pelos restantes packages.



Classe	Descrição
Constants	Classe que guarda todas as constantes utilizadas nas restantes packages.
Statistics	Representa as estatísticas de cada jogador.

3.3. Padrões de desenho

MCV (model-view-controller): de forma a que possíveis alterações na interface gráfica do jogo não implicassem uma alteração drástica na lógica ou vice versa, estas foram separadas em diferentes packages, permitindo assim uma maior robustez e uma maior clareza em termos de legibilidade do código.

State: a class "State" é uma classe abstrata extendida pelas classes presentes no package States, de forma a que a transição entre interfaces gráficas (menus) seja implementada a partir de uma máquina de estados.

Observer: A verificação de golos marcados e do respetivo jogador que marcou são registados recorrendo à interface ContactListener presente no LibGdx.

3.4. Ferramentas, bibliotecas e tecnologias

O projeto foi desenvolvido no **Android Studio** e a framework **Libgdx** foi amplamente utilizada. Desta forma, será pertinente referir o uso dado às bibliotecas mais utilizadas do Libgdx:

O **Box2D** foi utilizado de forma a implementar um modelo de física complexo no jogo e abstrair-nos de muitos problemas relacionados com a física de um jogo de futebol(colisões, velocidades, acelerações, etc).

A **AI** foi necessária na implementação do modo Singleplayer, mais especificamente no controlo de jogadores que não são controlados pelo utilizador.

Por fim, também recorremos ao **TexturePacker** e **Scene2D** para criar a interface de utilizador.

Em relação ao desenvolvimento do network, decidimos utilizar a biblioteca **KryoNet**, uma vez que permitia uma comunicação Cliente/Servidor bastante eficiente e simples, ótima para aquilo que queríamos fazer.

3.5. Dificuldades

Ao longo do projeto sentimos certa dificuldade na utilização de alguma das bibliotecas referidas anteriormente.

A implementação donetwork com o uso do KryoNet e, por vezes, a falta de documentação sobre esta biblioteca tornou a resolução de alguns problemas, que apareceram ao longo do desenvolvimento do projeto, difíceis de resolver.

O uso do Box2D e o desconhecimento de como funcionava a API desta biblioteca dificultou-nos a implementação inicial da física do jogo.

A implementação de testes unitários devido à quase inexistência de documentação sobre o uso de JUnit4 em libGdx.

3.6. Testes unitários

Como já foi dito acima, tivemos bastantes dificuldades relativos à implementação de testes unitários com o libGdx em Android Studio. A fraca / inexistente documentação que conseguimos encontrar não foi suficiente para implementarmos os testes, acabando, depois de muito pesquisar, por seguir em frente.

No entanto, para compensar a falta de testes automáticos, dedicámo-nos a realizar bastantes testes manuais, entre eles:

- O jogador consegue "atravessar" o limite das linhas brancas do campo, mas não consegue atravessar as linhas que definem a totalidade do ecrã;
- A bola colide com as linhas brancas do campo;
- O jogador não colide com a linha de golo, podendo entrar perfeitamente na totalidade da baliza;
- A bola colide com a linha de golo, mudando o estado de jogo e consequentes informações relativas a um golo;
- Após a marcação de um golo, todos os elementos jogáveis da aplicação (jogadores e bola) são movidos para as suas posições iniciais;
- O display dos golos faz o correto display do número de golos de ambas as equipas;
- Apenas uma das equipas, tanto no modo Singleplayer como no Multiplayer pode dar início ao jogo, desativando os colisores que impedem a outra equipa de invadir o campo da adversária sem esta ter ainda tocado na bola;
- Após apanhar um "power up", o jogador ou a sua equipa é afetado por as características desse poder;
- No modo Multiplayer, caso um cliente com o mesmo nome se tente conectar a um room com um cliente com esse nome, é enviado de volta ao menu principal;
- No modo Multiplayer, caso um cliente pertença à mesma equipa de um outro cliente que já esteja no mesmo room, esse cliente é enviado de volta ao menu principal;
- A animação da explosão é gerada na mesma posição da localização da bola quando esta entrou na baliza;

4. Conclusões

Em relação aos objetivos inicialmente definidos, ambos os elementos concordam que, de forma geral, todos foram cumpridos.

Contudo, se tivéssemos a oportunidade de implementar melhorias, iríamos incidir sobre a inteligência artificial dos jogadores controlados pelo computador e sobre o modo multiplayer (network).

Assim, visto que a AI atualmente implementada não retrata da melhor forma o comportamento de jogadores reais, seria um desafio interessante melhorar a utilização da framework Libgdx.AI.

Relativamente ao multiplayer, a adição de mais rooms para que mais jogadores possam jogar ao mesmo tempo seria uma mais valia para o jogo em si.

Também gostaríamos de melhorar, em termos estéticos, a interface com o jogador, de forma a tornar o jogo mais belo.

O desenvolvimento do projeto ao longo do semestre foi igualmente distribuído pelos dois elementos do grupo, portanto a contribuição de cada elemento será equitativa.

Em suma, o desenvolvimento deste projeto permitiu ao grupo aplicar e melhorar os conhecimentos sobre as competências adquiridas ao longo do semestre bem como adquirir novas competências em tecnologias desconhecidas.

Aprendeu-se a trabalhar com uma nova frameworke e deram-se os primeiros passos relativamente ao networking (coisa que nunca tínhamos realizado).

Depois da experiência com o libGdx, com certeza que esta framework fará parte das ferramentas que usaremos no futuro para o desenvolvimento de aplicações/jogos.

5. Referências

Libgdx Tutorial Series - http://www.gamefromscratch.com/page/LibGDX-Tutorial-series.aspx.

Libgdx repository - https://github.com/libgdx/libgdx.

KryoNet repository - https://github.com/EsotericSoftware/kryonet