

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 3](#_Toc453082531)

[2. Manual de Utilização 4](#_Toc453082532)

[2.1. Funcionalidades 4](#_Toc453082533)

[2.2. Instalação e arranque do programa 4](#_Toc453082534)

[2.3. Modo de utilização 5](#_Toc453082535)

[2.4. Ficheiros de entrada e de saída 9](#_Toc453082536)

[3. Conceção, Implementação e Teste 10](#_Toc453082537)

[3.1. Estrutura de packages 10](#_Toc453082538)

[3.2. Estrutura de classes 11](#_Toc453082539)

[3.3. Padrões de desenho 16](#_Toc453082540)

[3.4. Mecanismos e comportamentos 16](#_Toc453082541)

[3.5. Ferramentas, bibliotecas e tecnologias 16](#_Toc453082542)

[3.6. Dificuldades 16](#_Toc453082543)

[3.7. Testes unitários 16](#_Toc453082544)

[4. Conclusões 17](#_Toc453082545)

[5. Referências 18](#_Toc453082546)

# Introdução

No âmbito da unidade curricular Laboratório de Programação 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 singleplay ou multiplay, controla um jogador, tendo como objetivo derrotar a equipa adversária.

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](https://github.com/rsafeup/ChampionsField).

# Manual de Utilização

## Funcionalidades

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

Assim, referentemente 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”.

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. A aplicação cliente, inicializada a partir do Andoid, caso haja um host coneta-se ao Servidor e fica em espera até que haja jogadores suficientes ligados para que o “match” se inicie. 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 todas as aplicações Cliente.

## Instalação e arranque do programa

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

No modo Singleplayer, basta selecionar a opção “Run ‘Desktop’” ou “Run ‘Android’” para correr a aplicação. 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.

É também possível executar a aplicação utilizando os ficheiros executáveis. O Servidor é executado através do ficheiro “ChampionsFieldServer.jar” enquanto que o Cliente é executado com recurso ao ficheiro “ChampionsFieldClient.jar”.

No modo Singleplayer não há restrições quanto ao dispositivo utilizado e à rede ao qual está conectado.

Por outro lado, no modo Multiplayer para que ocorrer conexação entre o Cliente e o Servidor é necessário que os respetivos dispositivos estejam conectados à rede “Eduroam” da UP. É também pertinente voltar a frisar que a aplicação Servidor tem que ser, obrigatoriamente, executada em primeiro lugar.

## Modo de utilização

Executando a aplicação “ChampionsFieldClient.jar”, 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.

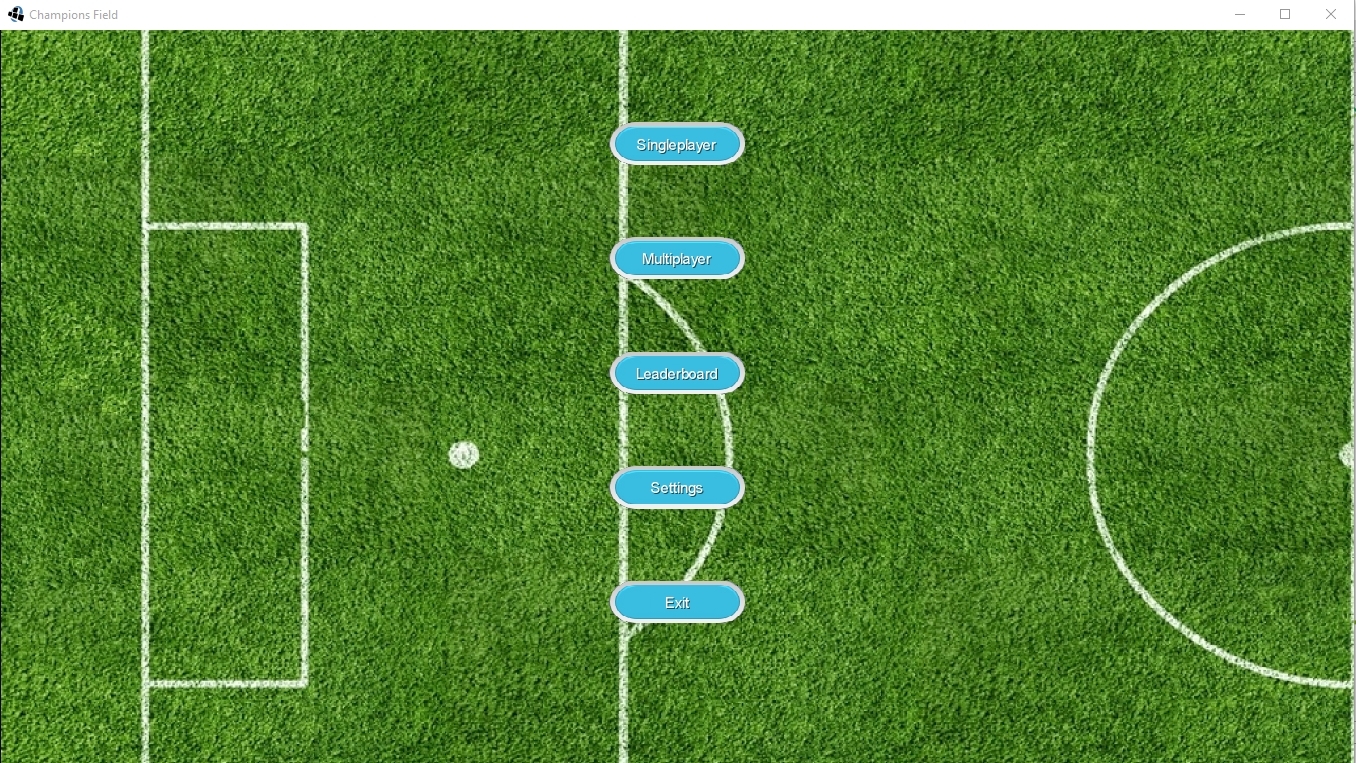


Figura 1 - Menu principal

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

Uma vez que criado 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 cheia.

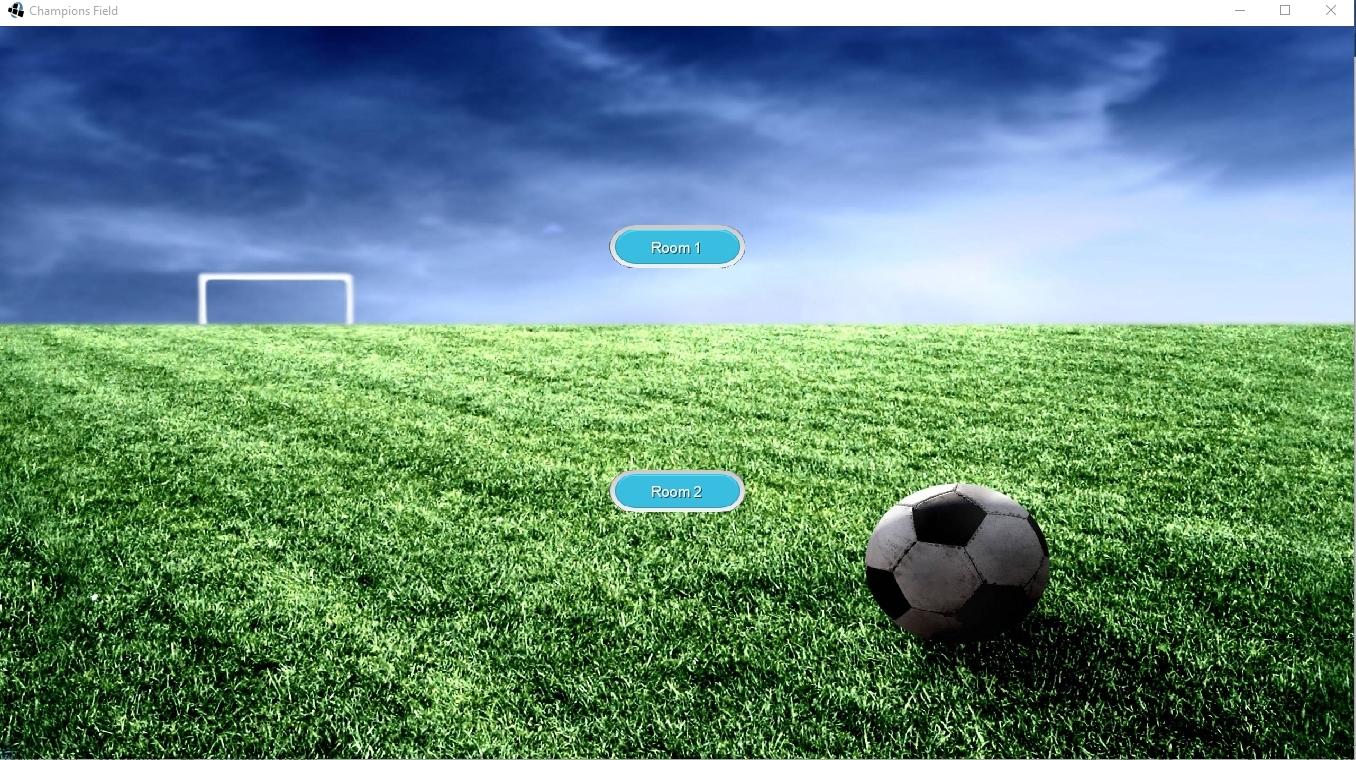


Figura 6 – Multiplayer lobby

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 realizarem updates e por conseguinte a fazer o display sincronizado do jogo.

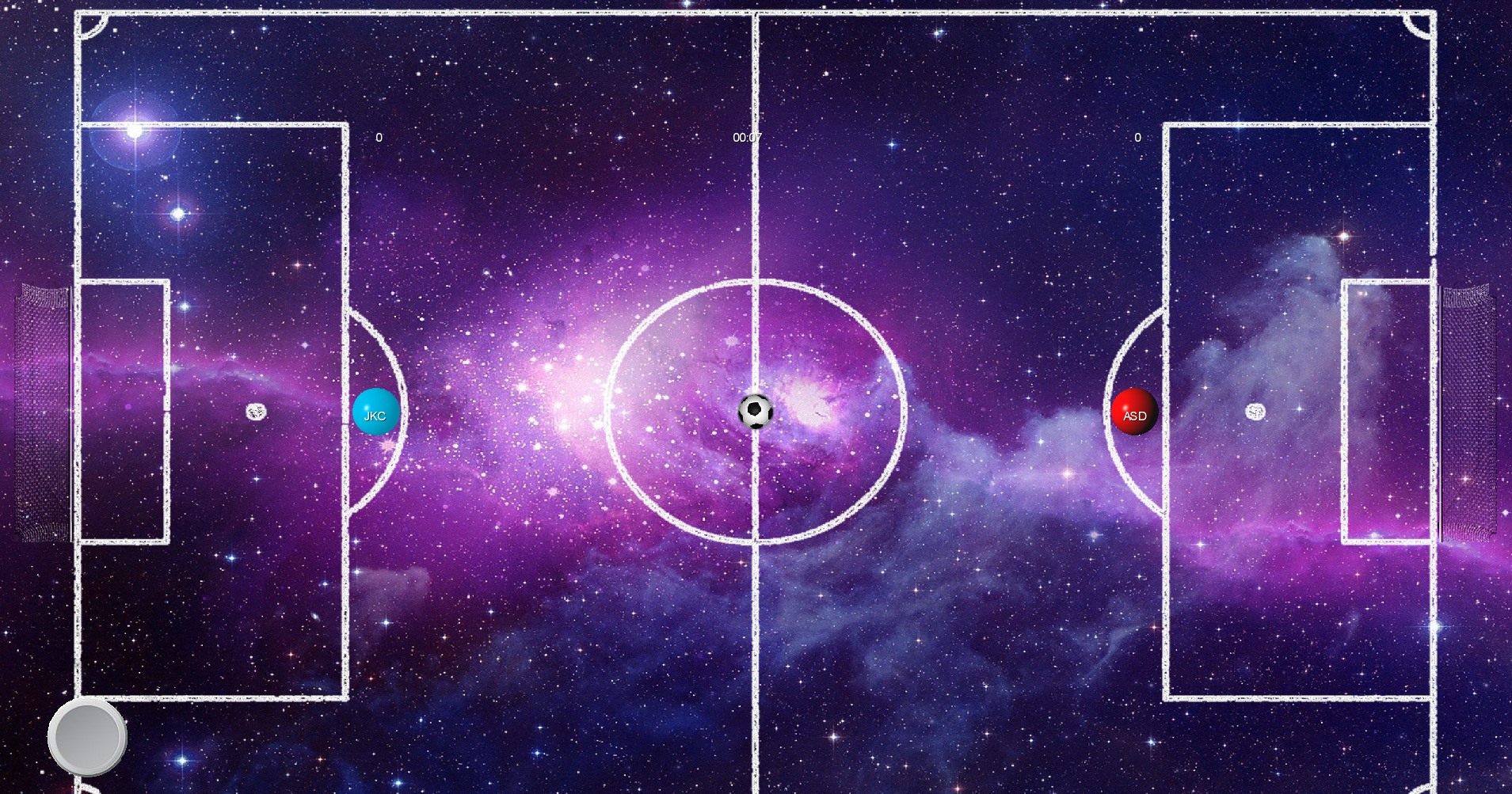


Figura 7 – Multiplayer match

## Ficheiros de entrada e de saída

A aplicação “ChampionsFieldClient.jar” 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.

# Conceção, Implementação e Teste

## Estrutura de packages

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

## 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 | |  |
| GameStateManager | |  |

Package **Logic**

|  |  |
| --- | --- |
| 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 |  |
| Team |  |
| 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 |  |
| SinglePlayState |  |
| MultiPlayState |  |

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 |  |
| MPClient |  |
| MPServer |  |
| MatchInfo |  |
| PlayerInfo |  |

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. |

## Padrões de desenho

## Mecanismos e comportamentos

## 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 ao jogo e abstrair-nos de muitos problemas de física (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 framework KryoNet, uma vez que permitia uma comunicação Cliente/Servidor bastante eficiente e simples.

## Dificuldades

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

A implementação do network com o uso do KryoNet e, por vezes, a falta de documentação sobre esta framework 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.

## Testes unitários

# Conclusões

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

Contudo, se tivessemos 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.

//Faz aqui um paragrafo sobre as melhorias do network, mete palha à vontade.

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, mas bastante .

# 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>