**Faculdade de Engenharia da Universidade do Porto**



**Mestrado Integrado em Engenharia Informática e Computação**

2º ano

**Laboratório de Programação Orientada a Objetos - EIC0020**

Ano Letivo 2014/2015

“ **Air Hockey** ”

Turma 3, Grupo 3

**Estudantes**

Diogo Belarmino Coelho Marques

up201305642@fe.up.pt

Pedro Miguel Pereira de Melo

up201305618@fe.up.pt

Índice

[Índice de figuras 3](#_Toc421217119)

[1. Manual (aplicação *desktop*) 4](#_Toc421217120)

[1.1 Menu inicial 4](#_Toc421217121)

[1.2 Menu “PREFERENCES” 5](#_Toc421217122)

[1.3 6](#_Toc421217123)

[Menu “SINGLEPLAYER” 6](#_Toc421217124)

[1.3 Modo singleplayer 7](#_Toc421217125)

[1.4 Modo multiplayer 9](#_Toc421217126)

[2. Manual (aplicação *Android*) 13](#_Toc421217127)

[2.1 Menu inicial 13](#_Toc421217128)

[15](#_Toc421217129)

[2. Informação do Projeto 16](#_Toc421217130)

[3. Organização/Estrutura do código 17](#_Toc421217131)

[3.1 arkanix.c 17](#_Toc421217132)

[3.2 bitmap.c 17](#_Toc421217133)

[3.3 generate.c 17](#_Toc421217134)

[3.4 input.c 17](#_Toc421217135)

[3.5 level.c 18](#_Toc421217136)

[3.6 menu.c 18](#_Toc421217137)

[3.7 rtc.c 18](#_Toc421217138)

[3.8 score.c 19](#_Toc421217139)

[3.9 serial.c 19](#_Toc421217140)

[3.10 sprite.c 19](#_Toc421217141)

[3.11 ss.c 19](#_Toc421217142)

[3.12 timer.c 20](#_Toc421217143)

[3.13 vbe.c 20](#_Toc421217144)

[3.14 video.c 20](#_Toc421217145)

[4. Diagrama de casos de utilização 21](#_Toc421217146)

[5. Diagrama de pacotes (*packages*) 22](#_Toc421217147)

[6. Diagrama de classes 23](#_Toc421217148)

[6.1 *Package* lpoo.proj2 23](#_Toc421217149)

[6.2 *Package* lpoo.proj2.audio 23](#_Toc421217150)

[6.3 *Package* lpoo.proj2.gui 24](#_Toc421217151)

[6.4 *Package* lpoo.proj2.logic 24](#_Toc421217152)

[6.5 *Package* lpoo.proj2.net 24](#_Toc421217153)

[7. Avaliação 25](#_Toc421217154)

# Índice de figuras

[Figura 1 – menu inicial 4](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975114)

[Figura 2 – botão singleplayer (estado normal e selecionado) 4](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975115)

[Figura 3 – botão multiplayer (estado normal e selecionado) 4](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975116)

[Figura 4 – botões de opção para o controlo da barra 5](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975117)

[Figura 5 – botão “quit” (estado normal e selecionado) 5](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975118)

[Figura 6 - menu modo multiplayer 5](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975119)

[Figura 7 – botão “local game” (estado normal e selecionado) 6](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975120)

[Figura 8 – botão “serial port” (estado normal e selecionado) 6](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975121)

[Figura 9 – botão “<” (estado normal e selecionado) 6](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975122)

[Figura 10 – último nível do modo singleplayer do Arkanix 7](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975123)

[Figura 11 – blocos do jogo 7](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975124)

[Figura 12 – barra do jogador, com respetiva bola 7](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975125)

[Figura 13 – área de jogo 7](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975126)

[Figura 14 – informação do nível a ser jogado 7](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975127)

[Figura 15 – tabela das melhores pontuações 8](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975128)

[Figura 16 – pontuação acumulada pelo jogador 8](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975129)

[Figura 17 – vidas do jogador 8](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975130)

[Figura 18 – ecrã de introdução do nome 8](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975131)

[Figura 19 – partida típica no modo multiplayer 9](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975132)

[Figura 20 – número de vidas e pontuação de um jogador 9](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975133)

[Figura 21 – temporizador do modo multiplayer 10](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975134)

[Figura 22 – ecrã do jogador vencedor 10](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975135)

[Figura 23 – ecrã mostrado numa situação de empate 11](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975136)

[Figura 24 – ecrã mostrado em caso de desistência de um jogador 11](file:///C:\Users\MARQUES\Documents\1.docx#_Toc407975137)

# Introdução

# Manual (aplicação *desktop*)

## 1.1 Menu inicial



O menu inicial do Air Hockey surge quando se inicia a aplicação. Através dele o pode iniciar uma nova partida no modo singleplayer, modificar os controlos da barra ou entrar no menu do modo multiplayer. Existem ao todo quatro botões distintos neste menu:

* **Botão *“SINGLEPLAYER”*** – inicia uma nova partida no modo *singleplayer*.



* **Botão *“MULTIPLAYER”*** – inicia uma nova partida no modo *multiplayer.*



Figura 1 – menu inicial do Air Hockey

* **Botão “PREFERENCES”** - possuí dois estados, *Mouse* (por defeito) e *Keyboard* que indicam qual o dispositivo a utilizar para controlar a barra durante o jogo no modo singleplayer.



* **Botão “CREDITS”**



* **Botão “EXIT”** – encerra a aplicação quando pressionado.



## Menu “PREFERENCES”

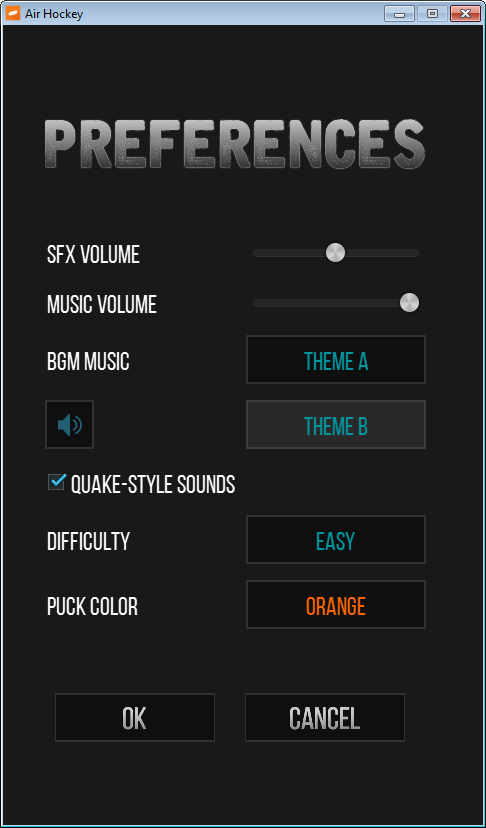
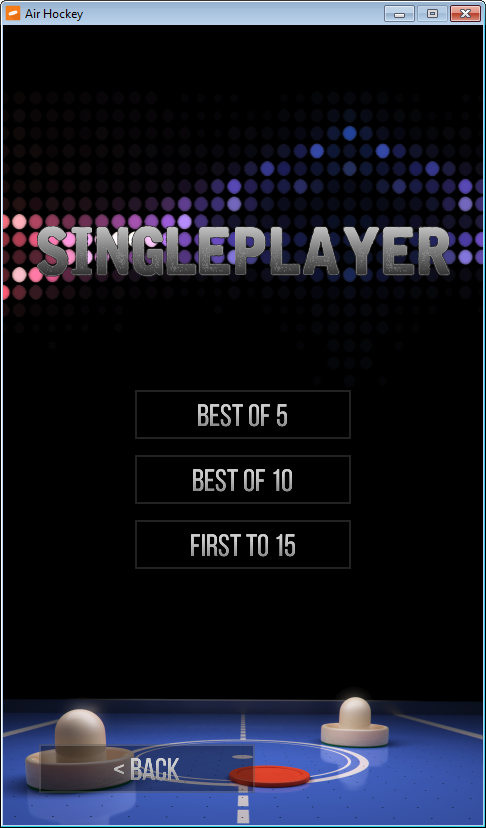
* **SFX VOLUME**
* **MUSIC VOLUME**
* **BGM MUSIC**
* **QUAKE-STYLE SOUNDS**
* **DIFFICULTY**
* **PUCK COLOR**
* **Botão “OK”**
* **Botão “Cancel”**

Figura 2 – menu de configuração do Air Hockey

## 

## Menu “SINGLEPLAYER”



O menu do modo *singleplayer* permite iniciar uma nova partida no modo *singleplayer* (um único jogador); **do tipo local**, em que ambos os jogadores se enfrentam no mesmo computador ou **do tipo em série**, no qual ambos os jogadores se enfrentam em computadores diferentes.

* **Botão “*BEST OF 5*”** - inicia uma nova partida do tipo “melhor de cinco”
* **Botão “*BEST OF 10*”** - inicia uma nova partida do tipo “melhor de dez”
* **Botão “*FIRST TO 15*”** - inicia uma nova partida do tipo “primeiro a chegar aos quinze pontos”

Figura 2 – menu “singleplayer” do Air Hockey

* **Botão “< BACK”** – permite sair do menu *multiplayer*, regressando ao menu inicial.



## 1.3 Modo singleplayer

No modo singleplayer, o jogador terá de ultrapassar três níveis distintos. Para tal necessita de controlar uma barra e uma bola, de forma a destruir todos os blocos presentes em cada um dos níveis. O utilizador possui um total de três vidas, com as quais terá de ultrapassar todos os níveis do Arkanix. O jogador perde uma vida sempre que não conseguir desviar a bola com a barra, deixando-a tocar no fundo da área de jogo.

O jogador possui ainda uma pontuação, que vai aumentando consoante o número de blocos que destruir.

Na figura seguinte pode ver-se o último nível do Arkanix:

**

Figura 3 – último nível do modo singleplayer do Arkanix

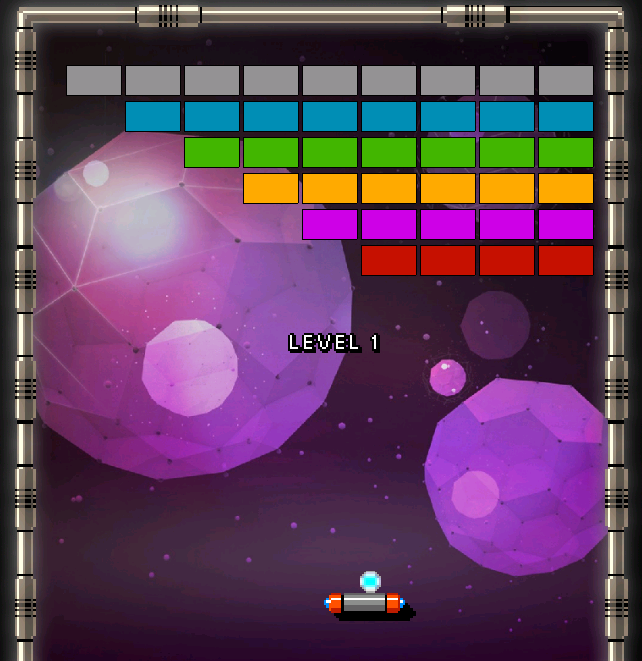
A componente principal deste modo é a **área de jogo** (figura à esquerda), estando toda a ação da partida aqui confinada. No topo da área de jogo encontram-se os **blocos** a eliminar pelo jogador.

Figura 4 – blocos do jogo

Na parte inferior da área de jogo encontra-se a **barra** e a **bola**, inicialmente presa. O utilizador deve libertar a bola na posição que lhe parecer estrategicamente mais conveniente, necessitando apenas de pressionar a tecla *Enter* ou o botão esquerdo do rato consoante esteja a utilizar o teclado ou o rato, respetivamente.



Figura 5 – barra do jogador, com respetiva bola

No centro da área de jogo pode ainda ver-se uma **etiqueta** que indica o número do respetivo nível a ser jogado.

****

Figura 6 – informação do nível a ser jogado



Figura 7 – área de jogo

No lado direito da área de jogo, o jogador pode consultar **a tabela de melhores pontuações** (figura à direita), onde se mostram as melhores classificações no modo singleplayer até ao momento da partida. Ao lado de cada pontuação é possível ver-se o nome do jogador que a obteve bem como a data desse feito.

Figura 8 – tabela das melhores pontuações

Acima da tabela de pontuações o jogador pode consultar a sua **pontuação** em todos os instantes da partida (figura à direita). A pontuação de um jogador incrementa sempre que a bola atinge um dos blocos na área de jogo. Aos blocos de diferentes cores está atribuída uma diferente pontuação (ver anexo).

Figura 9 – pontuação acumulada pelo jogador

Acima da área de jogo, no canto superior esquerdo pode ver-se as restantes **vidas do jogador** (figura à direita). O jogador começa sempre uma partida com três vidas, únicas para os três níveis. O jogador perde uma vida sempre que a bola tocar no fundo da Área de Jogo sem ser defletida pela barra. Quando isto acontece o número de vidas é também atualizado no ecrã.

Figura 10 – vidas do jogador

Se a pontuação do jogador estiver entre as melhores, mesmo que não tenha conseguido superar todos os níveis, terá direito a inserir uma nova pontuação da tabela de melhores pontuações, sendo mostrado o ecrã seguinte.



Figura 11 – ecrã de introdução do nome

O jogador deverá então utilizar o teclado para inserir o seu nome. A pontuação será automaticamente inserida na tabela de pontuações, assim que carregar na tecla *Enter*.

Quando uma partida no modo singleplayer termina o utilizador é redirecionado novamente para o menu iniciar. Deve-se notar que o estado do Botão Controlo assume sempre o valor de defeito (*Mouse*) após cada partida (independentemente do modo jogado).

O jogador pode abandonar o modo singleplayer pressionando a tecla ESC do teclado ou o botão direito do rato caso, se estiver a utilizar, respetivamente, o teclado ou o rato para controlar a barra.

## 1.4 Modo multiplayer

No **modo** **multiplayer**, é possível defrontarem-se dois jogadores.

Assim sendo, existem em simultâneo duas áreas de jogo iguais (*splitscreen*), independentes uma da outra, correspondendo cada uma a um jogador diferente. A mecânica de jogo no modo multiplayer é igual à do modo singleplayer, sendo cada barra controlada do mesmo modo que no modo singleplayer.

Uma novidade deste modo é a presença de um **temporizador** (em segundos) que vai decrementando no decorrer da partida. O tempo de uma partida no modo *multiplayer* é 90 segundos.

Na figura seguinte pode ver-se uma partida do Arkanix no modo multiplayer:

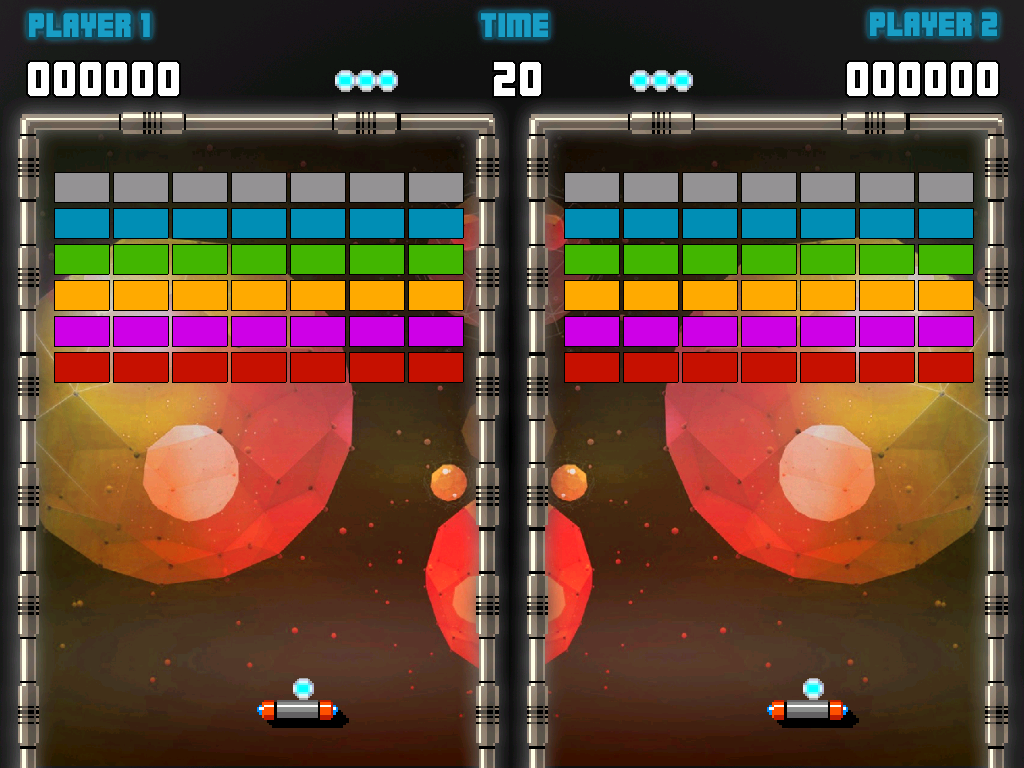


Figura 12 – partida típica no modo multiplayer

Acima de cada uma das áreas de jogo pode ver-se as pontuações de cada jogador bem como o respetivo número de vidas (em semelhança ao modo singleplayer).



Figura 13 – número de vidas e pontuação de um jogador

É possível ver entre as áreas de jogo o temporizador, constantemente atualizado e que delimita o tempo máximo até ao fim da partida.



Figura 14 – temporizador do modo multiplayer

É sagrado vencedor o jogador que no menor tempo conseguir uma maior pontuação (destruindo assim um maior número de blocos).

Sempre que se chega ao fim de uma partida surge o ecrã do vencedor.

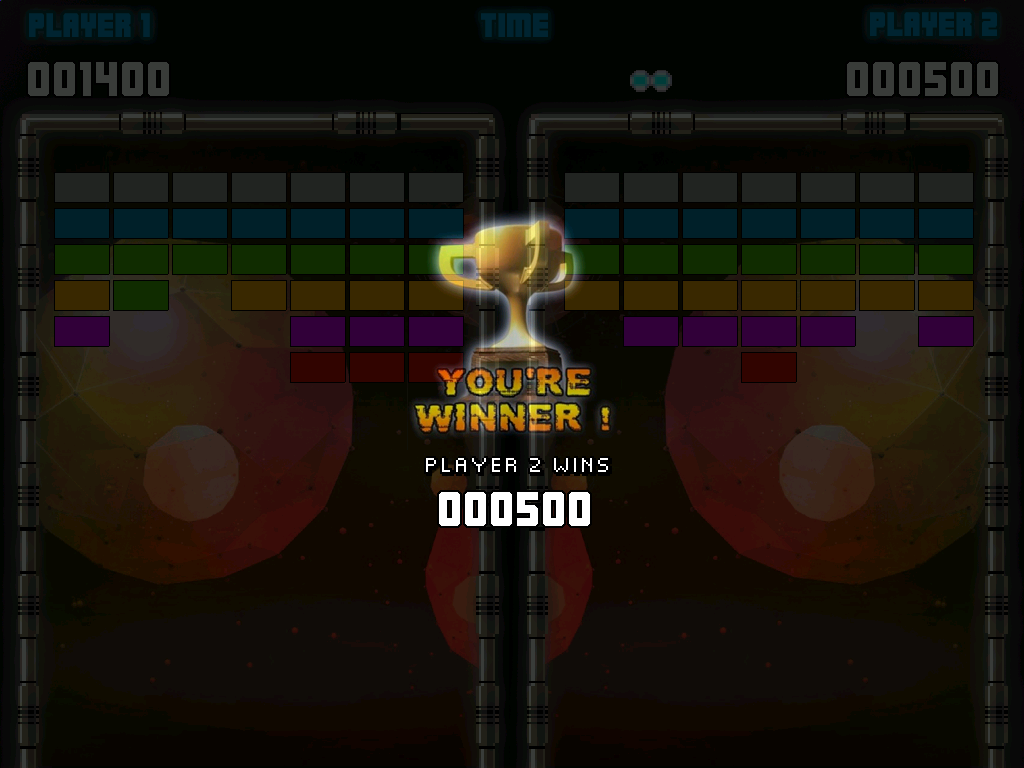


Figura 15 – ecrã do jogador vencedor

Aqui pode ver-se o jogador vencedor e a pontuação obtida por este no decorrer da partida.

Se o temporizador chegar a zero sem que haja vencedor, a partida termina e é mostrado o ecrã de empate (figura em baixo). Ainda em caso de empate, é mostrada a pontuação obtida por ambos os jogadores.

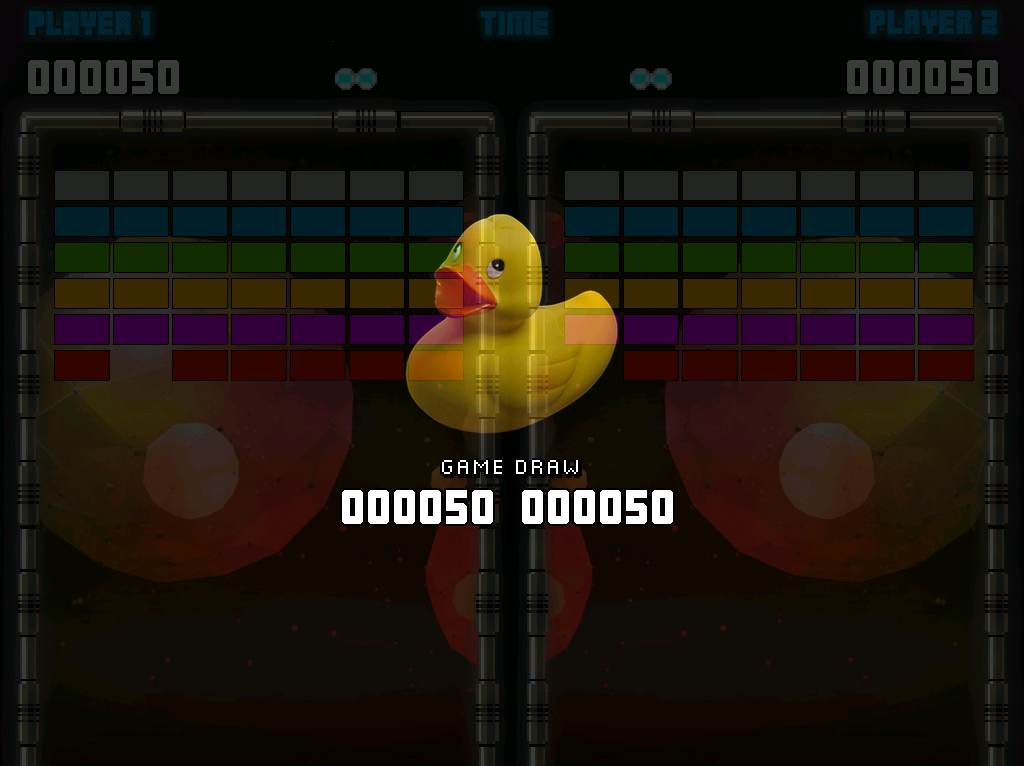


Figura 16 – ecrã mostrado numa situação de empate

Um jogador pode ainda desistir, utilizando os mesmos atalhos definidos na secção anterior. Nesse caso, o adversário sagra-se vencedor e é mostrado um ecrã a avisar do sucedido.

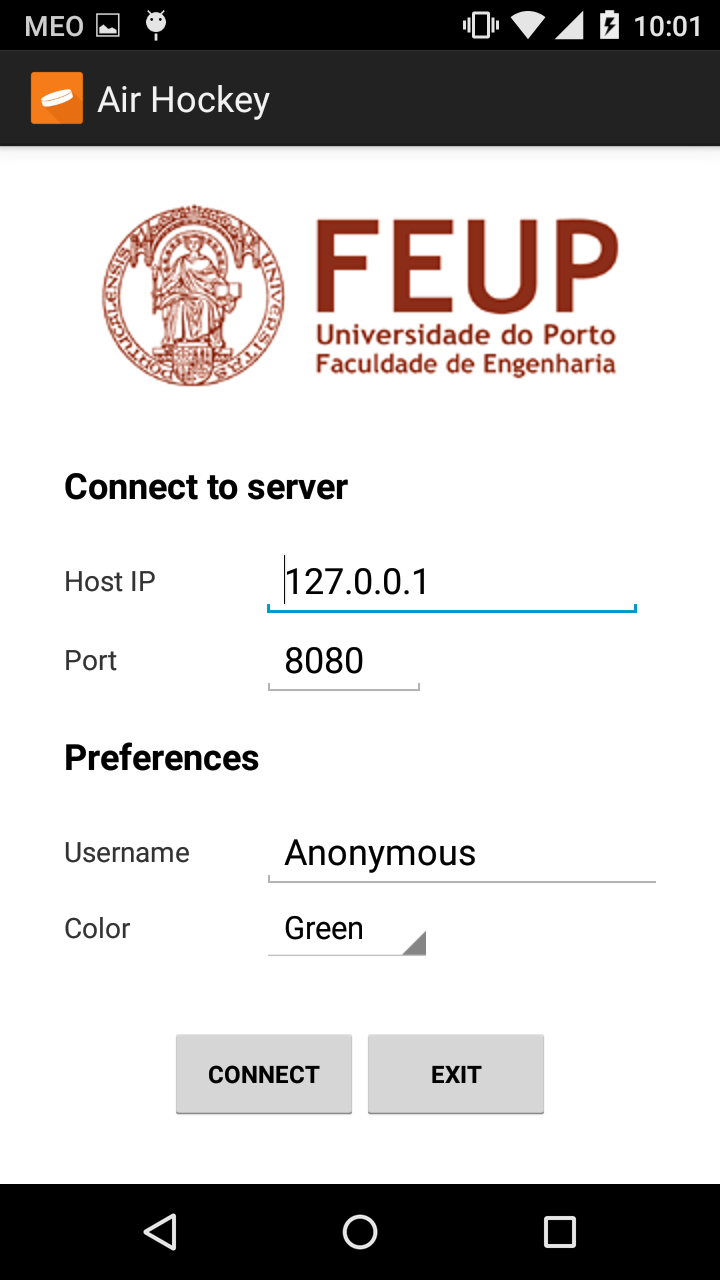


Figura 17 – ecrã mostrado em caso de desistência de um jogador

No modo multiplayer não existe tabela de pontuações pelo que no fim de cada partida os jogadores serão redirecionados para o menu inicial, após 5 segundos, sem que as melhores pontuações obtidas por eles sejam guardadas.

# Manual (aplicação *Android*)

## 3.1 Menu inicial

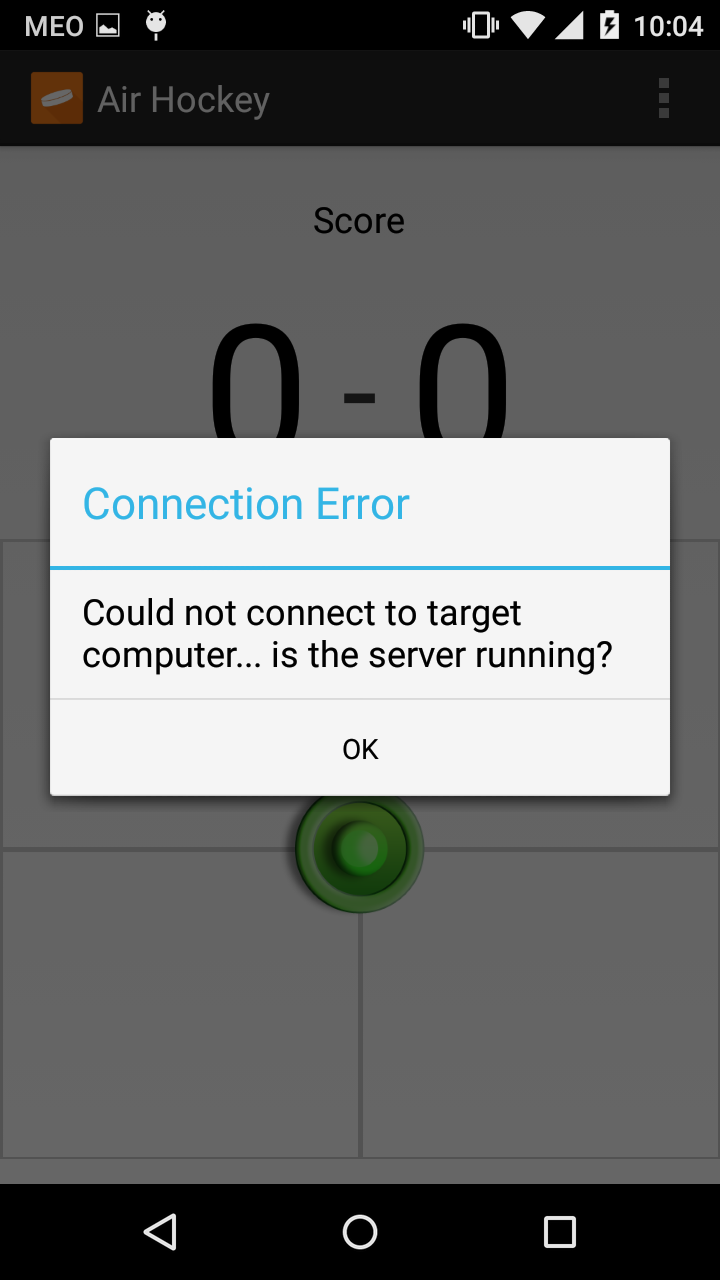
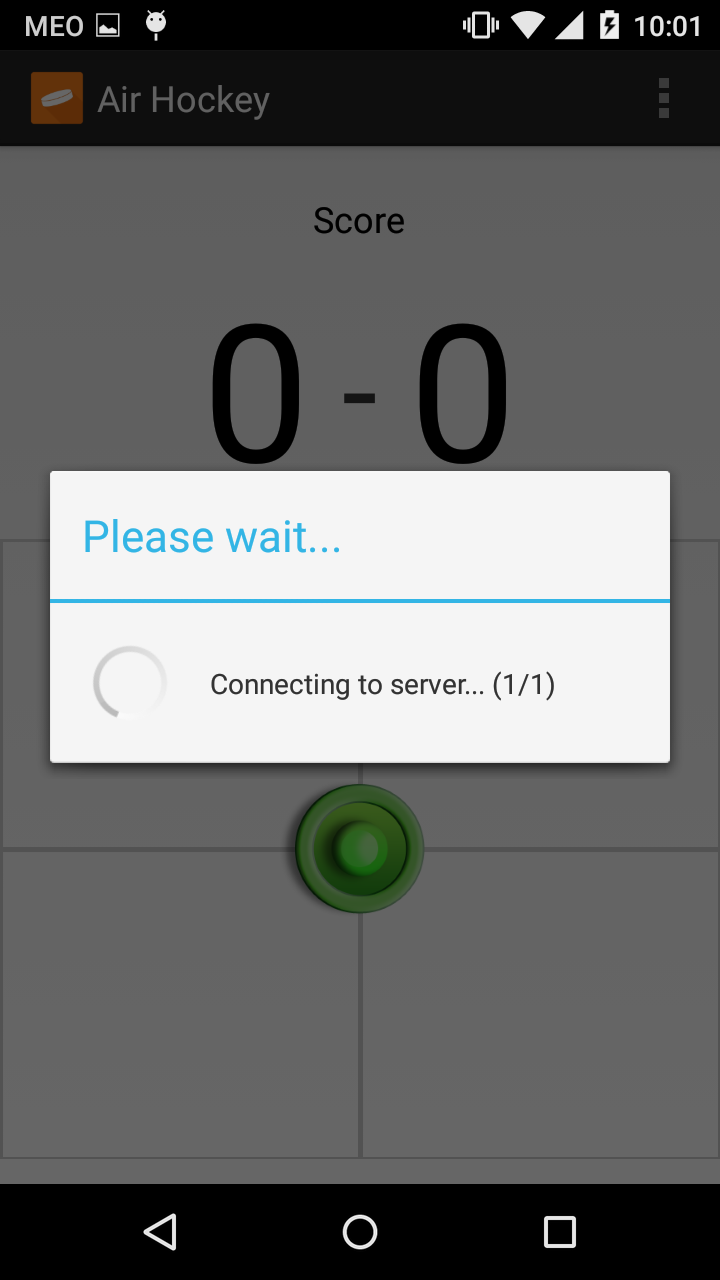


Quando o utilizador inicia a aplicação *Android* do cliente do *Air Hockey bla bla bla...*

* Host IP / Endereço IP – endereço local / *internet* onde o servidor se encontra a correr
* Port / Porta – porta TCP do servidor
* Username / Nome – nome ou alcunha do utilizador
* Color / Cor – cor do *paddle* do utilizador

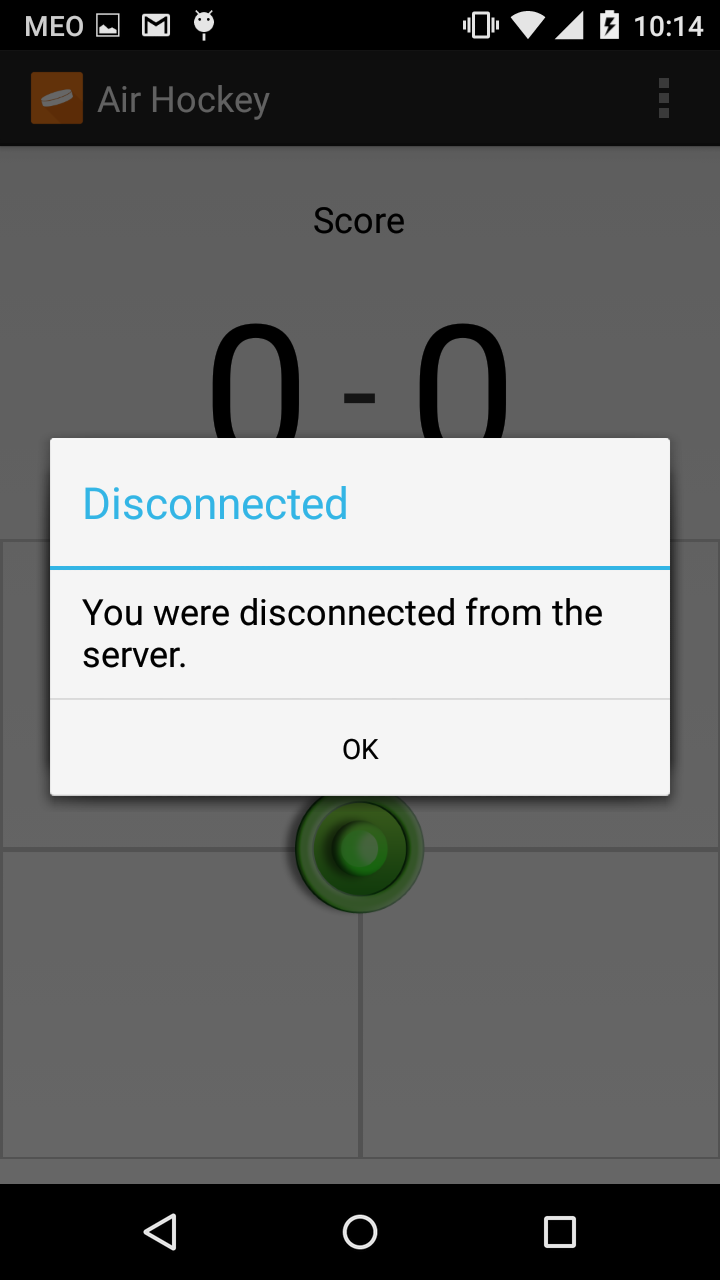
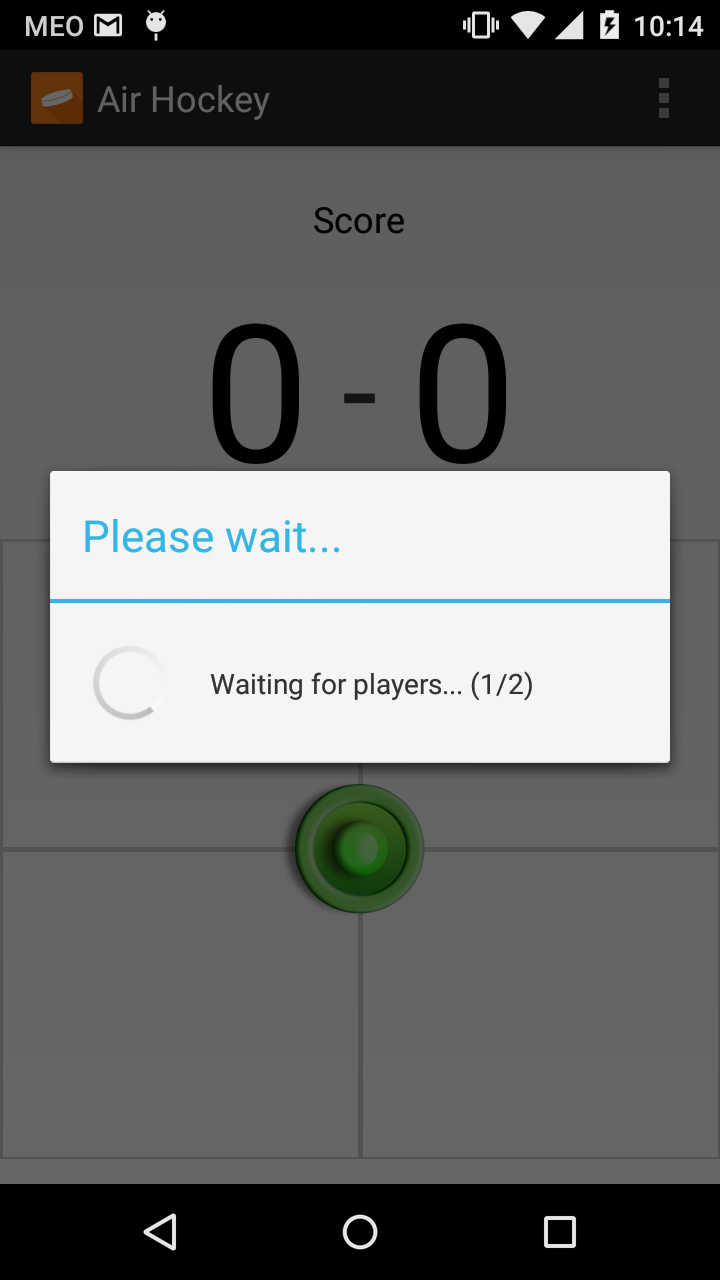
É possível escolher uma das seis cores disponíveis para o *paddle* do utilizador: vermelho (*red*), azul *(blue*), amarelo *(yellow)*, verde *(green)*, laranja *(orange)* e violeta *(purple)*.

Ao carregar no botão “Ligar” é criada uma nova *activity* na aplicação do cliente que representa o campo do jogador.



Após uma tentativa de ligação falhada é apresentada uma janela de diálogo para informar o utilizador de tal acontecimento. Ao pressionar o botão “OK”, o utilizador é encaminhado de volta ao menu inicial da aplicação. Como os dados que introduziu na *activity* principal não são apagados durante a transição para a *activity* do Game, o utilizador pode voltar a conectar-se ao servidor carregando simplesmente no botão “Ligar”.

# 



# D:\Screenshot_2015-06-06-13-26-03.png

# (se o utilizador não tiver nenhuma ligação de rede ativa)

# 2. Informação do Projeto

Na implementação dos menus (inicial e modo *multiplayer*) utilizaram-se os seguintes dispositivos:

|  |  |  |
| --- | --- | --- |
| Dispositivo | Finalidade | Interrupção (S/N) |
| *Rato* | Navegar nos menus e interagir com os botões. | S |
| *Placa de Vídeo* | Desenhar no ecrã o menu, os botões e as animações. | N |
| *Timer* | Atualizar o ecrã. | S |

Na implementação do modo singleplayer utilizaram-se os seguintes dispositivos:

|  |  |  |
| --- | --- | --- |
| Dispositivo | Finalidade | Interrupção (S/N) |
| *Rato* | Controlar a barra e abandonar modo, por opção do jogador. | S |
| *Teclado* | Controlar a barra e abandonar modo, por opção do jogador. | S |
| *Placa de Vídeo* | Desenhar no ecrã o nível, blocos, barra, bola, pontuação, vidas e tabela de pontuações. | N |
| *Timer* | Atualizar o ecrã. | S |
| *RTC* | Obter a data atual para a tabela de pontuações. | N |

Na implementação do modo multiplayer utilizaram-se os seguintes dispositivos:

|  |  |  |
| --- | --- | --- |
| Dispositivo | Finalidade | Interrupção (S/N) |
| *Rato* | Controlar a barra e abandonar modo. | S |
| *Teclado* | Controlar a barra e abandonar modo. | S |
| *Placa de Vídeo* | Desenhar no ecrã o nível, blocos, barra, bola, pontuação, vidas e tabela de pontuações. | N |
| *Timer* | Atualizar o ecrã. | S |
| *RTC* | Controlar o temporizador e definir um alarme para *timeout* das abas de final de jogo. | S |
| *Porta Série* | Enviar/receber informação do teclado (em partidas *multiplayer* através da porta série apenas). | S |

# Concepção e implementação

## 

## 4.1 Bibliotecas, tecnologias e ferramentas utilizadas

**Bibliotecas utilizadas no desenvolvimento do projeto:**

* **LibGDX** (http://libgdx.badlogicgames.com) – a grande vantagem do LibGDX é que não precisamos de construir todo o jogo apenas para Android ou diretamente nele. A biblioteca permite que o desenvolvimento e teste dos projetos criados com esta biblioteca em computador e, com poucas linhas de código, o projeto possa ser também publicado para outras plataformas mais modernas, tais como Android, iOS, HTML5.
* **Kryonet** (https://github.com/EsotericSoftware/kryonet) - *Kryonet* is a Java library that provides a clean and simple API for efficient TCP and UDP client/server network communication using NIO. *Kryonet* uses the [Kryo serialization library](https://github.com/EsotericSoftware/kryo) to automatically and efficiently transfer object graphs across the network. Esta biblioteca tem a vantagem de corer e também permitir a comunicação entre processos.

**IDEs utilizadas no desenvolvimento do projeto:**

* **Eclipse Luna** (https://www.eclipse.org/luna) – IDE principal utilizada no desenvolvimento da aplicação *desktop* e na finalização da aplicação Android
* **Android Development Tools Plugin** (http://developer.android.com/tools/sdk/eclipse-adt.html) **–** extensão para o *Eclipse* que permite o desenvolvimento de aplicações *Android*
* **Android Studio** (https://developer.android.com/sdk/index.html) – IDE oficial utilizada no desenvolvimento de aplicações Android

**Ferramentas utilizadas no desenvolvimento do projeto:**

* **Hiero** (https://github.com/libgdx/libgdx/wiki/Hiero) **–** utilizada na criação das *bitmap fonts* para os menus, apresenta funcionalidades mais avançadas relativamente à ferramenta *BMFont*, tendo sido utilizada sobretudo para aplicar efeitos de degradê e contorno nos tipos de letra. Integrada na distribução LibGDX.
* **TexturePacker** (https://github.com/libgdx/libgdx/wiki/Texture-packer) – utilizada no empacotamento das texturas dos controlos da *interface gráfica* num único ficheiro e criação do respetivo dicionário de texturas. Integrada na distribuição LibGDX.
* **BMFont** (http://www.angelcode.com/products/bmfont) **–** para a criação de *bitmap fonts*, ferramenta mais antiga e com menos funcionalidades do que a *Hiero* anteriormente referida, foi utilizada na criação dos tipos de letra de cor
* **Paint.NET** (http://www.getpaint.net/index.html) **–** utilizada na criação das imagens do jogo, para a *interface* gráfica dos menus da aplicação *desktop* e na adaptação dos *drawables* da aplicação Android aos vários tamanhos de ecrã e diferentes densidades de *pixels* (MDPI, HDPI, XHDPI)

## 4.2 Padrões de desenho

|  |  |
| --- | --- |
| Singleton | * AudioManager * B * C |
| Abstract Factory | * A * B * C |
| Strategy | * CPUPaddle * B * C |
| Visitor | * CollisionProcessor * B * C |
| Observer | * ServerListener * ClientListener * C |
| Facade | * AudioManager * B * C |
| State | * GameRunningState * WaitingState |
| Template Method | * GameRules * B * C |

## 4.3 Testes unitários

Como se trata de um jogo marioritariamente gráfico, assente na simulação física, não foram implementados testes unitários automáticos em *JUnit*. Um dos casos de teste possível seria simular uma partida em que ambos os jogadores marcavam golos e testava () situações de vitória, empate e derrota.

No entanto, foram realizados testes manuais ao funcionamento e estabilidade do programa, tanto na aplicação *desktop* como na aplicação do cliente para dispositivos *Android*. Segue-se uma lista dos testes que foram realizados:

* A
* B
* C

A aplicação Android foi testada nos seguintes dispositivos, com diferente *hardware*, resoluções de ecrã e *pixel densities*:

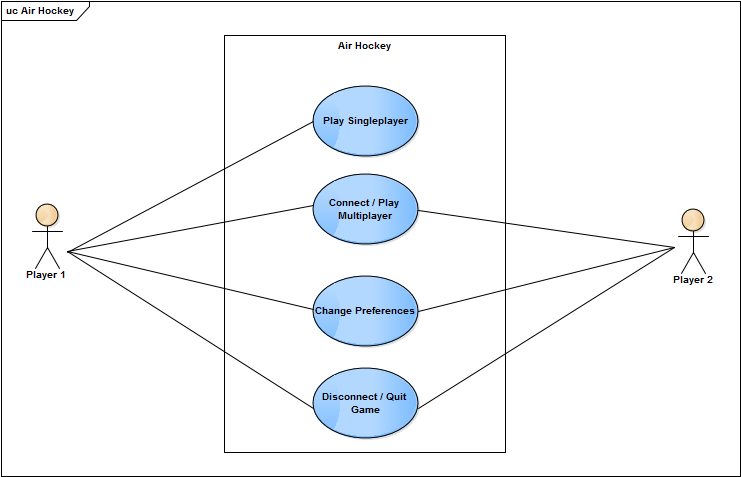
* Motorola Moto G (XHDPI, resolução de ecrã 720x1280, densidade 326 dpi)
* Huawei Ideos X5 (HDPI, resolução de ecrã 480x800, densidade 242 dpi)
* Samsung Galaxy S Duos 2 (HDPI, resolução de ecrã 480x800, densidade 233 dpi)

## Mecanismos de comunicação

A comunicação em rede implementada consiste na troca de objetos entre cliente e servidor. Estes são objetos de classes relativamente minimalistas, tendo sido desenvolvidas especificamente para este propósito: transferir a menor quantidade de informação entre os dispositivos ligados, e deixar as tarefas de processar o movimento ...

* GameOver – utilizado pelo servidor para informar todos os clientes que a partida chegou ao fim e quem se sagrou o vencedor da mesma; este objecto não requer parâmetros
* PlayerLogin – utilizado pelo cliente para informar o servidor que este se tentou ligar ao mesmo; é enviado ao servidor o nome/alcunha do jogador e a preferência desta para a cor do *paddle*
* PlayerConnected – utilizado pelo servidor para informar os restantes clientes que um novo cliente se juntou ao servidor
* PlayerDisconnected – utilizado pelo servidor para informar os restantes clientes que um cliente se desligou do servidor, abandonando a partida
* ServerFull – utilizado pelo servidor para informar o cliente que se tentou conectar de que o servidor atingiu a sua capacidade máxima (2 jogadores); é enviado o número máximo de jogadores permitidos no servidor
* UpdatePaddle – utilizado pelo cliente para informar o servidor da nova posição do *paddle* correspondente a esse cliente; são enviadas as coordenadas *x* e *y* da nova posição
* UpdateScore – utilizado pelo servidor para informar todos os clientes de uma situação de golo e atualizar a respetiva pontuação; são enviadas as pontuações de ambos os jogadores, *p1Score* e *p2Score* sempre que o servidor recebe este pedido

# Diagrama de casos de utilização

****

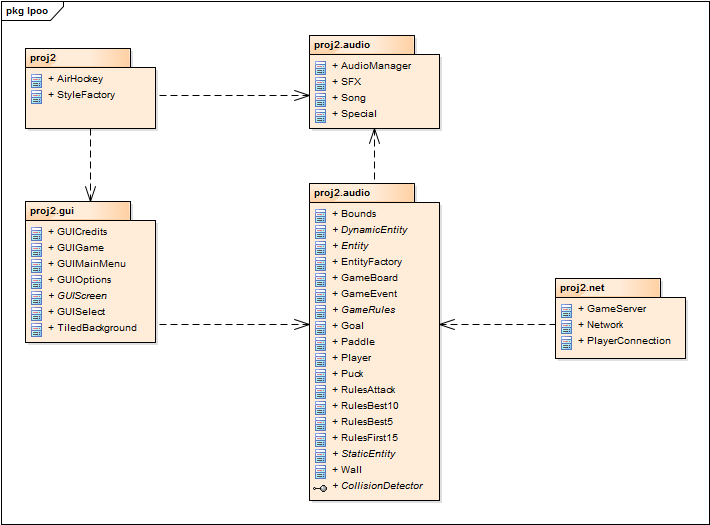
***Player* 1**

* Iniciar uma nova partida no modo *singleplayer*
* Iniciar uma nova partida no modo *multiplayer* (criar um servidor e/ou juntar-se a uma partida já existente)
* Alterar a configuração global (volume dos sons, volume da música, música de fundo, cor da bola)
* Alterar a configuração do modo *singleplayer* (dificuldade do adversário)
* Alterar o perfil do jogador (alcunha e cor do *paddle*)
* Abandonar uma partida no modo *multiplayer*, desconectando-se do servidor
* Encerrar a aplicação

***Player* 2**

* Iniciar uma nova partida no modo *multiplayer* (criar um servidor e/ou juntar-se a uma partida já existente)
* Alterar o perfil do jogador (alcunha e cor do *paddle*)
* Abandonar uma partida no modo *multiplayer*, desconectando-se do servidor

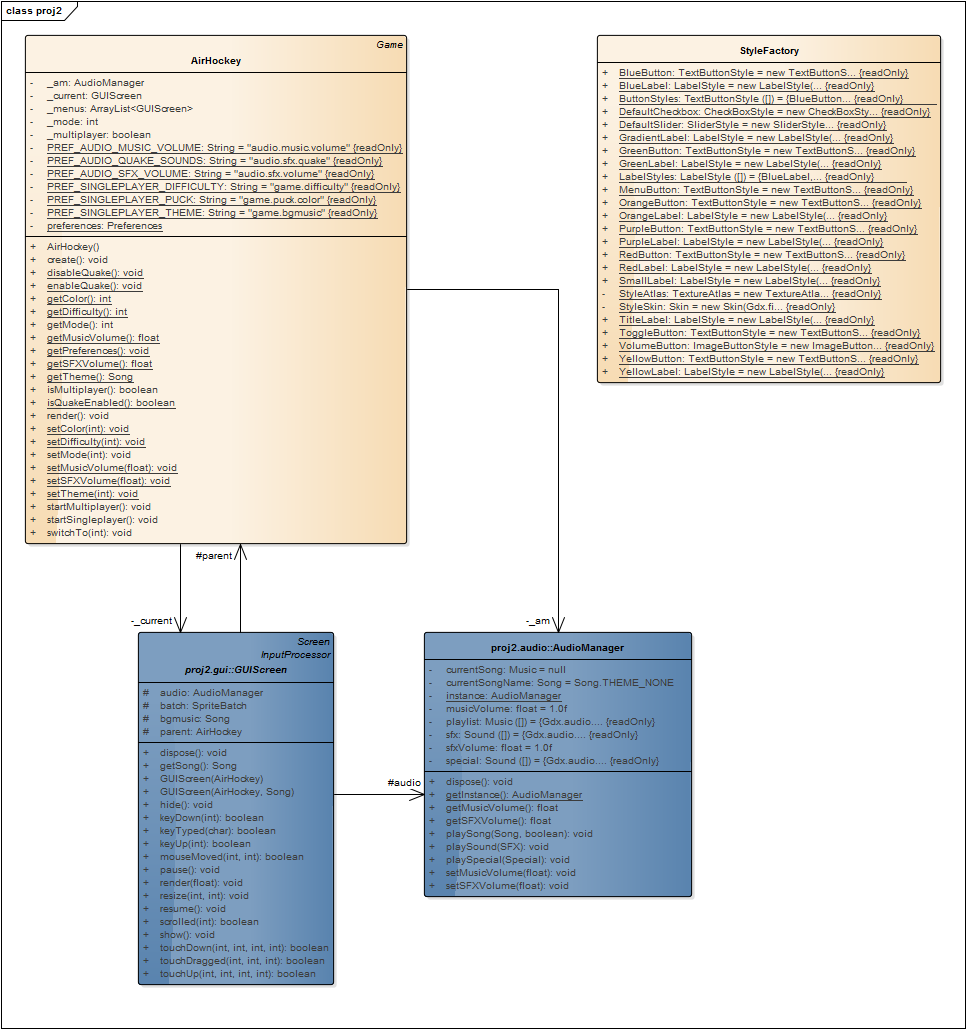
# 6. Diagrama de pacotes (*packages*)



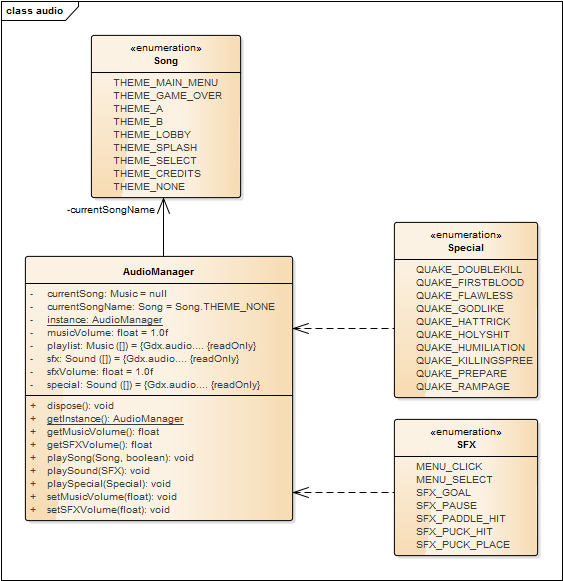
|  |  |
| --- | --- |
| *Package* | Descrição |
| proj2 |  |
| proj2.audio |  |
| proj2.gui |  |
| proj2.logic |  |
| proj2.net |  |

# 7. Diagrama de classes

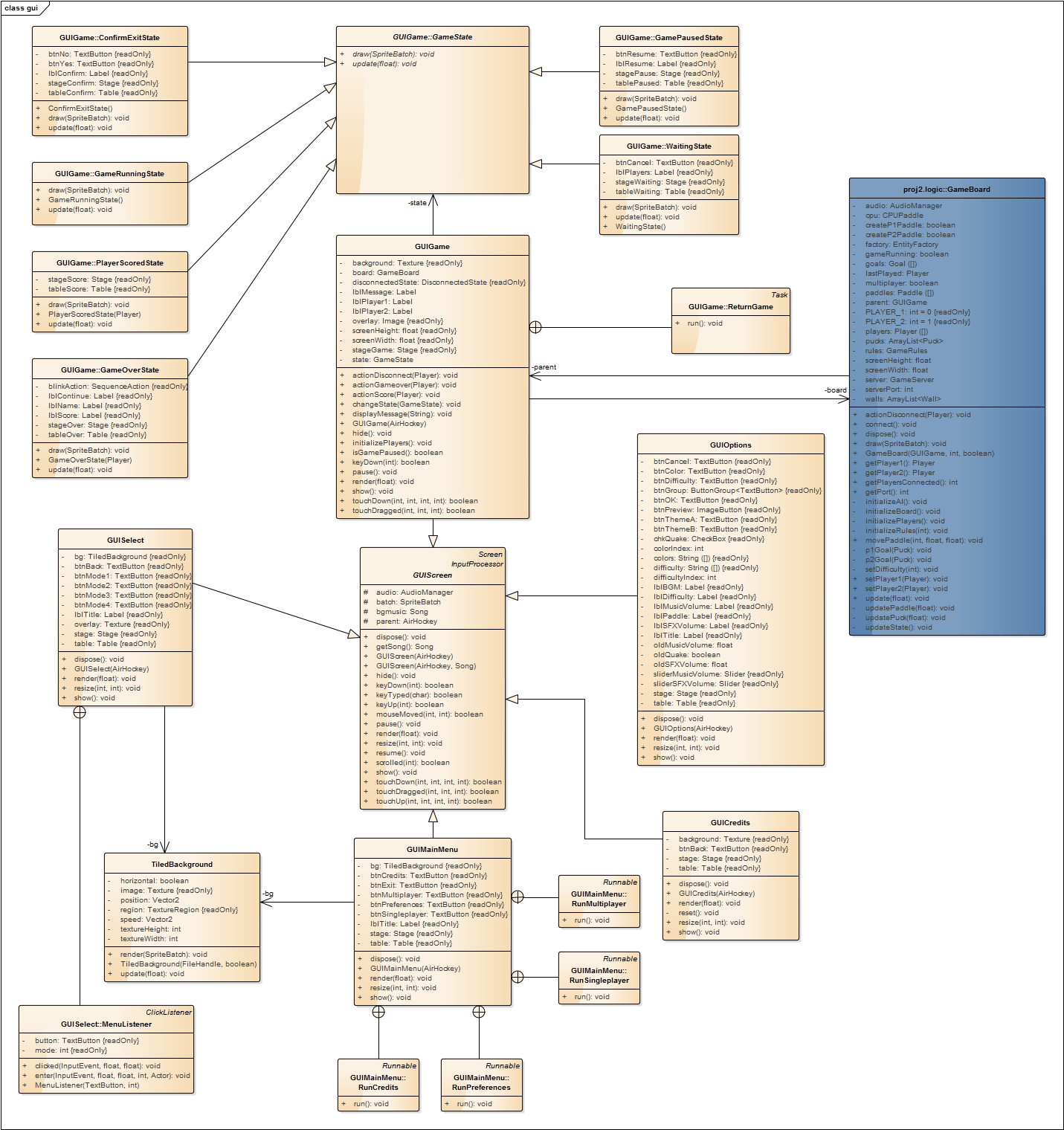
## 7.1 *Package* lpoo.proj2



## 7.2 *Package* lpoo.proj2.audio



## 7.3 *Package* lpoo.proj2.gui



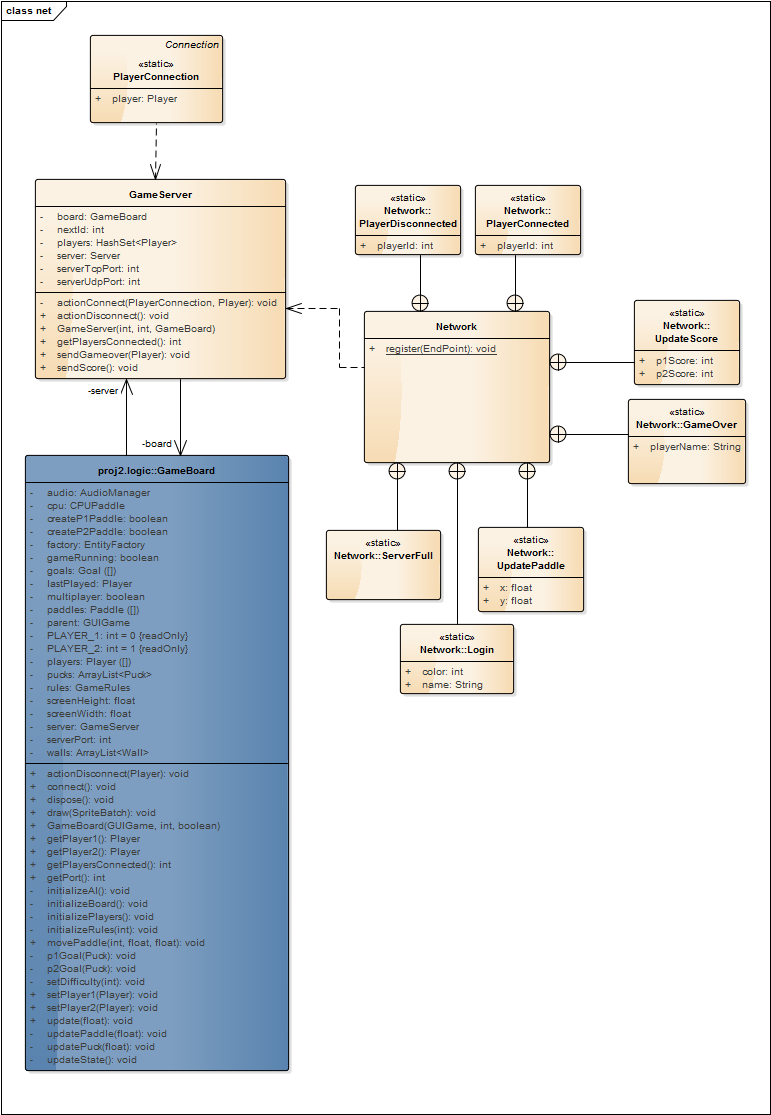
## 7.4 *Package* lpoo.proj2.logic

A porta série implementada transmite e recebe dados a uma taxa de bits fixa de 2400 bauds durante todo o seu tempo de funcionamento. A transmissão é feita por varrimento (*polling*), isto é, é verificado o estado do *Transmitter Holding Register*, e caso este esteja vazio (sem dados para enviar nem receber) e tenha ocorrido uma interrupção do teclado, envia o respetivo *scancode* para o computador de destino.

Quando o utilizador entra no modo *multiplayer* em série, o jogo fica em estado de espera (é apresentada a mensagem “*WAITING FOR PLAYERS...”* até receber um pacote específico do computador ao qual está ligado. O computador do adversário, por sua vez, também envia o mesmo pacote enquanto não receber uma resposta. Os pacotes são enviados a um período constante, a cada 0.5 segundos. Asosm, os jogadores não precisam de entrar simultâneamente nomodo *multiplayer*em série e um dos jogadores pode entrar no jogo a qualquer momento.

A receção dos dados enviados é feita no destino através de interrupções geradas pela *UART*. O *handler* recebe a interrupção, verifica o tipo de interrupção (foram ativadas apenas interrupções do tipo *Reciever Data Available* e *Line Status Register*), e processa o *scancode* recebido. É chamada a função *paddle\_keyboard\_update()* para atualizar a posição da barra do adversário e as teclas especiais que pressionou (*Enter*, barra de espaços, *Esc*).

## 7.5 *Package* lpoo.proj2.net



# 7. Conclusão

Embora tenham sido explicados os vários excertos de código em linguagem C fornecidos para serem utilizados nos laboratórios e no projeto final; estes excertos, na sua maioria "uma orientação para o que deveria ser feito" e por vezes bastante genéricos, foram abordados de um ponto de vista pouco prático.

Em consequência, deparamo-nos com a dificuldade de passar da "teoria à prática", não por não se conhecer a matéria, mas porque mesmo acompanhados dos diapositivos das aulas teóricas, dos apontamentos aí tirados e dos guiões das aulas laboratoriais não se sabia por onde começar a escrita do código (por nunca ter havido um verdadeiro contacto com a parte prática).

Assim sendo, sugerimos que o professor construa com os alunos pequenos excertos de código na aula e que os execute, podendo até mostrar propositadamente formas incorretas de escrever o código e explorar as consequências que daí advêm. A oportunidade de contactar com esta componente mais prática seria uma grande ajuda na realização dos trabalhos práticos.