

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
1. Manual (aplicação <i>desktop</i> ).....	5
1.1 Menu inicial .....	5
1.2 Menu “PREFERENCES” .....	6
1.3.....	7
Menu “SINGLEPLAYER” .....	7
1.3 Modo singleplayer .....	8
1.4 Modo multiplayer.....	10
2. Manual (aplicação <i>Android</i> ).....	14
2.1 Menu inicial .....	14
.....	16
2. Informação do Projeto .....	17
3. Organização/Estrutura do código.....	<b>Error! Bookmark not defined.</b>
3.1 arkanix.c .....	18
3.2 bitmap.c.....	<b>Error! Bookmark not defined.</b>
3.3 generate.c.....	<b>Error! Bookmark not defined.</b>
3.4 input.c .....	<b>Error! Bookmark not defined.</b>
3.5 level.c.....	<b>Error! Bookmark not defined.</b>
3.6 menu.c.....	<b>Error! Bookmark not defined.</b>
3.7 rtc.c .....	<b>Error! Bookmark not defined.</b>
3.8 score.c.....	<b>Error! Bookmark not defined.</b>
3.9 serial.c.....	<b>Error! Bookmark not defined.</b>
3.10 sprite.c.....	<b>Error! Bookmark not defined.</b>
3.11 ss.c.....	<b>Error! Bookmark not defined.</b>
3.12 timer.c.....	<b>Error! Bookmark not defined.</b>
3.13 vbe.c.....	<b>Error! Bookmark not defined.</b>
3.14 video.c.....	<b>Error! Bookmark not defined.</b>
4. Diagrama de casos de utilização .....	20
5. Diagrama de pacotes ( <i>packages</i> ) .....	20
6. Diagrama de classes.....	22
6.1 <i>Package</i> lpoo.proj2.....	22
6.2 <i>Package</i> lpoo.proj2.audio.....	23
6.3 <i>Package</i> lpoo.proj2.gui .....	23
6.4 <i>Package</i> lpoo.proj2.logic.....	25

6.5	<i>Package lpoo.proj2.net</i> .....	25
7.	Avaliação.....	26

## Índice de figuras

Figura 1 - menu inicial .....	5
Figura 2 - botão singleplayer (estado normal e selecionado) .....	Error! Bookmark not defined.
Figura 3 - botão multiplayer (estado normal e selecionado) .....	Error! Bookmark not defined.
Figura 4 - botões de opção para o controlo da barra .....	Error! Bookmark not defined.
Figura 5 - botão “quit” (estado normal e selecionado) .....	Error! Bookmark not defined.
Figura 6 - menu modo multiplayer .....	Error! Bookmark not defined.
Figura 7 - botão “local game” (estado normal e selecionado) .....	7
Figura 8 - botão “serial port” (estado normal e selecionado) ....	Error! Bookmark not defined.
Figura 9 - botão “<” (estado normal e selecionado) .....	Error! Bookmark not defined.
Figura 10 - último nível do modo singleplayer do Arkanix .....	8
Figura 11 - blocos do jogo .....	9
Figura 12 - barra do jogador, com respetiva bola .....	9
Figura 13 - área de jogo .....	9
Figura 14 - informação do nível a ser jogado .....	9
Figura 15 - tabela das melhores pontuações .....	9
Figura 16 - pontuação acumulada pelo jogador .....	9
Figura 17 - vidas do jogador .....	9
Figura 18 - ecrã de introdução do nome .....	10
Figura 19 - partida típica no modo multiplayer .....	11
Figura 20 - número de vidas e pontuação de um jogador .....	11
Figura 21 - temporizador do modo multiplayer .....	11
Figura 22 - ecrã do jogador vencedor .....	12
Figura 23 - ecrã mostrado numa situação de empate .....	12
Figura 24 - ecrã mostrado em caso de desistência de um jogador .....	13

# 1. Introdução

## 2. Manual (aplicação *desktop*)

### 1.1 Menu inicial



Figura 1 - menu inicial do Air Hockey

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*.



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



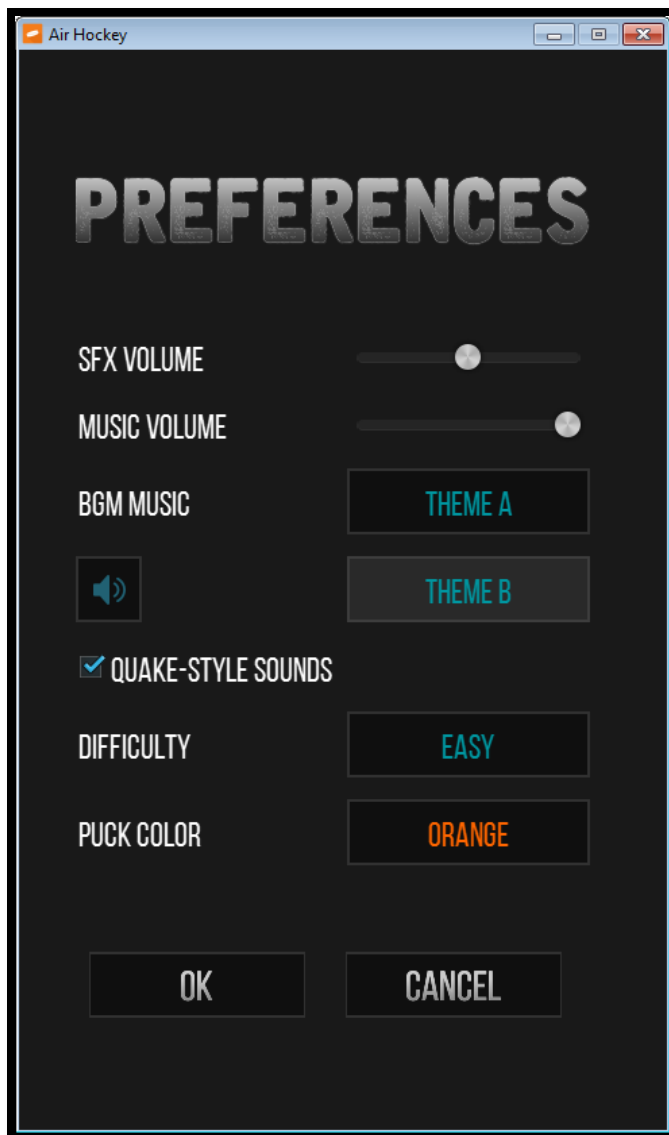
- ✓ Botão “CREDITS”



- ✓ Botão “EXIT” - encerra a aplicação quando pressionado.



## 1.2 Menu “PREFERENCES”

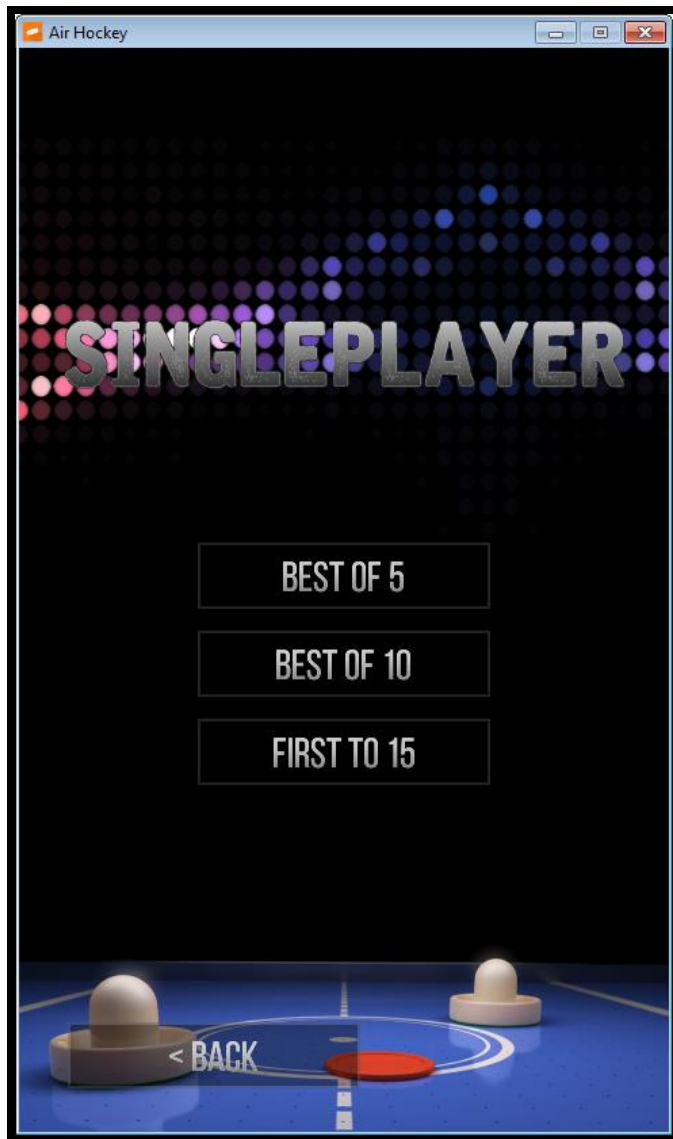


- ✓ SFX VOLUME
- ✓ MUSIC VOLUME
- ✓ BGM MUSIC
- ✓ QUAKE-STYLE SOUNDS
- ✓ DIFFICULTY
- ✓ PUCK COLOR
- ✓ Botão “OK”
- ✓ Botão “Cancel”

Figura 2 - menu de configuração do Air Hockey

### 1.3

#### 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”

**BEST OF 5**

✓ Botão “*BEST OF 10*” - inicia uma nova partida do tipo “melhor de dez”

**BEST OF 10**

✓ Botão “*FIRST TO 15*” - inicia uma nova partida do tipo “primeiro a chegar aos quinze pontos”

**FIRST TO 15**

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.

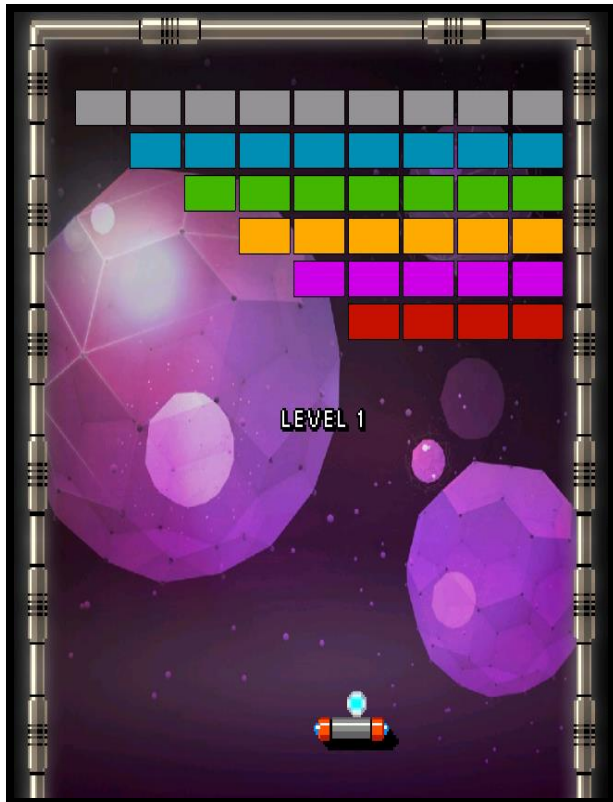


Figura 7 - área de jogo

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.

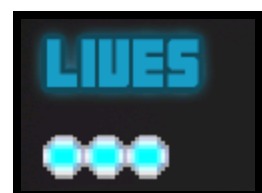


Figura 10 - vidas do jogador



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.

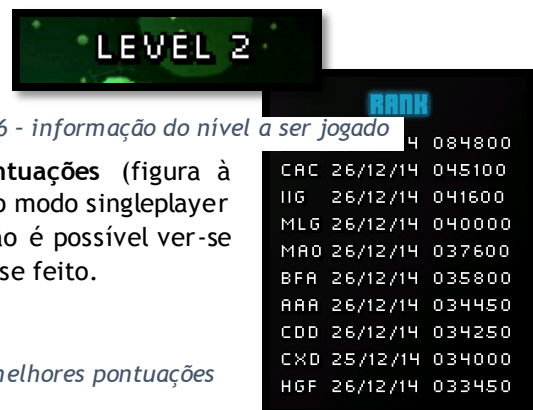


Figura 6 - informação do nível a ser jogado

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.

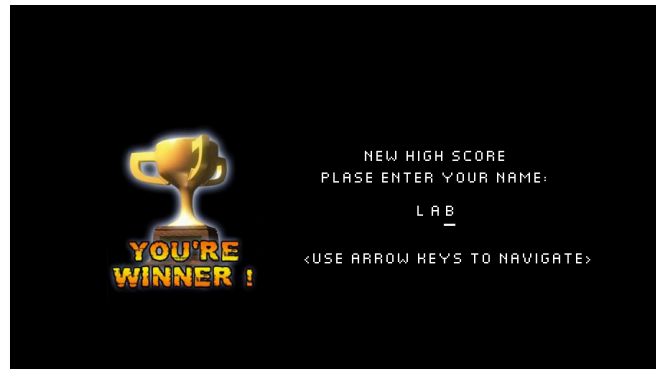


Figura 11 - ecrã de introdução do nome

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.

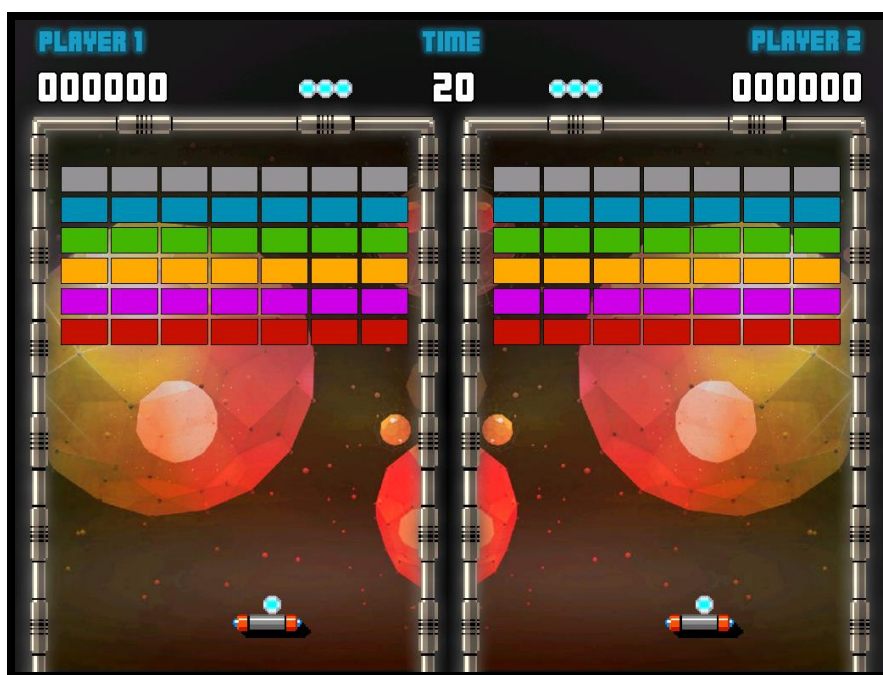
## 1.4 Modo multiplayer

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.

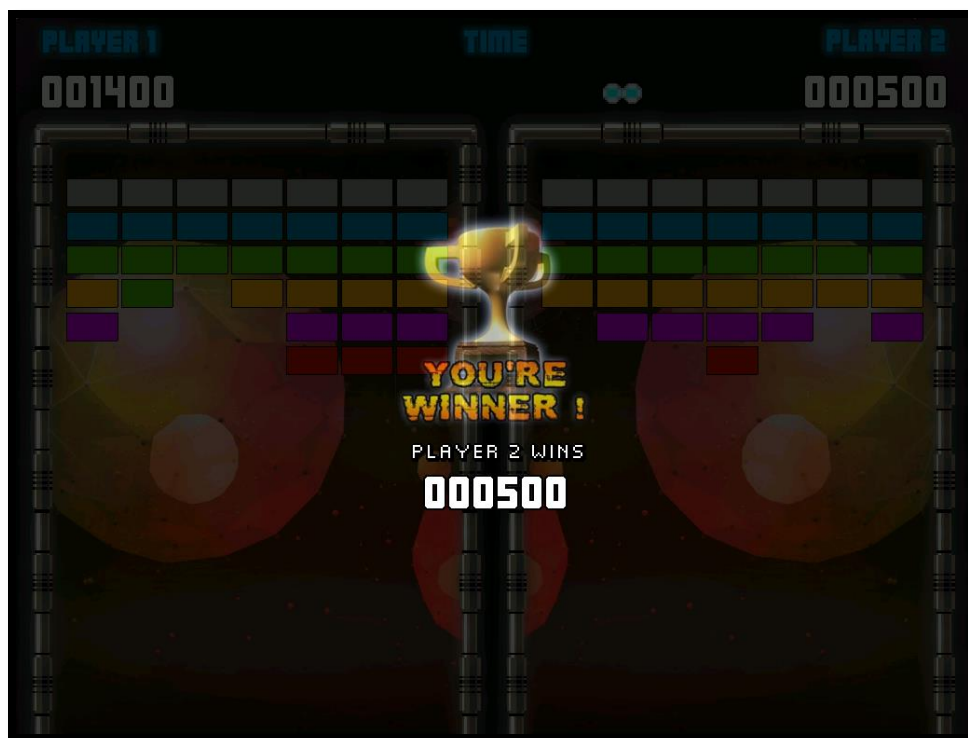


Figura 15 - ecrã do jogador vencedor

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.

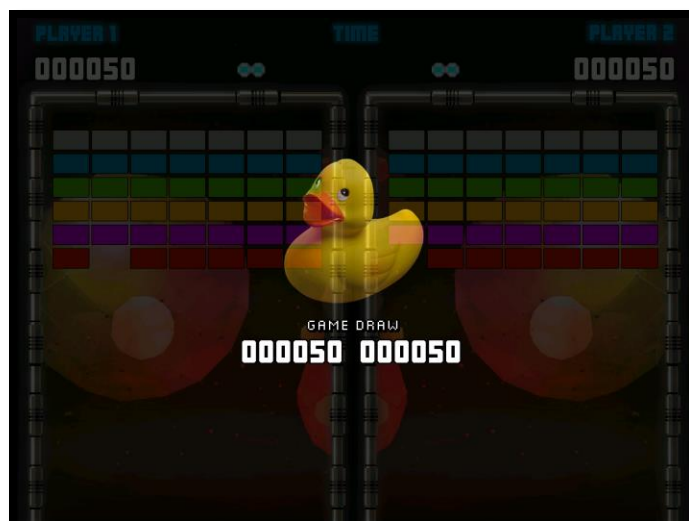
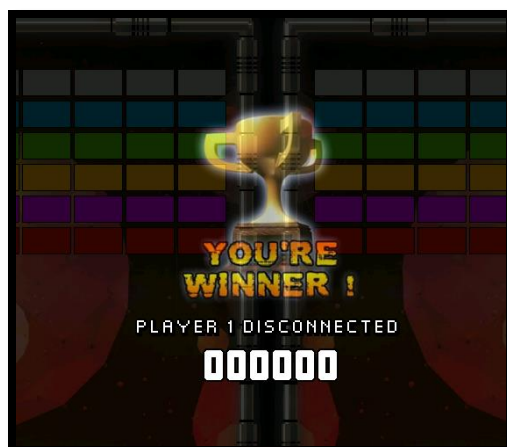


Figura 16 - ecrã mostrado numa situação de empate

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.

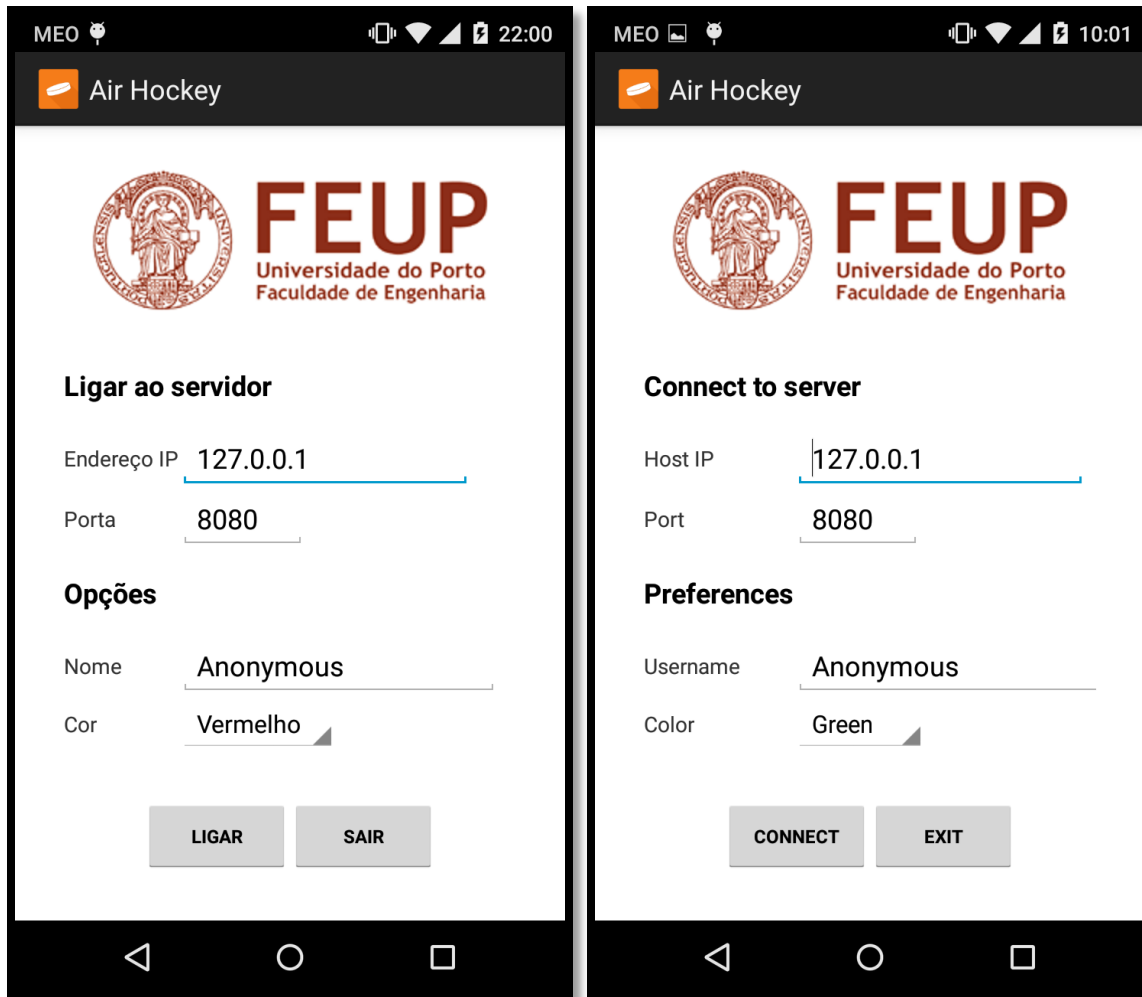


*Figura 17 - ecrã mostrado em caso de desistência de um jogador*

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.

### 3. Manual (aplicação *Android*)

#### 3.1 Menu inicial

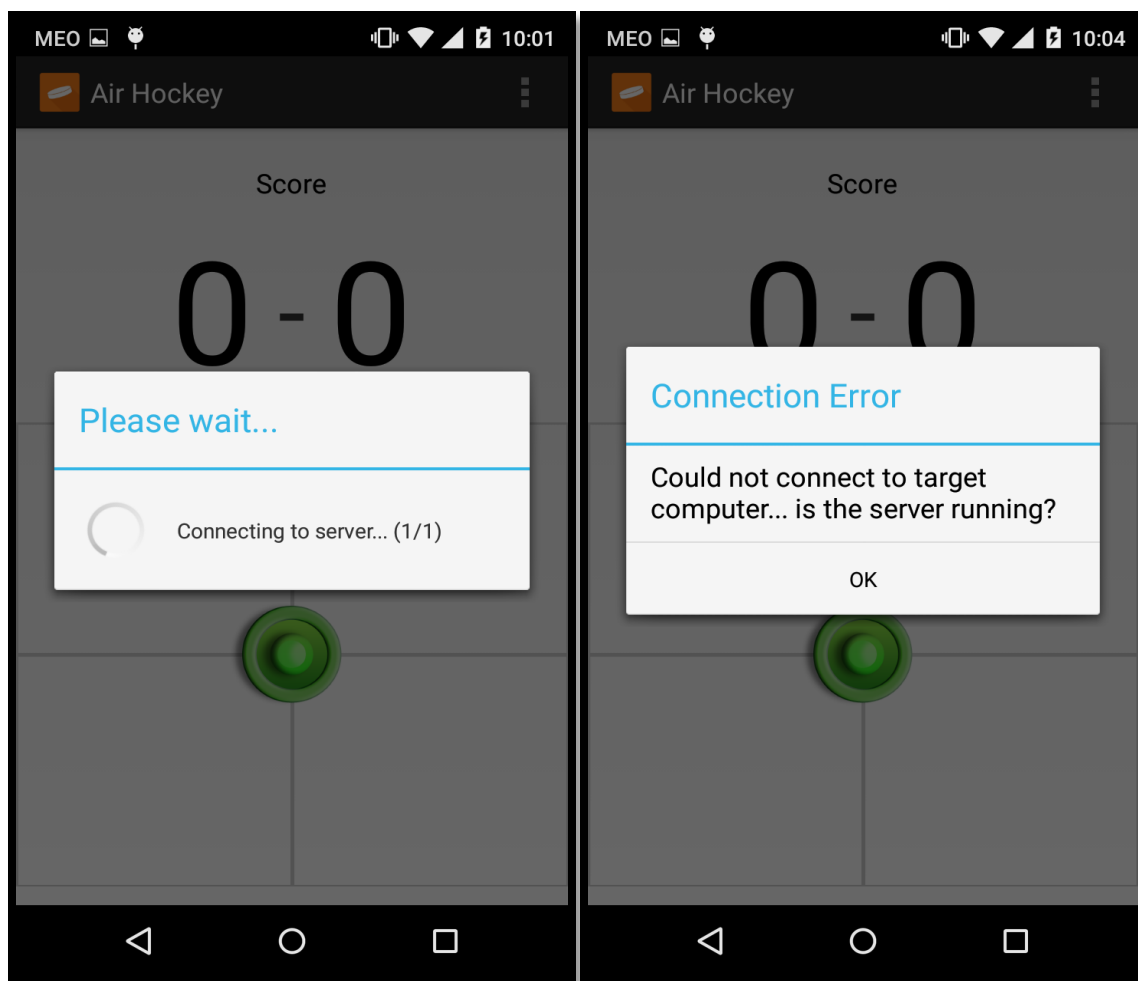


Quando o utilizador inicia a aplicação *Android* do cliente do *Air Hockey* bla bla bla...

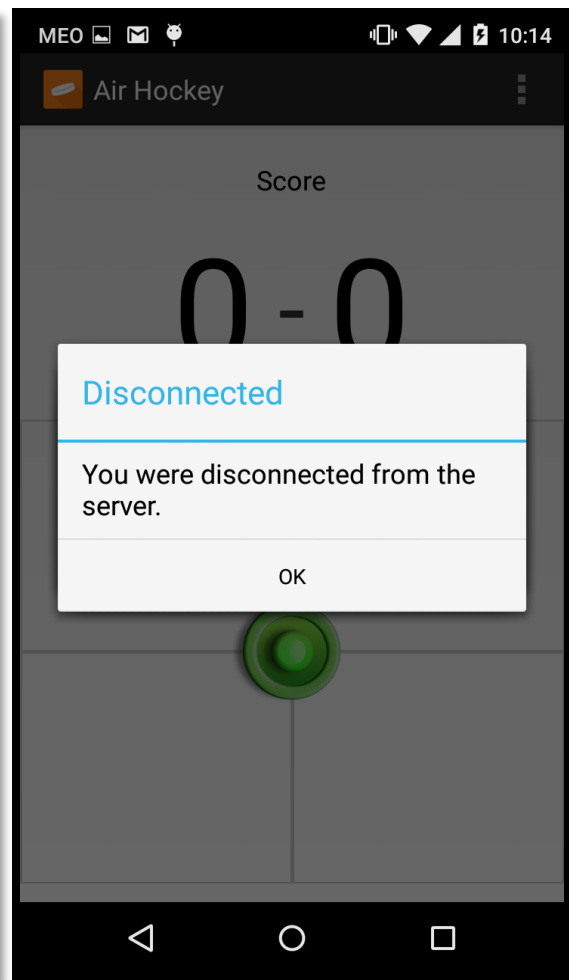
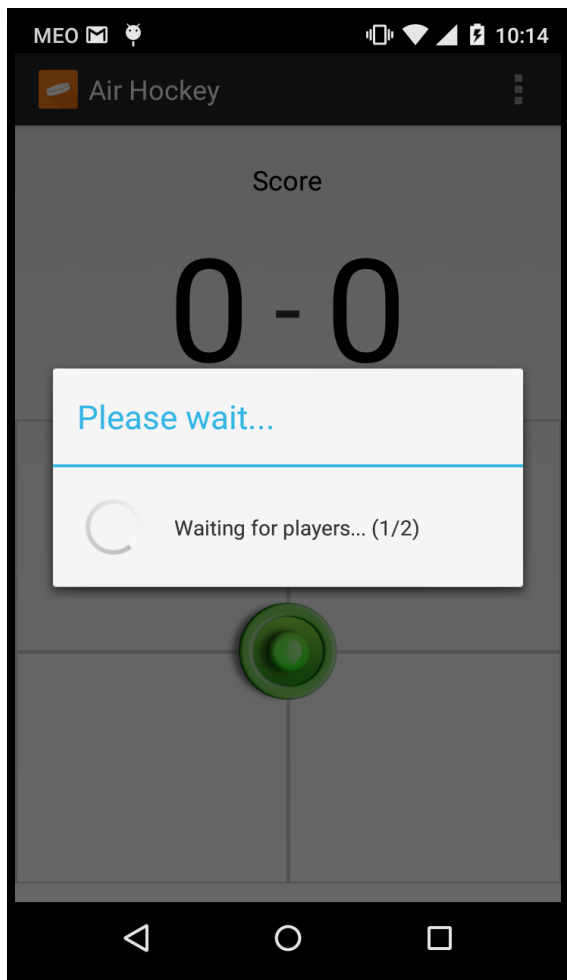
- ✓ Host IP / Endereço IP – endereço local / *internet* onde o servidor se encontra a correr
- ✓ Port / Porta – porta TCP do servidor
- ✓ Username / Nome – nome ou alcunha do utilizador
- ✓ Color / Cor – cor do *paddle* do utilizador

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”.





## 2. Informação do Projeto

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

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

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

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

## 4. Conceção e implementação

### 4.1 Bibliotecas, tecnologias e ferramentas

#### 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](#) to automatically and efficiently transfer object graphs across the network. Esta biblioteca tem a vantagem de correr 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 distribuição 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

Contém as funções que permitem ler e desenhar informação no ecrã contida em ficheiros do tipo *bitmap* (extensão BMP). Para que tal fosse possível, criou-se a uma *struct* denominada **bitmap\_t** que guarda o cabeçalho do ficheiro e a informação de cor nele contida.

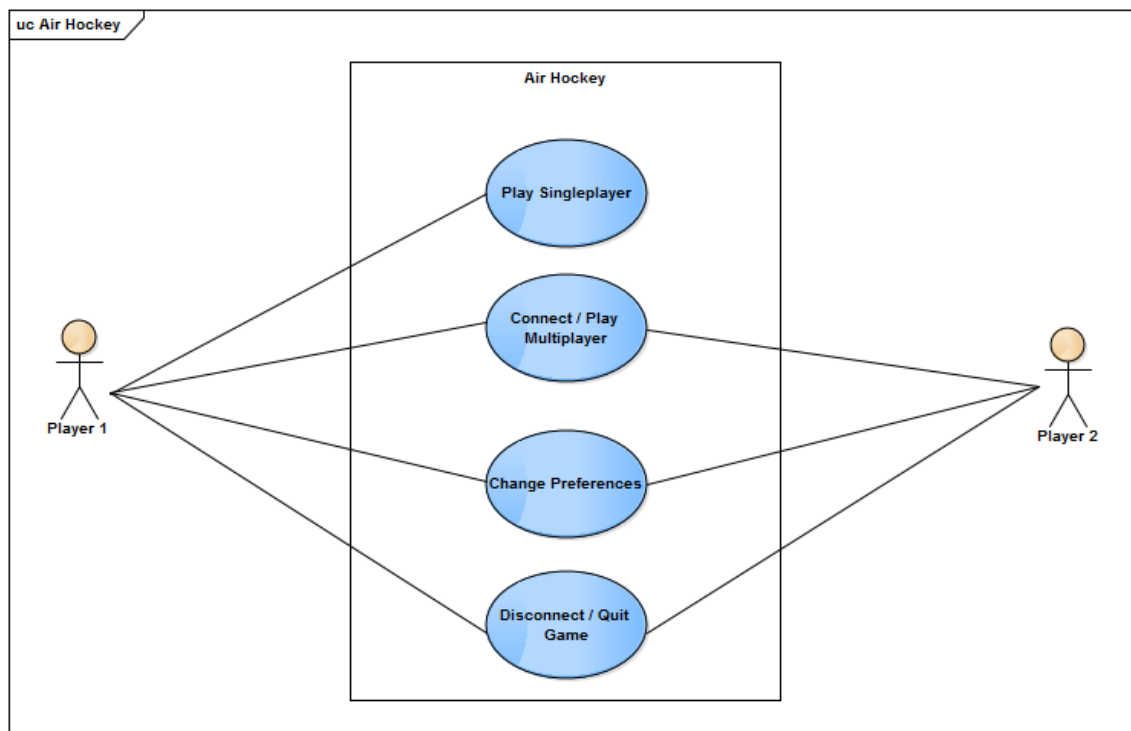
- Peso relativo deste módulo: 5%
- Membro responsável: Pedro Melo

## 4.3 Mecanismos de comunicação

Contém quatro funções que geram todos os níveis do Arkanix (para ambos os modos *singleplayer* e *multiplayer*), preenchendo as áreas de jogo dos diferentes níveis com os diferentes blocos nas suas devidas posições.

- Peso relativo deste módulo: 8%
- Membro responsável: Diogo Marques, Pedro Melo

## 5. Diagrama de casos de utilização



### Jogador 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

### Jogador 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*)

A função *arkanix\_change\_state()* é responsável pela transição entre estados (*menu*, *singleplayer*, *multiplayer*). Esta função, por sua vez, chama as funções *arkanix\_singleplayer()* e *arkanix\_multiplayer()* que inicializam e geram os níveis para o modo *singeplayer* e *multiplayer* respetivamente. É também responsável por ativar as interrupções para os restantes periféricos *on-demand*, isto é, quando forem necessários para o modo de jogo em questão.

## 7. Diagrama de classes

### 7.1 *Package lpoo.proj2*

O modo gráfico VESA utilizado neste projeto foi o 0x117 (1024x768, 16 bits por pixel). As imagens utilizadas encontram-se no formato bitmap e para tal foram criadas funções para leitura e desenho de imagens no mesmo formato. A nossa implementação suporta também imagens *bitmap* com largura ímpar, uma limitação que surgiu logo no início do projeto por acidente, quando nos deparamos com erros gráficos no desenho das imagens. Dada a escassez de informação objetiva na *internet* para o nosso problema, não sabíamos se este era causado por erros no nosso código ou por limitações do formato, o que mais tarde viemos a descobrir.

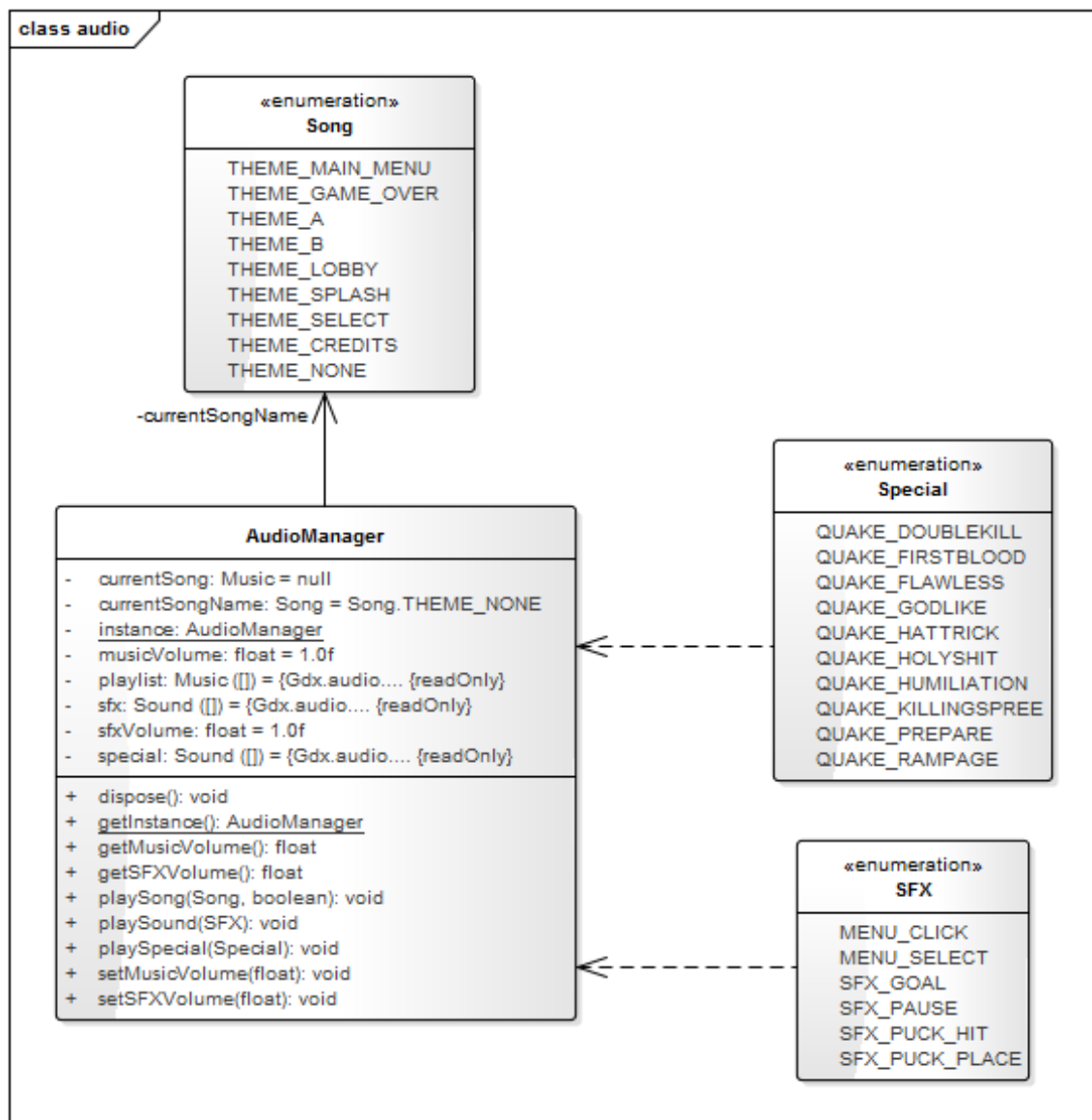
Foi criada uma macro para gerar cores no formato de bits empacotados 5:6:5 (*make\_rgb565*) que tem como argumentos três números inteiros de 8 bits (de 0 a 255) que representam cada componente de cor do sistema *RGB* (*vermelho*, *verde* e *azul*). Além desta, existem outra três macros (*BLUE*, *RED*, *GREEN*) que fazem o processo inverso, isto é, a partir de uma cor representada neste formato (*16 bits*) extraem as suas componentes de 8 bits.

As macros foram essenciais na construção dos blocos e na construção do algoritmo de *alpha blending* para conseguir efeitos de transparência entre dois *buffers* nos ecrãs, que pode ser observado nos ecrãs de final de partida do modo *multiplayer*.

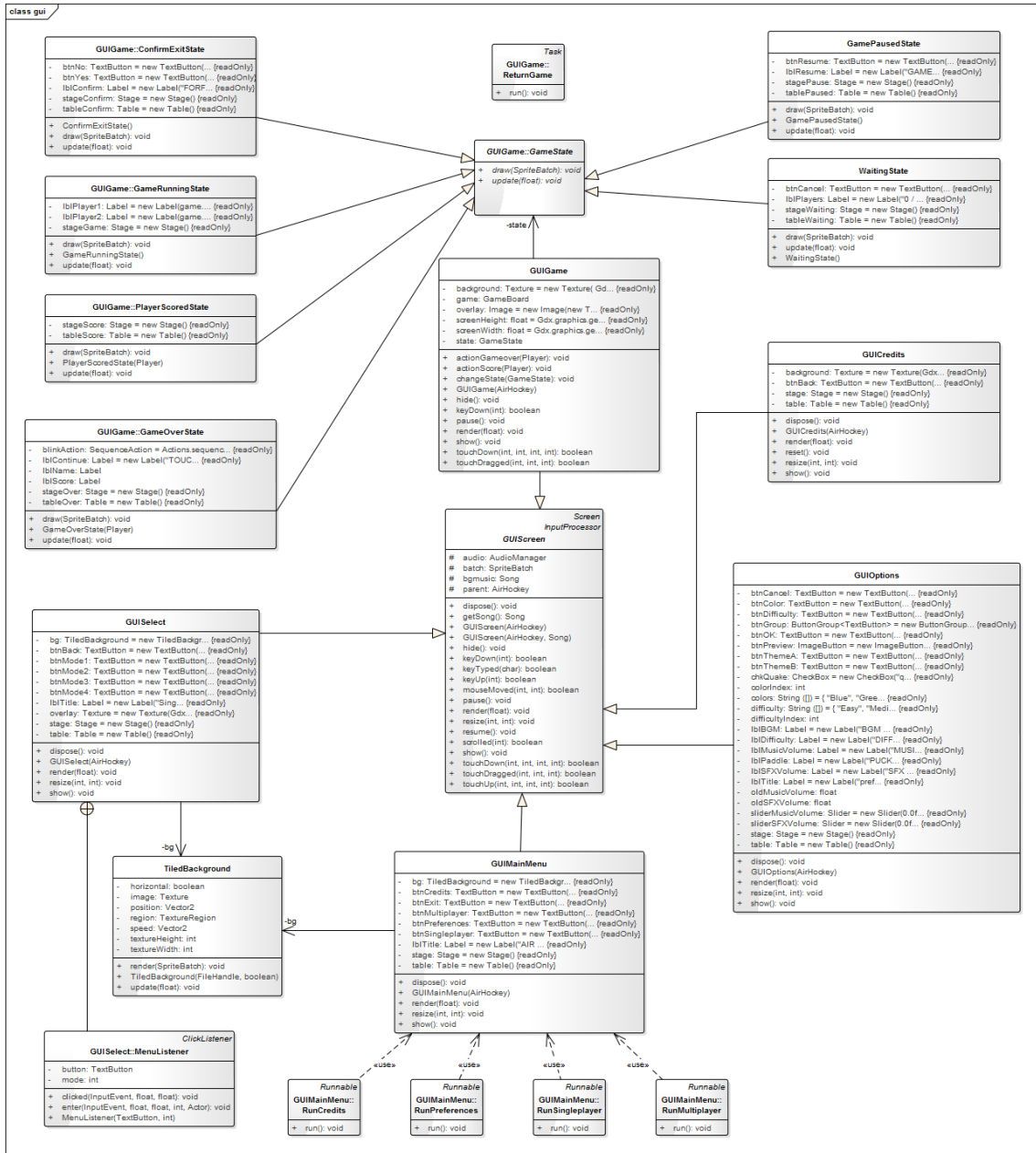
Para desenhar no ecrã recorreu-se a técnicas de *double buffering* e sobreposição de camadas. Tanto o modo *singleplayer* como o módulo *multiplayer* apresentam três buffers:

- *background* - permanente, não pode ser alterado por nenhuma função do jogo, guarda apenas a imagem de fundo do nível, que é convertida do formato *bitmap* para *buffer* aquando a inicialização do nível.
- *blocks\_layer* – atualizado sempre que o jogador destrói um bloco ou perde uma vida; é para este *buffer* que o *background buffer* é copiado; além do fundo, são desenhados nele os blocos existentes na área de jogo do(s) jogador(es), a pontuação e as vidas respetivas.
- *active\_buffer* - é o buffer “ativo”, aquele que é atualizado com maior frequência durante o decorrer do jogo, e onde é desenhada diretamente a camada ativa do jogo (a barra e a bola de cada um dos jogadores). É o único *buffer* que é copiado para a memória gráfica no final de cada interrupção do *timer* (período de refrescamento do ecrã) - pode conter apenas o *blocks\_buffer*, que por sua vez já contém o *background*, se a camada ativa não for atualizada. Numa situação típica, este *buffer* é a sobreposição dos três referidos anteriormente.

## 7.2 Package lpoo.proj2.audio



### 7.3 Package lpoo.proj2.gui





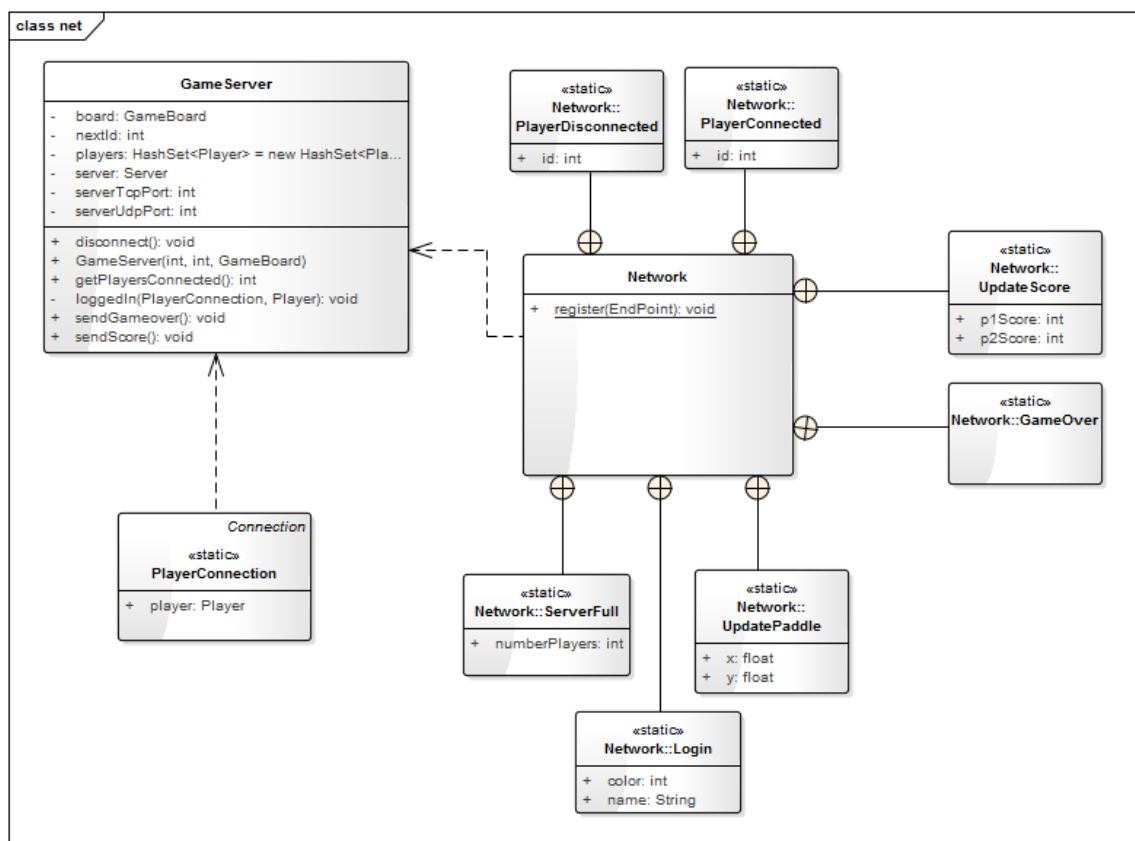
## 7.4 Package lpoo.proj2.logic

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. Assim, os jogadores não precisam de entrar simultaneamente no modo *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 *Receiver 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.