

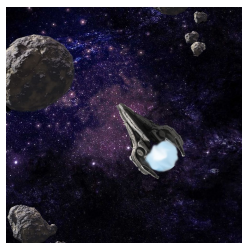


FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Laboratório de Programação Orientada por Objetos

MIEIC

ASTEROIDS



07/06/2016

Turma: 2MIEIC06

Grupo: 7

Autor: Nuno Filipe Sousa e Silva (up201404380@fe.up.pt)

Índice

1. Introdução
2. Instalação
3. Modo de utilização
4. Conceção e implementação
 - 4.1. Estrutura de packages
 - 4.2 Diagramas de classe
5. Padrões de Desenho Utilizados
6. Diagrama de estados
7. Bibliotecas e tecnologias utilizadas
8. Dificuldades encontradas
9. Testes Realizados
10. Conclusão
11. Referências

1. Introdução

Este relatório tem como objetivo esclarecer o funcionamento do segundo projeto que realizamos na unidade curricular Laboratório de Programação Orientada por Objetos bem como apresentar algumas componentes como as bibliotecas, modelo UML, ... de forma a ser bastante fácil e rápido ter uma ideia quase total da arquitetura do trabalho, paradigmas de programação e modo de utilização sem recorrer a uma análise do código.

O projeto baseia-se numa adaptação do “Asteroids”, um jogo arcade bastante popular, lançado pela Atari em 1981, para um modelo que possa ser jogado num dispositivo androide.

O objetivo do jogo é destruir o maior número possível de asteroides sem ser atingido por estes.

A nave perde uma vida quando é atingida por um asteroide e fica invencível por um pequeno período de tempo (o ecrã fica danificado quando a nave é atingida).

A nave tem um número máximo de vezes que pode ser atingido, 3, e perde quando o total de vidas é 0.

Atingir um asteroide irá aumentar o score do jogador e consoante este aumento irá influenciar respetivamente a dificuldade do jogo e os pontos atribuídos por cada asteroide destruído. Ao atingir um asteroide ele, aleatoriamente, pode deixar um pacote de “vida” que irá acrescentar 1 vida à nave, e também pode despedaçar-se em asteroides mais pequenos.

2. Instalação

Instalar normalmente num dispositivo android o ficheiro .apk

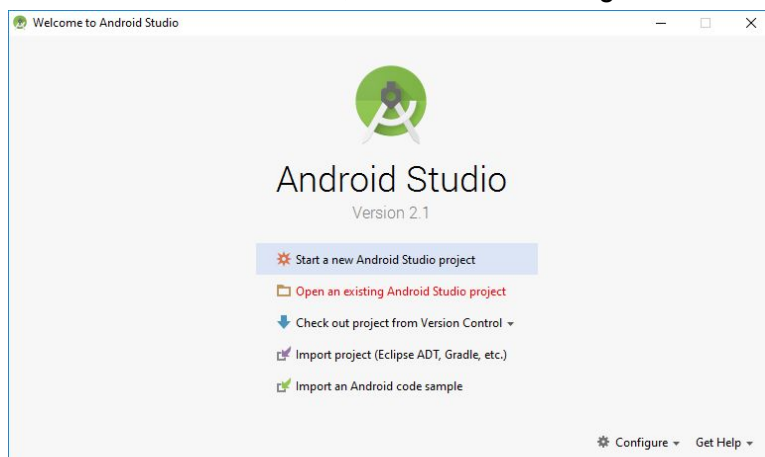
Ficheiro .apk: <https://feupload.fe.up.pt/get/McT5IG7Ohq6e1Y0>

ou

Fazer download do projecto:

Pasta projecto: <https://feupload.fe.up.pt/get/btDugxuLJy12u92>

- Descompactar o ficheiro .zip.
- Abrir o Android Studio, e na “Welcome Page”:



- Carregar em “**Open an existing Android Studio project**” e seleccionar a pasta “**Asteroids**” que foi descompactada.

- Após adicionar o projecto, abrir o mesmo e correr em “**Run ‘app’**” ou **shift + F10**.
- Escolher o dispositivo ou o emulador que desejar e iniciar a aplicação.

3. Modo de utilização

Ecrã Inicial

Ao iniciar a aplicação, irá ser apresentado o Ecrã Inicial do jogo.



Fig. 1 - Ecrã Inicial

No ecrã inicial existem 3 botões:



Fig. 2 - Botão "about"

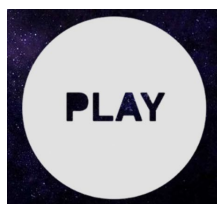


Fig. 3 - Botão jogar

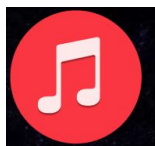


Fig. 4 - Botão do som em estado "On"

Ao carregar no botão “about”, **fig. 2**, ira aparecer uma janela com informação sobre a aplicação.

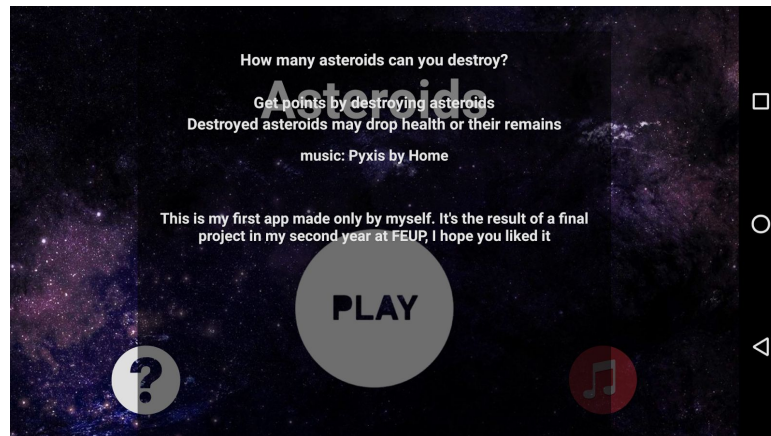


Fig. 5 - Janel “about”

Para fechar a janela e voltar ao ecrã inicial, **fig. 1**, toca-se no ecrã fora da janela ou carrega-se no botão “back”

O botão do som, **fig. 4**, diz o estado do som, se esta ligado ou não. Na imagem da **fig. 4**, o som está ligado, se carregar no botão de som, ele desativa o som e passa para a imagem:

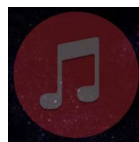


Fig. 6 - Botão som em estado “Off”

que simboliza que o som e a música esta desligado. O estado do som fica guardado mesmo quando se feche a aplicação. Para voltar a ligar o som, é só carregar novamente no botão do som e ele passa da **fig.6** para a **fig.4**.

Ao carregar no botão de jogar, **fig. 3**, o jogo inicia:

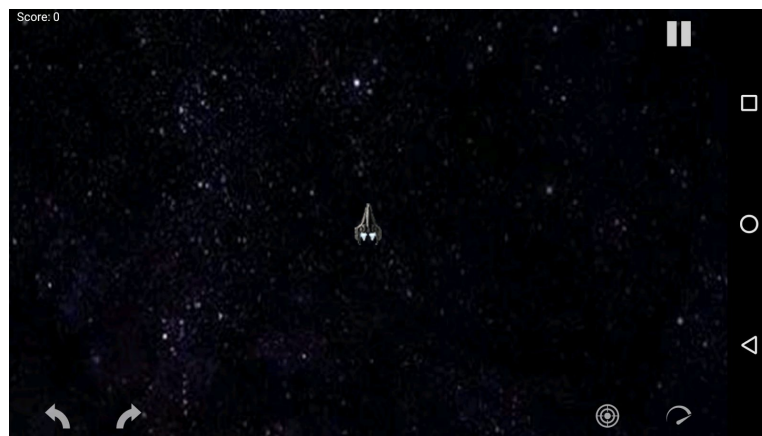


Fig. 6 - Ecrã de jogo

Existem várias ações que o utilizador pode fazer.

Pausar o jogo

Para pausar o jogo, o jogador carrega nestes dois botões:

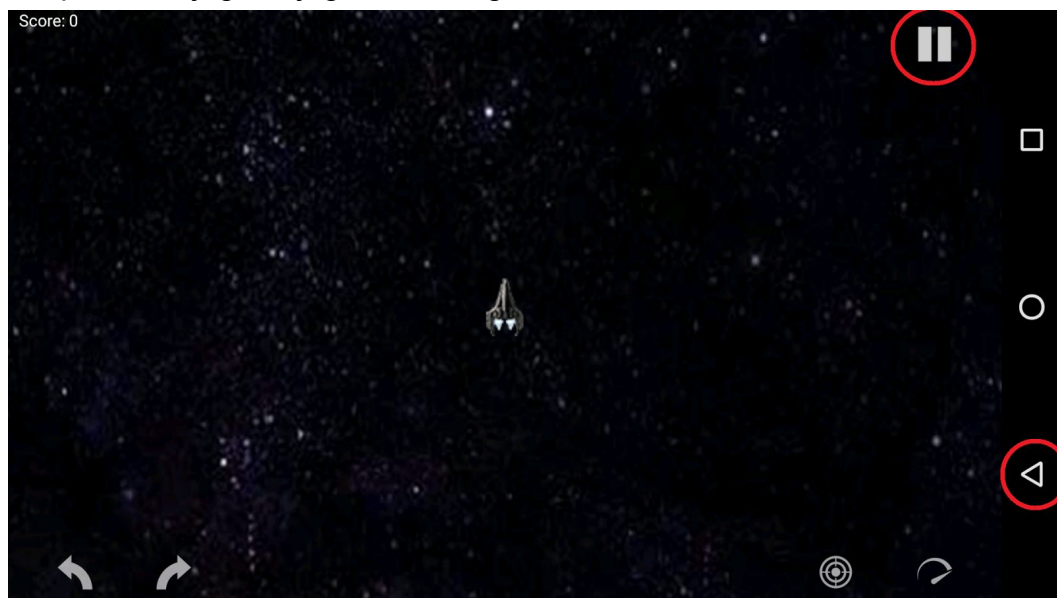


Fig. 7 - Ecrã de jogo a ilustrar os botões que pausam o jogo

Ao pausar o jogo, irá aparecer a janela que indica que o jogo esta em pausa.

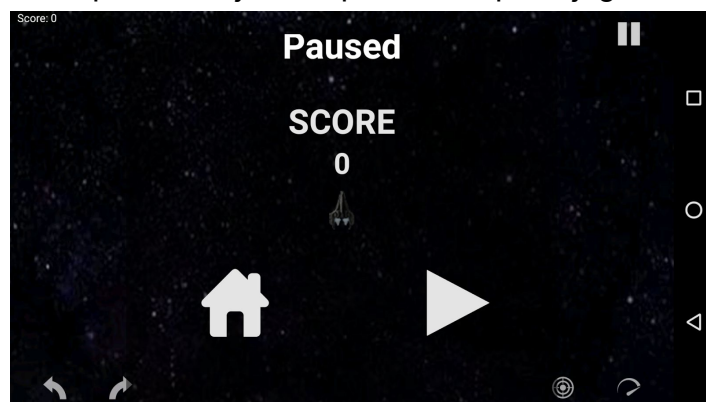


Fig. 8 - Ecrã de jogo a ilustrar os botões que pausam o jogo

A janela mostra o score atual do jogo assim como a opção de retomar o jogo, **fig. 9**, ou sair e voltar ao ecrã inicial, **fig. 10**



Fig. 9 - Botão “home”



Fig. 10 - Botão retomar

Para retomar o jogo carrega-se na **fig.10** ou fora da janela de pausa ou no botão “**back**”.

Jogar o jogo

O jogador controla o jogo com estes 4 botões:

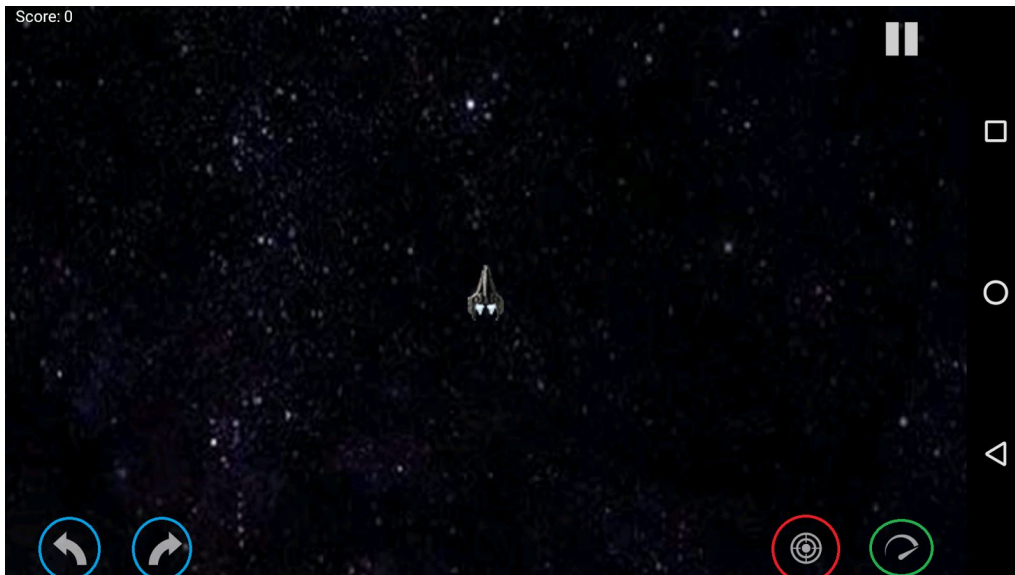


Fig. 11 - Ilustração dos botões que controlam a nave

Botões de direção: Rodam a nave para a esquerda e para a direita.

Botão de disparar: Dispara um projétil que é usado para destruir os asteroides.

Botão de aceleração: Acelera gradualmente a nave até chegar à velocidade máxima.

A nave desloca-se como se estivesse no espaço, mas com algum atrito que vai abrando o movimento.

O jogador tem de ter cuidado com os asteroides, porque eles podem atingir a nave e danificá-la. Quando a nave é atingida, fica a piscar o que simboliza que esta é invencível e não pode sofrer dano. Isto dura uns breves segundos.

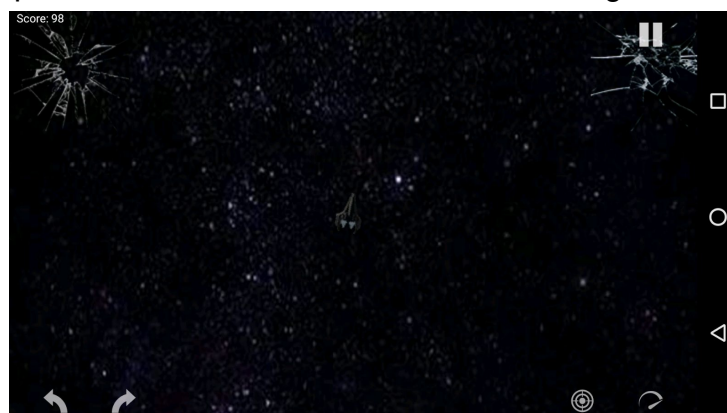


Fig. 12 - Ilustração do jogo após a nave ter sido atingida 2 vezes (note nos cantos superiores o ecrã danificado)

Ao destruir os asteroides a pontuação aumenta. O cálculo da pontuação é baseado na velocidade e tamanho dos asteroides.

Os asteroides podem deixar cair “pacotes de vida” que ficam no mesmo sitio por um tempo limitado ou até o jogador o apanhar.

Esses pacotes acrescentam 1 vida à nave até um máximo de 3 vidas e reparam visualmente o ecrã.

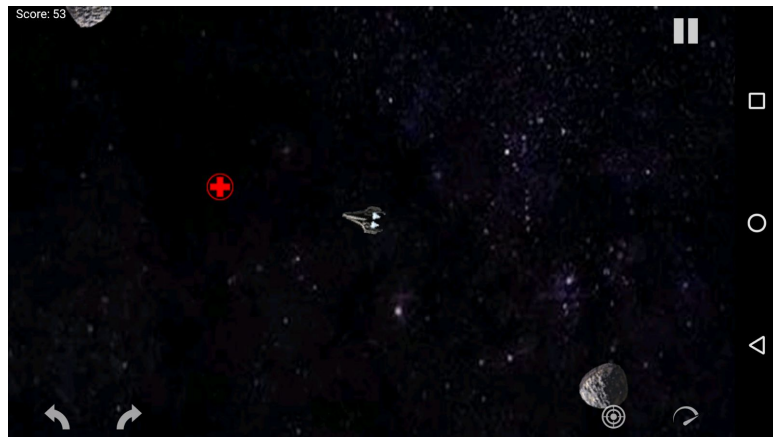


Fig. 13 - Ilustração de um pacote de vida no jogo

Ao serem atingidos, os asteroides também podem despedaçar-se em múltiplos destroços.

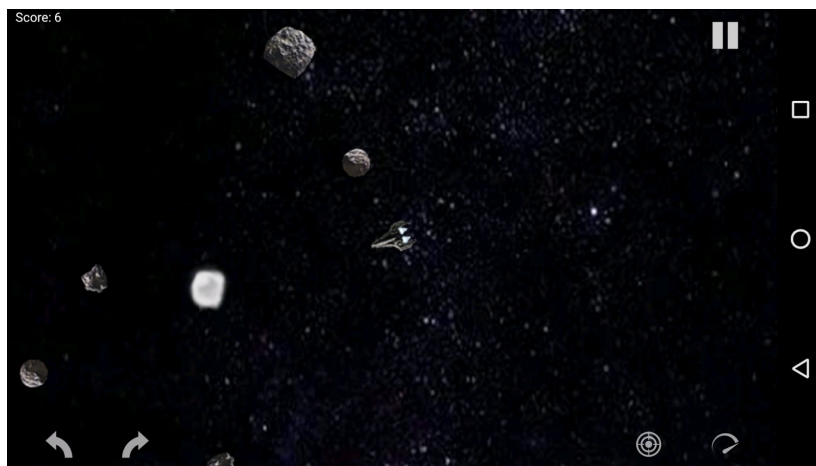


Fig. 14 - Ilustração de um asteroide a dividir-se em múltiplos asteroides

Game Over

Quando o jogador perde, irá aparecer a janela de “GAMEOVER” onde mostra a pontuação máxima registada e a pontuação que fez no jogo.

O utilizador terá dois botões, o botão da **fig. 9**, botão “home” e o botão “reload”.

Para iniciar novo jogo pressiona-se no botão “reload”.

Para voltar ao ecrã inicial, carrega-se no botão “home” ou no botão “back”.

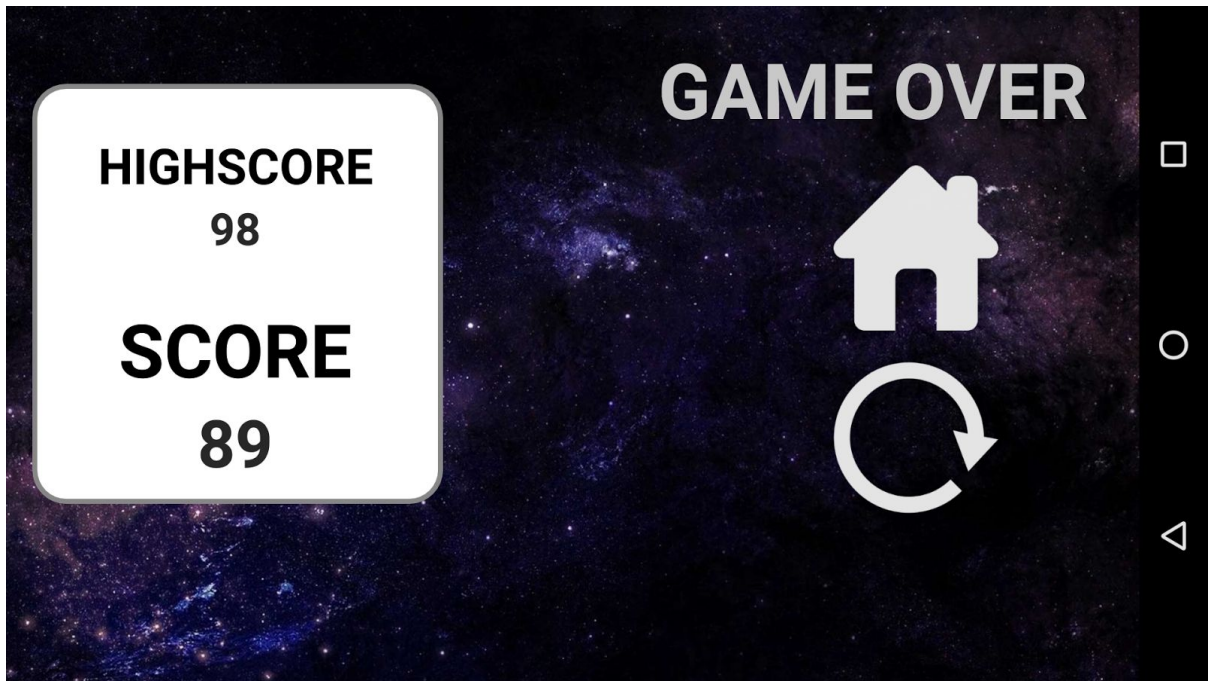
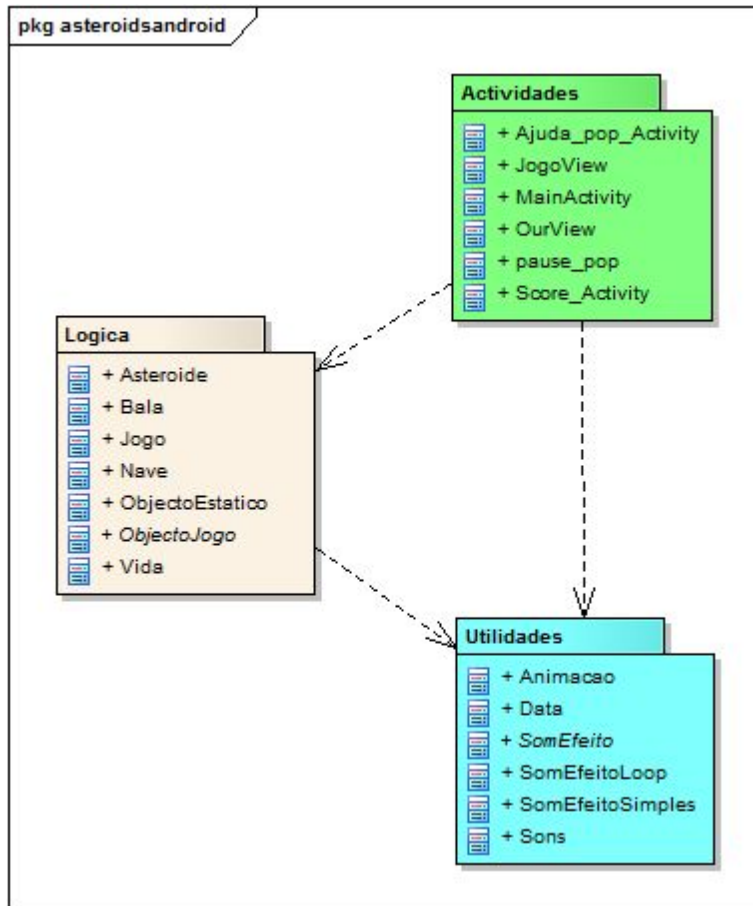


Fig. 15 - Janela “GameOver” em orientação landscape

4. Conceção e implementação

4.1. Estrutura de packages

Contém a estrutura de packages e as suas dependências:



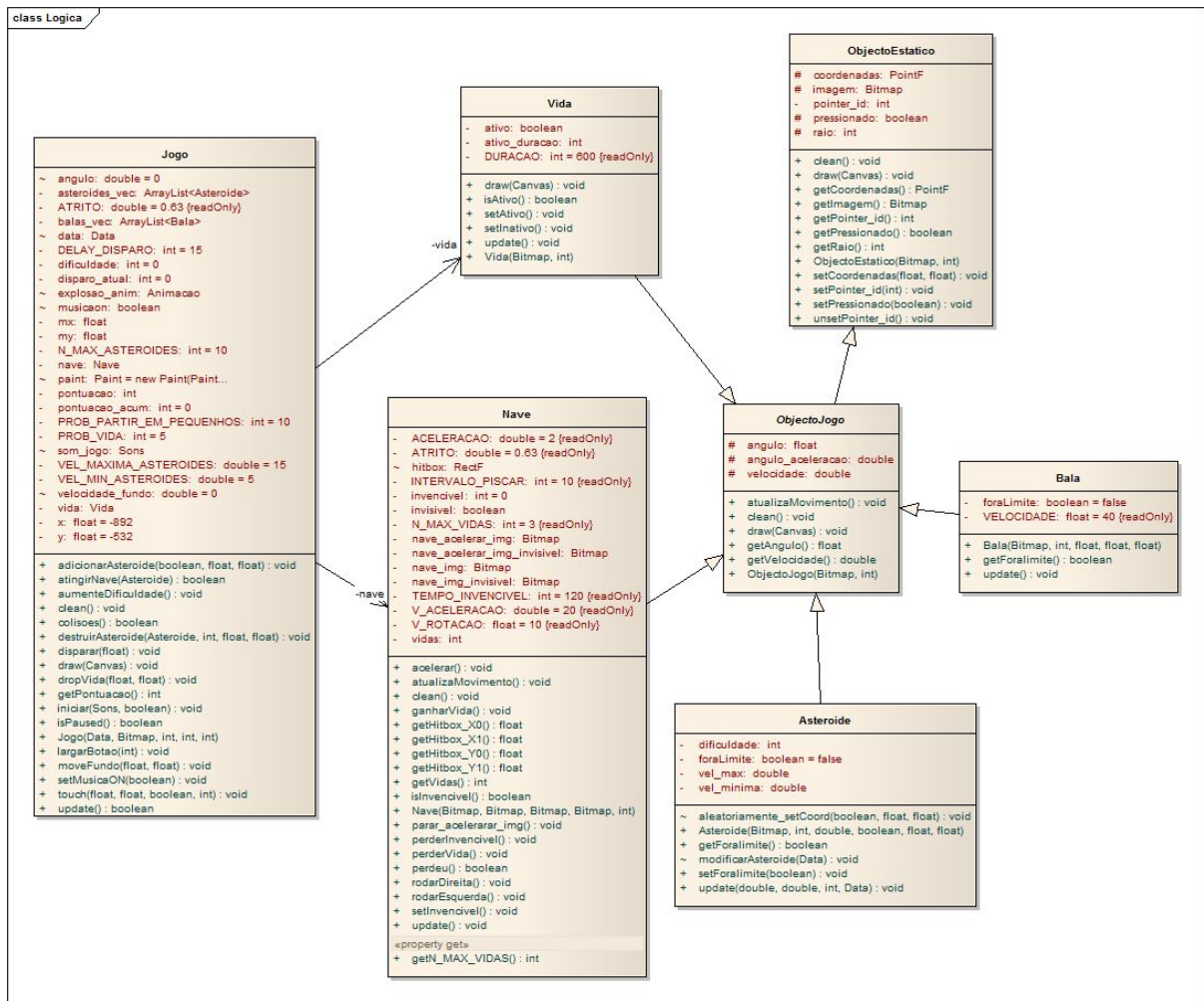
Package	Descrição
Utilidades	Contém os recursos do jogo assim como classes que gerem os sons e animação.
Lógica	Contém toda a lógica do jogo assim como os objectos envolvidos no mesmo.

Actividades

Contém as diversas “activities” da aplicação

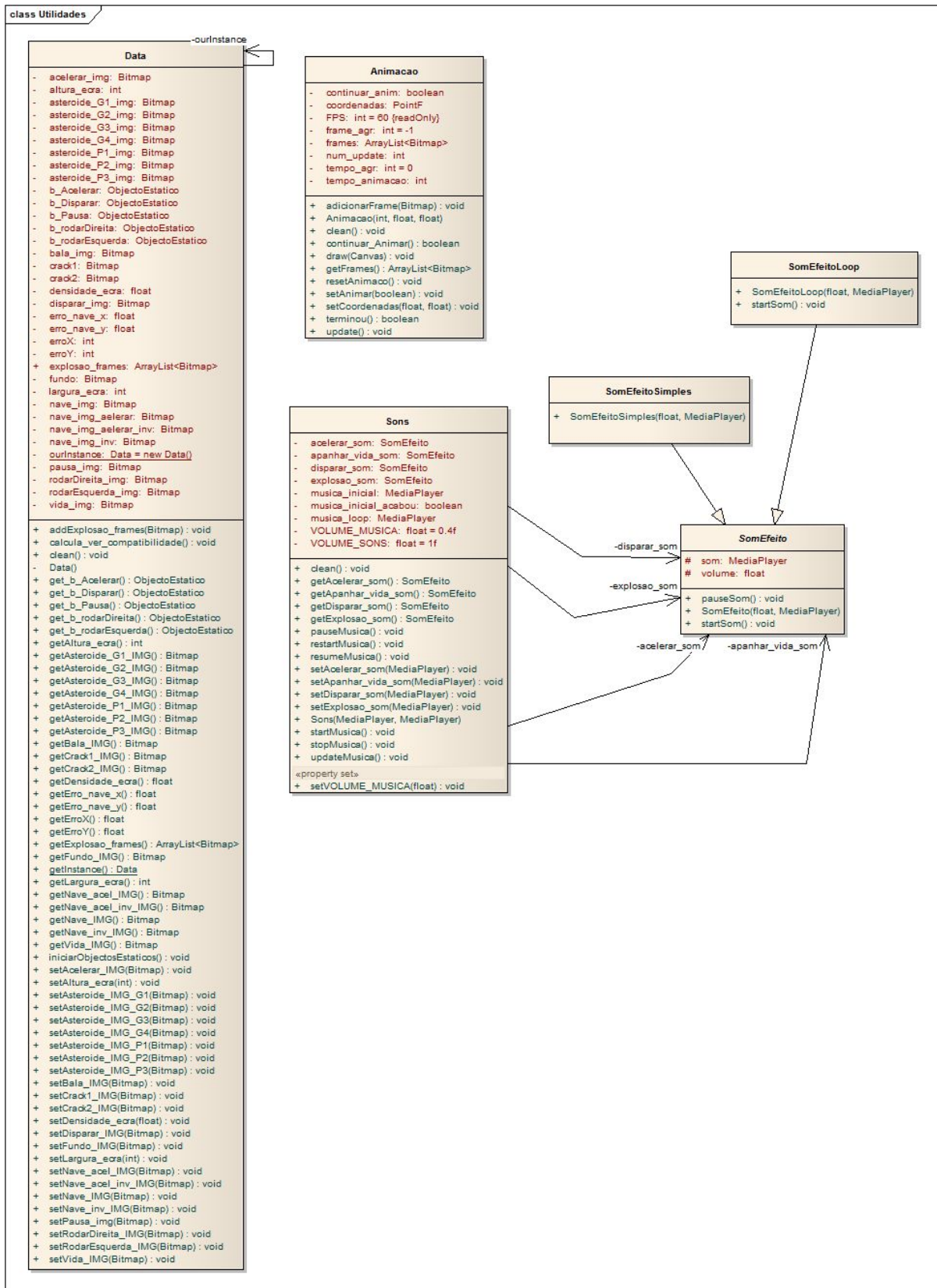
4.2 Diagramas de classe

Package Logica:



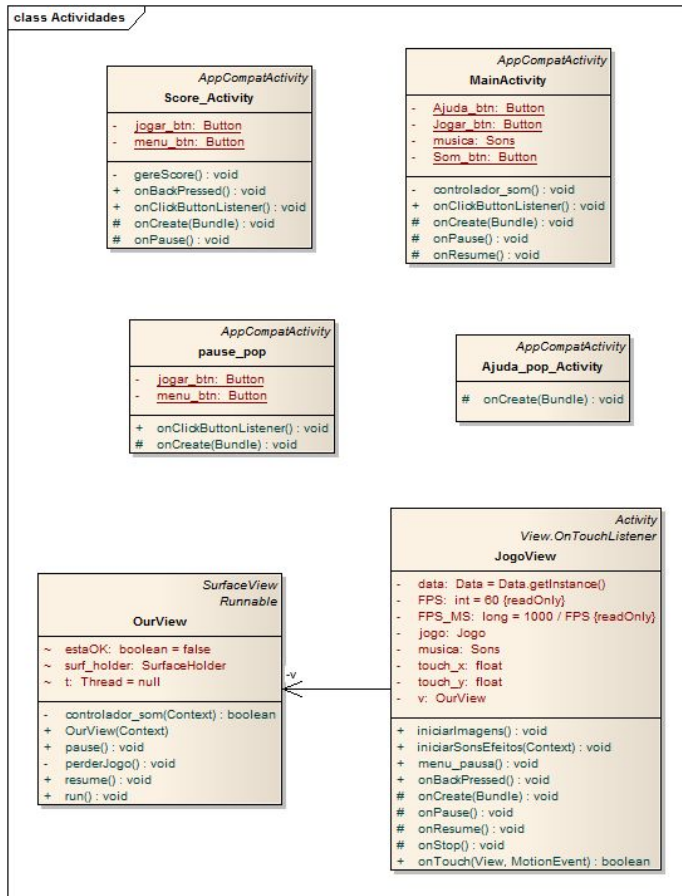
Classe	Descrição
Jogo	Gere todas as ações do jogo
Nave	Objecto controlado pelo utilizador
ObjectoJogo	Representa o que é em comum dos objectos na qual há movimento e interação
ObjectoEstatico	Representa objectos que ficam parados no jogo
Bala	Representa o projectil que a nave dispara
Vida	Sistema usado para regenerar a nave
Asteroide	Objecto que danifica a nave e que o jogador tem de destruir

Package utilidades:



Classe	Descrição
Data	Classe singleton usada para encapsular informação e variáveis
Som	Gere todos os sons do jogo
SomEfeito	Classe abstract que contém o que é comum aos sons de efeitos especiais
SomEfeitoSimples	Usado para sons simples
SomEfeitoLoop	Usado para sons que ficam em loop
Animacao	Anima um conjunto de frames

Package Actividades:



Classe	Descrição
JogoView	Contém a view personalizada
OurView	View personalizada onde corre o jogo
pause_pop	Janela de pausa
Ajuda_pop_Activity	Janela do botão “about”
Score_Activity	Janela que se mostra quando se perde
MainActivity	Janela principal, ecrã de inicio

5. Padrões de Desenho Utilizados

Criação

Object Pool Design - reutilizar os asteroides que passam o limite do jogo, fazer com que os mesmos apareçam noutra posição qualquer com outra direção e velocidade. O mesmo para as balas e vida.

Abstract Factory - utilizar tendo em conta as similaridades entre os vários objetos do jogo.

Estuturais

Private Class Data - para reduzir o número de atributos na classe "Jogo". A classe Data usa este design pattern.

Adapter Design - por causa das diferenças entre os asteroides, nave e balas, por exemplo.

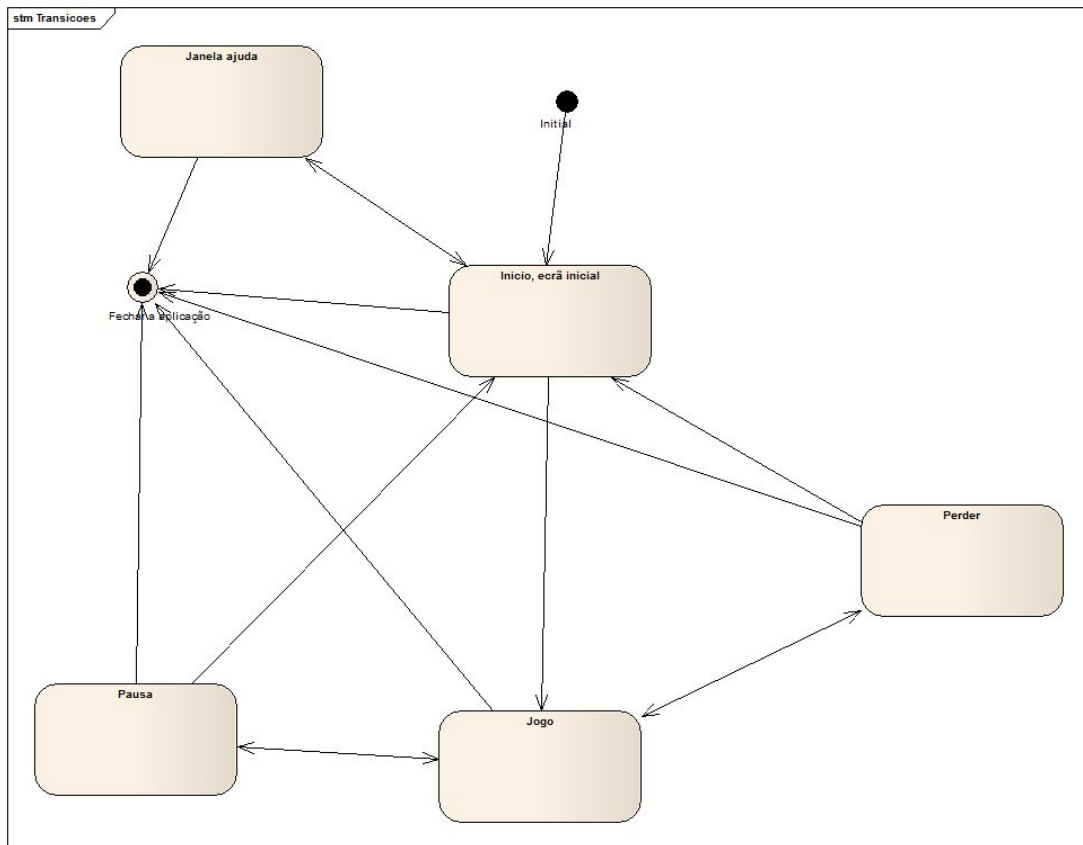
Comportamentais

Iterator - métodos "update" nas várias classes.

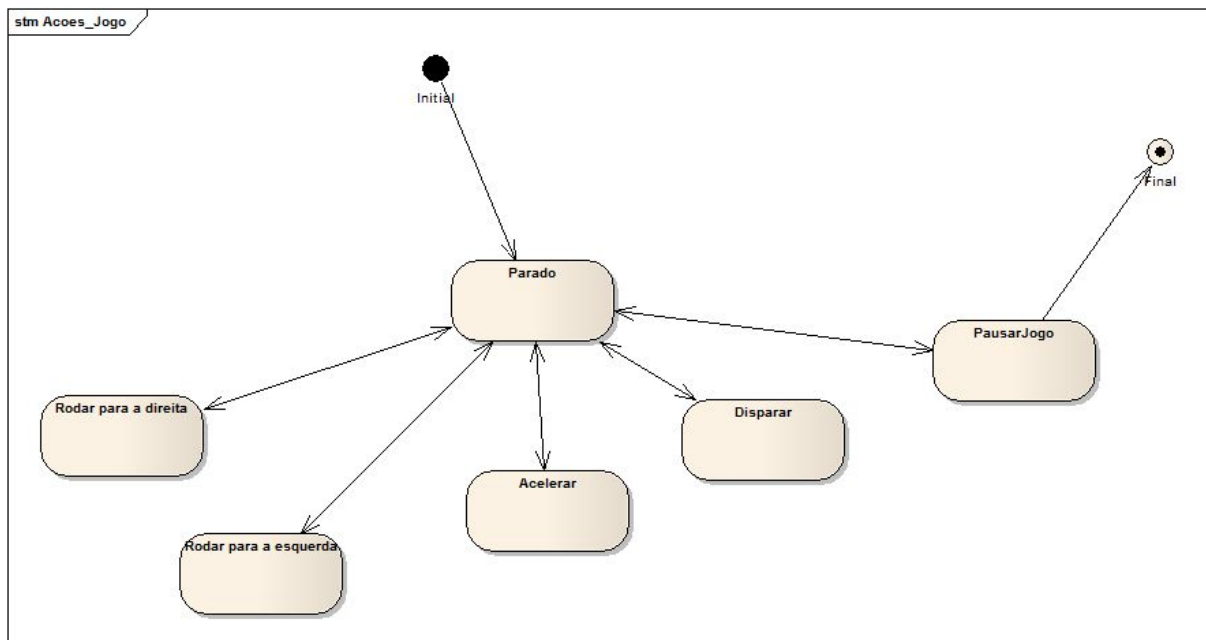
State - muda ações dependendo de estados de alguns objetos, como colocar o pacote de vida como inativo quando é apanhado ou o tempo se esgota, ou quando a nave é atingida e fica invencível.

6. Diagrama de estados

Transições entre atividades da app



Transições entre ações que o utilizador pode ter com o jogo.



7. Bibliotecas e tecnologias utilizadas

Foram apenas usadas as bibliotecas standard do android e o Android Studio.

8. Dificuldades encontradas

A aplicação não ficava bem em outros dispositivos e inicialmente tinha memory leak, por causa dos bitmaps..

9. Testes Realizados

Foram feitos os testes previsto na entrega intermédia:

- Movimentos da nave.
- Nave não passar os limites estabelecidos.
- Movimento dos asteroides.
- Colisão entre os objetos.
- Perder vida quando a nave é atingida.
- Perder o jogo quando as vidas da nave forem 0.
- Verificar se a nave dispara a bala.
- Verificar se a bala disparada pela nave vai na direção correta.

Também foram feitos vários testes manuais como:

- Verificar se a música para quando se minimiza a app.
- Verificar se o estado do som e os próprios sons ficam ativos, ou não, quando se fecha a app e se reinicia a mesma.
- Confirmar que desenha corretamente no ecrã.
- Confirmar a compatibilidade em vários dispositivos através do emulador, telemóveis e tablets de diferentes tamanhos e densidades de ecrã.
- Confirmação de que a pontuação máxima fica guardada entre sessões de uso da app.
- Confirmação visuais das colisões, para além dos testes unitários, entre os diferentes objetos de jogo.
- Teste dos botões em diferentes dispositivos através do emulador, tablets e telemóveis de diferentes tamanhos e densidades de ecrã.
- Confirmação de que os recursos são libertados quando não são necessários, melhorando assim o desempenho da aplicação e prevenindo o encerramento por falta de memória da aplicação.
- Confirmação de que o ecrã também é reparado quando a nave é raparada.
- Confirmação visual de todas as animações do jogo.
- Confirmação da correta transição da música para o loop da música.
- Confirmação do correto uso do som.
- Confirmação de todas as transições possíveis da aplicação, ou seja, foi testado todos os botões e tipos de toques possíveis, incluindo botões do telemóvel como o botão “back”.

10. Conclusão

Inicialmente estava inserido num grupo de 2, mas o meu colega desistiu de fazer o projecto, o que fiquei eu sozinho no desenvolvimento do mesmo. Outra dificuldade que tive foi ter de usar emuladores e não poder testar em dispositivos físicos, sendo que para testar o multi-touch, tive de pedir a colegas que experimentassem o jogo e que me dessem feedback de problemas que possam ter ocorrido.

Assim sendo, cumpri todos os objetivos estabelecidos, e foram acrescentadas outras funcionalidades como regeneração, música, som, criação de animações e outros detalhes que tornaram a aplicação mais apelativa.

A maior dificuldade foi acabar tudo a tempo com uma aplicação o mais bem feita possível, visto que foi tudo feito sozinho. Desde a edição e até criação dos sons, até à criação e edição das imagens e animações. Também foi feita e testada, através de emuladores, a compatibilidade para vários dispositivos android.

No final do desenvolvimento, cheguei à conclusão que podia ter usado métodos diferentes e provavelmente melhores na resolução dos problemas encontrados, como por exemplo usar “Soundpool” em vez de “MediaPlayer” nos sons e também, pelos comentários dos meus colegas, devia ter usado libgdx o que me iria simplificar o desenvolvimento do jogo.

No final, apesar de o jogo não ter muitas componentes úteis em rede, também queria implementar algo que utilizasse redes, como utilizar o dispositivo apenas como controlador e jogar no computador ou televisor.

11. Referências

Android Developers - <https://developer.android.com/> (06/06/2016)