

08/01/2018

Beatriz Abreu e Francisco Santos

Computação Móvel

Conteúdo

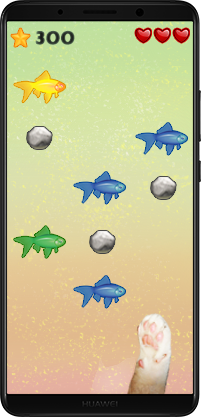
[Introdução 2](#_Toc534755408)

[Implementação do projeto 3](#_Toc534755409)

[Conclusão 5](#_Toc534755410)

# Introdução

De forma a elaborar o projeto final da UC Computação Móvel, projetou-se um protótipo de aplicação – um jogo *singleplayer* denominado *Kitty Scratchy Fishy* do tipo *hyper casual*. Para ver um clipe do jogo, clique [aqui](https://media.giphy.com/media/4PZHxWBDVM5TBTFUoN/giphy.gif).

Este jogo conta a história de um gatinho esfomeado que, quando foi passear, se deparou com um rio cheio de peixes. O objetivo deste felino é saciar a sua fome e, para o cumprir, terá que apanhar o maior número de peixes que por aquele rio passem. Todavia, para a infelicidade deste pequeno animal, o rio contém também pedras; portanto, deverá desviar-se destas enquanto cumpre a sua missão.

Assim que o gato perder todas as suas três vidas (perde uma vida por cada pedra que apanha), o jogo termina e ao utilizador será apresentado um ecrã com a sua pontuação e, para comparação, o top de pontuações mais altas no jogo. Neste ecrã, estará exposto um botão de *share,* para que o jogador possa partilhar a sua pontuação por diferentes plataformas, como o *Facebook*.

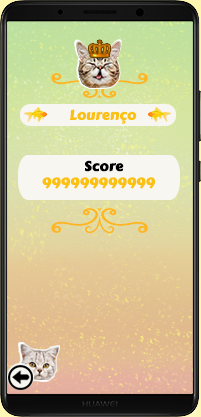
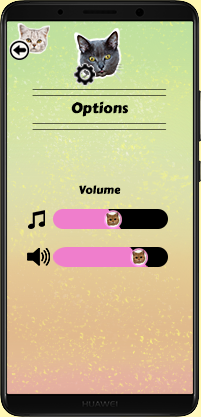
*Kitty Scratchy Fishy* não tem tempo limite, terminando apenas quando o número de vidas é nulo. No entanto, o nível de dificuldade vai aumentando à medida que o jogador progride 🡪 Quanto maior sua pontuação, mais rápido o jogo se tornará.

Figura 5 - Ecrã Options

Figura 4 - Ecrã de detalhe LeaderBoard

Figura 3 - Ecrã Leaderboard

Figura 2 - Menu Principal

Figura 1 - Ecrã de jogo

A aplicação terá um ecrã principal com um menu constituído por vários botões – “*Play*”, “*Options*” e “*Leaderboards*”. Cada um dos botões referidos anteriormente cria um novo ecrã (ou seja, cada um é referente a uma *activity*), sendo que é no ecrã “*Play*” que está disposto o jogo. Os restantes ecrãs servem, respetivamente, para ajustar as opções de som do jogo e para verificar quais as melhores pontuações de sempre no jogo (de todos os seus jogadores, carregadas de uma base de dados). Enquanto joga, o jogador deparar-se-á inúmeras vezes com botões “*back”*, para que possa, a qualquer momento, voltar atrás com a sua escolha.

O público-alvo desta aplicação abrange todos os indivíduos que possuam um telemóvel e tenham idades compreendidas entre os 6 e os 25 anos de idade, uma vez que o jogo poderá camuflar-se tanto como um jogo infantil ou como um jogo de brincadeira para jovens e adultos.

# Implementação do projeto

De forma a implementar as funcionalidades acima descritas numa aplicação móvel, decidimos fazer um projeto de *Android Studio* sem utilização de *libGDX*, para que seja mais simples a tardia implementação de *Firebase* no projeto.

Começou-se pela elaboração de um plano de trabalho, em que projetamos toda a ideia de aplicação (todos os ecrãs com todas as suas ações e botões discriminados). De seguida, optou-se por seguir as instruções do docente, que nos instruiu sobre como fazer um jogo básico sem utilizar a biblioteca externa acima referida e assim iniciamos a implementação do projeto em *Android Studio*.

Primeiramente construímos o *design* do menu principal, com o seu painel de fundo personalizado com uma imagem e três botões (para aceder aos ecrãs de jogo, opções e *leaderboards*); Esta ação seguiu-se pela adição de novas *activities*, demonstradas quando pressionados determinados botões do menu principal, como:

* *GameActivity* (é nesta atividade que a magia acontece – é chamado o método *“setcontentview”* com a *SurfaceView* (denominada *GameView*) criada como argumento e o jogo é criado recorrendo a todos os recursos gráficos e não gráficos nele inicializados);
* *LeaderBoardActivity* (esta atividade recorre à *real time database* do *Firebase* para demonstrar os nomes dos utilizadores (guardados através de uma autenticação incógnita) e as suas pontuações (que aparecem quando pressionado o ecrã));

Em adição às atividades acima referidas, a *MainActivity* gere o menu principal, tratando informação relacionada com sons (música do menu principal e booleano referente ao estado do som, que poderá ser ativado ou desativado) e botões que encaminham o jogador para o ecrã de jogo ou para o menu de *leaderboards*.

Ecrãs da aplicação em detalhe:

* Menu principal:
  + Este ecrã é controlado pela *MainActivity* (ou *MainActivityNoSound*, caso o som tenha sido desativado do jogo) e permite ao jogador entrar no ecrã de jogo ou no ecrã de *leaderboards,* ou ainda desativar o som do jogo.
* Ecrã de jogo:
  + Este ecrã é acedido quando pressionado o botão *“Play”* na *MainActivity* e é controlado pela *GameActivity*, permitindo ao jogador explorar o mundo de jogo criado em *GameView* com recurso a *SurfaceView,* coletando peixes e desviando-se de pedras. Neste ecrã estão presentes vários componentes, como retângulos (que determinam colisões), uma lista de *Sprites* que guarda todos os elementos gráficos de interação com o jogador (a pata do jogador, peixes, pedras e avisos), sons de jogo ou ainda elementos cruciais ao jogo, como contagem de vidas e pontuação.
* Ecrã de *leaderboard*:
  + Acedido no final de cada jogo ou quando pressionado o botão *“Leaderboard”* na atividade principal, este menu (constituído por um painel de fundo, uma série de caixas de texto e um botão de retroceder) permite ao jogador ver o top 10 de rankings no jogo. Sempre que acedido, este painel abre primeiramente uma mensagem para que o utilizador lá introduza o seu nome, que poderá ou não constar na tabela de líderes, conforme o resultado obtido no jogo.



Figura 6 – Ecrã de jogo

Figura 6 – Ecrã de leaderboards

Figura 5 - Menu principal

Numa fase mais tardia do desenvolvimento da aplicação, decidimos adicionar uma série de sons conforme as ações do jogador:

* Quando coleta um peixe dourado, ouve um som subtil de um gato a miar;
* Para além de desenhar um aviso na tela quando é atingido por uma pedra, o jogador poderá ouvir também um miar bastante revoltado.

Em adição aos sons acima referidos, foram também selecionadas três melodias para o jogo – uma para o menu principal, uma para o ecrã de jogo e uma outra para o ecrã de *leaderboards*.

Todas as alterações anteriormente referidas resultaram num jogo, denominado *Kitty Scratchy Fishy,* cujas mecânicas são:

* Coletar peixes:
  + A pontuação obtida por esta ação varia conforme a raridade do peixe (os peixes mais comuns são os peixes azuis, seguidos pelos verdes e finalmente pelos dourados).
* Desviar de pedras.

# Conclusão

A elaboração deste projeto tornou-se bastante complicada devido a várias dificuldades sentidas. Todavia, penso que o nosso grupo teve um balanço bastante positivo, terminando um projeto finalizado e completamente jogável e interativo.

Uma das primeiras dificuldades sentidas na elaboração do projeto ocorreu na implementação de *Firebase* no projeto, onde necessário despender bastante tempo devido a erros que este causava num dos nossos dispositivos relacionado com incompatibilidade de *SDK*; todavia, este erro foi tardiamente solucionado. Apesar de termos eventualmente resolvido este primeiro problema relativo ao *Firebase* com que nos enfrentamos, foram surgindo mais dificuldades relativas à plataforma que acabamos por resolver com a ajuda do docente.

A barra de regulação de som que tínhamos inicialmente no projeto passou a ser um botão *on/off*, visto que a barra requeria a criação de uma nova *activity* para todo o caso desnecessária, visto que o utilizador pode já regular o som da aplicação através da barra de som do seu telemóvel. Adicionalmente, foram também removidos alguns ecrãs anteriormente presentes no jogo (por exemplo: os ecrãs das *figuras 4 e 5*), visto que não eram necessários na nova implementação do projeto, e foram ainda melhorados os *assets* usados inicialmente, tornando o jogo mais harmonioso e coeso.

Para efeitos de *game design*, foram ainda adicionados mais objetos de jogo, que têm agora velocidades aleatórias, aumentando o nível de dificuldade do jogo.

Com este projeto, pudemos melhorar as nossas capacidades de desenvolver programas com a linguagem de *Java* e com o *framework* *Android Studio*. Desta forma, concluímos que com as bases que neste projeto implementamos conseguimos agora desenvolver qualquer tipo de aplicativo *mobile* na *framework*.