

# VitalPaint

Course: Introdução à Computação Móvel

Date: Aveiro, 29/11/2017

Authors: 80210: Bruno José Pires Silva  
80317: Fábio Miguel Tomaz dos Santos

Project abstract: VitalPaint - uma aplicação que permite jogar paintball com os amigos para seguir a posição dos companheiros de equipa e detetar inimigos, bem como criar perfis de jogo.

Table of contents:

[1 Introduction](#)

[2 Application concept](#)

[3 User experience design process](#)

[4 Architectural plan for the solution](#)

[5 Implemented solution in Android](#)

[6 Conclusion](#)

[7 References and resources](#)

# 1 Introduction

A aplicação deverá ser usada num ambiente de jogo de paintball entre equipas, e tem como objetivo dar a cada jogador uma maior perceção da área circundante, através de uma vista por satélite do local de jogo e das posições dos colegas de equipa, e de inimigos (quando estiverem perto) representados com marcadores no mapa. Cada jogador terá também um perfil com informação das suas derrotas e vitórias e foto de perfil, capa e descrição que podem ser alterados.

## 2 Application concept

### **Descrição da aplicação (funcionalidades e workflows)**

A aplicação destina-se a todos aqueles que pretendam jogar paintball com 1 ou mais pessoas, conhecidas ou desconhecidas, desde que estejam devidamente registadas na aplicação e numa zona próxima entre si. Isto deve ser feito num espaço outdoor, de preferência com sítios para os jogadores não sejam facilmente visíveis pelos inimigos.

Um jogo é iniciado através de um lobby, que pode ser definido por qualquer jogador, devendo estar no local onde o jogo ocorrerá (pois as coordenadas do utilizador que criou o lobby serão registadas neste), e definir a que horas o jogo começa. O lobby ficará criado sendo possível que qualquer pessoa num raio de 10 km aceda ao lobby. Após entrar no lobby os jogadores deverão escolher em que equipa ficam. Quando chegar a hora de início do jogo definida, todos os jogadores no lobby que estão colocados numa equipa terão o jogo iniciado num mapa com visão de satélite. Os jogadores podem aproveitar o tempo que falta até ao início da partida para combinar estratégias (entre equipa) para a partida, bem como afastarem-se / esconderem-se da equipa inimiga.

Quando um jogador é atingido, este deverá indicar tal ato pressionando o botão “fui atingido!”, ficando assim fora do jogo e podendo ver os restantes elementos da equipa a jogar no mapa. Espera-se que todos os jogadores tenham um bom espírito de jogo e que não façam batota ao não pressionar o botão “fui atingido!” na devida situação. De qualquer maneira, num jogo de paintball, quando um jogador é atingido, este ficará “marcado” com uma bola de tinta, por isso não será difícil identificar um batoteiro.

O jogo termina quando todos os elementos da equipa tiverem sido atingidos.

É possível criar um perfil com foto de perfil, capa, nome e uma curta biografia, bem como consultar o histórico de derrotas/vitórias, o número de vitórias/derrotas, e adicionar jogadores como amigos, e consultar o perfil de amigos.

### **Os seguintes conteúdos não foram incluídos no projeto:**

- Inicialmente, estava previsto que o utilizador que criou o lobby dava início ao jogo pressionando um botão, sendo o jogo iniciado também para os restantes elementos do lobby.
- Estava previsto implementar um outro modo de jogo, que consistia em duas equipas protegerem um ponto no mapa e a outra equipa teria que alcançar esse

ponto.

### 3 User experience design process

A planificação da interface de usuário e de workflow, foi feita de modo a que seja rápido ao utilizador iniciar um jogo, através de um menu lateral sempre disponível (exceto nos ecrãs de jogo, lobby e fim de jogo). A partir desse menu é possível aceder a qualquer funcionalidade da aplicação: adicionar amigos, ver perfil de amigos, ver informação de histórico de jogos, e etc..

Durante o jogo, o ecrã nunca entra em modo de bloqueio se o utilizador não tocar na tela durante um longo período de tempo, sendo possível que o utilizador consulte o mapa o sempre que precisar o mais depressa possível, sem ser necessário desbloquear o telemóvel.

Foi usado uma versão do material design compatível com versão android inferiores a 5.0 de modo a que aplicação fosse compatível com dispositivos com versões android mais antigas.

Os esquema de cores usado foi branco para o fundo do ecrã, ou azul escuro nalguns ecrãs e diferentes tonalidade de laranja para botões e outros elementos, e vermelho para as barras de título dos ecrãs. O menu lateral tem um design simples e muito fácil, onde consta na parte superior o nome do utilizador, foto de capa e foto de perfil. De modo semelhante, também o perfil dos jogadores tem as fotos destes, bem como as suas derrotas/derrotas num pie chart.

Não foi tido em conta vistas para tablets, pois esta aplicação deve ser usada em dispositivos pequenos para uma melhor movimentação durante o jogo.

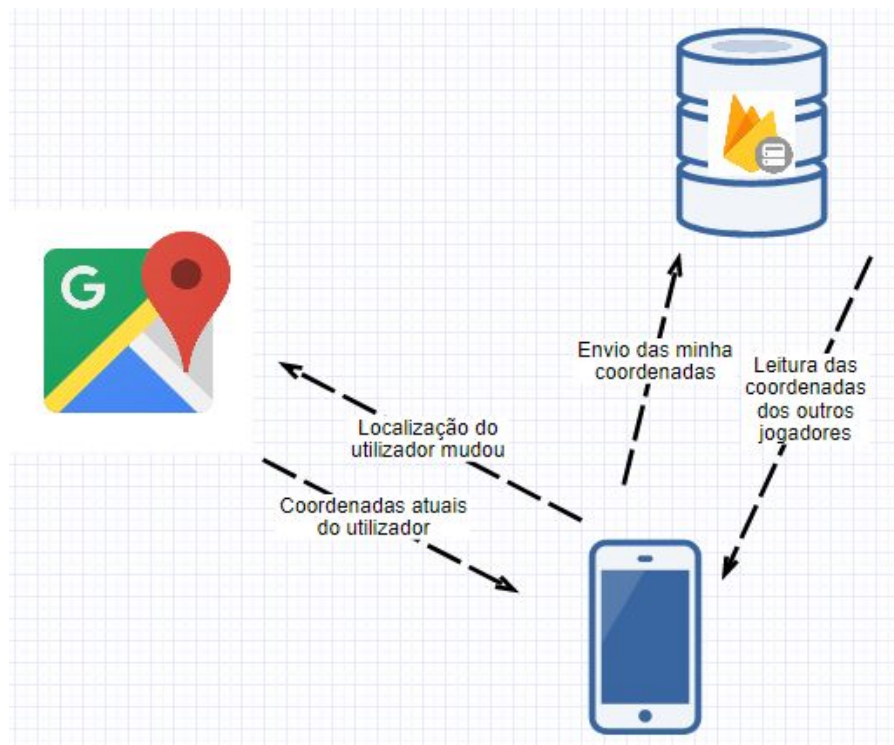
O ícone da aplicação foi criado pelo grupo e representa o ícone que será visível no dispositivo para aceder à aplicação, bem como a imagem que representa a aplicação:



### 4 Architectural plan for the solution

Para o jogo seria necessário obter em tempo real as coordenadas dos jogadores e apresentá-las no mapa de jogo com o menor delay possível. Por isso, após algumas alternativas consideradas, o grupo optou por usar a firebase realtime database, e colocar

cada jogador em enviar as suas coordenadas sempre que estas mudassem para a firebase database, ou seja, sempre que o método `onLocationChanged` (da google Maps API) fosse ativado. Sempre que os dados são atualizados na database existe um listener que lê os dados atualizados e coloca no mapa os marcadores com o email de cada jogador da equipa. Note-se que jogadores jogadores de diferentes equipas só verão os marcadores uns dos outros se estiverem a menos de 100 metros entre si. O diagrama abaixo resume este processo:



A informação associada a cada utilizador é guardada também na database do firebase. No projeto, a classe `UserDataManager` contém métodos que irão, através de listeners, solicitar dados à firebase database, e assim que os receber, colocá-los na classe `UserData`, que contém getters e setters para os dados do user. Qualquer outra classe que pretenda obter informação do utilizador, deverá “subscrever-se” ao listener de `UserDataManager`, e implementar a interface `UserDataManager.UserDataListener`, e consequentemente o método `onReceive`, que é chamado sempre que o listener recebe os dados do firebase, e passa a esse método os dados recolhidos, podendo então os dados serem acedidos. Note-se que na classe `UserDataManager` está também incluído um método para, dada uma string, substituir pontos, traços e outros caracteres por vírgulas, pois como para cada utilizador do firebase é guardado como key no nó o email do utilizador, e o firebase não aceita pontos, traços e outros caracteres nas keys, é necessário fazer esta substituição. O email original é guardado dentro desse nó como filho.

Relativamente á lógica de criação de lobby, o utilizador que cria o lobby irá criar no firebase um novo nó filho dentro do nó “Games” cujo nome é igual ao nome do lobby dado. A informação é escrita na firebase recorrendo à classe `GameData`, que inclui os atributos a

serem guardados no nó do lobby criado.

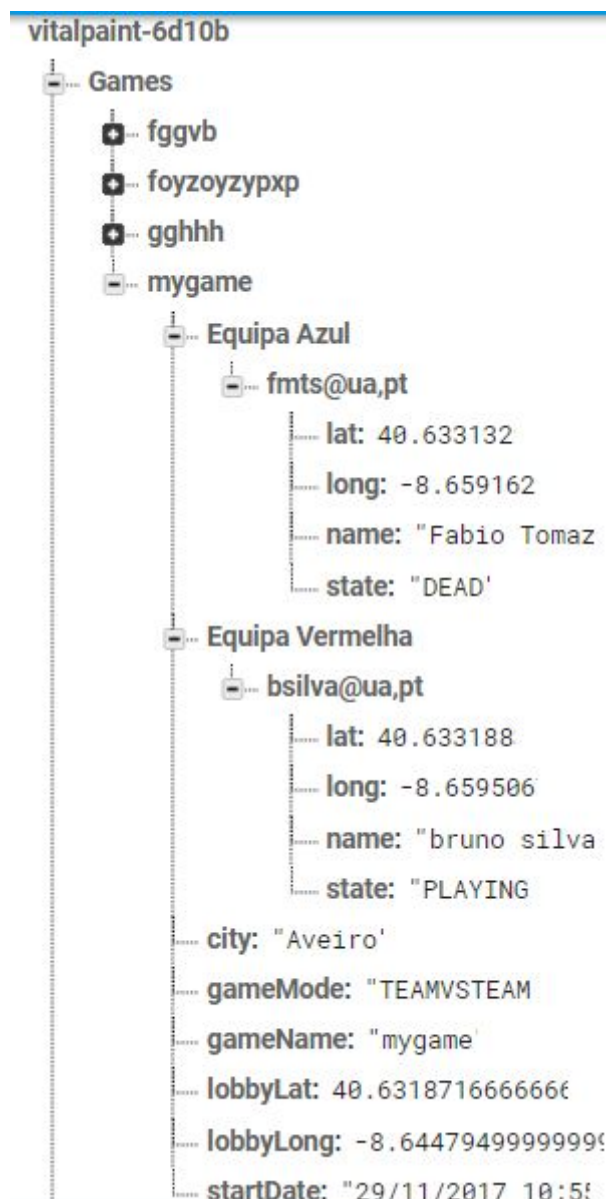
Na listagem de lobbys, quando o user navega para essa atividade, é efetuado uma leitura dos dados da database referente aos lobbys que ainda não iniciaram a partida e que se encontram perto do utilizador. Se houver um novo lobby na área, este irá aparecer na listagem de lobbys. No processo de leitura dos lobbys da database, os valores são guardados usando como auxílio a classe Lobby, com atributos referentes à informação de cada lobby.

No fim de cada jogo, dependendo do resultado do jogo, os dados do utilizador são atualizados e a informação sobre a partida é acrescentada no nó "GamesPlayed". A escrita desses dados é usada recorrendo à classe "GameSession".

## 5 Implemented solution in Android

A aplicação usa a google maps API para obter a localização (latitude e longitude) do jogador e apresentar no mapa de jogo (com vista de satélite). Juntamente com esta API é usado a Firebase Realtime DataBase para obter a posição dos outros jogadores. Cada jogador enviará as suas coordenadas sempre que estas mudem para a DataBase do FireBase e consequentemente os outros jogadores irão ler esses valores na DataBase e colocar no mapa marcadores (cuja cor varia consoante a equipa de cada jogador). A leitura

das coordenadas para quando estes forem atingidos, ou seja, assim que pressionarem o botão "Fui atingido!"



Cada lobby é guardado na FireBase DataBase com os dados do jogo criado (data de início, nome do lobby) e também com a latitude e longitude do utilizador que criou o lobby, bem como a zona a que correspondem às coordenadas, sendo que a aplicação obtém periodicamente nesta atividade a localização atual do jogador, ou usa a última posição já conhecida se não for muito antiga.

À medida que os jogadores se juntam a equipas no lobby, o email destes irá ser adicionado como filho ao nó com o nome da equipa em que se juntaram, que por sua vez é filho do nó do lobby. Cada jogador tem a ele associado a informação acerca do seu estado (a jogar ou "morto") e a sua latitude e longitude.

A figura ao lado mostra o esquema dos nós usados para um lobby e uma partida, com

1 jogador nas duas equipas.

A Atividade de juntar a lobby mostra todos os lobbys que preencham os seguintes requisitos:

- A hora de início do jogo definida é superior à hora atual, ou seja, jogos que já começaram/terminaram não irão aparecer na listagem, não sendo assim possível entrar em partidas que já estão a decorrer.
- A distância entre as coordenadas indicadas no lobby e as coordenadas correspondentes à posição atual do jogador que está a ver a listagem é inferior ou igual a 10km (semelhante à atividade de criação de lobbys).



A informação dos utilizadores registados na aplicação é também guardada na firebase database, e inclui o email, nome, número de vitórias/derrotas, amigos, e informação mais detalhada dos jogos em que este participou.

A figura ao lado mostra precisamente um exemplo dos dados de um utilizador na database do firebase.

Para a autenticação dos utilizadores foi usada a Firebase authentication, permitindo criação de conta nova, sendo cada utilizador identificado com uma password, sendo que esta pode ser alterada a qualquer altura.

Por fim, para armazenar as fotos de perfil e de capa de cada utilizador, foi usado o firebase storage para fazer o upload das imagens sempre que o utilizador mudar para um foto nova, e para obter a foto sempre que necessário. As imagens são guardadas da seguinte forma:

gs://vitalpaint-6d10b.appspot.com				FAZER UPLOAD DE ARQUIVO	+	⋮
<input type="checkbox"/>	Nome	Tamanho	Tipo	Última modificação		
★ As regras padrão de segurança exigem que os usuários se autentiquem				SAIBA MAIS	DISPENSAR	
<input type="checkbox"/>	📁 User Profile Photos/	—	Pasta	—		

gs://vitalpaint-6d10b.appspot.com > User Profile Photos				FAZER UPLOAD DE ARQUIVO	+	⋮
<input type="checkbox"/>	Nome	Tamanho	Tipo	Última modificação		
★ As regras padrão de segurança exigem que os usuários se autentiquem				SAIBA MAIS	DISPENSAR	
<input type="checkbox"/>	📁 bsilva@ua.pt/	—	Pasta	—		
<input type="checkbox"/>	📁 fmts@ua.pt/	—	Pasta	—		
<input type="checkbox"/>	📁 youngf3@live.com.pt/	—	Pasta	—		

gs://vitalpaint-6d10b.appspot.com > ... > bsilva@ua.pt				FAZER UPLOAD DE ARQUIVO	+	⋮
<input type="checkbox"/>	Nome	Tamanho	Tipo	Última modificação		
★ As regras padrão de segurança exigem que os usuários se autentiquem				SAIBA MAIS	DISPENSAR	
<input type="checkbox"/>	📄 headerPic	4,04 KB	image/webp	29 de nov de 2017		
<input type="checkbox"/>	🖼️ profilePic	2,49 MB	image/jpeg	29 de nov de 2017		

### Problemas sentidos durante o desenvolvimento:

Inicialmente, não considerámos que o processo de leitura de database era assíncrono, e não síncrono, o que resultou em algumas exceções de ponteiro nulo, pois esperávamos usar uma variável cujo valor seria atribuída pela leitura da DB, mas devido à sua natureza assíncrona, o programa avançava sem a essa variável ter sido atribuído algum valor.



## 6 Conclusion

A parte mais difícil do nosso projeto foi aprender a usar a firebase realtime database, não só pelo facto de esta ser assíncrona, mas também porque existiu uma dificuldade inicial em acrescentar novos nós no sitio correto e da maneira como se pretendia.

Houve bastantes momentos em que tivemos que colocar em prática aquilo que aprendemos nas aulas (fragmentos, modificar a UI apenas na main thread, usar serviços backend), e outros nos quais podemos aprofundar ainda mais os nossos conhecimentos. Felizmente, a documentação e ajuda online para o desenvolvimento eram vastas, facilitando por um lado o processo de desenvolvimento.

A quando de fazer testes à componente de jogo da aplicação, não foi possível utilizar os emuladores de android, pelo que teve de ser feito em dispositivos reais, e portanto foi necessário reconstruir a apk nos dois dispositivos cada vez que queríamos testar alguma alteração ao código do jogo. Adicionalmente, testes com mais do que 1 elemento em cada equipa foram difíceis de fazer tornando-se muitas vezes frustrante e atrasando o desenvolvimento da aplicação. Além disso, cada construção do gradle era demorada.

## 7 References and resources

Documentação Firebase:

<https://firebase.google.com/docs/database/>

e alguns links usados para suporte a escrita/leitura para a DB:

<https://stackoverflow.com/questions/37031222/firebase-add-new-child-with-specified-name>

<https://stackoverflow.com/questions/40013089/how-to-add-new-child-in-firebase-database-and-android>

Documentação Google Maps API:

<https://developers.google.com/maps/documentation/android-api/start>

Material Design - manter compatibilidade:

<https://antonioleiva.com/material-design-everywhere/>

Fazer listViews customizáveis:

<http://www.viralandroid.com/2016/02/android-listview-with-image-and-text.html>

## Project resources

Project resources for the Android module:

- Code repository: <http://code.ua.pt/projects/vitalpaint/repository>
- Ready-to-deploy APK:  
<https://www.dropbox.com/s/eokbyrokhilysq7/vitalPaint%20apk.zip?dl=0>
- Android Play store page: A aplicação não se encontra na playstore



- Promotional video:  
<https://www.dropbox.com/s/bwig34qsy2ltprw/My%20Movie.mp4?dl=0>