SCIENCES
SORBONNE
UNIVERSITÉ

Project: Crossing Roads

Renault@lrde.epita.fr

LRDE

Demo Video

Global Description

Objectives

- Teach you Android by coding a simple Game (~25 hours of work)
- Make you familiar with Map and (reverse) geolocalisation
  - **Remember to register for Google API**
  - **and activate geolocalisation and reverse geolocalisation**
- Force you to deal with Web communication
  - **You will have to deal with UI thread**

Mandatory (15 pts)

- What you can see on the demo video
- 4 screens
  - **the world**
  - **the game**
  - **the profile**
  - **the statistics**

World's Details

Level Description

- A Level is composed of
  - The name of the level (levelname): for instance Paris
  - The latitude
  - The longitude
  - The difficulty (an integer between 1 and 5)
  - The url of the background
  - The url of the car (left to right)
  - The url of the car (right to left)
  - The url of the pin (for the map)

https://www.lrde.epita.fr/~renault/teaching/ppm/game.txt

- Each field is separated by a **#**
- **Each level description is in only one line!**

This file will be update for the evaluation!

- **Only new levels can be added, no modifications for prior levels**

World's Details

Level Description

- A Level is composed of
  - The name of the level (levelname): for instance Paris
  - The latitude
  - The longitude
  - The difficulty (an integer between 1 and 5)
  - The url of the background
  - The url of the car (left to right)
  - The url of the car (right to left)
  - The url of the pin (for the map)

Data can be fetched using asynchronous task or threads

https://www.lrde.epita.fr/~renault/teaching/ppm/game.txt

- Each field is separated by a #
- Each level description is in only one line!

This file will be update for the evaluation!

- Only new levels can be added, no modifications for prior levels

Game Details

- Important: the roads will be always positioned in exactly the same position

- You have to generate and move cars
  - Best idea: have a list of cars that are repeatedly updated
  - User Timer.scheduleAtFixedRate method to move cars

- You must deal with the bird
  - Cannot go backward!
  - Detect long press and move the bird consequently
  - Detect X-axis moving to force left or right

- Detect Collision (smooth mode is ok)

Result's Details

- A list of 4 elements
  - name
  - hometown
  - time
  - level name

https://www.lrde.epita.fr/~renault/teaching/ppm/results.txt

  - Each field is separated by a **#**
  - **Each result description is in only one line!**

- **The user must be geolocated to display results**
- **Note that a player can play multiple time before accessing the results list**

Profile Details and Music

- Profile is the Simplest screen
  - A picture
  - A name
  - A motto

- The music is playing all around the game
  - No restart when changing screen (**please!**)
  - Use a Service to do that
  - You can have songs for each level if you want it

- You are free to choose the bird (or other) you like the most

Tips

- Reverse geocoding from your actual position

```
https://maps.googleapis.com/maps/api/geocode/json?
latlng=48.9167,2.2&key=YOUR_APIKEY
```

- FloatingButton for actions in the map
- Fix the orientation to landscape so that you don't have to handle rotation
- Use Handler for moving the bird and onTouhEvent
- Fix the size of your images
- If you opt for multiples activity startActivityForResult could be a good choice

Options (1/2)

- Support rotation on all screens of the App. (2.5 pts)
  - During the game your bird must be correctly relocated
- Persistence across re-opening (2.5 pts)
  - Save results, name, and motto and other stuff
- Score sharing (2.5 pts)
  - At the end of the game or by selection on the score screen
- Widget for displaying the bests results (2.5 pts)
- Allow user to create its own level (2.5 pts)
  - Create background, fix image for cars, etc.

Options (2/2)

Sort list of results (2.5 pts)

- According to user name, levels, time, location (all options must be available)

Offer the possibility to record a game (2.5 pts)

- And allows the user to browse the videos
- Be careful to lag...

Connect with iOS through "Bonjour" (2.5 pts)

- Share results with "bonjour"
- Develop an iOS application that displays these results

Other ideas welcome !

- Just ask me so I can validate your option

Submission

An archive *.tar.gz containing

- README.txt: this file must be at the root of your archive
  - **Explain your architecture**
  - **Describe what are the implemented options**
  - **Describe the problems you have <encountered**
- The APK of your application: this file must be at the root of your archive
  - **You can find it**
- The source of your application
  - **Clean them!**
  - **Document them!**

The name of your project must be Student1_Student2

Summary

A simple application

- Less than one day of work for an experimented developper
- Regroup all elements required by a modern application
  - Connection to Google API
  - Parsing
  - Touchs events
  - Multiple Sceens

From this application you can build your own

- Pokemon Go
- Ingress
- Crossy Road
- Temple Run