FEUP - LPOO Report

Henrique Ferrolho - henriqueferrolho@gmail.com
Hugo Sereno Ferreira

Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

Abstract: Quick summary about the applications developed for LPOO
Keywords: android, app, monty hall problem, snake

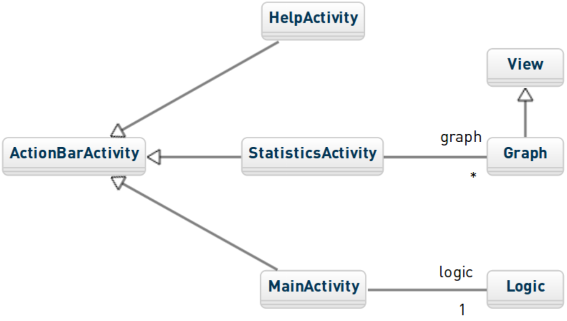
1. Monty Hall Problem

1.0 Summary

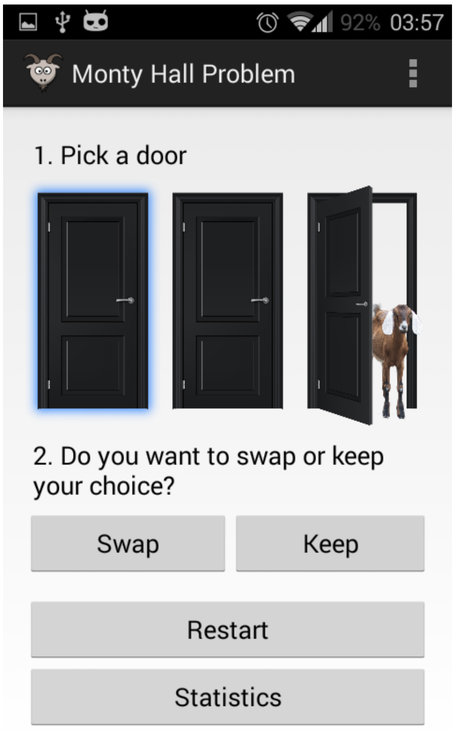
Monty Hall Problem is my first android app ever and the goal behind its development was to launch myself into the android world.
This app gave me the opportunity to learn about android's basic building blocks, activities life cycles, custom views and much more.

The app itself consists of an illustration and simulation of the Monty Hall Problem. The user is presented with three closed doors. After pressing one of the doors, one of the remaining doors opens. Upon seeing that door content, the user can either press "Keep" or "Swap", to keep the selected door or to swap the selection to the remaining unselected door respectively. Afterwards, the player wins if the selected door has a car, or loses if it has a goat.
Whenever the user plays the monty hall problem, the results are recorded. The application has a statistics activity where the user can view his/her saved results. Among this statistics are: the number of times played, times swapped and times kept; as well as the number of times the user won and lost for each decision.
There is also a Help activity which contains the Monty Hall Problem description, concept and how it works.
This app supports two different languages: English and Portuguese.

1.1 UML



1.2 Screenshot


Monty Hall Problem

1. Pick a door

2. Do you want to swap or keep your choice?

2. Snake

2.0 Summary

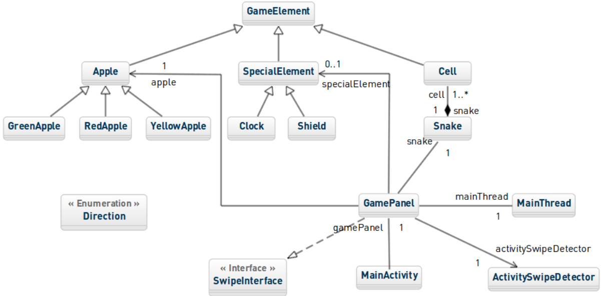
My goal while developing snake was to learn how I could implement my own gestures, my own game cycle and furthermore how to completely control the update frame rate and draw frame rate properly.
Since the game logic and structure was not my main focus, snake was the perfect game to accomplish my goal: short, simple, and easy to implement.

Snake ended up being a nice and fun application:

The controls are simple: swipe up to go up, swipe left to go left, etc.
If swiping is not the user's preferred method to control the snake, he/she can also change the snake's direction using clicks: if the snake is moving horizontally, tap anywhere above the snake's head to make it go up, or below to make it go down; if the snake is moving vertically, tap anywhere to the left of its head to make it go left, or anywhere to the right to make it go right.

Moreover, there are three kinds of apples which the snake can eat, each with different scores.
Besides the classic apples, there are two special items: the clock and the shield. The first slows down the snake for 10 seconds; the second creates a shield around the snake, granting the player the ability to go against walls or making the snake eat itself (both making the snake lose the shield and therefore losing these capabilities until another shield is caught).

2.1 UML

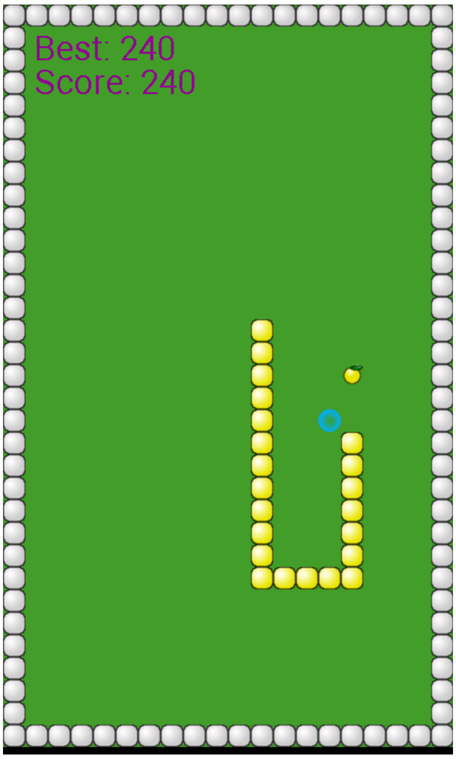


2.2 Design Patterns

Snake makes use of the following design patterns:

- Flyweight – used to load game bitmaps: every apple of the same kind shares the same bitmap, which is loaded only once at game initialization;
- Observer – used to control the snake current movement direction.

2.3 Screenshot


Best: 240
Score: 240