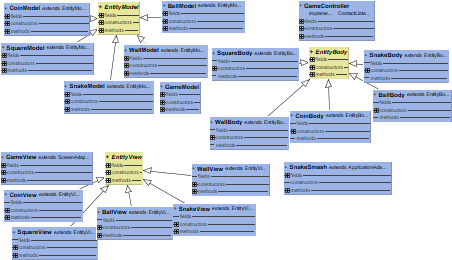
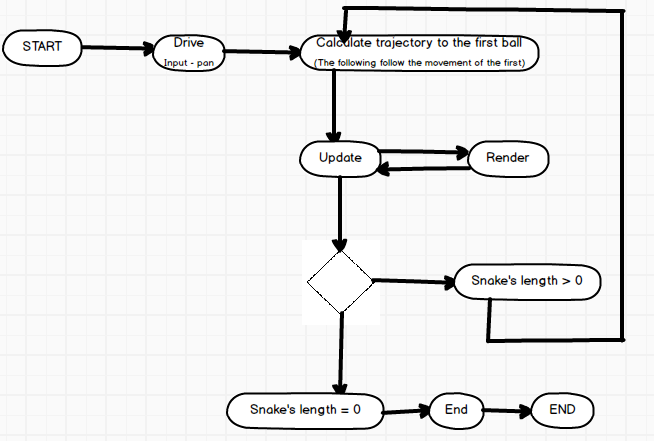
LPOO - Final Project Intermediate Check-Point

1. Architecture Design a. UML

[](https://user-images.githubusercontent.com/28101797/39429272-4888c704-4c82-11e8-8ec4-12125633ab25.png)

b. Behavioural Aspects[](https://user-images.githubusercontent.com/28101797/39429299-606d80ee-4c82-11e8-8a51-94633d411122.png)

c. Design Patterns

• Singleton - This will be used in game class because it knows all but there should be only one game class accessible to all.

• Strategy – This will be used in different levels of game because the snake should have different behaviours.

• Template – This will be used to draw the world after it’s updated.

1. GUI Design

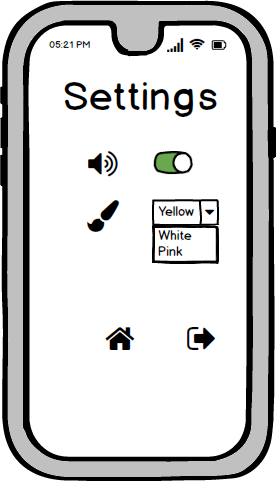
a. Main Menu

Here we have a mock-up of the main  
menu of the game. The game will start when the player chooses one of the three existent difficulties (beginner, intermediate and impossible). In this menu we will also be able to go to the settings or the scores menu, share the game results on social media and exit the game.

[](https://user-images.githubusercontent.com/28101797/39446024-f756d1d6-4cb4-11e8-89ef-ae50f1ee944e.png)

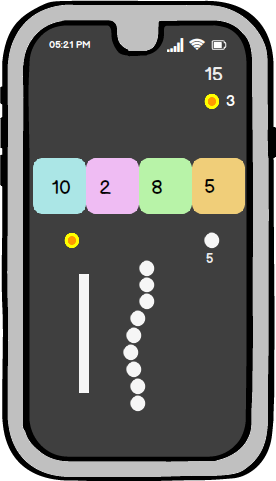
b. Settings Menu

In the settings menu we will have na option to remove/add sound to the game. The player will also get to choose the color of the snake he will be playing with. After changing the settings, the player can return to the main menu or just exit the game.

[](https://user-images.githubusercontent.com/28101797/39446089-2bba89ae-4cb5-11e8-9d3d-b636337aa6df.png)

c. Game Mode

The objective of the game is to get the largest Snake possible. When the snake collides with a circle, the number of circles indicated will be added to the snake body. Then, when colliding with a square the snake loses the number of circles shown inside the box. The player will lose the game if the snake disappears (number of circle = 0). In the game, the player may also collect coins, that will increase the final game score. The number of coins gained and the current score will be shown in the game.

[](https://user-images.githubusercontent.com/28101797/39446134-5767fc26-4cb5-11e8-865e-b2f4e23ce464.png)

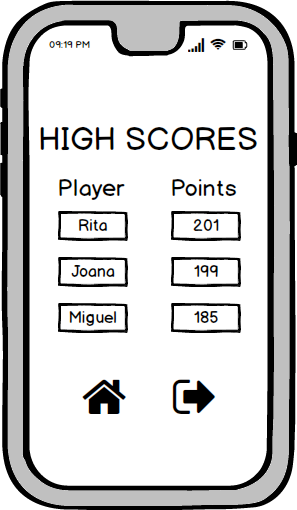
d. Game Over

When the game is lost, the player can return to the main menu, leave the game or play again. the score will appear on the screen and a crown will indicate if a new high score is reached.

[](https://user-images.githubusercontent.com/28101797/39446173-857b811e-4cb5-11e8-89d1-e593ac02609e.png)

e. Scores Menu

This menu will show the highest scores obtained in the game. The player may return to the main menu or leave the game.

[](https://user-images.githubusercontent.com/28101797/39446102-39908d80-4cb5-11e8-8ebb-d592c69adc28.png)

1. Test Design – Unit tests

a. Test collisions with the different objects of the game: squares, walls, circles and coins;

b. Test if when the snake disappears (number of circles = 0), the game is over;

c. Test if the snake gains the number of circles indicated when colliding with a circle;

d. Test if the snake loses the number of circles indicated when colliding with a square and the square disappears;

e. Test if the number of coins increases by one when colliding with a coin;

f. Implement test for the diferente levels: beginner, intermediate and impossible;

g. Test buttons behaviour (play, exit, settings, sound, select levels,…);

h. Test snake movement;