

# Minigolf Mobile Application Proposal and Design Document

Salman Hamid, Std#: 1802968

sh18670@essex.ac.uk

## Introduction:

The following document will outline the design elements of the Minigolf mobile game application. The objective of the application is to provide a fun and enjoyable experience for the user while maintaining a simplistic “Pick up and play for two minutes” user experience. The goal is to provide a game the user can play for a few minutes, several times a day.

The game will consist of 9 levels, with the possibility of adding additional stages later. A future implementation may incorporate a procedural level generator to provide the user with potentially infinite stages to play.

The business model of the game will be ad-based. There will be space provided at the top and bottom of the app screen to accommodate banner ads.

## Background:

The idea for the app came about through analysis of mobile game player habits and trends. Most mobile users spend time on their phones in short bursts when they are waiting for something else, (i.e. waiting in a queue, on the bus or train). Therefore traditional game ideas involving narrative and plot is not feasible as in Computer and Console games.

Therefore the game needs to be simple in its scope, yet still provide engaging gameplay. Looking at other successful titles such as Angry Birds and Cut the Rope shows that there is a large market for simple level based games involving physics.

## Features:

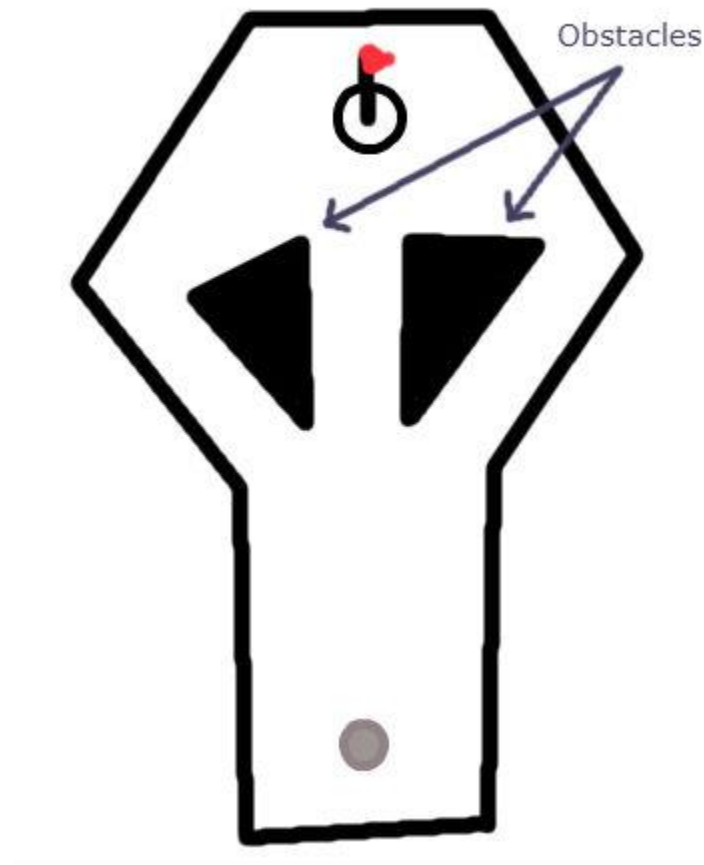
The game will initially consist of nine levels. The basic gameplay will be of players needing to press down on the golf ball on the screen and dragging in a specific direction to launch the ball. The player will often need to ricochet the ball off various obstacles and props on the level in order to pot it in the hole. They will be scored based on how many shots they take to reach the hole.

The following gameplay features will be utilized in the design of the levels:

### Obstacles

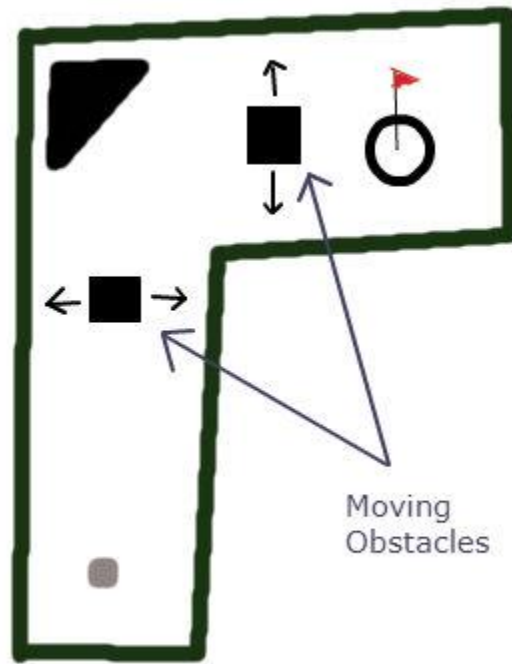
The most basic element of the minigolf game, obstacles are physical objects that can collide with the golf ball as is appropriate in the physical world. These can range from the very walls of each of the individual courses to wooden blocks placed to obstruct the path to the hole.

The obstacles can be both a hindrance and a tool to be used by the player. Because they interact consistently with the ball, the player can find creative uses for them by bouncing the ball off them to hit the hole from a different angle.



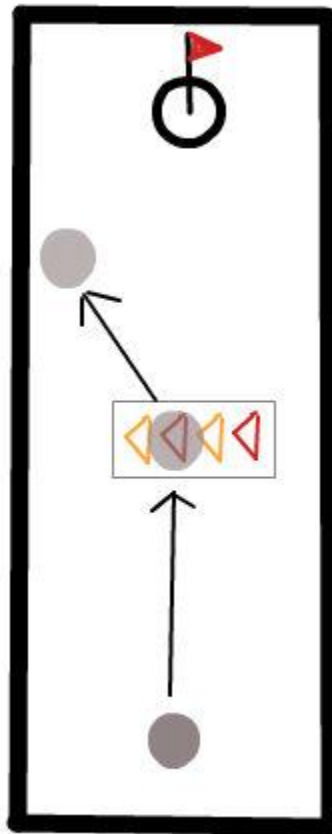
### Moving Obstacles

These are slightly more complex than regular obstacles as they move from place to place on the course. They still interact with the ball as solid objects, and are therefore able to be collided with, however they can also apply a force onto the ball due to their own movement. Players will need to pay attention to their movement pattern in order to bypass them successfully.



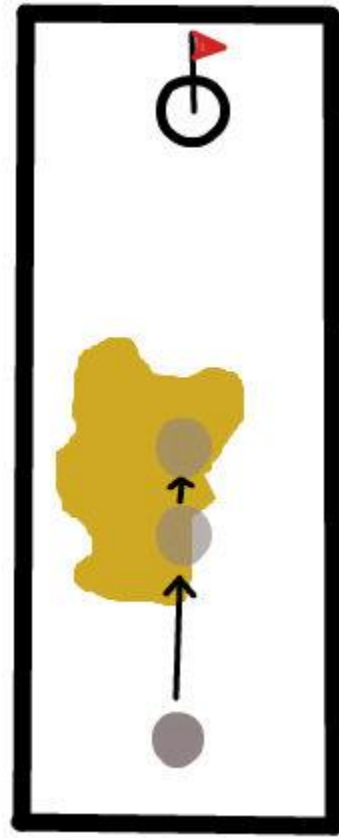
### Speed Boosts

A unique gameplay element, speed boosts are areas on the ground that apply a force on the ball as it passes over them. They are not physical object, and cannot be collided with, but will instead impart a force onto the ball in a specific direction. These can both hinder and aid the player, depending on how they are used.



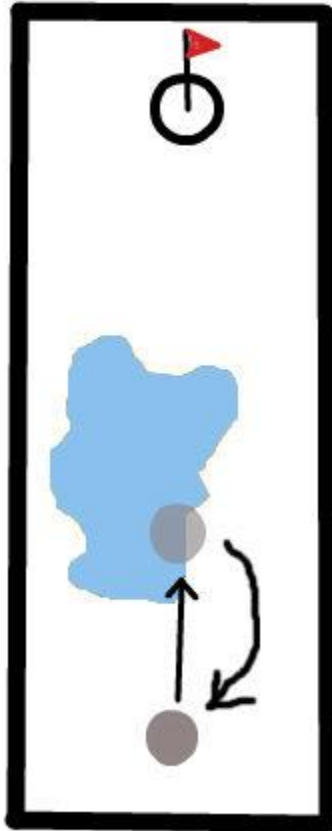
### Sandpits

Sandpits are areas on the course where the drag coefficient is significantly higher. This means that Sandpit regions are places the player will want to avoid going through.



### Water Hazards

Water Hazards are dangerous traps on the course. If the player knocks their ball into a Water Hazard, their ball is lost and they are sent back to the starting location.



### Design & Structure:

The application will consist of a simple menu interface with which the user may navigate between different screens. The game will keep a track of how many stages the user has unlocked by keeping track of an integer that can vary between 1 and 9. 1 means that the user has only the first stage available to them while 9 means that the user has unlocked the final stage.

### Activities

#### Main Activity

The first of these is the Main Menu, which will be our main activity and the one users will first land on. From this activity users may press on either the Stage Select or the Options buttons to navigate to these other activities.

#### Options Activity

In the options activity, users are able to alter some global variables that control whether the game music and sound effects will play or not.

#### Stage Selection Activity

From the Stage Selection Activity, the user is able to navigate to two other activities; the user may click on any of the stage buttons to move to the Gameplay activity, or the back button to return to the main activity.

#### GamePlay Activity

This activity will have nine different fragments, each of which corresponds to a unique stage. Clicking the pause button will open a dialog from which the user can restart or return to the stage selection activity.

### Plan:

The first part that will need to be implemented is a basic framework for the app. This basic application will feature only a Main Menu, a Stage selection Activity and a blank Gameplay Activity. This is only expected to take a day to implement as it will be a bare-bones functional design.

Following this, the next part is the framework of the physics engine. The engine that will be implemented and extended is the 3<sup>rd</sup> party library AndEngine [1]. The AndEngine will be used to create classes to represent the walls of the course and various obstacles. This part is expected to take three to five days, with another day allotted for learning and unexpected bugs. After that, once the framework is implemented, two days will be allotted for the design of the nine levels themselves.

Finally, the remainder of the allotted time for the assignment will be spent implementing other game features such as score tracking and polishing design. This part is expected to take two to three days, but due to allowances for bugs and inexperience, may take up to a week.

### References

- [1] N. Gramlich, "AndEngine," 2011. [Online]. Available: <https://github.com/nicolasgramlich/AndEngine>.