*Jack Maguire*

*Candidate Number: 0109*

*Candidate Centre: 51818*

*apex turrets*

# Table Of Contents

[Table Of Contents 1](#_Toc29309638)

[Analysis 2](#_Toc29309639)

[Research 2](#_Toc29309640)

[Background and Problem Definition 2](#_Toc29309641)

[Analysis of Similar Systems 2](#_Toc29309642)

[Problem With Current System 3](#_Toc29309643)

[Identification of Users, Their needs and acceptable limitations 3](#_Toc29309644)

[Identification of Users and their needs 3](#_Toc29309645)

[Acceptable Limitations 3](#_Toc29309646)

[Objectives 3](#_Toc29309647)

[Design 5](#_Toc29309648)

[Technical Solution 5](#_Toc29309649)

[Testing 5](#_Toc29309650)

[Evaluation 5](#_Toc29309651)

# Analysis

## Research

### Background and Problem Definition

Currently, at KCS, we have about 1000 pupils and numerous members of staff, and other than free periods, we have two breaks, one in the morning for 20 minutes, and one in the afternoon starting at either 12:35 or 1:15 (sometimes there is an extra lesson), and ending at 1:40 (for sports), or at 2:10 (for lessons afterwards). In the other half of the school (the junior school), there was a well-equipped playground, with monkey bars, play-frames and table-tennis tables. There was also the library (homework was forbidden in the day). In the Senior School, however with vastly more pupils, and there are only spaces in which to gossip, or the library in which there are computers (homework only), or the ICT Suite (in which Games are banned).

This creates a problem, that the people who don’t gossip, can get easily bored, with only homework, reading or sitting around to do. My proposed solution, a Tower Defence Game aims to solve that, in that the IT Staff allow self-coded Games, and that no previous pupils have made large scale games, similar to any standard of Game Development (pupils are taught only python at a basic level, and most students take a dislike to the Subject due to the teacher for the school being shouty, and that you are only taught in the first year (Year 9)).

A Tower Defence Game, is a game which is 3rd person based, and one has enemies to kill, but instead of shooting them like most games, you build turrets or towers to kill the enemies. These cost money, but as enemies die, they give you money, and you can build more towers. If the enemies get to the end of their path however, then you lose lives, and if you have 0 lives, then Game Over. You win the level, if you kill all of the enemies. As you progress through the game, there are more types of enemies, and towers, and you can upgrade the towers. It resembles some other games like Factorio, in that you try to automate it so that you don’t have to do anything. In more advanced versions again, you have power-ups or heroes.

## Analysis of Similar Systems

One good example of a tower defence game is Kingdom Rush, a popular mobile game that features a lot of stuff. The UI is clear and well-defined, and there are a lot of levels, to the point where I have not yet come close to finishing the game. It features no ads, and the only way of them making money, is buying more in-game currency (used for things like ‘Freeze all Enemies’), or buying heroes (they sit where you tell them, and kill enemies as an AI would in a typical shooting game).

However, this game is only available on mobile, which whilst effective for planes or journeys, limits its available processing power, and neutralises its uses for my use case (phones are banned throughout the day for my part of the senior school), and downloading emulators is not allowed, as are any other exe files, as well as other solutions for playing this. Images of the first level are shown below.

A picture containing indoor, table, grass

Description automatically generatedA picture containing indoor, grass, table, cake

Description automatically generated

## Problem With Current System

## Identification of Users, Their needs and acceptable limitations

Questions for Clients:

Q1: What do you currently do during our breaks?

Q2: What problems are there with this?

Q3: How do you think these problems could be tackled with my proposed solution?

Q4: Do you believe that there are any essential features that are currently missing?

Q5: Are there any features with my proposed system you find unnecessary?

Q6: How do you think new 4th Formers will react to this system, different to the old one?

Q7: Do you see any possible limitations with this system?

Q8: Do you have any ideas about UI, or anything else to do with the Front-End?

### Identification of Users and their needs

### Acceptable Limitations

## Objectives

1. The game should not be repetitive.
   1. There should be different turrets.
      1. Firstly, there should be at least two different turret templates.
      2. Then there should be upgrades available for each turrets (at least one, but more should be able to come.)
         1. An upgrade constitutes of an increase to a statistic positively, such as more damage, or shooting more often.
   2. There should be different enemies
   3. There should be different levels
      1. First, will be set levels, in which enemies come and go, in a set order (determined by a config file), and score will be determined by money remaining, money used, and hearts remaining.
      2. Then there will be an infinite ‘gauntlet’ mode, in which enemies are thrown semi-randomly (there will be checks not to be too slow, or too fast), and score is determined based on factors like time between enemies, enemies killed and the above factors for the set levels.
2. The game should be able to differentiate users.
   1. A user can select a difficulty, and have that kept for them.
   2. Each user should be able to keep a unique account.
      1. A password will be required, of a certain length.
      2. As well as a username of a certain length.
      3. The password will be hashed using SHA256, and stored in that way.
      4. The attributes of each player scored locally:
         1. Overall Score,
         2. Levels complete and score for each,
         3. Gauntlet high score;
      5. As a new user is created, a new file is created with the attributes of that player.
3. The game should be as efficient as possible.
   1. Storage Space Efficiency
      1. All turret, enemy, and level data will be stored on a Github repository, and fetched at runtime.
4. The players should be able to share statistics.
   1. The email will be stored, so a python script could be used to email others about the score.
   2. There will be a leaderboard for set levels, and for the Gauntlet.
      1. On the leaderboard, it will be in rows, sorted by score, and only display the top 10, and it will show the username, a message, and their score.
         1. In case of any unsavoury language, each word will be checked against a database (<https://raw.githubusercontent.com/RobertJGabriel/Google-profanity-words/master/list.txt>), and censored appropriately (eg. F\*c\*).
5. Running Example:
   1. The user will be presented with a main menu, with a Welcome Screen and two buttons to either login or to see the leaderboard.
      1. If they click on the leaderboard, then it will show it, as in 4.2.1
   2. If they click login, a new window will open to login, with an simple GUI to create a user, or to login.
      1. The details are stored locally on a cfg file.
   3. Once the user has logged in, they will be able to choose to replay a level, or to progress to the next.
      1. If they choose to progress to the next level, it will be the same as a previous only a different level shown.
         1. Then the level will be shown, with a grid system for the tiles, an enemy base, a home base, and GUI for purchasing towers, as well as to inform about lives and money.
            1. The grid system is that each icon has an ID, and dependant on a cfg file, it dynamically shows different tiles to change up the level.
            2. At the enemy base, the enemies spawn, and they slowly make their way across ‘path’ squares, and attempt to get to the home base. If they reach the base, then the player loses an amount of lives, and points for the end of the level. If the player runs out of lives, then Game Over.
            3. To combat the enemies, the player can buy turrets, and these can shoot the enemies, to kill them before they reach the base. The turrets get better, but so do the enemies, and turrets can be upgraded.
            4. In the shop, the player buys the turrets.
      2. The player could also play the Gauntlet, which is the same, only the level never ends until game over.
   4. If they choose to replay a level, they play through it as a normal level, and when it finishes, the old data from the player’s file is deleted, and the new data is inserted.

# Design

# Technical Solution

# Testing

# Evaluation