# Introduction

The gaming industry has one of the highest revenues, with an estimate of making 217 billion USD in 2022. The revenue of the gaming industry is still growing at a rapid rate due to the increase in mobile gamers that have completely changed the industry forever. However, due to the mainstream platforms that are charging high fees for the distribution of games, it means that the developers are being forced to bring back the development of web-based games.

In this project I am developing a web-based game application that will be made using HTML, CSS and JavaScript. I am creating this report alongside the game which will display the plan for the project of the COMP1004 module. This project will follow the software development lifecycle to ensure that the development of the game will be structured. This means that I will be able to plan and manage my time more efficiently, resulting in a more successful completed game that will meet the requirements and deadlines.

The report shows the steps I have taken to complete this game, highlighting the issues that may have had an impact on time when designing and developing the game, and any concerns that I have discovered, which may be legal social and ethical, throughout the project. It will also present the requirements that I will follow to ensure the success of the game, which will also help with the development of the architecture. I will finish the project by talking about the sprints and how they helped with the planning of tasks for each week. An evaluation will follow, explaining how the project went and whether it was a success.

# Software development lifecycle

For this project I have decided to use the scrum methodology. I believe that following this specific methodology will result in a more successful game. This is due to the sprints that will be taking place every two weeks in the project. This structure allows the tasks to be reviewed more often and it will ensure that only specific tasks will be completed during each 2-week sprint. This means that the team will have a stronger idea of what they are doing, keeping them motivated to stay focused and to complete the tasks within the sprint.

These are the software development life cycle steps:

* Planning
* Requirements Analysis
* Design
* Implementation
* Testing
* Deployment
* Maintenance

Following these steps ensures that the result of the project is completed by the deadline, with any issues that may arise is tackled much more efficiently. It could take much longer to complete a project and many unexpected issues could occur if this structure is not followed.

The main items that are featured in the scrum model are:

## Product backlog

This is used as a plan for the product to show the priorities of each task that will need to be completed. This is to ensure that the main tasks are being completed first to get the overall functionality and then any other added details that might want to be included can be added at the end if there is enough time. This is a very useful document that helps keep the user on track if it is followed in order of the tasks to be completed.

## Sprint backlog

The start of the sprint will display the tasks that need to be taken place for the next two weeks and what items of the project will be worked on. At the end of each sprint, the project owner will decide which items have been completed. The completed software will also be displayed. Throughout the project there will be daily scrum meetings which will highlight the work completed on the previous day and the work to be completed the following day.

## User stories

The user stories are featured in the product backlog but are not put in order of priority. They are used to show what the user actually wants from the game. These are especially important to follow because the users will be playing the game and therefore implementing what they would like will overall result in a successful game.

# Game design document

## Executive Summary

### Game Concept

The player has to get the ball in the hole in as little as shots as possible. There will be 9 levels or “holes” that the player will compete in. At the end of the 9 holes the player will be able to see the leaderboard which will display the overall score and how many minutes were spent playing.

### Genre

The genre of the game will be an online retro sports game.

### Target Audience

The game will be targeted toward people who play golf. Males will make up the majority of audience that will enjoy this game as they are the majority of golf players. It will be aimed at ages 18-24 as people in this age category will be more likely to play games and golf.

### Project Scope

The requirements that have been asked are to track the players username, score, level and minutes spent playing, which will be stored in a flat file using json. There needs to be some custom art for the game pages and the game needs to be fully functioning.

## Gameplay

### Objectives

The objective of the game is to get the ball in the hole in as little goes as possible. The players with the least scores will be displayed on the top 10 leaderboard.

### Game progressions

The player will move onto a different level each time the previous level has been completed. I will be implementing 9 levels for the player to complete.

### In game GUI

The home page will feature buttons which will allow the player to start the game, see the controls, open the leaderboard and exit the game. When the game has started, there will be on screen buttons which will allow the user to set the angle and power of the shot and there will be a button to click to hit the ball.

## Mechanics

### Rules

The player will only be allowed a maximum of 10 shots before automatically being moved onto the next level.

The ball cannot leave the game canvas.

If the ball gets put in the whole the player will move onto the next level. After the 9th level the player will see their score and hope to be on the leaderboard.

### Physics

The will have similar physics to games like 8 ball pool and raft wars. The player will aim using an arrow and drag down a bar to get the power of the shot. For example, If the player aimed upwards and did full power, the shot would go very high and not far.

## Game Elements

### Characters

Customizable characters will be implemented in the game. Players will be able to choose from a different range of characters that I will create myself and a range of colors for the ball. I will implement over 10 characters that will each have a pixel theme.

### Leaderboard

During each level the score will be displayed on the top of the screen. At the end of a the game there will be a leaderboard showing the score for each of the levels and the overall total score. There will also be a column for the amount of minutes spent player. I want to implement the leaderboard so that the data for the player will save each time it is played so that the player can see the previous progress.

### Level Design

Each level will have a different theme to it, for example one level may have an underwater theme and next will be set in the space. This will keep the user more engaged in the levels and always wondering what he next level theme will be.

## Assets

### Music

Each level will have different music that will be related to its theme. This will give the user the full experience of the levels theme. There will also be music that will be playing on the home screen and once the game has finished.

### Sound Effects

Every time the player hits the ball a different sound effect will play depending on how hard the ball has been hit. If it is hit gently then there will be a soft sound, but if it is hit with more power there will be a hard sound.

### 2d/3d models

I will be implementing 2d character models which I will be creating myself. The background will be a 2d model that I will get from an assets store. If I cannot find some backgrounds, I will be creating them myself. The ball will also be 2d and implemented from an asset store.

# Project Plan

## Sprints

## Product Backlog

A list of text on a white background

Description automatically generated

# Project vision and Background

RetroGolf’s vision is to deliver a playable SPA web game that will give off a mini golf experience, combining aiming mechanics from games like raft wars to 8ballpool. Players will find themselves in a nostalgic journey, competing in nine unique levels to get the lowest scores possible. The game will be engaging and competitive fun, with features of customizable characters and golf equipment and also a leaderboard to keep track of your score and minutes spent playing.

The aim for Retro Golf has emerged from a passion for nostalgic games and a desire to relive the joy of classic mini golf into the modern age. The inspiration has come from timeless games like raft wars to 8 ball pool. The development of Retro Golf has not only been fueled by the love for gaming, but also by the ambition to create a vibrant online community. Within this community, Players will be able to connect, compete and share their experiences.

Through my extensive research and analysis, I have discovered an expanding audience for web based games. Furthermore, I have identified that as the gaming industry continues to develop, there are very few mini golf games that not only offer immersive gameplay mechanics, but also focus on the sentimental value of retro aesthetics. With an increasing demand for engaging online games, RetroGolf aims to provide players with a nostalgic journey whilst introducing fresh and exciting elements into traditional mini golf gameplay.

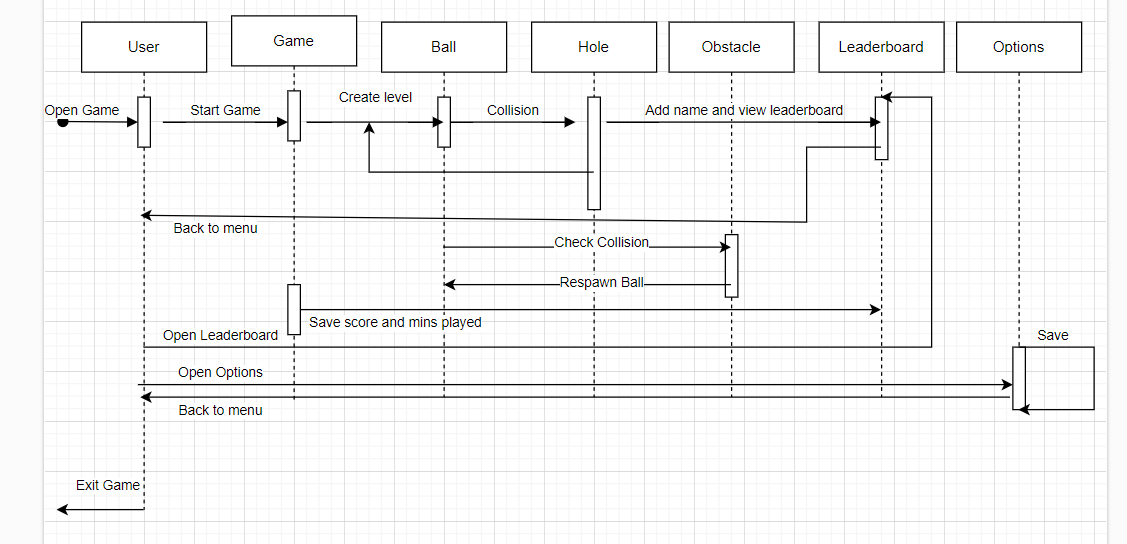
# User stories and associated user case scenarios

# Architecture – UML models

## Package diagrams

## Class diagrams

## Sequence diagrams



# Sitemap

A diagram of a game

Description automatically generated

# Wireframes

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Noted issues and constraints

# Poster

# Github repo link

# References