SCC 210 Group Project:

Mini Golf



Group B4

Lewis Allen, Chris Hicks, Alex Morris, Joseph Richardson,

Sam Sanad & Tom Wright

Contents

[Introduction and Project Ideas (20%)](#_p80mk6qh3pf7)

[Arcade](#_acdck6m0e084)

[Sport](#_1fob9te)

[Simulation](#_7omh5ao164nu)

[Strategy](#_hnnx0yujdxoi)

[Platformer](#_3dy6vkm)

[Game Design (40%)](#_tszdly22le2x)

[Specification](#_lnwjloq8rap)

[Selected Game](#_ckrlh27y86fs)

[Gameplay Features](#_ajduhkmcorzc)

[Pursuing and Achieving Goals](#_qggjr7b4wwsj)

[Constitutive](#_jymcu0ke7uv8)

[Operational](#_qyw5iz4kn3w3)

[Implicit](#_xngovaxizdhy)

[Software Engineering (15%)](#_hw7bx6bwzbxx)

[Requirements](#_j24xlvbx3q3p)

[Functional Requirements](#_7z7kcelem8tj)

[Non-Functional Requirements](#_6m2uqcjwq5u3)

[Acceptance Tests](#_2eq10sewgyo9)

[Usability Tests](#_qmib3irhbmt3)

[Implementation Plans (15%)](#_bc81i7ahnvn0)

[Task List](#_pm65bme4w1th)

[Implementation Milestones and Deliverables](#_loguc5weyt6b)

[Activity Network with Critical Path](#_dyrpbsmk8mb9)

[Gantt Chart](#_t8by4qst2azp)

[Contingency Plan and Risk Assessment](#_q3jx12mduh53)

[Final Design](#_1gp9zx5h3hrs)

[See Appendix B for larger images](#_iu3heqherzsp)

[Appendix A: Initial Design Idea Sketches](#_4ik1m9v0xisl)

[Appendix B](#_5vmpb3ij12tl)

[Final Design](#_q1y2i0sgt2nd)

# Introduction and Project Ideas (20%)

The aim of the project is to choose and create a 2D style game, using programming skills learned throughout the scope of the Computer Science degree and those developed independently. The game should be chosen based on its ability to engage the player to continue and not become repetitive or boring for as long as is possible. Using the skills, we have learned throughout the course, we aim to produce a well-designed game, starting with the project specification and finishing with a fully functional delivered game. The first step in creating a game, is choosing a suitable game idea to pursue. We began by individually submitting ideas to the group, at which point they were discussed and evaluated based on their merit and ability to support the deliverables required by the assignment.

See Appendix A for initial sketches of the following ideas.

## Arcade

Circular Block Breaker

Taking influence from the classic Block Breaker, the idea of this is to create a game where a circular layout of blocks is gradually destroyed by the player by rebounding the ball, like playing tennis with a brick wall. This varies from the traditional block breaker game as the game would exist in a circular space, as opposed to the classic flat layout. The blocks would rotate with several layers of blocks inside each other and would have a platform on the outside of the circle where the ball would bounce off of. To add more features of the game features such as speed and time can be implemented to allow the game to run faster making it harder, adding a time trail and a task to make the game more challenging, more levels and multiple bounces to break a block.

Reason for elimination:

The idea of this game was quite minimal and we felt that the ability to keep a player interested wasn’t there. One specific issue was concern over how to continue expanding the game. If the player won the game, we were concerned about how we could develop the game further without constant repetition, which would affect the player's’ ability to stay enticed in the game. This game wasn’t abstract enough, the player would just be looking a ball and some blocks rotating, without scenery and without an environment for the player to associate with. In order for this to be a valid choice it would have to be developed far beyond the original ideas we had.

## Sport

Mini-golf

Players must hit a ball into a hole at the end of the course, with the score taken relative to the par of the hole. As the game continues more obstacles would be added, such as windmills, slopes and pipes, to increase the difficulty of the game.

Card Memory Game

Idea of the game is to be shown cards, remember the information on them, then they are removed from the screen and certain questions are asked to see how well you have remembered the information. The game has some features which can be added to increase complexity, time constraints, and special bonus rounds that could present very difficult scenarios. Overall, the bonus features will have limited scope.

Reason for elimination:

We chose to not pursue this game for a few reasons. First being that it is far too repetitive, having a set of cards and using your memory there is only so much you can do, there is little scope to make it exciting, it would feel like a chore to play. Secondly, there are a large number of similar games already available, making another one has little purpose as they are all very similar. Finally, it's very simple - the project we are looking for needs to have a decent level of difficulty for us to implement, and have the ability to expand on it above the base idea.

## Simulation

A&E Simulator

A time management game where patients come in with varying injuries and the player has to allocate staff and other resources in order to treat/save them. This would involve making sure that the most life threatening injuries are treated first, people are seen in a reasonable time and that staff have time off to rest. Extensions could include logistics management, i.e. making sure that staff have the right amount of medicine and other equipment, and that administrative staff have the ability to maintain the hospital and keep it clean.

Goals and challenges could involve ensuring that all patients are seen within a certain amount of time, handling major catastrophes where large numbers of patients come in in a very short amount of time, and handling strike action by staff. Further development could come from making each patient require multiple stages of treatment in different areas and/or by different staff members.

Reason for elimination:

This game would have relied heavily on overlapping elements that would have been extremely challenging to make work together as desired. We also felt that the interface that would be needed would be very complicated and overwhelming to a player causing confusion and frustration. In addition, AI (Artificial Intelligence) would have had to be created to make the actors move and interact properly. This would have been a huge challenge as we would need separate AI for doctors, ambulance drivers, patients and other actors.

## Strategy

Tower Defence

Place towers of varying types around a map with the one aim to destroy a wave of enemies which spawn in at a known point, this generally follows 2 paths, where you make the maze yourself, or the maze is pre-built and you have to make towers in the available room - both forms require upgrading of the towers you build to defeat increasingly difficult enemies and has limited lives which decrease as you let enemies pass through and get to their checkpoint. Game type has seen a lot of use and is fairly successful, has low scope for creativity and high scope for additional features, however games can last a very long time and is likely to be difficult to differentiate from other games of similar make.

Reason for elimination:

We chose not to pursue the idea of a tower defence game, as we felt that the limited scope of the game hindered our ability to be creative and display our game design abilities. Additionally, we believed that there wasn't a significant amount of new content we could include after the first level, which could result in the player getting bored. When we compared the different ideas we had each brought to the table, we decided as a group that although we felt we could produce a good tower defence game, this was not the idea to pursue with regards to showing our abilities both in group work and in software design.

## Platformer

Moodle Run

The ISS servers have gone down on results day and you’re the only duck that can bring them back up! Make your way from Carter Pond past all the familiar campus landmarks in order to reach the ISS building in time to bring Moodle back online for the thousands of students anxious to get their exam results.

Involving multiple stages featuring different colleges, departments, and campus landmarks platform your way to the goal avoiding obstacles such as slow moving students, tea trolleys and building work to make sure that the students can get their results.

A simple platformer it combines the regular themes with a number of Lancaster University ideas. Common ideas such as platforms, a countdown timer, enemies and power-ups will also be present in order to make the game appealing to the player with the use of a time based high score table giving encouraging replay-ability.

Reason for elimination:

Whilst the concept of Moodle Run would have worked well as a 2D game, we felt it was too repetitive, didn’t have enough variety and had been conceptually explored to its fullest. Moreover, the audience for the game would be very small as it would have little meaning to people who aren’t students at the university. Due to the bespoke theme of the game it might also have required a greater emphasis on design and graphics than actual programming, which whilst important, isn’t the main objective of the project.

# Game Design (40%)

## Specification

### Selected Game

A simple golf mini-putt game with short, separated courses that increase in difficulty as the player progresses. The courses will challenge the player by imposing a target number of shots, obstacles that require precise timing and shot accuracy. The player will be able to view a scoreboard that records the best score for each course, the player will be able to replay each course to try and improve on their score.

We chose this as our game design idea after careful deliberation between all group members, as we believed this to be the idea that would allow us to utilise our abilities in both project management and game design. The ever-increasing complexity of levels and the whimsical nature of mini-putt would allow for creativity and expression in design. The scope of the game is also wide enough for our group to put our own spin on the game idea, rather than creating an almost identical copy of an already existing game.

Mini-putt was unanimously chosen as our game idea, after a vote was taken from all group members. After drawing up sketches and writing descriptions of multiple games, it quickly became clear that mini-putt was the idea that we were most passionate about. Everybody contributed ideas and thoughts about how the game could be designed and expanded, which suggests that this was the idea that we all felt most comfortable pursuing for both the planning and the implementing stages.

Game Concept:

The player must guide a ball to the hole at the end of the course in as few strokes as possible. New challenges will be introduced to the player as they continue through the game to keep it fresh and interesting.

To keep the game interactive and lessen how quickly it becomes stale, there will be multiple ways in which the player can traverse the course to get to the hole, which alone offers greater gameplay depth by allowing the player to make more interesting strategies. Furthermore, power ups will be placed on the course which can be picked up by the player which will alleviate a disability (such as sand slowing movement), adding even greater depth, as the player can choose to try and pick these up in an attempt to achieve a better score.

The game is kept competitive by having a high score for each course that the player can choose to replay and attempt to beat. Friends can compete to beat these scores, making it a more social game that can be played as a group.

In order for the game to play fluidly the ball will react realistically with its surroundings, audio feedback will occur when the ball is initially hit, visual appearance of the power ups on the course (and a continuing visual that remains after it has been picked up), obstacles will be clearly identifiable and have clear boundaries so the player knows how to traverse it just by looking at it. Game will play by the familiar mouse input to set direction of travel, and strength of shot - there will be clear visuals of the object hitting the ball, and the power behind the shot.

An additional feature which players can use is to make their own course, which they can then play - they have the choice to make the course, set the gradient of parts, place obstacles, place powerups, and place location of the hole. This also allows the player to create a course for a friend, so multiple people can get involved when playing this game, making it a group activity and fun for all involved rather than just the main player.

### Gameplay Features

* Record best score for individual course
* Player will use mouse input to set angle and strength of the shot
* Dedicated button to reset ball position and shot counter
* Upon level completion player will be able to re-play or continue to the next level
* Moving obstacles that temporarily block the ball from passing
* The ball will bounce off the sides of the course realistically
* Power-ups that make the course easier *(Power-ups negate sand pit slow down, etc. but player will have to go purposefully go after them)*
* Audio feedback for when the ball is hit or the course is completed
* Level creator for player to create own courses
* Option to view a brief overview of the rules
* Multiple paths to the same hole with one being short but more challenging and another being longer and easier
* Visual indication of what power-ups the player has collected ***(HUD icon)***

Game Design Principles

How Game Design is Influenced

Mini-golf is a simple game to understand, even to those who have never played before. All the player must understand is that they must get the ball into a hole at the other end of the course in as few strokes as possible.

To keep the player interested and challenged, we have decided to add obstacles that make the player manage the power and direction of each shot more closely and also keeping the game fresh and minimizing repetitiveness. The new obstacles include sand traps, which reduces the speed of the ball, gradients that increase the speed of the ball and ice areas that slow the ball much less than normal.

These new obstacles are added gradually as the player completes more levels so there is a smooth curve of difficulty while also allowing the player to learn how to deal with each obstacle. Adding new obstacles will keep the game challenging and spreading them between levels will prevent the player from being too frustrated.

### Pursuing and Achieving Goals

Short Term Goals

* Reach the hole in as few strokes as possible
* Collect power-ups

Medium Term Goals

* Beat the par of a course
* Complete all unlocked courses
* Unlock new courses

Long Term Goals

* Achieve improved scores
* Complete all levels with par score or better

Goal Feedback

* Game will track the current score and best score for each level
* At the end of a course the player can compare their score to the highscore
* Award stars based on how well the player performed
* No more locked levels shown

1. **Game Interactivity**

The UI (User Interface) for navigating through pre-game screens is simple and comprises of named buttons that are clear to the player. When on a course the UI is very simple with only the current number of strokes and the cursor being shown. A graphic is shown when the player is interacting with the ball to shown the power and direction of the stroke.

When in the level editor, the main two UI elements are a list of the usable tiles and a visual array into which the tiles can be placed. There are several smaller buttons the player can use such as a button to save and one to exit.

1. **Feedback**

* The location of the ball to show current progress on a course
* The current number of strokes
* The par score of the current level
* The power and direction of a shot when clicking
* Sound effects such as performing a stroke, potting or environmental effects
* Power-up icons to show what has been collected

1. **Gameplay Variety**

As mini-golf can feel repetitive after a short time, we decided to create obstacles that will be introduced to the player as they progress through the available levels. Easier obstacles are used in earlier levels that introduce a minor challenge, but later level obstacles will require the player to carefully time and position strokes.

Like obstacles, power-ups expand the variety of gameplay by adding new effects to how the ball reacts to the environment. For example, one power-up could increase the bounce of the ball, while another may cause less slowdown of the ball. These power-ups will cause the player to rethink how to aim a stroke from when playing without any power-ups.

Having multiple routes to the end hole of a course will introduce a risk and reward element of gameplay, as if the player takes the shorter route they will be challenged more with a narrower course or different obstacles. If done well the number of strokes will be lower than if the player took the longer but easier route. This choice will allow the player to change how much they are being challenged on their own time.

1. **Game Consistency and Fairness**

Each course has the same basic tiles meaning that any effect that a specific tile has will be consistent throughout the game. However, certain tiles will be more difficult by design and these tiles will only be used in later levels but will still be consistent even in the later courses.

The in-game physics being a close match to real world physics will help the player estimate their strokes better as the ball will react how the player would expect. This will improve the fairness of the game as the player won’t have to change their understanding of basic physics to perform accurate strokes.

As the game displays the angle and power of any given stroke, the player expects two strokes to be of the same power if the same input is given. Having all inputs to the ball processed fairly is important to preventing the game from becoming frustrating to the player.

Reasons for outright failure are infrequent, with the main one being the ball falling out of the boundaries of the course which will result in the ball being returned to the start of the course. Success, similarly, has one condition for success: getting the ball into the hole. Having only two conditions for success and failure means that the player is very easily able to determine what they did correctly or incorrectly.

1. **Avoid Repetition**

The course selection screen allows the player to skip directly to the latest unlocked course when the player returns to the game. This screen also allows for courses to be replayed at any time should the player wish to improve on their previous score or to practise more before attempting future courses.

The inclusion of power-ups, new obstacles and alternative routes on courses means that a given course can be replayed multiple times and the player has a different challenge with which to contend. This coupled with new courses that can vary greatly in difficulty means that the game is unlikely to feel repetitive after only a short period of gameplay.  
Game Rules

* The ball must start from a predetermined point on each course
* If the ball goes out of boundaries, it is returned to the course with a penalty
* A course is completed only when the ball reaches the hole
* A player can shoot from a ball trap or move the ball to the green at a one stroke cost
* A maximum of ten(?) strokes per course
* The power-ups are optional, but the player must purposefully pick them up (IE don’t put them in the middle of the course).

#### Constitutive

* If the player clicks on the course and drags back, then the power variable will be incremented and enforced upon release
* Direction is implemented by the direction of the mouse post initial click
* The level starts with 1 ball at the designated start point
* Once the player lets go the ball will travel in said direction at the designated speed
* The ball’s movement will be impacted by a range of factors such as changing surfaces or gradients that both modify the speed of the ball
* The ball should disappear once it reaches the hole thus ending the level
* The ball must bounce off barriers and obstacles, if appropriate, and will go out of boundaries if hit into a barrier with too much power
* The ball must continue to build momentum whilst traveling through pipes
* The ball must follow physics laws so the ball moves as would be expected in the real world
* A par will be set for each hole as a target for the player
* Each stroke increments the stroke counter by one for that course

#### Operational

* The player must hit the ball into the hole to complete the course
* If the ball is hit out of boundaries it will be returned to the course with a penalty
* There will be a set par for the level/hole that the player must aim to play to or under (least shots possible)
* Each stroke the player causes the total number of strokes for the level to be increased by one
* Obstacles added to the courses must be contended with to successfully complete the course

#### Implicit

* Make every movement worthwhile
* Avoid objects that obstruct the route
* Play within the games rules
* Don’t attempt to break the game
* Stay within the defined boundaries

# 

# Software Engineering (15%)

## Requirements

### The following section will serve to outline the features of the game, how it will be played and the output the user is presented with from a programming perspective. It also serves as a criteria that we must meet in order to make a complete game.

### Functional Requirements

Input

* Primary input shall be a computer mouse
* The input shall be simple and intuitive
* The input shall be taught to the player within the game
* The game will be a fixed resolution displayed in a desktop windows
* The game will have a screen to display the best scores for each course
* The game shall be able to be exited at any time
* The player shall be able to choose the direction and speed of the ball

Output

* The game shall have a menu screen which allows the player to select a course, or courses
* The game shall feature music and sounds; however, they will not be required in order to play
* The game shall make appropriate sounds when the ball interacts with its environment
* The game shall feature credits listing those who worked on it
* The game shall display the current shot total

Game Design

* Level must be completed within a certain amount of strokes
* The game shall be coded in Java
* The game shall be playable across multiple platforms (PC, Mac, Linux)
* The game shall feature multiple courses
* The game shall track number of shots per course and overall separately
* The game shall take number of strokes player makes and time taken to complete course and implements this into a score board
* The game shall randomly generate scores for the player to beat.

### Non-Functional Requirements

* The game should be fast to respond to player input
* The game should run at least at 60fps to give the player a smooth experience
* The ability for the player to see the full course
* Each level will be stored in a 2D array

## 

## 

## Acceptance Tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **When** | **Action** | **Result** | **Pass** |
| AT\_01 | On the main menu | Player clicks on the ‘start’ button | Presented with a screen to select a course | Yes |
| AT\_02 | On main menu | Player clicks “Player Levels” button | Presented with a list of player made courses | No |
| AT\_03 | On the main menu | Player clicks the ‘high scores’ button | Presented with the high score of each course | Yes |
| AT\_04 | On the main menu | Player clicks the ‘exit’ button | The game window closes | Yes |
| AT\_05 | On the main menu | Player clicks the ‘credits’ button | A list of people's names scrolls across the screen | Yes |
| AT\_06 | On the main menu | Player clicks the ‘level editor’ button | Presents an interface for the player to create new courses | Yes |
| AT\_07 | On the main menu | Player clicks the mute sounds option | Game sounds stop being played | Yes |
| AT\_08 | On course selection | The player clicks on an available course | The course is loaded and presented to the player | Yes |
| AT\_09 | On course selection | The player clicks a course that is locked | The game would notify the player that the level is still locked | Yes |
| AT\_10 | On a course | Player presses the ‘Esc’ key on the keyboard | A pause menu is displayed with options to resume or exit the level | Yes |
| AT\_11 | On a course | Drag the mouse backwards from the golf ball | The player is shown a graphical representation of direction and power of the shot |  |
| AT\_12 | On a course | The player releases a mouse click after dragging away from the ball | The ball will move in the direction defined by the location of the cursor | Yes |
| AT\_13 | On a course | Player clicks the reset button | The ball is returned to the tee off area and shot count is reset | Yes |
| AT\_14 | On a course | The ball goes out of boundaries of the course | The ball position and score are both reset |  |
| AT\_15 | On a course | The ball reaches the hole at the end of a course | The player is presented with a screen showing their score and the option to continue to the next level | Yes |
| AT\_16 | On level editor | Player selects an available tile and a slot on the level array | A tile of the selected type will appear where the player selected |  |
| AT\_17 | On level editor | Player clicks on an invalid location to place a tile | The tile is prevented from being placed without affecting the player |  |
| AT\_18 | On level editor | Player places two tiles that aren’t open next to each other | The player is notified that the ball cannot pass between the two tiles |  |
| AT\_19 | On level editor | Player clicks the save button | A screen to name the level is shown |  |
| AT\_20 | On level naming screen | Player enters name and clicks save | The designed level is saved to a file using the entered name |  |
| AT\_21 | On course | Player completes level, score is saved to high score list | Score is saved to high score list | Yes |
| AT\_22 | Menu | Player clicks high score option | High scores are displayed | Yes |
| AT\_23 | On Course | Ball collides with object/wall | Object bounces appropriately | Yes |
| AT\_24 | On Level Editor | Player selects a course to load | Course is loaded |  |
| AT\_25 | On Course | Ball collides with power-up | Power-up is applied to ball for specified amount of time |  |

## 

## 

## 

## Usability Tests

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Potential issue** | **Result** | Pass |
| Buttons | Buttons must be clear and readable, this can be achieved with short , basic wording and interaction e.g. button lights up. | Results in efficient, stress-free navigation. | Yes |
| Image Buttons | Images may be used as buttons for the course for example. Images need to stand out as a button e.g. when hovered over feature. | Results in familiarity to buttons seen in previous buttons linking the two for player understanding. | Yes |
| Button Layout | Buttons should be appropriately placed, corresponding buttons should only be available. | Results in uninterrupted game play giving the players a clear playing field, buttons like mute and restart would be available on a level. | Yes |
| Text | Text should be bold and short to get the point across easily. | Results in a faster learning curve in the game e.g. Tutorial | Yes |
| Layout | Navigation design needs to be simple and repetitive so players can easily conceptualize it . | Results in players recognising the design in the game, this allows navigation to be easy and familiar. | Yes |
| Spacing | Buttons and images need to be appropriately spaced so confusion is limited. | Results in easy and fast understanding of layout. | Yes |
| Game interaction | How the player plays the game should have visual demonstration on how to operate the actions available. | Results in quick learning of how to play the game. |  |
| Relative information | The information that is given must be true to its area for example menu can’t be on a course level, snow level cannot be on a sunny level etc. | Results in easy understanding and gets rid of unnecessary confusion. This allowing greater focus and enjoyment on the gameplay. |  |
| Size | Size of the game, whether it is the frame or the course or even the size of the ball. Size has to be to a familiar scale, if the golf ball is the size of a beach ball this is not going to be familiar and may resolve in hesitation. | Results in familiar understanding, if the game did not have this familiarity, players could be hesitant elsewhere. Having the Sizes be in a realistic scale improves understanding. | Yes |

# Implementation Plans (15%)

## Task List

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Task** | **Start** | **End** | **Duration** | **Dependencies** | **Acceptance Tests** |
| **1.0** | **Design** | **16/1/17** | **22/1/17** | **7 days** |  |  |
| 1.1 | Design Main Menu | 16/1/17 | 18/1/17 | 3 days |  |  |
| 1.2 | Design Level Selection screen | 17/1/17 | 19/1/17 | 3 days |  |  |
| 1.3 | Design Level Editor Interface | 18/1/17 | 20/1/17 | 3 days |  |  |
| 1.4 | Design course pause menu | 19/1/17 | 21/1/17 | 3 days |  |  |
| 1.5 | Design on-course player input feedback | 20/1/17 | 22/1/17 | 3 days |  |  |
| **2.0** | **Game level art creation** | **23/1/17** | **27/1/17** | **5 days** |  |  |
| 2.1 | Tile design and creation | 23/1/17 | 26/1/17 | 4 days |  |  |
| 2.2 | Background creation | 26/1/17 | 27/1/17 | 2 days |  |  |
| **3.0** | **Level Design** | **17/1/17** | **22/1/17** | **6 days** |  |  |
| 3.1 | Course creation | 17/1/17 | 22/1/17 | 6 days |  |  |
| **4.0** | **Gameplay Coding** | **19/1/17** | **10/2/17** | **22 days** |  |  |
| 4.1 | Player Input | 19/1/17 | 23/1/17 | 5 days |  | AT\_10, AT\_11, AT\_12, AT\_13 |
| 4.2 | Ball Movement | 19/1/17 | 24/1/17 | 6 days | 4.1 | AT\_12, AT\_14 |
| 4.3 | Edge Detection for tiles | 23/1/17 | 27/1/17 | 5 days | 1.0, 4.1, 4.2 | AT\_23 |
| 4.4 | Course completion criteria | 24/1/17 | 29/1/17 | 6 days | 1.0, 4.1, 4.2, 4.3 | AT\_15 |
| 4.5 | Interactions with course and obstacles | 25/1/17 | 31/1/17 | 7 days | 1.0, 4.1, 4.2, 4.3 | AT\_23 |
| 4.6 | Ball power-ups | 26/1/17 | 1/2/17 | 7 days | 1.0, 4.1, 4.2, 4.3, 4.5 | AT\_25 |
| 4.7 | Implementation of menus | 30/1/17 | 4/2/17 | 6 days | 1.0, 2.0, 3.0 | AT\_01, AT\_02, AT\_03, AT\_04, AT\_05, AT\_06, AT\_07 |
| 4.8 | Course selection menu | 1/2/17 | 6/2/17 | 6 days | 1.0, 2.0, 3.0 | AT\_08, AT\_09 |
| 4.9 | High score saving | 8/2/17 | 10/2/17 | 3 days | 1.0, 4.1, 4.2, 4.3, 4.4, 4.5 | AT\_21, AT\_22 |
| **5.0** | **Level Editor coding** | **13/2/17** | **10/3/17** | **25 days** |  |  |
| 5.1 | Interface | 13/2/17 | 22/2/17 | 10 days | 1.0, 2.0, 3.0, 4.1, 4.7 | AT\_16 |
| 5.1.1 | Tile selection | 13/2/17 | 18/2/17 | 6 days | 1.0, 2.0, 3.0, 4.1, 4.7 | AT\_16 |
| 5.1.2 | Tile placement | 17/2/17 | 22/2/17 | 6 days | 1.0, 2.0, 3.0, 4.1, 4.6 | AT\_17, AT\_16 |
| 5.2 | Course validation | 27/2/17 | 3/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0 | AT\_17, AT\_16 |
| 5.3 | Saving player courses | 6/3/17 | 10/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0 | AT\_19, AT\_20 |
| 5.4 | Loading past courses | 6/3/17 | 10/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0 | AT\_24 |
| **6.0** | **Testing** | **13/3/17** | **17/3/17** | **5 days** |  |  |
| 6.1 | Black box testing | 13/3/17 | 17/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0, 5.0 |  |
| 6.2 | White box testing | 13/3/17 | 17/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0, 5.0 |  |
| 6.3 | Player testing | 13/3/17 | 17/3/17 | 5 days | 1.0, 2.0, 3.0, 4.0, 5.0 |  |
| **7.0** | **Presentation** | **20/3/17** | **22/3/17** | **3 days** |  |  |
| 7.1 | Write presentation | 20/3/17 | 22/3/17 | 3 days | 1.0, 2.0, 3.0, 4.0, 5.0, 6.0 |  |

## 

## 

## Implementation Milestones and Deliverables

**Design Deliverables**

This covers the design of the screens and menus the player sees when they have started the game. These screens should be easily understandable and reflect an industry standard to allow the player to quickly grasp what each button and piece of information is there to represent.

Deliverables:

* Main menu screens
* Level select screens
* Level edit interface
* Game pause menu

**Game Level Art Creation Deliverables**

This section represents the artistic portion of the in-game interface. The measurable deliverables are the non-menu interfaces provided to the player in-game.

Deliverables:

* Designed tiles, usable as course components
* Background designs

**Level Design Deliverables**

This section covers the course layout design and creation. Courses must be created carefully to ensure they are can be completed and fun for the player.

Deliverables:

* Completable courses

**Gameplay Coding Deliverables**

This section covers the coding of the gameplay. Successful completion of each deliverable will provide the player with more functionality within the game.

Deliverables:

* Player input in game
* Ball movement
* Border collision detection
* Obstacle collision
* Power-ups and relevant effects
* Menus implemented
* Score is recorded

**Level Editor Coding Deliverables**

This section covers the portion of the game which allows the player to create their own courses from a selection of pre-existing tiles provided to the player.

Deliverables:

* Tiles can be selected and placed
* Course can be validated as completable
* New courses can be saved
* Courses can be loaded
* Created courses can be played

**Testing Deliverables**

This section covers the testing portion of the games design. The game must be tested and improved until all testing is successful.

Deliverables:

* Completed black box testing table
* Completed white box testing table
* Completed player testing and feedback

**Presentation Deliverables**

This section covers the assessed presentation portion of the assignment. After the completion of the game, each group member must provide a self-reflective report about their experiences and the contribution they made to the group.

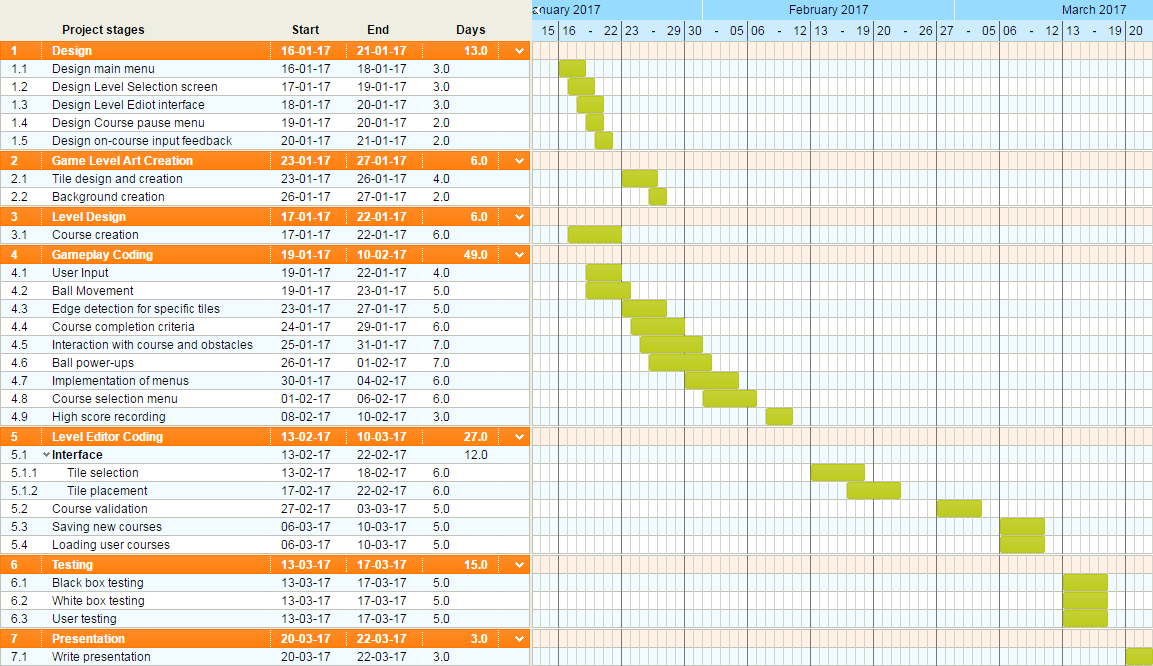
Deliverable:

* One self-reflective report per member of the group

## Activity Network with Critical Path



## Gantt Chart



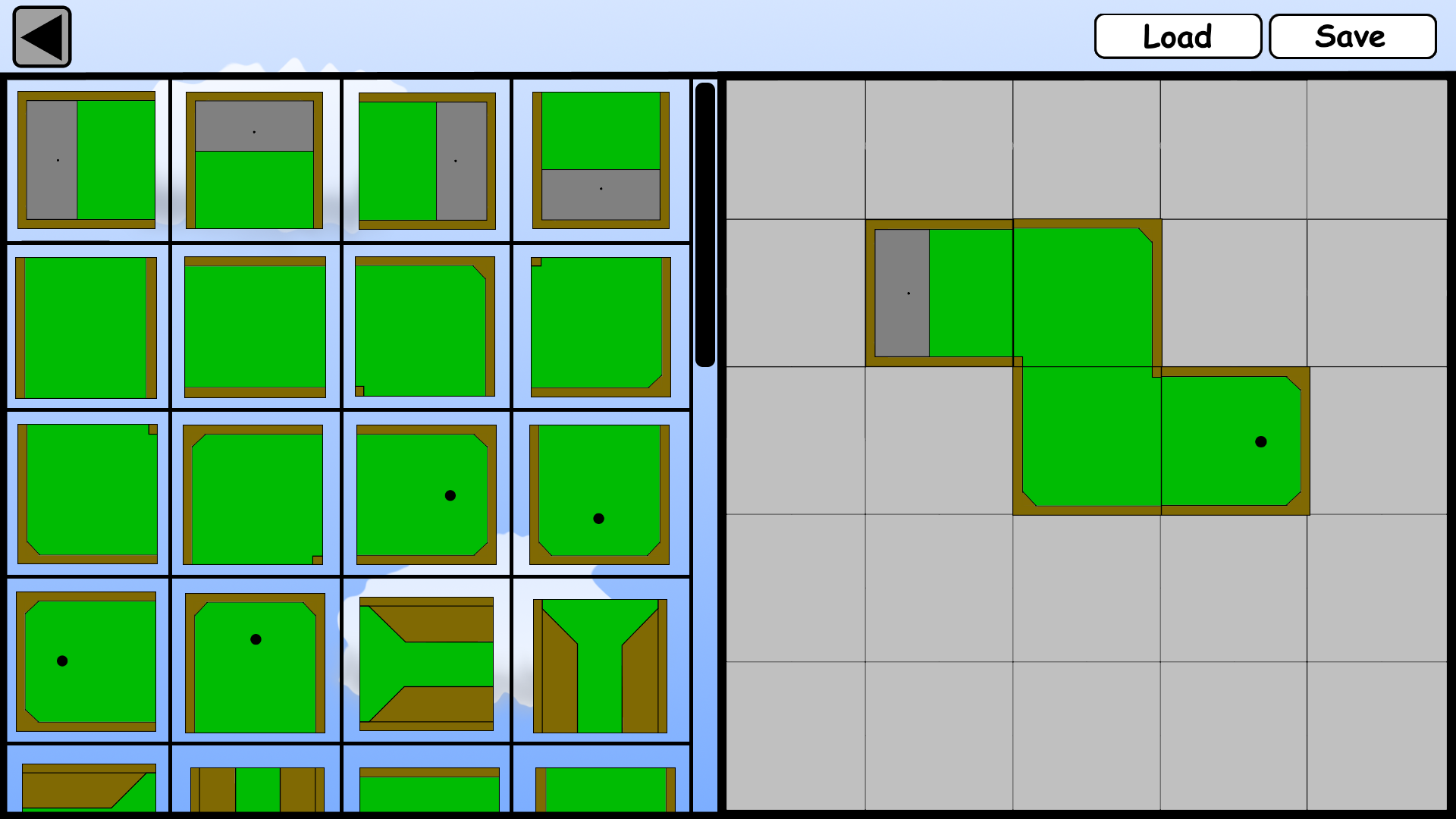
## Contingency Plan and Risk Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Risk | Description | Likeliness | Impact | Contingency Plan |
| 1 | Group member cannot attend meeting | If a group member has something come up such as a doctor's appointment that causes them to miss a meeting. | Fair | Low | The member should let the group know prior to the meeting allowing the other members to divide the workload between them or allocate that member work that they can complete another time, independently of the group before the next meeting. |
| 2 | Group member has extended period of illness/leave | If a group member was to fall seriously ill, drop out or have to miss several meetings in a row. | Low | High | The member should let the group know of their situation and a rough idea of how long they will be absent. The group must then see what upcoming work could be completed by that member and if not then divide the extra work load between the other members. |
| 3 | Work becomes corrupted | If our documents did not upload to Google Drive. | Low | Low to high | If the work was saved locally prior to being uploaded to Google Drive then the documents can be easily reconstructed, however if not then the group will have to redo their specified tasks. If certain members have their work saved locally then they should help the other members with their work. |
| 4 | Internet goes down and we cannot access Google Drive | If the university’s internet goes down we would lose access to the group work which will be stored on Google Drive. | Low | Low | The work may have been saved locally so it can still be edited, albeit with more difficulty sharing and editing it. If it has not been saved locally and the internet has gone down prior to our meeting, we could work from home. Moreover we could use mobile data to create hotspots to access the internet from. |
| 5 | Underestimate the required time for tasks | When allocating tasks, the group has underestimated the complexity/quantity of work due for completion. | Fair | High | Depending on the type of work it could be distributed among the other members granted they also aren’t behind on work. If the future work is dependent on the current task and there isn’t a deadline nearby then the completion date could be extended slightly. |
| 6 | Member(s) fall behind with other work | Group member(s) could fall behind on work due to external workload from other modules or a lack of effort etc. | Fair | Fair | If the work is crucial then it could be a problem, however if not the group can identify why the work isn’t being completed and address it. If the member/s aren’t/isn’t working hard enough then they need to increase their effort or risk a poor reflection in the end of project assessment. If they have too much other work or are just struggling with the task then other members can help out. |

# Final Design

# Our final design has improved significantly since our original concept, as it now has numerous features that will make the game a much more enjoyable experience. For example, we have designed an intuitive and easy to use input system that makes it easy for the player to enter their desired direction and power for the shot. Throughout the game the interfaces we use are simple yet convey to the player what everything does without making them feel overwhelmed.

# In addition to the user interface we have added new concepts to the game, some of which aren’t normally associated with mini golf games, such as power-ups and alternative routes within the same course. These features of the game we feel will not only add more challenge to our game but also replayability. Many golf games will give you the same experience no matter what level you play, our game on the other hand enables the player to play in a whole different way with the aforementioned features changing the experience every time a level is played. We feel like the extra gameplay elements in conjunction to the number of levels will engage the player for a long time without feeling repetitive or boring.level select.pngMain menu.pngtemp ball input.pngtemp ball input angle.png

The largest feature we have added is the Level Editor which we decided to create for two main reasons. The first was to make it easier for ourselves to create the levels that will be included with the game, and secondly to add an element to our game that is not usually seen in other golf games which should increase the interest in the game. An extra feature of this is that players will be able to save and share their own levels with others.

Overall the game offers a both a short term challenge and long term challenge which will both hold the attention and interest of the player. Should the player exhaust the built in levels there is near endless potential with the Level Editor allowing for the player to create however they wish.

(See Appendix B for larger images) /\* <http://www.technologystudent.com/designpro/eval1.htm> \*/