

## **COSC 345 Year Long Project.**

**Liam Chapman, Calum Hunter, John Kim.**

### **What we are going to build:**

For our year long project we will be building an offline game based on pocket billiards. The specific type of game we decided to base our program on is called eight-ball pool. In this popular version of the sports there are sixteen balls; a cue ball, a black eight ball, seven striped balls, and seven solid balls. A player aims to sink their respective balls (either solid or striped) into any of the six pockets that are along the rail by striking the cue ball with the cue.

For our program, we will be creating three playable modes: Local player versus player, Player versus AI, and a unique Trickshots mode. 'Local player versus player' mode will allow two individuals to play a game on a single machine, and 'Player versus AI' mode will allow a single individual to go up against our AI opponent. The 'Trickshot' mode will consist of various different levels where the individual will be given a single attempt to sink a specified ball into a specified pocket. The level difficulties will be determined by various obstacles being placed on the table and also by changing the environment and the balls in play.

Our game will take keyboard inputs from the user to determine the angle and strength of their shot, removing the need for a mouse. We will also be creating our own physics library, which will allow us to optimise the code; removing unnecessary functions while giving us more freedom of use. The AI will also be created by ourselves, but will include a range of randomness, preventing it from always winning with 100% accuracy all the time.

### **Who is going to do what:**

Dividing the workload between everyone in the group, we agreed there will be certain tasks requiring everyone's input, but also allocated each member to lead specific tasks based on our individual strengths. Due to the small nature of our group, we also agreed to aid each other in our allocated tasks to increase efficiency. For the creation of the game screen, cue ball, and trickshot levels, everyone's input will be needed.

As the first task to complete, group work on the game screen will give us an opportunity to explore SFML and learn how to best use the library to our needs. To create the cue ball, we decided everyone should trial our physics library together. Each of us will get a chance to provide input into the implementation of the physics library, and working together will also allow us to correct any bugs and problems as a group. To

create the trickshot levels, all of us will bounce ideas off one another, making sure that each level is winnable, but also challenging enough for the players. Creating the trickshot levels together, will also allow everyone to add their unique ideas into the game. We will also be working together on the AI, as we expect it to produce the most amount of problems.

Liam has been tasked to lead the creation of the physics engine, AI, and the menu screens. He is a third year computer science major that has completed the COSC 360 paper in summer school. We decided he is the best mathematician in the group, with a solid knowledge of vectors, and decided he will lead the creation of our physics library. He also has previous experience making game menus, and working with AI algorithms from his COSC 360 paper.

Calum has been tasked to lead the creation of the game table, game manager and the UI. He is also a third year computer science major and has completed the COSC 360 paper with Liam. His experience in 2D graphics, creating game managers, and UI elements, was our reasoning to have him lead his specific tasks. We believe that it would be most efficient for us to have the game manager and UI elements lead by the same person to maintain consistency.

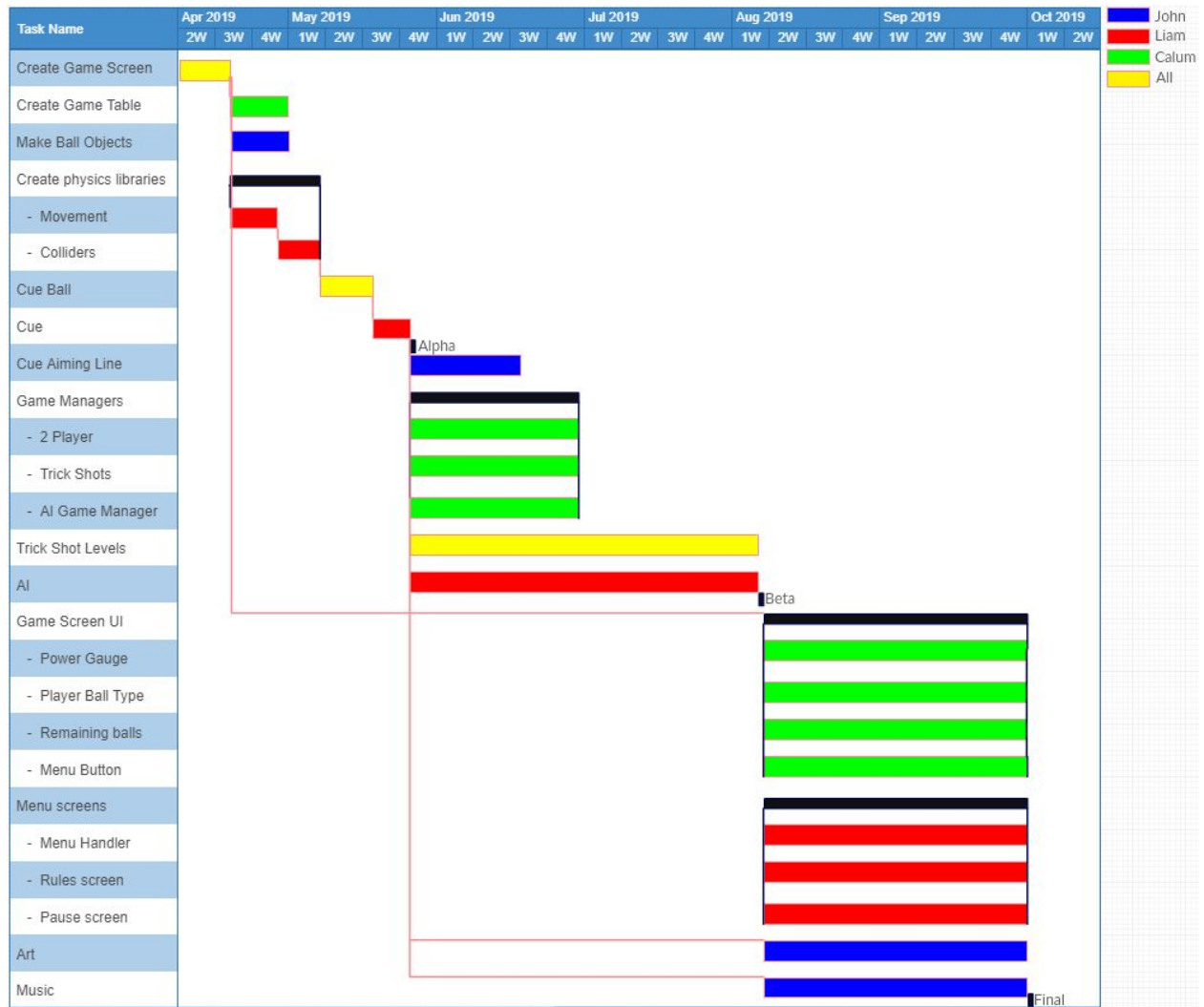
John has been tasked to lead the creation of the pool balls, cue-aiming line, music & art, and documentations. He is currently completing a DipGrad in Computer Science and has a background in Psychology and Philosophy. He has less knowledge on coding and game development, however by allocating him the ball creation and the cue-aiming line, we aim to have him continue to develop his skills to be used later in the project. He has some background knowledge of music & art, and was allocated to lead in documentation to make use of his experience in writing reports.

### **How we will build it:**

We will be using C++ as our main coding language, implementing third party libraries to aid us in coding our program. We will be using Visual Studio Code as our IDE and have decided to use a free, third party library called SFML. SFML will mainly help us implement the GUI, Windows, Audio, and System components while keeping our program compact. Our program will be version controlled using Github as our online repository and Sourcetree as our local repository. To track and document bugs, we decided to go with Bugzilla as it is open source and trusted by various users online. We plan to meet twice a week; once to determine what tasks to complete, and once to discuss our progress and any issues that we encountered. Each member will write a small report on their individual work, which will then be compiled into a single document to keep a record of our overall progress. We have also agreed to make time for extra meetings, should we fall behind schedule due to unforeseen circumstances. For our

project, we do not believe that there will be any hardware or software dependencies. We will also be using Make to build our project and will use Java to create our test harnesses.

## How long it will take to build:



We estimate that the entire project will be completed by the end of September; our alpha release of the program to be made on the fourth week of May, and our beta release to be made on the second week of August. Asides from our deliverables, our main milestones will be the completion of the physics libraries, game managers, game screen UI, and the menu screens. Of these, we will be working on the game screen and the menu screens, in parallel, as they will not be dependant on one another. These times are only estimates and will most likely change throughout the year as we expect

to run into problems along the way. In some cases however, we expect that certain tasks may be completed earlier than the timeframe we have given.

### **Which standard layout we will be using:**

We will be coding our program following the K & R standard layout.

### **What program out there already does what ours will do:**

The concept of eight ball pool is not new, and as such there are multiple versions of the game available to play on various devices. Already there are popular flash variations of the game available to play online and there are other variations available to download & play on mobile devices running iOS and Android. Researching online showed that most variations had 3D like images for the pool balls and were usually playable online against other players, AI opponents, or had local player versus player modes.

### **How our program will differ to the programs already available:**

'Small is beautiful' - for our version of eight ball pool, we are aiming to create a program which is very lite, challenging, but enjoyable to play. Our game will be offline and will allow for a local player versus player game mode, a player versus AI game mode, and a puzzle style trickshot mode as well.

Our key selling points will be that it is a lite program, 1000 lines of code or less, and the inclusion of our trickshot mode. Most variations of pool games we looked at, only allowed for gameplay against opponents. We wanted to differentiate ourselves by including a game mode which does not require an opponent, but still provides a challenge to overcome. Our game mode will also allow players to enjoy a slower paced game which they can complete at their own leisure.

The trickshot mode will include various different levels of difficulty, where a player is given a scenario on the pool table, and is required complete a task. The task involves the player attempting to sink a specific ball into a specific pocket, but they will only be allowed one attempt. If they fail, they will be given the option to try again, or to go back to the main menu. Each successful attempt will advance the player to another level with a similar task, with a different scenario, until all levels have been cleared. We will create each scenario by placing various obstacles on the table, and we also plan to experiment with our physics library to alter the interactions the cue ball would have with the environment.