COSC345 A1

“Escape Owheo”

Who is in the Team?

Oscar Crowe: Hi my name’s Oscar, this is my last semester at university before graduating with a BSc. I'm most comfortable working in java or any other OO language but over the last year I’ve definitely become more comfortable with C. At the moment I'm just trying to get as much experience as possible with new skills whether it be continuous integration or making ASCII art so that I'm better prepared to leave uni with confidence in my skills.

Callum Sullivan: Hi, my name is Callum, I'm in my final semester at Otago University working towards my BSc majoring in Computer Science. Throughout my time in Otago, I've become most proficient with the Java and Python languages and had no experience with coding before arriving here (though I’ve come to love it). I enjoy problem-solving and breaking down complex issues into their simpler components and I hope to bring this skill set to the team. This is my first time working with C++, so far I’m enjoying learning the new syntax and I’m excited to see where the development of our game gets us and hope we have something to be proud of by the end of the semester.

Anthony Deng: Hi, my name is Anthony, and this will be my final semester before graduating with a BSc. I am most competent when working in Java or Python, so working with C++ is a little out of my comfort zone. However, I see this as a challenge that can become beneficial in improving myself as a developer. I think this software development paper will be a good experience, both in learning and creating, and will be helpful with whatever path I may choose after graduating.

Lachlan Graham: Kia ora, my name is Lachlan and I am in my final semester before graduating with a BSc. I mainly work in C or Java, so C++ has been a refreshing change with elements of both languages mashed together. The only other exposure to C++ was in Visual Computing in semester 1 2024, which relied heavily on external libraries and CMake to create a viable program (which is fair enough). I am looking forward to this paper's final program requirement of ‘no external libraries’ to see how flexible and coder friendly we can make the code in case some random person on the internet wants to add a level themselves.

What are we going to make? / How will we make the game?

The overarching genre of our group's program is “escape room puzzle game”. We intend to have an ASCII art depiction of walls in a 4 walled room. Each wall will have some sort of hint, clue, or mini-puzzle that will help you progress into the next room. Each member of the group

will be planning one room for escaping. A positive of this is the classic features of escape room games (zooming into objects, moving from wall to wall, shifting the dialogue/exposition box etc.) can be repeatable functions within the code that we can share and implement in each other's rooms. Another plus with the design of the game is adding an extra room is easily done, allowing for bonus levels if we so choose.

The game will be built in C++, this is due to the object-oriented nature of the language. Using C++ promotes good code maintainability as many aspects of our game can be broken down into objects that are easily reusable for different rooms. Each puzzle room will be built in a way that enforces a sequence of actions or moves that allows the room to be cleared. User input will be taken to allow the player to navigate around the room, zoom in for detail and solve the specific puzzles. We will also develop testing we write ourselves for the display, graphics and input to enforce good standards. The code base will also run through a continuous integration pipeline each time it is pushed to GitHub so that it can be built properly.

In terms of our schedules and goals for the project, our team gantt chart as well as the hard deadlines will be our guides for delivering code on time. Smaller tasks will be delegated between our members based on our different areas of strength or weakness, the integration of these features will be done through different GIT branches that we will keep at maintainable sizes so that we are not merging huge amounts of code every two weeks. Lachlan's device will serve as the main branch because he has the only laptop that can run the program (Windows laptop compared to the other members' Macs), all other group members will check out different branches on lab machines and integrate features through them.

How is the game back to basics?

Although all of us have learnt alot of different skills and tricks over our degree we want to be able to showcase we can build something very simple very well. Effective planning, good test coverage and good communication will allow us to achieve this.

Our game can be considered basic as it revolves around the idea of using ASCII art in the terminal to represent the game. This is because ASCII is only made using printable characters and does not require any special third-party libraries to generate or render. There is no complicated game logic or fancy rendering tricks, we simply want to create an visually appealing and interesting game that is enjoyable to play. Because of the simple nature of the development it will be much easier to extend the game by adding more puzzles/rooms which gives us alot more flexibility down the line. It also means testing of the game will hopefully be straight forward. The only things we need to worry about are the users input, how the display reacts, and how the puzzles are solved.

What is similar? And what Interest is there?

Terminal based dungeon games are not uncommon, and neither is the ASCII graphics scheme they use. NetHack is a good example of a game set in the command line where you descend floors fighting off enemies with an end goal of retrieving the “Amulet of Yendor”. And although we have a different concept of first vs. third person, the way the game is able to create levels with ASCII art can serve as a great reference point for our future work.

Another game that we are drawing inspiration from / will end up being similar to is Terminal Town. This follows more with the puzzle type game we are trying to make. However, this game implements puzzles as folders / files in your directory. And using certain command line tools is the navigation you need to access different resources to take over a fictional town. We think the use of files and the directory structure of your own device could be a unique way to make the puzzles in our game more interactive and interesting.

The way we can gauge customer interest for this sort of game is seeing how similar concepts or formats perform in the market. The above two games are smaller and lesser known but much closer to what we are going to make. A game that isn't exactly in line with ours but still draws similarities is Papers Please. The graphics are very pixelated, you only ever stay in one room and the gameplay is based around solving incrementally harder puzzles at a border crossing. Despite being so simple the game has sold over 5 million copies and has been played by some of the largest youtubers in the world. What this tells us is that the complexity of a game does not determine its popularity, if you can create a unique story and do it in a creative way then you'll be able to draw people in.

How is the game going to be ‘cool’? (and other cool points)

Because of the operating systems our group runs (¾ mac) the only time anyone but Lachlan is able to write code is in the labs. This felt like a detriment at the start but actually has become quite useful. We now make most of our big decisions in person as a group, and can pair program, which for most of us is brand new. Even with Andie the majority of code written was developed and pushed remotely, there was no cooperation or live feedback. We hope this unusual style of working will actually serve us in the long term and create a program everyone is much more familiar with.

Our team has decided to use discord as our communication method. Originally we were using a messenger chat but sending code between each other in that was really impractical. Discord lets us send entire code files and preview them with syntax highlighting which is obviously very helpful. And because not everyone can run the code on their computer its great to be able to make remote suggestions and then send them through so Lachlan can integrate them fully. The discord also has voice chat rooms which we havnet used yet but last semester specifically when some of us were having to smash out group etudes it was much better to be able to call and discuss a problem rather than having to type things out.

Although the game might be simple in nature we want the graphics that accompany it to be visually appealing. Each wall in the puzzle room should be able to be zoomed into to reveal more detail and hidden easter eggs. We hope the level of visual detail and artistic style is one of the main points that sets our game apart from the rest. And because the game is based on the Owheo building we are going to make the users of the game feel like they are actually there, the terror of being stuck in lab G.06 should make the puzzle solving process even more frantic than it usually would. The puzzles are going to vary from visual tricks, to ciphers, to keyboard / mouse movements and hopefully file directory mischief. The game will feel interactive like it has taken over your device and your only choice is to leave.
