### C++ Report

**Specification:**

So, this will be my first C++ project. I want to make this project something to remember because I hope to look back and learn from this project. I've always been into that 1980s video game retro vibe.

What is it?: Well, the game I have created is "The Maze Game". The game will be 24 by 24, with three enemies, one start and one finish. The reason why I choose a maze game is that I wanted to create a game where the player can think about the right move always. I want the player to be connected to the game. There're walls that they can't touch, dead ends and enemies which can kill you.

The plot of the game: It's Earth many years in the future; there's a task force that has to take down the enemy. One of the task force members has been doing puzzle drills to increase his awareness. However, there was a spy and told the enemy that one of their soldiers is going drills. So, they built a maze to make this a hard game to give them move time. You "Mike Razer" and helicopter 4 has just been destroyed; you're the only survivor but you have to face a MAZE. to get to the drop off point.

Data types:

* Boolean
* Data
* Decimal
* Double
* Integer
* Object
* String

Key features I used in the program:

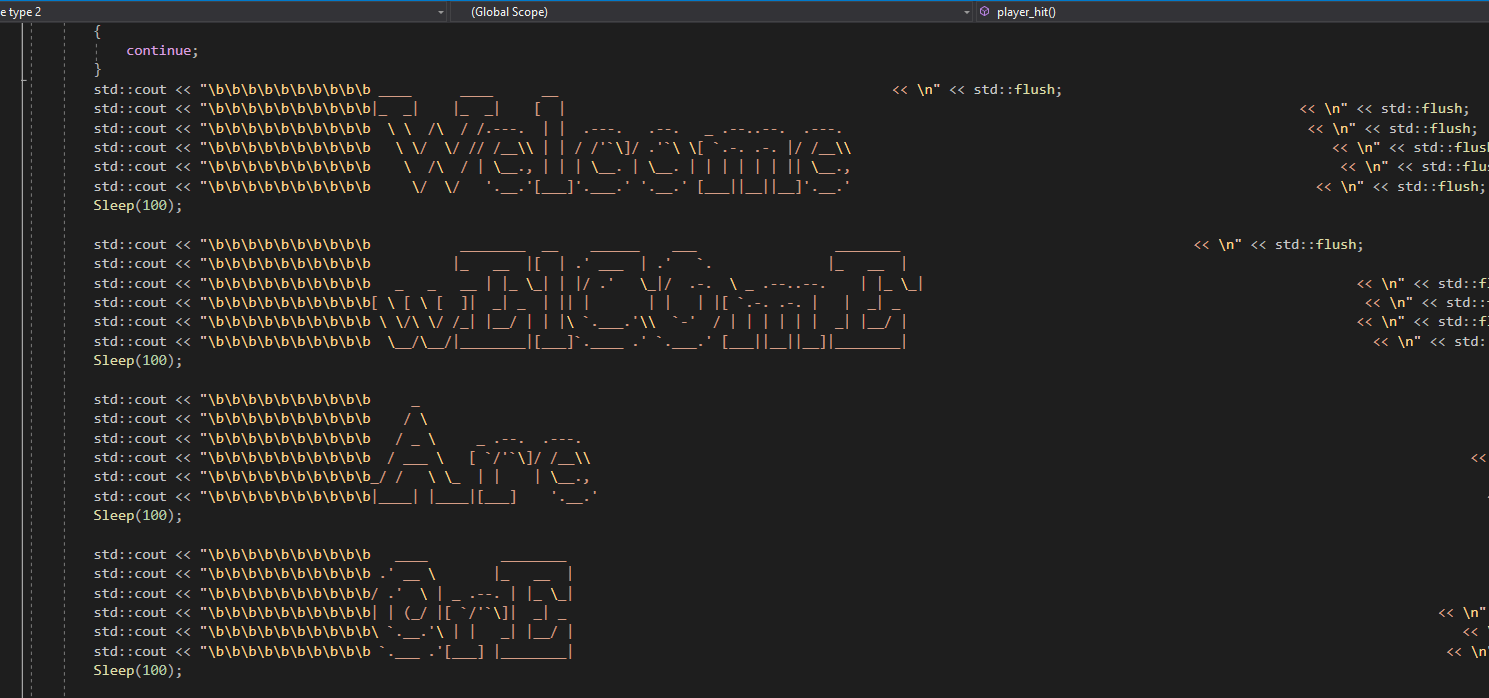
bool CheckWinCondition(int playerX, int playerY)

This code a key feature because when the player gets to the endpoint, the end message will appear and that's because of that code. The code will activate when the player gets to the X and Y point. This is a must needed code because the player will know that they have won or even lost and that's something essential in this game.

cout << "Hello Maze Runner use [w], [a], [s], [d] to move around so that you can escape this maze. press any [w] or [a] or [s] or [d] and then press [enter] to start playing the MAZE GAME\n"; //might chnage beacuse i want to make it more 1980s vibes

cout << "presse (e) then return to exit the game\n";

Movement is what the player needs to do. Embedding the controls for the player is needed. Now gaming on the computer has changed from the simple arrow keys. It's now the universal control of WASD and an interactive button of E or I (i chose E). I think adding this for the player is necessary.



By coding the title screen in this type of style is something that I needed to add because it gives this simulation, futuristic feeling to the game. having two types of the same word in a different style will draw the player in. I know it takes around 30 seconds to load but that's the feeling I wanted. This dark utopia world that's controlled by AI. (I've talked more about the title screen in the design section)

**The Design:**

Title screen:

|  |  |
| --- | --- |
| The Title screen is having this glitch feature. When the player presses start then these words will show up at the speed of 100: sleep (100) and each word will repeat four times because of the command  If (I == 4)  {  Continue;  }  “Welcome are you ready maze runner  Loading...”  (Image 1)  When it has repeated four times then a simple instruction will appear. The controls will show up which are “[w], [a], [s], [d] “and (e) to start the game. I think having this feature is this matrix's, 80s type theme. I wanted to have this feeling because classic type games are always great to play. Having the matrix theme is great because it comes down the screen fast. (Image 2) | Image 1    Image 2 |

End screen and end point:

|  |  |
| --- | --- |
| The end screen is a very important part of the maze game. There are two endings. The first ending that I programmed is when you lose. Following the theme of the 80s game, "game over" will show up; because the game has a loop feature in it, there will be another message for the player to play again. Adding the loop code is super important when you make a game or any sort of program because it's just a suitable code to add.  The second ending is when the player finds the exit and wins the game. There will be an image of a helicopter because it's showing "communication team 5" picking you up. I've also added the loop code even if the player has won the game. |  |

The player:

|  |  |
| --- | --- |
| YOU the player has four keys "WASD". These four keys are the only way to move around the maze. When you start playing to know that you in the maze there will be a letter "M" and this is the player. The "M" means maze runner. The maze runner (The player) has three lives. |  |

The enemy:

|  |  |
| --- | --- |
| The enemy will show you in three positions in the maze and you can see them say the letter "K" because their main mission is to stop you. |  |

The Maze:

|  |  |
| --- | --- |
| The original idea for the maze was 10 by 10 but I did some tests; it was just easy. Having the maze 10 by 10 with no enemies wasn't the correct idea. So, having a bigger maze (24 by 24) and adding enemies would make the whole game so much harder.  The maze is 24 by 24. The walls are displayed as "#" and if the player touches it, then they will lose a life. The paths where the player can move is "\_". On the path will also be three enemies, so watch out. |  |

Lives:

|  |  |
| --- | --- |
| When I first started, I was going to add five lives but I thought the game would have been so easy and the maze at the time was only 10 by 10. So, by making the maze 24 by 24, adding three enemies and dropping the lives by two. I want to make the game, have a great story and quite hard. So, dropping the lives was important. It will make the player more aware; they'll think more because every more is costly.  When the player does hit the wall or one of the three enemies then a message will show up at the bottom of the maze to tell you. |  |

**Test:**

Testing is a very important entity when making a program. By having the first idea not programmed that well, I had to make a game that was better, bigger and complex. By increasing the maze to 24 by 24 I would be able to have a successful test. increasing the maze gave me the freedom to unlock new ways to add more features. I was always working out where the start would be for the player (0,1). The finish was something that was the hardest. I had to keep test the end coordinates so that when the player gets to the end the end message will come up. I was getting it wrong for a few days but testing the outcome worked.

Another feature that I had to test was the damage done by the enemies. One of the enemies was doing the full three damage and that would have made the game super hard for the player. The code

void player\_hit()

{

player.posX = 1;

player.posY = 1;

player.hp--;

}

This code helps out the whole game because it will determine that one enemy will only do one damage. But the player still has to look out for the other two because the player can still hit both and lose.

The title screen/ start screen was a compact detail because I had to see when I start up the game that all the text would be in line. It took a few hours to test but the title screen and even the end screen both works; they all fit in place. I think that these title screen/end screen took a few hours because I had to have the real word then adding a diverse style. Having these different styles of text is something that gives character to the game. I want the player to be eager to play. So, testing out these styles is a must needed for this game.

The most crucial part of the game is the player. I had to test the controls and movement of the player. I had to see if the maze didn't move. Well, the prototype went very wrong. So, after a lot of research and testing the board stayed where it is; the player was able to move with ease. Movement is important when making a game because moving is a course of action, a way to tell a story, a path to success. This is by far one of the most important countenances.

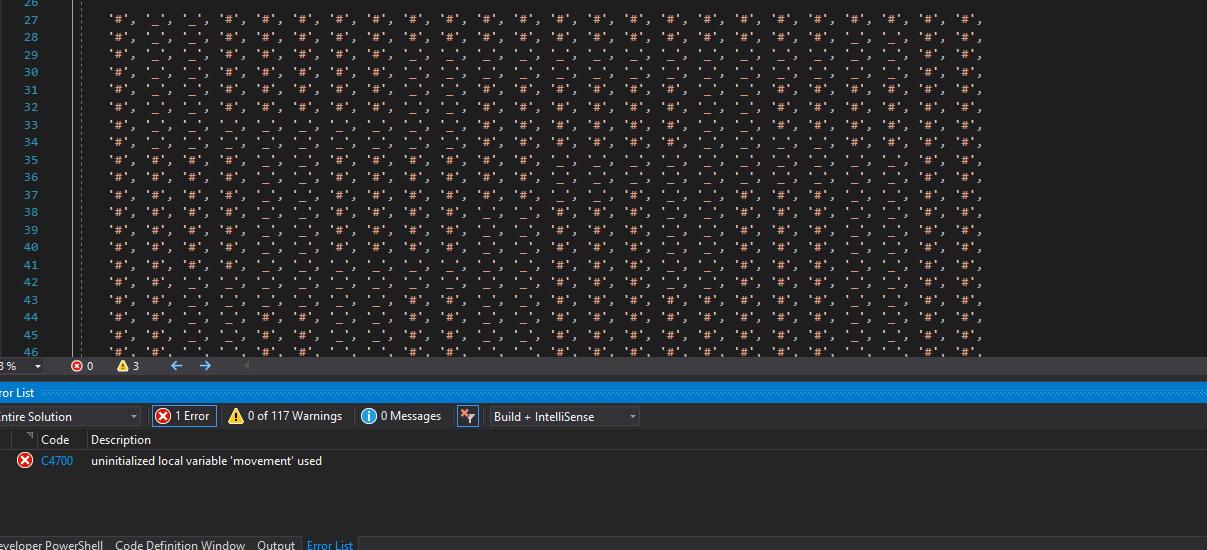
The messages in the game are something for the player to see. The message is to tell the player if they hit a wall or one of the three enemies. This was something that was tested consuming but it's needed. The code

else

cout << " Maze Runner this is communication team 5, you just hit the wall, Over..."; //this is when the player is hit

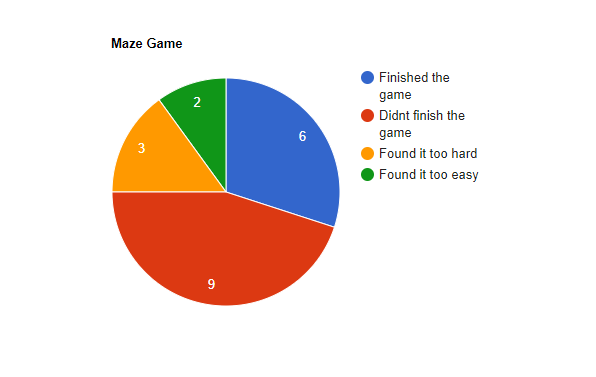
break;

This would show up when the player lost damage.

This image above is something I saw when testing. I start up the game and error C4700 comes up but the game shows up and works fine. This must be a software error because everything is working fine.

So, I decided to let 20 people play my game. I needed to have this data so that I have correct and realistic feedback. the percentage is higher for not completing the game and finding the game hard. Is this a negative towards my game?

Well, I don't think so because if I made a game that is too easy then I think that the player wouldn't find the game that fun. But I do understand making the game too hard can be rather annoying for the player. I need the player to be drawn into the game and I know that the game is challenging but the story behind the game will urge the player on so that they can finish and leave the maze.



**Evaluation:**

What worked: So, what worked in this program. I think the way how I got the look of the theme went the way I wanted it to go. The 1980s feel is a classic addition to the whole game. The title screen, end credits and the maze are what I had in mind. The way the whole program/game flows as a whole were super important for me. By adding a story behind it is something I wanted for the player because I want the player to get close to the maze runner (you). The maze is something that was very hard to make but the outcome is what I wanted. Some paths take the player the wrong way, paths where it's too narrow; they could lose a life and three enemies. these three enemies are a great addition to the game because I thought that the game would be so easy if it didn't have three enemies and also, I added a feature on the wall which also take damage. Having two things that can take damage makes the whole game super fun.

What didn’t work: So, I can say that three things that are not working. The first would have to be that it's really hard to beat the game. I've to try my hardest to beat it but I'm always dying. But I can say that the game over message show ups. So, I know that when the player gets to the finish point the victory message will appear. The second wrong thing is that I start up the game it says there's an error but the game shows up working fine. Maybe it's a software error. The third feature which is not working is when the player moves and there a small lag. This does make the game a lot harder. This could be a great outcome because it does make the game a lot harder for the player but it's something that could annoy the player. I'm not saying it's a huge problem but as the programmer it's something.

What could have been improved: I think there are many things I can improve about this program/ game and that is the maze. If I would make this game again, I would increase the maze from 24 by 24 to 100 by 100. The reason why is because I would have liked two finish points, more paths, more enemies. The enemies would have moved liked in Pacman, they would come to the player making it hard, making the maze walls move to increase the level of difficulty. It might be small changes but for me this would have made the game better, harder and more creative. I'm happy with what I coded because it's a start maybe this idea can come real. I think the title screen and the end credits could have flowed better but I do like how it was made. But I would change the speed and the position of the text next time.