

For the testing code (on left), I began with coding the fundamentals of the project, which was checking if the number inputted was divisible by 3 and 5. I completed this by first creating a variable “n” to act as a counter and set its value to 0. I then created another variable "divisible\_3” and set it to false, this was to check if the number was divisible by 3 or not. Then, I created an if statement to check if the number was divisible by 3, and if it was set the variable “divisible\_3” to true. I did the same for checking if the value was divisible by 5. To execute this, 3 more if statements were created, checking if the variable “n” was divisible by 3 and 5, just 3 or just 5. The way this was done was by in a way replicating what was inputted by the user to the variable “n”, so that when the user inputted a string variable, it would not divide it to check if it is divisible by 3 or 5. As the game requires the user to count up from 1, the “n” variable is just added by 1 each time, replicating the input by the user. This also allows to check if the user inputs a wrong number, by checking if the user inputs match the variable “n”.

For phase 2, I began with coding and testing the basic requirements of the project like checking if the number is divisible by 3 and 5, etc.

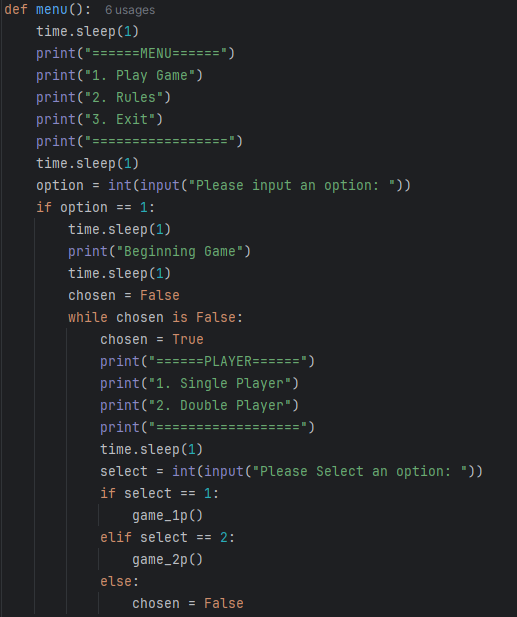
Below is an example of the code being executed

Phase 2: Coding (3Hrs 0mins)

For the Planning Phase, I began with analysing the criteria given for the project to understand what needed to be done. After understanding the criteria / requirements, I begin planning to see if I could break the task into multiple smaller tasks to be completed. I then made a flowchart of the game project (see attached) to make it easier to visualise and plan out my next steps completing the task.

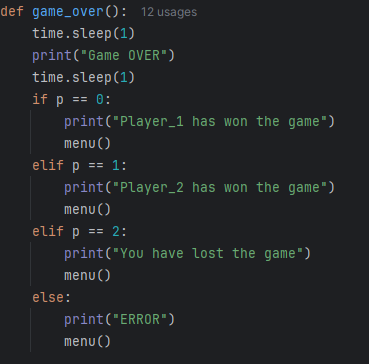
Phase 1: Planning (30 Mins)

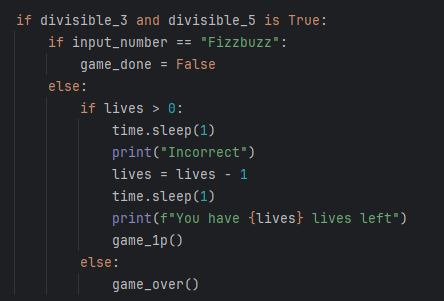
RDT FizzBuzz Console Game Project

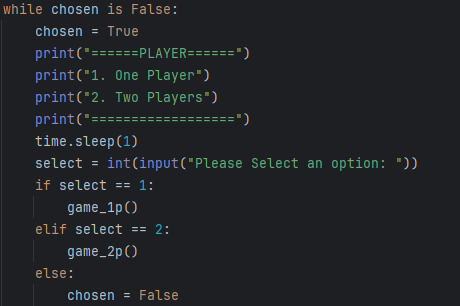


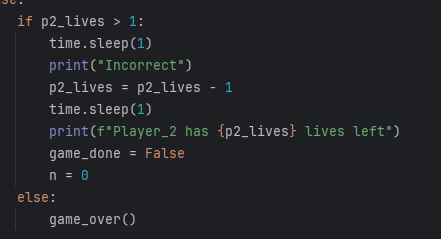
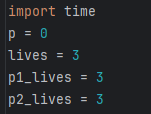
Also for test 2, I improved on the game over section, showing who won at the end of the game.

For test 2, I created a basic menu, including a play game option, rules option and exiting the game. I also added the choice to pick between one player or two players.





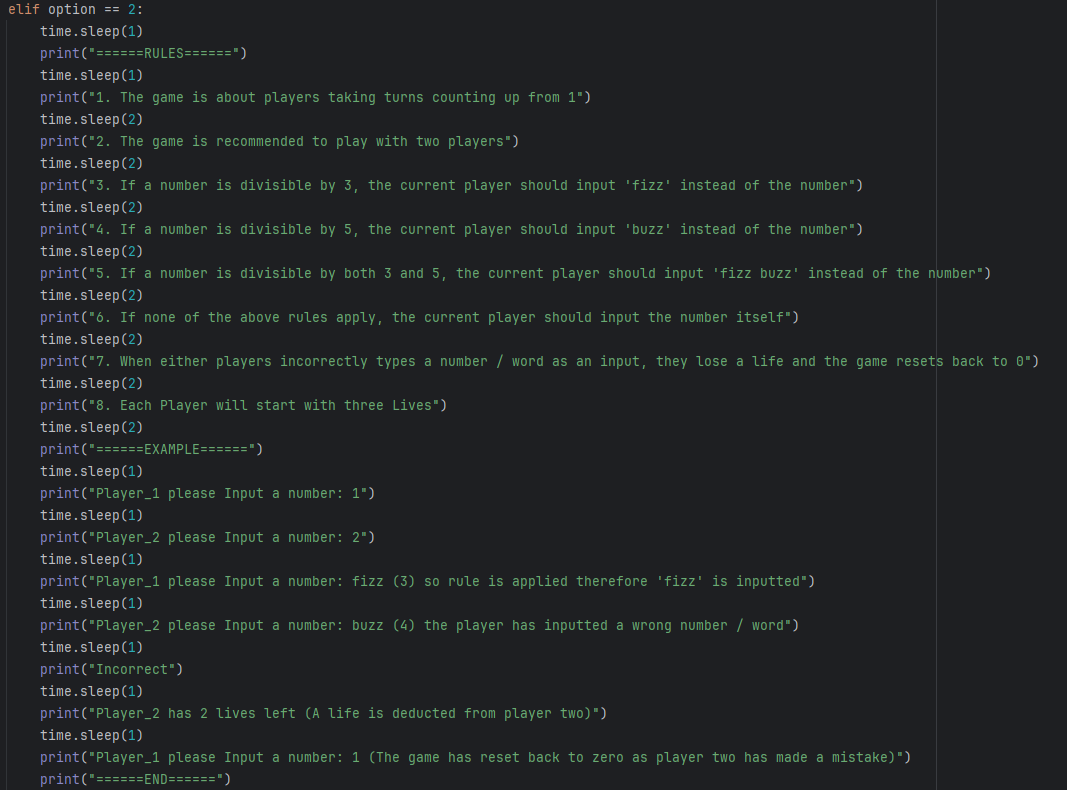


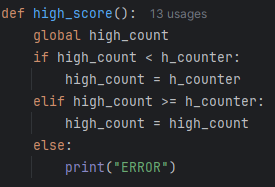
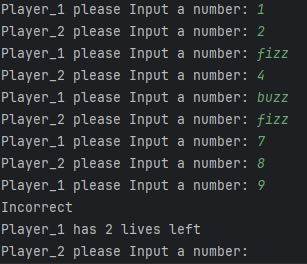
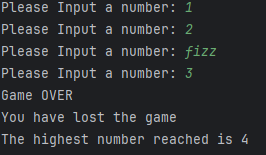


For test 3, I also created a section in the main menu called rules and example to explain the rules to the user, how the lives system worked and a typed up example round, with annotation on how it worked to help the player understand the game. (See next page)

I also created a simple menu for the player choosing option, so the player could decide to play alone or with someone else. (Below are the variables storing both the single player lives, two player lives and player count (p))

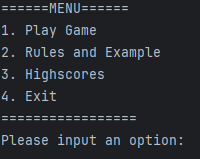
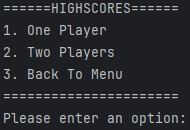
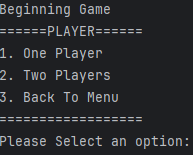
For test 3, I implemented a bonus idea where the player would have lives instead of losing the game instantly, I did this for both the single player and two player options, outputting that the player was wrong and how many lives they had left





I created a high score system to keep track of the highest number reached by the players. It keeps track of both the single player and two player games together, but resets every time the application closes due to not being able to save externally. (See above)

Below is the user interface of the console, how it would look like to the user when executing the code



For the final phase, the testing phase, I and some volunteers tested out the game to both get some feedback and to test the game for any bugs or glitches in the code. I conducted multiple testing, one playthrough was playing the game normally, while the second playthrough was to try and break the game. On my second playthrough, I came across some bugs like the menu not working or the game not registering the high scores but it was easily fixable. Some testing that I made were within the code, I tried to implement some bonus features like a countdown but could not seem to get it to properly work, which was a bummer as I was hoping to make a Hardcore mode, with One life and a time counter but only managed to code the One life mode.

Phase 3: Testing (30 Mins)

Overall, I found this task to be quite enjoyable, getting to code something is always interesting in my opinion. I learnt new ways to implement code that I had never usually used before and learnt how to code some elements of games like a high score etc. Throughout the task, I faced some challenges that helped me learn and understand more about python and how the code is implemented. I am inspired to do more projects in my own time, including learning how to code a timer countdown.

Summary: