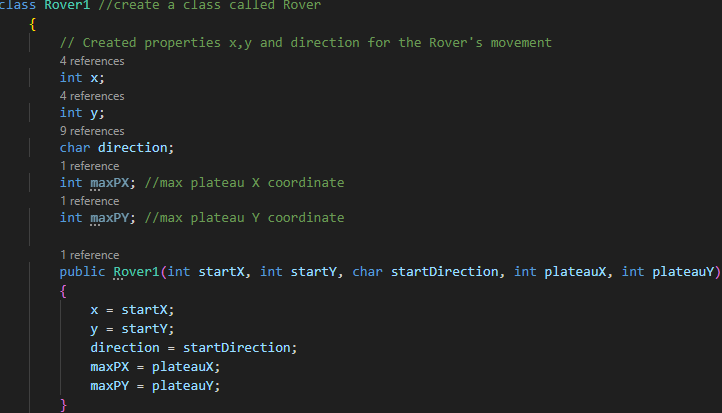


Time Taken: 7 hours

To begin this coding project, I copied all the typical tasks from the PowerPoint and put it within VS Code, this allowed me to easily look back at the code when needed. Besides the typical task, I labelled what was needed (for example the first one, it required the x,y,z coordinates, directions and were commands.)

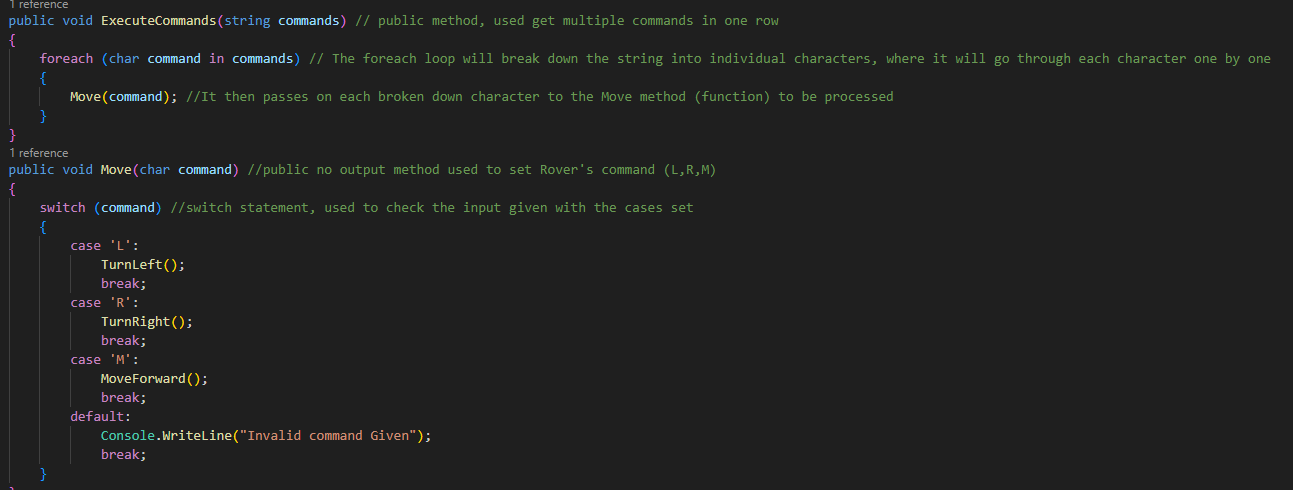
RDT Mars Rover Task:

After listing out the requirements, I took some time to plan and assess what needed to be done. I broke down the task into 3 main components, the Rover (it’s movement, it’s direction’s both initial and final), the plataue (The top right coordinates, the boundaries) and the user input (the x,y,z coordinates, the plateu, and validating user input)

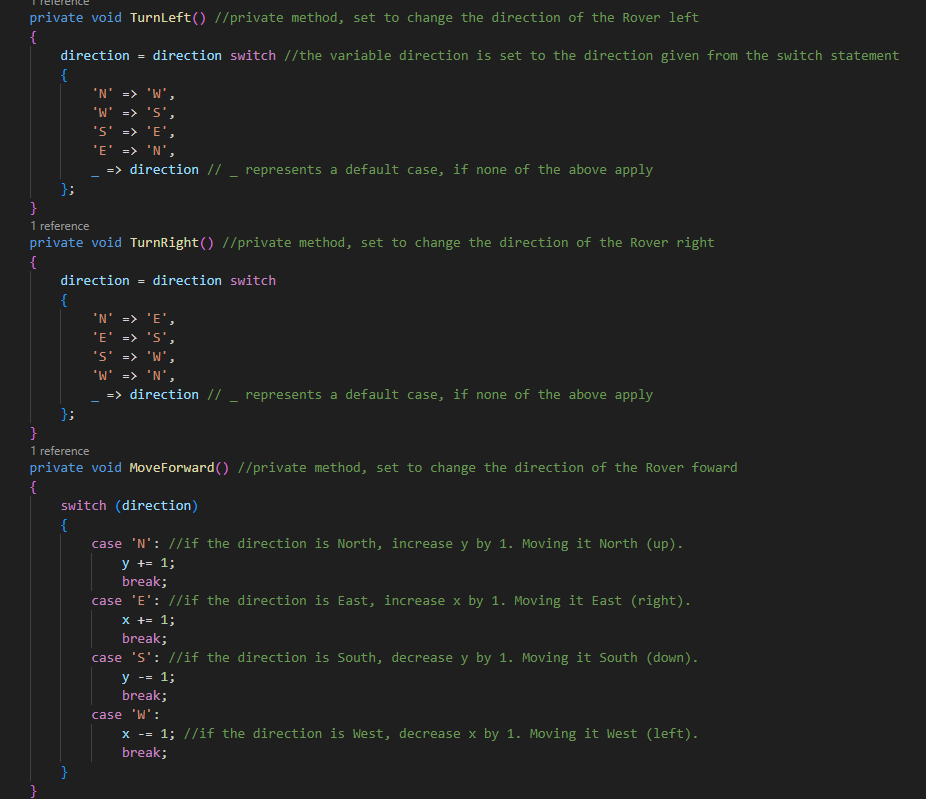


I began by coding the Rover, I began by creating x,y and direction (z), the max plateau X and Y properties for the rover. I then created a public constructor assigning the created variables to the new one within the constructor.

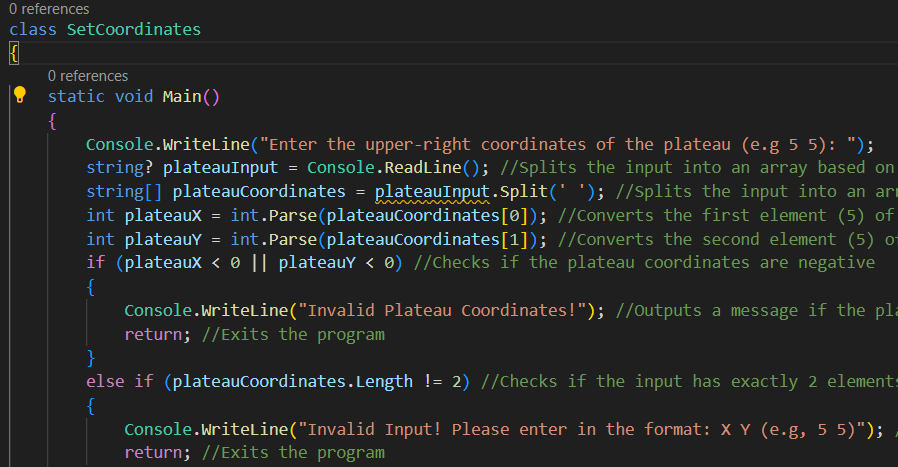
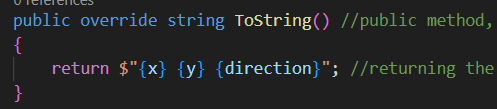
[BELOW] I then created a public method used to set the rover’s commands like (L, M, R). It makes use of the switch statement and cases. Once I completed this. I realised that the commands could only be typed in once a line as it was classed as a character rover1.move(‘M’).



So I created another public method. This one to separate a string into separate character using the foreach loop so that the method (Move) could manually go through each Character and give an input out

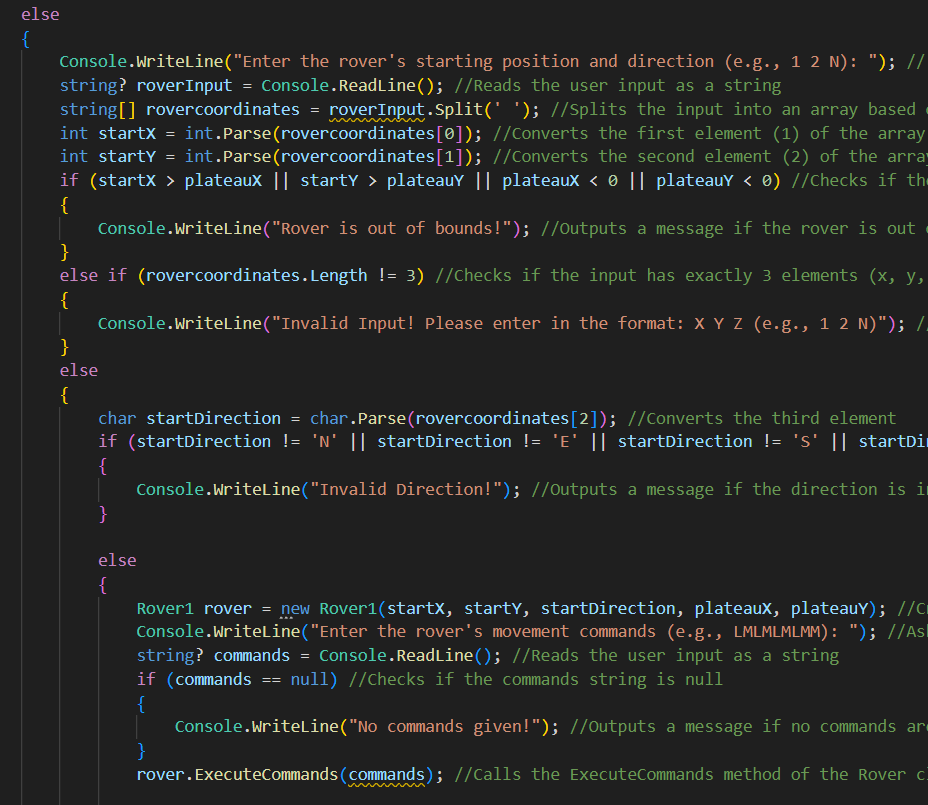
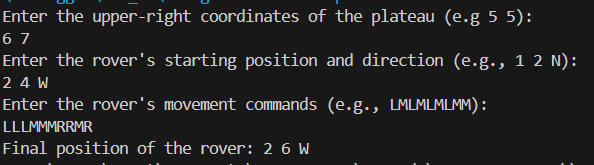
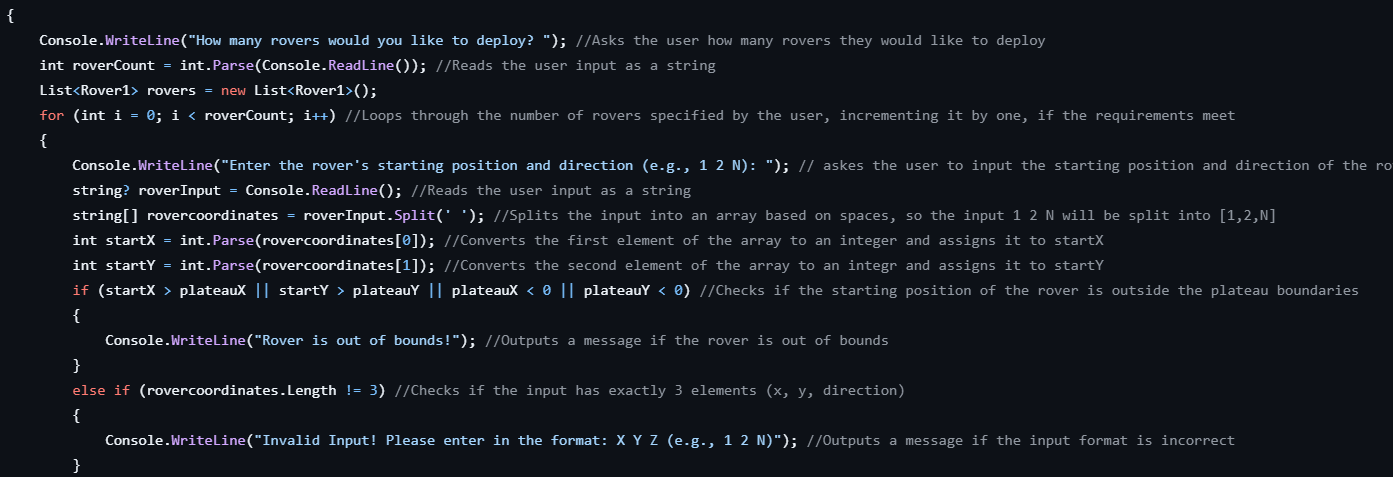


For the Rover Controls, I Created 3 private methods, Turnleft(), Turnright(), and MoveForward(). To accomplish the initial and final movements, I created a variable called direction and linked it to the switch statement beforehand. I then listed out the direction changes “ N to W, W to S etc. I repeated this for the TurnRight(). For the forward movement. I created Another switch, once again linking it to the direction. Then I added multiple cases, stating the movements on an axis grid based on the direction the rover is facing



To be able to get user input, I created a new class called Setcoordinates. I then printed the requested user input and saved the input into the string plateauInput. I then split the given input into two based on the spacings given and save it into plateau coordinates. I then took the split Input and saved it into plateauX for the X Coordinates and PlateauY for the Y coordinates

To output the coordinates, I created a public method, to return the coordinates



Overall, I found this Coding Task A big step up to what I was used to code. I found some bits quite similar to Python, But Most of the parts were different to the language I was Used to. Learning C# was quite difficult, with its new syntax, the way variables were implemented and the coding style in general, to learn but with a bit of practice I got the hang of it. The hardest part of this task in my opinion was the debugging and coding phases. I struggled to find why my code went wrong, and had to go through my code over and over again, till I solved it.

Once all the basic requirements were met. I moved on to the last requirement. Multiple rovers. I created a loop that kept repeating till the desired amount of rover was met

I struggled to find a reason why the code would not work. After troubleshooting over and over again, I finally found the issue that caused this error. The if startdirection requirements stated that the requirements was “and” not “or” as I had used the || instead of &&. After fixing it and testing it. The system worked. Below Is the output.

I then did the same thing for accepting the Rover’s initial input and The Rover’s commands. From here I ran into multiple issues, where I could write numbers or strings that did not correlate to the codes required and the system would still continue. To solve this, I had to create if statements to test and try to see If the inputs would match the required strings / characters / integers required. Upon testing, I ran into another error. This time, even though the correct input was put into the console, it would always show invalid directions and the system would close.