Team Computing

Project Document

Alan Gavin, Kenneth Feeney

After we looked at the project as a whole we decided upon the best approach to take. We wrote up our progression plan and intended to stick to it as best we could. We arranged the tasks in a chronological order with the more simple tasks preceding the more difficult tasks, such as manoeuvring the obstacle.

We had a clear vison of what we needed to have done each week to ensure we met our deadline. In the first week we both worked on two different aspects of the project. In theory this meant we could complete our jobs much faster. However, in reality, we found each of us encountered far too many problems individually to work alone and that the best approach would be to work closely together. We could also see that other groups were having a lot of problems when work was split up too. This new strategy worked well as we found we fixed bugs much quicker when the two of us together examined the code. Initially the outline of our progression plan was as follows…

* Week 1- Navigating the grid efficiently. Which consisted of coding for turning and stopping at the edge based on the line\_count variable.
* Week 2- Reading in black and white squares, number of rows and columns and total number of squares. All this information was to be mapped to an external file.
* Week 3- Manoeuvring the obstacle and mapping its position on the grid. Also determining the colour of the square the object is placed on
* Week 4 – Optimizing code. Improving speed and eliminating flaws and discrepancies.

At the end of each lab we looked at the next few tasks to be done. Each of us would take the problem and during the week would write up some pseudocode and logic to have ready for the next lab. This helped us tremendously as we had something to work off as soon as we landed into the lab.

We did encounter several aspects of the project which turned out to be very time consuming. Together we worked through solving the various problems, keeping a logbook of our failed approaches. The coding didn’t come as natural to us as it did to some other groups but we worked very well together which somewhat made up for this.

Dividing tasks was based on our individual strengths. When it came to the actual making of the program for the robot, we split the work into research and coding. Alan completed the majority of the physical coding with Kenneth helping with reading resources and task plan. This was not set in stone however as we both helped each other out on all aspects of the project. It helped that we get on well, as some other groups did not and we found it particularly easy to communicate and keep each other updated.

GitHub proved to be very useful towards the end of the project. Both of us were working from home to solve various problems and the tweaked code was easily passed back and forth. We also divided the code into separate snippets with each task getting its own individual compiled program. This ensured that each program performed its task without any errors. In the end we would join all these together to hopefully create a bug free program.

We learned a huge amount over the course of this project. Neither of us knew how to complete the tasks straight off the bat, but with reading, trial and error we grasped each individual concept as we went along. Our teamwork skills also seen a huge improvement. An example of this is when we couldn’t figure out a bug within our code. Reading line by line we each proposed possible solutions and through a tedious process of trial and error finally figured out a solution. It was also refreshing to see a physical implementation of a program we had written which was a change from our usual practises.