**Programming Things Individual Report**

Emran Hamdu B9016310

**Introduction**The start of the project was a slow start as we struggled to find a solid project that would be appropriate for 4 people. We have come to the conclusion of making an RC robot that can be navigated by the user. To ensure that the project is worthy of four people we decided on several features such as a Geiger counter, heat sensor, light sensor, camera, Xbee, and Zumo. Using all these features can be combined to make a solid project, to make a build, that can be used to detect any hazardous or dangerous area before any human is in harm’s way. It will essentially make sure that there is nothing that can be harmful to a human being before entering the room.

We have created a project that can control the Zumo robot using GUI created in python, that pushes a Lego built structure holding all the components needed such as the MKR which is a Wi-Fi 1010 Arduino board that is being used to wire the light and heat sensors. The light and heat sensors are held together on a breadboard along with the MKR board. Lastly, the Geiger sensor is being held on the bottom of the obstruction to detect radiation.

When we first got together, we created a list of aims for the team to meet in order to complete the project and upload the features on Trello for everyone on the team to see, and here is the list:

* Control the Zumo wirelessly using Xbee
* Have a way to connect the sensors wirelessly and read the data on users’ device
* A way to hold all the components on a structure so that they can be moved together
* A GUI with information on screen for the user to read and evaluate
* Lastly a live camera feed that can be used to navigate the room without being inside the room

**Development**

For our methodology, we mainly followed the Kanban methodology as we stuck to Trello to see what needed to be completed, what was in progress and what is already completed.

Our main sensor, which we wanted to make sure works, was the Geiger sensor, which will be used to detect any radiation-emitting near the vicinity of the robot. How the Geiger sensor works is by having a gas trapped in a glassed tube and if there is any radiation near the gas, it will collide and pushes the electron away. In the glass container, there is a small wire that can be used to detect whenever the elector gets pushed, which sends current through the wire. Which we can then use to collect the data whenever there is a current that gets sent through, it will send a high signal to the MKR board pin.

This is how we can detect the radioactivity of that certain area, once the pin is high the code will count how many times it passes per 15 seconds and will send back to the users the amount of pin being high in that 15 sec. If there is a low count then, theoretically it should be no radioactive vis versa, high meaning radioactive. This code will essentially tell the board to read the radioactive activity it reads and log it so that it can save how many times it ticks per fifteen seconds. How this works, is essentially it will keep count of how many times it ticks over a certain period of time that you set on the Arduino code and then that will be given back to the user, which can be visual in the serial monitor.

A picture containing electronics, circuit

Description automatically generated

Figure : Geiger Counter Arduino

After the Geiger counter was set up and can give feedback to the user, it was time to test if it can detect anything radioactive. Following this link: <https://www.mentalfloss.com/article/63768/7-most-radioactive-items-your-home>, we tested various objects to see if the Geiger counter will be able to pick it up. However, we were unlucky and items such as bananas, Brazilian nuts, the fire exit signs, etc do not seem to have a strong reaction to the Geiger counter, however, the banana was the most radioactive out of them. Although banana may have sufficed, it wasn’t enough to keep a steady and strong signal to detect any radiation which we can use to give a good example in our demonstration of the project. So, we searched for more and found that the smoke detector emits high levels and so we opted for that instead.

The other sensor we used was a digital heat sensor. We originally intended to use an analogue version but there was too much hassle and disturbance that was caused by trying to use MKR which caused a large variance when reading the temperature of the room.

Another sensor we used was a light sensor that essentially checks the level of the light around the surroundings using the voltage. All this information gathered is sent to a WebSocket server to be displayed on the GUI live. To ensure that the GUI is connected with the MKR, the user’s device must be connected to the same network to work.

Other components we planned on using were Pixy Cam or OV7670 as our camera so that the user can get live feedback while controlling the robot. However, due to some complications with ordering and delivery, we could not get our hands on either of these so we had to change the camera to an old phone that can share the camera view using an IP Camera. Using a camera allowed the usability of the robot to be even more remote as the users would not even need technically be in the same vicinity as the robot to be able to navigate it. How the camera was set up is by getting the getting frames of the video and updating it on the Tkinter app, which is what we used for our GUI. After that, to give better usability of the application, the sensor’s information is inputted onto the label.

Once everything was done, all that was left to do was get all the components installed onto one solid mode of transport, so that the Zumo can be used to transport all the sensors into different parts of the room. To do this, we used Legos, as they have enough shapes to allow you to design any type of shape, but the other key reason for using this was that it has sturdy, so they will be able to support all the weight of the equipment, such as the power bank.

A picture containing floor

Description automatically generated

Figure : Lego Structure

The final thing that was left is the GUI so that the user can see and interact with the system with limited restraints. The GUI was split into sections so that it would be easier to develop. One of the key aspects of the GUI was the thread, which is essentially a PySerial module that is connected to the serial port with Xbee. The second thread was to connect all the data gathered using the sensors to then update the fields and values currently visible to the user.

**My Contributions**

My role within the project was mainly the supportive role, as I did not get key aspects of the project with sole responsibility but rather would help out where I can. The Lego construction was designed by me and my other team member where we looked for possible solutions that could help control the Zumo smoother and keeps all the components within the build and still work functionally. The Lego construction was one of the key aspects of the project, due to the fact, that the build needed to be robust enough to keep the sensors and other components within the user’s control. Once the structure was built enough to carry and support all the electrical components, the next issue we had to overcome was the USB connections we are going to need to and from the power bank, which was handled by making some blank space along the edge, so the user had easy access to it.

Another issue we had to deal with was keeping the Geiger counter on the Lego construction securely, but at the same time making sure that it can still detect radiation. This was resolved with a quick solution by using the extra free space below the construction to mount the Geiger counter and this was done by using rubber bands. We could not use anything stick as it could interfere with the board of the Geiger counter.

Whenever the team was in a debugging stage and need help with finding solutions, I tried to help by looking for possible alternatives code that could be used instead researched online with a similar issue or trying to figure out the solutions ourselves with trial and error.

**Conclusion**

Overall, the whole project was a success, we met the requirements set out to achieve with the functioning components that were implemented. Although the project may have been a success, there were some key issues that could have been dealt with better if we had more time. For starters, the Lego structure could have been done with more polishing so that there is no wasted energy rotating the structure. The main reason why this happened was due to the heavyweight of the payload of the components and power banks as well as the grip of the Zumo track was not always strong.

Another issue the project could make improvements on was using the original camera we originally intended to use, as this would have saved the time of setting up the camera with the IP Camera server and would be directly connected to the MKR which meant that it can be used to send the data over using that.

Finally, the last issue that I would have liked to change for the project, is to find better usage for holding the camera and Geiger counter as we are currently just using a rubber band to hold them in place. Maybe we could have set it up so that it can fit into place perfectly using other objects other than Legos

In conclusion, I feel that the project was a great success with few required improvements that could have been made this project better. The electricals we mounted on with success and did not fall out when done so. The main thing I would say was my contribution to the team could have been better and maybe instead of supportive role I could have help add many other features.