**Programming Things Report**

I have completed task 1 – 5 with a great deal of issues that I encountered and attempted to resolve.

Task 1: Manual control.

One of the main issues encountered when it came to controlling the zumo manually was learning all of the variables, syntax within Arduino that I needed to know to be able to create this functionality, firstly beginning with how to read in values and output values from the zumo to Arduino and vice versa. Secondly, I needed to setup the xBee so that it was communicating correctly with the zumo, and allowing it to communicate through a GUI created in the Processing environment. And with that having to learn all of the variables, syntax needed to create a GUI within processing and opening a port to connect with the zumo wirelessly. Once I had figured out all of these things, sending keystrokes from the GUI to the zumo and having Arduino convert these keystrokes into commands that move the zumo was rather simple.

Task 2: Border Detection

I encountered many issues when working with the sensors within the zumo, to begin with, I had to find the library that was created for the sensors, so that I could receive data from the sensors and output it to the console. Once I had looked at one of the examples from the ZumoShield library (border detection) it helped me understand how the sensors are used and how to receive information from the sensors and use it to do certain things. The second issue was calibrating the sensors to work with the environment that I had the zumo in. so I had to create a test program to output the sensor information constantly so that I could adjust the limits for the border detection e.g. white background with black borders. The second issue I encountered was detecting if the zumo had hit a flat wall, or the zumo had just bumped into one of the corridor walls. This was a hard issue to resolve since in the Arduino environment, IF statements do not work like in other languages, so I had to write nested IF statements checking if each sensor had encountered a black line to then know if it had encountered a wall. Once this issue was resolved, task 2 was completed fairly simply.

Task 3: Turning Corners

Not many issues encountered with this task except learning more methods of how to create different modes inside the Arduino environment since in Arduino there is a loop that is constantly looping no matter what. Resolved this fairly easily by creating different functions with loops of their own to avoid the loop created by the Arduino environment.

Task 4: Searching a room

The first issue encountered was finding out how to retrieve information from the ultrasonic sensor. After looking through the example in a library that I downloaded I understood how to do this, and all I had to do was create a loop which moved the zumo inside a room and constantly checked whether an object is within x range. The main issue was after making the ultrasonic sensor work, is getting the zumo to flawlessly navigate the room and return. This was hard to overcome and still has not been perfected. I attempted to use the compass within the zumo, however, it seemed to be more inaccurate than using time delays when moving the zumo inside the room. So I resorted to just using delays to move the zumo left to right with consistent checks sent to the ultrasonic sensor to see if there have been any objects detected. The problem with using this method is that depending on the battery strength the movement of the motors differs, so different delays will mean different amounts of movement.

Task 5: The T junction

With everything that I had learned from the previous tasks, the only issue I encountered with this task was getting the zumo to return after reaching the end of a corridor and ignore all commands. I solved this by timing how long the zumo spent travelling in the corridor (minus any time it spent searching a room) and uses this total time when returning and ignores any commands for this amount of time. The zumo is then put back into the mode where it is allowed to be stopped to check other rooms.

Task 6.

Attempted to store each turn into an array, with a Boolean for each room that saved found if an object was found in a room. If nothing was found in the room it would not add the turn into the array, hence, ignoring this corridor since nothing was found. In theory my logic was correct, however, the return journey simply did not want to work, doing the return journey with delays is not accurate enough as the zumo would overshoot/undershoot turns and turn into a wall, and ruin the return whole return journey.