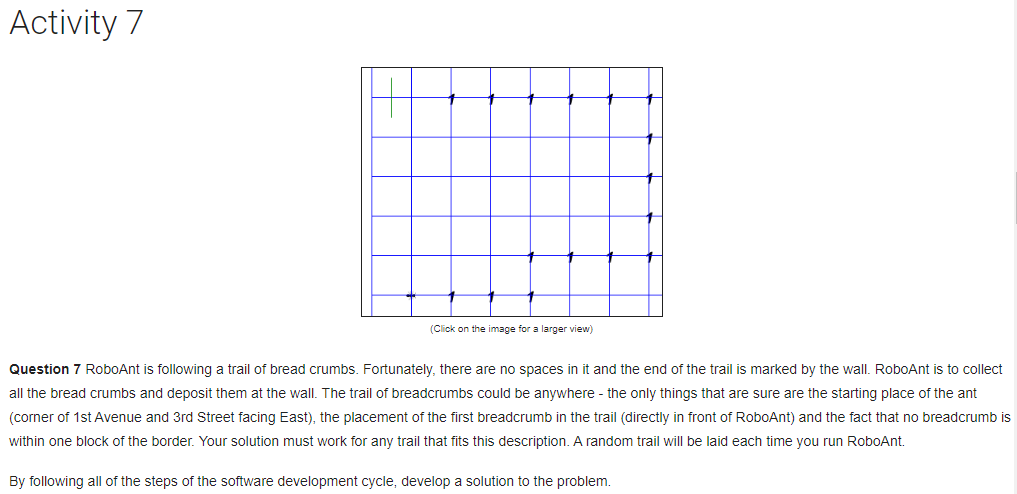
**(REMOVED PERSONAL DETAILS)– Digital Solutions FIA1 Final**

**EXPLORE- Analysis**

****

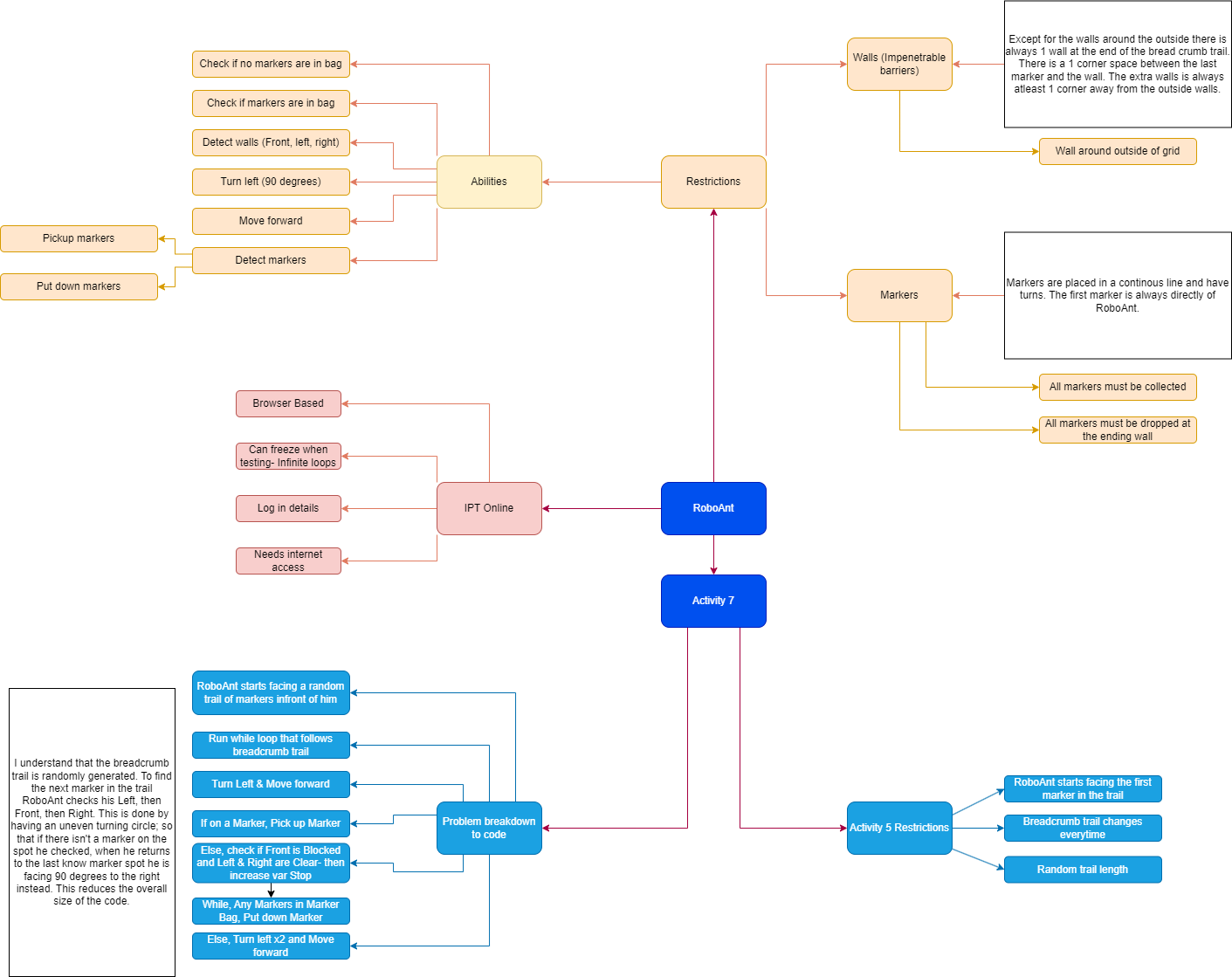
The starting point is fixed (Corner of 1st Avenue & 3rd Street facing East) with the first marker directly in front. The trail is always at least 1 corner away from the outer walls, reducing the need to factor in hitting them. There is always a wall one corner away from the last marker which can be used as a stopping variable.

There are no set challenges for this task.

The trail is completely random every time, therefore the solution must be universally compatible to work with any combination of markers. RoboAnt will have to check the corners around him to calculate whether he has moved onto a corner that has a marker on it.

Clear instructions stating that RoboAnt is following a trail of breadcrumbs (markers) and is to pick them **all** up then deposit them at the end.

**EXPLORE- Mind Map**

****

**EXPLORE- Criteria**

**Prescribed**

In activity 7, by 20/03/23 RoboAnt will complete:

1. Be able to find the next marker in the bread crumb trail and pick it up
2. Use the found marker’s position to find the next marker
3. Find ‘the wall’ and drop all markers.
4. Use iteration, selection, and nesting to complete the task.

**Self-Determined**

In activity 7, by 15/03/23 I will complete:

1. Create a main while loop to follow bread crumb trail using IF, ELSE IF’s.
2. Ensure when RoboAnt finds ‘the wall’ he drops all markers and increases the stop variable to exit the main loop.
3. Ensure the main loop works for **ALL** possible bread crumb trail combinations.
4. Complete task in under 30 lines of code
5. Use indentation, comments, white space, and proper variable names effectively.

**DEVELOP- Interface**

|  |  |
| --- | --- |
| **Starting interface**  There are many different breadcrumb trail combinations for RoboAnt activity 7. In need of this, the solution is built to work universally for any breadcrumb trail no matter the length or complexity. RoboAnt starts in the same spot each time (corner of 1st Avenue and 3rd Street facing East) with the first marker directly in front of him. The markers are placed either in front or to the adjacent sides of the marker behind it. There are no gaps in the trail and there is only 1 marker on each spot. | **Step 1**  For simplicity of the code, there is one function that is run the entire time. The first command is to turn Left and Move forward. |
| **Step 2**  If RoboAnt is on a Marker, he picks it up. | **Step 3**  If RoboAnt Isn’t on a Marker, it runs an If Else statement checking if the Front is blocked and Left & Right are clear. If they are it increases the stop variable, ending the main loop. It then will run a while Any Markers In Marker bad command to drop all markers at the end. |
| **Step 4**  Lastly is an Else command, Turn Left twice then Move Forward. When RoboAnt is checking the spaces around him, if there isn’t a marker this command returns him to the last space there was a Marker, since the turning circle is uneven, he is facing 90 more to the right. This reduces the overall amount and complexity of the code. This isn’t as efficient since RoboAnt checks every left instead of the front first- where markers are most likely to be. | **Overview**  The entire procedure for RoboAnt Activity 7 can be placed inside of a while loop (Except for the variables and Change delay command). RoboAnt first checks his left side for a marker, if there isn’t one there he returns to the last spot where there was a marker facing the opposite way. When the while loop runs again, he turns left, being 90 degrees right of the direction he just checked he then moves forward instead. Repeat again he is facing to the right. If there is a marker, RoboAnt starts the loop again, by starting to check the left etc. |

**DEVELOP- Algorithm**

|  |  |
| --- | --- |
| **Maps** | **PSEUDOCODE** |
|  | Increase Movement Speed to 100 Milliseconds  SET stop to 0  WHILE stop is equal to 0  Turn Left  Move Forward  IF On a Marker  Pick up Marker  ENDIF  ELSEIF Front is blocked & Left is clear & Right is clear  Increase variable stop by 1  WHILE Any Markers in Marker bag  Drop Marker  ENDWHILE  ELSE  Turn Left  Turn Left  Move Forward  ENDELSE  ENDWHILE |

**GENERATE- Code**

//Wk8 FIA1 Final V2.3

//Increase speed

Change\_Delay\_To(50);

//Stop Variable

var stop = 0;

//Main function

I used iteration (while loop) here because the length of the trail is unknown and because it is the most efficient.

while (stop == 0) {

Turn\_Left(); //Changes starting direction

Move(); //Moves onto new spot

This command ensures RoboAnt checks all sides of the current marker to find the next one in the trail.

if (On\_A\_Marker()) { //Checks new spot for marker

Pick\_Up\_Marker();

}

else if (Front\_Is\_Blocked() && Left\_Is\_Clear() && Right\_Is\_Clear()) { // Signals end of trail

stop++; //Ends function

while (Any\_Markers\_In\_Marker\_Bag()) { //Drops all markers

Put\_Down\_Marker();

}

}

If RoboAnt isn’t on a marker, this command returns him back to the last place there was a marker.

else { //Turns back & moves to last spot

Turn\_Left();

Turn\_Left();

Move();

}

}

//END

**EVALUATION- Error Checking**

|  |  |  |
| --- | --- | --- |
| **ERROR** | **TYPE OF ERROR** | **PROBLEM/SOLUTION** |
| Infinite Loop | Logical | Made a function call another function which calls the first function.  Rewrote the entire script to avoid this problem. |
| RoboAnt/ IPTO Online Crash | Logical | Don’t use While loops with a fixed variable.  Insert a condition inside the loop that changes the variable, so it becomes **false** for the primary loop otherwise it has no possible way to end. |
| Uncaught SyntaxError: Unexpected identifier 'else' On line: 22 | Syntax | Forgot to place an **(** at beginning of the while loop condition. |
| Uncaught SyntaxError: Unexpected token ';' On line: 17 | Syntax | Forgot to add the second **+** when increasing the increment of a variable. |
| Uncaught ReferenceError: Turn\_left is not defined On line: 23 | Syntax | Capitalised the L in Turn\_Left command. |

**EVALUATION- Evaluate Against Criteria**

I was successfully able to complete and achieve everything that was listed in the prescribed and self-determined criteria. I ensured that I gave myself reasonable time to plan out how I would complete the task. I ensured that I used the most efficient coding methods to reduce the overall size of the code.

I found that instead of running the main function as a function, I was able to make the function itself a while loop which reduced the overall size and complexity of the script.

Using a large range of IF & ELSE IF statements, I ensured that RoboAnt was able to follow the breadcrumb regardless of the combination or length. I also ensured that I used appropriate indentation, comments, white space, and proper variable names and kept the code length under 30 lines.

**EVALUATION- Suggested Improvements**

I feel that my solution has completed the required task effectively. I believe that my solution is the most efficient in terms of overall code length. Using much longer code, it would be possible to make RoboAnt move more efficiently such as checking the front first as this is where most markers are instead of checking the left first.