Digital Solutions

Assessment item: FIA1 Portfolio

Student name: James Macgillivray

Teacher name: Glen Nixon

Due date: 09/06/2023

Contents

[Activity 5 3](#_Toc137065451)

[EXPLORE 3](#_Toc137065452)

[Task Analysis 3](#_Toc137065453)

[Mind Map 4](#_Toc137065455)

[Criteria 5](#_Toc137065456)

[DEVELOP 6](#_Toc137065457)

[Interface 6](#_Toc137065458)

[Algorithm 7](#_Toc137065459)

[GENERATE 8](#_Toc137065460)

[Code 8](#_Toc137065461)

[EVALUATION 9](#_Toc137065462)

[Error Checking 9](#_Toc137065463)

[Evaluate Against Criteria 9](#_Toc137065464)

[Suggested Improvements 9](#_Toc137065465)

[Activity 7 10](#_Toc137065466)

[EXPLORE 10](#_Toc137065467)

[Task Analysis 10](#_Toc137065468)

[Mind Map 11](#_Toc137065469)

[Criteria 12](#_Toc137065470)

[DEVELOP 13](#_Toc137065471)

[Interface 13](#_Toc137065472)

[Algorithm 14](#_Toc137065473)

[GENERATE 15](#_Toc137065474)

[Code 15](#_Toc137065475)

[EVALUATION 16](#_Toc137065477)

[Error Checking 16](#_Toc137065478)

[Evaluate Against Criteria 16](#_Toc137065479)

[Suggested Improvements 16](#_Toc137065480)

[Appendix 17](#_Toc137065481)

# Activity 5

## EXPLORE

### Task Analysis

This means that each turn, coin, and length in the maze should be counted to be used to generate the percentages when the ant completes its journey.

Nested if statements were not used as it makes the code nicer to read. Reiterates the fact that the maze changes between tests.

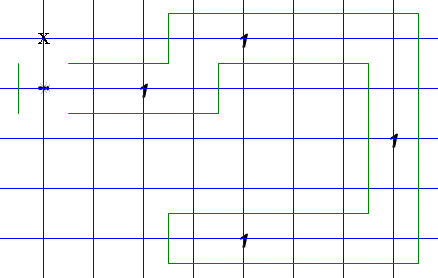
The main task which needs to be completed, is simple to interoperate

The image of the maze is very useful and gives many parameters not explicitly stated in the task. These include:

* There is only one path to the end of the maze
* The specific shape of the entrance to the maze

Gives Context for task. Has little use for completing the task.

Gives further detail for the tasks. Given the maze changes during each attempt, static code to complete a single maze cannot be used. The unknown amount of turns or length doesn’t really affect the final code.



You are at Maze City and dollar coins (markers) have been dropped randomly throughout the maze. There is no accurate map of the maze as it can be changed from day to day. It is unknown what the length of the maze is or how many twists and turns it makes.

**Question 5**

You are to program Robo to enter the maze and find all the money and return it to the entrance.

[Hint! You will need to use nested IF statements in your solution.]

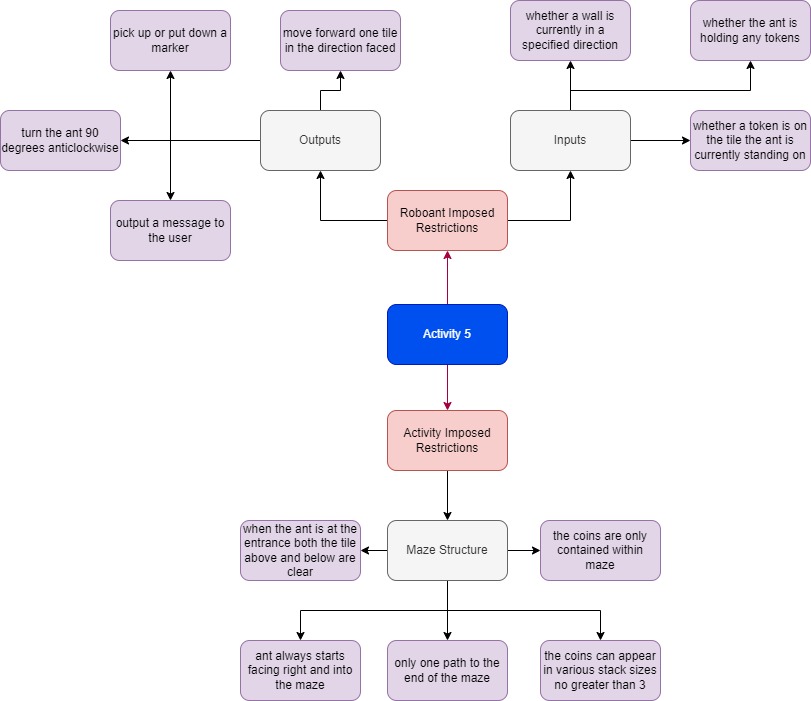
[Ensure your program works for coins in all possible positions in the maze, and also ensure your program will work for a maze that twists and turns randomly]

**Challenge**

* Display the number of coins as a percentage of the number of turns in the tunnel
* Display the number of turns as a percentage of the length of the tunnel

### Mind Map

This is relative to where the ant is facing and not absolute direction.



At each point the ant can only go left, forward or right except for the end where the ant must turn around

Can be chained to turn 180 or 270 degrees

### Criteria

In activity 5, by 9/6/23 RoboAnt will complete:

* Enter the maze and travel to the end without hitting any walls
* Pick up any coins along the way
* Return to the entrance of the maze
* Drop all collected coins
* Display Percentages as outlined in the task

In activity 5, by 9/6/2023 I will complete:

* Use while loops to loop the code effectively
* Use if statements to scan the area around the ant
* Keep code concise and space efficient
* Follow general coding principles with indentation, whitespace and comments

## DEVELOP

### Interface

|  |  |
| --- | --- |
| Start:  Both left and right of the ant are clear initially which will break navigation loop, so an initial move is used | Main function:  Every tick the ant scans its surroundings and determines whether it is standing on a coin and if so, pick it up then checks the direction of the next valid square and moves in that direction. This process repeats until it has reached a square where no forward tiles are valid |
| Once it has been determined that the ant has reached the end of the maze it turns around | Once the ant has returned to the entrance it simply drops all of its collected tokens |

### Algorithm

|  |  |
| --- | --- |
| Flow Chart | Pseudocode  WHILE true  IF standing on a token  Pick up token  END IF    IF returned to entrance of maze  Drop All tokens  End code execution (break)  END IF    IF left is clear  Turn left  END IF  IF right is clear  Turn right  END IF  IF path front is clear  Remain facing forwards  END IF  IF front, left and right is not clear  Turn Around  END IF    move forward  END WHILE |

## GENERATE

### Code

|  |  |
| --- | --- |
| Change\_Delay\_To(100) //to speed up testing  Move(); //initial move into the maze to fix navigation issue  //define counter variables required for challenge  //and kill variable to end main function  var turns = 0;  var moves = 0;  var coins = 0;  //main loop  while (true){    //check if on marker, if true pick it up and add to coin counter  while (On\_A\_Marker()){  Pick\_Up\_Marker();  coins++;  }    //check if returned to entrance, if true place  //down markers and exit main loop  if (Left\_Is\_Clear() && Right\_Is\_Clear()) {  while (Any\_Markers\_In\_Marker\_Bag()){  Put\_Down\_Marker();  }  break;  } | //check if at end of maze, if true turn around  if (Front\_Is\_Blocked() && Right\_Is\_Blocked() && Left\_Is\_Blocked()){  Turn\_Left();  Turn\_Left();  }      //check if left is the way to proceed,  //if true turn left and increase turn counter  if (Left\_Is\_Clear()){  Turn\_Left();  turns++;  }  //check if right is the way to proceed,  //if true turn right and increase turn counter  if (Right\_Is\_Clear()){  Turn\_Left();  Turn\_Left();  Turn\_Left();  turns++;  }    Move(); //move now that ant is facing the correct way  moves++; //increase move counter  }  //Calculate and diaplay percentage for challenge  Say("Coins/Turns "+50\*coins/turns+"%");  //50 is used rather than 100 because maze  //length is half the moves the ant takes  Say("Turns/Length "+100\*turns/moves+"%"); |

## EVALUATION

### Error Checking

|  |  |  |  |
| --- | --- | --- | --- |
| Error | Error Cause | Error Type | Solution |
| Infinite loop detected | Interpreter seems to simulate the code before displaying and if code doesn’t terminate after set period, it throws error | Logic | Using a for loop when testing and implementing while loop after code was functional |
| Uncaught SyntaxError: Unexpected token '(' On line: 25 | Using break as a function rather than a statement | Syntax | Removing “()” after break statement |
| Can’t Move there is a wall | Attempting to move into a wall | Intentional | Correcting the path of the ant |
| Uncaught SyntaxError: Unexpected token ';' On line: 20 | Using *variable*+; rather than *variable*++; | Syntax | Adding a plus at the end of the line |
| Integer outputed as “NaN%” | Not initialising variable as integer. “NaN” is an acronym for not a number (developer.mozilla.org, 2023) | Logic | Setting the variable to zero rather than leaving it as null |
| Can’t Pick Up Marker, no marker | Attempting to pick up marker when not standing on one | Intentional | Using if statement to verify ant is on marker before picking up |

### Evaluate Against Criteria

All points as described in the criteria and the activity analysis section of the report have been achieved. This includes but is not limited to entering the maze and travel to the end without hitting any walls, dropping all collected coins at the entrance, keeping code concise and efficient and displaying percentages outlined in the task’s challenge.

### Suggested Improvements

As the development process for activity 5 was relatively simple, all improvements that were thought of during the development process have been implemented. During the development process an improvement implemented was using a break statement rather than setting a variable to break the while loop. This was an improvement as the code immediately exited the loop rather than exiting after the code had reached the end of the loop. The only perceivable change this made to the user was removing the erratic movements of the ant at the when the ant had returned to the entrance.

# Activity 7

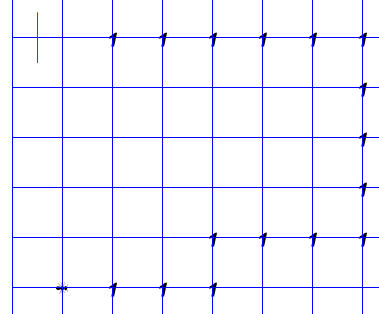
## EXPLORE

### Task Analysis

The main task in the activity. It is quite open ended only requiring the ant to move all tokens to a certain tile.

Gives Context for task. Has little use for completing the task.

Contrary to what this seems to imply, the fact that there are no walls makes the task more difficult because the only input for marker location checks the ant’s current tile unlike walls which can be detected from any adjacent tile.



Question 7 RoboAnt is following a trail of bread crumbs. Fortunately, there are no spaces in it and the end of the trail is marked by the wall. RoboAnt is to collect all the bread crumbs and deposit them at the wall. The trail of breadcrumbs could be anywhere - the only things that are sure are the starting place of the ant (corner of 1st Avenue and 3rd Street facing East), the placement of the first breadcrumb in the trail (directly in front of RoboAnt) and the fact that no breadcrumb is within one block of the border. Your solution must work for any trail that fits this description. A random trail will be laid each time you run RoboAnt.

By following all of the steps of the software development cycle, develop a solution to the problem.

This picture shows the following information

* The trail does not cross itself at any point
* The trail does not turn in the same direction twice consecutively ensuring there is no more than two adjacent tiles to any crumb

General broad statement which can be applied to all roborant activities

Further information about the random nature of the task. Starting location of the ant does not really matter when coding. Location of the first token in the code would be a cause of bugs if it was behind or diagonal the ant but this is not the case. No breadcrumb being adjacent to the border is useful as it prevents bugs caused by the ant either moving into it or mistakenly thinking it is at the end of the maze

Means a general solution must be used and not a for a single specific trail

### Mind Map

This is relative to where the ant is facing and not absolute direction.

At each point the ant can only go left, forward or right except for the end where the ant must turn around

Can be chained to turn 180 or 270 degrees

### Criteria

In activity 7, by 9/6/23 RoboAnt will complete:

* Follow the trail of breadcrumbs using the previous one as navigation
* Reach the wall at the end of the trail
* Drop all breadcrumbs

In activity 7, by 9/6/2023 I will complete:

* Use while loops to loop the code effectively
* Use if statements to scan the area around the ant
* Keep code concise and space efficient
* Follow general coding principles with indentation, whitespace and comments

## DEVELOP

### Interface

|  |  |
| --- | --- |
| Move forward and check whether on marker, if so pick it up. Then repeat | When the ant eventually reaches a tile which does not contain a token, it backtracks to the previous tile tries the tile which was originally on its right, checks whether it is on a marker, if so, pick it up and go back to step one. If it does not contain a marker than it can be concluded that the marker must be on the right of the original tile it was on. Then the ant picks it up and returns to the first step |
| A similar movement can be seen with right turns. | Once it is detected that a wall is in front of the ant, all tokens are dropped and code execution stops. |

### Algorithm

|  |  |
| --- | --- |
| Flow Chart | Pseudocode  WHILE true  Move  IF at end of trail  Drop all tokens  End code execution (break)  END IF    ELSE IF on a marker  Pick up marker  END ELSE IF    ELSE  Turn around  Move  Turn left  Move    IF on a marker  Pick up token  END IF    ELSE  Turn around  Move  END ELSE  END ELSE  END WHILE |

## GENERATE

### Code

|  |
| --- |
| Change\_Delay\_To(100); //speed up testing  while (true) { //main loop  Move(); //initial move    //check if at end of trail  if (Front\_Is\_Blocked()) {    //drops all markers  while(Any\_Markers\_In\_Marker\_Bag()){  Put\_Down\_Marker()  }  break; //breaks out of main loop    //if not end of trail check if on token  }else if (On\_A\_Marker()){  Pick\_Up\_Marker();    } else { //if not on token return on initial move tile  Turn\_Left();  Turn\_Left();  Move();    //move left, which is right of initial state  Turn\_Left();  Move();    //check if right of initial tile has the token  if (On\_A\_Marker()){  Pick\_Up\_Marker();  } else {  //since token is not front or right of initial tile  //token must be left    //turns and moves to inital tile preparing for move in next loop  Turn\_Left();  Turn\_Left();  Move();  }  }  } |

## EVALUATION

### Error Checking

|  |  |  |  |
| --- | --- | --- | --- |
| Error | Error Cause | Error Type | Solution |
| Infinite loop detected | Interpreter seems to simulate the code before displaying and if code doesn’t terminate after set period, it throws error | Logic | Using a for loop when testing and implementing while loop after code was functional |
| Uncaught SyntaxError: Unexpected token '(' On line: 25 | Using break as a function rather than a statement | Syntax | Removing “()” after break statement |
| Can’t Move there is a wall | Attempting to move into a wall | Intentional | Correcting the path of the ant |
| Uncaught SyntaxError: Unexpected token ';' On line: 20 | Using *variable*+; rather than *variable*++; | Syntax | Adding a plus at the end of the line |
| Can’t Pick Up Marker, no marker | Attempting to pick up marker when not standing on one | Intentional | Using if statement to verify ant is on marker before picking up |

### Evaluate Against Criteria

All points as described in the criteria and the activity analysis section of the report have been achieved. This includes but is not limited to following the trial of tokens, picking up all tokens, dropping all tokens at the final wall, keeping code concise and efficient and displaying percentages outlined in the task’s challenge.

### Suggested Improvements

As the development process for activity 7 was relatively simple, all improvements that were thought of during this process were implemented. During the development process an improvement implemented was using a break statement rather than setting a variable to break the while loop. This was an improvement as the code immediately exited the loop rather than exiting after the code had reached the end of the loop. This stopped the ant from moving into the wall at the end and neatens the code.

## References

developer.mozilla.org. (2023). NaN - MDN Web Docs Glossary: Definitions of Web-related terms | MDN. [online] Available at: https://developer.mozilla.org/en-US/docs/Glossary/NaN.