



Brunel
University
London

SOFTWARE DESIGN

DETECT OBJECT

TAHMID AL SIFAT

BSC IN COMPUTER SCIENCE (1ST YEAR)

BRUNEL UNIVERSITY LONDON

Requirement Specifications

Functional Requirements

1. SwiftBot Mode Selection :
 - a) The user will have 3 separate QR codes containing 3 modes.
 - b) The program asks the user to scan a QR code using the bot's camera.
 - c) The user scans a QR code to select one of three modes .
 - d) Curious SwiftBot,
 - e) Scaredy SwiftBot and
 - f) Dubious SwiftBot
 - g) The program should display the chosen mode after scanning the QR code and execute the mode.
 - h) The bot should wander around at the beginning of every mode.
 - i) The bot's ultrasound sensors will be activated so that it starts measuring the distance from each object or obstacle it encounters.
2. Wandering Around :
 - a) The bot should set its underlights as blue.
 - b) The bot should move forward for 5 seconds at 20% Speed.
 - c) The bot should move right for 3 seconds at 20% Speed.
 - d) The bot should move forward again for 5 seconds at 20% Speed.
 - e) The bot should move right again for 3 seconds at 20% Speed.
 - f) The bot will keep moving forward at 20% speed.
3. The Curious SwiftBot Mode :
 - a) The bot detects an object using ultrasound sensors by calculating the distance of the object from the bot, the bot keeps storing the distance intakes.
 - b) The bot should maintain a 30 cm buffer zone between the SwiftBot and the object and the object has to be exactly 30 cm distant from the bot.
 - c) If the distance from the object is more than 30 cm, then the bot should move forward to make the distance 30 cm while underlights are set as Green.
 - d) If the distance from the object is less than 30 cm, then the bot should move backwards to make the distance 30 cm, while underlights are set as Green.

- e) When the distance of the object is exactly 30 cm, blink underlights as green.
- f) Upon establishing a buffer zone the bot should not move and will remain stationary.
- g) The program counts the object.
- h) The bot should take a picture of the object, save the image in a file path in the user's device.
- i) The program should start a 5 seconds timer.
- j) The bot should wait for 5 seconds and check if the object has moved.
- k) The bot should check the movement of the object by calculating if the distance has changed or not.
- l) The bot should adjust the position to maintain the buffer zone if the object has moved.
- m) After 5 seconds of no object detection or movement, the bot should wait for 1 more second and start moving again in a slightly different direction.
- n) The same process will be repeated unless the user wants to quit.

4. The Scaredy SwiftBot Mode :

- a) The bot detects an object using ultrasound sensors by calculating the object's distance from the bot, the bot keeps storing the distance intakes.
- b) The bot should detect an object if it is within a 50-cm distance.
- c) The bot should set the underlights as Red upon detecting the object.
- d) The bot should click an image of the object and save the image in a file path in the user's device.
- e) After detecting an object, the bot should blink the underlights as red.
- f) The bot should set the underlights as Red again.
- g) The bot moves backwards.
- h) The bot turns 180 degree.
- i) The bot moves forward again for 3 seconds.
- j) After the turning, the bot should remain stationary.
- k) The bot should check if the object has moved by checking if the distance is more than 50 cm.
- l) If the object has moved, then the bot should wander around again until it detects another object.
- m) If no new object is detected for 5 seconds, the bot should wait for 1 more second, change direction and start moving again.
- n) The same process will be repeated unless the user wants to quit.

5. Dubious SwiftBot Mode :

- a) The program selects a random number between 1 and 2
- b) If it is 1, then the program should execute the Curious Mode
- c) If it is 2, then the program should execute Scaredy Mode
- d) The mode ends after executing one of the mentioned modes

6. Termination Process :

- a) If the user presses the “X” button on the SwiftBot, the termination process begins.
- b) The program should calculate the total duration of the operation.
- c) The program should create a log file to save the execution log.
- d) The program should display and provide the user with two options.
- e) If the user wants to view the execution log, then the user should press the “Y” button on the SwiftBot.
- f) The execution log should contain the name of the mode it ran, the duration of the operation, the number of objects it encountered, the file path where the images are saved and the file path where the log file is saved.
- g) If the user does not want to view the execution log, then the user should press the “X” button again on the SwiftBot.
- h) If the X button is pressed again, then the program should display only the file path where the log file is saved.
- i) The program ends.

7. Command-Line Interface

The program should be controlled via a simple, interactive CLI that prints outputs and takes input from the user.

Non-Functional Requirements

- 1. The program should validate the scanned QR codes.
- 2. The program should reject invalid QR codes and display an error message instructing the user to scan the QR code again.
- 3. The program should also validate the button pressings on SwiftBot, if the wrong buttons are pressed, it should prompt the user to press them again.

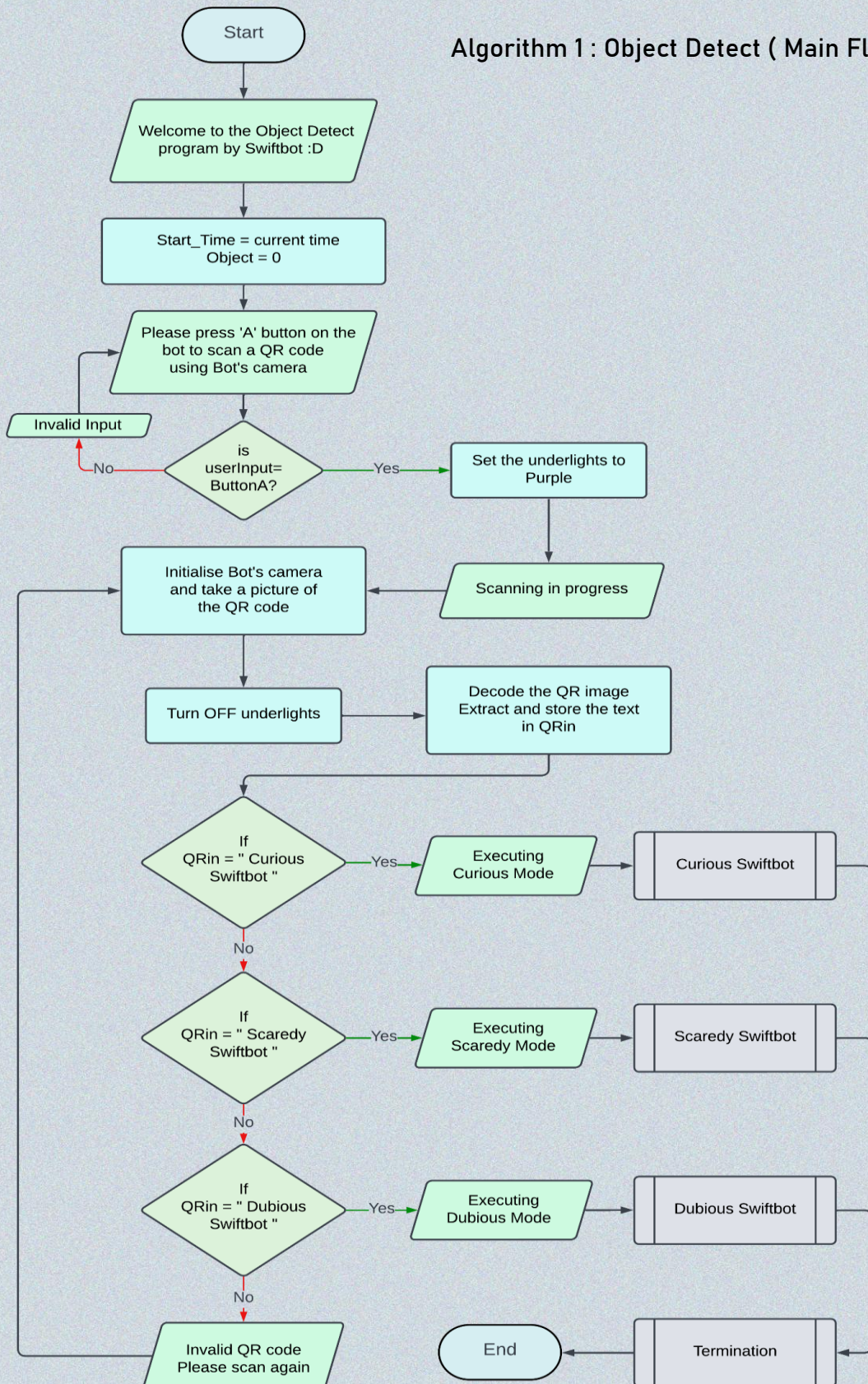
4. There should be display messages for the user when SwiftBot is :
- a) Wandering around.
 - b) Adjusting position.
 - c) Changing direction.
 - d) Detects an object.
 - e) Begins termination of the program.
 - f) Ending the program.

Additional Functionalities

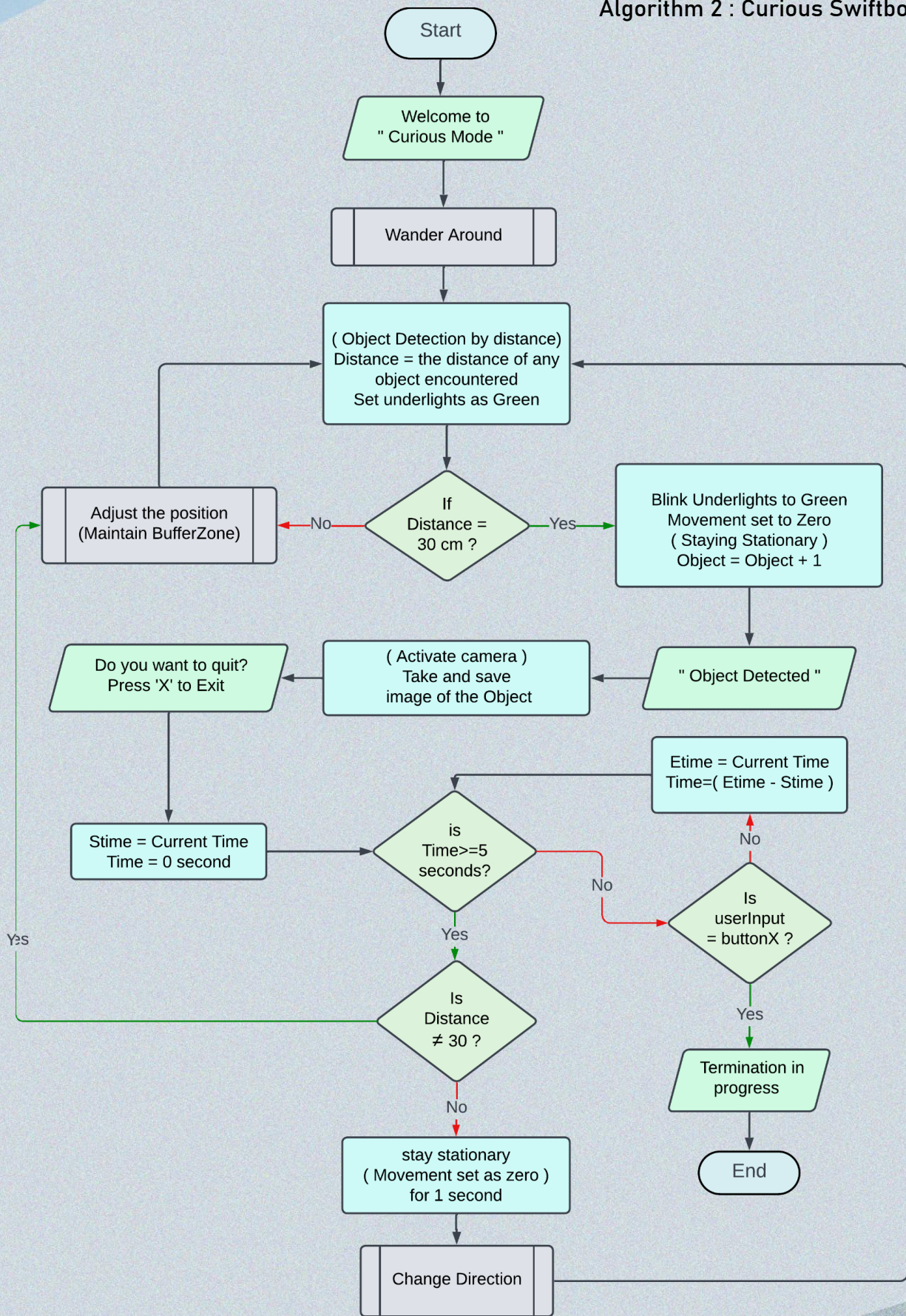
1. To scan the QR code, the program will ask the user to press the "A" Button on the SwiftBot to initiate the scan.
2. The program should validate if the user pressed the "A" button or not for the scanning of the QR code.
3. The program should display a message saying " Scanning in progress".
4. If the user presses any other button then the program should ask the user again to Press the "A" button.
5. While scanning the QR code, the underlight should be set to Purple.
6. After detecting an object, the program should ask the user with a display message if the user wants to terminate the program.
7. The program will ask the user to press the "X" button on the SwiftBot to proceed to termination. After the user presses X to terminate, the program displays " Termination in progress" before exiting.
8. The program should show a Welcome message at the beginning of the program: "Welcome to the Object Detect operation by SwiftBot :D ".
9. If the user presses "Y" inside the termination process, the underlights should blink twice as Orange.
10. If the user presses "X" inside the termination process, the underlights should blink twice as Yellow.

Algorithm Design (Flowcharts)

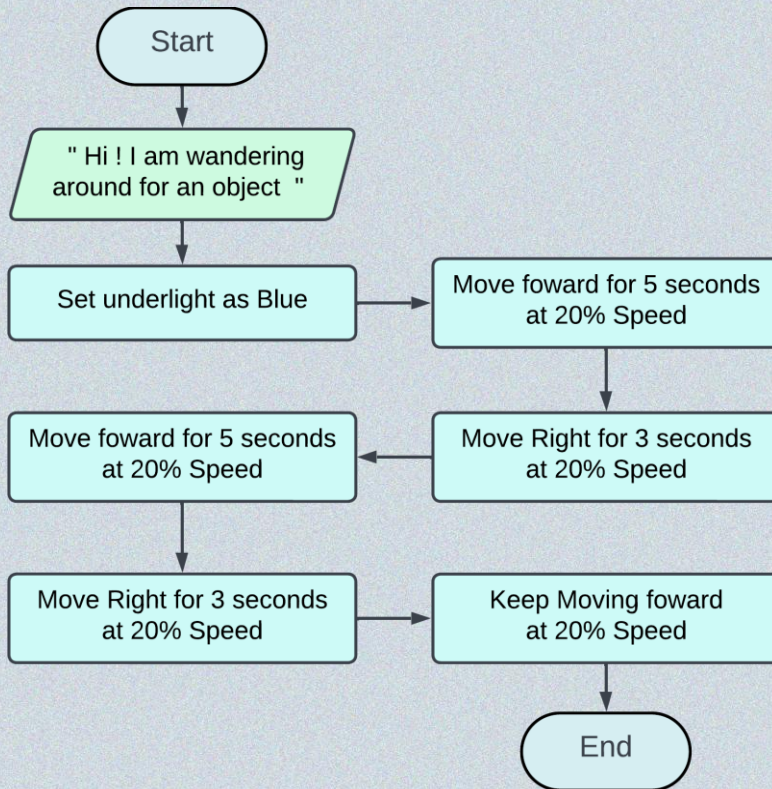
Algorithm 1 : Object Detect (Main Flowchart)



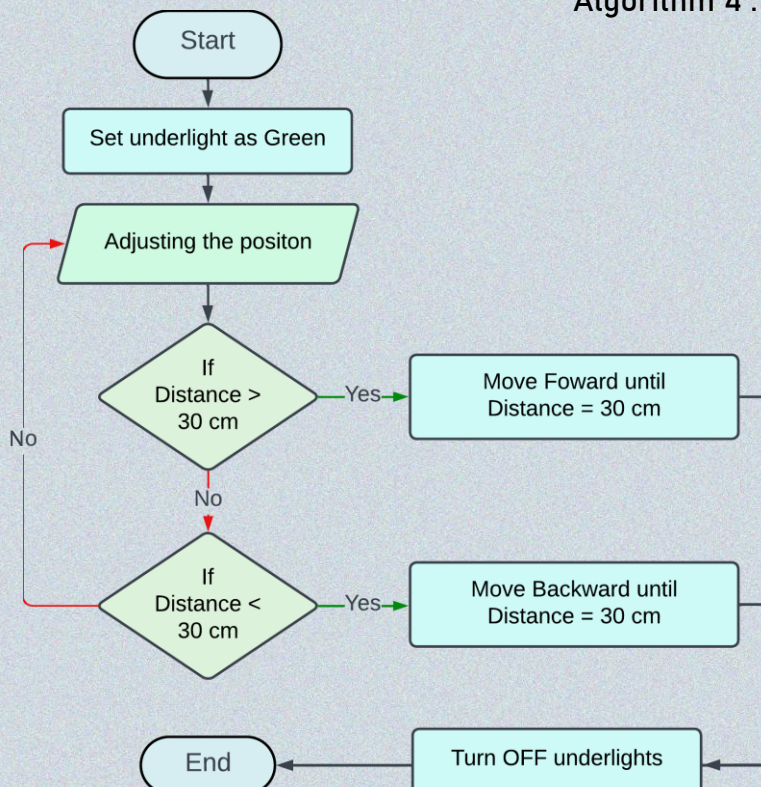
Algorithm 2 : Curious Swiftbot



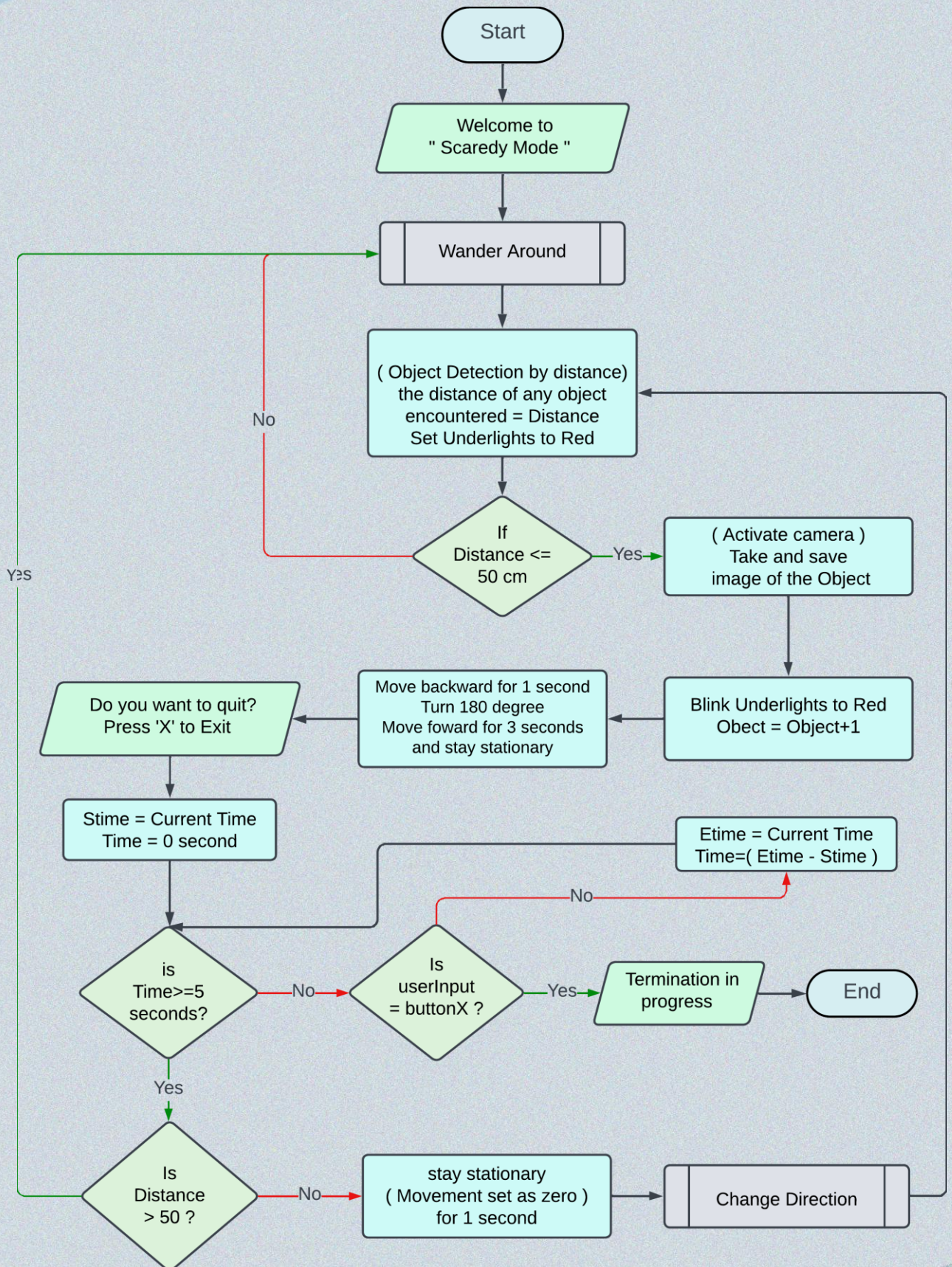
Algorithm 3 : Wander Around



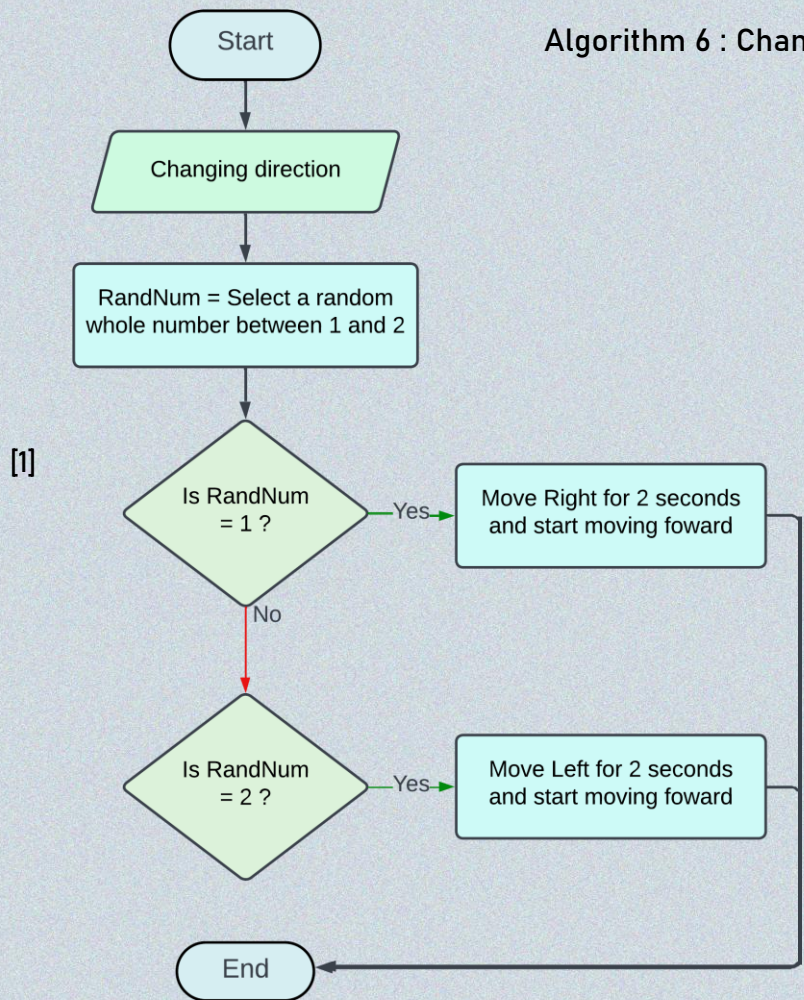
Algorithm 4 : Adjust the position (Maintain BufferZone)



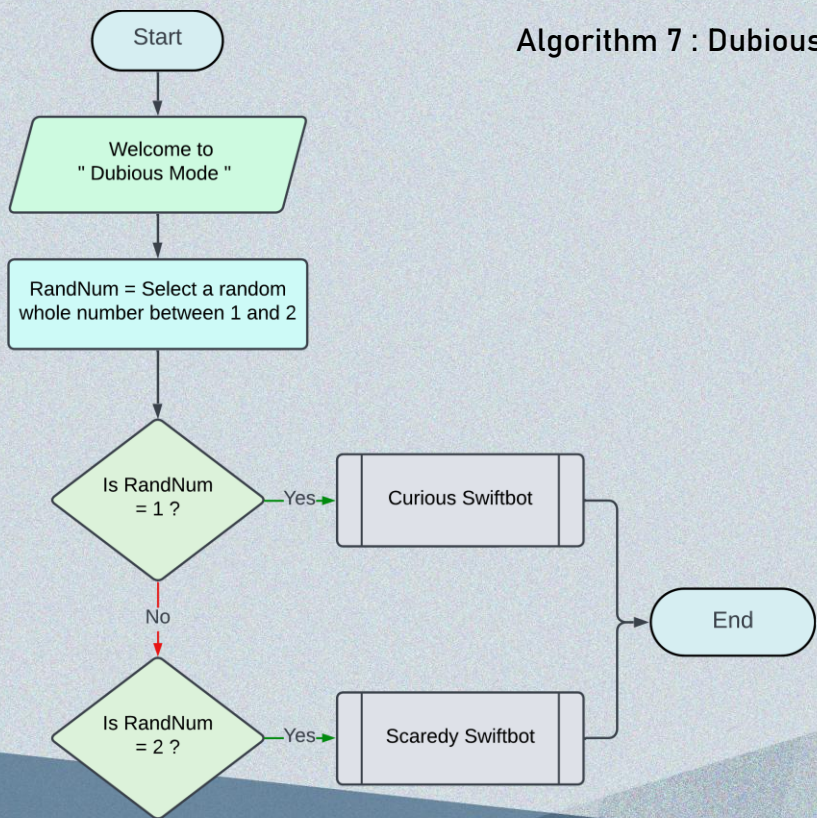
Algorithm 5 : Scaredy Swiftbot



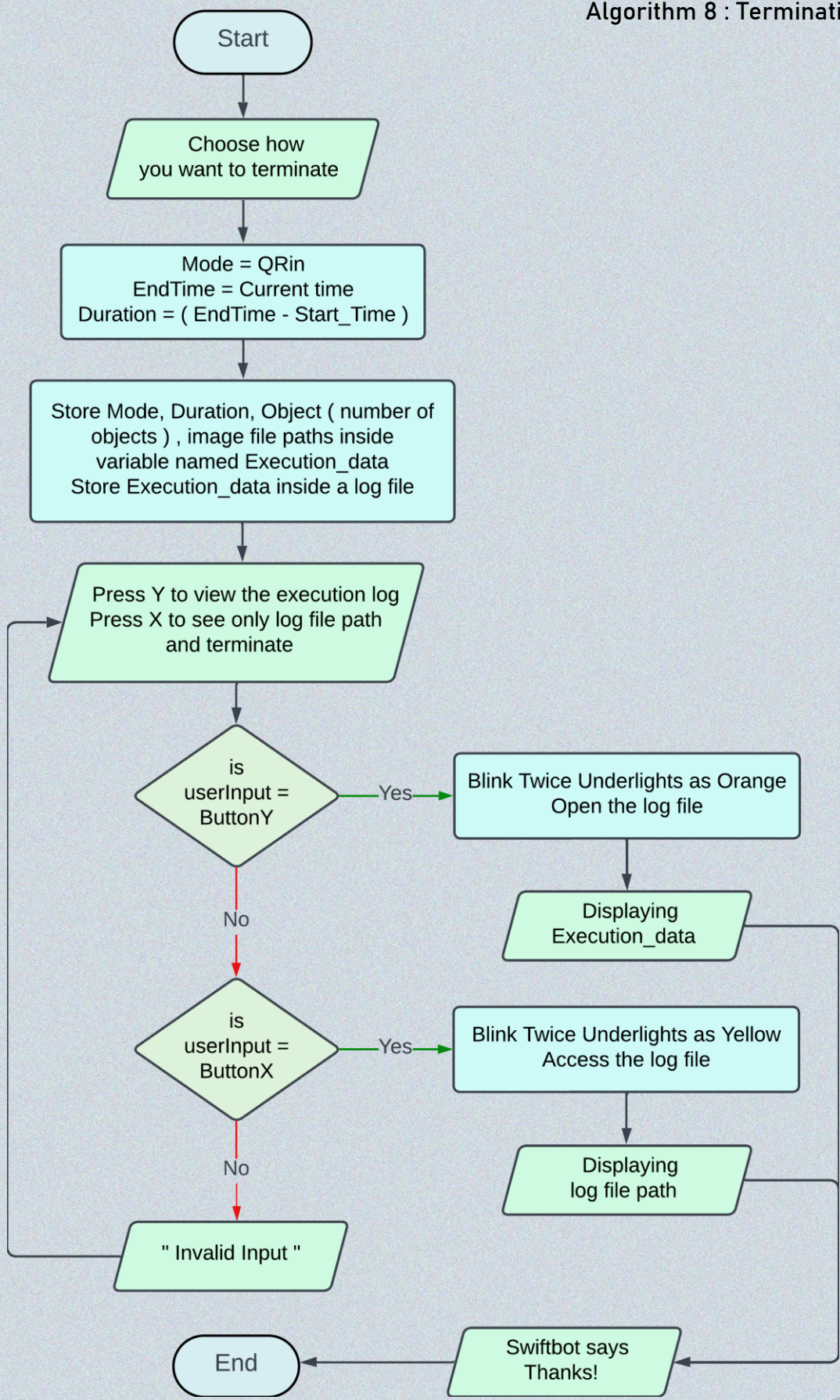
Algorithm 6 : Change Direction



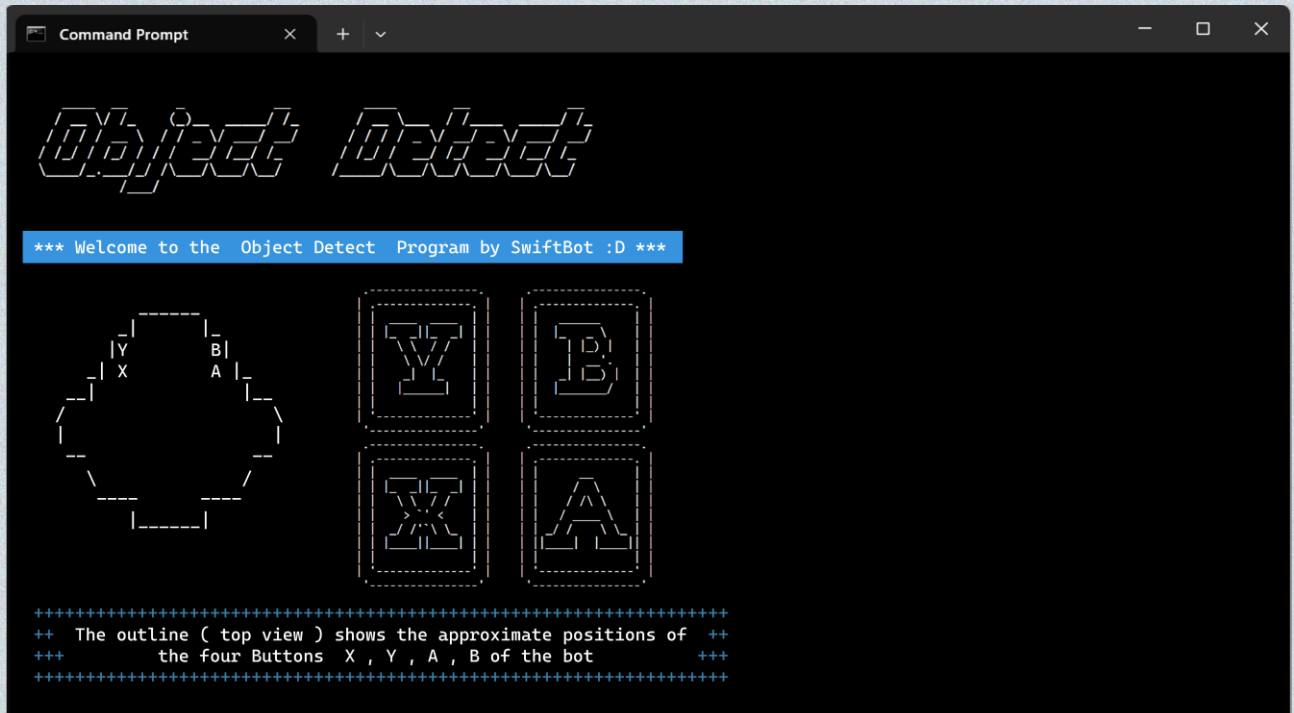
Algorithm 7 : Dubious Swiftbot



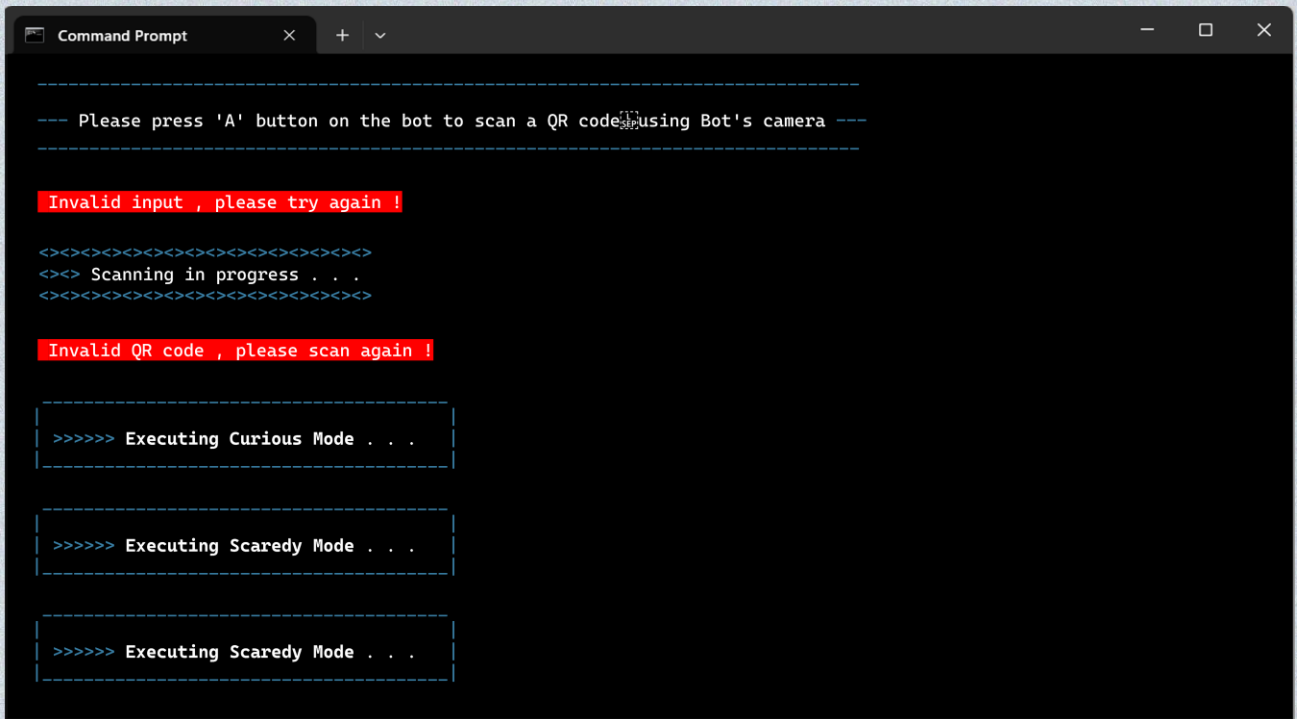
Algorithm 8 : Termination



User Interface Design (Prototype)



[2]




```
Command Prompt
*** Welcome to the Curious Mode ***

>>> Hi ! I am wandering around for an object <<<
<> Current distance is 30 cm <>
>>> Object Detected ! <<<
>>> Taking an image of the object <<<

Do you want to quit ?
Please press 'X' button to exit

>>> Termination in progress . . .

>>> Adjusting the position <<<

>>> Changing the direction <<<
```

```
Command Prompt
*** Welcome to the Scaredy Mode ***

>>> Hi ! I am wandering around for an object <<<
<> Current distance is 50 cm <>
>>> Object Detected ! <<<
>>> Taking an image of the object <<<

Do you want to quit ?
Please press 'X' button to exit

>>> Termination in progress . . .

>>> Adjusting the position <<<

>>> Changing the direction <<<
```



```
Command Prompt

Dubious Mode

*** Welcome to the Dubious Mode ***

Randomly selecting between
Curious Mode or Scaredy Mode

>>>>> Executing Curious Mode . . .

>>>>> Executing Scaredy Mode . . .
```

```
Command Prompt

Termination

>>> Choose how do you want to terminate <<<

Press 'Y' Button to view the execution log
press 'X' button to view only the log file path

Invalid Input ! Please press again

Mode of operation      : Curious Mode
Duration of operation  : 341 Seconds
Objects Encountered    : 6
Image file path        : F:\SwiftBot Images

>>> Execution log file path : F:\Assignment 2 , cs1704\SwiftBot Execution_log

Thank you
```