# IN621 Basic Behaviours SBA

|  |  |
| --- | --- |
| Assignment issued: | Week 5 |
| Assignment is due: | **31st August 2021** |
| Marks  *(Worth 10% of Final Course Grade.)* | **22 Marks** |

# Learning Outcomes Covered

## Design a robotics/automated solution to a specified problem following sound principles of interaction design.

## Use an appropriate software development platform to implement simple interactive robotics/automated systems.

# General Description and Conditions

* Make sure you have spare batteries. I will have some spares, but probably not enough for the whole class.
* **This SBA is open book**, but what you submit must be a genuine attempt at your own work.
* You have **one week** to “demonstrate” all tasks in this SBA. Tasks that are not demonstrated get a zero mark.
* SOFTWARE:
  + **All software written must be your own work** and be uploaded to Moodle before the end of the session. Software will be automatically checked for plagiarism and will get a zero mark if it is mostly not your own work.
  + **Your code must be written as a finite state machine (FSM).**
  + For servo behaviours you should be able to demonstrate **basic smoothing of your sensor data** before acting on it.
* You will not be expected to account for the general state of the room, conditions on the ground, or unexpected conditions in the test area. For example if your robot snags on the carpet or furniture.
* You may choose to **attempt this SBA in its entirety** a second time in order to improve your overall score.

# Ballistic Behaviors – No feedback from sensors.

Demonstrate the following:

1. **Drive a short distance 🡪 Stop**. This is a non-repeating sequence.
2. **Drive a short distance** 🡪 **Stop** 🡪**. Turn 90 degrees**. This sequence is repeated indefinitely.

# Servo Behaviors – Using feedback from sensors

Using the line-following board and your **line sensors**, demonstrate the following:

1. Starting on a white part of the board: **Drive forward 🡪 Stop when you detect a change in colour/brightness**. This is a non-repeating sequence
2. Starting on the white part of the board: **Drive forward 🡪 Turn left only when you detect a change in colour/brightness.** This sequence is repeated indefinitely.

Using the **accelerometer/gyroscope module**, implement and separately demonstrate the following behaviors:

1. **Tap-drive**: From a stopped state, **you will tap your robot and it will drive a short distance** before stopping. This is a repeating sequence.
2. **Collision detection**: Aiming your robot towards a wall, your robot will drive towards it. On contact with the wall, it **will immediately reverse** a short distance and then stop. This is a non-repeating sequence.

# Deliverables:

* **A video of your robot performing tasks is linked or submitted.**
* **Code for each behaviour** is separately uploaded to Moodle **by the due date/time for this SBA**

# Marking Schedule. Name:

**Task performance** *– maximum of 12 marks* **WEIGHTED AT 40%**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 2 | 1 | 0 |
|  | Robot fully performs tasks | Robot partially performs task | Robot does not perform task |
| Drive 🡪 Stop |  |  |  |
| Drive 🡪 Stop 🡪 Turn |  |  |  |
| Line drive 🡪 stop |  |  |  |
| Line drive 🡪 turn |  |  |  |
| Tap Drive |  |  |  |
| Collision Detect |  |  |  |

**Overall code quality** *– maximum of 10 marks* **WEIGHTED AT 60%**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 10 – 9 | 8 – 7 | 6 – 5 | 4 – 3 | 2 - 0 |
| Code quality | * Code is a FSM * Behaviours are evident (Servo and Ballistic) and well thought out. * Transitions between states are done very well.   Generally: Code is well written and follows good practice | * Code is a FSM * Behaviours are evident (Servo and Ballistic). Some logical issues (i.e. missing behaviours). * Transitions between states are done fairly well.   Generally: Code is ok. Some tendency to untidiness, but tries to follow guidelines. | * Code is a FSM * Behaviours roughly implemented. Missing, illogical or scope too small/poorly planned. * Transitions between states are poorly done, but work   Generally: Code is a hot mess. (But it does work, after a fashion) | * **Code is NOT a FSM** * Unable to discern behaviours in code * No transitions   Generally: Code barely fit for purpose. “Spaghetti” in Loop(), no methods used, etc, and mostly does not work | Extraordinarily poor quality work or no submission |
| Drive 🡪 Stop |  |  |  |  |  |
| Drive 🡪 Stop 🡪 Turn |  |  |  |  |  |
| Line drive 🡪 stop |  |  |  |  |  |
| Line drive 🡪 turn |  |  |  |  |  |
| Tap Drive |  |  |  |  |  |
| Collision Detect |  |  |  |  |  |

**Performance Mark** = (student mark / 12) \* 40

**Code Mark** = (student mark / 10) \* 60

**SBA Mark (Out of 100)** = Performance mark + code mark