Comp 1421 Foundations of Computing

Workshop6a Algorithms for Robots (Finite State Machines)

CBPrice October 2021

Color Coding for this Worksheet	
	Information
	Instructed Learning
	Autonomous Learning "Homework"
	Assignment material
	Research

All the activities below will prepare you for some of the examination questions (Assignment 2)

A note on the state variable and state definitions.

The states we use are all type **String**. This is so we can output them to the console with meaning. This is a little unusual, we would normally use the type **int** in any serious program.

The declaration of the states in the code look like this:

static final String STATE_1 = "STATE_1";

The words **static final** are variable "modifiers" and tell Java that the variable cannot change in the code, it is a CONSTANT. Also, the name of the variable **STATE_1** is written in upper-case throughout the code, to remind us that this variable is in fact not a variable at all, it's a CONSTANT! Finally, before the variable name, we write its type, in this case **String**.

A note on variables and what we are trying to do

These activities involve making the robot move in x-y space, and also change its "pose" by rotating to an angle "theta". They use the FSM approach. In each state, the values of the velocities are set to achieve a desired motion. Following the state code, the values of velocities are used to execute the dynamics (motion) of the robot. Here are the turtle methods:

turtle1.moveTo(x,y);	move immediately to the coordinates (x,y)
turtle1.turn(theta);	turn immediately to the angle *theta* in degrees

Activities

1 **A two-state Turtle FSM – "Reading Code"**

In this activity, you will ***read*** the code for a two-state robot FSM. In the first state, **STATE_1** the robot will move forward to a certain place. When it has reached that place, it will transit into the second state, **STATE_2** where it will stop.

(a) Open the file **robot_1.cde** which contains the entire code for the two-state Finite State Machine for a robot, as described above.

(b) Read the code (pass 1). In particular:
(i) Find the lines of code which declare the two states, and give them their names.
(ii) Find the lines of code where the current state is written to the console
(iii) Find the lines of code which *declare* and *assign* values to:
* the x,y, and theta values
* the x and y robot velocities, and the velocity of rotation ("omega")

(c) Continue reading the code (pass 2). There are two "chunks" of code; the first is the FSM code where the velocities (vX, vY, omega) are assigned values. The second chunk is where the dynamics (movement) is executed, where the FSM velocities are converted into movement. So you should look for the lines of code where:
(i) The dynamics are executed
(ii) The code for the two states is executed, in particular the *transition* from STATE_1 to STATE_2.

2 **Adding a third state to rotate**

(a) Open the file **robot_2.cde.** Rename STATE_2 to STATE_3 which will become the final state which makes the robot stop.

(b) Create a new STATE_2 to make the robot rotate until it has reached 90 degrees. Remember, you only change the velocity variables in the state code. You will need to experiment with the angular velocity **omega** to make the robot turn

3 **Adding a fourth state to move forward again**

Save your file **robot_2.cde** as **robot_3.cde** and add a state to make the robot move down after it has turned.

4. **Creatively create your own FSM**

Decide on a behaviour you wish the robot to execute and code this as a FSM