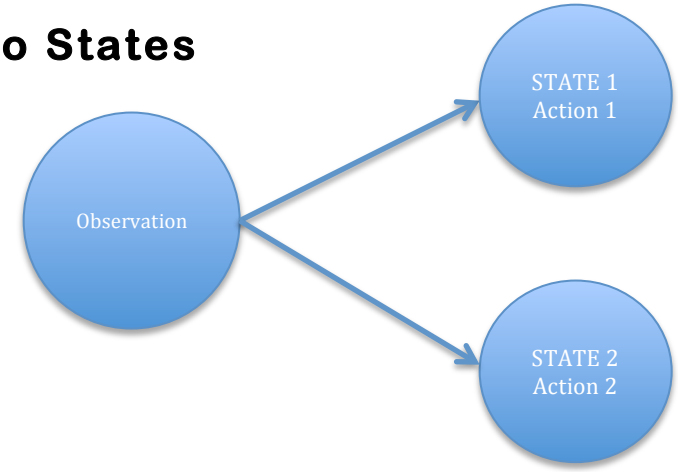
Finite State Machines

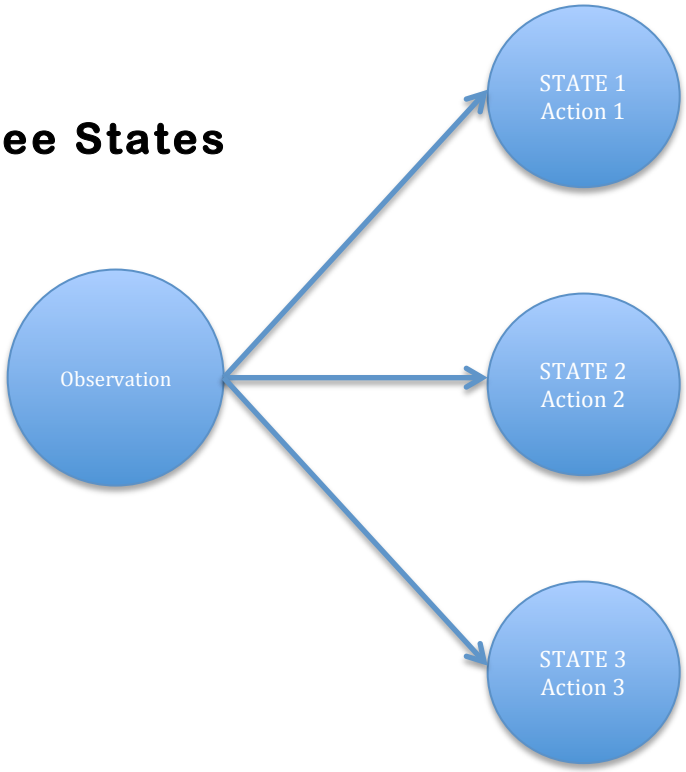
Definition

When we need to be able to take multiple actions for the same observation, we need **internal state**. Robots that use finite state machines have an internal state that tells them which action to take when they make a given observation. The robot starts with an initial internal state and knows the rules for when to switch between internal states. Using this information, the robot is able to keep track of which state it is in at all times.

Two States



Three States



Example

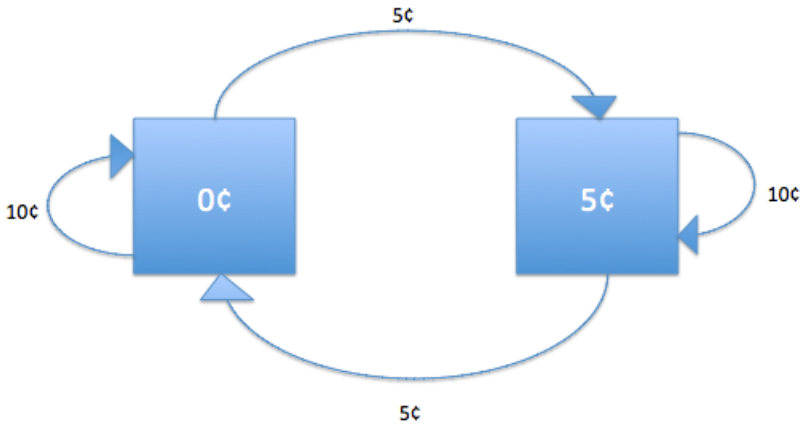
Vending Machine

In the video we looked at a vending machine that gave a soda for 10¢ and could take nickels (5¢) and dimes (10¢) as payment. The vending machine had two possible internal states: machine has 0¢ and machine has 5¢. Within each of the two states, there were two observations that the vending machine could make: customer puts in 5¢ and customer puts in 10¢. Based on the current internal state of the vending machine and the observation it made, there were two actions the vending machine could take: dispense soda and do not dispense soda. The vending machine starts with no money, so its initial state is the 0¢ state.

0¢ State		
Observation	**Action**	**Next State**
5¢	No Soda	5¢
10¢	Soda	0¢

5¢ State		
Observation	**Action**	**Next State**
5¢	Soda	0¢
10¢	Soda	5¢

Remember the state diagram from the video? We represented the different internal states with boxes and the different observations that triggered transitions between states with arrows between the boxes.



Note: We do not see the actions the vending machine takes represented on the state diagram. We only see the internal states and the observations that trigger new states.

Your Turn!

Now we'll consider a new example. See if you can fill in the blanks, the tables, and the state diagram!

Elevator

There is an elevator that can go to two floors, floor 1 or floor 2, and there are buttons for each of the two floors. If the elevator is on floor 1, and the button for floor 1 is pressed, nothing happens, and the elevator remains on floor 1. If the elevator is on floor 1, and the button for floor 2 is pressed, the elevator goes up until it has reached floor 2. Likewise, if the elevator is on floor 2, and the button for floor 2 is pressed, the elevator does nothing and remains on floor 2. If the elevator is on floor 2, and the button for floor 1 is pressed, then the elevator goes down until it reaches floor 1. The elevator begins on floor 1.

The elevator's two possible internal states are: _____ and _____.

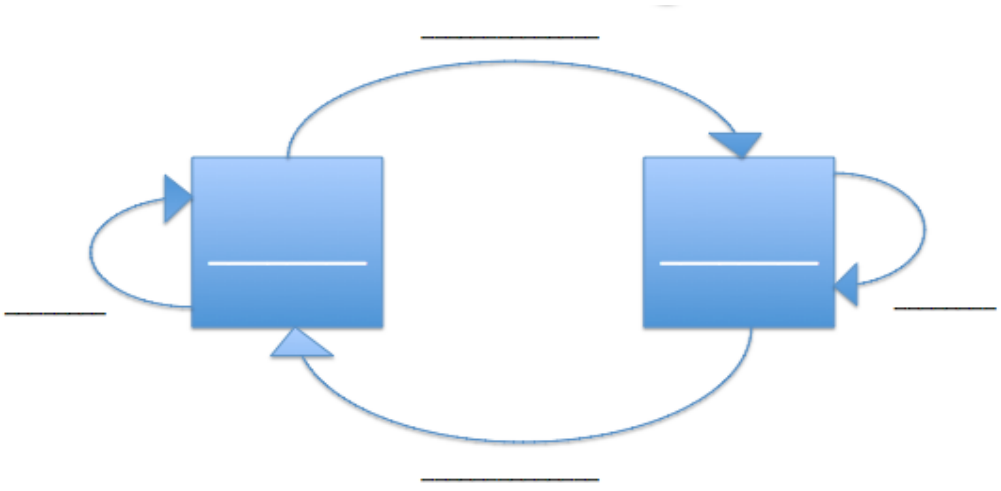
The two observations the elevator can make are: _____ and _____.

The two actions that the elevator can take based on its internal state and the observation it makes are: _____ and _____.

Hint: The actions are related to switching directions.

State: _____		
Observation	**Action**	**Next State**

State: _____		
Observation	**Action**	**Next State**



Lego Mindstorms Example

One application of the finite state machine concept using the Lego Mindstorms robot is following the line on both sides. You can watch a video of the robot performing this activity here: http://www.youtube.com/watch?v=tkytbh2au78

This example is an extension of the follow the line on the left example you completed in the behavior worksheet. You can use the code you wrote last time to help you complete this task. The tables from the Mindstorms example in the behaviors worksheet can help you remember which motors to activate in order to make the robot follow the line.

In the video, you can see that the robot takes two different actions when it makes the same observation, the yellow square. This means there are two different internal states. The yellow square triggers a change in internal state, and whenever the robot sees the yellow square, the action it takes is switch directions.

What are the two possible internal states? (Hint: Think about the two behaviors you coded your robot to perform last time)

_____ and _____

Using your answers above, can you fill out the tables for the two different states?

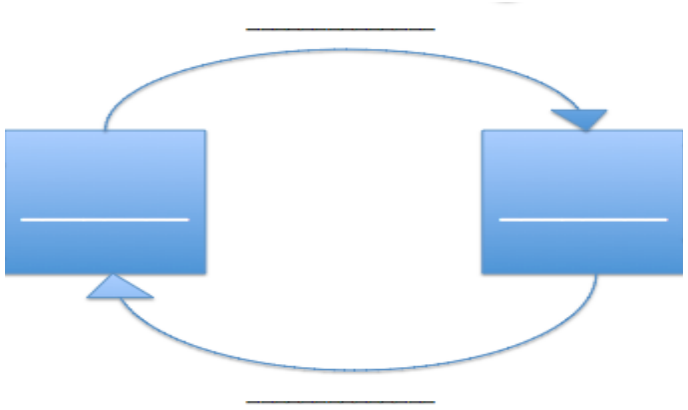
State: _____		
Observation	**Action**	**Next State**

State: _____		
Observation	**Action**	**Next State**

Now see if you can complete the state diagram for the finite state machine. Use the tables you filled out above and the previous examples of state diagrams in the worksheet to help you.

What goes in the boxes? _____

What do the arrows represent? _____



Now, use the NXT-G code to make your robot perform this task. Use the blocks of code you wrote for the following the line on the left and following the line on the right tasks, and add in a switch block that uses a variable to switch between states. When the robot no longer senses black or white (in other words, it senses yellow), the code should change the variable. Changing the variable changes the state. Don't forget that you still need a loop block around all of your code so that the robot performs the actions continuously.