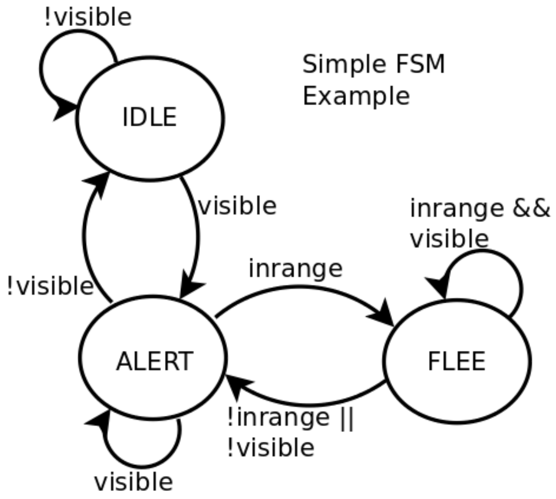
Simple Finite State Machine Example

W. Dobson   5-14-2018

The following represents the FSM implemented in the example code for assignment #3. It uses boolean inputs: *visible* and *inrange* to control the state transitions. Where *visible* means that this character can see the player and *inrange* means that the player has gotten too close causing the character to flee the situation. Although not in this implementation you may want to add a timer that used to extend/delay state transitions so the the character continues to flee for a while after the threat is removed.

State Bubble Diagram



State Transition Table

Present State	Transition Rules	Next State
IDLE	player not *visible*	IDLE
IDLE	player *visible*	ALERT
ALERT	player *visible* and not *inrange*	ALERT
ALERT	player *visible* and *inrange*	FLEE
ALERT	player not *visible*	IDLE
FLEE	player *visible* and *inrange*	FLEE
FLEE	player not *visible* or not *inrange*	ALERT

Note that typically the number of input variables (n) can control up to 2^n states. So for 2 inputs it can control up to $2^2 = 4$ states which is sufficient for the 3 states in this machine example.