**A.I. Notes**

**Finite State Machine’s (FSM) –** Are hypothetical Gears made up of one or more States in which only a single state can be active at any given time. The FSM has to transition from one state to another in order to perform the action associated with each state such as (Patrol, Chase and Attack).

**Implementation –**

Public enum State {Patrol, Chase, Attack}

Public enum Transitions {See Enemy, Reach Enemy, Hit By Enemy, None}

State currentState = state.Patrol;

Transition currentTransition;

Update (Game\_Loop)

//Search Phase

currentTransition = fillCurrentTransition();

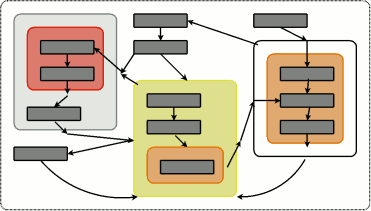
//Thinking Phase

Current State = ThinkAboutIt();

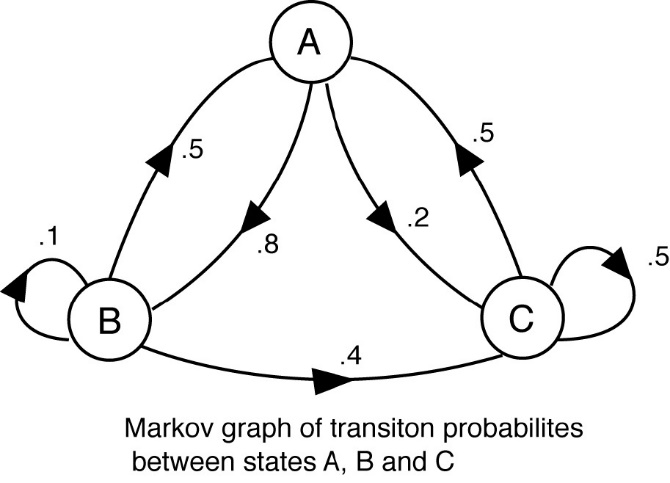
//Action Phase

ActionPhase();

**Hierarchical FSM’s –** overcome the limitations of a standard Finite State Machine while being able to maintain what a Finite State Machine can do by using a hierarchy systemwhich allows any state to be a sub-state of another state.



**Markov Model –** is similar to a FSM but a Markov Model uses multiple paths in which a state can be changed using probability.



**Difference Between Markov Model and a FSM –** A Finite State Machine follows it’s Unique Path through the Machine whereas a Markov Model may have multiple paths in which it can take depending on the probability of the path.

**Sensing Section –** The purpose of a sensing section for a NPC is used for patrolling/searching for a certain object/event/player in which will trigger a transition to another Section. Once the target is within range of the NPC it will transition from the sensing section to an action section

**Action Section –** The purpose of the Action section for a NPC is to determine what action to take when there is a target within an NPC’s range in which a specific action associated with the NPC and it’s target can be used.