**Finite-State Machine**

A finite-state machine is a model of computation made up of multiple ‘states’. Only one state can be active at any given time, so the machine transitions from one state to another to perform different actions. It acts as the ‘brain’ for AI in games.

Waiting

Moving

Following

Attacking

If the player is within 5 metres

If the player is not in sight

If the player is within 5 metres

If the player is within 1 metre

If the AI has no destination

If the AI reaches its destination

Finite-state machine of the Push-Push Penguin 3D AI.

**2016 Winter Exam**

Artificial intelligence (A.I.) can be useful in describing behaviours of Non-Playing Characters (NPC’s) in games. The implementation of A.I. is usually separated into sensing, thinking and acting sections. If we intend to implement a Finite State Machine (FSM) to deal with the thinking section…

1. What is an FSM? **(2 Marks)**
2. Discuss, with sample code, the implementation of a FSM referring to, with examples, the sensing, thinking and acting phases of the AI. **(6 Marks)**
3. What are Hierarchical FSM’s and Markov Models? What advantages do these have over FSM’s? **(4 marks)**

1

1. A FSM is any device that stores the status of something at a given time and can change the status/cause an action to take place for any given change.
2. A FSM can be implemented with a switch/case or if/else statement(s).

switch (NpcStatus) {

case (Status.Waiting):

GetNewDestination();

NpcStatus = Status.Moving;

break;

case (Status.Moving):

//The cases/if/else statements are the ‘Thinking’

if (PlayerInSight())

NpcStatus = Status.Following;

else if (!HasReachedDestination())

MoveToDestination();

else

NpcStatus = Status.Moving;

break;

case (Status.Following):

//The methods the if statements call are the ‘Sensing’

if (CanAttackPlayer())

NpcStatus = Status.Attacking;

else if (PlayerInSight())

MoveToPlayer();

else

NpcStatus = Status.Waiting;

case (Status.Attacking):

//The results of the if statements are the ‘Acting’

if (CanAttackPlayer())

AttackPlayer();

else

MoveToPlayer();

}

1. Hierarchical FSMs are an extension of FSMs, where any state can be a sub-state of some larger state. They are advantageous in that they provide ways to re-use logic by sharing transitions.  
   Markov Models are random process models which try to predict outcomes based on statistics derived from sequential data. They are composed of states and transitions.