

Spike Outcome Report

Number: 14

Spike Title: Soldier on Patrol

Personal: Chloe Mitterer 9725725

Goals:

Create a "soldier on patrol" simulation where an agent has two or more high-level FSM modes of behaviour and low-level FSM behaviour. The model must show (minimum)

- High level "patrol" and "attack" modes
- The "patrol" mode must use a FSM to control low-level states so that the agent will visit (seek/arrive?) a number of patrol-path way points.
- The "attack" mode must use a FSM to control low-level fighting states. (Think "shooting", "reloading" – the actual states and transition rules are up to you.)

Technologies, Tools, and Resources used:

- Sublime Text 3
- Python v 3
- Code from lab06 (path follow code)
- Spike 9 code (projectiles and shooting)

Tasks undertaken:

- Take out unnecessary code from the lab06 code
- Make the agent follow a continuous path, sticking close to the path.
- Create an enemy class and give world a list of enemies
- Create a finite state machine to control whether the agent is attacking or patrolling (seeking enemies or following the path)
- Make the agent shoot at one of the enemies, followed by the next enemy, followed by the next, and so on.
- Create another finite state machine within the attack state
- This state machine will control whether, in the attack state, the gun is either shooting or reloading.

What we found out:

- How nesting finite state machines works
- How to use finite state machines to control both high level and low level states