

Task 16 Spike: Soldier on Patrol

Context:

Agents in games can take advantage of layered state-machine designs with high-level modes of behaviour and lower-level step behaviour working together.

Knowledge/Skill Gap:

Developers need to be able to create layered finite state machine models for agents to use.

Goals/Deliverables: [CODE] + [SPIKE REPORT]

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)

- (a) High level "patrol" and "attack" modes
- (b) 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.
- (c) 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.)

Planning Notes:

- Start with the path following lab code and implement a FSM to control the high level "patrol" or "attack" modes.
- Utilise your agent marksmanship spike outcomes for the attacking mode behaviour.
- Be able to use the keyboard to add a new enemy agent to the game. (Makes testing much easier!) The enemy agent can/should be very simple to start with: doesn't move, doesn't fight back, appears at a random location, and can be easily killed. (Avoid adding too much complexity that doesn't align with the goals/deliverables and hence the spike gap.)
- The attacking agent will need to be able to "attack" the enemy. (A basic health/attack model is needed. Again - try to keep this simple to start with.)

Extensions:

- More high-level modes, more attacking modes, more agents, different regions of the map (recharge, hiding location) ...