State Machine Design

The environment simulated is a maze of nodes with a path flowing through valid nodes. It was made using an ascii map, with binary strings deciding the positions of nodes. 1 represents a valid node, and 0 represents an invalid node.

The AI swaps between two states: following and wandering. When the state machine agent is following, it targets the position of the player agent. The state machine agent has a speed that is half of the player agent.

When the program starts, the agent using the state machine starts by wandering around the maze. When it gets within 5 nodes of the player agent, the agent switches from wandering to following, where it chases the player agent until it reaches the player’s position. If the player gets further than 7 nodes away from the state machine agent, the state machine agent stops following and returns to wandering.

//what is wandering (how does it work mechanically)

//what is following (how does it work mechanically)

//what is gotopoint(how does it work mechanically)

To enhance the difficulty, the speed of the state machine agent would be able to be changed.

//How else could you influence the difficulty? 3 Examples