# Practical 9.1 – Finite State Machines

For this exercise, you are to use the FSM technique to implement a simple creature. The creature wanders randomly in an area containing food and obstacles. If the creature finds food, it eats it; if the creature finds an obstacle it changes direction to avoid the obstacle. The FSM diagram is shown below. A demo is on the I drive.

*FSM for Creature*

When in the Wandering state, the creature moves in a constant direction for a fixed number of timer ticks, then selects a new random movement direction. If it encounters an obstacle, the creature backs up a little, then moves into the Changing Direction state, where it selects a new randomly chosen direction before returning to the Wandering state. If it encounters food, it enters the Eating state. After entering the Eating state, the creature remains in that state for a fixed number of timer ticks. Its size then increases to allow the viewer to observe the effect of eating.

In the demo, creatures, obstacles and food are all descended from a common base class. A management class holds an array of each entity type and manages the high-level functionality. To help you get started, .h files for the base class (Thing.h) and the creature class (Creature.h) are on the I: drive.