Submission Due by 4pm on January 10th, 2023

Required:

This is a research project to bring together elements of the course that you have already covered in various modules. Create a new project and implement the following states using a state machine for a cat and squirrel simulation in a scene in Unity. You need to write the state machine for this and add it to both the cat objects and the squirrel objects. Do not use the built in FSM in Unity. There are many coding samples available online that you can reference.

Understanding the State Pattern using Super Mario

State Pattern using Unity

The State Pattern in C#

You have already covered the State Pattern in your studies and have implemented the Singleton in several modules as well. When you write the implementation here, you need to add a block comment at the start of your code to explain it. Do not comment each line individually, this is about understanding the concept and implementing it in the scenario given.

Submission:

Upload the **Zipped Solution** from the Unity game to the space on Learn Online. It is your responsibility to delete the temporary folders from this before upload.

Game environment:

Set up a bounding area for the cats and squirrel to exist within.

There are two cats which follow different predefined paths. Each cat has its own bed where it rests.

There are a configurable number of trees with nuts for the squirrel to collect. These provide the squirrels with energy and sustenance which they bring back to their nest to store.

There are four squirrels in the game collecting nuts along configurable paths.

The squirrel nest can be placed anywhere (random location on startup is acceptable) on the screen to begin.

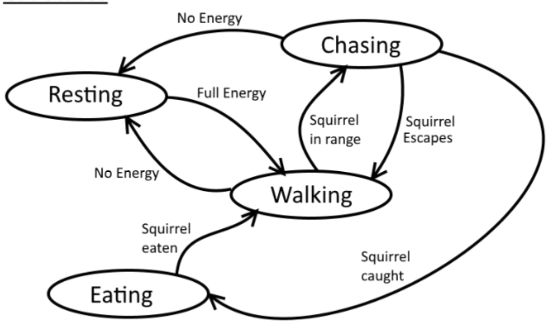
Chase Controller:

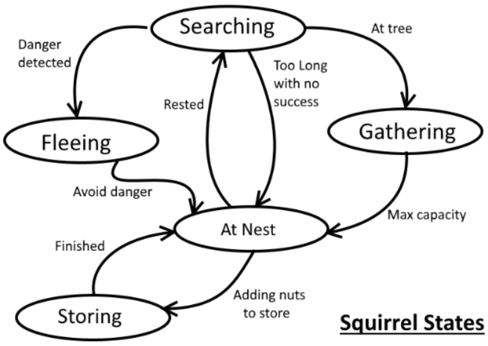
You need to create a singleton chase controller that will implement the automatic chase/flee scenario between the cat and a squirrel once they are within range. Cat and squirrel sprites will take turns to "move" during the chase. These moves use twice as much energy as usual. Squirrels will head straight to home and the cat will chase. If the squirrel reaches the nest first, then the cat must rest to regain energy.

The chase controller should use a simple GUI to show what happened ("squirrel escapes", "squirrel gets eaten").

During the chase, visual feedback should be given (for example, sprite is the green initially, yellow when energy is less than 60% and red when energy is less than 30%) on the energy levels of the cat or the squirrel.

Cat States





Squirrel States

Cats: Represented by a triangle sprite on the screen.

The cat has an energy value that can be set at the start.

Cat FSM

As the states change, the colour of the triangle should change to indicate this.

Starts in the resting state. Resting restores energy at a rate of 0.3 units per update. If a squirrel passes while resting, the cat can start running in pursuit.

Walking uses 0.4 units of energy per update so the cat must rest when energy gets too low. It can rest where it stops.

Chasing uses 0.8 units of energy per update.

If the cat catches a squirrel, then full energy is restored immediately.

When a squirrel is in range (you decide the range), “chasing” is automatic with no state changes until completion.

Squirrel: Represented by a circle sprite on the screen.

The squirrel has an energy value and payload capacity for nuts that are all configurable for testing.

Squirrel FSM Rules

As the states change, the colour of the circle should change to indicate this.

Starts in the “at nest” state and searches along a predefined path of points until it finds a tree or gets within chasing range of the cat (you decide the range).

Searching uses 0.1 unit of energy per update. If the energy value goes too low (you decide the threshold value), then the squirrel needs to return to the nest to restore energy at the rate of 0.2 unit per update.

In the fleeing state, the squirrel should head straight for the nest, but uses 0.3 units of energy per update while fleeing. If the squirrel goes out of range, then the cat starts patrolling again.

If the squirrel gets caught during a chase, then it is destroyed (eaten).

When the squirrel returns to the nest, it stores the nuts it has gathered at a rate of 0.2 units per update. The squirrel will have to wait (rest) if there are any other squirrels adding to the store of nuts or until energy is restored.

Marking Scheme:

- Create the Game Environment as specified – 15%
- Chase Controller implemented as specified – 30%
- Base Finite State Machines implemented as specified – 35%
- Cat fully implemented as specified – 10%
- Squirrel fully implemented as specified – 10%