Assessment –January 8th, 2021 before 1pm

Required:

Create a new project and implement the following states for the birds and bees simulation in a scene in Unity. You need to write the state machine for this and add it to both the bird objects and the bee 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 exported assets from the Unity game to the space on Learn Online. You may also zip

It is your responsibility to ensure the assets are exported correctly from the game and uploaded within the allotted time.

Game environment:

Set up a bounding area for the birds and bees to exist within.

There are two birds which follow different predefined paths. Each bird has a nest where it rests.

There are a configurable number of flowers in the room. They provide the bee with nectar to bring back to the hive.

The four bees fly around the game collecting honey along configurable paths.

The hive can be placed anywhere on the screen to begin.

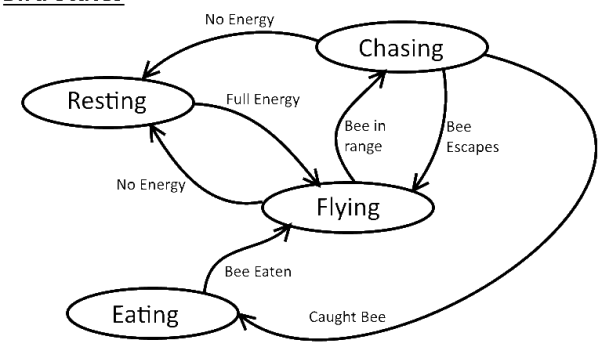
Chase Controller:

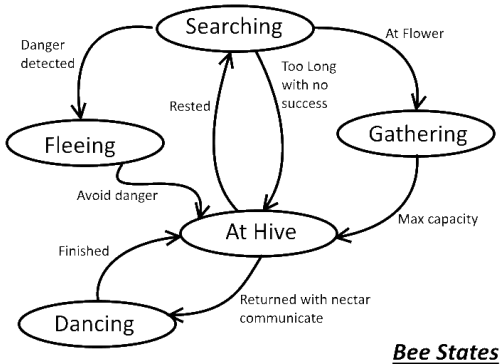
You also need to create a singleton chase controller that will implement the automatic chase/flee scenario between the birds and the bees once they are within range. Bird and bee sprites will take turns to “move”. The moves use twice as much energy as usual. Bees will head straight to home and the bird will chase. If the bee reaches the hive first, then the bird must rest to regain energy.

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

During the chase, a 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 bird and bee.

Bird States





Bee States

Birds: Represented by a triangle sprite on the screen.

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

Bird 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 bee passes while resting, the bird can start flying in order to pursue.

Flying uses 0.4 unit of energy per update so the bird must rest when energy gets too low.

Chasing uses 0.8 units of energy per update.

If the bird catches a bee, then full energy is restored immediately.

When a bee is in range (you decide the range), "chase" is automatic with no state changes until completion.

Bee: Represented by a circle sprite on the screen.

The bee has an energy value and payload value for nectar that are all configurable for testing.

Bee FSM Rules

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

Starts in the "at hive" state and searches along a predefined path of points until it gets within chasing range (you decide the range).

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

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

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

When the bee returns to the hive with nectar, it "dances" for other bees until the nectar is unloaded (0.2 units per update) and then rests until energy is restored and flight begins again.

Marking Scheme:

- Create the Game Environment as specified – 10%
- Chase Controller implemented as specified – 20%
- Finite State Machines implemented as specified – 40%
- Bird fully implemented as specified – 15%
- Bee fully implemented as specified – 15%