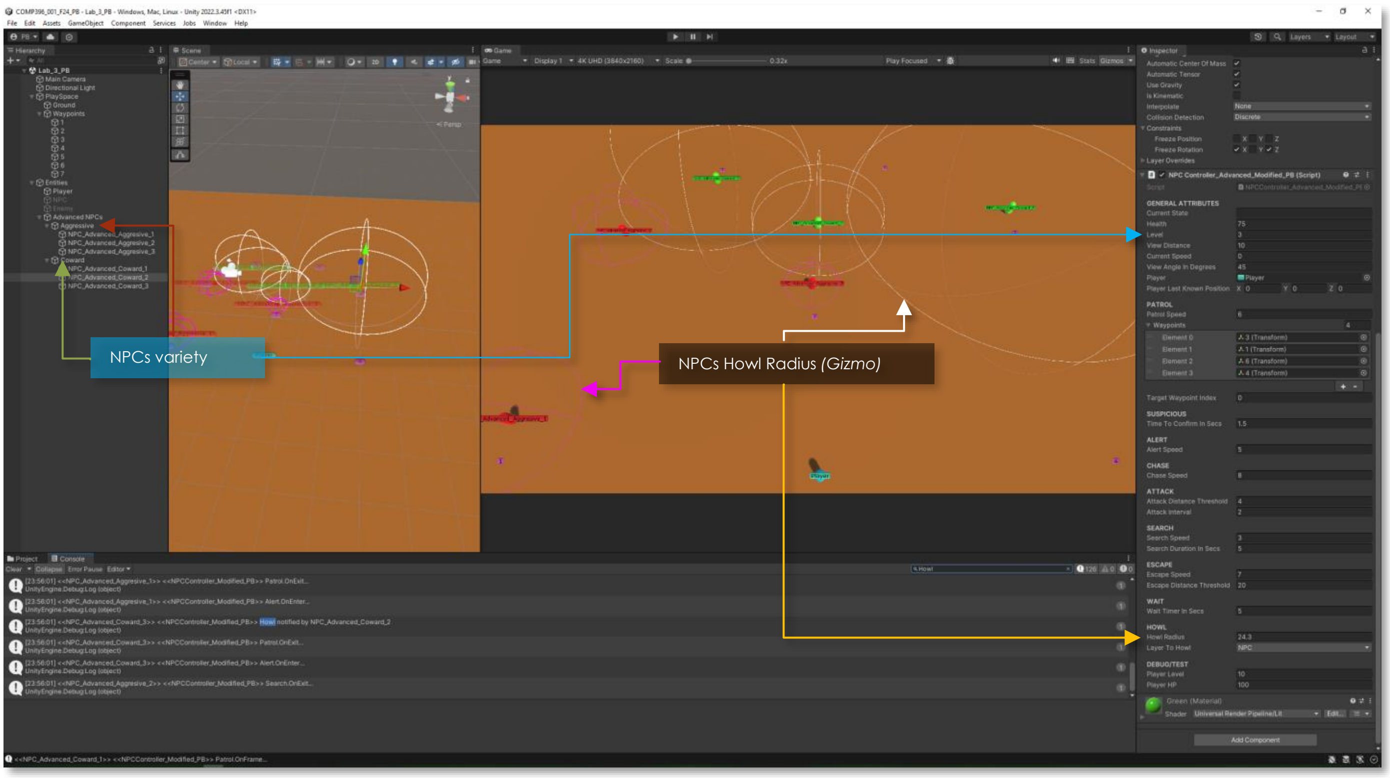
LAB – 3

ADVANCED FSM - 1
(Factory Design Pattern)

CHANGES FROM LAB – 2:

- Added State (More detail in Flowchart):
  - HOWL (Now total 9 states)
    - NPC, when it becomes suspicious has a HOWL (*notify*) radius to other NPCs.
    - Player's last known position is transmitted with *OnHowlNotified*
    - The rest of the NPCs act independently depending on their attributes.
- Total States Now:
  - *PATROL*
  - *SUSPICIOUS*
  - *ALERT*
  - *CHASE*
  - *ATTACK*
  - *SEARCH*
  - *ESCAPE*
  - *WAIT*
  - *HOWL*
- Now has multiple NPCs in scene with Coward and Aggressive NPC generalization. Attached video and Script.



```
[23:56:01] <<NPC_Advanced_Aggresive_1>> <<NPCController_Modified_PB>> Patrol.OnExit...
UnityEngine.Debug:Log (object)
[23:56:01] <<NPC_Advanced_Aggresive_1>> <<NPCController_Modified_PB>> Alert.OnEnter...
UnityEngine.Debug:Log (object)
[23:56:01] <<NPC_Advanced_Coward_3>> <<NPCController_Modified_PB>> Howl notified by NPC_Advanced_Coward_2
UnityEngine.Debug:Log (object)
[23:56:01] <<NPC_Advanced_Coward_3>> <<NPCController_Modified_PB>> Patrol.OnExit...
UnityEngine.Debug:Log (object)
[23:56:01] <<NPC_Advanced_Coward_3>> <<NPCController_Modified_PB>> Alert.OnEnter...
UnityEngine.Debug:Log (object)
[23:56:01] <<NPC_Advanced_Aggresive_2>> <<NPCController_Modified_PB>> Search.OnExit...
UnityEngine.Debug:Log (object)

<<NPC_Advanced_Coward_1>> <<NPCController_Modified_PB>> Patrol.OnFrame...
```

```
NPC Controller_Advanced_Modified_PB (Script)
GENERAL ATTRIBUTES
Current State
Health                      75
Level                       3
View Distance               10
Current Speed               0
View Angle In Degrees       45
Player                      Player
Player Last Known Position  X 0   Y 0   Z 0
PATROL
Patrol Speed                6
Waypoints                   4
  Element 0                 3 (Transform)
  Element 1                 1 (Transform)
  Element 2                 6 (Transform)
  Element 3                 4 (Transform)
Target Waypoint Index       0
SUSPICIOUS
Time To Confirm In Secs     1.5
ALERT
Alert Speed                 5
CHASE
Chase Speed                 8
ATTACK
Attack Distance Threshold   4
Attack Interval             2
SEARCH
Search Speed                3
Search Duration In Secs     5
ESCAPE
Escape Speed                7
Escape Distance Threshold   20
WAIT
Wait Timer In Secs          5
HOWL
Howl Radius                 24.3
Layer To Howl               NPC
DEBUG/TEST
Player Level                10
Player HP                   100
```

Fig 1.1: Screenshot showcasing multiple NPCs with different attributes and behaviours.

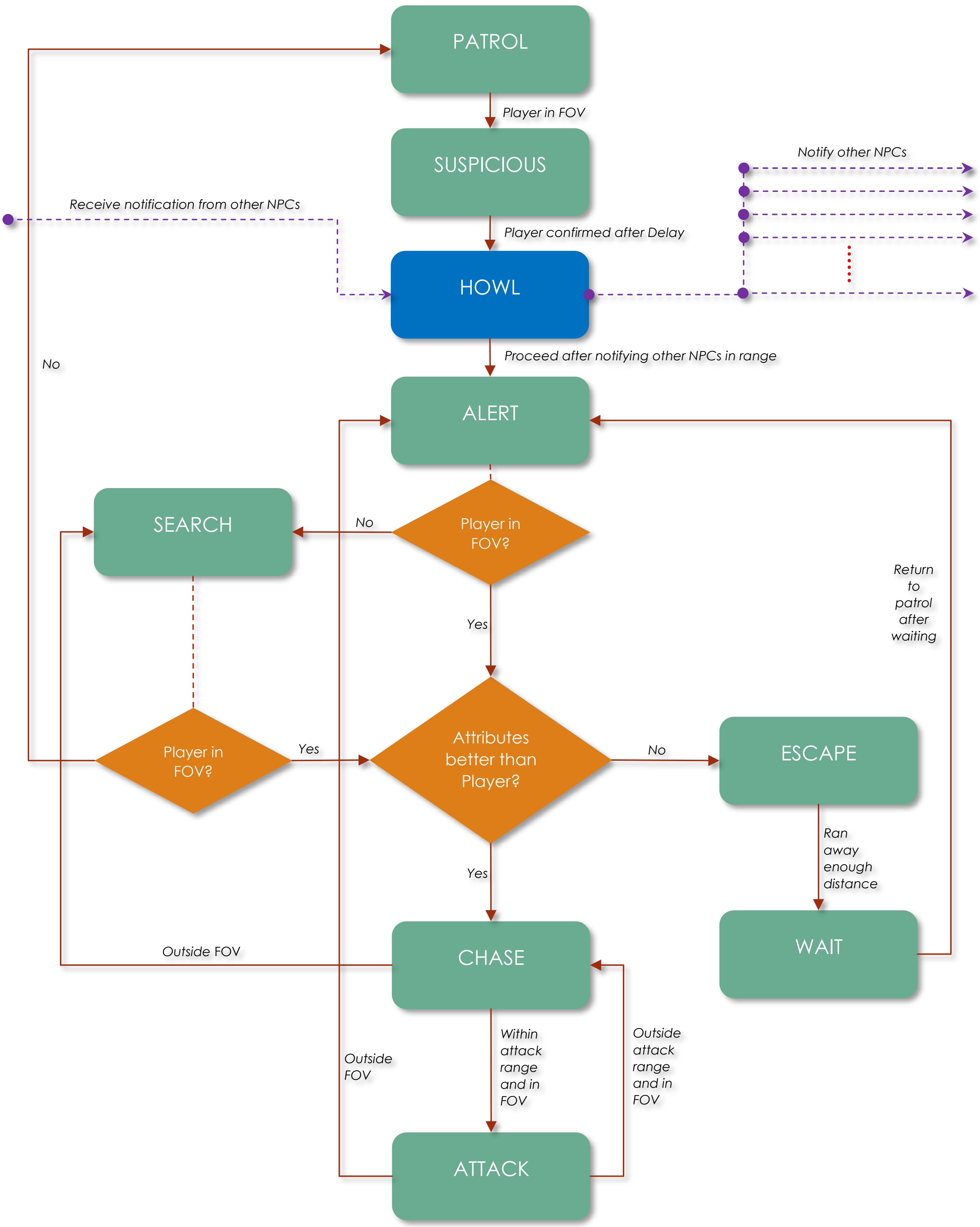

PATROL
Player in FOV
SUSPICIOUS
Notify other NPCs
Receive notification from other NPCs
Player confirmed after Delay
HOWL
Proceed after notifying other NPCs in range
No
ALERT
SEARCH
No
Player in FOV?
Return to patrol after waiting
Yes
Player in FOV?
Yes
Attributes better than Player?
No
ESCAPE
Ran away enough distance
Yes
WAIT
Outside FOV
CHASE
Outside FOV
Within attack range and in FOV
Outside attack range and in FOV
ATTACK

Fig 1.2: Flowchart showing the states now (Added HOWL).

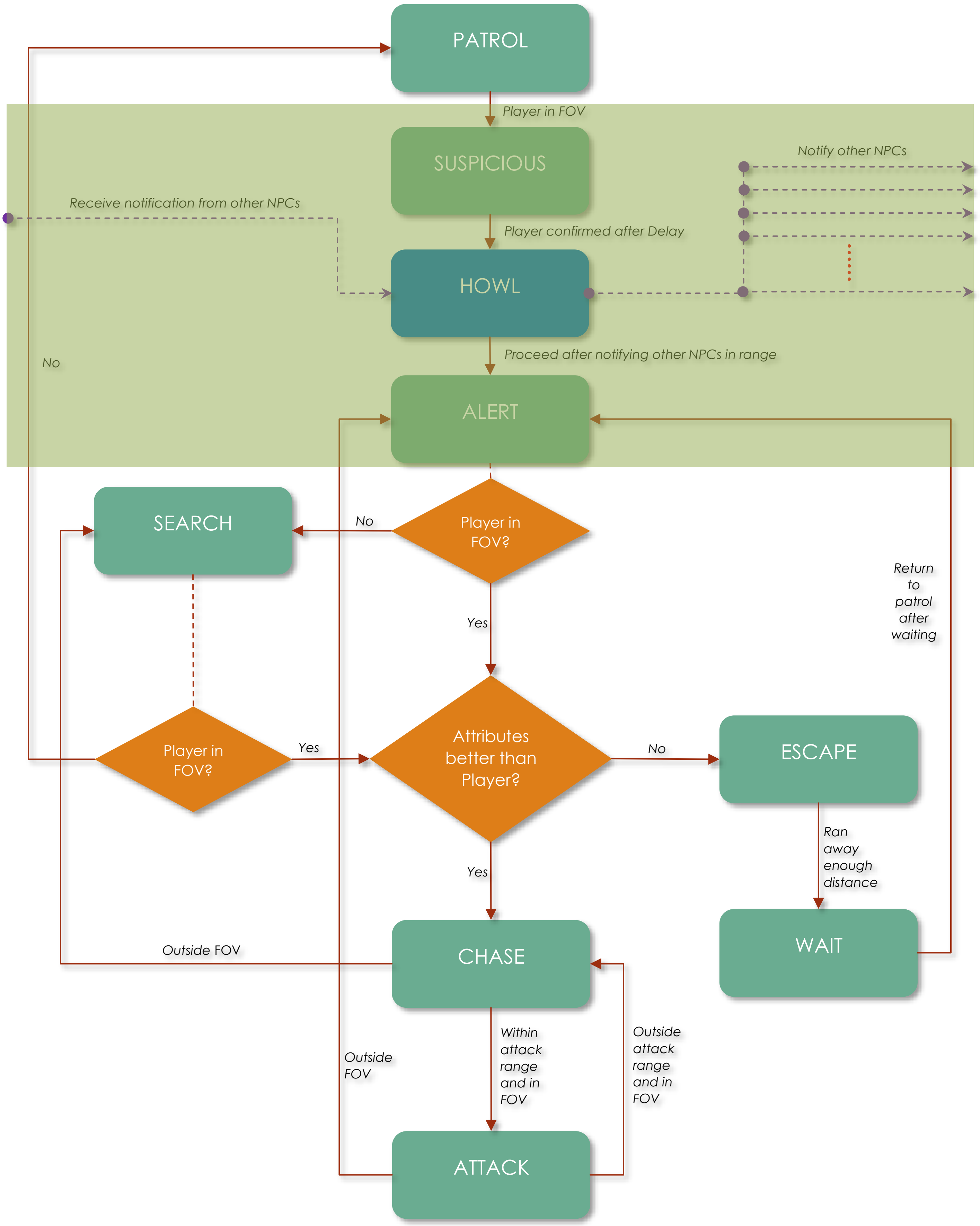

PATROL
Player in FOV
SUSPICIOUS
Notify other NPCs
Receive notification from other NPCs
Player confirmed after Delay
HOWL
Proceed after notifying other NPCs in range
No
ALERT
SEARCH
No
Player in FOV?
Yes
Return to patrol after waiting
Player in FOV?
Yes
Attributes better than Player?
No
ESCAPE
Ran away enough distance
Yes
WAIT
Outside FOV
CHASE
Within attack range and in FOV
Outside attack range and in FOV
Outside FOV
ATTACK

Fig 1.3: Flowchart showing the states affected by adding Howl (Highlighted in green)

- Changes needed for accommodation of extra state:
    - *added some logic to preceding and succeeding state*
    - *Added a state check in the newly added state*
    - *More attributes needed*
    - *More interdependent code*
    - *Since now every state has OnEnter, OnFrame and OnExit*
        - *Setting of certain attributes can be done exclusively like speed in OnEnter*

- Dependency of changes on number of changes
    - *The new state slotted just in between two states and required minor adjustments on OnFrame of both the preceding and succeeding state*
    - *Since, the transition is one way it was one way simple and less dependent on others*

- Nature of changes when states are removed:
    - *Certain states are more complex than others but since each state is now its own object, it is simpler to remove than simply switch/if*
    - *For complex changes that handle different attributes it will affect maybe all of the FSM else at least most of FSM*
    - *Simple logic states just need their transition (OnTransition to be replaced)*

- Predict dependency on the states you start with and number of transitions from/to the state to be removed
    - *Complex states like* ***ALERT*** *with 3 transitions will require some rewrite of the code since a lot of references will be giving error since it has state checking and other things*
    - *Simple states like* ***SUSPICIOUS*** *with 1 state will only require its state set from previous state to transition directly to its next set. (Think of deleting a node in a single linked list)*

- Sustainability of the method if frequent changes are expected
    - *This method is more sustainable than Switch/ENUM due to the fact that the logic is self-contained mostly and each state has its own object to handle making it easier to organize in code*
    - *Each state has its own OnEnter, OnExit and OnFrame. This helps to set certain attributes which used to bleed into different states like speed in the method we learnt in Lab 2.*
        - *For example:*
            - *In Lab 2, patrol speed was set by Alert state during transition to avoid making it getting set every frame in Patrol state*
            - *In Lab 3, because of OnEnter, the speed is set in OnEnter of Patrol instead of some other state making it more organized and easier to debug and understand.*
    - *Although it is more sustainable, it is still far from modular since every state is in a single big class and we still need to do some manual work which can be automated using different patterns and techniques*