**Overview.**

**Components:**

* **Ability:** Allows an Entity cast abilities. Entity abilities.
* **AIAction:** Scriptable object used by a state. Makes an entity execute a specific action or process.
* **AIDecision:** Scriptable object used by a transition to check a condition that returns a Boolean.
* **AIState:** Scriptable object the contains actions and transitions. Every time the state machine updates the state it executes first all the action/s and the checks the transition/s.
* **AIStateMachine:** Finite state machine used by the enemies to cycle through states. The CurrentState update process is executed by a fixed number of times each second.
* **AITransition:** Used by the state to check any possible state change/s for that state. Contains a decision, depending on the output of the decision it tells the state machine to change to a true or false state.
* **Enemy:** Implementation for all the possible entities that are enemies to the player. Forces the engine to add AIStateMachine, EnemyMovement and a Targetter as basic components for any enemy.
* **EnemyMovement:** The main component for the enemy movement. Uses a NavMeshAgent to navigate through the NavMeshSurface. Each NavMeshAgent has its own type, if needed, and each agent type uses its own NavMeshSurface for the navigation to towards the destination. Implements the EntityComponent class.
* **Entity:** Main component for every object that moves or executes actions in the game.
* **EntityComponent:** Main structure for all the components that can be added to any entity, ex: Movement, StateMachine, Abilities, etc.
* **EventManager:** Static class that manages events in the game. Uses EventHandler<> interface to communicate between the manager and the objects that receive events. Allows objects to Subscribe and UnSubscribe to events in the OnEnable and OnDisable callbacks.
* **Singleton:** Singleton design pattern used by any objects that is unique in the game. Used by polymorphism to make specific implementation.
* **SlotManager:** Used to control the number of enemies that can attack a certain object. Contains Waiting and Attacking slots. Allows a user to reserve a free slot, first it checks for available AttackingSlot/s if there is no available continues with the WaitingSlot/s. Returns null if there is no slot available.
* **Targetter:** Component in charge of the detecting trigger events and adding the corresponding target to the NearTargets list. Also update the CurrentTarget by a fixed number of times each second, according to the minimum distance to the target.

**Enemy:**

The Enemy class is an implementation of the Entity class. The Enemy must have a EnemyMovement component so it can navigate trough the scene, also the Targetter components allows it to target near Entities that come in within the Trigger range and its own AIStateMachine.

* The AIStateMachine updates inside a routine contained in a variable, for tracking purposes and control. The CurrentState update occurs a fixed number of times each second.

The Enemy also contains a NextAbility that can be set from other LookForAbilityCast decision. This then can be used by the CastNextAbility action to make the Enemy cast that Ability. This is in a scenario where we want the Enemy to cast any of its possible abilities in no order.