

State Machine:

Description:

StateMachine system works based on *StateMachineContext*. It means each object just has one *StateMachine* component but actually the object is able to have different state machines (Even each state can have a state machine too) because each state machine information is stored in one *StateMachineContext*.

For instance, we have one *StateMachine* component on each character but they have 3 different state machines: 1. Main 2. Secondary Movement 3. Combat and they work parallelly.

Each state machine has some states which you can check them in [CharacterStateMachine doc](#).

How does it work?

Each State has *Enter*, *Update*, *FixedUpdate*, *LateUpdate* and *Exit* which are controlled by *StateMachine*.

Each *StateMachineContext* has some information. Like it is for which object, what the start state is, what the current state is and so on. Also states instances are stored here. You can put other common variables of the states in a context.

StateMachine has some functions, they are:

- *Start*: It should be called in *Monobehaviour.Start* of the object which have *StateMachine* component. It is used for starting the state machine with *StartState* that is determined by programmer.
- *Update*: It should be called in *Monobehavior.Update* of the object which have *StateMachine* component. It is *Update* loop in the state machine.
- *FixedUpdate*: It should be called in *Monobehavior.FixedUpdate* of the object which have *StateMachine* component. It is *FixedUpdate* loop in the state machine.
- *LateUpdate*: It should be called in *Monobehavior.LateUpdate* of the object which have *StateMachine* component. It is *LateUpdate* loop in the state machine.

- *ChangeState*: For switching between states you should use this function.
- *Reset*: Changes state machine state to *StartState*.

How to use it?

Context classes should inherit *BaseStateMachineContext* class.

State classes should inherit *BaseState* generic class.

So, we can call each group of functions for each *StateMachineContext* to have multiple state machines and work parallelly or serially or as a nested state machine.