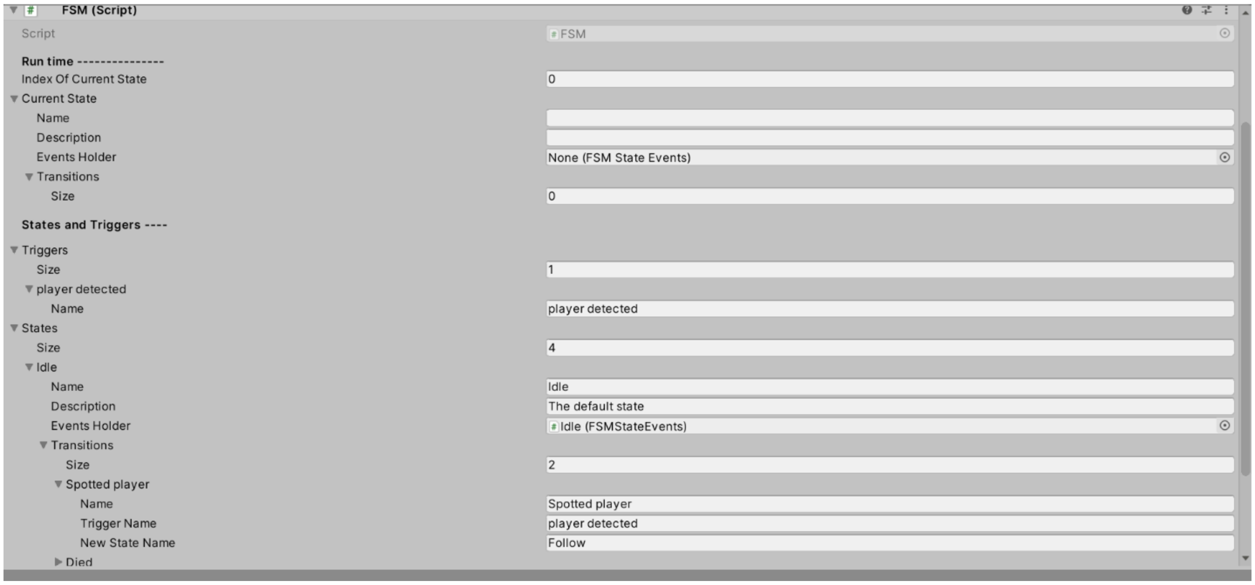
Author: Digx7

FSM (Script)

This is the main brains of a generic FSM (Finite State Machine). What FSM is for is merely to keep track of a variety of possible states that a given object can be in and determine when and what states it can transition into. This is very similar to the Animator window in unity, however, that one is built solely around animations. This FSM was built for AI originally but should be flexible enough to work with anything. *Please note that this FSM is still very much a work in progress, if any part of this documentation is out of date please let me know*



```
FSM (Script)
Script                          FSM
Run time ---------------
Index Of Current State          0
▼ Current State
    Name
    Description
    Events Holder               None (FSM State Events)
  ▼ Transitions
        Size                    0
States and Triggers ----
▼ Triggers
    Size                        1
  ▼ player detected
        Name                    player detected
▼ States
    Size                        4
  ▼ Idle
        Name                    Idle
        Description             The default state
        Events Holder           Idle (FSMStateEvents)
      ▼ Transitions
            Size                2
          ▼ Spotted player
                Name            Spotted player
                Trigger Name    player detected
                New State Name  Follow
          ▶ Died
```

Properties

PROPERTY:	FUNCTION:
RUN TIME	All of these items are altered and updated at runtime (during play mode).
INDEX OF CURRENT STATE	This is the number of the current state that is active. This can be altered to set what state will be the default state when entering play mode
CURRENT STATE	This will update at runtime to show all the details of the currently active state. This is only here for debugging purposes. Its properties are the same as the states down below.

STATES AND TRIGGERS	This is the main brains of the FSM and is where all of the main elements are
TRIGGERS	This is a drop-down list of all the triggers
SIZE	Adjust this number to determine how many triggers there are in this list.
TRIGGER ELEMENT	This is a single trigger in this list
NAME	This is the name of this trigger
STATES	This is a drop-down list of all the states
SIZE	Adjust this number to determine how many states there are in this list.
STATE ELEMENT	This is a single state in this list
NAME	This is the name of this state
DESCRIPTION	This is the description of this state
EVENTS HOLDER	This is a reference to a FSMStateEvents script
TRANSITIONS	This is a drop down list of all the transitions
SIZE	Adjust this number to determine how many transitions there are in this list
TRANSITIONS ELEMENT	This is a single transition in this list
NAME	This is the name of this transition
TRIGGER NAME	This is the name of the trigger that activates this transition
NEW STATE NAME	This is the name of the state that this transition goes to if triggered

Details

Triggers

Triggers are how the FSM knows when to transition from one state to the other. They only have one element, their name, which is case sensitive. This name is used by outside code to call the function in the FSM activateTriggerWithName(string name). When that function is called if the current state has any transitions using the given trigger it will transition to the new state.

States

A state is a single form, position, or configuration that the FSM can be in. For example, an enemy might have 4 states known as: Idle, Follow, Attack, and Dead. Only one State can be active at any given time and this can be seen under the Current State. In order to have a tangible effect on the rest of the game each state must be connected to a FSMStateEventHolder prefab under its Event Holder section.

This prefab has two events on it. A 'Start State' Event and a 'End State' Event. The 'Start State' Event is called when the given state is first transitioned to and the 'End State' Event is called when the given state is transitioned away from. These events are where you would connect anything that needs to happen on that given event.

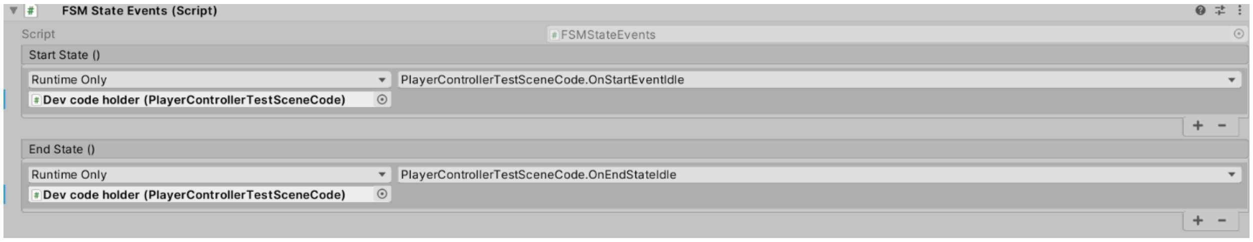
So for example: in the 'Start State' Event for the state 'Dead' on an enemy you could hook up the death animation, sfx, particle effects, and enemy death script.

Transitions

Each State has a list of transitions underneath it and are what control the transition of one state to another. Each transition has three name under it that are all case sensitive. First the 'Name' is the name of that given transition. Second is the 'Trigger Name' and is the name of the trigger that causes this transition. This name must perfectly match the name of one of the triggers listed above (spacing, punctuation, and capitalization matter). Lastly is the 'New State Name' and is the name of the state to transition to. This name also must perfectly match the name of one of the existing states (again spacing, punctuation, and capitalization matter).

FSMStateEventHolder (PreFab)

This is a small prefab with the FSMStateEvent script on it. This script holds the events for a single state in the FSM. *It is recommended that you rename each individual object to that of the state it is tied to for clarity.*



FSM State Events (Script)

Script: FSMStateEvents

Start State ()

Runtime Only | PlayerControllerTestSceneCode.OnStartEventIdle

Dev code holder (PlayerControllerTestSceneCode)

End State ()

Runtime Only | PlayerControllerTestSceneCode.OnEndStateIdle

Dev code holder (PlayerControllerTestSceneCode)

Author: Digx7

Properties

Property:	Function:
Start State	This event is called when the state attached to it is first transitioned to.
End State	This event is called when the state attached to it is transitioned away from.