Module 6

Animation State Machines

ANIMATION STATE MACHINE

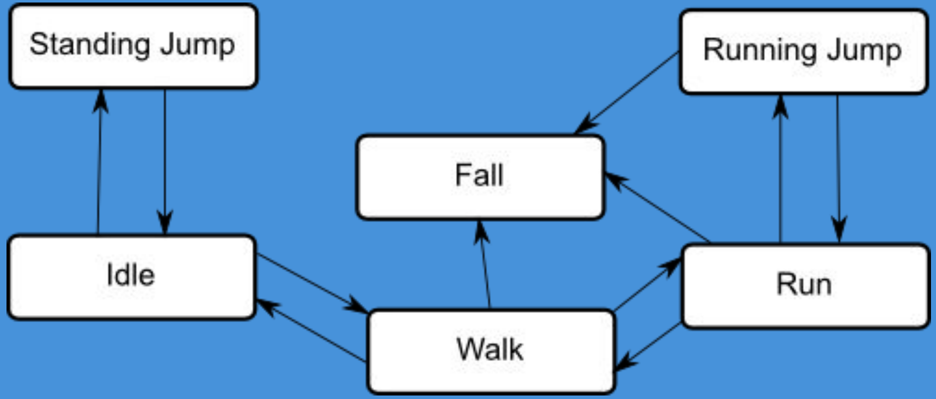
Animations in Games

- GameObjects often have multiple animations corresponding to different actions they can perform
  - Fighter has a swing attack, walking animation, and a jump animation
  - Game character's idle animation for standing and transition animation into walking animation
- Complex transitional logic
  - Can be hard to code
  - Need to do it for every gameobject with multiple animations
  - Can make the code VERY messy
- State Machine
  - States store animations
  - Only code transitions
  - Abstract away all of the gameobjects animations

Jump attack
Pull up
Roll
Run2
Sit down
Walk2

A State Machine for a Platformer

- State are the actions a player can take
  - Idle (do nothing)
  - Walk
  - Run
  - Jump: Standing and Running
- Player cannot transition from any state to any state
  - Can only running jump if already running.
- Whatever state the state machine is in is the animation that should be playing


$$\begin{tikzcd}
\text{Standing Jump} \arrow[d, shift left] & & \text{Running Jump} \arrow[dl] \arrow[d, shift left] \\
\text{Idle} \arrow[u, shift left] \arrow[dr, shift left] & \text{Fall} & \text{Run} \arrow[u, shift left] \arrow[l] \arrow[dl, shift left] \\
& \text{Walk} \arrow[ul, shift left] \arrow[u] \arrow[ur, shift left] &
\end{tikzcd}$$

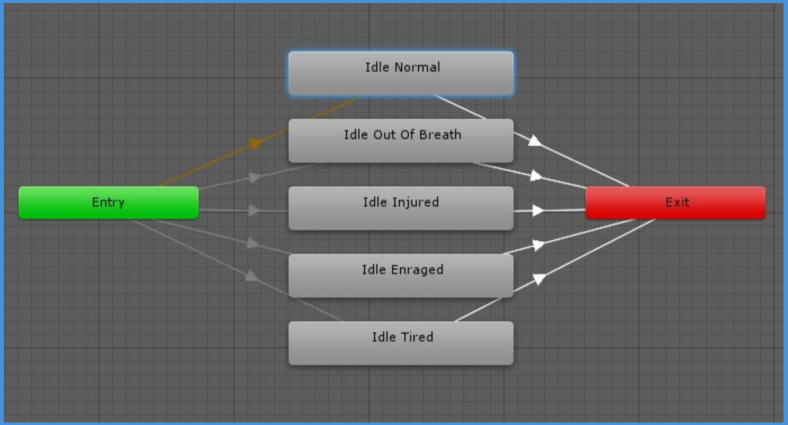
States and Transitions

- States represent a single animation
  - Walking state plays a walking animation
- State parameters
  - Allow game developer to execute code when animation starts, stops, or during.
  - How fast should the animation be playing?
  - Should the animation transition to another state once it is finished?
- States can have associated code (like for collision components)
  - Access some additional information in the game to determine what animation should be playing
    - Walk right or walk left animation based on the players direction

- Transitions are how / when the state machine should go from one state to another
- Some animations should only play once and should go to the next state when finished.
  - Death animation, Jump animation, Attack animation
- Conditional transitions check some additional information in the game
  - Pressing space runs the shooting animation but, if the player is out of ammo run the “uh oh” animation

States and transitions can be exposed through an interface to the rest of your game code allowing other parts of the game to induce a transition, set a state, or change parameters

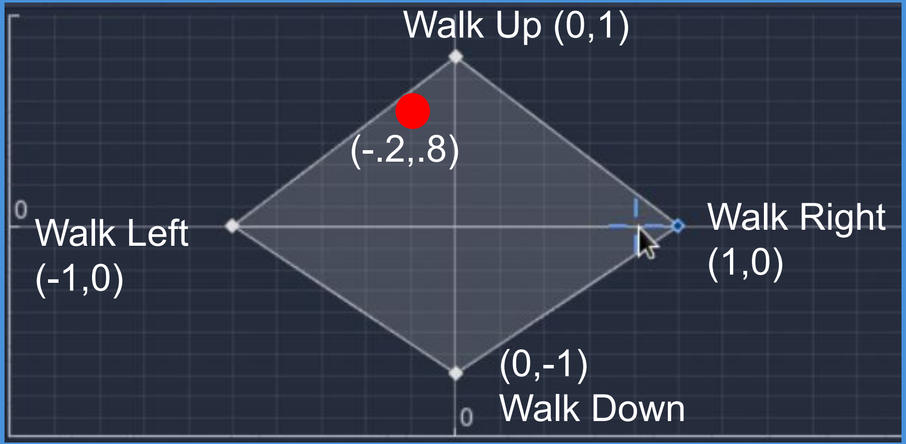
Sub-State Machines



- Sometimes a state machine can get very complicated
  - Lets package parts of it up and name them
- Sub-State Machine
  - Still a state machine
  - Add 2 special nodes to use it as a state in a parent state machine
- Need to transition into a sub-state machine and transition back out
  - Entry Node
    - Initial state when sub-state machine is entered
    - Immediately transitions to default state
    - Stores internal conditions to determine what the default state should be
  - Exit Node
    - When sub-state machine transitions to this node it returns to the parent state machine
- Sub-state machine is a separate and complete state machine
- State transitions can be from state to state, state to sub-state machine, and even sub-state machine to sub-state machine.

Blend Space

- We want to be able to automatically switch between animations based on the value of some variable
- Give each animation a coordinate
    - 2D space in this example but really can be any
- Based on the value of a variable choose the closest animation
- Essentially, another layer of abstraction where a single state in the animation state machine can hold a set of animations that are being blended together



Example:

Four animations for walking up, right, down, and left

Red dot is the current player direction (-0.2, 0.8).

The Walk Up animation is closest so we should use the walk up animation.

This is just the tip of the iceberg! Much more here: https://docs.godotengine.org/en/stable/tutorials/animation/animation_tree.html#blendspace2d

Additional Resources

Unity Manual: https://docs.unity3d.com/Manual/AnimationStateMachines.html

Unreal Manual:
https://docs.unrealengine.com/4.27/en-US/AnimatingObjects/SkeletalMeshAnimation/StateMachines/

Rive Tutorial: https://help.rive.app/editor/state-machine