

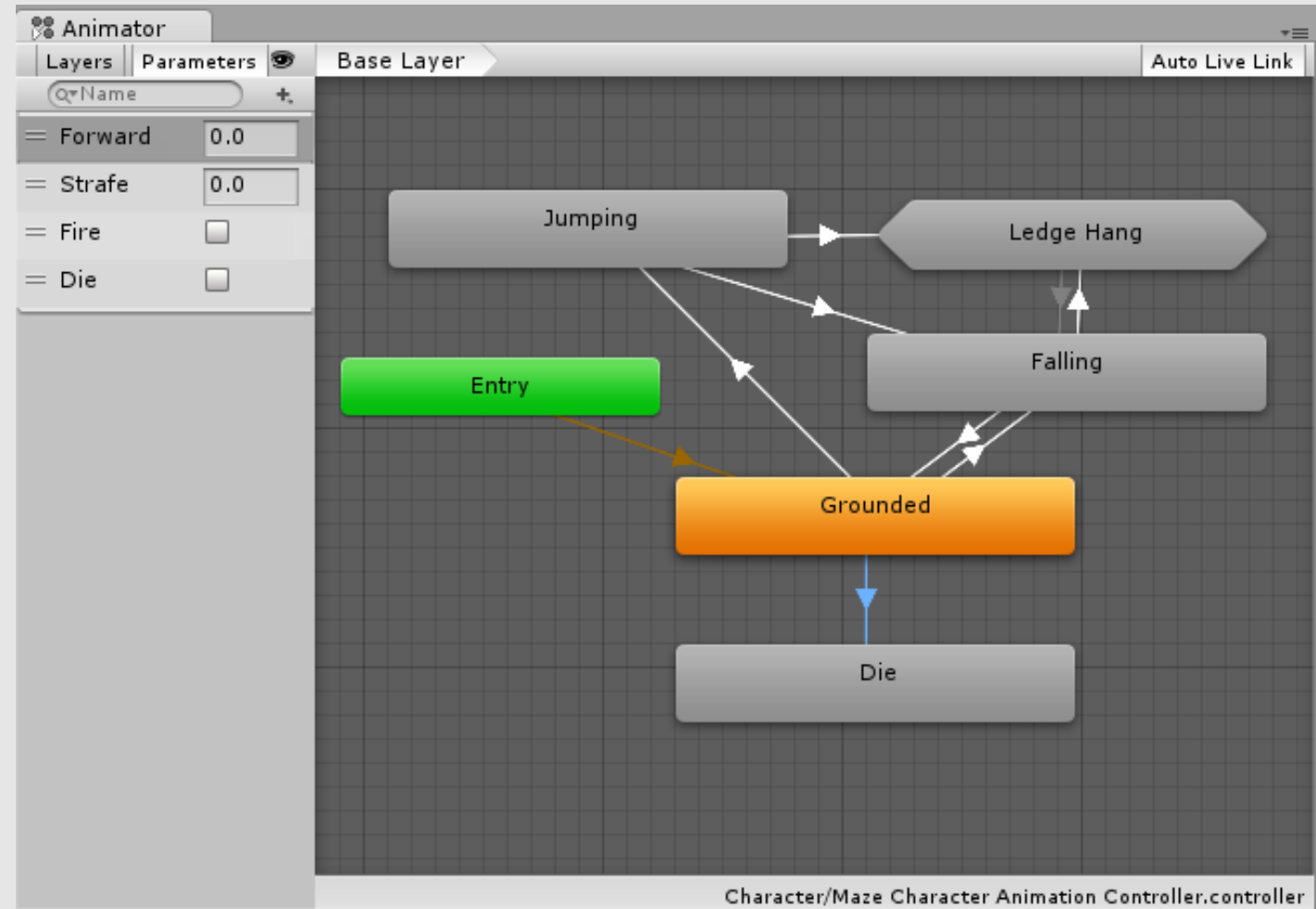
Game Dev with Unity3D

ANIMATOR



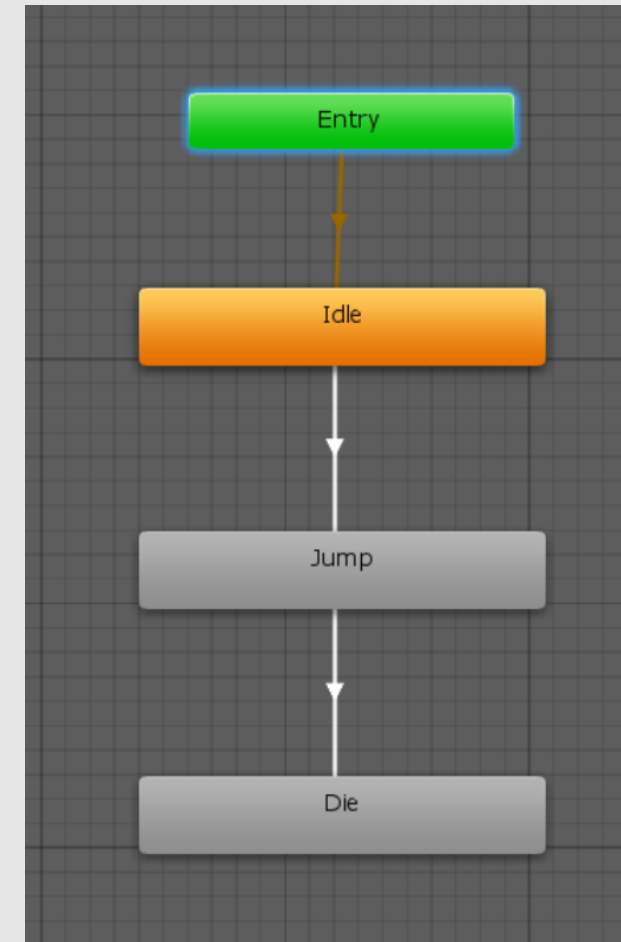
Moving Between Animations

- **Unity Animator** is the engine that runs and controls all animations in an object
- We speak with the Animator in order to control which animation should play



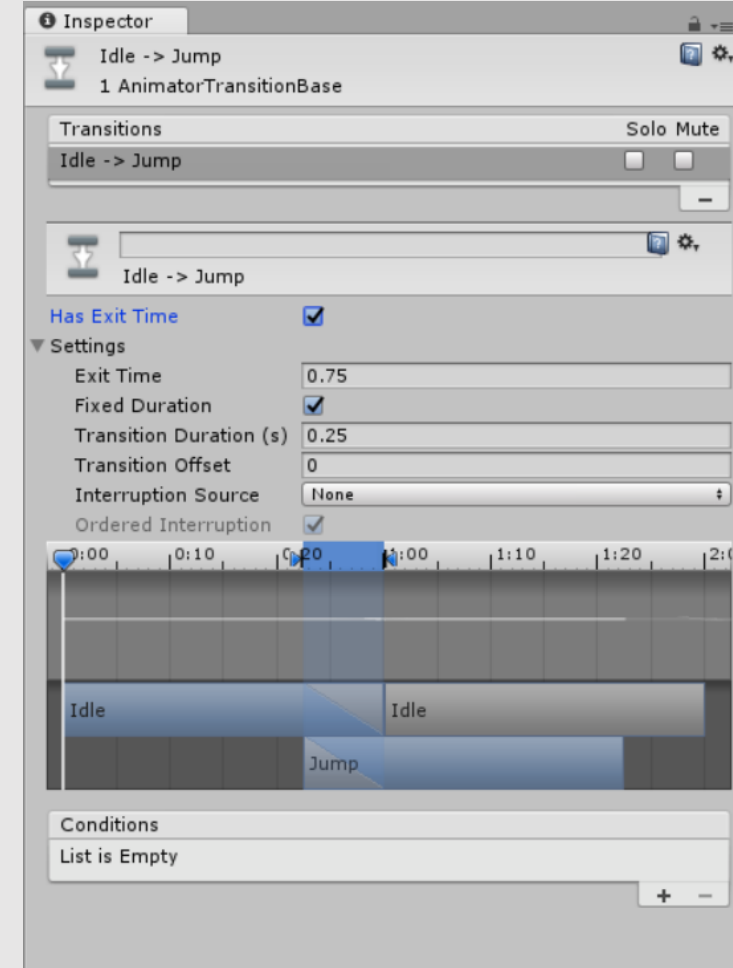
Moving Between Animations

- **Animator States** represent each state of the element
- Each state is connected to a single animation clip
- The states are the grey rectangles and the orange one
 - The orange one is the default. It will be the first to run
 - You can only have one default and many grey ones (normal states)



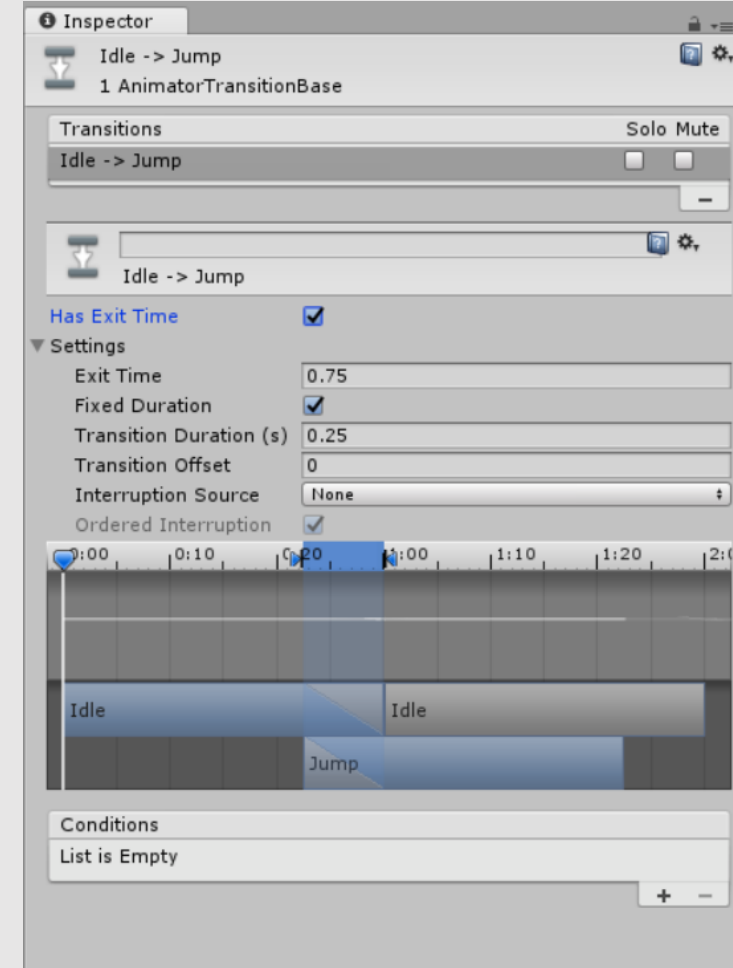
Moving Between Animations

- **Transitions** explain HOW to move from one state to another.
 - Do you want the animation to play once and then move back to an idle state?
 - Do you want the animation to stop immediately and play a Jump animation?



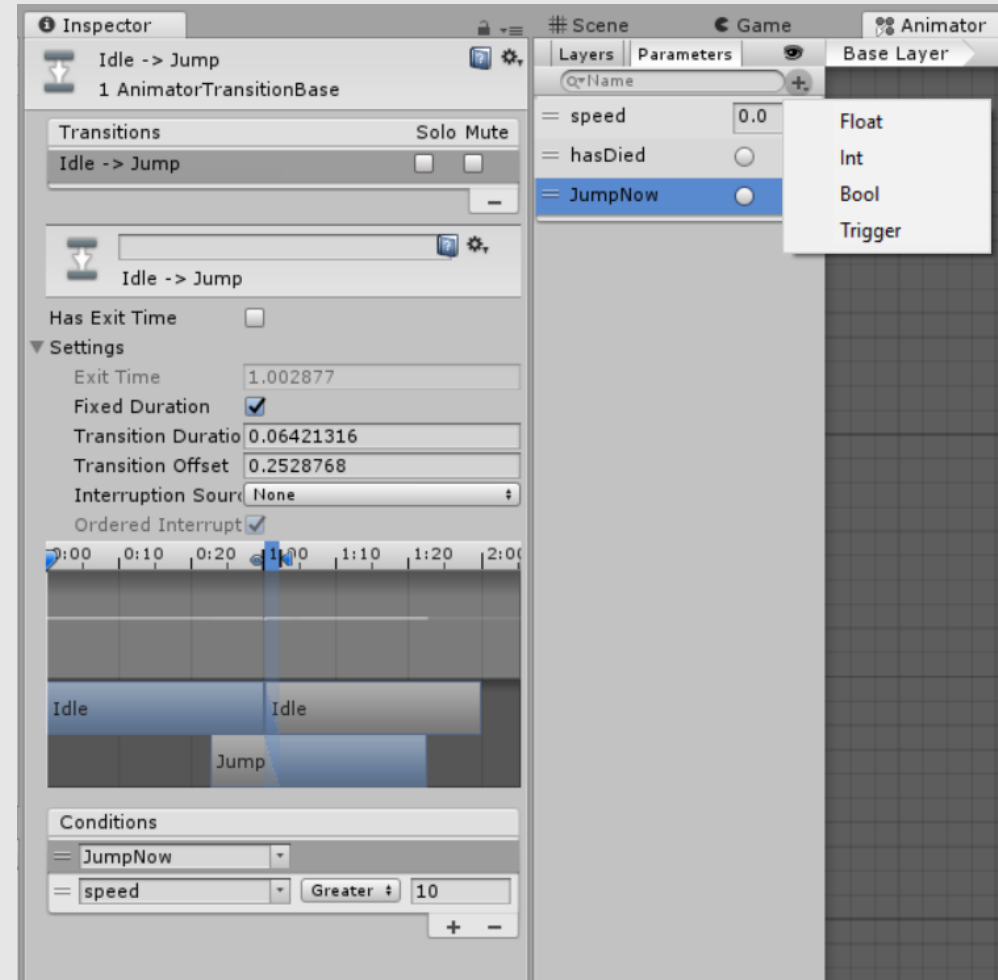
Moving Between Animations

- **Transition** most used features
- Has Exit Time: do you want the animation to finish playing before leaving it
- Transition Duration: length of duration
- Conditions: parameters that are used to trigger the transition to occur



Moving Between Animations

- Defining Parameters is made from the dedicated menu bar.
- Trigger is used for Boolean activated events. Once a trigger is activated, it is automatically being deactivated.



Questions and Exercises

(if you can answer this without testing, good job. otherwise, test it)

- Make a starship that has 2 animations:
 - One idle floating in place
 - Another that influenced by movement
 - Hint: this involves the use of 2 Animators on 2 GameObjects
- Look back on the first tutorial projects we looked over (Survival Shooter or Tanks). Find their Animators – and investigate how they did.

Triggering Animation with a Parameter

```
private Animator _animator;  
  
public string TriggerName;  
  
void Update () {  
    _animator = GetComponent <Animator>();  
    if (Input.GetKeyDown (KeyCode.Space))  
    {  
        _animator.SetTrigger(TriggerName);  
    }  
}
```