

Weekly breakdown - Week 8

Weekly objective: Students will understand the basics of motion design and how to create animation by code and by using the animation system. Students will be able to control transitions of multiple animations on one prefab/game object by building an animator and setting the transitions between animations by code.

Goals:

Students will be able to:

- Create basic parameter changes as animation by code
- Create advanced parameter changes using the animation system
- Understanding features of the animation window, keys, frames, ticks, properties
- Understand advanced concepts of animation like dope sheet, curves, tangent handles
- Understand basics of motion design and building loopable animations
- Govern state changes between multiple animations using the animator component
- Understanding the relationship between the game object and the animator component
- Understanding the connection between game object components and the animation window
- Understand the relationship between states and animations, and states and transitions, transitions and parameters
- Send data to prefabs and animators from code bases and UI updates
- Organize project hierarchies as quantities and varieties of external asset files grows

Lecture Topics:

- Content creation: creating prefabs and animations
- Animators, animations, transitions, parameters
- Creating animation by code or by the animation system
- Defining animation as parameter changes over time
- Animation systems allow for easy animation creation for design decisions
- Event systems and conditions for animation playing and switching
- Using events and user input to change animation states in animator
- Using graphic changes ie animations to tell user what is happening in the system
- Change of visual information is used heavily in UIs to give feedback to user
- Creating game object hierarchies with ease of animation in mind
- Opening up animation conditions to let user change animation features
- Review of game object components needed for animating, renderer, animator, collider
- Project organization is important as amount of files grow, animations, shaders, prefabs
- Generating an easy to understand folder hierarchy in the project
- Using the create menu for creating animators and animations
- Understanding the relationship between the game object and the animator component
- Understanding the connection between game object components and the animation window

- Animator states, any state, entry, exit, and user created
- Understand the relationship between states and animations, and states and transitions, transitions and parameters
- Understanding idea that parameters are public variables that other scripts can change
- Quick methods for using the animator and animation windows
- Naming conventions for animators, animations, and parameters
- Name animators as a noun, animations as a verb, and parameters as a transition
- Adding properties in animation window and adding keys to change the values over time
- Understanding features of the animation window, keys, frames, ticks, properties
- Animation is changes of what rendered to the monitor
- Hertz or refresh rates are how often are the images changed on the monitor
- Common Hz ratings are 60, 75, 120 Hz or frames per second
- Animation is just a trick to the eye, switch images fast enough to see motion
- Relationship of keyframes to curve interpretation based on frame rate and curve style
- Ensure first frame values are the same as default values of game object
- Ensure last frame values are same as first frame values for loopable animations
- History of animation starting from Walt Disney and the 12 principles of animation
- Using methods like update and pingPong to animate with code
- Use animation system for quick iteration on design work and then convert over to code
- Access children's components via animation
- Design one object then spawning multiple objects creates emergent behavior
- Animation of parameters can break UI if trying to change the same parameter
- Scaling, copying and pasting, and moving keys in the animation window
- Dopesheet vs curves, linear interpolation vs smooth interpolation
- Cleaning up starting and ending keys for looping animations
- Screen space vs world space UI
- Converting UI from screen space to world space
- Standard shader vs unlit shader, when to interact with the lighting system and not