Weekly breakdown - Week 11

Weekly objective: Students will understand how to organize and present complicated UI interfaces, organize rendering orders for 2D and 3D graphics, animate objects with multi-threaded code, and refactor prefabs and code to fix bugs.

Goals:
Students will be able to:

- Understand rendering order when handling 2D sprites for UI and 3D graphics
- Understand design methods for building clear sections of UI
- Group parts of UI in the hierarchy for easier reorganization
- Build UIs that have both 2D and 3D elements mixed together
- Create animations from code, not using the animation system
- Use multithreaded code to control timing situations like animations
- Refactor prefabs and code to fix bugs in UI features
- Write code for prefabs to hold data about themselves

Lecture Topics:

- Rebuild UI background from 3D quad to UI image
- Create multiple UI images to seperate UI elements
- Position clock in front of UI since its global space render UI
- Position AM/PM button in front of clock
- Group parts of UI in hierarchy for easier reorganization
- Rebuild animation part of UI
- How hierarchy of UI elements impacts their rendering order
- Repurposing UI gameobject with rect transforms to organize UI elements
- Design methods for building clear sections of UI
- Rebuild random toggles in animation part
- Aligning text for better readability, rotating text for vertical alignment
- Ensuring everything is clickable after reorg, adjust raycast targets
- Adding colored sections with headers for clear UI
- Creating hide button and positioning it offset from main UI
- Make UIAnimation script, first full animation by code
- Coroutines as seperate threads or loops
- Calling coroutines from a regular method call
- Lerp method uses start pos, end pos, and amount of time to move
- Using Vector3 in rect transform just like regular transform
- Checking if application is running in editor or standalone
- Move paint strokes behind the UI and enlarge strokes to compensate
- Removed mousePosition.x check on paint stroke since painting behind UI
- Hotkeyed 4-6 for spawn object and 1-3 for animation since can hide UI

- Added float variable to animator to remember initial speed setting to make random speed changes work when changing the overall speed setting from 0x-2x
- Changed speed text from secs to X, pushed max from 2 to 3
- Standardized paint strokes sizes, adjusted sizes of prefabs to match each other
- Fixed cylinder prefab animation to fit with rest of prefabs
- Try to break UI for user testing
- Created prefabData script for animation and color information
- Go over storing initial animation speed to allow for adjusting speed while keeping all random speeds
- Go over storing initial color to allow for adjusting color info
- Updated color method to update emission as well
- Reset animation speed to initialSpeed * animationSpeed after switching animation type
- Removed isAnimTypeRandom check in ChangeAnimationState
- Fixed animation speed sections when spawning new paint strokes and changing speed slider
- Cleaned up some code so that turning off random anim changes all previous paint strokes, and changing during random doesnt
- Updated ChangeShape and doing check for random shape before switch statement, creating bool variable for randomShape, now random prefab type toggle works in UI