Weekly breakdown - Week 13

Weekly objective: Students will understand naming, commenting, organization and refactoring processes for clean, legible code, and be able to read through and understand all code generated in this class project.

Goals:
Students will be able to:

- Refactor code for legibility
- Understand clear naming conventions for variables and methods
- Understand the organization and commenting principles for code blocks
- Safety-proof arithmetic results using Mathf.Clamp
- Understand collection data structures like dictionaries
- Read through and understand all code generated in this class project

Lecture Topics:

- Refactoring code as its growing and getting hard to read
- Group together your public, and private variables and methods
- Rename variables and methods as their purposes become clearer
- Comment methods to give a quick description of their purpose
- Move generic global variables to the top
- Move global variables specific to a method above and outside that method
- Organize and comment blocks of code in methods
- Reusing UI elements to build new parts of UI
- Changing default settings of UI elements
- Checking if we should randomize the shape int before spawning an object
- Use UI dropdown element to build idea of collections from list and array to dictionaries which are a list of mappings of one value to another, usually a human legible value mapped to a machine legible value, ie strings to ints
- Explain the connection of UI toggle to bool in another script which is checked in an if statement in the update loop so if user clicks the mouse button then we check if we should randomize some parameters before object creation then we send information back to the UI to should the results after the randomization of a parameter
- Review all code blocks in the ClickPositionManager update loop to ensure understanding of code as user interacts with the mouse
- Explain how to incorporate randomness into the lifetime of objects
- Use Mathf.Clamp to provide safety checks if your code by provide numbers that may not be appropriate to send to other methods, ie negative numbers to Destroy method
- Review the libraries/namespaces we are using with our scripts, UnityEngine, UnityEngine.UI, EventSystems
- Review that usually one class per script and inherits from MonoBehavior
- Review the structure of class with variables and methods, types of variables

- Review public/private, global/local, serializeField, multiple initializations
- Review if statement structures, nested ifs, switch statement, for, forEach, while
- Review logical and arithmetic operators
- Review user input, Input class, keyboard and mouse input
- Review raycasting, rays, screen/world coordinates, global/local coordinates, physics layers, xyz coordinates, transforms, position, rotation, scale
- Review ClickDetectorManager, PrefabData, UIAnimation, and Clock scripts
- Review animation by code, IEnumerator, coroutine, yield, multiple threads
- Review animator, animation, transition, parameter, shader, material
- Review camera, lights, UI, canvas, UI elements, eventSystem
- Review inspector, scene, game, project, console, build settings, building webGL