

Weekly breakdown - Week 2

Weekly objective: Student will understand user input, camera, raycasting, screenspace, worldspace, hierarchies, game objects, prefabs and build requirements.

In-class

Goals:

Students will be able to:

- Create groups of gameobjects in hierarchy
- Create, instantiate and destroy prefabs
- Understand difference of screen and world space
- Understand coordinate systems
- Understand Unity API for mouse and keyboard input

ACTIVITIES

- Moving from hard-coded values to public variables to incrementing variable change per update, input or timer based scenario
- Talking about input, down meant a left click, a holding meant a left drag, this lesson was interesting, i think its a good opportunity for a coding activity to try our left click, right drag, etc, maybe some key combination, alt-shift-r
- Prefabs, the whole lesson references ie public gameobject variable, being able to connect different prefabs or even randomly changing them during runtime to get very different paint strokes or experiences

Lecture:

Introduction to Unity part 2

- Hierarchy
- Prefabs
- Game objects
- Camera
- Screen space
- World space
- Coordinate systems

Input API

- Mouse get, up, down
- Mouse position
- keyboard getkey, down, up

Introduction to Scripting Continued

- Manager game objects
- Update, fixedupdate, start, awake, collisions

- If else switch statements
- References
- System and unity variables
- Float notation
- Arrays
- Pseudocode
- Colliders
- Rays
- layers

Raycasting (in-class)

- Mouse cursor in screen and world space (in-class)
- Mouse position (inclass)

Exercise: [Click Detector](#) at 4:50

- Create separate game object to detect mouse clicks
 - Create a new game object to listen for click and determining world positioning
 - Talk about manager objects
 - Create ClickPosition script
 - Make use of update method
 - If statement
 - Input interface class
 - GetMouseButtonDown(index from an array)
 - Arrays
 - psuedo code
 - Vector3 variable
 - Vector3.one pre defined structures
 - Built in unity method ScreenToWorld
 - Cameras and main camera
 - Reference to objects
 - F notation for floats, list out variables

CLICK POSITION METHODS

OnMouseDown()

attach to click target, target will react, no info

ScreenToWorldPoint(mouse position + distance)

best for 2D/isometric, straight-on view

ScreenPointToRay(mouse position) -> Collider

gets precise world point, requires GameObject + Collider, may need LayerMask

Plane.Raycast

gets precise world point, infinite range, requires slightly more calculation

Online:

Goals:

Students will be able to:

- Become more advanced as a scripter
- Understand how to use raycasting to click on objects in the game world
- Be able to build and upload a WebGL build to itch.io

Lecture:

Raycasting

- Mouse cursor in screen and world space (in-class)
- Physics (concepts in-class, rigidbodies and other unity - online)
- Ray cast hit (online)
- Mouse position (inclass)

Exercise: [Raycast using Colliders](#) at 10:30

- Create a plane
- Colliders
- Ray variable
- RaycastHit variable
- Layers for physics sorting, create layers and add objects
- Public variables to add the layer mask

Exercise: [Raycast usingPlane](#) at 17:30

- Struc Plane
- Constructors, need direction and one point on plane

Lecture:

Build Requirements

- Build settings
- Game view aspect ratio
- UI positions and rect transforms need to be corrected
- WebGL build process
- Test locally with edge or firefox, not sure of safari
- Itch.io upload process

Exercise: Build and upload experience to itch.io