

HW1: Gaze Interaction

CSE165 - Discussion 2

Agenda

- Homework Recap
- Intro to Raycasting
- Cursors
- Gaze Interaction
- EC: Cursor Animations & Teleport & Explosion(Briefly)

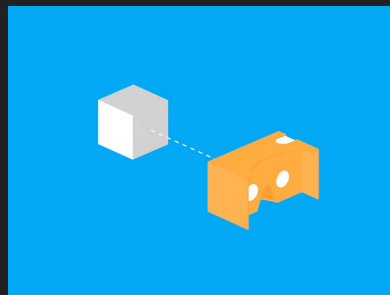
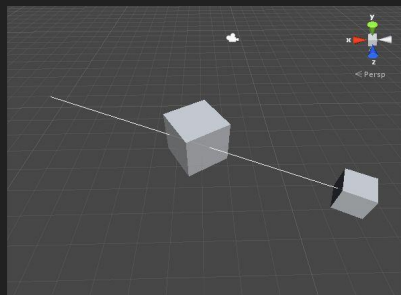


Homework Recap

- The assignment is up here <http://ivl.calit2.net/wiki/index.php/Project1W19>
- Deadline: Friday, January 25th
 - Course Schedule: <http://ivl.calit2.net/wiki/index.php/CSE165W2019Schedule>
 - You'll be presenting your project in-person in the VR Lab
- It is divided into two parts
 - Building a wall around you
 - Tearing down the wall
- For this assignment, you will only be using the HMD (no touch controllers)
- This slide will cover gaze interaction

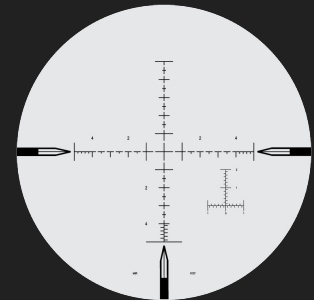
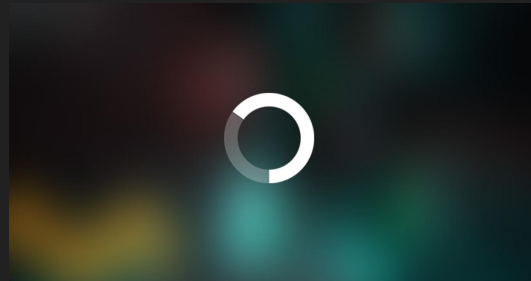
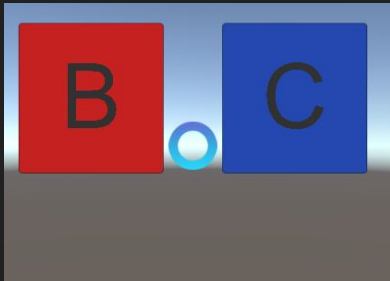
Introduction to Raycasting

- Raycasting: Projecting a ray in a direction until it hits something
 - Then retrieving the information about what the ray hit
 - This is to achieve “gaze interaction”. Camera is ray origin, and camera.forward is direction
- The Unity Documentation is always a good resource
 - <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>
- Given an initial position and a direction, checks if the ray hits anything
- Store the hit result in ‘out RaycastHit hitInfo’
- You can use Debug.DrawRay() to see the ray in scene view



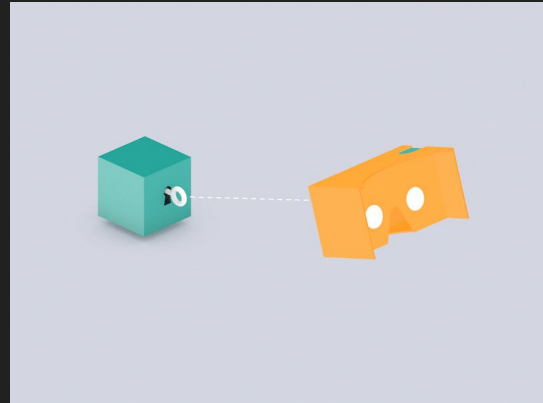
Gaze Cursors

- Create a UI -> Image object and use whatever image you want.
 - A canvas object should be automatically created
- Set the canvas Render Mode to World Space.
 - We want the cursor to be a 3D sprite, not a 2D image on a 2D plane
- Attach and place the cursor in front of the camera
 - This can be done by parenting the canvas to the camera, or using a script to move elements
- To always render the cursor over everything, you will need a custom shader.
 - <https://answers.unity.com/questions/878667/world-space-canvas-on-top-of-everything.html>



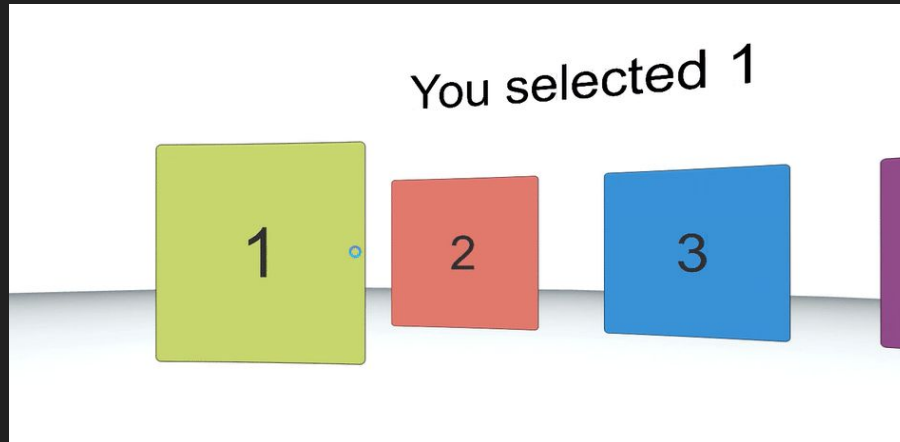
Gaze Interaction

- So how do we actually modify objects using gaze?
- Check if your raycast is dwelling on a brick for a long period of time
 - If you'd like, you can also check to make sure the user's head isn't moving too much
- To move the brick, translate it along the direction you are looking
- To visually indicate which block is being pushed or pulled:
 - Highlight: switch the shader/material of the block being pushed
 - Sparks: spawn particle effect at the point where the ray hits.
- You can access an object by using RaycastHit hitInfo
 - `hitInfo.collider.gameObject` will give you the actual object
 - This will always be the object that the raycast hit
 - Specifically, you access the object through the collider



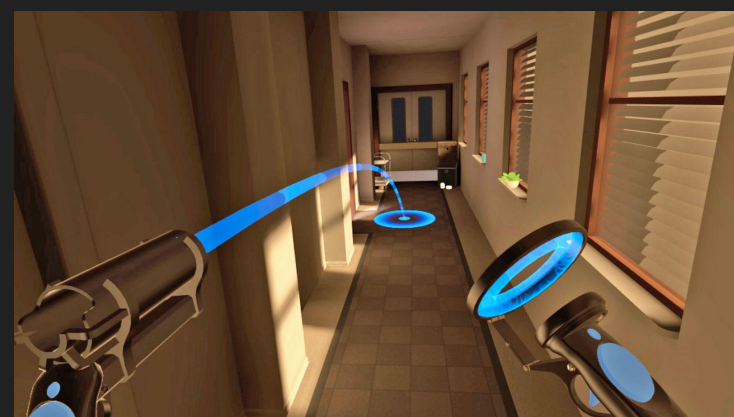
Cursor Animations (EC)

- One way to animate your cursor is to use Image Type - Filled
- Play around with the options see what works best
- Use a script to change the Fill Amount based on gaze duration
- To modify this object from a different script, give it a tag, and use `GameObject.FindGameObjectWithTag ("myTag")`



Teleport (EC)

- How can you use gaze interaction to move around a space?
- Before, users look at bricks to interact with them
- Where could a user look to teleport?
- How will the teleport be triggered?
- Can you use the existing gaze interaction system to implement teleport?



Extra Note

- UCSD **just** launched an Edx course about building mobile VR applications!
- Created by Prof. Schulze and CSE165 tutors (past and present)
- Lots of overlap with CSE165 material, and is taught in Unity3D
- Free to enroll and is self-paced!
- Launched at 4:00pm!

<https://www.edx.org/course/creating-virtual-reality-vr-apps-uc-san-diegox-cse190x>

or

<http://bit.ly/ucsdvredx>

Questions?

Feel free to ask on Piazza!

(Making your questions public is helpful to everyone!)