

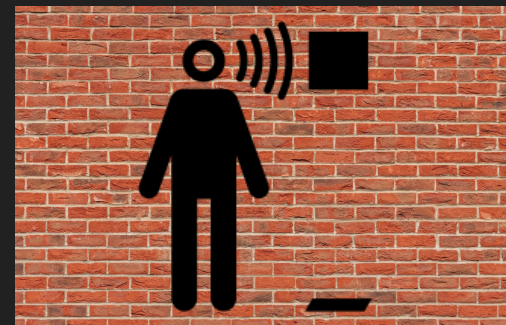
Unity & HW1

CSE165 - Discussion 1

Agenda

- Introduction to HW1
- Unity FAQ, Tips, and Tricks
- Unity & Git
- Using Unity and Oculus
- Dynamically Building a Brick Wall
- Brick Wall Stability
- Q & A

Next Week: Gaze interaction, dwelling, and timers



Homework Introduction

- The assignment is up here <http://ivl.calit2.net/wiki/index.php/Project1W18>
- Deadline: Friday, January 26th
 - Course Schedule: <http://ivl.calit2.net/wiki/index.php/CSE165W2018Schedule>
 - 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 week we will be covering building the wall through code
 - And how to make it stable!

Unity FAQ, Tips, and Tricks

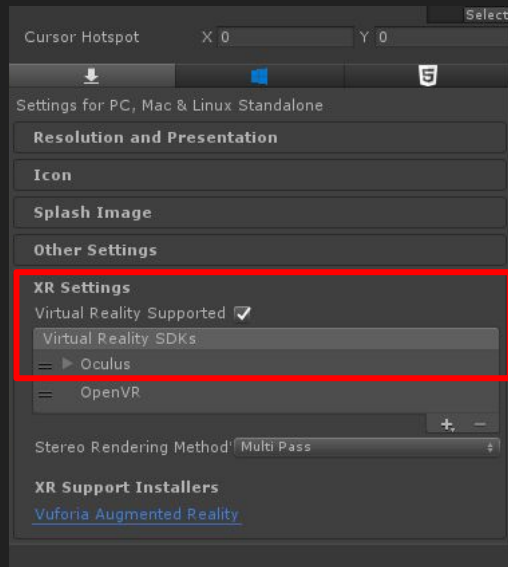
- Changes are not saved in play mode
 - You can set the playmode tint to be more obvious
 - Edit>Preferences>Colors>Playmode tint
- Unity versions: be sure to work on the same version of unity if you are working at home (you can download 2017.3 to the D: drive if you want)
- Duplicate objects: ctrl+d
- To easily align objects, hold v while dragging a vertex
- Always save your project. Make “Ctrl + S” a habit!

Unity & Git

- Unity can be a bit tricky to use with Git / GitHub, but you'll need to do it
- Make sure your repositories are **private!**
 - You get free private repositories as students on GitHub!
- Don't rely on saving work in the VR Lab - disks often get wiped
 - But when you're working, you should be working on the **D Drive** of the computer
- Great Guide: <https://robots.thoughtbot.com/how-to-git-with-unity>
 - Edit -> Project Settings -> Editor
 - Version Control / Mode: "Visible Meta Files"
 - Asset Serialization / Mode: "Force Text"
 - Save Changes!
- Git LFS helpful if you're pushing models, but it can be complicated
- Make sure you use a .gitignore! There's a good default one on GitHub
 - <https://github.com/github/gitignore/blob/master/Unity.gitignore>

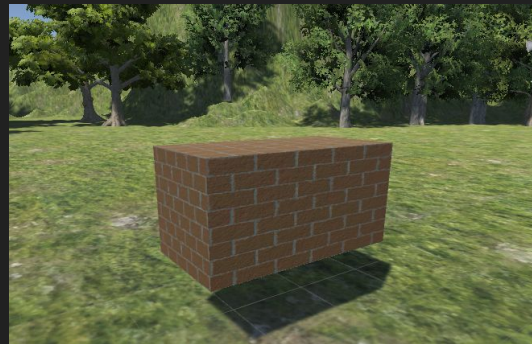
Using Unity and Oculus

- For this assignment, you don't need touch controllers - just the Oculus HMD
 - HMD = "Head Mounted Display"
- Unity has built-in XR support. Enable VR with a checkbox!
 - Edit -> Project Settings -> Player
 - Unity 2017: XR Settings -> Virtual Reality Supported
 - Unity 5.6: Other Settings -> Rendering -> Virtual Reality Supported
 - Make sure "Oculus" is selected as your Virtual Reality SDK
- VR will work with a regular "Main Camera"
- For future assignments with Touch controllers, need SDK
 - Get the "Oculus Integration" from the Unity Asset Store
 - <https://www.assetstore.unity3d.com/en/#!/content/82022>
 - Not needed for this assignment



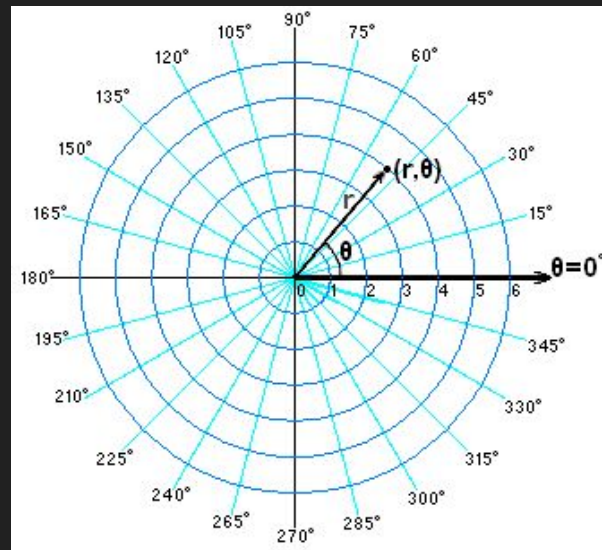
Dynamically Building Brick Wall: Start With A Brick

- Create a brick: Just add a brick texture to Unity's default Cube
 - You can just drag an image onto the cube to add it as a texture
- Save this cube as a "Prefab"
 - More on Prefabs: <https://docs.unity3d.com/Manual/Prefabs.html>
- To create a wall dynamically, you need to "Instantiate" your brick prefab
- Instantiate a prefab: `GameObject.Instantiate()`
 - <https://docs.unity3d.com/ScriptReference/Object.Instantiate.html>



Dynamically Building Brick Wall

- Remember polar coordinates?
 - Use radius r and angle θ to calculate coordinates.
 - $X = r \cos(\theta)$
 - $Y = r \sin(\theta)$
- Calculate the position and rotation of each brick:
 - Remember to add height
 - An understanding of Quaternions + Euler Angles is helpful
 - More info on next slides
- Keep the radius constant and just change the angle
- This is just one way to do it! Get creative!



Unity Rotations, Quaternions, and Euler Angles

- Whoa whoa whoa, wait, what's that "Quaternion" thing?!?!

```
public void OnTriggerEnter(Collider other) {  
    GameObject spawnedSphere:  
    Quaternion startRotation = Quaternion.Euler(Vector3.zero);  
}
```

- Rotations in Unity are stored as **Quaternions**
- Quaternions contain x, y, z, and w values.
- Quaternions are **complex** numbers... x, y, z are NOT the actual rotations!!
- However, thankfully, we can think of Quaternions in terms of **Euler Angles**
- **Euler Angles** are the rotations that we're familiar with
 - Angles in the X, Y, and Z axis. I.e. Rotated 90 degrees in x axis is (90, 0, 0).

Unity Rotations, Quaternions, and Euler Angles

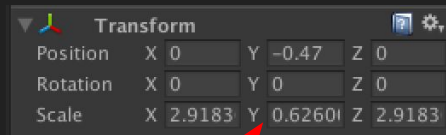
- Since Unity is our friend, it auto converts between Euler and Quaternions
- Easy method to use: Quaternion.Euler(Vector3 angles)
 - Returns a Quaternion using the specified angles

```
Vector3 newRotation = new Vector3(90.0f, 90.0f, 0.0f);  
transform.rotation = Quaternion.Euler(newRotation);
```

Sets rotation to 90, 90, 0

- We can also get Quaternions as Euler Angles
 - Just use quaternionValue.eulerAngles

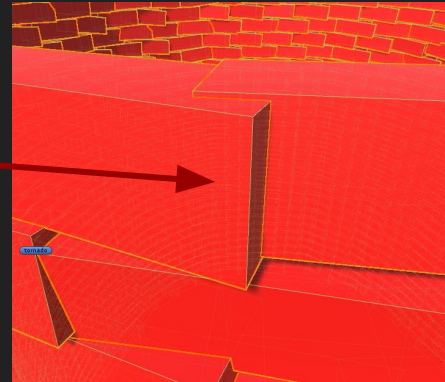
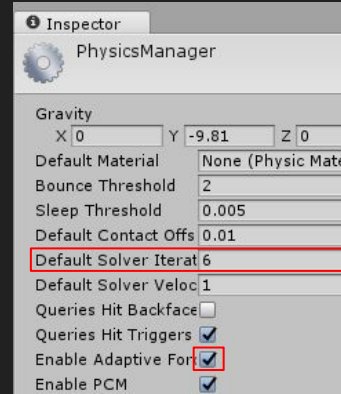
```
Vector3 eulerAngles = transform.rotation.eulerAngles;
```



- Note: The rotation values in the inspector are thankfully **Euler Angles**

Brick Wall Stability & Realistic Physics

- Brick walls initially have some weird physics/won't stay together
- A few options in the physics project settings:
 - Edit -> Project Settings -> Physics
 - Make sure "Enable Adaptive Force" is checked
 - Try increasing "Default Solver Iterations" to make the wall more stable
- You can also add a Physics Material to bricks, and change its properties
 - Particularly, the friction constants are most helpful
- Try modifying values on the Rigidbody of each brick
- You'll need to experiment a bit to get a stable/realistic wall
- Also make sure no brick colliders are overlapping!
 - Would cause explosions



Questions?

Feel free to ask on Piazza!

(Making your questions public is helpful to everyone!)