GNG 1103 - VR Project

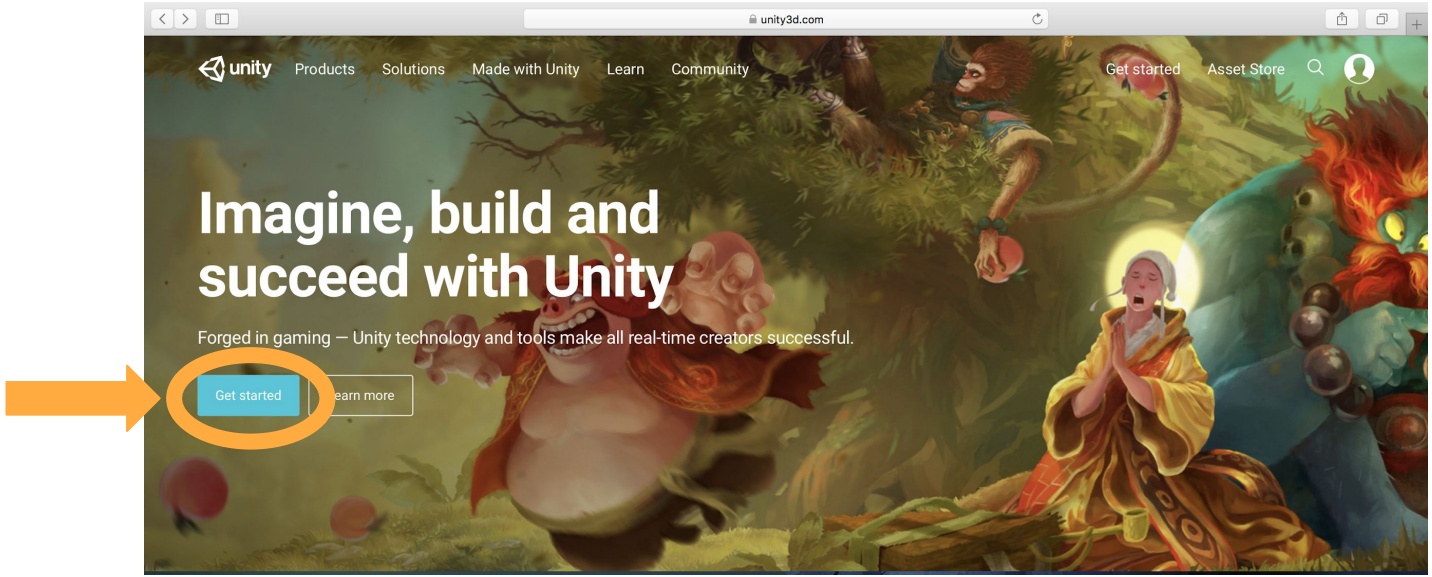
Tutorial

Pre-Lab Questions

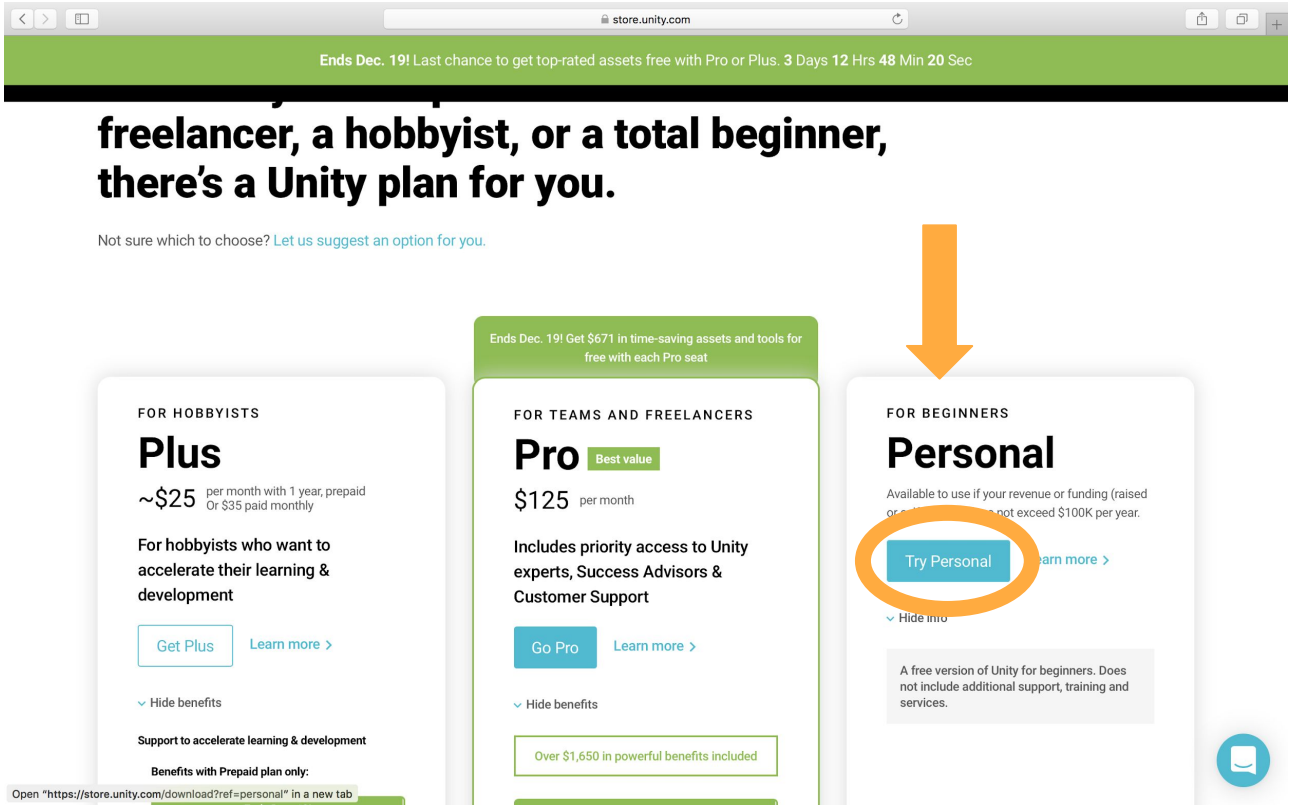
1. What is the software that you are using to build your VR games?
2. What are two deliverables that you need for the next client meeting in February?
3. When and where do the MakerSpace VR Challenge workshops occur each week?
4. What are the constraints in the project?
5. What programming language are we using to write scripts in our games?

Installing Unity3D

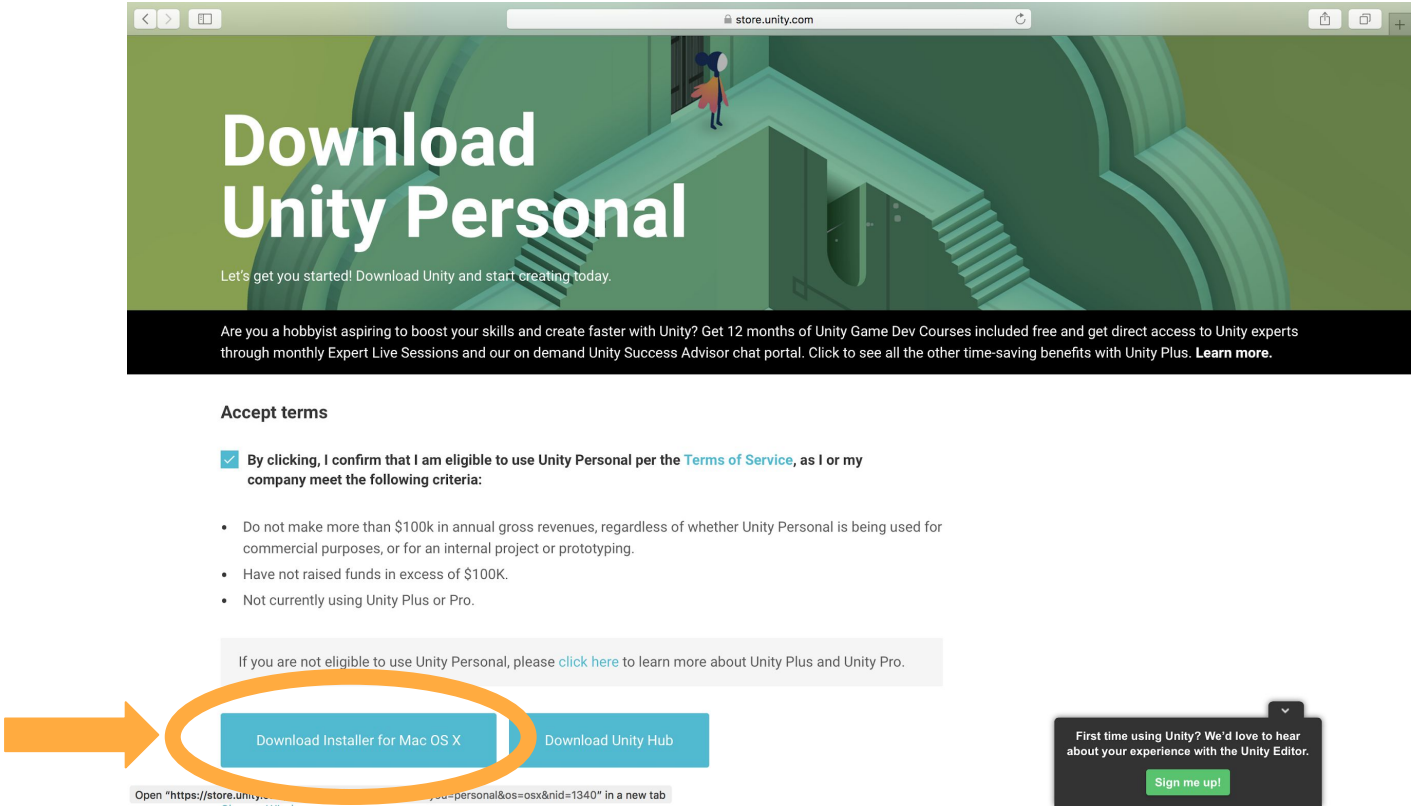
Go to Unity3D.com
Click on “Get Started”



Click on “Try Personal”



Download and Install Unity



VR Challenge Info

Slack Group
makerspacevrchallenge.slack.com

- Main communication platform for the competition
- Directly message me about questions
- I will be posting links for new workshop material
- The group can use it to share useful information with each other
- Anyone with a uOttawa email address can join
    - If you don’t have a uOttawa email address, talk to me after the session and I’ll add you

Github Repository

github.com/elishapruner/Makerspace-VR-Challenge

- Github repo has:
  - Source code for workshops
  - Powerpoint slides
  - Links to YouTube videos
  - Competition instructions

Project Constraints

Project Constraints

- Your game must have 3D models in it
- All 3D models in your game must be created by you
  - You cannot download any 3D models from the web or the Unity Asset Store
- You must build a game that fits within the constraints of chemotherapy and radiation therapy treatments
  - Chemotherapy - sitting on a chair, IV in one arm, can play game with one arm and move head
  - Radiation therapy - lying on a table, must stay very still, can move fingers
  - Avoid anything that may cause nausea
    - Stay away from fast moving things flying at you and bright colors
    - Head movement in the virtual world must be the same as in the physical world
    - ie. no teleportation → movements must be slow, teleportation may cause nausea

Next Client Meeting Deliverables

Next Client Meeting Deliverable → In February

1. A mockup of your game
2. A detailed schedule for how you will build it

Deliverable #1: A Mockup of Your Game

- The mockup will be hand drawn drawings of all of the components of your game
- Must include drawings for the following:
  - 3D models in your scenes, characters that move and objects that stay still
  - What your scene will look like
  - How things are going to move
  - Interactions that you will do with objects in your scene, how those interactions will work
  - How the user flows through the game
- Other elements
  - ie. Ideas for music and sound

Deliverable #2: Detailed Schedule

- A detailed schedule of how you will execute your project
  - Tell me exactly how you are going to build all of the components that you laid out in the mockup
- The schedule includes:
  - A list of tasks with an estimate of how many days each task will take
  - Assign the tasks to each group member, lay out who will do what and in what order
  - A break down of the number of weeks to build your game, what are the milestones each week
  - What are priority tasks → things that have to be done
  - What are wish-list tasks → things to do at the end

VR Workshop Schedule

VR Workshops

- The workshop explains many of the concepts that you need to build your games
- Workshops occur every Wednesday night at 6pm in the Design Commons room on the first floor of the STEM building next to the MakerSpace
- It is very important to attend the workshops
- If you have class and cannot attend, you can catch up and watch the YouTube Video links posted on the Github

VR Workshop Schedule

Week	Date	Topic
1	January 16, 2019	Intro to Unity
2	January 23, 2019	Real-World and Virtual Environments in Unity
3	January 30, 2019	3D Modelling in Unity
4	February 6, 2019	VR Interactions for Chemotherapy and Radiation Therapy
5	February 13, 2019	VR Workflow for Game Development
6	February 20, 2019	Reading Week
7	February 27, 2019	Character Modeling, Rigging, and Animation in Blender
8	March 6, 2019	Student Led Tutorials
9	March 13, 2019	Student Led Tutorials
10	March 20, 2019	Student Led Tutorials

Workshops that are very important for your projects

- This week’s workshop on creating environments for your world
- Next week’s workshop on 3D modeling → since 3D modeling is mandatory for your projects
- Week 4 on interactions for chemotherapy and radiation therapy

Today’s Tutorial

Creating a simple virtual world

- Building simple 3D objects in Unity
- Use a skybox
- Camera - write a simple script to move the camera in the scene
- Designing Materials
- Texturing Objects

Create a New Project in Unity

Create a New Project in Unity

- Open Unity
- In ‘Location’, find a folder in your documents where you want to store the Unity project
- In ‘Project Name’ give the project a name
- In ‘Template’ choose ‘3D’
- Click on ‘Create Project’

HDR Skybox

HDR Skybox

- High Dynamic Range (HDR) is a file format similar to JPEG or PNG
  - HDR images have more lighting information than a typical JPEG file
  - Current iPhones use the HDR image format for photos
- For VR, the HDR images provide the color and also light the scene
- HDR images are used in your scenes to make the sky

In Unity

- Click on the ‘Asset Store’ tab and search for ‘Free HDR Sky’ and click on it
- Click on the ‘Download’ button
- Now click ‘Import’

Add the Skybox to the Scene

- In the file menu, to to ‘Window’ → ‘Rendering’ → ‘Lighting Settings’
- In the Lighting Settings tab, click on the circle beside ‘Skybox Material’
- Choose ‘Skybox_Daytime’ or ‘Skybox_Sunset’
    - Alternatively you can drag the material directly into the Skybox Material box

Week 3 of workshops

- We will be creating our own custom HDR Skyboxes of real-world scenes

Basic 3D Objects

Adding 3D Objects to the Scene

- Right click in the ‘Hierarchy’ window and go to
  - 3D Object → Cube
  - 3D Object → Sphere
  - 3D Object → Capsule
  - 3D Object → Cylinder
  - 3D Object → Quad
  - 3D Object → Plane
- Rename, duplicating, or deleting the 3D object
  - Right click on the object and choose rename, duplicate, or delete
  - For renaming you can also slow double click
- Save your scene

Transformations

- Make sure the ‘Inspector’ window is visible
- Click on an object and you can see all of its values in the Inspector window
- You can use the arrows in the scene to translate the object in the x, y, z directions
- You can also do the translations more precisely in the Inspector window under ‘Transform’
  - Translate in x, y, z
  - Rotation in x, y, z
  - Scale in x, y, z
- In Unity when playing in VR → 1 unit = 1 meter
- Play with these features move the objects around, rotate them, and change their scale

Camera

Camera

- Click on ‘Main Camera’ in the Hierarchy window and look at its values in the Inspector window
- Camera view of the scene is shown in the ‘Game’ window
- Under ‘Transform’ in the Inspector window
  - Change the x, y, z position of the camera in the scene
  - Change the x, y, z rotation of the camera in the scene

Adding a Script to Move the Camera

- Create a new folder under ‘Assets’ called ‘Scripts’
  - Right click on Assets and go ‘Create’ → ‘Folder’
  - Name the folder ‘Scripts’
- Create a new script file called MoveCamera
  - Inside the Scripts folder right click and go ‘Create’ → ‘C# Script’
  - Double click on MoveCamera and it will open in a text editor or IDE
- IDE’s that you can use with Unity
  - Microsoft Visual Studio (comes with the unity install)
  - Atom, Sublime Text, VS Code
  - Rider (free with Jetbrains student account)
  - Pretty much anything works - i’ll be using Atom but use whatever you like

Adding a Script to Move the Camera

- Double click on ‘MoveCamera’ to open the file in your favorite text editor
  - Make sure the class is called MoveCamera, if not change the text here
- The class has two functions
  - Start()
    - Start runs once at the start of the program
  - Update()
    - Update runs every frame, updates 60 frames per second in Unity
- Depending on what you want your Unity program to do, you will generally start by putting your code in either the Start() or Update() functions

Adding a Script to Move the Camera

- Github Repo:
  - github.com/elishapruner/Makerspace-VR-Challenge
- Go to Github, go to Workshop 1, in the Start folder find the file MoveCamera.cs
  - Copy and paste the contents of the file into your own MoveCamera script
- Drag ‘MoveCamera’ onto MainCamera in the Hierarchy window to add the script to the camera
  - Alternatively click on MainCamera and go to the Inspector window, then drag MoveCamera into the empty space in the Inspector window
  - Alternative click on MainCamera and go to the Inspector window, click on ‘Add Component’, type MoveCamera in the search box

Adding a Script to Move the Camera

- Press Play to run the game
  - Click on or off on ‘Maximize on Play’ to either expand the game window on play or not
- Use the keyboard keys:
  - Right, left, up, and down keys to move the camera
  - Use the + or - key to zoom in and out
- Things to play with in the script
  - Change the speed value, in C# floats need an ‘f’ letter beside the number, otherwise it assumes it is a double. In VR we always use floats instead of doubles
  - Change the keys that I used to different keyboard keys
    - https://docs.unity3d.com/ScriptReference/KeyCode.html

Materials

Creating New Materials

- In the Asset folder, right click and add a new folder called ‘Materials’
- Right click in the Materials folder and go ‘Create’ → ‘Material’
  - Name the material ‘CubeMat’
- Repeat for all of the 3D objects in the scene
  - So you will have CubeMat, SphereMat, CapsuleMat, CylinderMat, PlaneMat, QuadMat
- Drag the material onto the 3D object that it is assigned to
  - Can drag onto the name in the Hierarchy window
  - Alternatively can drag onto the object in the scene
- Click on the material and look in the Inspector window
  - Beside Albedo change the color by clicking in the box
  - Change the Metallic and Smoothness values in the slider

Textures

Setting Up the Scene

- Create a new Scene and name it ‘Texture Scene’
  - In the Scenes folder, right click and go ‘Create’ → ‘Scene’
  - Alternatively go File → New Scene
  - Alternatively go File → Save As to save the previous scene into a new scene
- Add a skybox, a plane, and a capsule to the scene
  - Add the MoveCamera script to the MainCamera object to move the camera with the keyboard
- Click on the Capsule in the Hierarchy window and go to the Inspector window
  - Under ‘Transform’ move the x, y, z positions to x=0, y=1, z=0
- Click on MainCamera in the Hierarchy window and go to the Inspector
  - Under Transform change the x, y, z positions to x=0, y=1, z=-4

Download a Texture from the Asset Store

- Go to Google and search ‘unity asset store’
- In the search area of the Unity Asset Store search for ‘brick texture’
- Under the Price filter, move the slider so that it only searches for free assets, click apply to apply the filter
- Choose a brick texture that you like
- In Unity, go to the Asset Store tab, in the search bar copy and paste the name of the asset you want to download
    - Download and install the asset
- Under Assets create a new folder called ‘Textures’, and drag the installed folder into the Textures folder

Create a New Textured Material

- In the Materials folder, right click and go ‘Create’ → ‘Material’ and name the new material ‘TextureMat’
- Drag TextureMat onto the Capsule object in the Hierarchy window
  - Alternative drag TextureMat onto the capsule object in the Scene window
- Click on Capsule in the Hierarchy window and go to the Inspector window
  - Either click on the circle beside each material property to choose a texture image, or drag the texture image directly into the box beside the name
  - Add the image for Albedo, Metallic, Normal Map, Height Map, and Occlusion

Tutorial Task

Task: Create a House

- Create a new scene and name it ‘House’
- Using the basic 3D objects build a house using:
  - Cubes, quads, spheres, etc
  - Use the transforms like position, rotate, and scale to modify the shapes
- Use the Unity Asset Store to find free textures
  - Brick, roof, doors, windows, ground
- Use the MoveCamera script to rotate the camera around the house
- After you’re done, share a screenshot of your house
  - Share it with the group on Slack on the #Random channel
  - Share it on Twitter and Facebook

VR Interactions

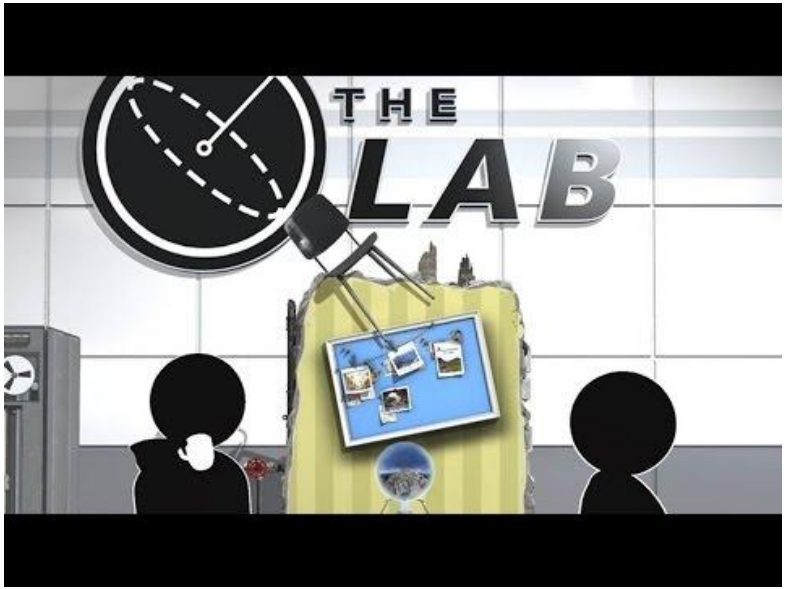
VR Interactions

- We are going to play some games, to understand more about interactions in VR and what is possible
- Bring paper and pens with you, you can start sketching out ideas
- I’m here to answer any questions you may have
- Take turns playing games, and watch on the monitor what is going on
  - What interactions the game uses
  - What kinds of 3D objects are in the scene, what can you create

SteamVR Tutorial

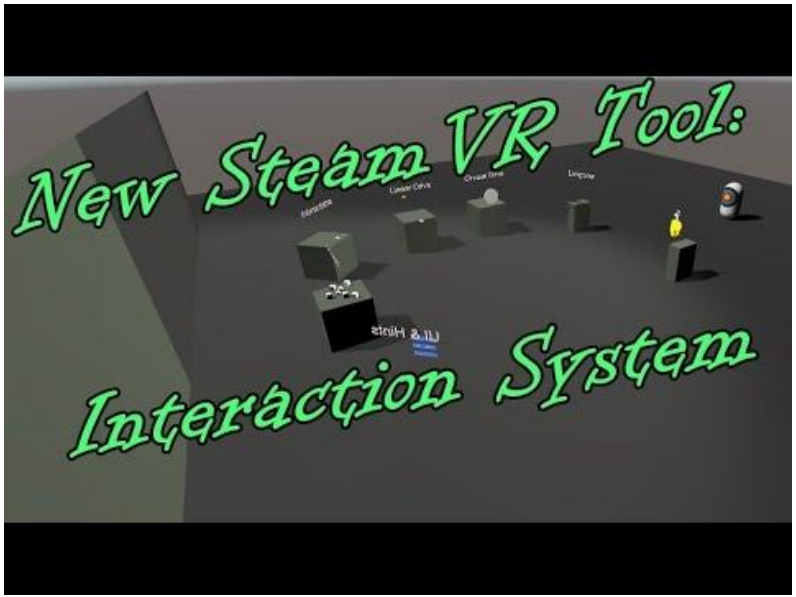
https://youtu.be/Kg7gPiz8-SU

SteamVR - The Lab



https://youtu.be/_OBLI6r2IZs

SteamVR Interaction System in Unity



https://youtu.be/EPQougskzn8