Developing with Unity

Learning Unity

Unity Documentation 1
Tutorials 1
Documentation 2
Unity Tips 2
Reparenting a GameObject 2
Quality Settings 2
Physics Settings 3
Physic Materials 3
Managing Project size 4
Networking 4

XR Development 4

Option 1: VR in Unity 4
Desktop VR 5
Mobile VR 5

Option 2: AR in Unity 5
2a: Vuforia 5
Image Target Preparation 6
Vuforia Targets 6
2b: ARFoundation, ARCore, and ARKit 7
ARKit 7
ARCore 7

Learning Unity

Unity Documentation

Tutorials

- Follow the Roll-a-ball tutorial to quickly get started with Unity.
- Tutorial videos by project and topic: http://unity3d.com/learn/tutorials
Suggested tutorial order:

- **Interface Tutorials**: Complete steps 1-6
- **Scripting**
  - For those familiar with C#: Start from the Awake/Start tutorials and complete to end
  - **3D Physics**: 1-9

Suggested Documentation:

- **Unity Interface**
- **Creating Scenes**
- **Assets**
  - **Primitive objects**
  - **Importing Assets**
  - **Notes on supported formats.**
- **Transformations in Unity**
- **Unity Scripting**
- **Input devices in Unity**

**Documentation**

- **Manual**
- **Class Reference**
- **Scripting Guide**

**Unity Tips**

**Reparenting a GameObject**

- If you want to change the parent of a GameObject in a script, you can assign the transform component of the new parent to the Transform.parent member variable of the child GameObject.
  - See **Transform.parent** for more details.

**Quality Settings**

- Unity allows you to set the graphical quality of its rendering pipeline. Quality settings are customizable and can be enabled/disabled for any platform Unity supports.
  - The Quality Settings Inspector panel can be opened by going to **Edit→Project Settings→Quality**.
  - At the top of the Quality Settings Inspector panel you will find a matrix of checkboxes.
    - The rows of the matrix represent the quality setting.
    - The columns of the matrix represent the platform.
  - To change the default quality setting, select the small black triangle in the default row under the platform you wish to deploy to, and in the drop down list, select the quality setting you want to use as default.
  - Below the settings matrix, you will find options that define the settings for the currently selected row in the matrix.
The Quality settings are broken into three categories, Rendering, Shadows and Other. Please see the Unity reference page on Quality Settings for a detailed breakdown of each setting.

Physics Settings

- Unity has a robust physics engine, but you may need to adjust the default settings in order to tune performance to your particular needs.
- To open the Physics Manager Inspector Panel, go to Edit → Project Settings → Physics.
- Please see the Unity reference page on the Physics manager for a detailed breakdown of each setting.
- Important Notes:
  - Gravity:
    - You may need to change the Gravity setting when using Tracking Targets in Vuforia.
      - Determine which axis will represent “down” in your system and change the magnitude and axis in the Gravity setting here.
  - The Bounce threshold will determine at what relative velocity bouncing will stop.
    - Setting the value too high may stop bouncing too quickly.
    - Setting the value too low may result in endless bouncing.
    - Determine what settings are appropriate for your system through experimentation.
  - Default Contact Offset
    - This sets how many meters two colliding GameObjects may overlap before the physics engine pushes them apart.
    - As with the Bounce threshold, determine what is right for your system through experimentation.

Physic Materials

- Unity allows you modify the effect of friction and “bounciness” on colliding GameObjects through Physic Materials (yup, there’s no final “s” in “Physic”).
  - To add Physic Materials to your project, go to Assets → Create → Physic Material.
  - You can change the settings of a Physic Material by selecting the material in Project View and editing the properties in the Inspector.
  - To add a Physic Material to a GameObject:
    - Drag the Physic Material onto the GameObject name in the Hierarchy View,
    - Drag the Physic Material onto the GameObject in the Scene View, or
    - Drag the Physic Material into (Inspector Panel) GameObject Inspector → Collider Component → Material.
      - If your GameObject does not have a Collider component, you will need to add one.
  - See the Unity Reference page on Physic Material for additional details.
Managing Project size

- The Unity Project may become very large as packages and assets are added.
- Consider removing unneeded packages.
- However, ensure that asset chains and required Prefabs are not removed.
  - Play it safe: Backup your project before attempting to clean it up!

Networking

- Unity has several options for networking. We recommend using PUN2 Free. Please refer to their documentation for instructions on using networking.
- If you are looking for a client-server architecture solution or other advanced features, we recommend using Mirror. Please refer to their documentation for more details.
- For comparisons between Unity networking solutions, please refer to this official post.

XR Development

You have 2 options for XR development in 4172: AR or VR (please see the appropriate sub-sections below). If you have time, try both! Lastly, if you would like to borrow one of the lab’s VR devices, let us know!

Option 1: VR in Unity

Feel free to deploy to either Oculus, Vive, Windows Mixed Reality or a Mobile VR solution when the time comes. We will be making Oculus Quest devices available later this semester.

Unity’s VR tutorials are a great introduction to VR development. First go through these three pages:

- **XR Overview**: Please complete:
  - Resources for Getting Started with VR
    - Please see below for more information on VRTK
• Getting Started with VR Development

If you are using one of the lab’s Oculus Quests, please check out the following pages:
  ○ Oculus Quest Best Practices
  ○ Oculus Quest Getting Started

Desktop VR

Vive (3rd party tutorial here), Oculus, and Microsoft have their own tutorials/samples. Please try to follow one of them.

Please note, in 2018 SteamVR drastically changed how it handles input. Please see the tutorial by Eric Van de Kerckhove on RayWenderlich.com here to learn how to use SteamVR 2.0 input.

Mobile VR

VR is also supported on mobile devices. Please see one of the following tutorials for mobile VR support.
  ● Unity VR (with built in mobile support) tutorial - same as above
  ● Unity + Oculus + Mobile tutorial
  ● Unity + Google Cardboard SDK tutorial

Please come speak to us to discuss optimization techniques for VR and AR applications on mobile devices. You can deploy to a mobile device from your laptop.

Option 2: AR in Unity

There are a number of ways of creating AR applications using Unity. Vuforia is a 3rd party cross-platform library for Unity that has been the standard for AR in Unity for years. It is fairly stable and quite robust. Unity provides an alternative, ARFoundation, which connects to platform-native AR libraries: ARCore and ARKit (Android and Apple, respectively, although ARCore will also run on Apple devices), which can sometimes perform better on certain devices.

2a: Vuforia

![Vuforia logo]

Vuforia is an AR platform that is compatible with Unity and allows for rapid development of AR applications. Please see Vuforia tutorials here.
Vuforia deploys to Android, iOS, iPadOS, and Windows UWP. You can also use Vuforia in "Play Mode" in Unity running on a Windows or macOS computer.

**Vuforia Targets**

- Vuforia supports a number of target types.
    - Create Vuforia account and login
    - Select Develop in navigation menu.
    - Select Target Manager in the navigation sub-menu.
    - Select **Add Database** and set up the new database.
    - Upload images to database using **Add Target**
      - **Natural Features and Rating**: Creating high rating targets.
    - Download targets to a local device database by selecting **Download Dataset**. (You may need to refresh the page first.)
      - Select **Unity Editor** to create a Unity package
    - Import the Unity package to your project.

**2b: ARFoundation, ARCore, and ARKit**

Unity also supports AR functionality native to iOS and Android. In the latest Unity, this is accomplished using [Unity’s ARFoundation](https://unity.com) (relevant blog post [here](https))..

**ARKit**

Apple provides [ARKit](https://developer.apple.com) for iOS devices

- **Unity’s ARKit support and tutorials** [here](https://unity.com)
- Require a macOS desktop for development
- Please see the [Apple Developer](https://developer.apple.com) page if you have not developed with iOS before.

**ARCore**
ARCore

Google provides [ARCore](https://developers.google.com/ar) for Android devices
- Unity’s ARCore support and tutorials [here](https://developer.unity.com/)
- Supported on Windows/macOS/Linux
- Please see the Android Development page if you have not developed with Android before.

As of the writing of this document, Vuforia tracks printed targets better than ARKit and ARCore, and will run in “Play Mode” on Windows and macOS computers. However, ARKit and ARCore track natural features in the surrounding environment better than Vuforia. ARCore will run on Android, iOS, and iPadOS (though some advanced features are available only on select Android devices), while ARKit will run only on iOS and iPadOS. Please take into consideration whether multi-platform support is critical in your project before deciding on using ARKit, ARCore or Vuforia.