git clone https://bitbucket.org/scopear/scopeholol enstutorial.git

(Kindly clone the starter repository above)

(or download)
https://bitbucket.org/scopear/scopeholol enstutorial/get/bfced75ad9b1.zip

Augmented Reality Quickstart

About us

We’re Brent, Kyle, and Jason.

We work on augmented reality apps (and more) at Scope AR.

We’re a local startup building augmented reality applications for industrial purposes.

WorkLink provides authoring and viewing of AR work instructions.

Remote AR provides live remote assistance over VOIP with AR annotations.

SCOPE AR
PROFESSIONAL SERVICES

TITLE: **Bandsaw**

USE CASE: In-place Maintenance & Operating Instructions

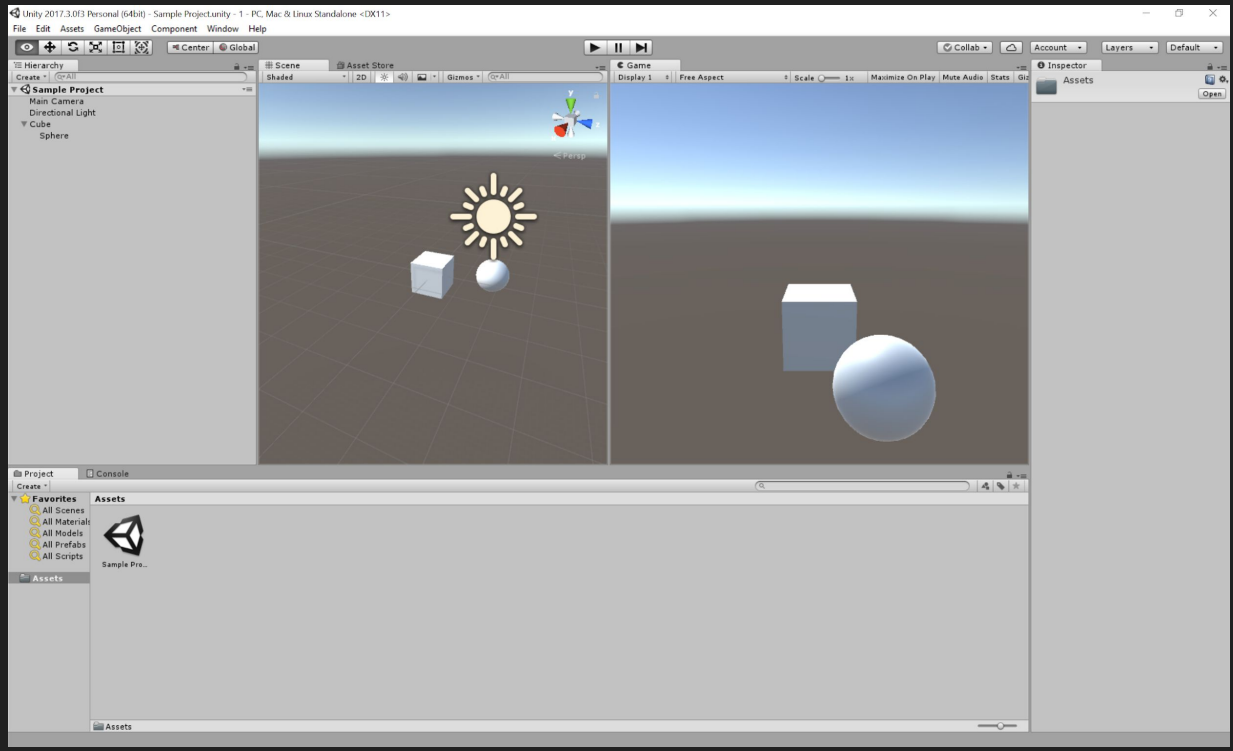
SUMMARY: This project demonstrates what can be achieved even in circumstances when 3D models of equipment are not available. New simple models can be created, or more comprehensive ones developed through various techniques. Instructions can even be created which directly reference the working environment via arrows, text, photos and video placed in the exact location needed.

Agenda

1. Getting started with Unity
2. Creating a new HoloLens project
3. Making a game
    - Shooting a ball
    - Voice Commands
    - Interacting with the “real world”
    - Creating and placing a target
4. Free Time Suggestions
    - Implement A Voice Command
    - Increase ball velocity with hold gesture
    - Bounce noises
    - Score UI

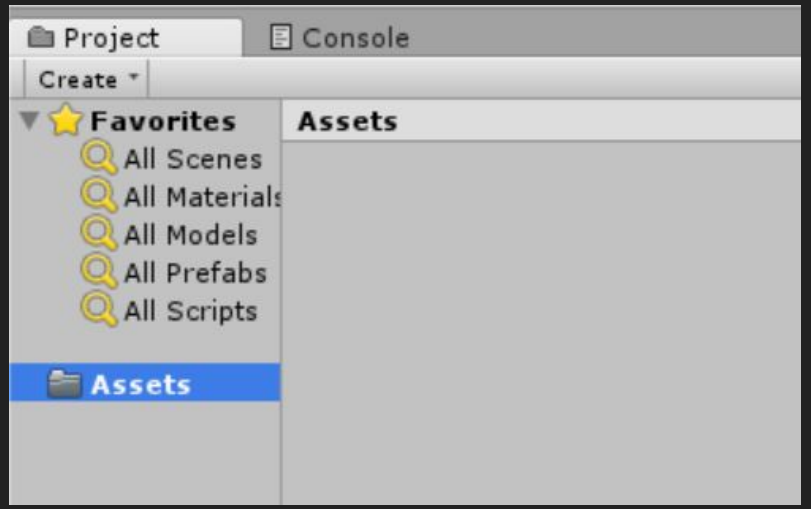
Getting to know the Unity UI

- Project View
- Scene View
- Game View
- Hierarchy
- Inspector
- Console



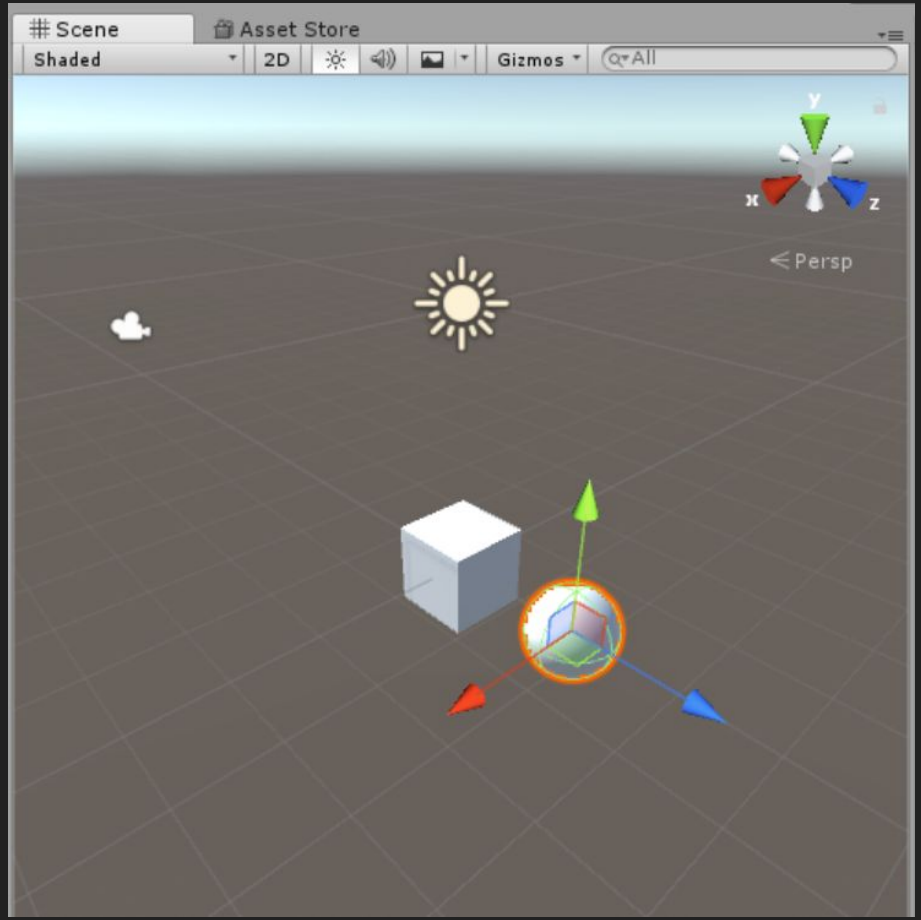
Unity - Project Window

- View and manage the assets and scripts in your project
- Think “Solution Explorer” in Visual Studio



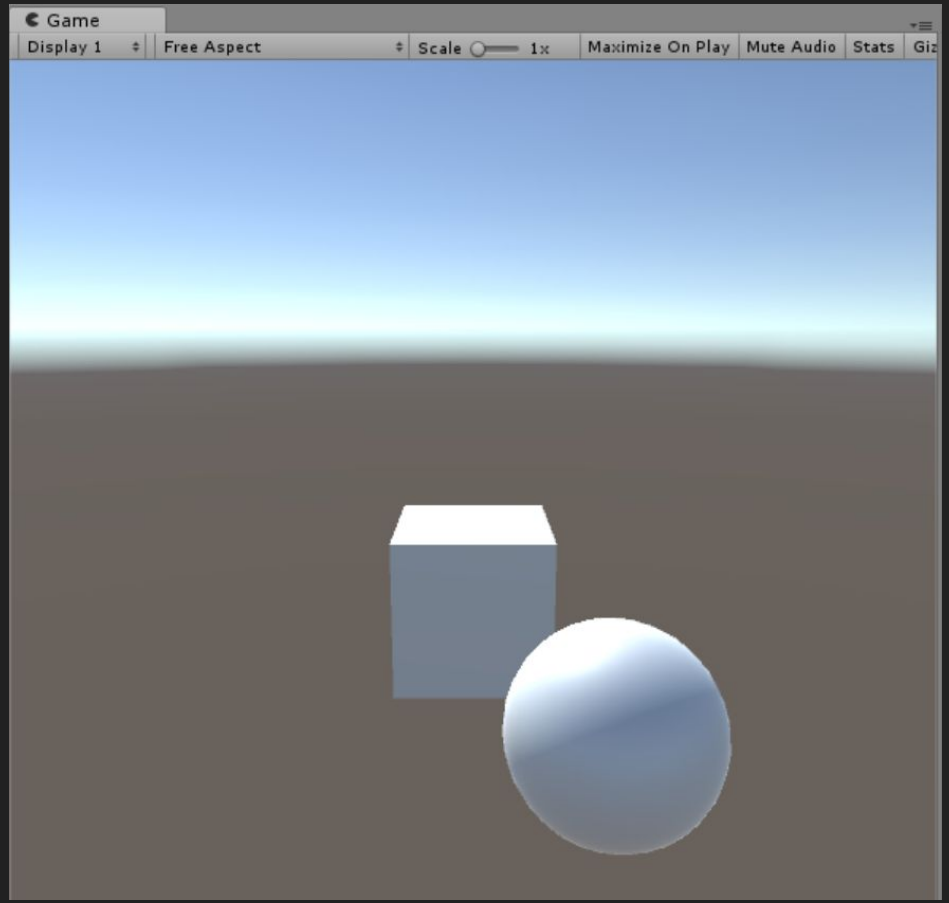
Unity - Scene View

- Editable preview of game objects in 3D space
- 3D debug view
- Objects can be selected and manipulated to edit properties



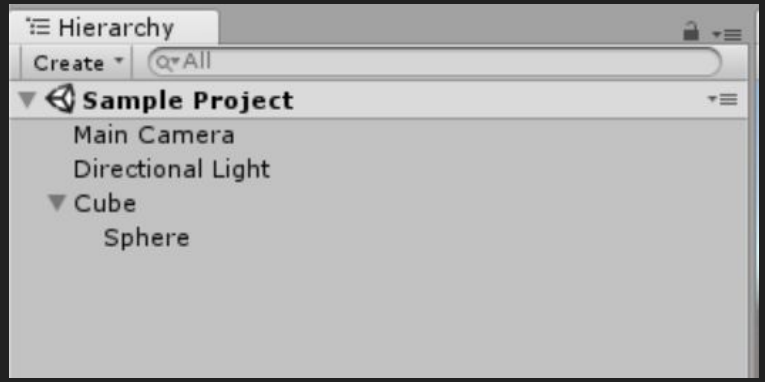
Unity - Game View

- Rendered view from the cameras in your game or app
- Shows what the user will actually see



Unity - Hierarchy Window

- Contains all the game objects in your scene
- Parent-child relationship allows for transformations relative to the parent
- Example: when “Cube” is translated on the x-axis by 10 units, “Sphere” follows the same translation



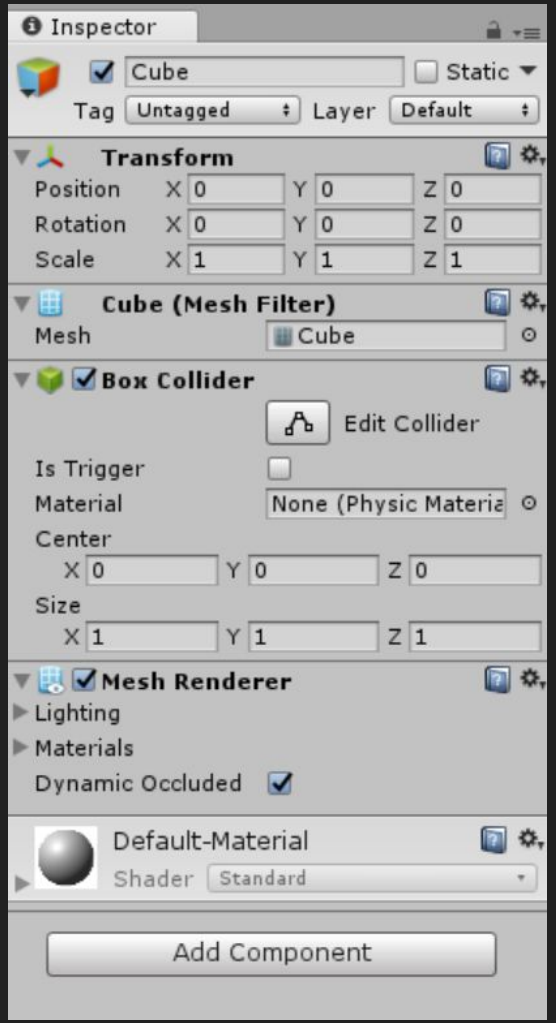
Exercise

Creating and manipulating game objects with the Hierarchy Window and the Scene View

1. Hierarchy Window
   - Right click (on empty space) -> 3D Object -> Cube
   - Right click on Cube -> 3D Object -> Sphere
2. Scene View
   - Click to select the sphere
   - Drag any of the axes to move it away from the cube
3. Press “Play”
4. Fiddle around
   - Try translating the parent cube (by dragging an axis in the Scene View)
   - Try translating the Main Camera and watch the Game View react

Unity - Inspector Window

- Allows editing of selected game object in the Hierarchy Window
- Components (like scripts, game physics hookups, rendering properties, etc.) can be attached to the game object here
- Think “Property Manager” in Visual Studio



Exercise

Scripting a Game Object

1. Hierarchy View -> Select the cube
2. Inspector View -> Add Component -> New Script -> Create and Add
3. Inspector View -> “Gear” icon on script -> Edit Script
4. `Update()` gets run every frame by the game engine. Write a block of code to toggle the cube’s position every other second. You might need:
   - `Mathf.Round(Time.time)` to get the number of seconds since started
   - `this.gameObject.transform.position` to get/set the object’s 3D position

Exercise

One way to do it:

```
void Update ()
{
    Vector3 position = Mathf.RoundToInt(Time.time) % 2 == 0
        ? new Vector3(0, 1, 1)
        : new Vector3(0, 0, 1);

    this.gameObject.transform.position = position;
}
```

Unity for HoloLens Development

- The same, but compiled for HoloLens
- Your head now controls the camera rather than player input
- Mixed Reality Toolkit makes the mixed reality parts easy

Mixed Reality Toolkit

- Microsoft’s toolkit to get you up and running fast with mixed reality development in Unity
- Hookups for using:
  - Spatial mapping
  - Gaze tracking
  - Gestures
  - Voice commands
  - etc.
- We’ll use this release: https://github.com/Microsoft/MixedRealityToolkit-Unity/releases/tag/2017.2.1.4
- Getting started guide: https://github.com/Microsoft/MixedRealityToolkit-Unity/blob/master/GettingStarted.md

HoloLens How-To

- Air tap
- Bloom
- Voice commands
- Pairing and compiling
  - Select “Device” in deployment target drop down
  - On HoloLens goto Settings > Update > For Developers and click “Pair”

Building a Mixed Reality Game