



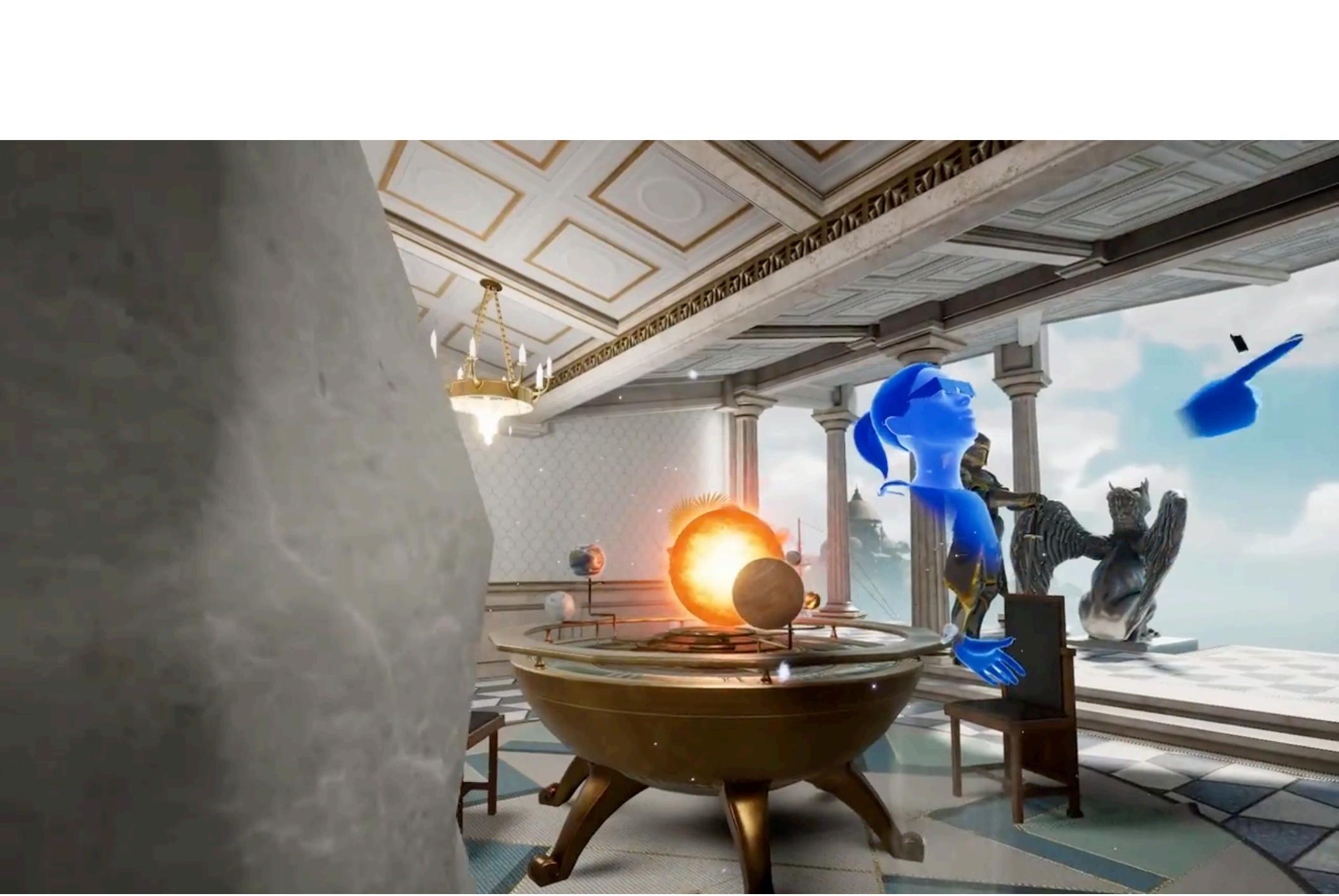
Oculus Go

TECH 421 - Future of Digital Media

TECH 3706 - AR/VR in Architectural Environments



oculus connect







bigscreen



Milestone: 10/17

App Delegate
Listens for and responds to system-level events

Comes with:
appStarted()
aboutToEnterBg()
didEnterBg()
willResume()
didResume()
willTerminate()

View Controllers
Instances of Class to handle view states (see UI chart)

Send UI events to StateManager and receive UI changes from StateManager

State Manager
Singleton accessible to any other class
All program logic goes through here.

closestBeacons[5]
currentStoryScene
possibleNextStoryScenes[]
conditionsForAdvancingStory
new Authorizer()
new LocationManager()
new DataManager()
new AudioPlayer()

startLocationChecks()
updateClosestBeacons()

startStory()
isStoryReadyToAdvance()
doesLocationMeetConditionsToAdvance()
advanceStory()
pauseStory()

saveStoryProgress()
loadStoryProgress()

Authorizer
Class to handle checking permissions and hardware and prompting user to authorize use.

isLocationEnabled()
isBluetoothEnabled()
isWifiEnabled()
areHeadphonesIn()
areAllEnabled()
listenForChangesToStates()
askPermissionForState()

Location Manager
Class to handle beacon ranging and GPS

geoFenceRegions[]
beaconRangingRegions[]

start/stop beaconRanging()
start/stop GPSTracking()
isInGeofence()

Audio Manager
Class to handle playing audio

currentlyPlayingAudio
audioReadyToPlay

play/pause/stop/resume()
setFgBgMix(ratio)
prepAudioOnDeck([audioReferences])

Data Manager
Class to hold references to audio files on disk, hold the structured data file with story logic, and update new versions of both as needed

storyDataObject
referencesToAudioFilesOnDisk[]
mostRecentWeather
stateOfRelevantWorksOfArt[]
new WebConnectionManager()

checkIfNewVersionOfStory()
downloadNewMedia()
verifyNewMedia()
clearOldMediaFromDisk()
returnReferenceToAudioOnDisk(ID)

WebConnectionManager
Class to access external, frequently-changing data like weather and art-on-view.
Possibly also to handle syncing groups over webSockets

getOverheardStoryData()
getOverheardStoryVersion()
getWeather()
getArtStatus([artWeWant])

Key:
Class (Blue Box)
Properties/Instances (Red Box)
Methods (Green Box)

State Manager

Singleton accessible to any other class
All program logic goes through here.

```
closestBeacons[5]
currentStoryScene
possibleNextStoryScenes[]
conditionsForAdvancingStory

new Authorizer()
new LocationManager()
new DataManager()
new AudioPlayer()
```

```
startLocationChecks()
```

```
updateClosestBeacons()
```

```
startStory()
```

```
isStoryReadyToAdvance()
```

```
doesLocationMeetConditionsToAdvance()
```

```
advanceStory()
```

```
pauseStory()
```

```
saveStoryProgress()
```

```
loadStoryProgress()
```

Authorizer

Class to handle checking permissions and hardware and prompting user to authorize use.

```
isLocationEnabled()
```

```
isBluetoothEnabled()
```

```
isWifiEnabled()
```

```
areHeadphonesIn()
```

```
areAllEnabled()
```

```
listenForChangesToStates()
```

```
askPermissionForState()
```

Location Manager

Class to handle beacon ranging and GPS

```
geoFenceRegions[]
beaconRangingRegions[]
```

```
start/stop beaconRanging()
```

```
start/stop GPStracking()
```

```
isInGeofence()
```

Audio Manager

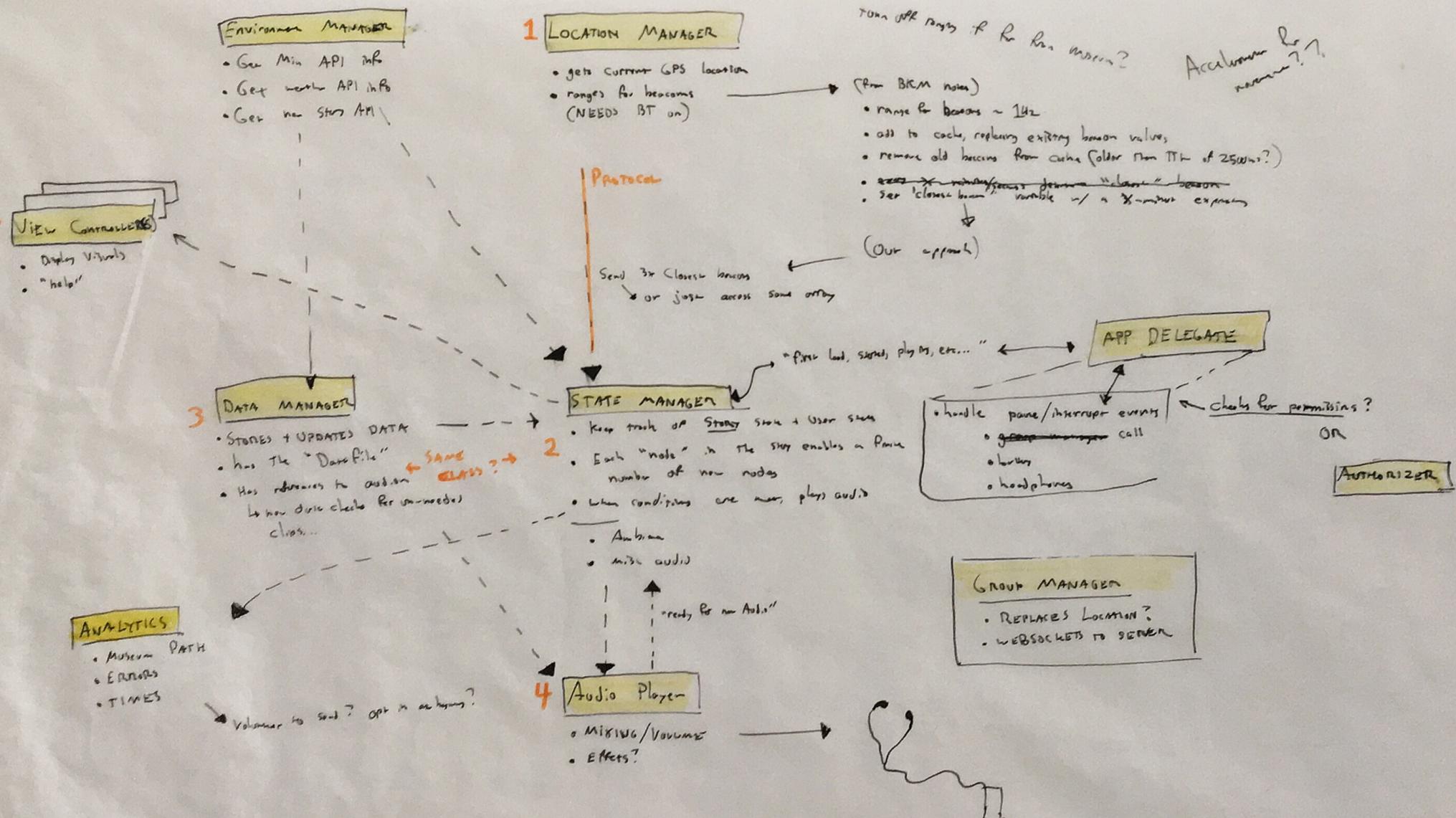
Class to handle playing aud

```
currentlyPl
audioR
```

```
play/pause/stop/resume()
```

```
setFgBgMix(ratio)
```

```
prepAudioOnDeck([audioRefe
```



Due: 10/24

10/26
In Class Showcase +
Discussion

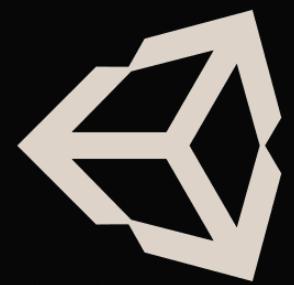
All Materials due 10/31:

Documentation

- Video of the program
- Planning documentation

Unity Project File

Finished (Compiled) Program



unity



MixedReality Toolkit

(formerly HoloToolKit)

<https://github.com/Microsoft/MixedRealityToolkit-Unity>

Platform



PC, Mac & Linux Standalone



Universal Windows Platform



iOS



tvOS



Android



Tizen



Xbox One



PS Vita



Universal Windows Platform

Target device

HoloLens

Build Type

D3D

SDK

10.0.14393.0

Build and Run on

Local Machine

[Copy References](#)



Debugging



Unity C# Projects



Development Build



Autoconnect Profiler



Scripts Only Build

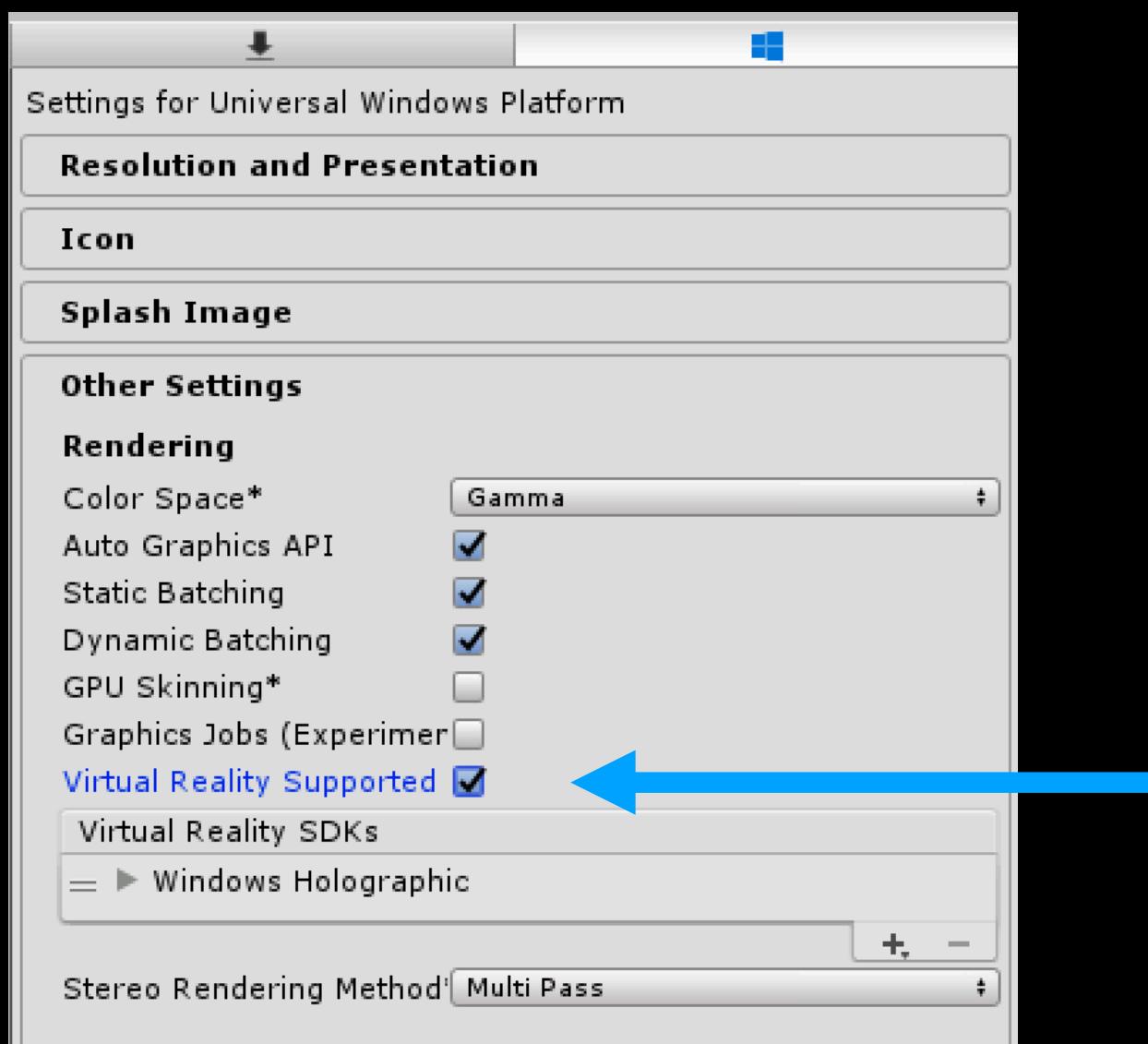
[Learn about Unity Cloud Build](#)

Switch Platform

Player Settings...

Build

Build And Run



Useful components in HoloToolkit



The background of the slide features abstract, wavy patterns in shades of dark blue, teal, and light grey, creating a dynamic and modern look.

Importing the package
will create a folder
called “HoloToolkit” in
your “Assets” folder.

This is full of stuff to
make your life easier.

Prefabs to include:

- **HoloLensCamera**

- (Make sure to delete “MainCamera” from your scene)

- **Input Manager**

- **InteractiveMeshCursor**

Easy-to-use Interaction Scripts

Assets/HoloToolkit/Input/Scripts/Interactions/TapToPlace

Attach to a GameObject to allow you to pick it up and move it around with an air-tap. Hololens will try to drop the object at the point you're looking at.

Assets/HoloToolkit/Input/Scripts/Interactions/HandDraggable

Attach to a GameObject to allow you to pick it up and hold it with an air-tap and move it through space as long as you hang on to it

More Easy-to-use Interaction Scripts!

Assets/HoloToolkit/Utilities/Scripts/Billboard

Attach to a GameObject to make it always rotate to face you.

Assets/HoloToolkit/Utilities/Scripts/Tagalong

Attach to a GameObject and it will follow your view around, ensuring that some part of it is always visible. Combine with **Billboard** for a great approach to menus and instructions.

Advanced Interaction Scripts: Events

What if you want to trigger your own behaviors on taps and gazes?

`HoloToolkit/Input/Scripts/InputEvents/`

These are scripts that define events you can listen for. You can't add these to objects directly...

Advanced Interaction Scripts: Events

`HoloToolkit/Input/Scripts/InputEvents/`

These scripts define the methods you have to have in your script to listen for these events.

Examples on the github (with comments)

WIRED:
At Oculus' Developer Summit,
VR Progress Is a Game of Inches

<https://www.wired.com/story/oculus-connect-virtual-reality-progress/>

Until those billion people get into headsets, though, the company seems to be doing its best to make VR easier to use—and as persistent as possible once you're using it.

<https://www.wired.com/story/oculus-connect-virtual-reality-progress/>



Oculus Go

TECH 421 - Future of Digital Media

TECH 3706 - AR/VR in Architectural Environments