Overview of AR Applications

Tue, June 30 (Week 2)
The Loop of Every Interactive 3D Software

Loop per Frame:

Input -> CPU -> GPU -> Monitor

Inputs:

Keyboard/Mouse/Touch/Trackers/Controllers/Cameras
Between Every Frame... (< 1/30 seconds)
The Structure of a 3D Software

1. Collect the input and interpret it.
   : Tracking and Camera Detection

2. Update the virtual world inside.
   : The Application-specific Logic

3. Render the virtual world.
   : Meshes, Textures, and Lights
The Structure of a 3D Software

1. Collect the input and interpret it.
   : Tracking and Camera Detection

2. Update the virtual world inside.
   : The Application-specific Logic

3. Render the virtual world.
   : Meshes, Textures, and Lights
CPUs/GPUs
The Structure of a 3D Software

1. Collect the input and interpret it.
   : Tracking and Camera Detection

2. Update the virtual world inside.
   : The Application-specific Logic

3. Render the virtual world.
   : Meshes, Textures, and Lights
The Structure of a 3D Software

1. Collect the input and interpret it.
   : Tracking and Camera Detection

2. Update the virtual world inside.
   : The Application-specific Logic

3. Render the virtual world.
   : Meshes, Textures, and Lights
Game Engine

Unity XR Tech Stack

Mixed and Augmented Reality Studio (MARS) → AR Apps → VR Apps

AR Foundation → XR Interaction Toolkit

XR Subsystems
- Display
- Input
- Environment
- Faces
- Raycast
- Camera
- Planes
- Image Tracking
- Object Tracking
- Meshing

Unity XR SDK
- ARCore XR Plugin
- ARKit XR Plugin
- Oculus XR Plugin
- Windows XR Plugin
- Magic Leap XR Plugin
- VSP/3rd Party XR Plugin
Game Engine

Unity XR Tech Stack

Mixed and Augmented Reality Studio (MARS) → AR Apps → VR Apps

AR Foundation → XR Interaction Toolkit

XR Subsystems

- Display
- Input
- Environment
- Faces
- Raycast
- Camera
- Planes
- Image Tracking
- Object Tracking
- Meshing

Unity XR SDK

Provider Implementations

Developer Tools

XR Plugin Framework
Trade-off Cost of Relying on Pre-existing Code

1. Collect the input and interpret it. : Tracking and Camera Detection  
   - Small

2. Update the virtual world inside. : The Application-specific Logic  
   - Large

3. Render the virtual world. : Meshes, Textures, and Lights  
   - Small
Trade-off Cost of Relying on Pre-existing Code

1. Collect the input and interpret it.  
   : Tracking and Camera Detection  
   \[\text{Use Engine}\]

2. Update the virtual world inside.  
   : The Application-specific Logic  
   \[\text{Write Yourself} = \text{Application Code}\]

3. Render the virtual world.  
   : Meshes, Textures, and Lights  
   \[\text{Use Engine}\]