

Introduction to Unity and MiddleVR

David J. Zielinski

DiVE Virtual Reality Lab
Duke University

Overview

- Unity Overview
- Editor Interface
- Scripting
- MiddleVR

Unity - Why even use a game engine?

Simplify development of games.

- Game Mechanics
 - physics, AI
- Rendering Effects
 - shadows, lighting
- I/O Abstraction
 - input devices
 - output devices



Unity - What is it?

- Commercial game engine (Free and Pro)
- Cross platform
- Popular

Unity - Commercial Game Engine

Pro Version offers:

- LOD
- Occulsion culling
- Profiler
- Full screen post processing effects (Oculus)
- Quad Buffered Stereo (for zSpace and DiVE)
- ... and more!

Unity - Cross Platform

Multiple Build Targets (Input/Output Abstraction)

- Windows
- Linux
- iOS
- Android
- Web
- ... and more!

Unity - Popular

- 2 million registered users
- Large asset store (models, scripts)
- Utilized in other Duke classes

Unity - Competitors



UNREAL
ENGINE

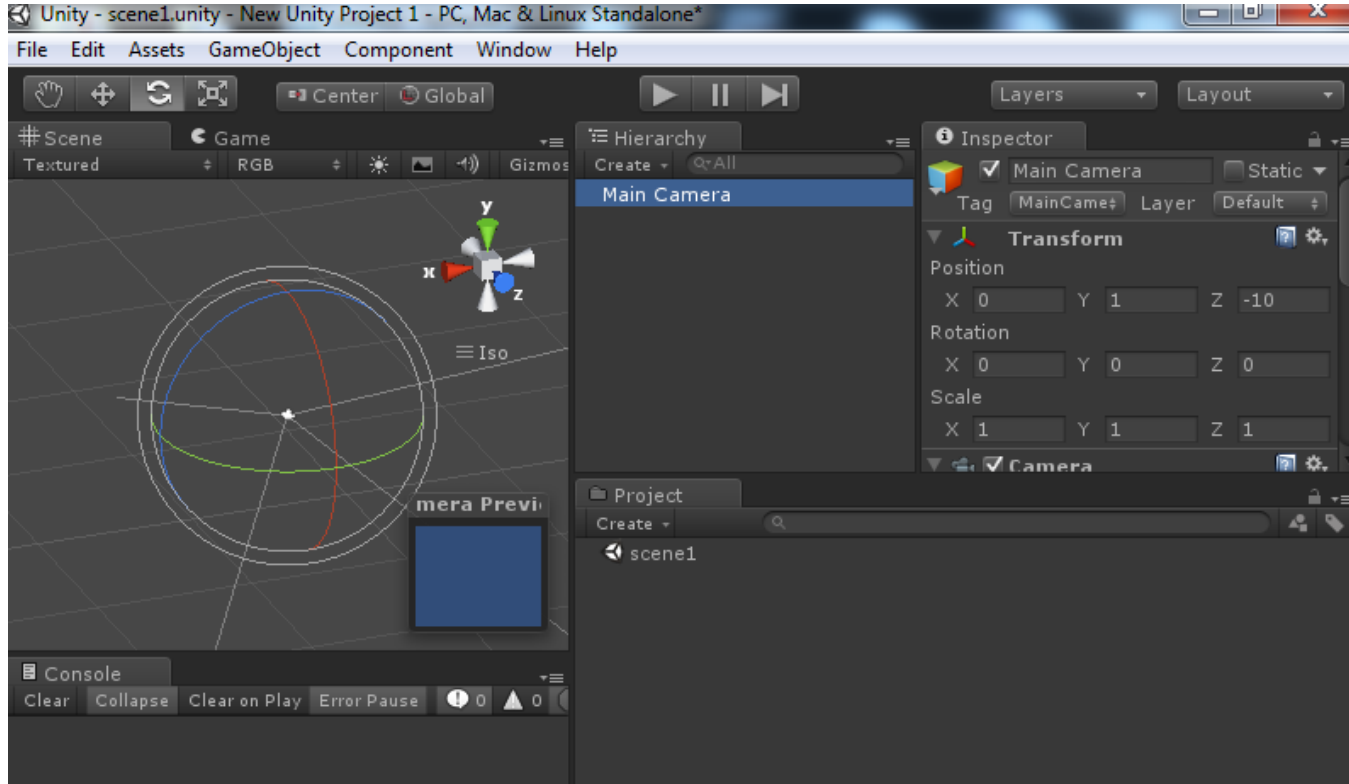


Unity - Understanding Projects

A Unity project...

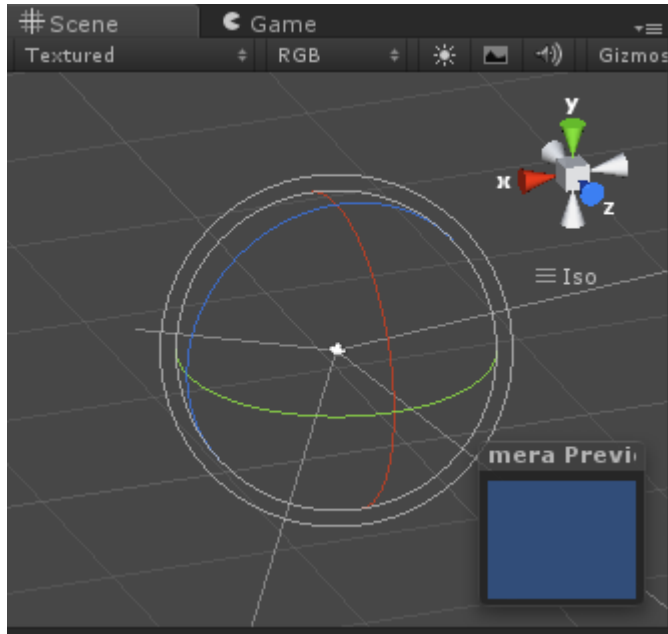
- is actually the directory
- will contain all the assets for your game
- can contain multiple scenes

Unity Editor - Windows



Scene
Hierarchy
Inspector
Project
Console

Unity Editor - Scene Window



Orbit:

alt+left mouse

right mouse button

Pan:

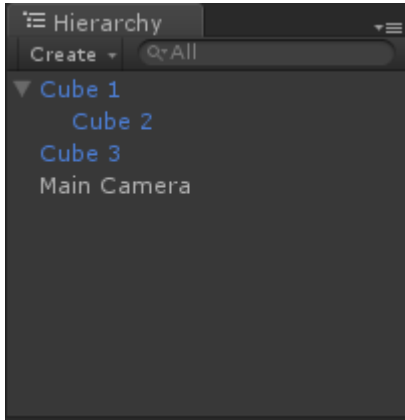
middle mouse

Zoom:

scroll wheel

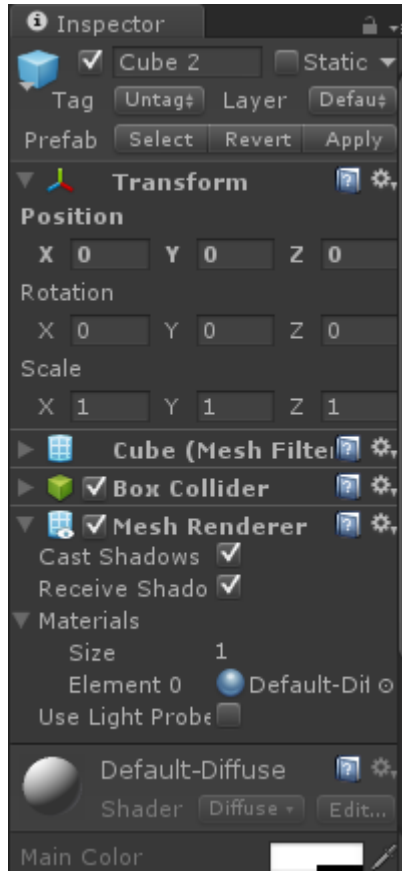
alt+ right mouse

Unity Editor - Hierarchy



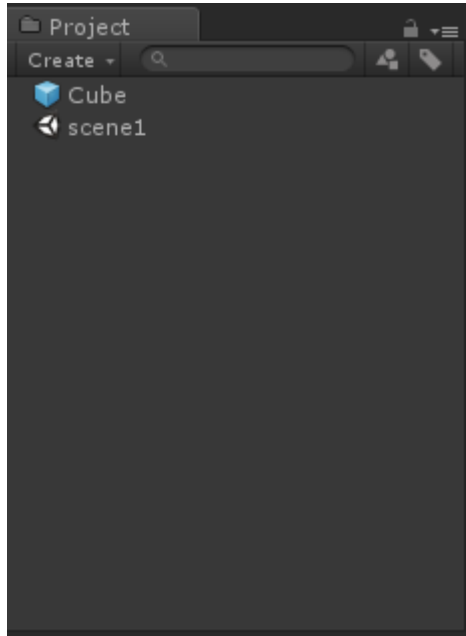
- Show objects in current scene
- Can drag objects underneath another object.
- Selecting an object will change the view in the inspector window (next slide)
- Tip: can use 'f' key to center on object in scene view

Unity Editor - Inspector



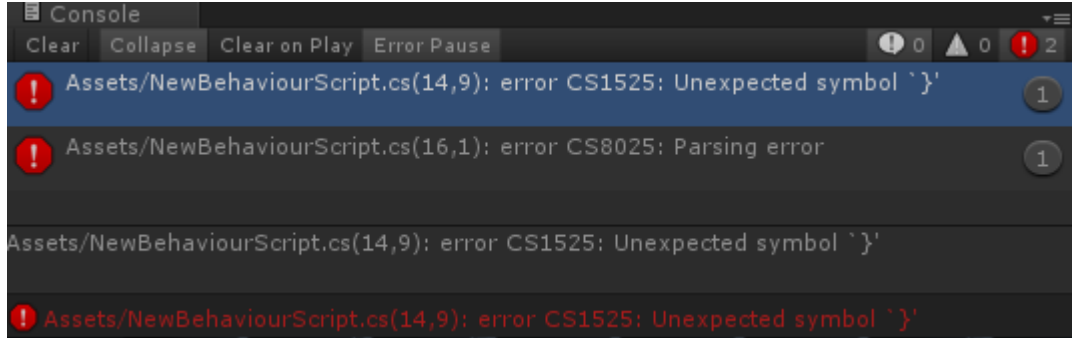
- Shows currently selected game object.
- Lists all components of that game object.
- Scripts will show up here as well.

Unity Editor - Project



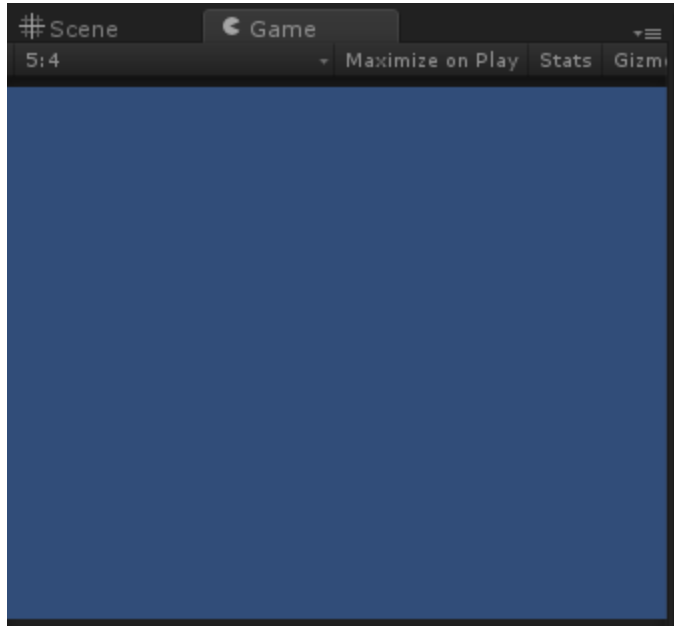
- Lists all assets in your project

Unity Editor - Console



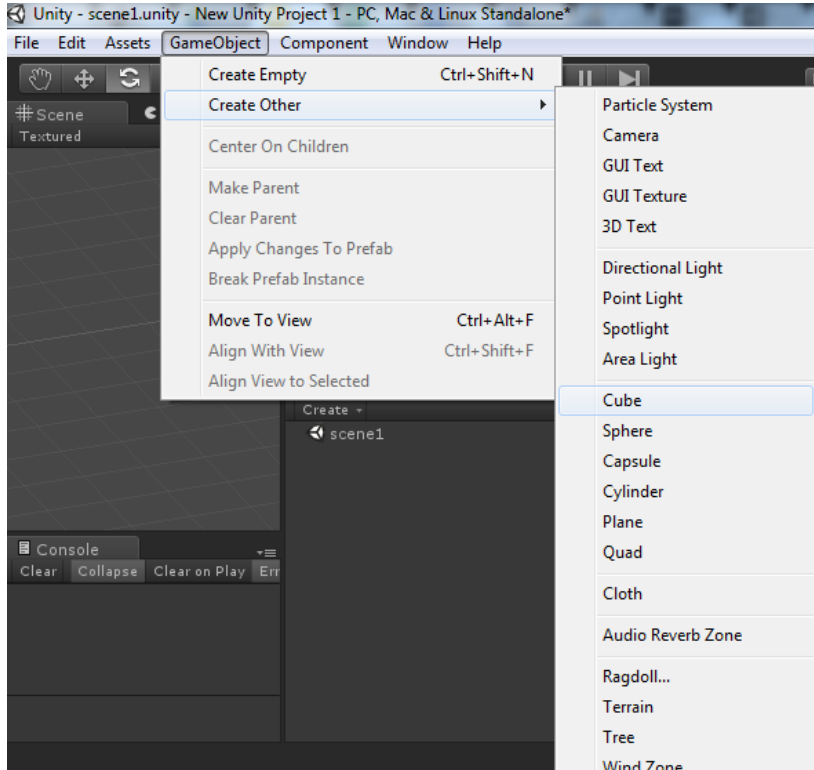
If there are errors in your script, they will show up here when you press play.

Unity Editor - Game Window



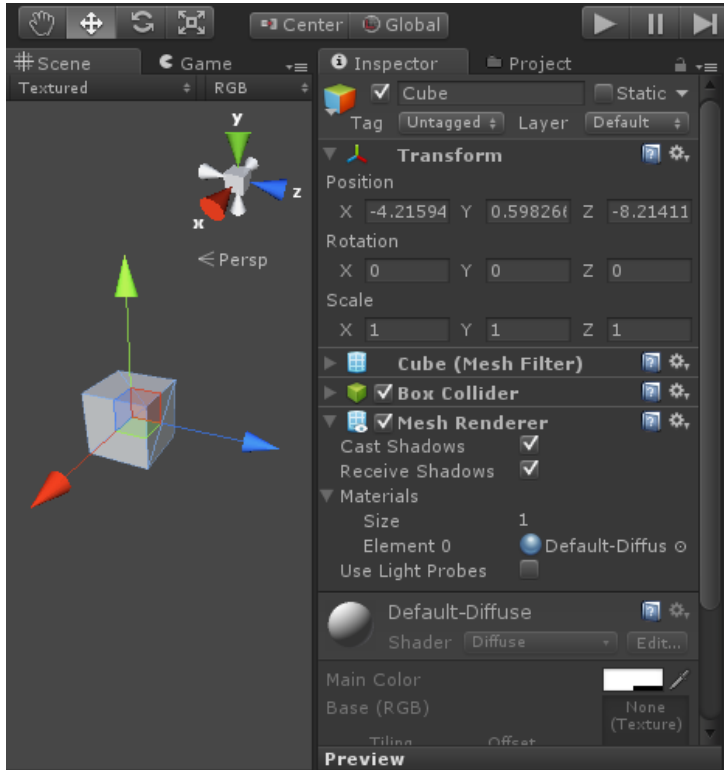
- When you press play the scene window will switch to the game view window.
- **IMPORTANT:** values adjusted while in play mode will revert when play ends.

Example - Adding a Cube



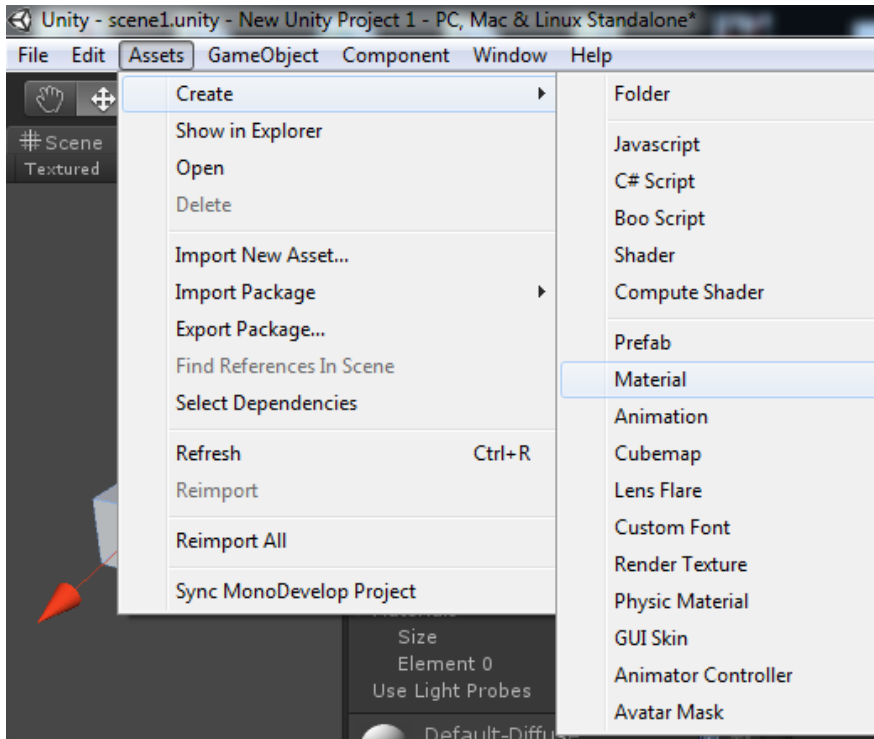
GameObject
Create Other
Cube

Example - Moving a cube



- can use handles in scene view to move
- can adjust values in inspector view
- Tip: click value and use mouse to quickly adjust

Example - add material to cube

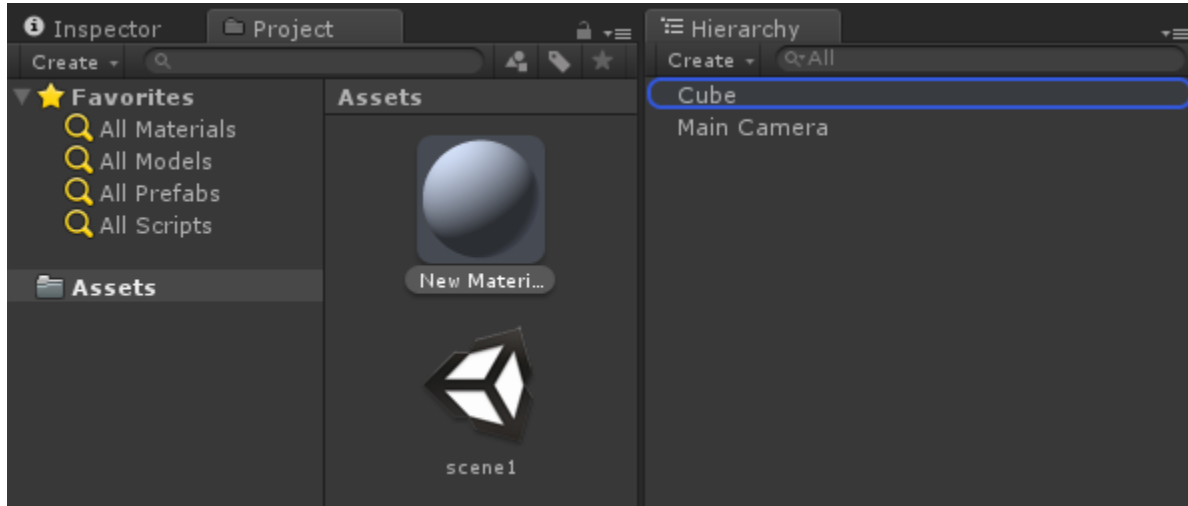


Assets

Create

Material

Example - add material to cube part2



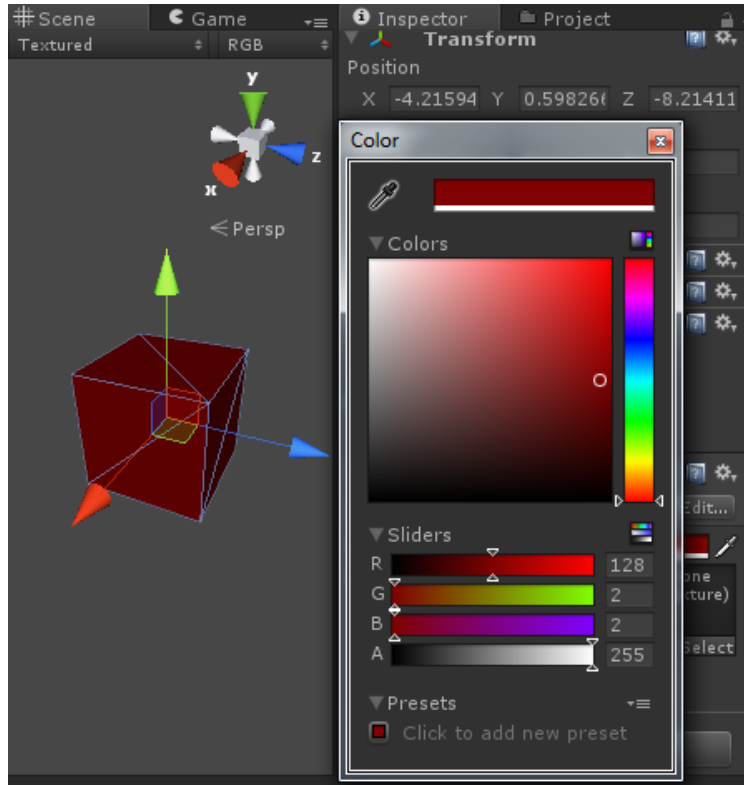
Drag material from project window onto Cube in Hierarchy window.

Example - Material added to cube

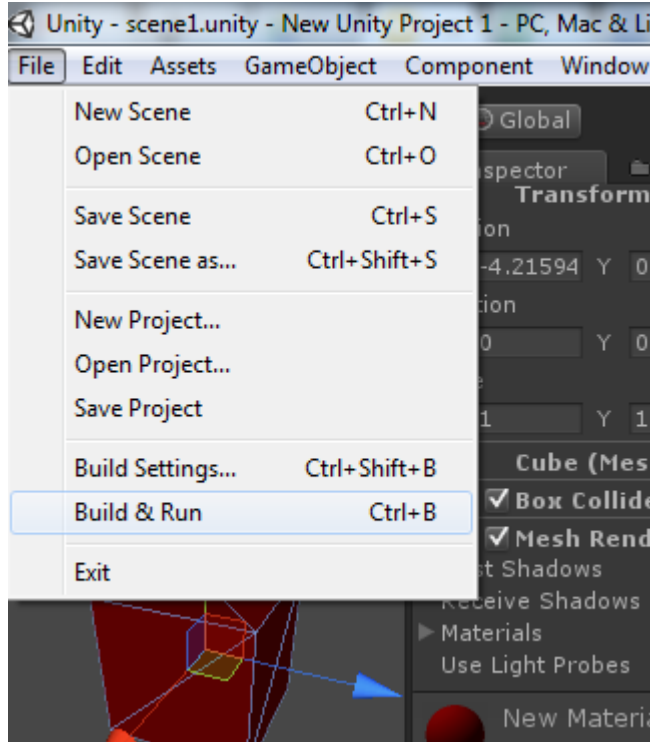


- We can change the shader used
- We can change the color by bringing up the color picker

Example - Color picker



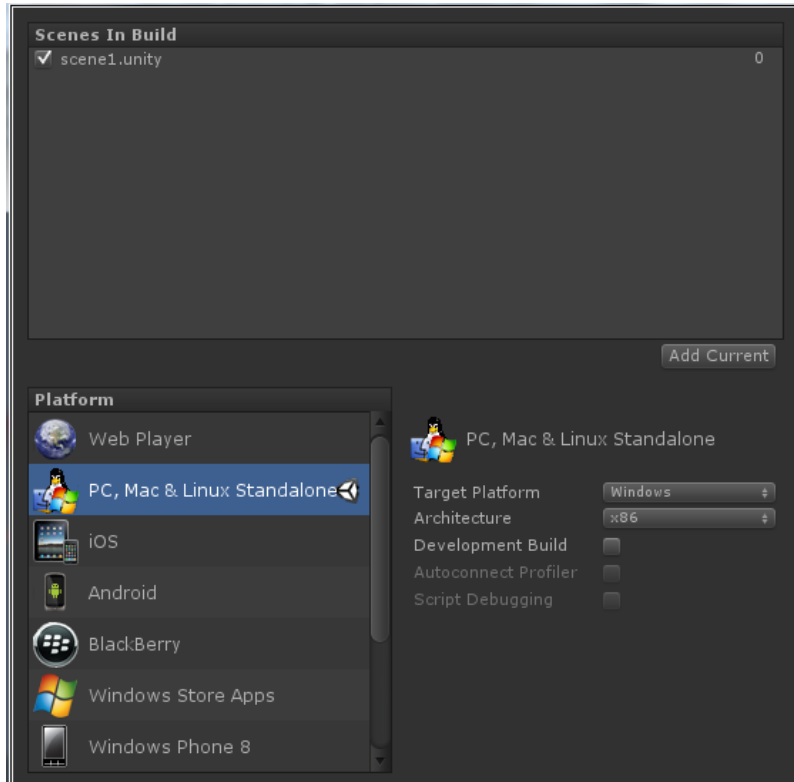
Building



File

Build & Run

Building - Platform Chooser



- This lets you generate a standalone exe
- Tip: make sure scene is in build.
- Look at “Scenes In Build” window.

Scripting

- UnityScript (similar to JavaScript)
- Boo (similar to python)
- C#

Which language to use?

Answer: C#

- interoperability with libraries written in C#
 - MiddleVR is written in C#
- less errors
 - JS Implicit variable declaration can cause problems
- advanced features available if needed
 - Properties, delegates, lambdas, generics

Comments

// for single line comments

/* for multi line comments */

Variables

- Various Data Types
 - bool, char, byte, float, double, int, string, ...
- Public vs Private keyword
- Pass by reference: use ref instead of &

Example:

```
public int playerHealth=5;
```

Functions (called methods in C#)

- allows us to compartmentalize and reuse code effeciently

```
public int SquareANumber(int number)
{
    return number * number;
}
```

If Statements

```
if (m > 10)
{
    //do something
}
else
{
    //do something
}
```

Basic Functions

```
void Start()  
{  
    //called once. do any initialization here  
}  
void Update()  
{  
    //called every frame. move objects here  
}
```

For Loop

```
for (int i = 1; i <= 5; i++)  
{  
    Debug.Log("hello");  
}
```


Example - Rotating Cube

```
void Update()  
{  
    transform.Rotate(0,speed,0);  
}
```

Example - Rotate Cube Properly

```
void Update()  
{  
    transform.Rotate(0,speed*Time.deltaTime,0);  
}
```

MiddleVR

- Commercial Plugin
- Support for 3D interaction devices
- Stereoscopy (active, passive)
- Multi-screens / multi-computers
 - (needed for CAVE-type systems)

MiddleVR Licensing

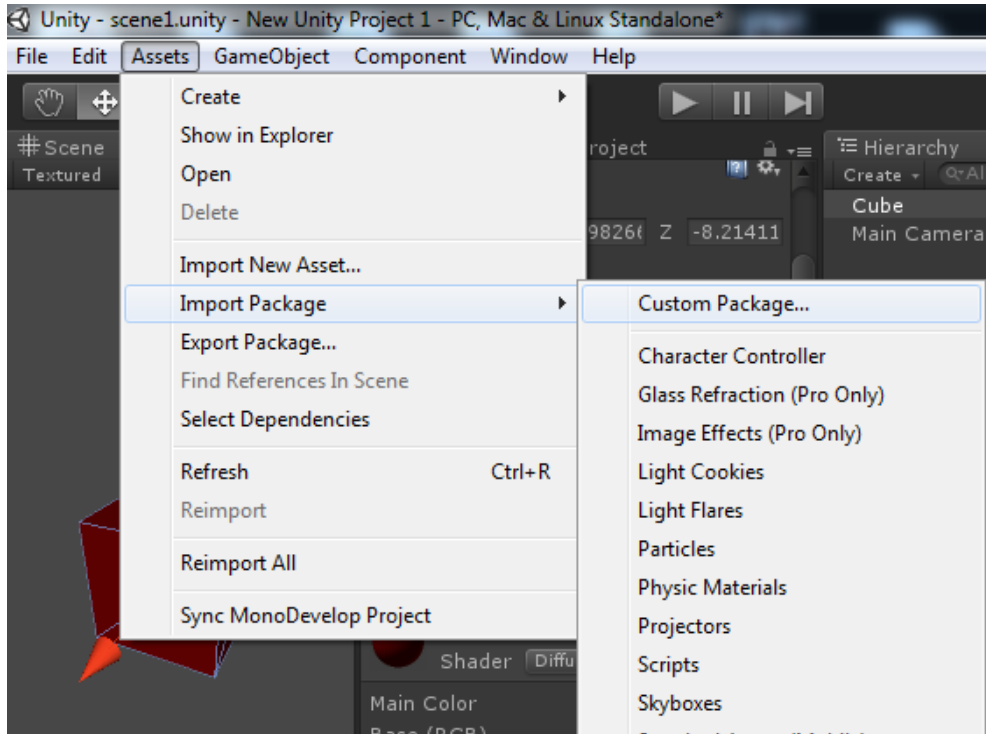
Free Version:

- export for oculus
- simulator for desktop development

Academic (licensed) Version:

- Build/Export to zSpace
- Build/Export to DiVE
- Can be node locked or floating license.

Add MiddleVR to Unity Project

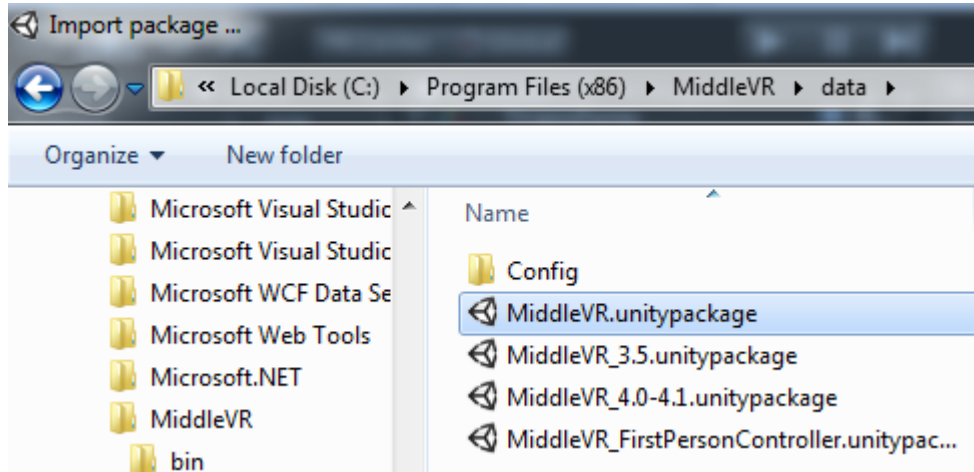


Assets

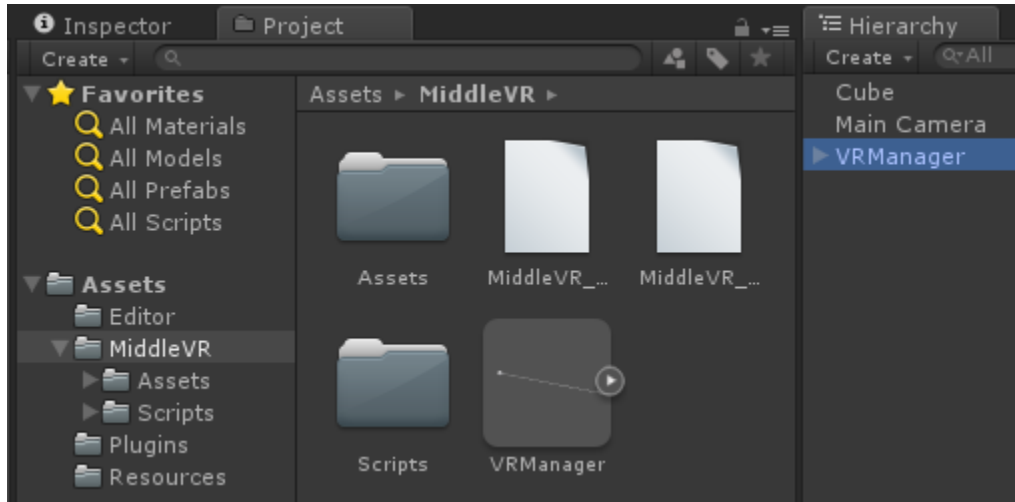
Import Package

Custom Package

Find the MiddleVR Package

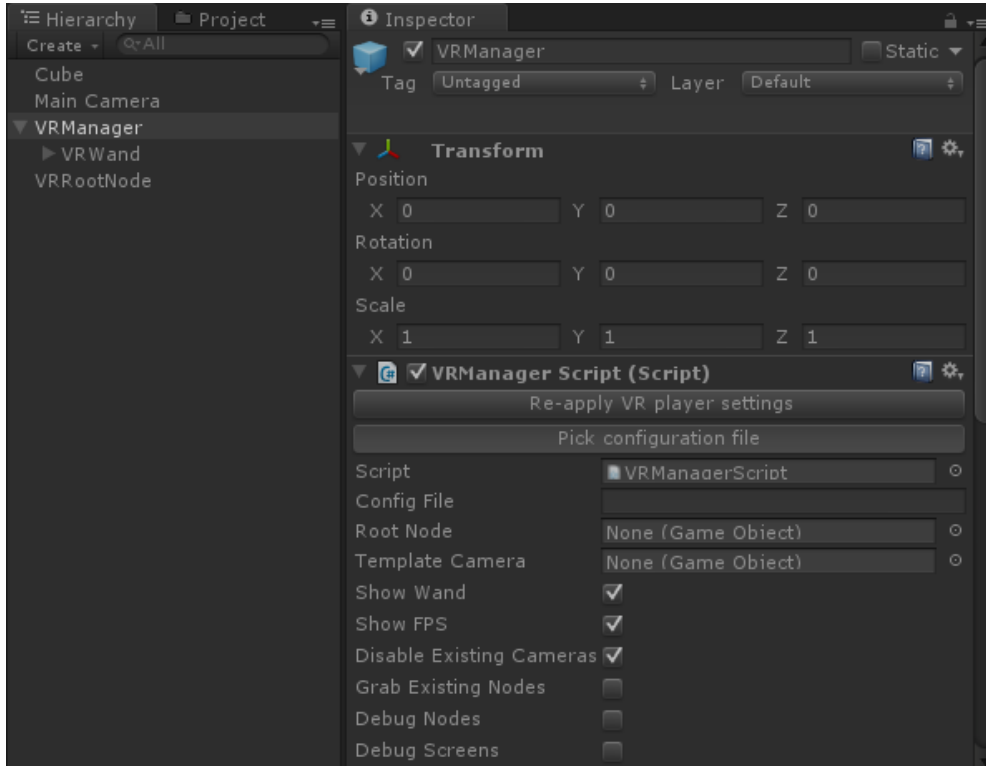


Adding to Scene



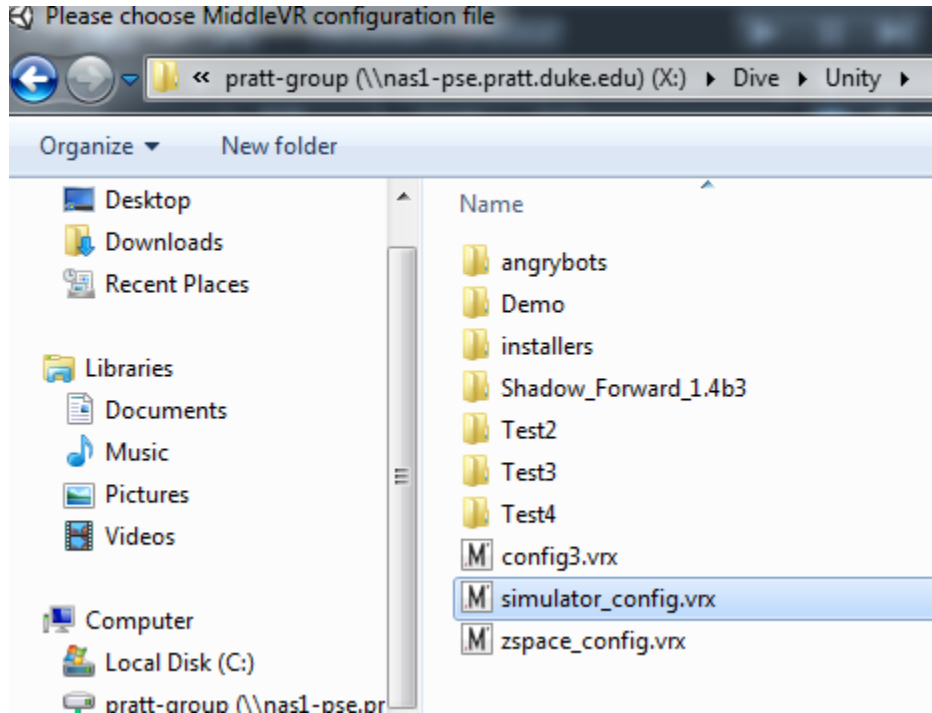
Need to drag
“VRManager”
from project to
Hierarchy window

Pick Config File



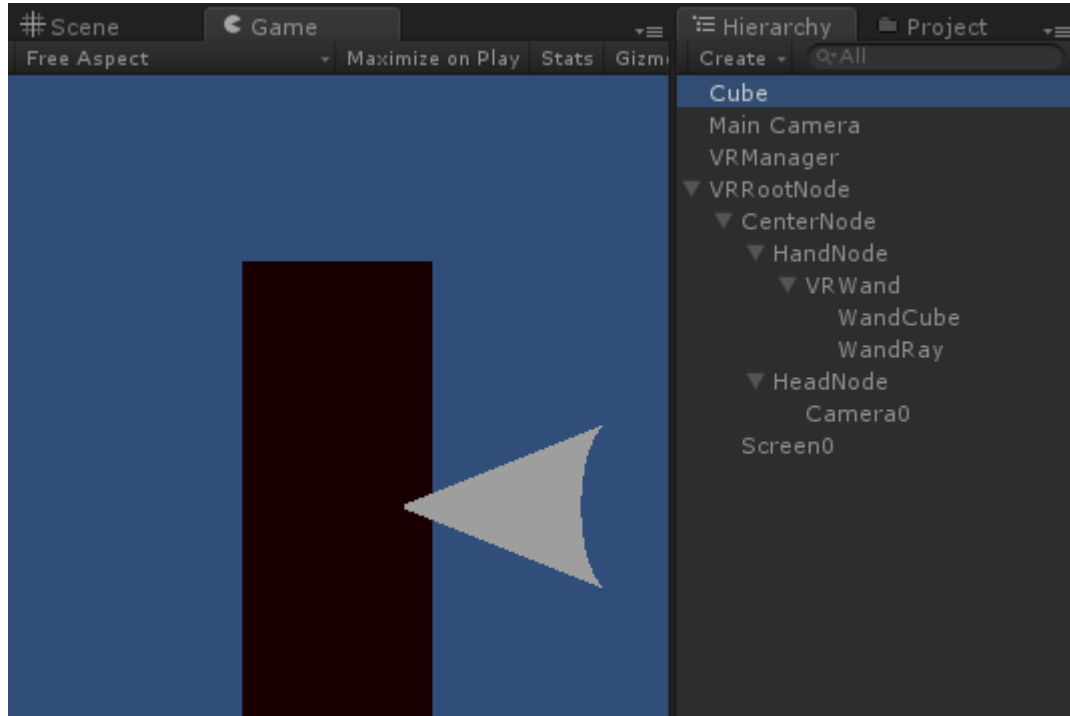
- Select VRManager.
- Many options in inspector window.
- Pick configuration file.

Pick Config File - Part 2



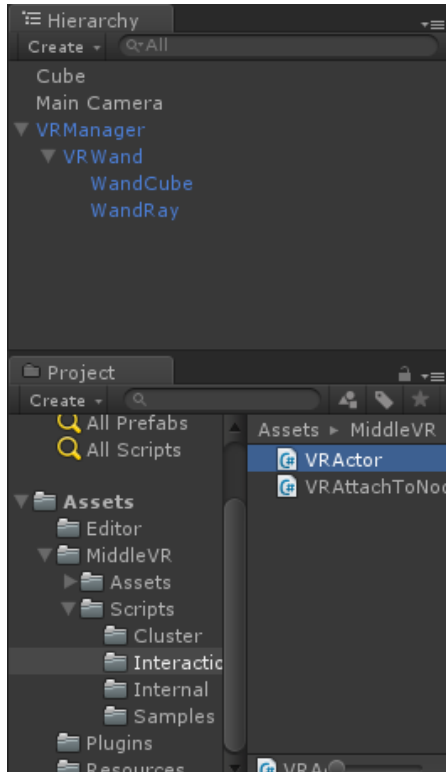
- I've been working (in progress) on some config files.
- Located on the DiVE fileserver.

Now Click Play



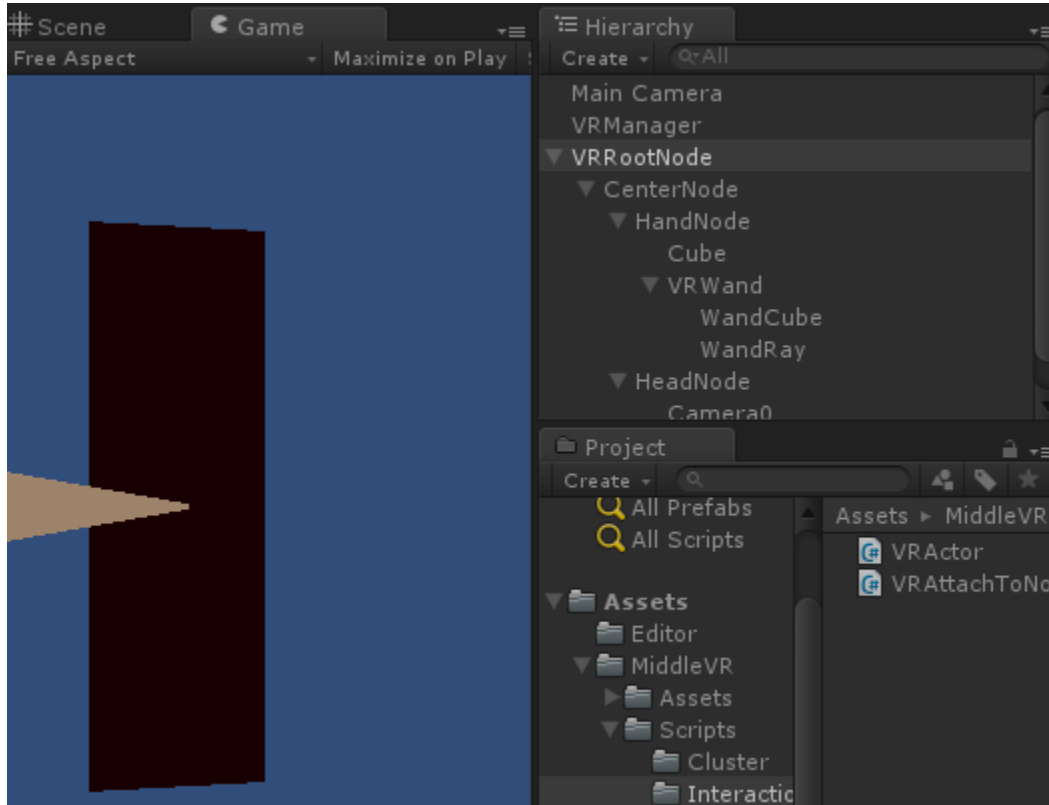
We can see that VRRootNode is created dynamically.

Virtual Hand Technique



- MiddleVR has easy ways to enable Virtual Hand technique.
- Add VRActor script to object we want to pickup

Virtual Hand Demo



- touch an object and click, the object is moved underneath the handnode

Learn More

Unity

<http://docs.unity3d.com/Documentation/Manual/index.html>

Walker Boys Studio - Video tutorial

http://www.walkerboystudio.com/html/unity_training___free___html

Lynda

<http://www.lynda.com/Unity-3D-tutorials/Unity-3D-35-Essential-Training/96677-2.html>

Middle VR

<http://www.imin-vr.com/doc/current/>

Thank You!

Questions?