

---

# BEGINNING VIRTUAL REALITY DEVELOPMENT

@MISSLIVIROSE



## Virtual Reality

NOUN

*the computer-generated  
simulation of a three-dimensional  
image or environment that can be  
interacted with in a seemingly real  
or physical way by a person using  
special electronic equipment*



# THE VR DEVICE ECOSYSTEM

DESKTOP // MOBILE

# HEAD MOUNTED DISPLAYS

Head Mounted Displays (HMDs) can be desktop-powered devices or mobile-powered devices

- Desktop: Separate display device self-containing screen, content provided externally from a separate PC
- Mobile: Content provided and displayed by a smart phone inserted into the casing



# DESKTOP VIRTUAL REALITY

Powered from external computers

Hardware includes separate screens

Treated as an external display at the hardware level

Software runs on the desktop and is passed to the display

Examples: Oculus Rift, HTC Vive, FOVE

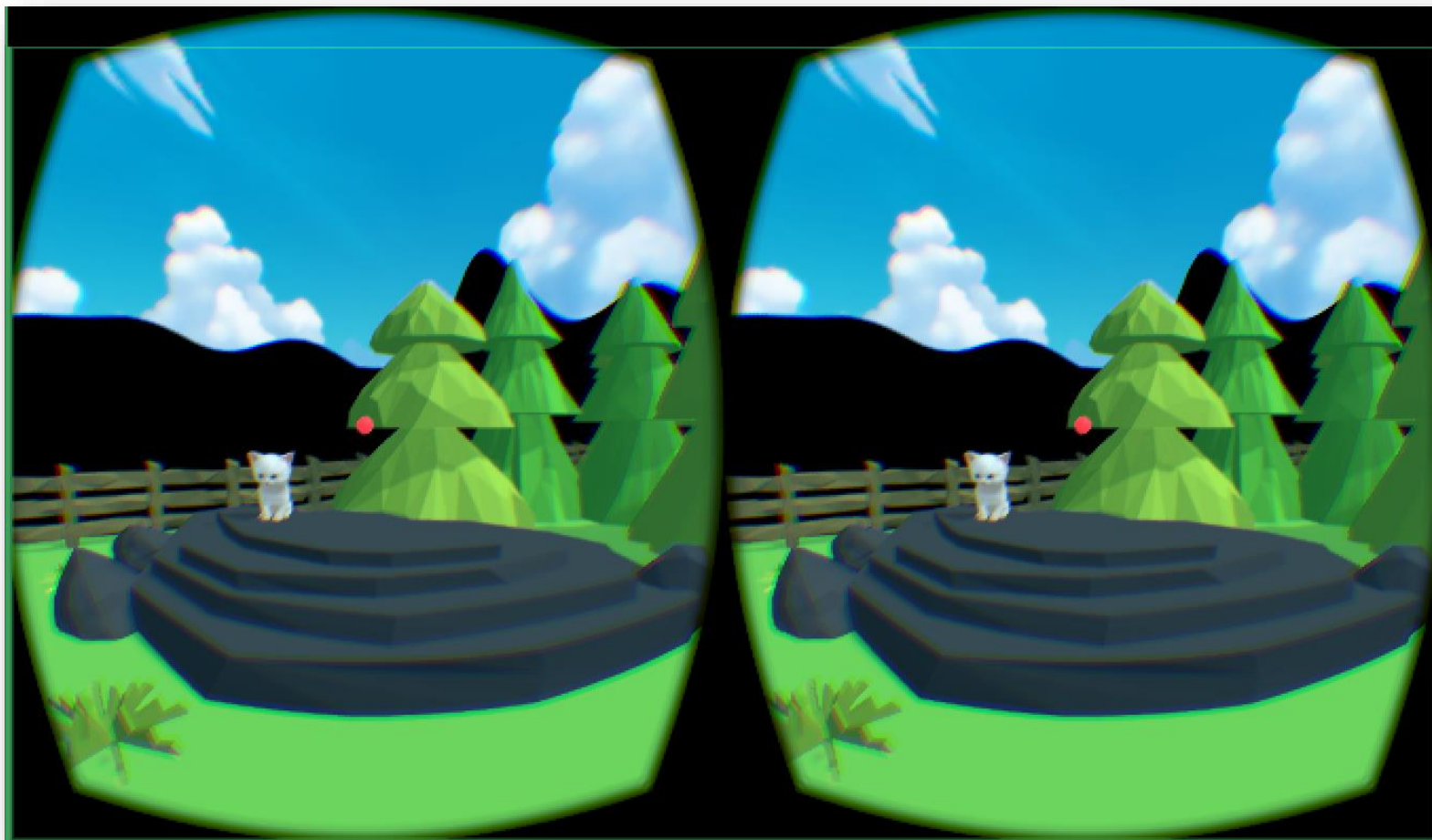
# MOBILE VIRTUAL REALITY

- Headsets are containers that support stereoscopic rendering for mobile phones
- Applications are typical smartphone apps
- The display is the phone screen
- Usually, almost all computational aspects are done on the phone rather than the headset
- Examples: GearVR, Homido, FreeFly, Cardboard



# OVERVIEW OF A VR APPLICATION

FEATURING KITTENVR



## KittenVR

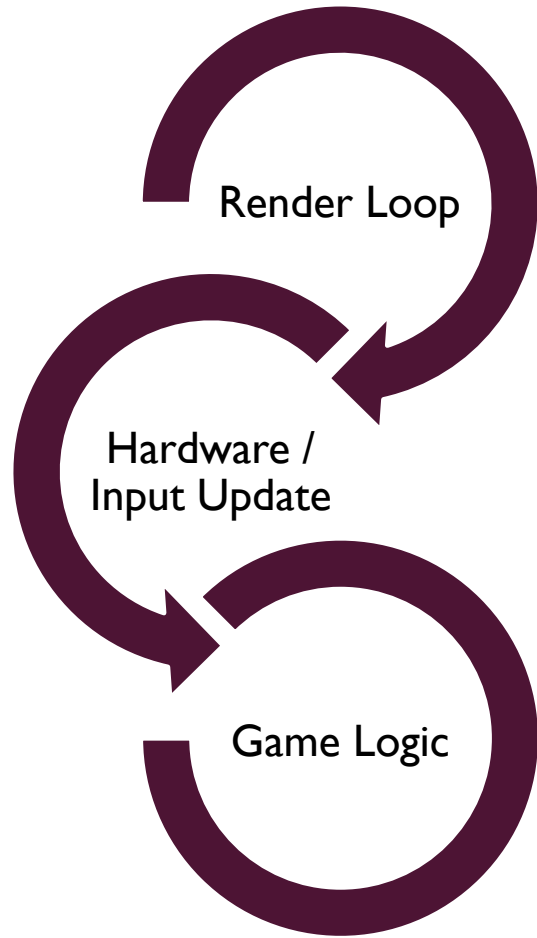
Made with Unity

C# Scripting

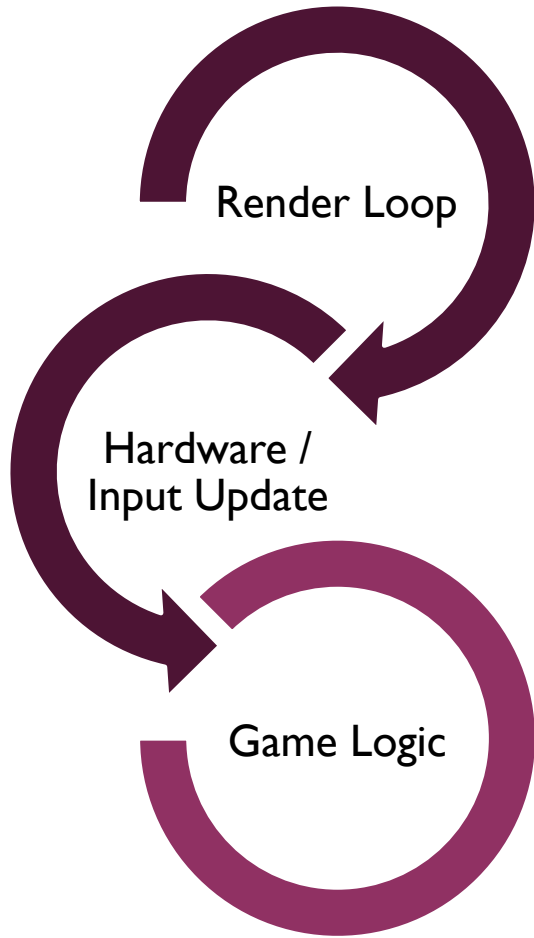
Oculus Rift (PC)

[www.kittenvr.com](http://www.kittenvr.com)





# Lifecycle of a VR App



# Game Logic

- Defines the behavior of your application components
- Models and textures are defined for your *scene*
- Scripts control behavior of the scene objects
- Generally doesn't rely on the status of the hardware, but will likely contain checks for changes

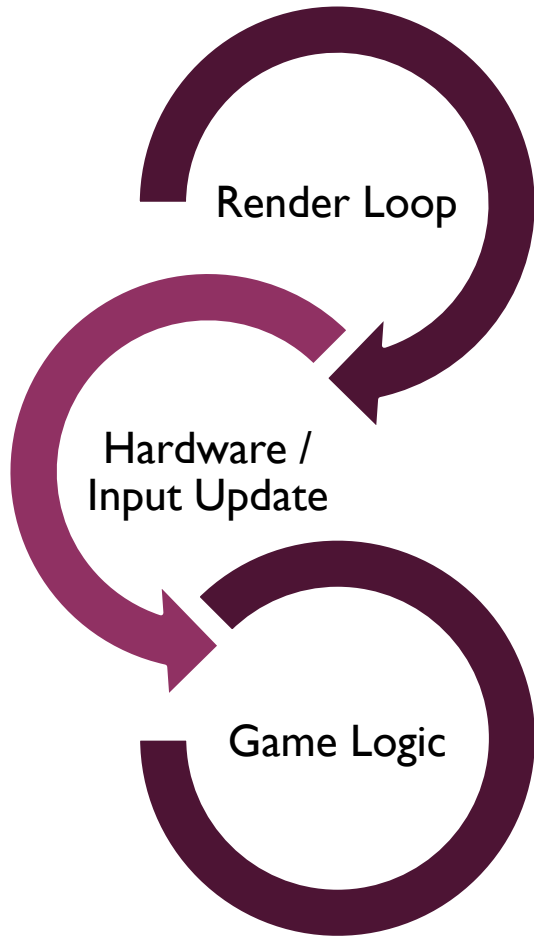
## KittenVR

Cartoon Environment

Kitten AI Behavior script

Gameplay script

Oculus Character Controller



# Hardware / Input

- Check on the orientation of the device to calculate how the camera should render objects in the scene
- Evaluate if there are actions triggering different behaviors in your scene
- Update scene variables or components based on input checks

## KittenVR

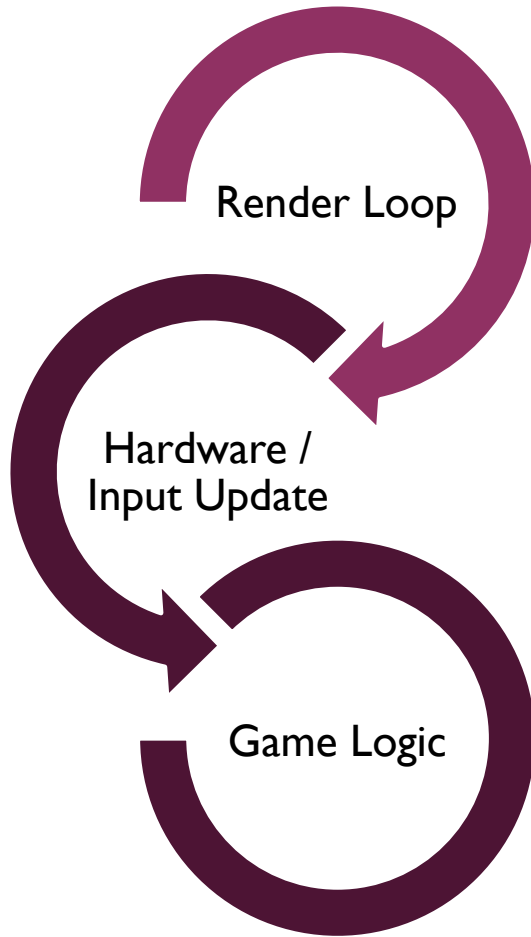
Check for Oculus orientation change

Analyze for Xbox button presses matching the “collect” action

Move character with joystick changes

# Render Loop

- Evaluate the scene appearance based on the orientation of the headset
- Apply effects based on changing scene (shadows, lighting, etc.) dynamically to match the user's expectation
- Draw to the headset / display
- Example: If player has moved under a tree, update the shadows cast by the player accordingly



## Playback

## Required Refresh Rate

Film / TV

24 – 48 FPS

Console Gaming

30 FPS

PC

50-60 Hz

VR

75-120 Hz

# PRESENCE IN VR

- Consider how your user will want to interact with the environment
- Environment scale will impact how realistic a scene feels to a player
- Non-autonomous character actions will break the sense of self



# DISPLAY PERFORMANCE & CONSIDERATIONS

- Low Framerates = Motion Sickness
- “Tearing” around the edges of the display
- Shadowing (duplication) of scene objects at high framerates
- Motion blur



# TOOLS OF THE TRADE

UNITY, UNREAL ENGINE, WEBVR (OH MY!)

# DEVELOPMENT OPTIONS

- Native development with the Oculus SDK
- Game Engines:
  - Unity
  - Unreal
- Web:
  - WebGL VR support through WebVR
  - Three.js

## KittenVR

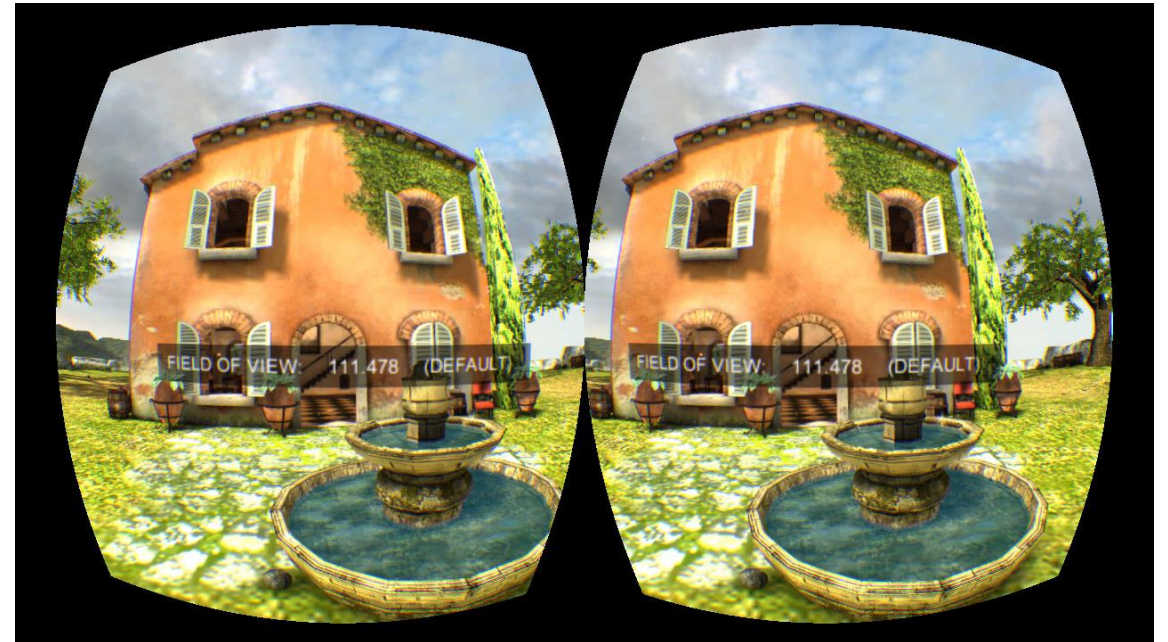
Used the Oculus Unitypackage plugin for Unity 4.6

Upgraded to use built-in option for VR support in Unity 5.X



# OCULUS SDK

- Add stereoscopic, distortion, and update effects within the graphics pipeline of your application
- Render to a texture specifically output to the Oculus device as well as to a non-stereoscopic screen view
- Great for custom game engines adding in VR support with the Rift
- More customization and control granted
- Versions available for desktop and mobile



Source: [developer.oculus.com](http://developer.oculus.com)

# UNITY

- Game Engine from Unity Technologies
- Visual editor for environment building & scripting
- Languages: C#, UnityScript (a JS derivative), Boo



# UNITY:VR MODE

- Unity 5.1+ has support for VR built in
- Requires Oculus Runtime (available for Windows)
  
- Implementing it:
  1. Edit -> Project Settings -> Player
  2. Under “Other Settings” check the ‘Virtual Reality Supported’ box
  3. Will look normal until the Oculus is plugged in

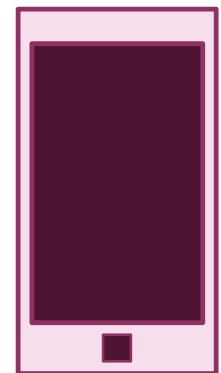
# UNREAL ENGINE

- Game Engine from Epic Games
- Visual editor for environment building & scripting
- Language: C++



# WEB VR

- Flexible back end development
- Front-end Rendering:
  - WebGL
  - Three.js
- Desktop & mobile support with one codebase
- [www.webvr.info](http://www.webvr.info)



# BENEFITS OF WEBVR

- No installation of application is needed – run in the browser on any device
- Automatically support mobile and desktop devices
- Utilize the current set of tools and libraries for JavaScript
- Easily switch between VR and non-VR mode
- Offload interfacing with hardware to the browser

# DOWNSIDERS OF WEBVR

- Performance hits for non-native graphics
- Non-standard support for some browsers
- Less insight and control into bugs with hardware
- Relies on browser detection of devices

# PLANNING A VR APPLICATION

- Evaluate how the medium lends itself to your game or application
  - Minimize the amount of locomotion
  - Choose one platform to target at a time
- Read up on best practices, but don't lock things in
- Prepare to iterate - a lot!



# IMPLEMENTATION OF A VR APP - MY STRATEGY

- Build out the basics in your environment so that you have a rough understanding of what your application components will do
- Script the behaviors for gameplay elements & the rest of the application components
- Finalize the environment with lighting, textures, effects

# TESTING YOUR VR APP

- One person QA is not quite enough!
- Variations in height, gender, life experiences, comfort with technology, etc. will all impact your user to a much greater extent in VR
- Performance is hugely important to prevent motion sickness

# READY TO BUILD?

- 3D Development
- Scripting
- Animation & 3D Modeling
- Download Unity / Unreal

## MY RESOURCES

- Blog: <http://livierickson.com/blog> for short-form VR development tutorials, reviews, and news
- GitHub: <http://github.com/misslivirose> for all the code in my VR experiments (including Kitten VR!)
- Informatica Feminale course materials: [http://misslivirose.github.io/informatica\\_feminale](http://misslivirose.github.io/informatica_feminale)
- Learning Unity: <http://livi.link/learnunity>