

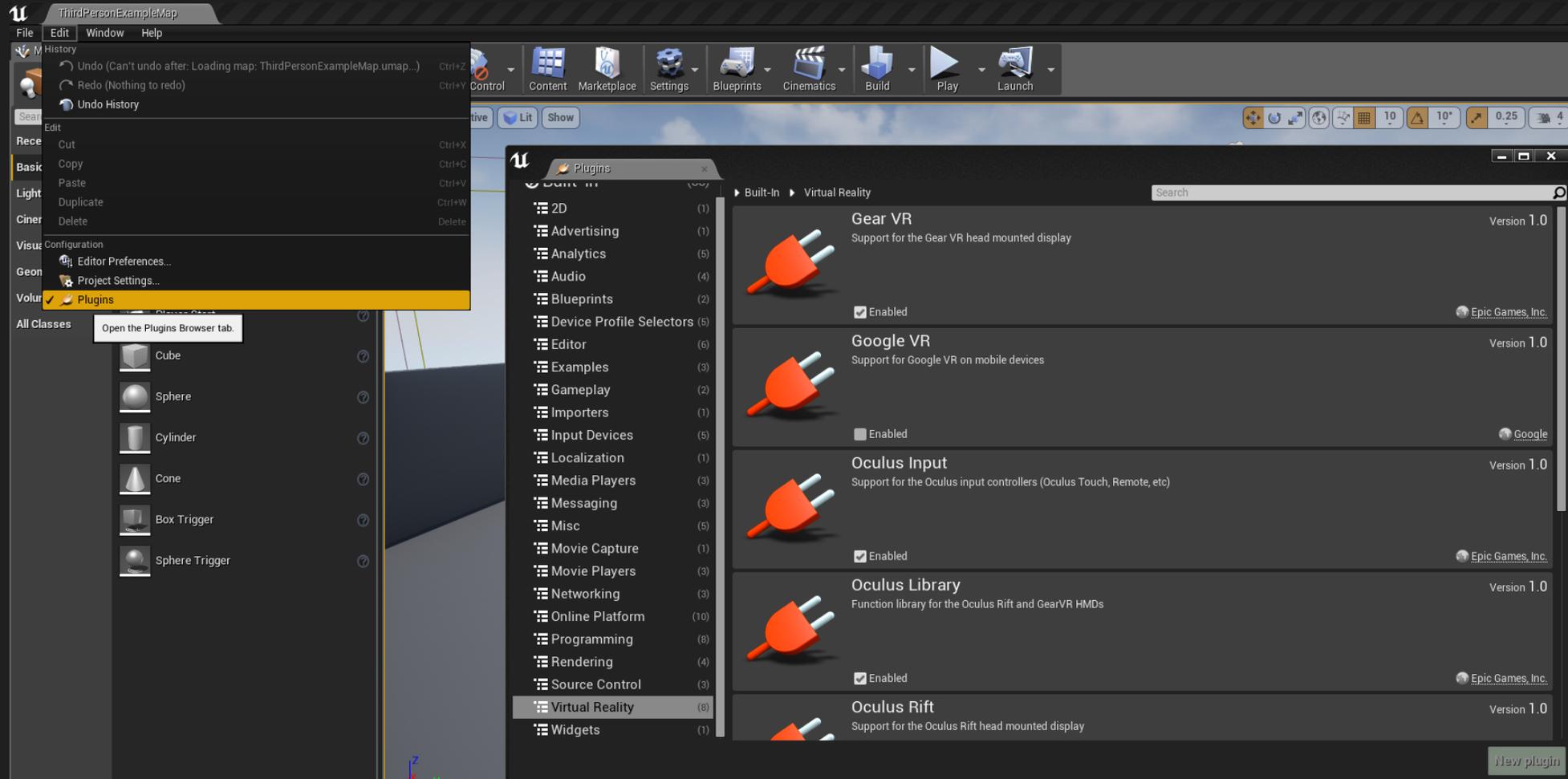
Shipping a VR Game with **UNREAL ENGINE 4**



VR Platform Support



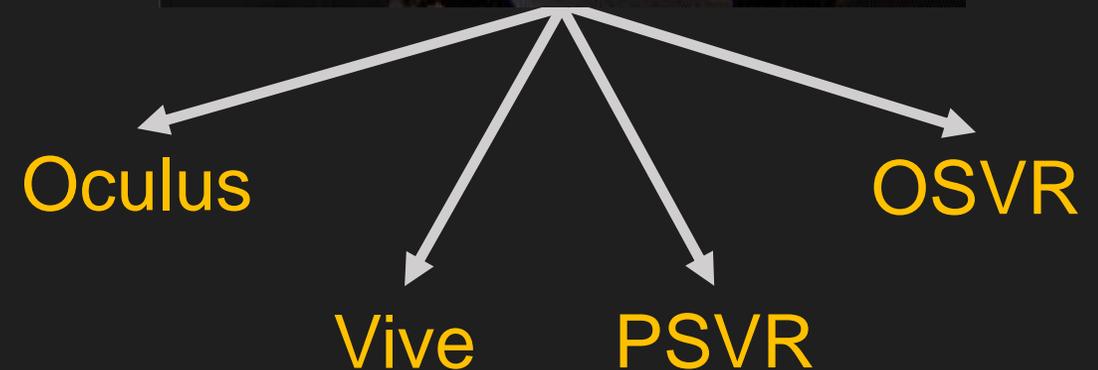
VR Platform Support



VR Platform Support

All of these platforms go through UE4's **common VR interfaces**, so you can **make your content once**, and **deploy it anywhere**.

- Unified Camera System
- Motion Controller System
- Optimized rendering paths
- Low-latency optimizations



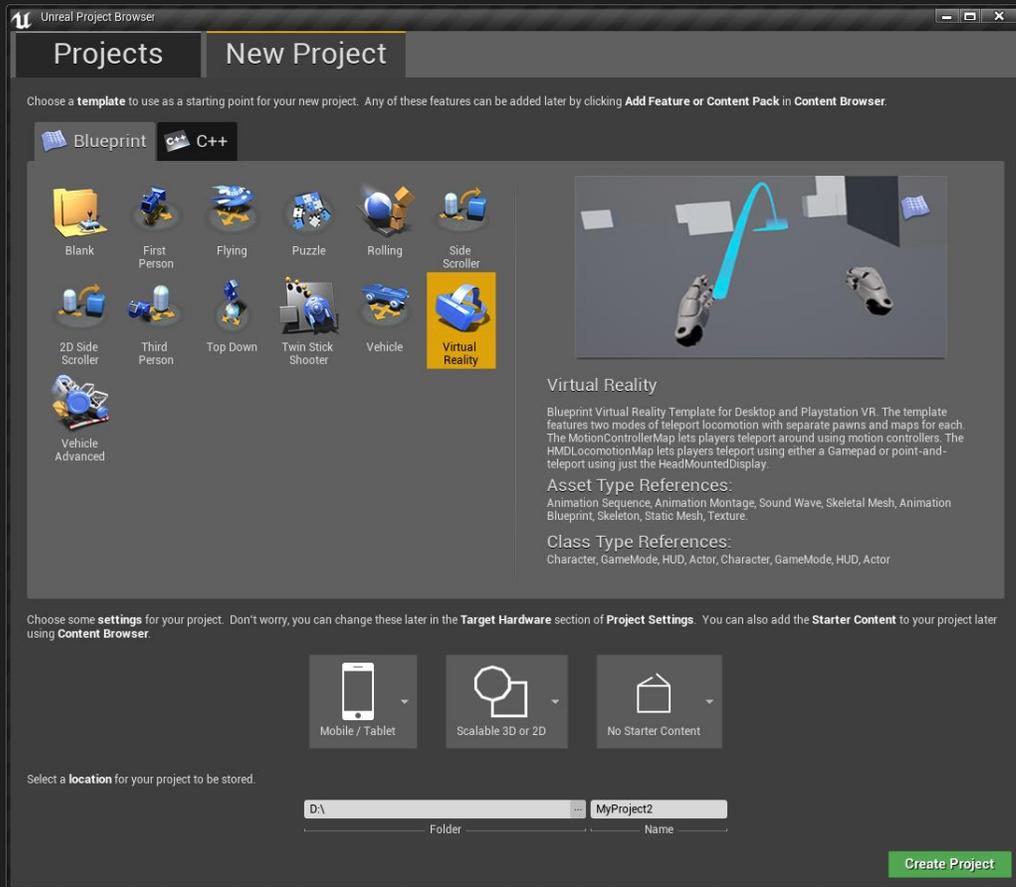
VR Preview

You can quickly play your game projects in VR with the **VR Preview** option right off the **Main Toolbar**.

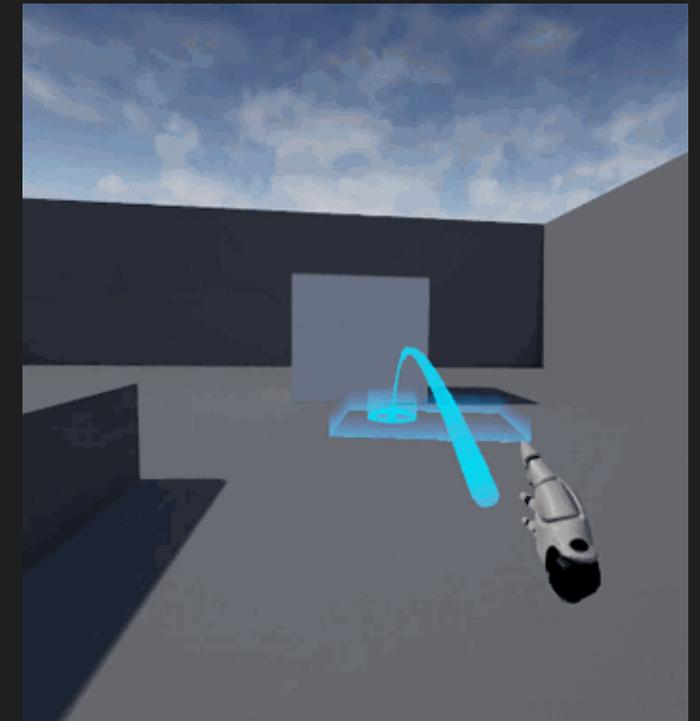
- Launches Connected Device
- HMD Tracking Auto Enabled
- Device **MUST** be connected prior to loading your project



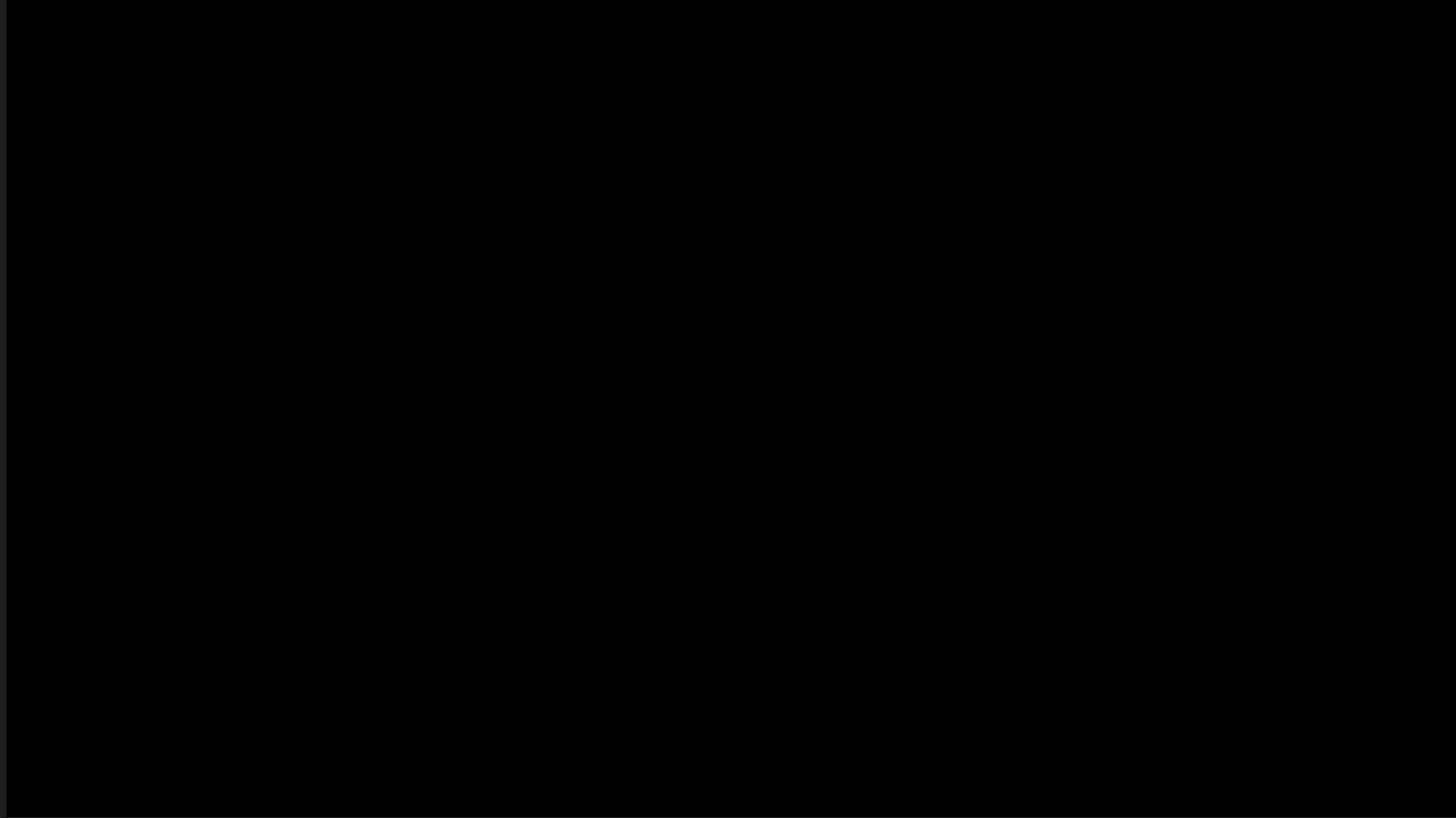
Where Do I Start?



New - 4.13 VR Template



VR Editor – Create VR in VR



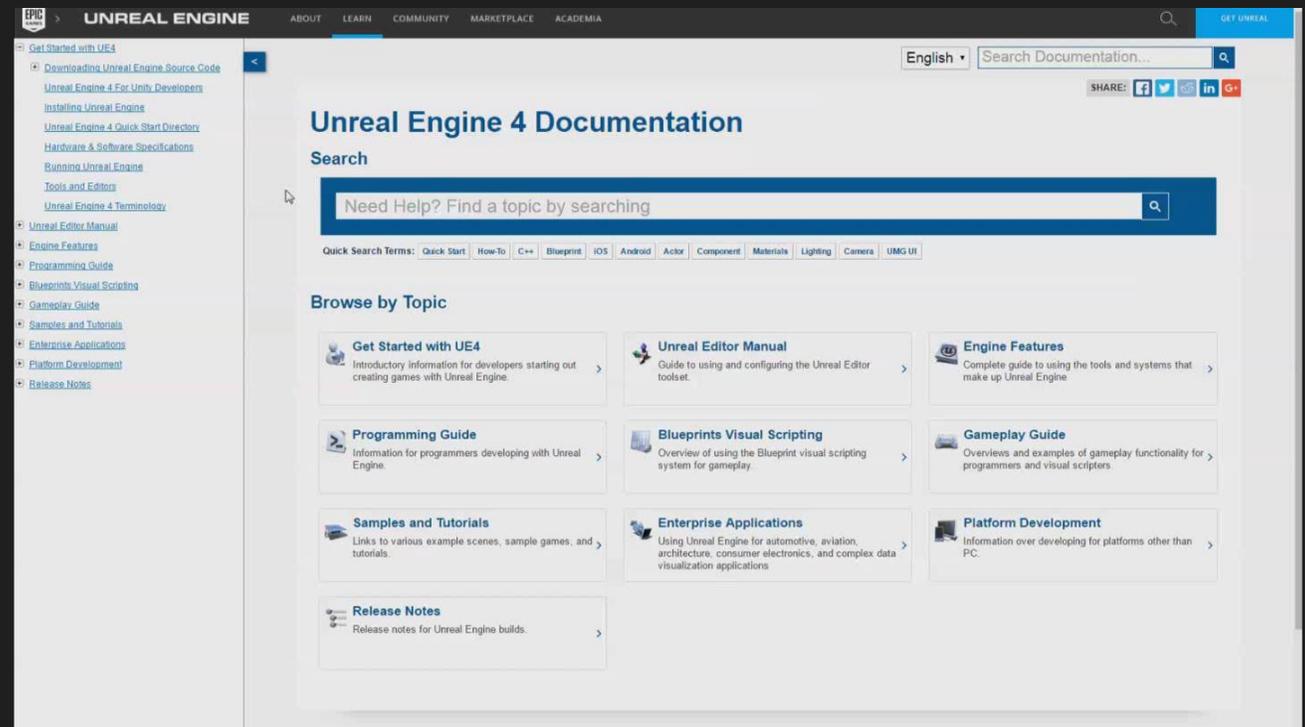
Enabled from **Editor Preferences** -> **Experimental** -> **Enable VR Editing**



https://docs.unrealengine.com

The **Unreal Engine 4 documentation** pages offer many different forms of **Learning Resources** for you to choose from.

- Getting Started Guides
- Step-by-step Tutorials
- Technical Reference Pages
- Video Tutorials
- Sample Projects
- Updated Weekly!



https://answers.unrealengine.com

Ask on AnswerHUB

- Post/View Questions/Answers
- Browse by Topic
- Epic Staff Responses
- Great for Specific Questions

The screenshot shows the Unreal Engine AnswerHub website. The top navigation bar includes the Epic Games logo, 'UNREAL ENGINE', and links for 'ABOUT', 'LEARN', 'COMMUNITY', 'MARKETPLACE', and 'ACADEMIA'. A search bar and a 'GET UNREAL' button are also present. Below the navigation, there's a 'UE4 ANSWERHUB' section with a user profile 'Wes Bunn 4.9k', a 'Post A Question' button, and a search input. A welcome message states: 'AnswerHub is a resource for developers to help each other succeed with UE4. If you have a problem, it's a great place to search for solutions and ask questions. If you have knowledge to share, want to establish yourself as an expert, or make connections, please jump in and answer questions. Though Epic keeps an eye on questions being asked, AnswerHub only works with your participation!'. Below this, there are tabs for 'Recommended Questions', 'All Questions', 'Hottest Questions', and 'Unanswered Questions'. The 'Recommended Questions' section lists three questions: 'Splitscreen toggle' (Blueprint Scripting, UE 4.13), 'Blueprinting RESOLVED' (Blueprint Scripting, UE 4.12), and 'MasterServers with blueprinting' (Blueprint Scripting, UE 4.12). A sidebar on the right contains a 'Using the UE4 AnswerHub' section with a 'Quick Start Help' button and a 'Sections' list including 'Installation & Setup', 'Using UE4', 'Rendering', 'Packaging & Deployment', 'Blueprint Scripting', 'C++ Programming', 'Legal & Licensing', 'Bug Reports', 'Everything Else', 'Marketplace', 'Japanese', and 'Korean'.



<https://www.twitch.tv/unrealengine>

Training & Community Live Streams

- Topic Based Live Training
- Live Q & A
- New Feature Breakdowns
- Developer Demos
- Every Tues/Thurs a 2PM EST



Tips for VR Development

Maintain your Target Frame Rate

For a VR experience to feel smooth, your game needs to run **at 75 fps (Oculus DK2) or even 90 fps** (Vive and Oculus Retail) depending on the device. To see the current frame rate type in **“stat fps”** or **“stat unit”** (for more detailed breakdown) in your console when running the game.

HMD Device	Target Frame Rate
DK1	60 FPS
DK2	75 FPS
Rift Retail	90 FPS
Vive	90 FPS
Gear VR	60 FPS
PSVR	Variable up to 120 FPS

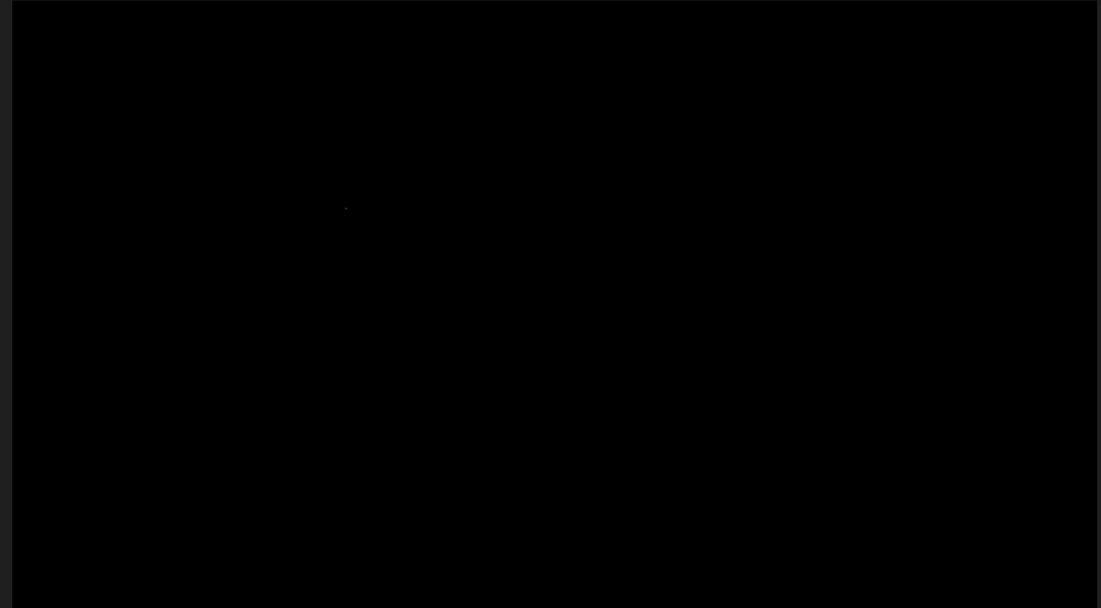


Tips for VR Development

Enable **VR Instanced Stereo Rendering**

Lets us use a single draw call to draw both the left and right eyes, saving CPU (and some GPU) Time.

- In *Bullet Train*, we saved about 1.75ms on the CPU, and 0.75ms on the GPU.
- Currently works on **PC** and **PS4** with support for other devices soon.
- Enable it from the **Rendering / Project Settings**.

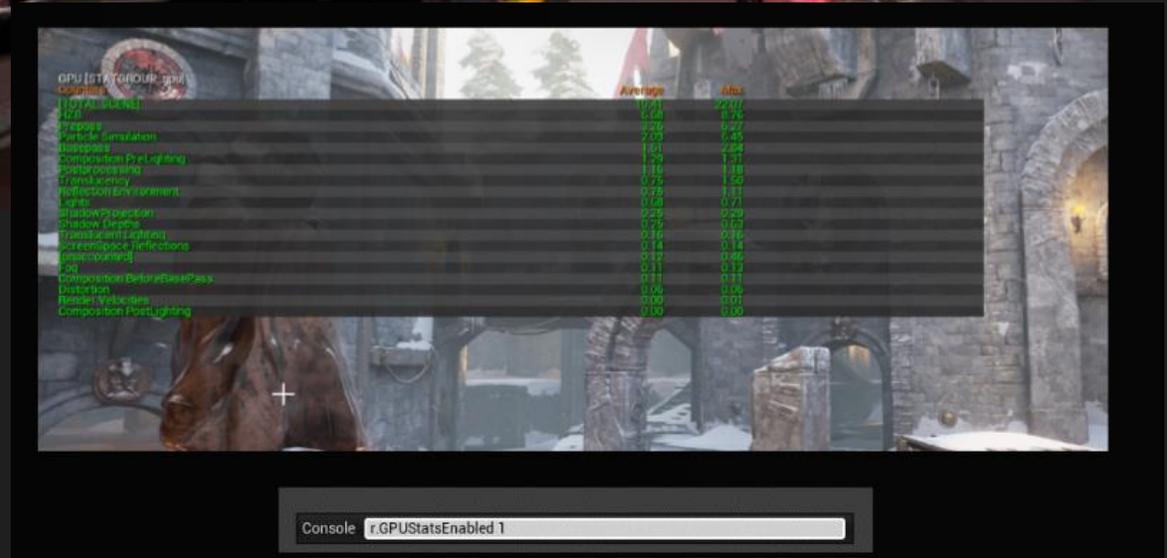


Tips for VR Development

Performance is **KEY!**

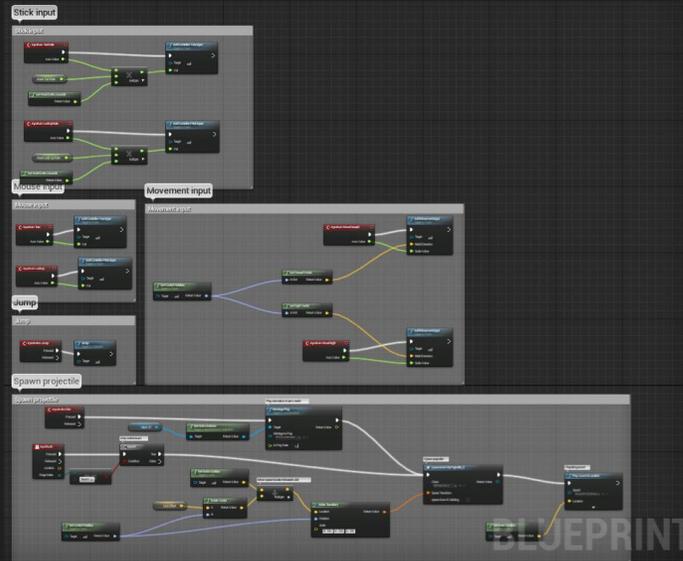
Use the **Profile Tools** early and often.

- **CPU Profiling**
 - Stat Commands
 - stat SceneRendering
 - stat Game
- **GPU Profiling**
 - GPU Visualizer (**Ctrl+Shift+Comma**)
 - Experimental Real Time GPU Profiler
 - r.GPUStatsEnabled 1 in the console
 - stat GPU



Tips for VR Development

Convert Expensive Blueprints to C++ Code



As of 4.12, you can begin experimenting using the **Blueprint to C++ Conversion** for **extra performance** on console and mobile platforms. We'll be improving support in the future!

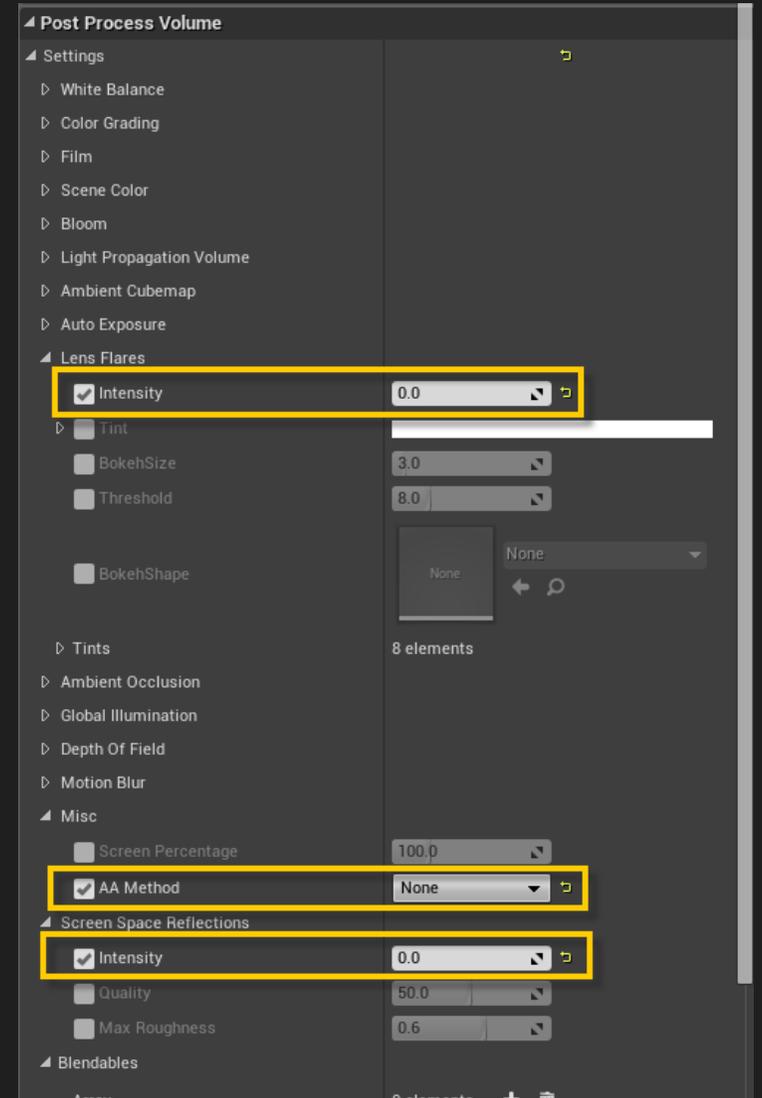


Tips for VR Development

Disable Heavy Post-Processors

Rendering requirements for VR is demanding and many of the advanced **Post Processing features** that are enabled by default **should be disabled**.

- Lens Flares
- Screen Space Reflections
- Temporal AA
- Screen Space Ambient Occlusion
- Bloom

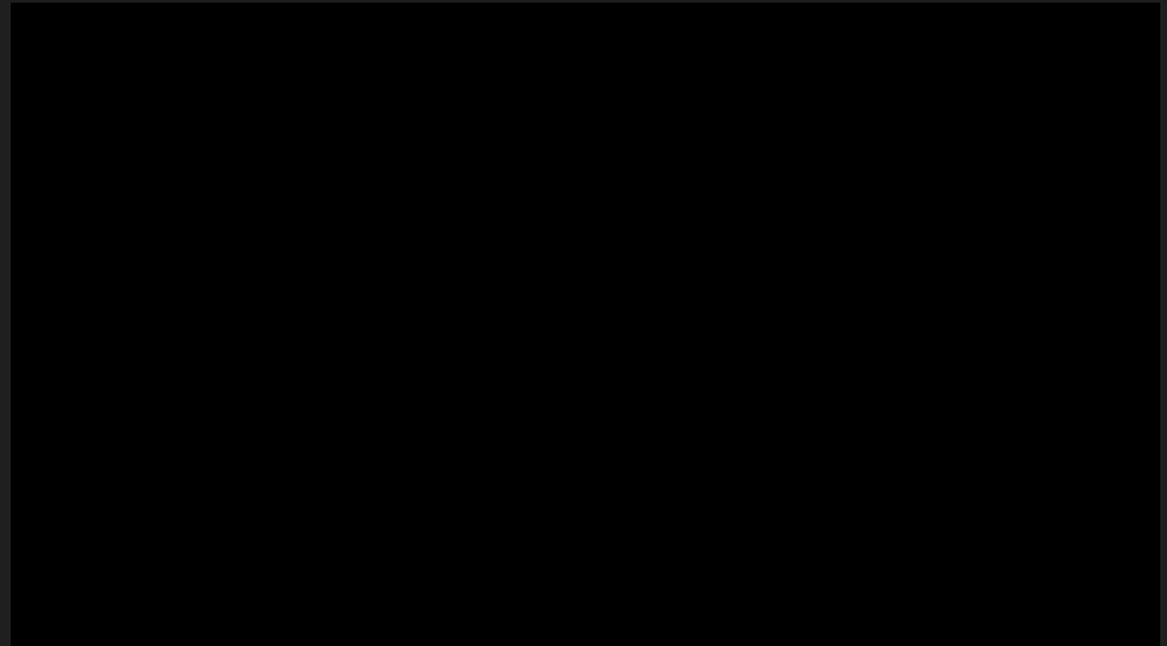


Tips for VR Development

Lighting Tips for VR

Use **Static Lighting** over Stationary or Dynamic, **baked lights are the best** option for VR environments.

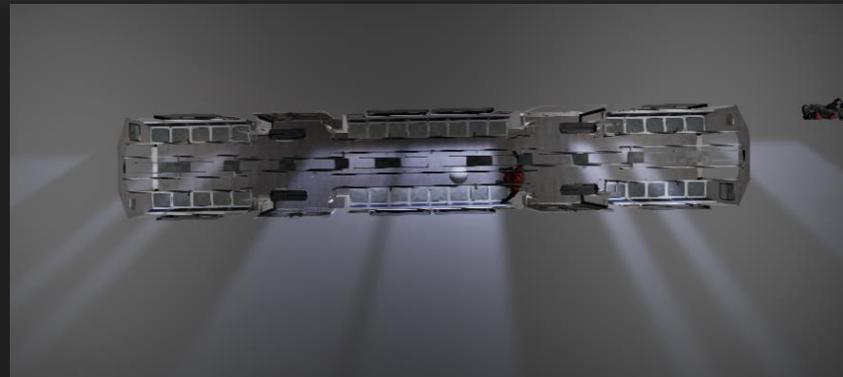
This scene used originally 70+ dynamic lights! **We utilized the GBuffer in order to fake the Lighting**, which is something we also did in our Showdown Demo.



Tips for VR Development

Lighting Tips for VR

Since we know the scene normals and depth, we can project a pre-made light / shadow texture onto the world. Much more efficient!



Tips for VR Development

Lighting Tips for VR

- Make sure your Stationary / Dynamic Lights **do not overlap**.
- If using Dynamic Shadows, only have **one shadowing light**.
- Dynamic Objects should have “**Single Sample Shadow from Stationary Light**” enabled.
- Use **Stat LightRendering** to see current lighting costs.
- Use **Reflection Capture resolution limits** and stick to them.



Tips for VR Development

Effects for VR

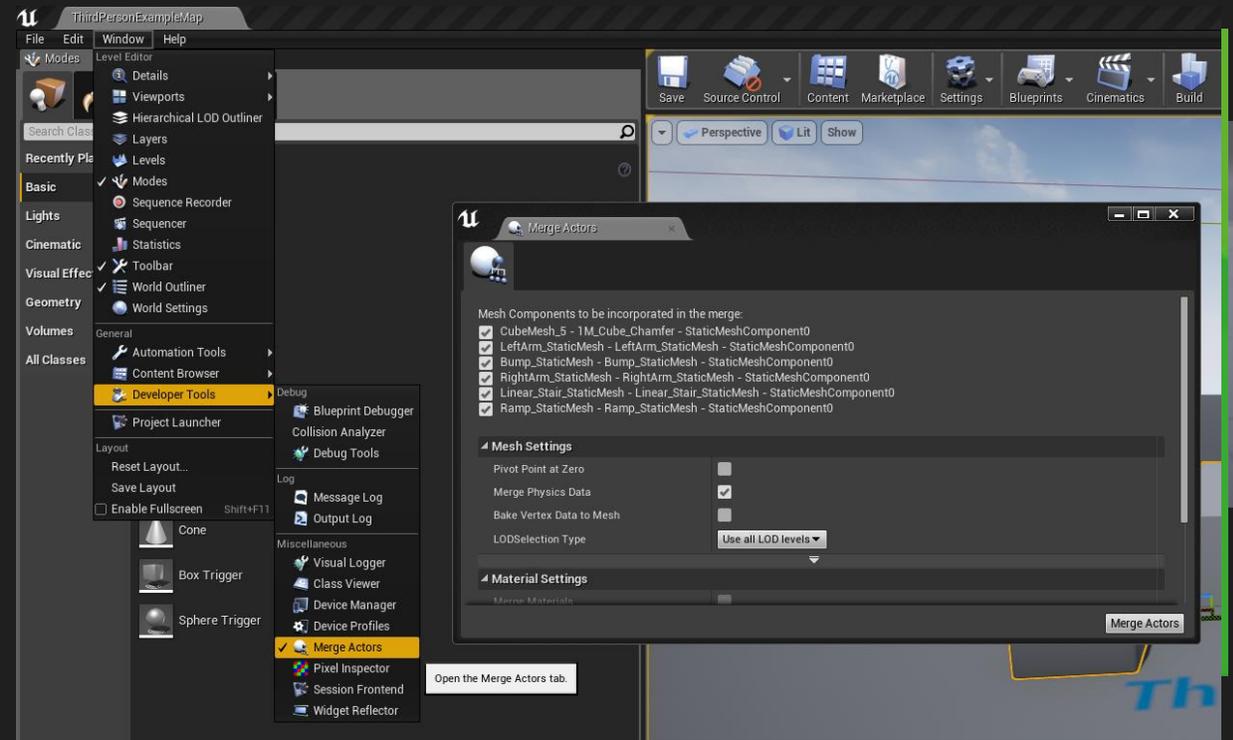
- **Mesh based** VFX work the best for VR.
- Camera Facing particles **do not hold up well** in VR.
- Use the **Dither Temporal AA Material Function**:
 - Can make Opacity masked objects look Translucent.
- Use **Stat Particles** console command.
- Infinity Blade Effects Pack via Marketplace – **Free Stuff!**



Tips for VR Development

Environment Tips

- Use **Reflection Probes** instead of Screen Space Reflections
- Use **HLOD** and aggressive **LOD**
- Be mindful and **limit Material usage**
 - **Alt + 8** for Shader Complexity
- Do not hesitate to **Merge Assets**
- Do not be too modular
 - (**Big elements = less draw call**)
- Use **stat SceneRendering** for info!



THANK YOU!



We're CloudGate Studio



Jeremy Chapman - CCO
@thejerenator

Steve Bowler - President
@gameism



You Might Remember Us From



Simple Mechanics; Emotional Connection



Deeper Interactions; Greater Presence



We Shipped Two Steam VR Games in Five Months With Two People

How would you go about doing that?

- Allow fear to take hold and paralyze you
- You're going to have to be fast/nimble
- Throw out existing preconceptions about game development
- Assess the existing pipeline problems/Solve for those
- Use UE4 because it will make your life easier



The Practice Run

Two Guys. Three Weeks. One 90FPS Demo

The Brookhaven Demo was made in 3.5 Weeks

- Simple Interacts: Shoot All The Things
- Lack of Movement
- Buy or Scrounge for All Assets
- Managed entire store presence/PR/marketing



Can Lightning Strike Twice

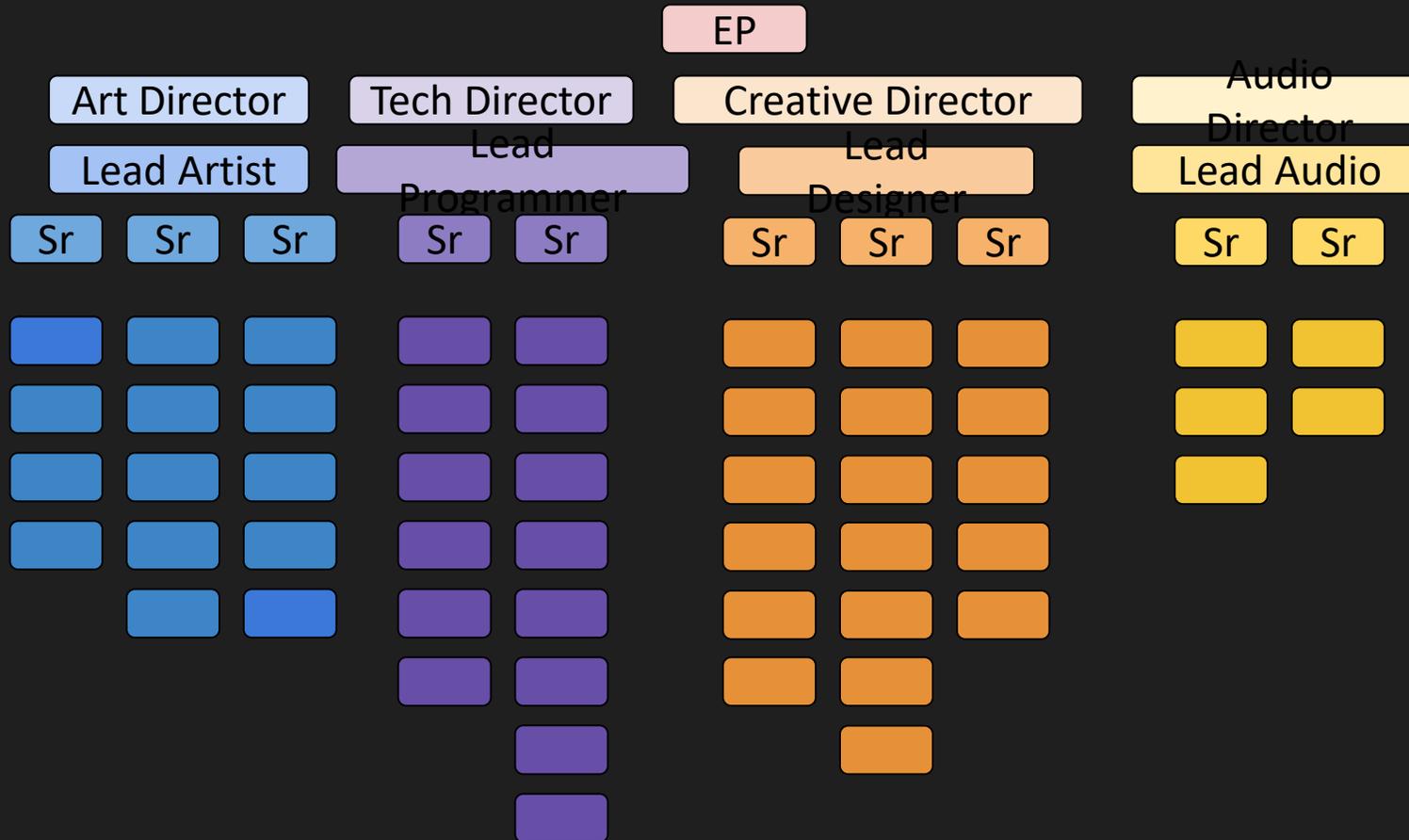
How Can We Expand On What We Learned

Can We Get a Sellable VR Game Out in 3 Months?

- Need to Prototype at Lightning Speed
- Innovative Movement + Interact Mechanics
- Purchase All Content Assets
- Manage not only store but payroll/taxes too
- Try not to die



AAA Game Dev Teams/Pipelines = Too Slow



- Inflexible
- Glacial Decisionmaking
- Low Risk/High Return



We're Going To Have To Redefine "Team"



- Producers: Just, no.
- Art Team: Too slow for what we need.
- Programmers: Too expensive. Also too slow.
- Audio: Sorry.
- UI: You're also cut.
- "But What About": We cut them



Content is The Most Cost Ineffective Part of Game Development

- Assets take a lot of time to make and finalize
 - Art
 - Audio
 - Animation (and we're ex-Lead Animators!)
- Our Strength as a Studio lies elsewhere:
 - Mechanics
 - Presence



\$72



How Do We Make Up That Content Deficit?

ART: Marketplace Assets



Not All Art Assets Are Created Equal

- Asset may not meet your needs
- We have to “fix” every asset
 - Collision
 - Materials
- Contract an Art Friend

This is *still* the most cost effective



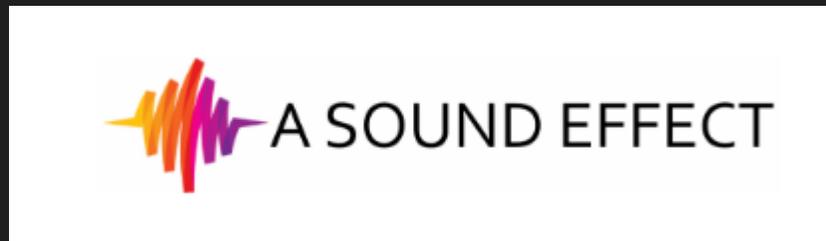
How Do We Engineer Audio?

AUDIO: Marketplace Assets

STOCK SOUND EFFECT
SFX
\$3
0:01 / WAV / STEREO / 294.3 KB
See item details
ADD TO CART \$3

RELATED RESULTS

- Cartoon Arp 0:02
- Cartoon Eek 0:03
- Duck Quack 02 0:01
- Cartoon Roll Up 0:02
- Future Boing 0:02
- Robotic Elephant 0:02
- Cartoon Pipe 0:03



CONTENT SEARCH

Home Categories

Asian Audio Bundle
Evil Mind
\$29.99
★★★★★ (2) Music

Asian 2 Audio Bundle
Evil Mind
\$59.99
Not Yet Rated Music

Puzzle Audio Bundle
Evil Mind
\$29.99
★★★★★ (2) Music



The Only Perfect Audio is Music

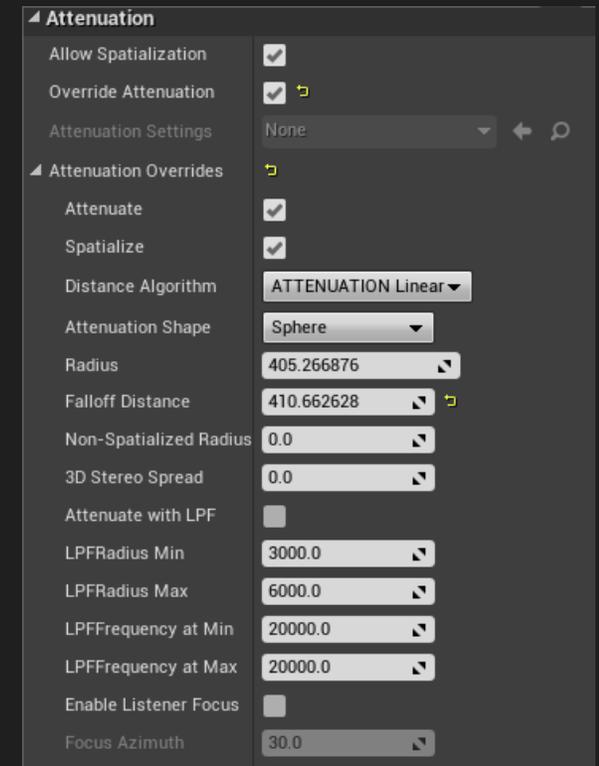
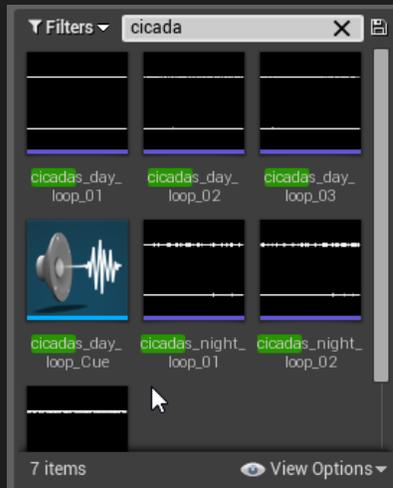
- We have to “mix” every asset
 - Dino sounds are always 3 animals blended
 - Most SFX require some “layering”
 - Audacity works: is free
- Music is only thing that plugs straight in
 - Downside: works this way for everyone else

This is *still* the most cost effective use of our time/money



Unreal Audio Does The Rest

- 3D Spatialization and Attenuation is Built In
- Literally Drag and Drop SFX In



How Do We Make New Features?

CODE/DESIGN: UE4 Blueprints

```
BaseCharacter.cpp
→ ABaseCharacter
    // Called every frame
    @void ABaseCharacter::Tick(float DeltaTime)
    {
        Super::Tick(DeltaTime);
    }

    // Called to bind functionality to input
    @void ABaseCharacter::SetupPlayerInputComponent(class UInputComponent* InputComponent)
    {
        Super::SetupPlayerInputComponent(InputComponent);
    }

    //Implement CalculateHealth
    @void ABaseCharacter::CalculateHealth(float Delta)
    {
        Health += Delta;
        CalculateDead();
    }

    //Implement CalculateDead
    @void ABaseCharacter::CalculateDead()
    {
        if (Health <= 0)
            IsDead = true;
        else
            IsDead = false;
    }

    //Implement PostEditChangeProperty
    @if WITH_EDITOR
    @void ABaseCharacter::PostEditChangeProperty(FPropertyChangedEvent& PropertyChangedEvent)
    {
        IsDead = false;
        Health = 100;

        Super::PostEditChangeProperty(PropertyChangedEvent);

        CalculateDead();
    }
#endif
```

Solution Explorer

- Solution: TwinStickShooter (2 projects)
- Engine
- Games
 - TwinStickShooter
 - Config
 - External Dependencies
 - Source
 - TwinStickShooter
 - BaseCharacter.cpp
 - BaseCharacter.h
 - TwinStickShooter.Build.cs
 - TwinStickShooter.cpp
 - TwinStickShooter.h
 - TwinStickShooter.Target.cs
 - TwinStickShooterEditor.Target.cs
 - TwinStickShooter.uproject

Properties

ABaseCharacter VCodeClass

C++	ABaseCharacter
(Name)	ABaseCharacter
File	c:\unreal\project\TwinStickShooter
FullName	ABaseCharacter
IsAbstract	False
IsObject	False
IsManaged	False
IsSealed	False
IsTemplate	False
IsValue	False

=



UE4 Blueprints Disrupt Game Development

Blueprints are faster than code for new features

- Democratizes development: Lowest Common Denominator
- Communication “Tech Barrier” is eliminated
- New Blueprints integrate easily
- No need for a “merge” and risk breaking the game
- GO BEYOND PROTOTYPING. SHIP BLUEPRINTS.

If you can think it, you can make it.



It's a Little Bit Like This

We Found a Language We Can Build Virtual Worlds With



UE4 Makes It Possible

CloudGate Would Not Exist Without UE4

- Codeless development wasn't possible in UE3
- We couldn't do what we do Unity (No Blueprints)
- Two Devs. Two Premium VR Games. Five Months.

You Can Do It, Too



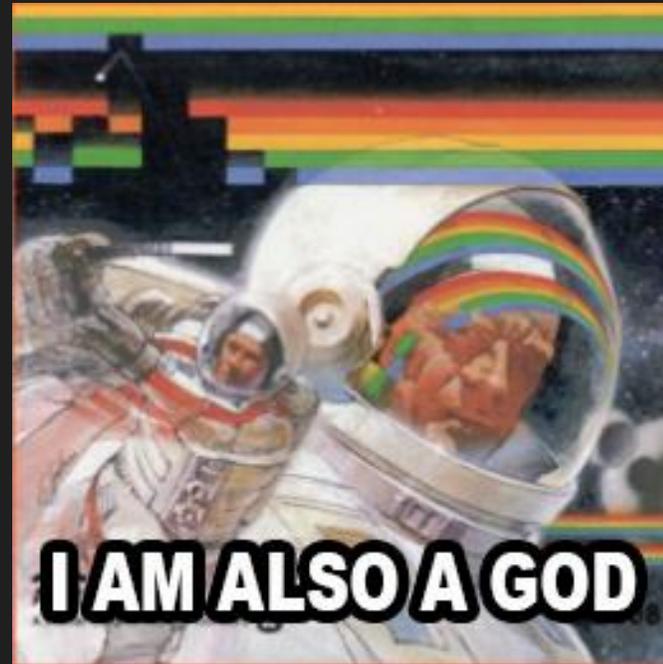
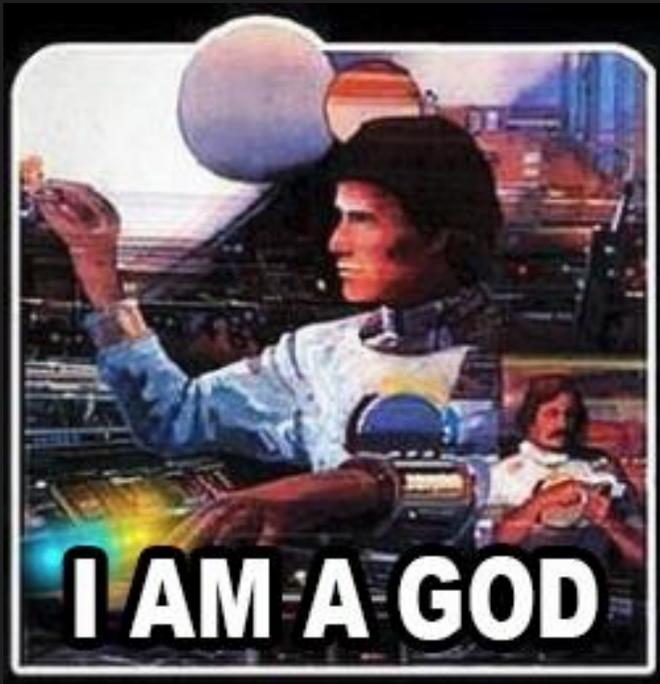
How We Build VR Worlds in UE4

Jeremy Chapman - CCO



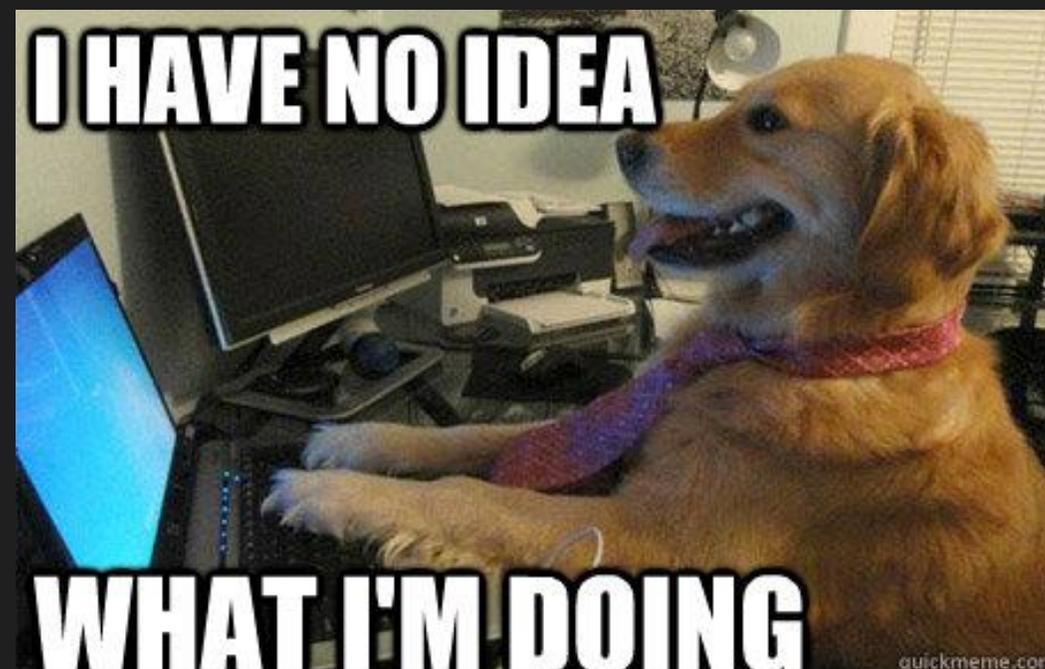
Our Reputation

How We Hope People Perceive Us



Full Disclosure

We Have No Idea What We're Doing



With That Out of the Way

We're Going to Show You How We Do It

- Fail Faster
- You don't know until you try
- Don't get attached
- "You're Going to Build it 2-3x Anyway"

-The CloudGate Motto



BluePrints for VR

Proven in Production

- Blueprints are powerful
 - You can ship them
 - Extremely rapid prototyping
 - Our VR Locomotion went through many stages of iteration very quickly
- You can create an industry standard in an afternoon



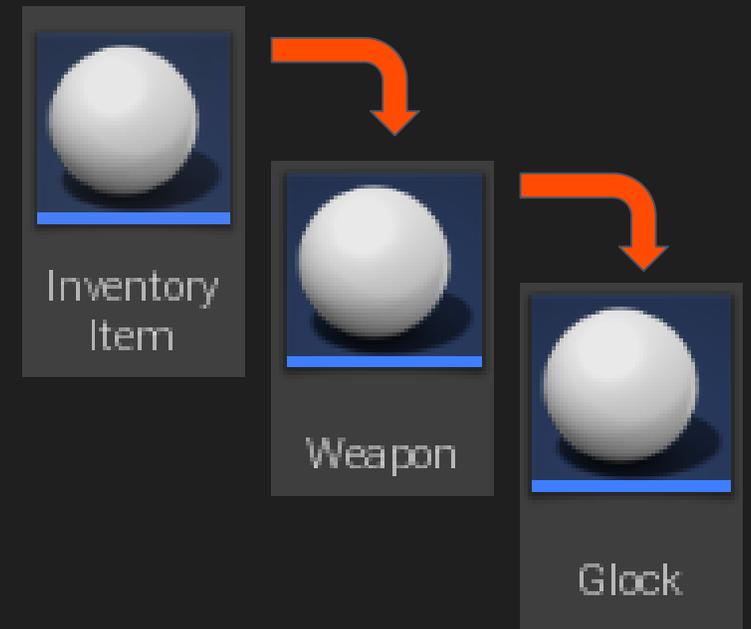
Problems Small VR Teams Face

- Avoiding Repetitive Gameplay
 - Be more than a tech demo
- Even Avoiding Repetitive Work
- Optimization and Performance is a never-ending battle
- No one else to blame:
 - “SOMEONE BROKE THE BUILD”
 - (it was probably you)



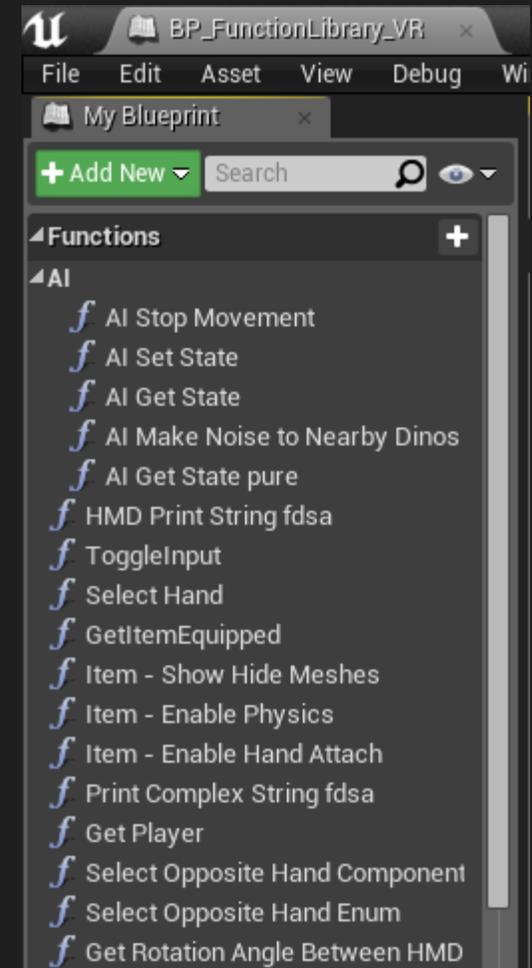
Solutions: Reusing Blueprint Actors

- Avoid single use BluePrints
 - Level BluePrints great for prototyping/proof of concepts
 - Can't be reused
- If you think you *might* use it elsewhere, make it a BluePrint class



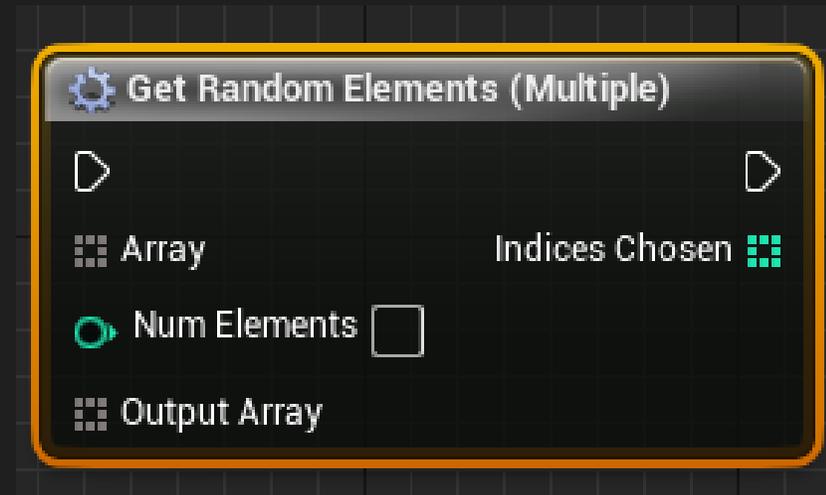
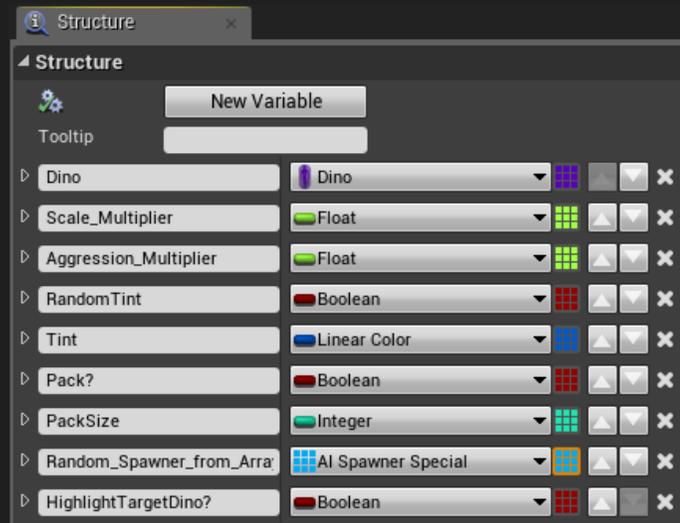
Reusing Tools

- Function and Macro Libraries
 - Create one tool, apply it to any actor
 - Saves time and effort
 - Easy changes/iterations
 - Makes debugging much easier



Variation

- Modifying parent class variables in child classes allows for easy procedural content
 - Randomize classes, amounts, colors, HP, speed, etc.



BluePrints Inside BluePrints

Yo Dawg, BluePrints Inception Joke Goes Here

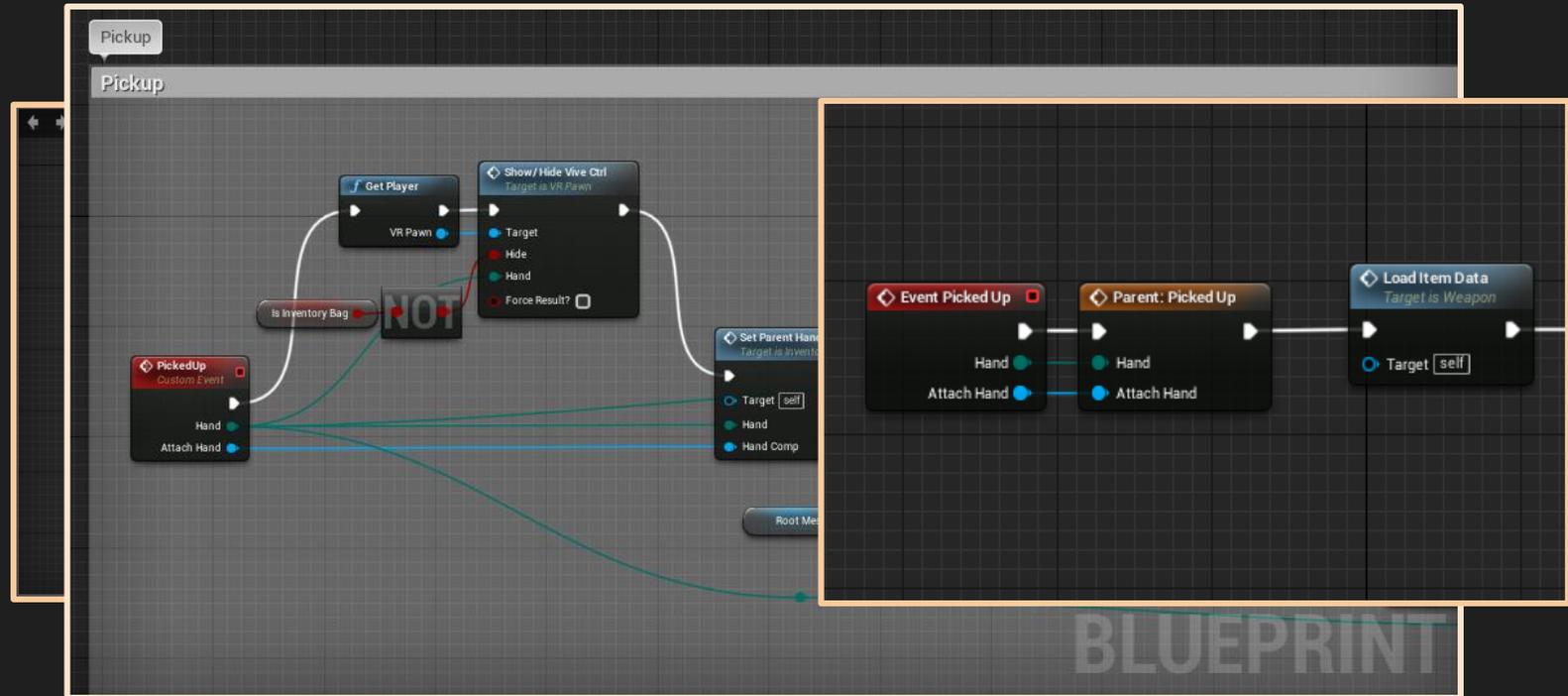
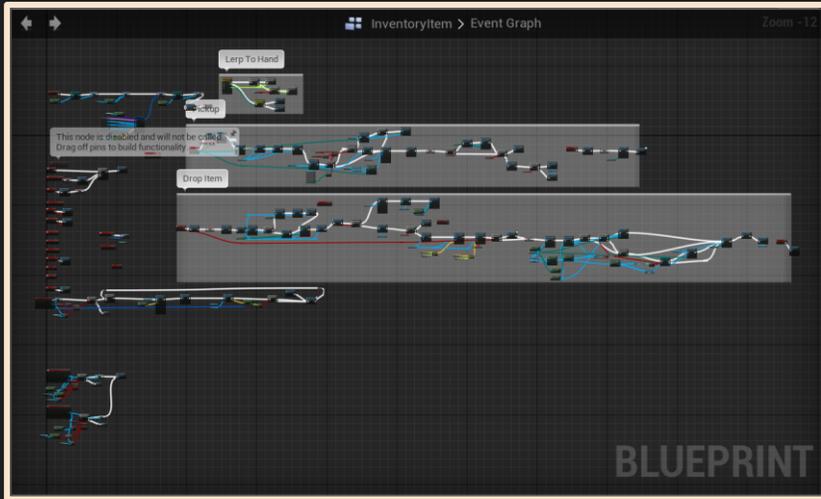
- Multiplying Your Force Multiplier
- Make your BPs Ala Carte
- Nest BluePrints as Child Actors Inside Larger BluePrints
- Faster Content Creation
- Simpler/Faster to Fix



Complex Parent Classes

And Their Simpler Children

ITEM CLASS



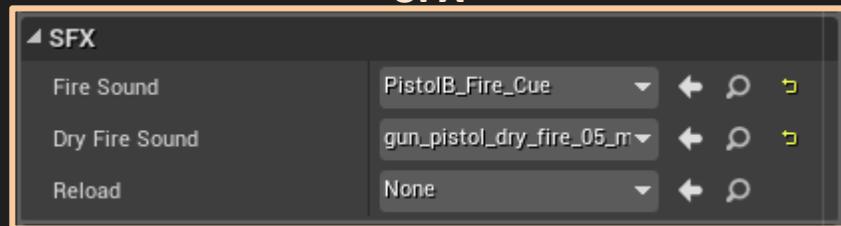
Keep It Simple, Stupid

Your Final Child Content Should Probably Just Be Vars

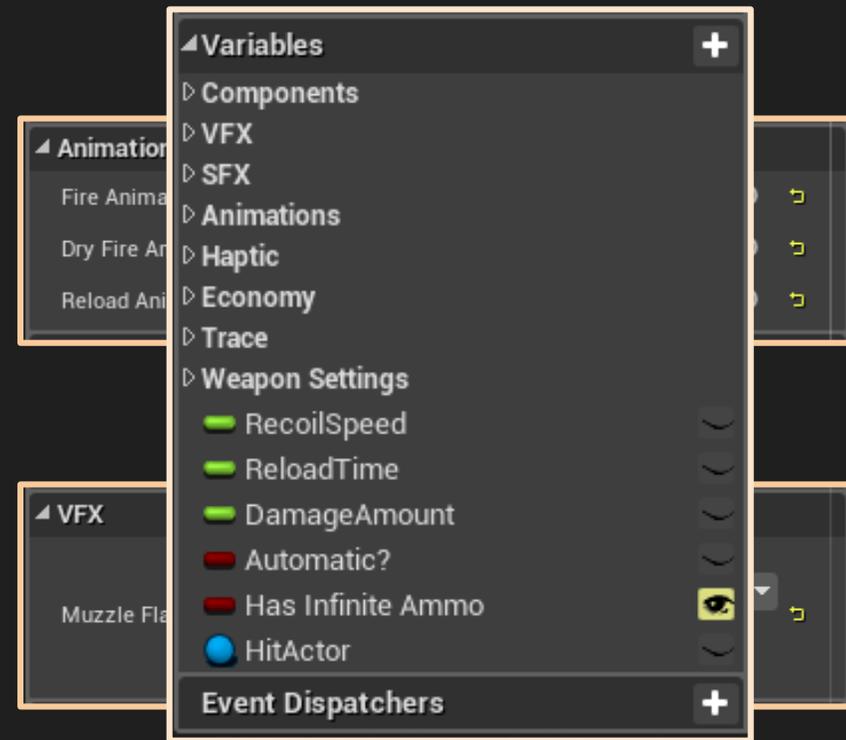
ENUMS-INTS-FLOATS-BOOLS



○ PFX SFX



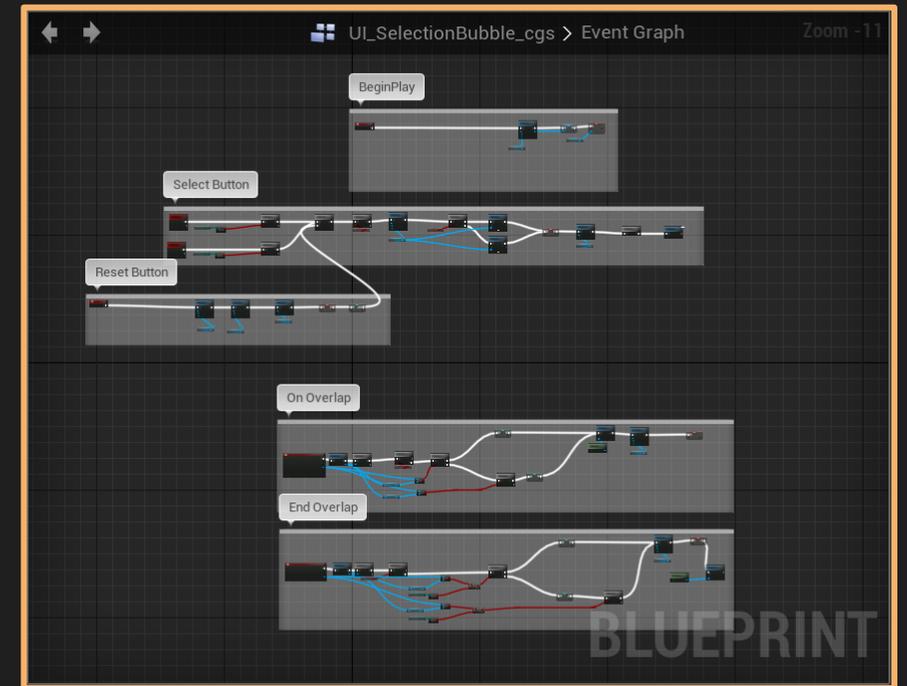
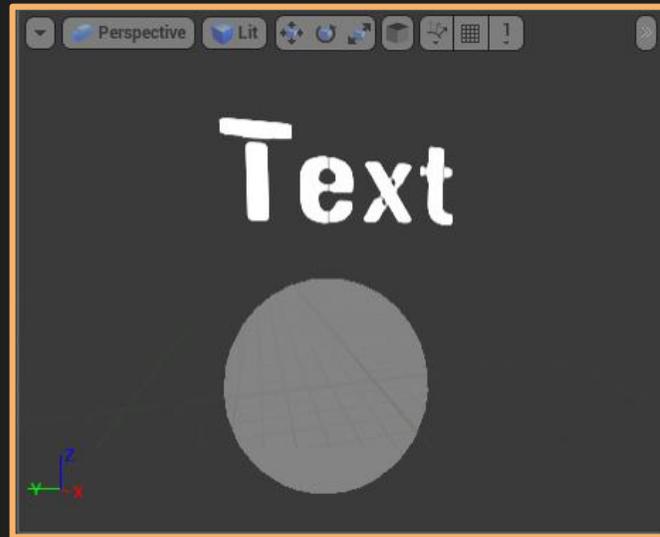
Vars



Simple Example: Menus

First, Make The Buttons

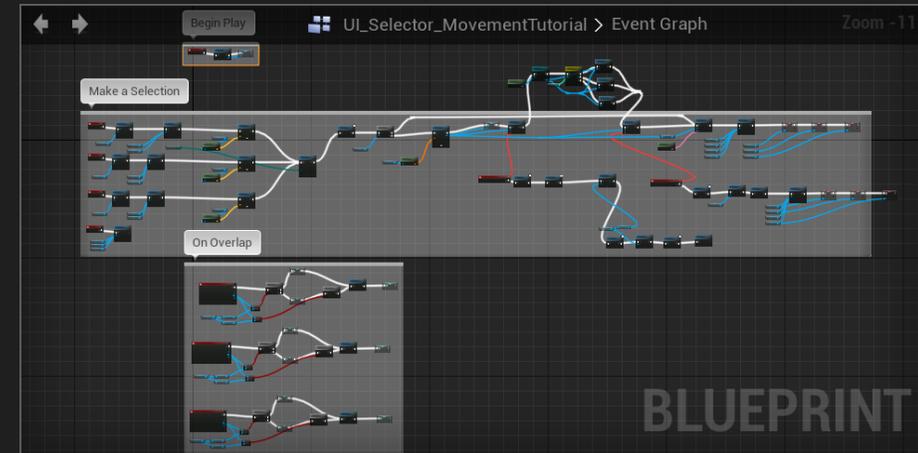
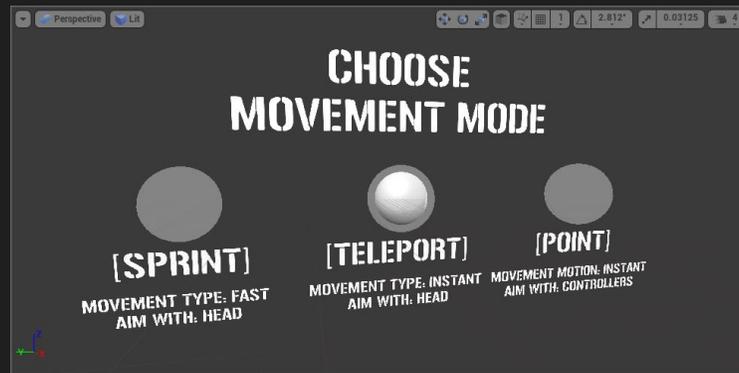
- Self Contained
- Fully Functional
- Sends an Activation Function to Parent



Create Custom Menus from Buttons

Use the Button BluePrints as Child Components

- Modular Construction
- Easier/Faster to Build
- Fairly Easy to Fix/Debug



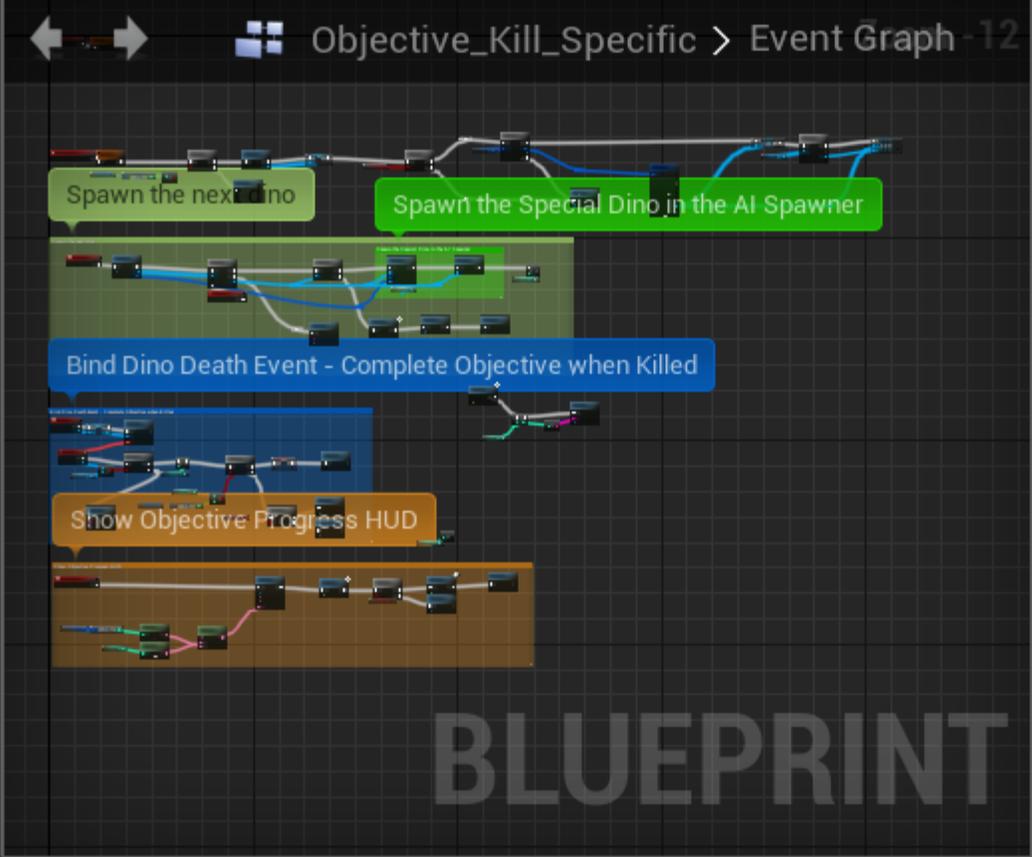
Components

+ Add Component

- Objective_Kill_Specific
- DefaultSceneRoot (In)

Compile Save Find in CB Search Class Settings Class Defaults

View Cons Pick Choc Rem Spav Even



My Blueprint

+ Add New

Graphs

- EventGraph

Functions (29 Over)

- Construction Scri
- Spawn Objective
- Pick Next Closes
- Choose New Spa
- Remove Closest

Macros

Variables

- Dinos to Kill
- Target Dino
- Use Random
- Show Kill Pro
- AI Special Spa
- Dino Spawn Ir
- Special Dinos
- test
- Objective Con

Details

Search

Default

- Dinos To Kill: 0 elements
- Use Random Spaw:
- Show Kill Progress:
- Dino Spawn Index: 0
- Special Dinos Kille: 0
- Test: 0
- Objective Complet:
- Mission Objective:
- Time Limit: -1
- Show HUD:
- Is Primary Objecti:

Actor Tick

- Start with Tick Ena:
- Tick Interval (secs): 0.0
- Allow Tick Before I:

Rendering

- Actor Hidden In Ga:
- Editor Billboard Sc: 1.0

Replication

- Only Relevant to O:
- Always Relevant:
- Replicate Moveme:
- Net Load on Client:
- Net Use Owner Rel:

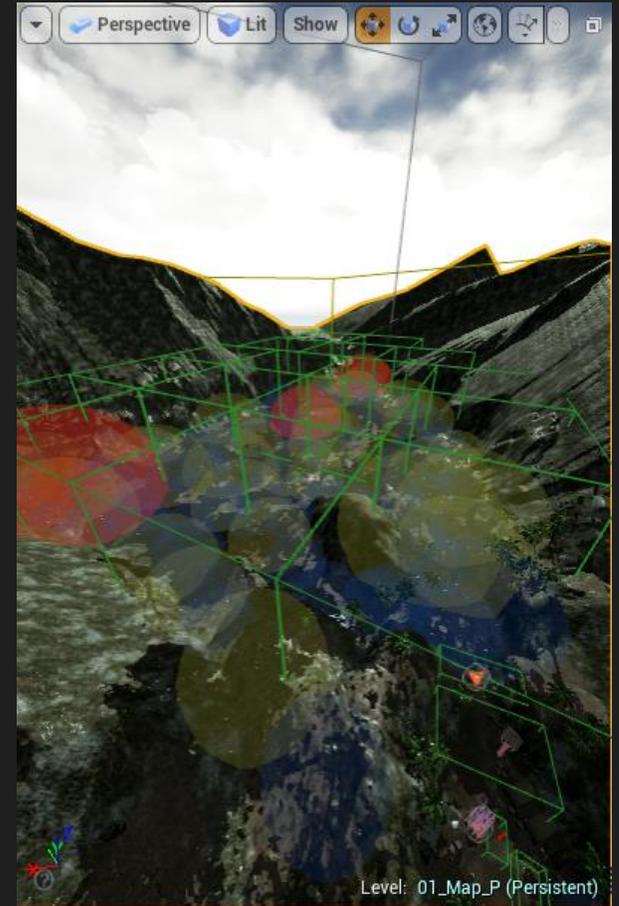
Compiler Results Find Results

Clear



Other Gameplay Challenges

- Actor Streaming
 - Only spawn actors when needed
 - AI
 - Ammo, Gun, Loot, Etc.
 - Unload actors when no longer needed
 - Keeps actor count/CPU load low



Lots of Little Pieces

In the right order can lead to great complexity



Ship It!

First Thing: Make a Development Build

- Do This Around Your First Playable
- SURPRISE: THERE WILL BE ERRORS
- You MUST fix these errors before you continue
- If you don't, you are only compounding your non-shippable issues.



Making a Build in UE4

It's Easier Than You Might Think

- You don't need a Programmer to do it
- No, seriously, you don't need a Programmer to do it
- UE4 made it easy for you
- If you're still not comfortable, sure, ask a Programmer...



We Got Some...Excuses

Our Favorite Examples of “The Tech Veto”

- “Programmers Have to Do It”
- “You Have to Use our Build Farm”
- “It Will Take Us At Least a Week to Set Up For You”
- “You Don’t Know Steam Pipe”
- “We Don’t Have Time To Help You”



Okay, We're Doin' it Live

We Refused to Settle for "No"

- We w
- Realiz
- Made
- We ju



If We Can Do It, You Can Do It

Unreal and BluePrints have Democratized Game Dev

What a difference a year makes

Any one of you could be standing here right now

“An idea without execution is just hallucination”

--Thomas Edison



Shipping and Beyond

Steve Bowler - President



If You're Doing It Yourself

Know That The Store Itself Takes Awhile

- You're going to need iconic flat art
 - Screen Shots
 - Tons of file sizes/formats
 - Even an .ico file
 - TRAILERS
 - Trailers in VR = epic p.i.t.a.
- Tons of details about your game
- Steam's Checklist makes this easy



So You're Starting a Steam Business

Good Luck. You're Going to Need It.

- You'll need some form of corporation
 - LLC -- Cheapest, difficult with multiple founders
 - C-Corp -- Expensive, but easy "slam dunk" for us
- This gets you a Federal Tax ID for your business
- You need a Corporate Bank Account to create a Steam Page
- Running a Store Page/Game is challenging
- Running your own company is even more challenging



This is What We Could Do In 5 Months

With Just Two People Using UE4



- How can your company use UE4 BluePrints?
- What could you accomplish with 6 people?
- What could you accomplish with 12 people? 40? 100+?
- We hope some of you leave here today and try this at home!



Thank You For Coming!

Special Thanks to Our Friends at Valve and Epic

Tim Sweeney
Dana Cowley
Chance Ivey
Nick Whiting
Nick Chester
Alan Noon
Ray Davis
David Stelzer
Wes Bunn

Chet Faliszek
Augusta Butlin
Tom Giardino
Matt Nickerson



Questions

