Showcasing the Unreal Engine 4 through 3D Game Development

Ian Wilder

Outline

- Background
- Development process
- Programmed features
- UE4 Editor
- Problems encountered
- Demonstration
- Possible future goals

What is Unreal Engine 4?

- Newest edition of Epic Games' Unreal Engine


STREET FIGHTER V


Disney
SQUARE ENIX
KINGDOM HEARTS
III


GEARS
OF WAR 4
Entertainment

What is Unreal Engine 4?

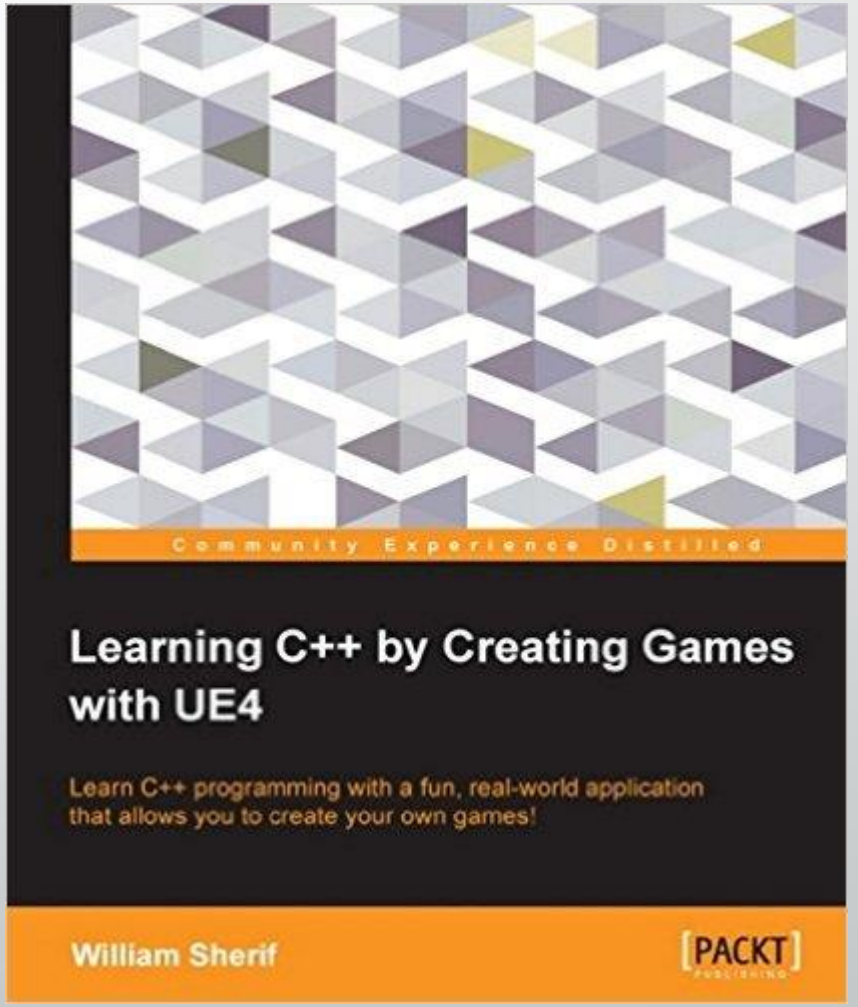
- Newest edition of Epic Games' Unreal Engine
- Released May 2012
- Became open source March 2014 for $19/month
- Complete suite of game development tools
- Free in March 2015
  - 5% royalty on revenue over $3000

What is included with Unreal Engine 4?

- Full access to engine library
  - Written in C++ and C#
- Free tutorials
- Free updates
- Documentation
- Marketplace to sell content
- Excellent online community

Development – Beginning

- Start with learning C++
- Reading documentation and API
- LOTS of YouTube tutorials
- *Learning C++ by Creating Games with UE4*
- Studied finished projects provide by Unreal
- Experimented with small projects



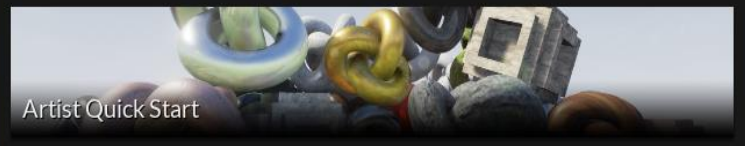
Community

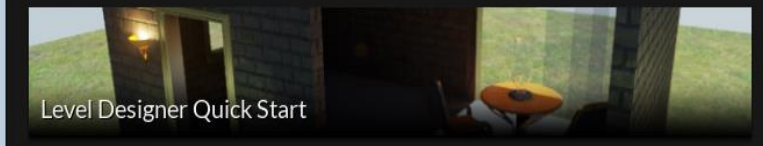
Learn

Marketplace

Library

Begin Your Journey

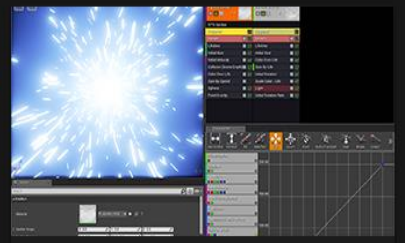




Broaden Your Horizons

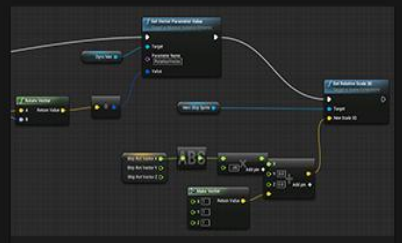
UE4 For Unity Developers

Translate your Unity knowledge into UE4 so you can get up to speed quickly.



Tools and Editors

Find the right tool for the job with this directory of editors and asset types available to you in Unreal Engine 4.



Blueprint Jump Starts

Blueprints are a visual scripting system that help you quickly add features to your game. These mini-tutorials will get you started.

Content Example Directory

Covering everything from Animation to VR, check out this guide to the examples contained in the Content Examples project below.

Community

Learn

Marketplace

Library

Showdown VR Demo

Showdown is a VR action cinematic that takes you through a guns-blazing scenario in bullet-time slow motion.

Contains

Infiltrator Demo

High-end demonstration of rendering and cinematic capabilities

Contains

Content Examples

This museum-style project has a collection of maps with stands that demonstrate specific features!

Contains

A Boy and His Kite

Explore Epic's GDC 2015 demonstration showcasing Unreal Engine 4's open world, cinematic and photoreal capabilities.

Contains

Open World Demo Collection

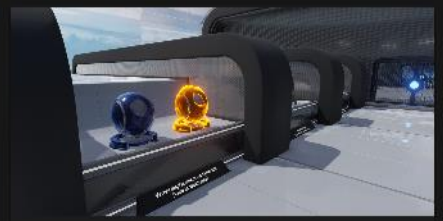
A collection of realistic assets from Epic's Open World demo shown at GDC 2015. Use them to create new levels or add them to existing ones.

Contains

Water Planes

This collection of watery surfaces provides examples of different water shaders and Blueprints to configure them.

Contains



Features Tour 2014

Epic's GDC2014 Demo! This project walks you through Unreal Engine features like the new material system and Blueprints.

Contains

Matinee

The Matinee example shows how to create highly stylized cinematic sequences using Unreal Engine 4's Matinee Editor.

Contains

Landscape Mountains

Explore high-end landscapes in UE4 with this

Realistic Rendering

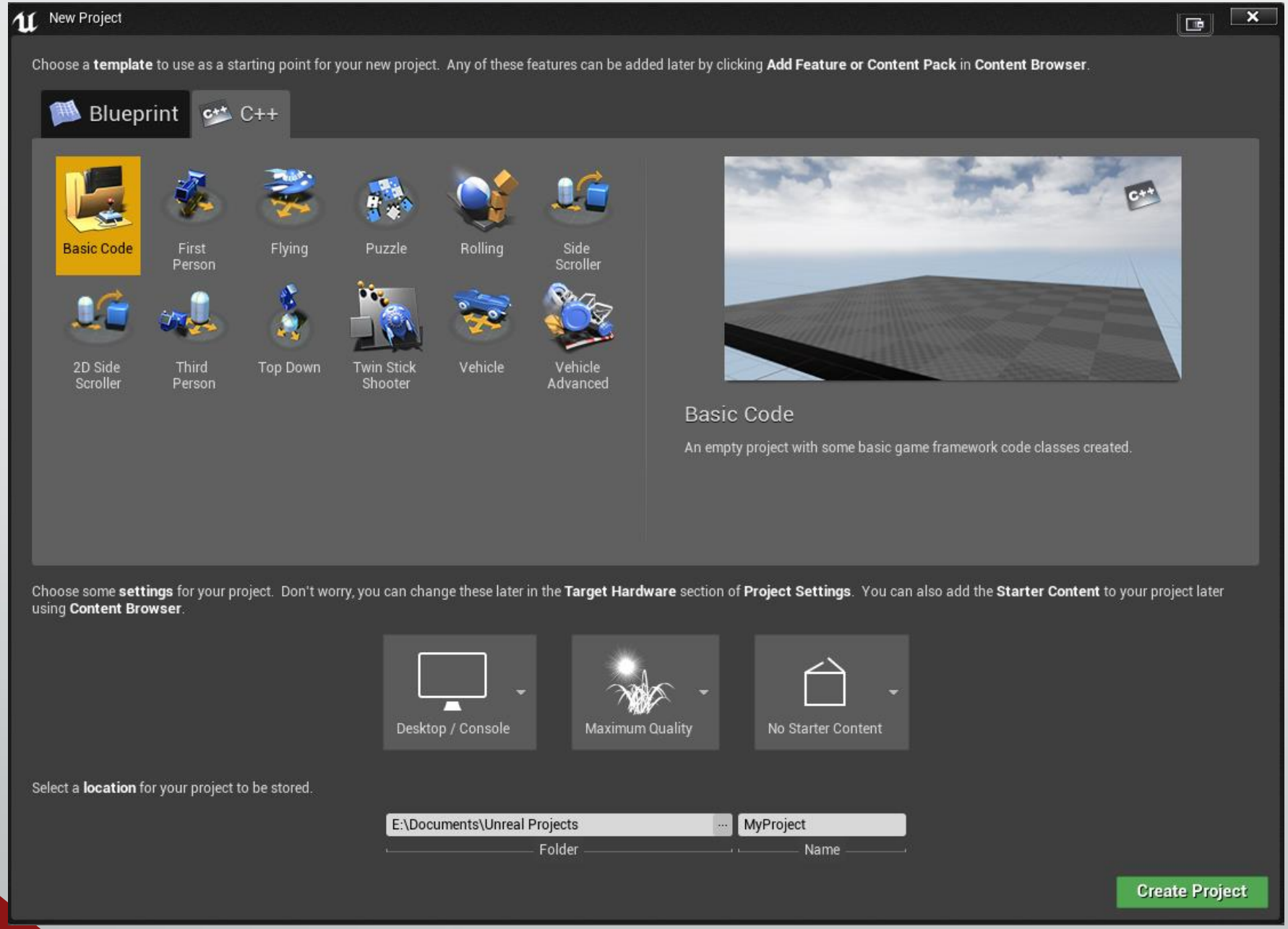
This example shows off the realistic rendering

Particle Effects

This project shows off a variety of particle systems

Elemental Demo

Experiment with the popular real-time


New Project
Choose a template to use as a starting point for your new project. Any of these features can be added later by clicking Add Feature or Content Pack in Content Browser.
Blueprint
C++
Basic Code
First Person
Flying
Puzzle
Rolling
Side Scroller
2D Side Scroller
Third Person
Top Down
Twin Stick Shooter
Vehicle
Vehicle Advanced
Basic Code
An empty project with some basic game framework code classes created.
Choose some settings for your project. Don't worry, you can change these later in the Target Hardware section of Project Settings. You can also add the Starter Content to your project later using Content Browser.
Desktop / Console
Maximum Quality
No Starter Content
Select a location for your project to be stored.
E:\Documents\Unreal Projects
MyProject
Folder
Name
Create Project

Development – Final Game

- Genre decision
  - Third Person Action
- Concept: “Coin” game
  - Short
  - Good amount of programmable gameplay
  - World design experimentation

Development – Creating Batteries

```
#include "Pickup.h"
#include "BatteryPickup.generated.h"

/**
 *
 */
UCLASS()
class BATTERYCOLLECTOR_API ABatteryPickup : public APickup
{
    GENERATED_BODY()

public:
    // Sets default values for this actor's properties
    ABatteryPickup();

    //Override WasCollected func (implementation)
    void WasCollected_Implementation() override;

    //Access battery power lvl in public
    float GetPower();

protected:
    //Set amount of power given
    UPROPERTY(EditAnywhere, BlueprintReadWrite, Category = "Power", Meta = (BlueprintProtected = "true"))
        float BatteryPower;
};
```

Development – Creating Batteries

```cpp
#include "BatteryCollector.h"
#include "BatteryPickup.h"


//Set default values
ABatteryPickup::ABatteryPickup()
{
    GetMesh()->SetSimulatePhysics(true);

    //base power level of battery
    BPower = 150.f;
}


void ABatteryPickup::WasCollected_Implementation()
{
    //Use base pickup behavior
    Super::WasCollected_Implementation();
    //Destroy battery
    Destroy();
}

//report power lvl
float ABatteryPickup::GetPower()
{
    return BPower;
}
```

Development - Spawning

```
FVector ASpawnVolume::GetRandomPointInVolume()
{
    FVector SpawnOrigin = WhereToSpawn->Bounds.Origin;
    FVector SpawnExtent = WhereToSpawn->Bounds.BoxExtent;

    return UKismetMathLibrary::RandomPointInBoundingBox(SpawnOrigin, SpawnExtent);
}
```

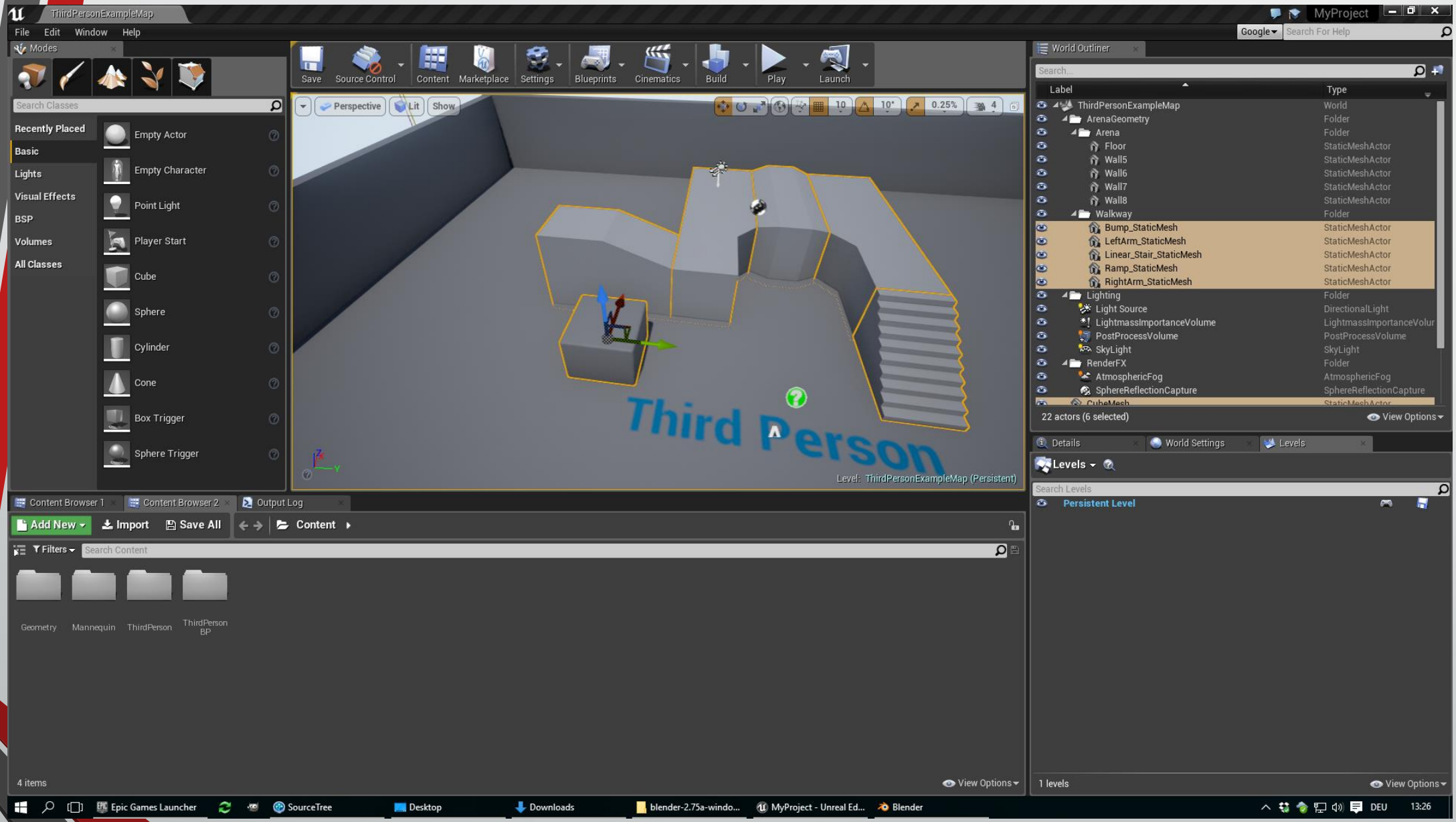
```
    //Get random location
    FVector SpawnLocation = GetRandomPointInVolume();

    //Get random rotation for item
    FRotator SpawnRotation;
    SpawnRotation.Yaw = FMath::FRand() * 360.0f;
    SpawnRotation.Pitch = FMath::FRand() * 360.0f;
    SpawnRotation.Roll = FMath::FRand() * 360.0f;

    //spawn pickup
    APickup* const SpawnedPickup = World->SpawnActor<APickup>(WhatToSpawn, SpawnLocation, SpawnRotation, SpawnParams);

    SpawnDelay = FMath::FRandRange(SpawnDelayRangeLow, SpawnDelayRangeHigh);
    GetWorldTimerManager().SetTimer(SpawnTimer, this, &ASpawnVolume::SpawnPickup, SpawnDelay, false);
}
```

Development – Game states

```
//If game is playing
case EBatteryPlayState::EPlaying:
{
    //spawn vol active
    for (ASpawnVolume* Volume : SpawnVolumeActors)
    {
        Volume->SetSpawningActive(true);
    }
}
break;
//If game is won
case EBatteryPlayState::EWon:
{
    //spawn vol inactive
    for (ASpawnVolume* Volume : SpawnVolumeActors)
    {
        Volume->SetSpawningActive(false);
    }
}
break;
//if game is lost
case EBatteryPlayState::EGameOver:
{
    //spawn vol inactive
    for (ASpawnVolume* Volume : SpawnVolumeActors)
    {
        Volume->SetSpawningActive(false);
    }
    //block input
    APlayerController* PlayerController = UGameplayStatics::GetPlayerController(this, 0);
    if (PlayerController)
    {
        PlayerController->SetCinematicMode(true, false, false, true, true);
    }
    //disable/ragdoll character
    ACharacter* MyCharacter = UGameplayStatics::GetPlayerCharacter(this, 0);
    if (MyCharacter)
    {
        MyCharacter->GetMesh()->SetSimulatePhysics(true);
        MyCharacter->GetMovementComponent()->MovementState.bCanJump = false;
    }
```

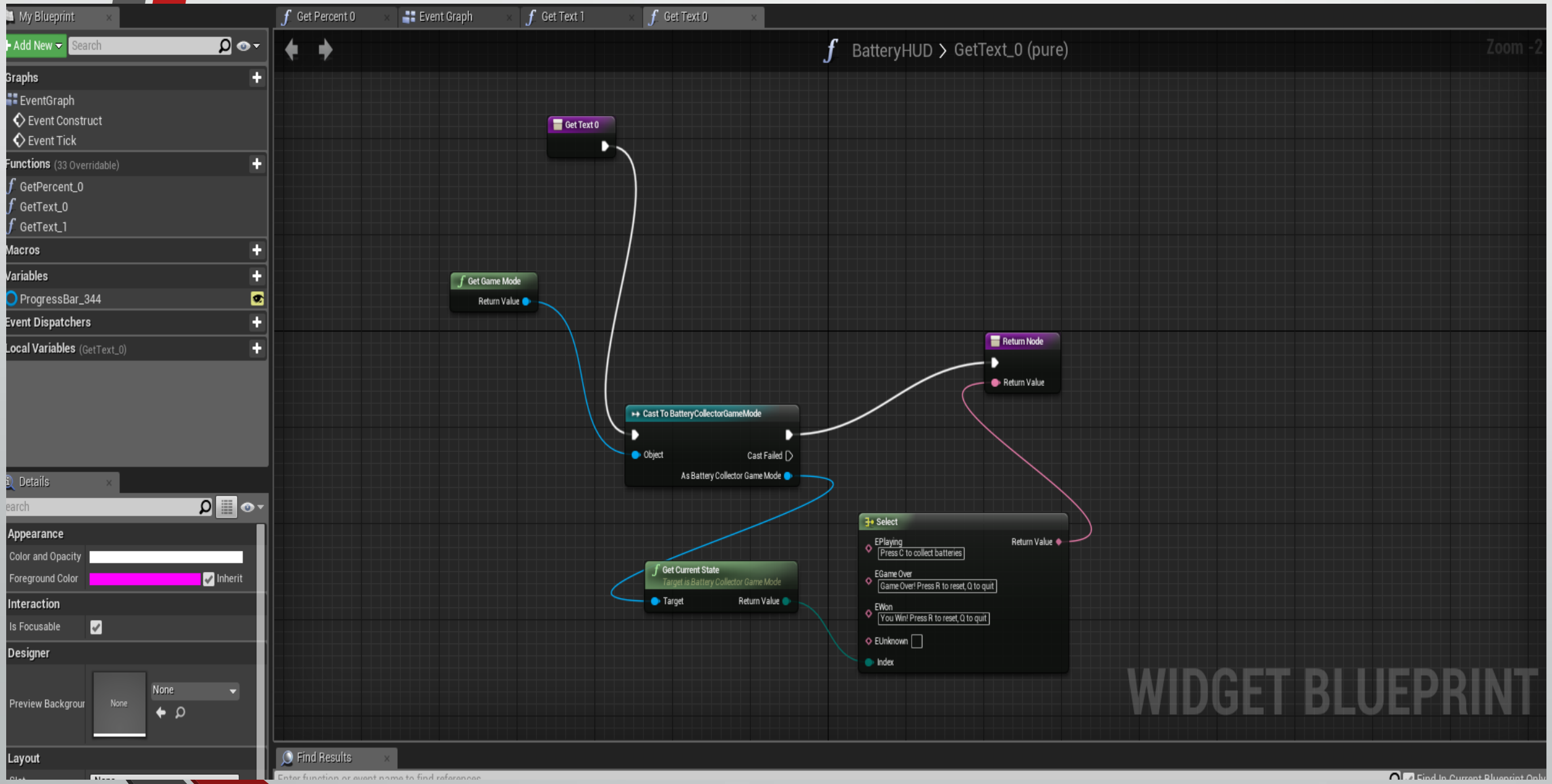
UE4 Editor

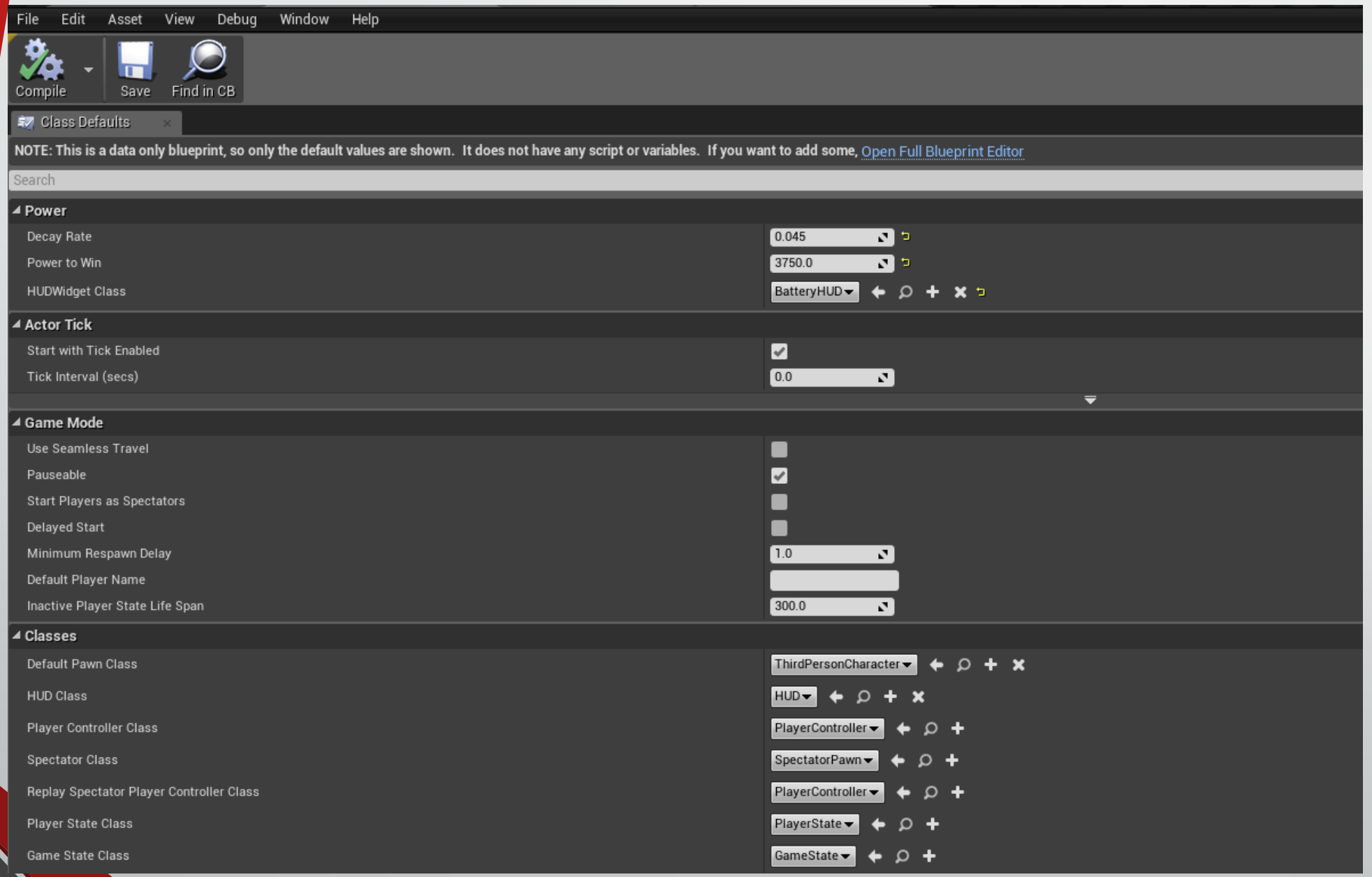
- Similar to Blender
- Creation of meshes, materials, and other effects
- Import assets from other projects
- Physics engine included
- Allows instant compiling of code
- Create and modify blueprints
- Creative side
- Test game in editor before packaging


ThirdPersonExampleMap
MyProject
File Edit Window Help
Modes
Save Source Control Content Marketplace Settings Blueprints Cinematics Build Play Launch
Search Classes
Recently Placed
Basic
Lights
Visual Effects
BSP
Volumes
All Classes
Empty Actor
Empty Character
Point Light
Player Start
Cube
Sphere
Cylinder
Cone
Box Trigger
Sphere Trigger
Perspective Lit Show
Third Person
Level: ThirdPersonExampleMap (Persistent)
World Outliner
Label Type
ThirdPersonExampleMap World
ArenaGeometry Folder
Arena Folder
Floor StaticMeshActor
Wall5 StaticMeshActor
Wall6 StaticMeshActor
Wall7 StaticMeshActor
Wall8 StaticMeshActor
Walkway Folder
Bump_StaticMesh StaticMeshActor
LeftArm_StaticMesh StaticMeshActor
Linear_Stair_StaticMesh StaticMeshActor
Ramp_StaticMesh StaticMeshActor
RightArm_StaticMesh StaticMeshActor
Lighting Folder
Light Source DirectionalLight
LightmassImportanceVolume
PostProcessVolume
SkyLight
RenderFX Folder
AtmosphericFog
SphereReflectionCapture
22 actors (6 selected)
View Options
Details World Settings Levels
Levels
Search Levels
Persistent Level
1 levels
View Options
Content Browser 1 Content Browser 2 Output Log
Add New Import Save All
Content
Filters
Search Content
Geometry Mannequin ThirdPerson ThirdPerson BP
4 items
View Options
Epic Games Launcher SourceTree Desktop Downloads blender-2.75a-windo... MyProject - Unreal Ed... Blender
DEU 13:26

Blueprints

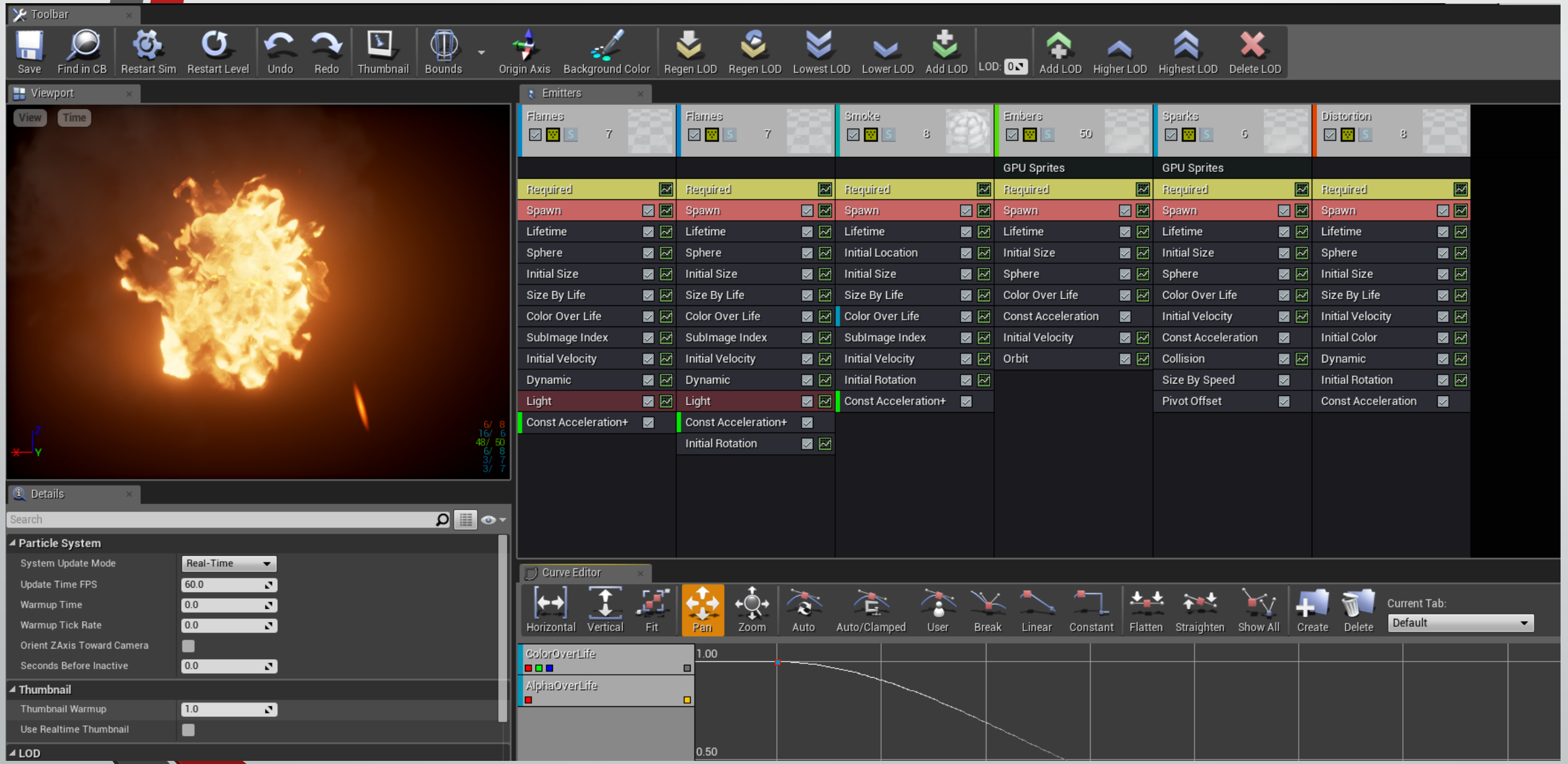
- Programming for those with little to no experience
- Visual programming
- Separate editor inside of UE4 editor
- Connect nodes and functions with wires
  - Properties of a character
  - HUD creation
  - User-to-world interaction
- Faster for small things


My Blueprint
Add New
Search
Graphs
EventGraph
Event Construct
Event Tick
Functions (33 Overridable)
GetPercent_0
GetText_0
GetText_1
Macros
Variables
ProgressBar_344
Event Dispatchers
Local Variables (GetText_0)
Details
Appearance
Color and Opacity
Foreground Color
Inherit
Interaction
Is Focusable
Designer
Preview Backgrour
None
Layout
Get Percent 0
Event Graph
Get Text 1
Get Text 0
BatteryHUD > GetText_0 (pure)
Zoom -2
Get Text 0
Get Game Mode
Return Value
Cast To BatteryCollectorGameMode
Object
Cast Failed
As Battery Collector Game Mode
Return Node
Return Value
Get Current State
Target is Battery Collector Game Mode
Target
Return Value
Select
EPlaying
Press C to collect batteries
Return Value
EGame Over
Game Over! Press R to reset, Q to quit
EWon
You Win! Press R to reset, Q to quit
EUnknown
Index
WIDGET BLUEPRINT
Find Results


File
Edit
Asset
View
Debug
Window
Help
Compile
Save
Find in CB
Class Defaults
NOTE: This is a data only blueprint, so only the default values are shown. It does not have any script or variables. If you want to add some, Open Full Blueprint Editor
Search
Power
Decay Rate
0.045
Power to Win
3750.0
HUDWidget Class
BatteryHUD
Actor Tick
Start with Tick Enabled
Tick Interval (secs)
0.0
Game Mode
Use Seamless Travel
Pauseable
Start Players as Spectators
Delayed Start
Minimum Respawn Delay
1.0
Default Player Name
Inactive Player State Life Span
300.0
Classes
Default Pawn Class
ThirdPersonCharacter
HUD Class
HUD
Player Controller Class
PlayerController
Spectator Class
SpectatorPawn
Replay Spectator Player Controller Class
PlayerController
Player State Class
PlayerState
Game State Class
GameState

Particles

- Visual effects
- Range from smoke and fire to complex effects
- Drag and drop into editor


Toolbar
Save
Find in CB
Restart Sim
Restart Level
Undo
Redo
Thumbnail
Bounds
Origin Axis
Background Color
Regen LOD
Regen LOD
Lowest LOD
Lower LOD
Add LOD
LOD: 0
Add LOD
Higher LOD
Highest LOD
Delete LOD
Viewport
View
Time
Emitters
Flames 7
Flames 7
Smoke 8
Embers 50
Sparks 6
Distortion 8
GPU Sprites
GPU Sprites
Required
Spawn
Lifetime
Sphere
Initial Size
Size By Life
Color Over Life
SubImage Index
Initial Velocity
Dynamic
Light
Const Acceleration+
Initial Location
Initial Rotation
Const Acceleration
Orbit
Collision
Size By Speed
Pivot Offset
Details
Search
Particle System
System Update Mode
Real-Time
Update Time FPS
60.0
Warmup Time
0.0
Warmup Tick Rate
0.0
Orient ZAxis Toward Camera
Seconds Before Inactive
0.0
Thumbnail
Thumbnail Warmup
1.0
Use Realtime Thumbnail
LOD
Curve Editor
Horizontal
Vertical
Fit
Pan
Zoom
Auto
Auto/Clamped
User
Break
Linear
Constant
Flatten
Straighten
Show All
Create
Delete
Current Tab:
Default
ColorOverLife
AlphaOverLife
1.00
0.50

Challenges Encountered

- Learning specifics of Unreal Engine 4
- Engine requires powerful computer
- Blueprints can be finicky
- Most of online community uses blueprints
  - Minor inconvenience
- First build was too easy
  - Added obstacles, changed spawn rate and decay

Demonstration

- Backstory
  - Robot has become too dangerous
  - Left on self-destructing lab in sea
  - Using parts of the lab to power himself up to escape
- Song credit: “Ignition” by TobyMac
  - Matches fast pace of game
  - Keep going, be strong

Demonstration

Future Goals

- Timer
- Battery count
- Cutscenes
- Different levels

Questions?