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
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
Begin Your Journey

Get Started with UE4

Artist Quick Start

Level Designer Quick Start

Programmer Quick Start

Broader 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 title, guide to the examples contained in the Content Example project below.
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.

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.

Select a location for your project to be stored.
Development – Final Game

• Genre decision
  • Third Person Action

• Concept: “Coin” game
  • Short
  • Good amount of programmable gameplay
  • World design experimentation
#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_Implmentation();

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

protected:

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

```cpp
FVector ASpawnVolume::GetRandomPointInVolume()
{
    FVector SpawnOrigin = WhereToSpawn->Bounds.Origin;
    FVector SpawnExtent = WhereToSpawn->Bounds.BoxExtent;
    return UKismetMathLibrary::RandomPointInBoundsBox(SpawnOrigin, SpawnExtent);
}
```

```cpp
// 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::RandRange(SpawnDelayRangeLow, SpawnDelayRangeHigh);
    GetWorldTimerManager().SetTimer(SpawnTimer, this, &ASpawnVolume::SpawnPickup, SpawnDelay, false);
```
Development – Game states

```cpp
// 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
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
Particles

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