

Game Programming in UE4

Game Framework & Sample Projects

Gerke Max Preussner

max.preussner@epicgames.com

Game Framework History

UE1 and UE2

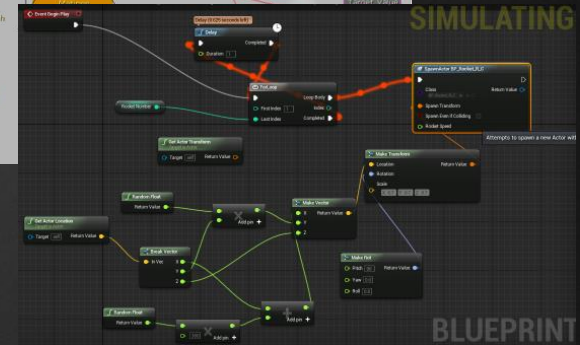
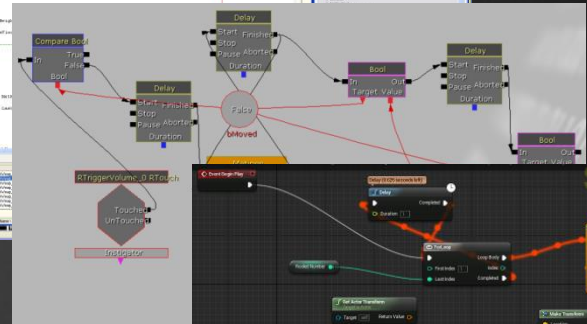
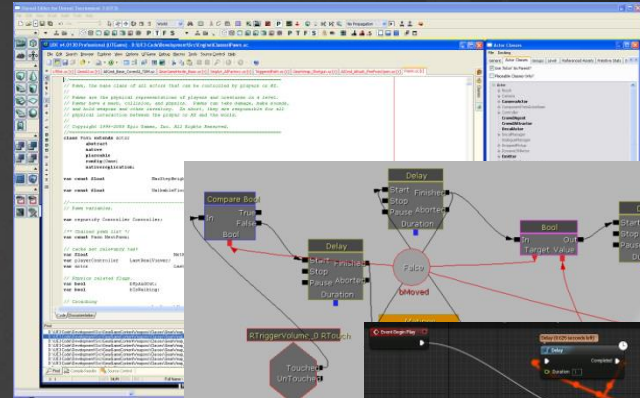
- Designed for First Person Shooters (FPS)
- UnrealScript game scripting language

UE3

- Kismet Visual Scripting added
- More modular game classes
- But still very FPS centric

UE4

- UnrealScript replaced with Blueprints
- Game genre agnostic
- Lots of sample projects!



UnrealScript vs. C++ vs. Blueprints

UnrealScript was:

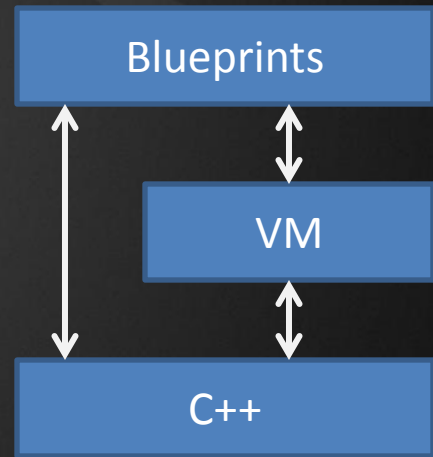
- An object-oriented scripting language
- Similar in syntax to C, C++, Java, but also somewhat different
- Compiled to virtual machine byte code
- Adding interesting features, such as States, Timers, Delegates

Blueprints are:

- A visual scripting system that is artist and designer friendly
- Using the same virtual machine as UnrealScript
- Almost as powerful as UnrealScript, and in some ways even better

C++ has:

- Always been part of UE game programming
- Tight bi-directional integrations with the virtual machine
- Been greatly improved in UE4 to replace UnrealScript for coders



So What Is The Game Framework?

Overview

Why Use It?

Set Of Foundation Classes

- Provide basic structure and functions of your game
- You derive from the classes that you need
- Fill in the details for your particular game

What's Missing?

- Game genre specific implementations
- No concept of health and death
- No built-in classes for weapons, inventory, etc.

So What Is The Game Framework?

Overview

Why Use It?

Reduced Learning Curve

- Hides the low-level details of game engines
- Benefit from 20+ years of game developing experience
- Many samples and tutorials for solving common tasks

High Productivity

- Focus on what really matters to you: making your game
- Solves many tricky problems in games (movement, input, etc.)
- Greatly reduces boilerplate code needed for all games

Scalability

- Smoothly grow your game & team from prototype to AAA

Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is an Actor?

- Entity in a game level
- Usually contains one or more Actor Components
- Supports network replication for multiplayer games

Things to know about Actors

- Don't have Location, Rotation (stored in root component)
- Created with `SpawnActor()` method
- Must be destroyed explicitly with `Destroy()` method
- Will not be garbage collected during the game

Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is an ActorComponent?

- Reusable functionality that can be added to an Actor
- Contain the most interesting functionality & events
- Also accessible in Blueprints!

Example Components:

- Scene Component – Adds transforms and attachments
- Primitive Component – Adds collision and rendering
- UAudioComponent, UArrowComponent, UInputComponent, ULightComponent, UMeshComponent, UParticleSystemComponent and many more!

Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

PrimitiveComponent Event Examples

- Hit – Called when bumping into a wall
- Begin/EndOverlap – Walk into or out of a trigger
- Begin/EndCursorOver
- Clicked/Released
- InputTouchBegin/End
- Begin/EndTouchOver

Fundamental Concepts

Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is a Pawn?

- An agent in the world
- Optionally possessed by a Controller
- Usually handles movement and input

Things to know about Pawns

- Good place to implement health
- No movement or input code by default



Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is a Controller?

- A brain that can possess a Pawn
- PlayerController: Represents a human player
- AIController: Computes AI behavior for Pawns

Things to know about Controllers

- Possess one Pawn at a time
- Can persist after possessed Pawn dies

Fundamental Concepts

Actors

Components

Pawn

Controller

Character

HUD

GameMode

PlayerController

- Interface for players to agents
- Handles touches, clicks, keyboard
- Showing/hiding mouse cursor
- Good place for non-agent code
- Menus, voice chat, etc.
- Many other useful options

DIRECT CONTROL ASSUMED



Fundamental Concepts

Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is a Character?

- Special Pawn that can walk
- Comes with useful Components

Things to know about Controllers

- Handles collision
- Client-side movement prediction
- Much improvement from UE3



Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is a HUD?

- Head-Up Display
- Responsible for in-game UI

Things to know about Controllers

- Immediate-mode drawing API
- No tools for building HUDs
- UMG will provide all the things!

Fundamental Concepts



Actors

Components

Pawn

Controller

Character

HUD

GameMode

What is a GameMode?

- Implements the game rules
- Configures default classes for Pawn, Controller, HUD, etc.
- Can be accessed from anywhere (`GetGameMode()`)

Things to know about GameModes

- Only exists on the server and single player instances!
- GameState is used to replicate game state to clients
- Default game mode can be set in Project Settings
- Per-map overrides in World Settings

Other Important Concepts

Input

Collision

Replication

Axes & Actions

- Can bind to mouse, keyboard, touch, etc.
- Defined in Project Settings

Input Processing Order

1. PlayerController
2. Level Blueprint
3. Possessed Pawn



Other Important Concepts

Input

Collision

Replication

Various Collision Query Functions

- Line traces (ray casts)
- Geometry sweeps
- Overlap tests

Simple Collision

- Box, sphere, capsule, convex
- Authoring tools in Editor
- Used for movement, physics, etc.

Complex collision

- Actual graphics triangles
- Used for weapons, kinematics, etc.



Other Important Concepts

Input

Collision

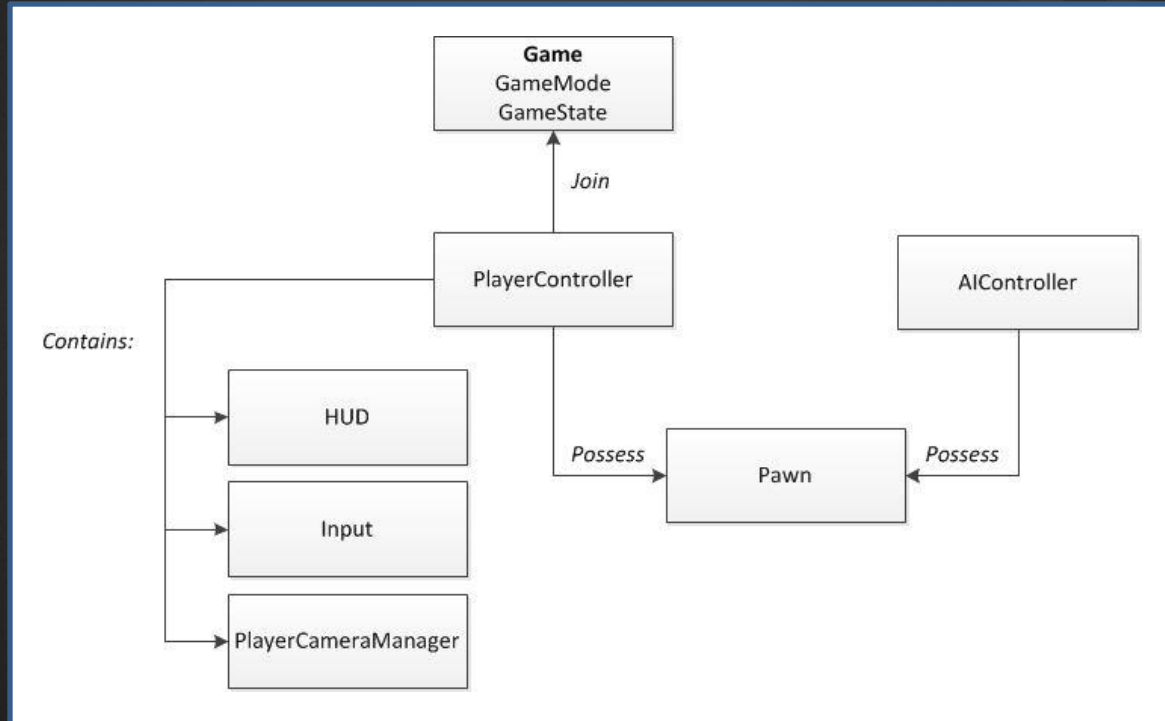
Replication

Transfer game state between Server and Clients

- Remote Procedure Calls (RPC)
- Data transfer (variables, structs, dynamic arrays)
- Editor supports multiplayer PIE testing



Game Framework Flowchart



Questions?

Documentation, Tutorials and Help at:

- AnswerHub: <http://answers.unrealengine.com>
- Engine Documentation: <http://docs.unrealengine.com>
- Official Forums: <http://forums.unrealengine.com>
- Community Wiki: <http://wiki.unrealengine.com>
- YouTube Videos: <http://www.youtube.com/user/UnrealDevelopmentKit>
- Community IRC: #unrealengine on FreeNode

Unreal Engine 4 Roadmap

- imgtfy.com/?q=Unreal+engine+Trello+

