Programming in UE4

A Quick Orientation for Coders

Gerke Max Preussner

max.preussner@epicgames.com

UNREAL ENGINE

Programming

Is Awesome

Sucks

But...

Because you can:

- Create something from nothing
- Bring dead matter to life
- Improve the human condition
- Impress your girl/boyfriend/cat
- Make a good living in the process

Programming

Is Awesome

Sucks

But...

Because:

- Programmers are crazy
- Programming languages & tools suck
- All code is bad and buggy
- There is never enough time to do it right
- You are always behind the curve

Peter Welch: http://stilldrinking.org/programming-sucks

Programming

Is Awesome

Sucks

But…

Don’t get discouraged!

- If it was easy, a monkey could do it!
- Don’t be afraid of programming languages
- Don’t get discouraged by complex code bases

There are ways to make your life easier

- Know your tools and keep learning
- Software Design and Architectural Patterns
- Coding Guidelines & Best Practices
- Transfer knowledge with your peers

Getting Started

Tools:

- Windows: Visual Studio, UnrealVS, Visual Assist X (recommended)
- MacOS: XCode

For everything else see:

https://docs.unrealengine.com/latest/INT/Programming/QuickStart/

Common Blockers

Compiling

Acronyms

Entry Point

Compiling is handled through UBT

- UBT – Unreal Build Tool
- Solution/Projects in Visual Studio and Xcode are a lie!

Common Blockers

Compiling

Acronyms

Entry Point

Acronym Soup (and Code Names, too)

- UBT – Unreal Build Tool
- UHT – Unreal Header Tool
- UAT – Unreal Automation Tool
- UFE – Unreal Frontend
- BP – Blueprint
- CDO – Class Default Object
- INI – Text Based Configuration File
- Cooking – Optimizing game content
- Lightmass, Persona, Cascade, Swarm and other tools
- etc. pp.

Common Blockers

Compiling

Acronyms

Entry Point

I Want To Understand the Engine - Where Is the Main Loop?

- LaunchEngineLoop.cpp
- It's really complicated (and everybody hates it)
- Please don't bother with this – start with our tutorials!

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

We Use Prefixes for All Types

- U – UObject derrived class, i.e. UTexture
- A – AActor derrived class, i.e. AGameMode
- F – All other classes and structs, i.e. FName, FVector
- T – Template, i.e. TArray, TMap, TQueue
- I – Interface class, i.e. ITransaction
- E – Enumeration type, i.e. ESelectionMode
- b – Boolean value, i.e. bEnabled

Everything in Unreal is Pascal Case (Upper Camel Case)

- Function names and function parameters, too
- Even local and loop variables!

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

UObjects Work Around Limitations in C++

- Run-time reflection of class properties and functions
- Serialization from/to disk and over the network
- Garbage collection
- Meta data
- Also: Blueprint integration

Decorate regular C++ Classes with Magic Macros

- UCLASS – for class types
- USTRUCT – for struct types
- UFUNCTION – for class and struct member functions
- UPROPERTY – for class and struct variables

Unrealisms

- Type Names
- UObjects
- Basic Types
- Strings
- Macros

```
// Example (not actual UE4 code – omitting some more advanced details)

USTRUCT()
struct FVector2D
{
        UPROPERTY()
        float X;

        UPROPERTY()
        float Y;

        UFUNCTION ()
        float GetLength() const;
};
```

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

Fundamental Types

- We don't use C++ integer types (char, short, int, long, etc.)
- Custom typedef's for ints & strings in GenericPlatform.h (int32, uint32, uint64, TCHAR, ANSICHAR etc.)
- Numeric type traits in NumericLimits.h

Common Structures

- FBox, FColor, FGuid, FVariant, FVector, TBigInt, TRange
- And many more in Core module

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

Containers

- TArray, TSparseArray – Dynamic arrays
- TLinkedList, TDoubleLinkedList
- TMap – Key-value hash table
- TQueue – Lock free FIFO
- TSet – Unordered set (without duplicates)
- And many more in Core module

Delegates

- Unicast and multicast delegates
- Also thread-safe variants

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

Smart Pointers

- TSharedPtr, TSharedRef – for regular C++ objects
- TWeakPtr – for regular C++ objects
- TWeakObjPtr – for UObjects
- TAutoPtr, TScopedPtr
- TUniquePtr
- Similar to boost:: & std:: implementations
- Also thread-safe variants

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

String Types

- FString – Regular string
- FText – Localized string, used heavily in Slate UI
- FName – String hash, used heavily in UObjects

String Literals

- TEXT() – Creates a regular(!) string, i.e. TEXT("Hello");
- LOCTEXT() – Creates a localized string, i.e. LOCTEXT("Namespace", "Name", "Hello");
- NSLOCTEXT() – LOCTEXT with scoped namespace, i.e. NSLOCTEXT("Name", "Hello");

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

FNames are case-insensitive!

- Can't rename 'MisSpelled' to 'Misspelled' in the Editor
- Can't name a property 'Blast' if one 'bLast' exists

Unrealisms

Type Names

UObjects

Basic Types

Strings

Macros

Logging
- UE_LOG, also GLog->Logf()

Assertions
- check(), checkSlow(), ensure()

Localization
- LOCTEXT_NAMESPACE, LOCTEXT, etc.

Slate (UI Framework)
- SLATE_BEGIN_ARGS, SLATE_ATTRIBUTE, etc.

Best Practices

Guidelines

Principles

Coding Guidelines

- Posted on http://docs.unrealengine.com
- There are some inconsistencies in the code base
- If in doubt, follow existing style in current code file

Naming Conventions

- Choose descriptive names that are as short as possible
- Also for local and loop variables!
- Avoid your own acronyms

Best Practices

Guidelines

Principles

General Principles

- KISS, YAGNI
- Composition vs. inheritance
- Avoid tight coupling of code and modules
- Many trivial instead of few complicated components

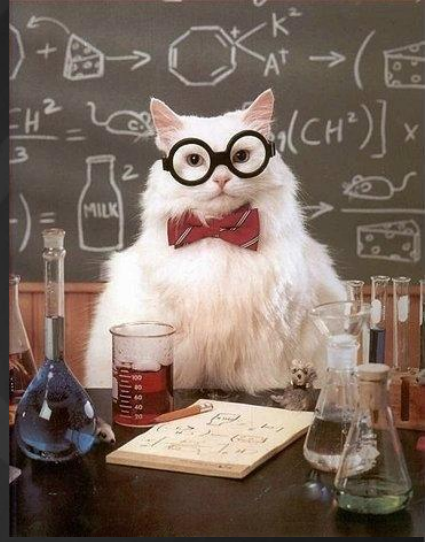
Design Patterns

- SOLID (especially S, O, L and I; DI is not elegant in C++)
- Hollywood Principle (especially for Slate & game code)
- GOF, EIP

Methodologies

- DDD, TDD (we support unit tests), AOP

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

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