Introduction to Unreal Engine 4 Game Development Using Blueprint

HackYourFuture, Amsterdam, January 13, 2019
by Mohammad S. Babaei

What is Unreal Engine 4?

- It is the fourth iteration of the Unreal technology which has been in development since 1998 for the first-person shooter game Unreal and later Unreal Tournament by Epic Games
- It is one of the most successful and widely used game engines today which has been used by professionals and beginners in game development industry in order to craft different kind of computer games ranging from 2D to 3D games, single player to massive multiplayer games, and even AR and VR games and simulations

ELI5: What is a Game Engine?

- When building a videogame, there are plenty of problems that need to be solved, tackled, or taken care of before artists, designers, and even programmers in the development team can get to the fun stuff
- Lots of these problems and issues are generic/common issues faced by video game developers during the development cycle
- Why not create a set of tools (a.k.a. game engine) to take care of all those repetitive often boring stuff, letting the team members jump directly into the fun bits of creating a game?
- Game engines achieve that by providing specialized tools tailored for every team's member's needs or requirements
- Sometimes these tools are visual with a user interface on top to interact with; e.g. for artists and designers in the team; and sometimes it is in the form of code - known as API or Frameworks - for other programmers in the team

Tools and Editors available through UE4Editor

- Level Editor
- Material Editor
- Blueprint Editor
- Behavior Tree Editor
- Persona Editor
- Cascade Editor
- UMG UI Editor
- Matinee Editor
- Sound Cue Editor
- Paper2D Sprite Editor
- Paper2D Flipbook Editor
- Physics Asset Tool Editor
- Static Mesh Editor
- Media Player Editor

What makes Unreal Engine 4 so great?

- With its code written in C++, Unreal Engine features a high degree of portability and is a tool used by many game developers today
- Many AAA games and thousand of indie games for wide range of platforms have been developed in Unreal Engine which proves it's a battle-tested tool for game development
- The source code for Unreal Engine is available free of charge which allows game developers to add new features and extend the engine for their in-house requirements
- Although originally it was intended for creating first-person shooter games, it has been successfully used in a variety of other genres, including stealth, fighting games, MMORPGs, and other RPGs
- Using and creating games in Unreal Engine 4 is free of initial costs. By accepting the EULA, you agree to pay Epic a royalty equal to 5% if your revenue from your game pass beyond $3,000.00 per calendar quarter; the same applies for games published on Oculus Store except when it goes beyond $5,000,000.00 per quarter (for more info consult the EULA please)

Why should I choose Unreal Engine for my first or next Game?

- Unreal Dev Grants Program: Epic has created a $5,000,000 development fund to provide financial grants to innovative projects built in and around Unreal Engine 4. Awards range from $5,000 to $50,000, and there are no strings attached: you continue to own your IP, are free to publish however you wish, and can use the grant funds without any restrictions or obligations to Epic
- If you need to to port and support a wide range of platforms and hardware, Unreal Engine makes this much much easier for you
- Its business model reduces the initial development costs for both small and large teams significantly, although some might not like the 5% lifetime royalty
- For people who need official support or do not like the 5% royalty, a Custom License is also available
- Since a lot of people are using it on daily basis, and the community is so supportive, finding support or answers to some of your questions might be easier than some other free tools

Unreal Engine 4 Supported Platforms

Supported development platforms are:

- Windows
- MacOS
- Linux (No binary version is available, supported through building from source)
- FreeBSD (unofficial port)

Games created in Unreal Engine 4 could target the following platforms:

Microsoft Windows, PlayStation 4, Xbox One, Mac OS X, iOS, Android, AR, VR, Linux, SteamOS, and HTML5

Please note that Xbox One, PlayStation 4 and NIntendo Switch are only available to dev teams who obtain a valid DevKit from their respective companies

Is there any limitations (license-wise) for using Unreal Engine 4?

Yes, Unreal Engine's usage terms and conditions are governed by a EULA which puts some restrictions on its usage. For example, the following applications are not acceptable or requires a Custom License from Epic Games:

- Releasing a game under a copyleft license (e.g. GPL, LGPL, CC BY-SA). Though, permissive licenses such as BSD License, MIT License, Microsoft Public License, or Apache License are acceptable
- Creating certain types of games such as gambling-related, serious games, or military usage

Please consult the EULA for further information!

Do I have to code in order to create a game?

tl;dr Yes and No!

ELI5: What is Blueprint Visual Scripting System?

- A visual scripting system designed for people that do not come from a computer science or programming background such as artists or level designers
- Instead of writing lines of code by a traditional programmer, Blueprint provides a way of scripting those some lines of code in a visual manner by connecting a series of nodes which have some functionality attached to them in order to create the same functionality that was traditionally only available to programmers
- In many ways Blueprint works identically to writing code, the only difference is Blueprint allows scripting in a more visual way by connecting nodes instead of writing lines of code

Clarification: Blueprint vs Blueprints

The term **Blueprint** (sometimes abbreviated as **BP**) refers to the Visual Scripting System, while the term **Blueprints** refers to the objects that are created as a result of using the Blueprint Visual Scripting System

A successor to Kismet and UnrealScript?

- UnrealScript and Kismet were provided as scripting choices by Unreal Engine 3 (UE3) and Unreal Development Kit (UDK)
- Kismet is a Turkish word meaning destiny or fate; and is originated from Arabic
- Kismet 2 or publicly known as Blueprint, is a successor and evolution to Kismet
- UnrealScript is a defunct Java-like compiled to bytecode scripting language in UE4
- But, does that mean Blueprint is also a replacement for UnrealScript? Yes and no. Gameplay programming and everything that UnrealScript was used for in the past can still be handled through code using C++. At the same time, while Blueprints are not meant as a replacement for UnrealScript, they do serve many of the same purposes that UnrealScript handled.
- Kismet scripts were bound to levels and not reusable while Blueprints could be either Level Blueprint (per level) or Class Blueprint (reusable)
- There are also other types of Blueprints

Various Types of Blueprints

- Blueprint Class
- Data-Only Blueprint
- Level Blueprint
- Blueprint Interface
- Blueprint Macro Library
- Blueprint Utility (a.k.a. Blutility)

The Blueprint Anatomy

- Components Window
- Construction Script
- Event Graph
- Functions
- Variables

Who is Blueprint ideal for?

- Ideal for beginners who want to learn UE4 but have no programming experience yet
- Ideal for artists and game designers who have no experience in programming and want to create quick prototypes or simple games.
- Ideal for programmers working in a team that want to make part of the game logic available to artists or game designers in order to offload some of their work

Blueprint: Pros and Cons

Pros:

- Highly productive specially for prototyping, creation, and iteration
- Shallow learning curve

Cons:

- 20x slower than C++, although Epic Games introduced Blueprint Nativization in the later versions of UE4 in order to circumvent performance issues by compiling Blueprints to C++; should be enabled manually, and still slower than hand-written C++
- Does not have access to all engine features and API
- Not as extensible as C++, specially for large projects
- Although UE4 has good source control support for Blueprint (Git, Subversion, Perforce), it is not as good as C++ due to nature of Blueprint assets (text vs binary)

C++: Pros and Cons

Pros:

- 20x faster than Blueprint
- Unrestricted access to all features and API of the engine

Cons:

- Steep learning curve
- Less productive than Blueprint
- Limited subset of C++ which sometimes looks and feels ugly

The choice of tool: Blueprints vs C++?

C++ Advantages

- **Faster runtime performance**: C++ logic is significantly quicker than Blueprint logic
- **Explicit Design**: by exposing variables or functions from C++ you have more control over exposing members, so you can a formal "API" for your class and avoids creating overly large and hard to follow Blueprints
- **Broader Access**: in addition to access to more engine functionality, functions and variables exposed in C++ can be accessed from all other systems
- **More Data Control**: access to more specific functionality for loading and saving data
- **Network Replication**: replication support in Blueprints is straightforward, but tight control over replication bandwidth or timing is only possible through C++.
- **Better For Math**: C++ is more suitable for math-heavy operations
- **Easier to Diff/Merge**: since C++ code and data is stored as text it's easier to manage under source control software

The choice of tool: Blueprints vs C++?

Blueprint Advantages

- **Faster Creation**: prototyping brand new systems is often faster in Blueprint
- **Faster Iteration**: much quicker to modify Blueprint logic and preview inside the editor than to recompile the game, although hot reload can help
- **Better For Flow**: visualizing "game flow" in C++ is complicated
- **Flexible Editing**: designers and artists without specific technical training can create and edit Blueprints, which makes Blueprints ideal for assets that need to be modified by more than just engineers.
- **Easier Data Usage**: storing data inside Blueprint classes is much simpler and safer than inside C++ classes; Blueprints are suitable for classes that closely mix Data and Logic

A perfect mix: C++ and Blueprints

Suitable for Gameplay Logic

- C++ Classes
- Blueprint Classes
- Custom Systems (e.g UE4 Materials Editor, Sequencer Tracks, and AI Behavior Trees)

Suitable for Data

- C++ Classes
- Blueprint Classes
- For Data you have more options than C++ and Blueprints: Config Files, Data Assets, Tables, Placed Instances, Custom Systems as with Gameplay Logic (e.g. Save Games)

In the real world, a mid to large-scale project uses a balanced dose of each. In fact C++ and Blueprints are tightly coupled in Unreal Engine 4.

What about the aesthetics, art, game design, and storytelling?

- Well, creating an awesome game would require specialized expertise in many fields (similar to how a movie is created) and usually cannot be done by one person alone (although there are some exceptions)
- As a result a rich games asset marketplace exists for UE4 filled with 2D assets, 3D models, textures, materials, 3D characters, weapons, environment assets, animations, particles and visual effects, and music and sfx assets
- There are also plenty of general purpose reusable Blueprints classes or C++ plugins available through UE4 Marketplace which can be used from both Blueprint and C++; e.g. Finite State Machines, Object Pools, FPS Kits, …
- In later slides, I'll also introduce other resources in order to get yourself familiarized with various game development skills other than programming; in fact it is a norm among indie and small developers for one person to accept and fill various development rules in the team

Nah! Games are boring! What else can I do in Unreal Engine 4?

- **Simulation or Real-Time Design**: in fact NASA, McLaren, and Microsoft AirSim (open source, hosted on GitHub) utilize Unreal Engine 4 this way
- **AR/VR**
- **Creating** generic art, animation, vfx, and music assets, or pre-built reusable Blueprints, or C++ plugins and selling them to other UE4 developers on UE4 Marketplace which proves to be a profitable business for some developers; considering the 88% (developer) / 12% (store) revenue share and more than 6.3 million users or 8 million downloads from Marketplace since its launch in 2014 up to July 2018

How to obtain a copy of Unreal Engine 4?

There are two ways to obtain a copy of Unreal Engine 4 for free:

- A binary version through Epic Games Launcher documented here https://docs.unrealengine.com/en-US/GettingStarted/Installation mostly suitable for Blueprint only projects

A Farsi/Persian translation is also available at https://fa.babaei.net/blog/unreal-engine-installation/

- Building Unreal Engine 4 from GitHub source which is the recommended method for mid to large C++ projects; please note that it's a private GitHub repository and requires an initial registration and activation through Epic Games Launcher

Downloading source code documentation: https://docs.unrealengine.com/en-us/GettingStarted/DownloadingUnrealEngine

Building from source documentation for Windows, MacOS, and Linux: https://docs.unrealengine.com/en-us/GettingStarted/DownloadingUnrealEngine

Let's create a 3D third-person endless runner game in 2 hours!

We are going to implement a prototype game similar to Jungle Run or Subway Surfers in pure Blueprints in just two hours!

Where to go from here?

- Free Unreal Engine 4 learning resources
- Paid Unreal Engine 4 learning resources
- Hardcore Unreal Engine 4 / C++ game development learning resources
- Resources to learn more about game design and development in general
- Hardcore game development learning resources (technical and highly advanced)
- Getting involved with active unreal engine 4 game development communities
- Even more game development communities

Free Unreal Engine 4 learning resources

- Official Unreal Engine 4 Documentation - https://docs.unrealengine.com/
- Official Unreal Engine 4 Youtube Channel - https://www.youtube.com/user/UnrealDevelopmentKit/
- Unreal Engine 4 Wiki - https://wiki.unrealengine.com/

Paid Unreal Engine 4 learning resources

- Unreal Engine 4 Books, Videos, Courses from Packt Publishing https://www.packtpub.com/
- Unreal Engine 4 Courses from Udemy

Hardcore Unreal Engine 4 / C++ game development learning resources

- **Shooter Game** from Epic Games Launcher - https://docs.unrealengine.com/en-us/Resources/SampleGames/ShooterGame
- Tom Looman's **Epic Survival Game** Series - https://github.com/tomlooman/EpicSurvivalGameSeries
- **Unreal Engine 4 Mastery: Create Multiplayer Games with C++** by Tom Looman - https://www.udemy.com/unrealengine-cpp/
- **Unreal Engine C++ Developer: Learn C++ and Make Video Games** by Ben Tristem and Sam Pattuzzi - https://www.udemy.com/unrealcourse/

Resources to learn more about game design and development in general

- **The Art of Game Design: A Book of Lenses 2nd Edition** by Jesse Schell
- **Level Up! The Guide to Great Video Game Design** by Scott Rogers
- **How Games Move US: Emotion by Design** by Katherine Isbister - https://mitpress.mit.edu/books/how-games-move-us
- **Amnesia Fortnight** (2012, 2014, 2017) by Double Fine in conjunction with Humble Bundle - https://www.doublefine.com/fortnight
- **Blender 3D: Noob to Pro** a featured book on Wikibooks and shared effort by numerous artists, authors, and editors in order to get familiarized with 3D content creation using free and open source tool Blender 3D
- **Extra Credits** - https://www.youtube.com/user/ExtraCreditz/

Resources to learn more about hardcore game development (technical and highly advanced)

- **Game Engine Architecture, Third Edition** by Jason Gregory - https://www.gameenginebook.com/
- **The Nature of Code: Simulating Natural Systems with Processing** free and open source book by Daniel Shiffman - https://natureofcode.com/
- **Handmade Hero** series by Casey Muratori - https://handmadehero.org/
- **ThinMatrix** - https://www.youtube.com/user/ThinMatrix
- **Thebennybox** - https://www.youtube.com/user/thebennybox
- **edX Computer Graphics Course** from The University of California, San Diego - https://www.edx.org/course/computer-graphics
- **Math for Game Developers** by Jorge Rodriguez - https://www.youtube.com/playlist?list=PLW3Zl3wyJwWOpdhYedlD-yCB7WQoHf-My

Getting involved with active unreal engine 4 game development communities

- **Official Unreal Engine Forums** - https://forums.unrealengine.com/
- **Official UE4 AnswerHub** - https://answers.unrealengine.com/
- **r/unrealengine** - https://www.reddit.com/r/unrealengine/
- **Unreal Slackers on Discord** - http://unrealslackers.org/
- **Unreal Engine 4 IRC** on freenode - http://webchat.freenode.net/?channels=%23unrealengine

Even more game development communities

- **r/gamedev** - https://www.reddit.com/r/gamedev/
- **gamedev.net** - https://www.gamedev.net/forums
- **TIGForums** - https://forums.tigsource.com/
- **Game Dev League** - https://discord.gg/0TYNJfCU4De7YIk8
- **Gamasutra** - http://www.gamasutra.com/
- **Handmade Hero Networks** - https://hero.handmade.network/forums

Any success stories for small devs?

Indie Game: The Movie (2012), it's a must see! - http://indiegamethemovie.com/

WAY TO THE WOODS (Unreal Engine 4 game) by a 16 years old Australian artist - https://redd.it/8274bo

Omno (Unreal Engine 4 and Blueprint only game), according to Kickstarter 3,471 backers pledged €97,769 to help bring this project to life. - https://redd.it/9vkxob

License

Feel free to share and adapt this presentation under (CC BY-SA 3.0) Creative Commons Attribution-ShareAlike 3.0 Unported License - https://creativecommons.org/licenses/by-sa/3.0/

Q & A

- Q?
- A.

OK! If I can't answer your questions in the Q & A time frame or your questions requires more explanation, I am also available on HackYourFuture's Slack workspace for further Q & A. Feel free to drop me a line in the #alumni channel!