

# Development a Software with Augmented Reality Using Unreal Engine 4

Vasyl Kushnir  
Ivan Franko Lviv National University  
Lviv, Ukraine  
vasyll195@gmail.com

Bohdan Koman  
Ivan Franko Lviv National University  
Lviv, Ukraine  
sonce\_28@ukr.net

Roman Shuvar  
Ivan Franko Lviv National University  
Lviv, Ukraine  
shuvar@electronics.lnu.edu.ua

**Abstract**—We are live in a time where technology take a big part in our life. To use technology we need to interact with them using interfaces. There are a lot of hardware manipulators like keyboard, joystick, microphone and programm manipulators like button, textfield, media player, etc. But one of the most interesting is augmented reality. It help us to develop not just a simple application or a game but an application that interacts with a whole real world and opens user a new beautiful mix of real and virtual world.

**Index Terms**—Unreal engine 4; augmented reality(AR); mixed reality; software

## I. INTRODUCTION

Augmented Reality is an interactive experience of a real-world environment whose elements are "augmented" by computer-generated perceptual information, sometimes across multiple sensory modalities, including visual, auditory, haptic, somatosensory, and olfactory. The overlaid sensory information can be constructive (i.e. additive to the natural environment) or destructive (i.e. masking of the natural environment) and is seamlessly interwoven with the physical world such that it is perceived as an immersive aspect of the real environment. In this way, augmented reality alters one's ongoing perception of a real world environment, whereas virtual reality completely replaces the user's real world environment with a simulated one. Augmented reality is related to two largely synonymous terms: mixed reality and computer-mediated reality[4].

Augmented, virtual and mixed reality technologies are evolving rapidly and becoming more powerful by the day. Even if you're starting with a simple project to learn AR, VR or mixed reality, Unreal Engine empowers you to build your team, your assets and your workflow with tools that can deliver on your creative vision and quality bar - now and in the future.

To create immersive experiences that are believable to the human mind, AR, VR and mixed reality require complex scenes rendered at very high framerates. There are no shortcuts. Because Unreal Engine 4 is designed for demanding applications such as AAA games, filmmaking and photoreal visualization, it meets these requirements and provides a solid foundation to build upon.

Both powerful and proven, the world's leading brands consistently choose Unreal Engine to bring their stories to life.

## II. AR SOFTWARE ALGORITHMS

A key measure of AR systems is how realistically they integrate augmentations with the real world. The software must derive real world coordinates, independent from the camera, from camera images[2]. That process is called image registration, and uses different methods of computer vision, mostly related to video tracking. Many computer vision methods of augmented reality are inherited from visual odometry.

Usually those methods consist of two parts. The first stage is to detect interest points, fiducial markers or optical flow in the camera images[2]. This step can use feature detection methods like corner detection, blob detection, edge detection or thresholding, and other image processing methods. The second stage restores a real world coordinate system from the data obtained in the first stage. Some methods assume objects with known geometry (or fiducial markers) are present in the scene. In some of those cases the scene 3D structure should be precalculated beforehand. If part of the scene is unknown simultaneous localization and mapping (SLAM) can map relative positions. If no information about scene geometry is available, structure from motion methods like bundle adjustment are used[3]. Mathematical methods used in the second stage include projective (epipolar) geometry,

geometric algebra, rotation representation with exponential map, kalman and particle filters, nonlinear optimization, robust statistics.[4]

To enable rapid development of augmented reality applications, some software development kits (SDKs) have emerged. A few SDKs such as CloudRidAR leverage cloud computing for performance improvement. AR SDKs are offered by Vuforia, ARToolKit, Catchoom CraftAR Mobinett AR, Wikitude, Blippar Layar Meta. and ARLab[3].

### III. AR DEVELOPMENT

The implementation of Augmented Reality in consumer products requires considering the design of the applications and the related constraints of the technology platform. Since AR systems rely heavily on the immersion of the user and the interaction between the user and the system, design can facilitate the adoption of virtuality[2]. For most Augmented Reality systems, a similar design guideline can be followed. The following lists some considerations for designing Augmented Reality applications:

#### A. Environmental/context design

Context Design focuses on the end-user's physical surrounding, spatial space, and accessibility that may play a role when using the AR system[2]. Designers should be aware of the possible physical scenarios the end-user may be in such as:

- Public, in which the users use their whole body to interact with the software
- Personal, in which the user uses a smartphone in a public space
- Intimate, in which the user is sitting with a desktop and is not really in movement
- Private, in which the user has on a wearable.

#### B. Interaction design

Interaction design in augmented reality technology centers on the user's engagement with the end product to improve the overall user experience and enjoyment. The purpose of Interaction Design is to avoid alienating or confusing the user by organizing the information presented[3]. Since user interaction relies on the user's input, designers must make system controls easier to understand and accessible. A common technique to improve usability for augmented reality applications is by discovering the frequently accessed areas in the device's touch display and design the application to match those areas of control. It is also important to structure the user journey maps and the flow of information presented which reduce the system's overall cognitive load and greatly improves the learning curve of the application[4].

#### C. Visual design

In general, visual design is the appearance of the developing application that engages the user. To improve the graphic interface elements and user interaction, developers may use visual cues to inform user what elements of UI are designed to interact with and how to interact with them. Since navigating in AR application may appear difficult and seem frustrating, visual cues design can make interactions seem more natural[2].

In some augmented reality applications that use a 2D device as an interactive surface, the 2D control environment does not translate well in 3D space making users hesitant to explore their surroundings. To solve this issue, designers should apply visual cues to assist and encourage users to explore their surroundings.

It is important to note the two main objects in AR when developing VR applications: 3D volumetric objects that are manipulatable and realistically interact with light and shadow; and animated media imagery such as images and videos which are mostly traditional 2D media rendered in a new context for augmented reality. When virtual objects are projected onto a real environment, it is challenging for augmented reality application designers to ensure a perfectly seamless integration relative to the real world environment, especially with 2D objects. As such, designers can add weight to objects, use depth maps, and choose different material properties that highlight the object's presence in the real world. Another visual design that can be applied is using different lighting techniques or casting shadows to improve overall depth judgment. For instance, a common lighting technique is simply placing a light source overhead at the 12 o'clock position, to create shadows upon virtual objects[3].

### IV. UNREAL ENGINE 4

In August 2005, Mark Rein the vice-president of Epic Games, revealed that Unreal Engine 4 had been in development since 2003. Until 2008, development was "basically" done by Tim Sweeney, CEO and founder of Epic Games. The engine targets the eighth generation of consoles, PCs and Tegra K1-based devices running Android announced in January 2014 at CES.[5]

In February 2012, Mark Rein said: "people are going to be shocked later this year when they see Unreal Engine 4". Unreal Engine 4 was unveiled to limited attendees at the 2012 Game Developers Conference, and video of the engine being demonstrated by technical artist Alan "Talisman" Willard was released to the public on June 7, 2012 via GameTrailers TV. This demo was created on a PC with triple GeForce GTX 580 (tri SLI) and can be run on a PC with a GeForce GTX 680. [5]

One of the major features planned for UE4 was real-time global illumination using voxel cone tracing, eliminating pre-

computed lighting. However, this feature has been replaced with a similar but less computationally-expensive algorithm prior to release for all platforms including the PC because of performance concerns. UE4 also includes new developer features to reduce iteration time, and allows updating of C++ code while the engine is running. The new "Blueprint" visual scripting system (a successor to UE3's "Kismet") allows for rapid development of game logic without using C++, and includes live debugging. The result is reduced iteration time, and less of a divide between technical artists, designers, and programmers. [1]

I could say: 'I'm going to convert this pillar into a blueprint [in the Engine] and add some sort of trap to it.' It means I can really go in and start enhancing my world with interaction that just would not have been possible without a technical artist, a designer and a programmer and now any one of those three can do all of it, provided they have the assets handy. The fact that I can just go in and say, 'If you're within X distance of this thing, start to glow and take my distance to it, normalize it zero to one and then just lerp [oscillate] between two different brightness values, so as I reach for something it gets hot'...that would have been something do-able but very difficult for anybody except a gameplay programmer. And he wouldn't have known how to set up the assets, but now any one of the three could do it.—*Alan Willard*

Development of the first generation Unreal Engine was carried out by the founder of Epic Games, Tim Sweeney. Inspired by John Carmack's pioneering programming work on *Doom* and then *Quake*, Sweeney started the engine in 1995 for the production of a game that would later become known as *Unreal*, a first-person shooter set in a medieval world with alien elements. After years in development, it debuted with the game's release in 1998, although licensees such as MicroProse and Legend Entertainment had possessed the technology much earlier, with the first licensing deal taking place in 1996.

Both software and hardware rendering were present in the foundational software, as well as collision detection, colored lighting, and a rudimentary version of texture filtering. The engine also provided a level editor, UnrealEd (formerly Unreal World Editor), that had support for real-time constructive solid geometry operations as early as 1996, allowing mappers to change the level layout "on the fly". Other features implemented during the engine's development included real-time direct illumination and light sourcing, which were respectively integrated in 1995 and 1997. In addition to having support for Microsoft Windows, Linux and Mac, *Unreal Tournament* also opened the platform to PlayStation 2 and, with the help of Secret Level, to Dreamcast. In 2000, Epic updated the engine with new improvements, including higher-polygon models and architecture, a skeletal animation system and large-scale terrain support.

The currently supported platforms are: Microsoft Windows, macOS, Linux, SteamOS, FreeBSD, HTML5, iOS, Android, Nintendo Switch, PlayStation 4, Xbox One, Magic

Leap One and virtual reality (SteamVR/HTC Vive, Oculus Rift, PlayStation VR, Google Daydream, OSVR and Samsung Gear VR).

Similar to Java (some of think to C++), UnrealScript is object-oriented without multiple inheritance (classes all inherit from a common Object class), and classes are defined in individual files named for the class they define. Unlike Java, UnrealScript does not have object wrappers for primitive types. Interfaces are only supported in Unreal Engine generation 3 and a few Unreal Engine 2 games. UnrealScript supports operator overloading, but not method overloading, except for optional parameters.

One of the key moments in Unreal Engine 4's development was, we had a series of debates about UnrealScript – the scripting language I'd built that we'd carried through three generations. And what we needed to do to make it competitive in the future. And we kept going through bigger and bigger feature lists of what we needed to do to upgrade it, and who could possibly do the work, and it was getting really, really unwieldy. And there was this massive meeting to try and sort it out, and try to cut things and decide what to keep, and plan and there was this point where I looked at that and said 'you know, everything you're proposing to add to UnrealScript is already in C++.

—*Sweeney, speaking to Gamasutra*. [5]

With Unreal Engine 4, Epic opened the Unreal Engine Marketplace in September 2014. The Marketplace is a digital storefront that allows content creators and developers to provide art assets, models, sounds, environments, code snippets, and other features that others could purchase, along with tutorials and other guides. Some content is provided free by Epic, including previously offered Unreal assets and tutorials. Prior to July 2018, Epic took a 30% share of the sales but due to the success of Unreal and *Fortnite Battle Royale*, Epic retroactively reduced its take to 12%. [5]

In 2014, Guinness World Records called the Unreal technology the most successful video game engine.

#### V. AR ON UNREAL ENGINE 4

To prepare unreal engine 4 for using to develop ar based software you must install ARCORE prerequisites to develop for IOS, android and windows.[1] Then you need to open the Unreal Project Browser and create a new Blueprint project with the following options:

- Select the Handheld AR template.
- Set the Target Hardware to be Mobile / Tablet.
- Set Graphics Level to Scalable 3D or 2D.
- Select No Starter Content

The result is a new UE4 project (ARProject), based on the Handheld AR blueprint template.

The HandheldAR Blueprint template pre-configures many of the required project settings for using augmented reality with Unreal Engine. However, when creating a project for the iOS platform, you will need to manually configure some settings [5].

To deploy the current Level to a device, you first need to make sure that you have the Level that you want to test open. For this tutorial, we will use the Handheld AR Blank Map level that was created by the Handheld AR template. Once the deployment starts, status dialog boxes are displayed towards the bottom of the **Unreal Editor** screen showing progress. App deployment can take several minutes[1]. The latest update to the Unreal Engine (4.19) is going all-in on augmented reality (AR). That's big news for the developers and tech pros who rely on Unreal to create games and applications, and who might be interested in exploring how they can port their work into an AR environment.[5]

Unreal's Unified Unreal Augmented Reality Framework (go ahead, say that five times fast) is positioned as a unified framework for iOS and Android AR apps. "The framework provides a single path of development, allowing developers to build AR apps for both platforms using a single code path". "The Unified Unreal AR Framework includes functions supporting Alignment, Light Estimation, Pinning, Session State, Trace Results, and Tracking."

There's also an AR-based project template, complete with an example project that details all aspects of Unreal's AR functionality.

By unveiling a unified AR development platform, Unreal is trying to position itself as an ideal solution for AR games and apps in an increasingly crowded market. Unreal has certain advantages in this fight, including a huge install base among game developers. The big question is whether AR developers will gravitate toward a unified platform, or choose to stick with specific tools for iOS and Android; as with so much software, ease-of-use will help determine the Unified Unreal Augmented Reality Framework's adoption rate.

#### ARCore

- Augmented Images
- Vertical Plane Detection
- Cloud Anchors

#### ARKit

- Face Tracking

- 2D Image Detection
- 3D Object Detection
- Persistent Experiences
- Shared Experiences

#### ACKNOWLEDGMENT

Unreal engine 4 helps easy and fast to develop your augmented reality based application, build it and publish it into the store. Unreal engine 4 provides developer with a clear documentation and advices about how to develop your AR software in the best way.

Augmented reality is a technology that overlays a computer-generated image on a user's view of the real world, providing a composite view.

The Unreal Engine AR framework provides a rich, unified framework for building augmented reality apps with the Unreal Engine for both iOS and Android handheld platforms. The unified framework provides a single path of development for both platforms, allowing developers to build augmented reality apps for both platforms using a single code path. The Handheld AR Blueprint template provides a complete example project demonstrating the augmented reality functionality available in the Unreal Engine.

The unified AR framework includes support for basic AR functionality like Alignment, Light Estimation, Pinning, Session State, Trace Results, and Tracking.

However, the augmented reality story for Android and iOS is constantly evolving. As of Unreal Engine 4.20, we now support the advanced functionality available in the ARCore 1.2 and ARKit 2.0 releases.

#### REFERENCES

- [1] Epic games, "Unreal engine 4 documentation", July 2011.
- [2] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, M. Ivkovic, "Augmented reality technologies, systems and applications", 2010
- [3] Wendy E. Mackay, "Augmented Reality: Linking real and virtual worlds A new paradigm for interacting with computers", Université de Paris-Sud.
- [4] S. Siltanen, "Theory and applications of marker-based augmented reality," 2012.
- [5] "Wikipedia. Unreal engine 4." - [resource]: [https://en.wikipedia.org/wiki/Unreal\\_Engine](https://en.wikipedia.org/wiki/Unreal_Engine)