Week 2: Activity – Prepare Platforms & Tools

Research

**UE4 and mobile development**

Unreal and its mobile development environment has made very big strides the past few years. After the release of Fortnite we saw the release of Unreal Engine 4.20 which brought with it over 100 mobile optimizations tools as well as the ability to optimize gameplay across platforms as stated by Tim Slager [reference] T.S. 2018. Exploring Unreal Engine 420’s Mobile Optimizations. August 27. Exploring Unreal Engine 420’s Mobile Optimizations. [Online]. [10 June 2019]. Available from: <https://www.unrealengine.com/en-US/tech-blog/exploring-unreal-engine-4-20-s-mobile-optimizations>.

The blog also includes a Unreal Live Training stream that discuss the numerous optimizations, features and stability improvements the Unreal team has made to the engine and mobile platforms supported while also highlighting how Epic leveraged these tools to ship Fortnite on iOS and Android and overcome technical challenges such as memory, performance and battery life for the application.

**Repositories**

When researching into repositories I was looking for a solution that can integrate well into the Unreal Engine and handle big file transfers like textures and .fbx that will be included in my project. The two that stood out where GitHub and Perforce.

GitHub and perforce have many differences that set them apart from one another.

GitHub has a distributed Git model which allows developers to download the source code along with a full version history to their local machine where they can make commits, diffs and merge code fast. Each developer has their own copy of the repository which can be a security risk at big developer companies and make it hard to manage the application data. Typically, the repositories security levels of the project are assigned at a repo level to employees and the projects are broken into several different branches where teams end up dealing with cross repository dependencies that could be a nightmare if not controlled correctly.

Perforce Is Centralized has a centralized model that centres around storing everything in one place ensuring developers always have the latest version. All commits from developers, no matter where they sit across the world are done to the same server where the project is located. This provides improved auditability, traceability and communication because checked out work is easily visible to other developers on the team [Reference] Perforce, P. 2019. Git vs Perforce. April 11. How to Choose. [Online]. [12 June 2019]. Available from: <https://www.perforce.com/blog/vcs/git-vs-perforce-how-choose-and-when-use-both>

**AR Images and Objects**

I was looking deeper into AR this week and wanted to integrate Augmented images and objects into the application as well. I found that it is possible with Google ARCore and Unreal engine and is quite simple to do as well. Unreal can take images the user loaded into the engine and turn them into smart images the camera will recognize, and actions can be executed on recognition of the images. This gives me the ability to make something interesting in that I would be able to create a companion application with the original that would recognize images and cards that could be used as bonus content or power ups by the user. I found documentation on Augmented Images and how to integrate the feature into Unreal Engine at [Reference] Arcore, G.A. 2019. Augmented Images developer guide for Unreal. February 28. Using Augmented Images. [Online]. [13 June 2019]. Available from: https://developers.google.com/ar/develop/unreal/augmented-images/guide

References:

T.S. 2018. Exploring Unreal Engine 420’s Mobile Optimizations. August 27. Exploring Unreal Engine 420’s Mobile Optimizations. [Online]. [10 June 2019]. Available from: <https://www.unrealengine.com/en-US/tech-blog/exploring-unreal-engine-4-20-s-mobile-optimizations>

Epic unreal, U.E. 2018. Android Quick Start. July 16. Getting Started. [Online]. [10 June 2019]. Available from: <https://docs.unrealengine.com/en-US/Platforms/Mobile/Android/GettingStarted/index.html>

Perforce, P. 2019. Git vs Perforce. April 11. How to Choose. [Online]. [12 June 2019]. Available from: <https://www.perforce.com/blog/vcs/git-vs-perforce-how-choose-and-when-use-both>

Arcore, G.A. 2019. Augmented Images developer guide for Unreal. February 28. Using Augmented Images. [Online]. [13 June 2019]. Available from: https://developers.google.com/ar/develop/unreal/augmented-images/guide

Development

Setting up the environment using Unreal and Android works was quick and easy as all SDK downloads required were included with the engine inside the installation directory Epic Games\UE\_4.22\Engine\Extras\AndroidWorks. I continued to get a project template for AR up and running and did my first build for Android mobile on a Galaxy S7 Edge and it worked perfectly without any issues.

I have also continued to build out a basic fish tank and got it placed on a flat surface and had the AR track the object and its location. This was simple to execute with Unreal Engine 4.22 and can see the strides they have made when it comes to mobile development.

I decided to load in one image a painting into the art application and see if I could convert it into an Augmented Image. It too wasn’t complicated, and the image was recognized as soon as it came into view of the camera on my mobile device.

My development environment is now set up with AR functioning as intended and Unreal deploying to mobile successfully. I am starting on my basic prototype of the application and creating assets and basic blueprint functionality that would provide the base of my application. I have uploaded my project to my git repository as source control with GitHub is integrated and working with Unreal Engine as well.

Course work

I started setting up my development environments and SDK’s to start working on a prototype of the application I want to build. Using the Unreal Engine and developing for mobile Android specifically, I needed to setup the SDK’s and configure my environment inside Unreal engine to be mobile ready and have my repositories linked to my GitHub account.

A big discussion the past week was Platform and Tools. What will be used and why? I have decided to stick to the Unreal Engine and Mobile Android development using ARCore from google.

The other topic that came up the past week was Unity vs Unreal. Which engine is the one to use for the project going forward? I have worked on Unity for many years before and work with Unreal Engine atm in my professional career. This has given me a unique outlook on the engines and how they compare with one another.

I have chosen to build my application on the Unreal Engine due to the numerous new tools and advancements made in the mobile space. Unity has been the go-to engine for mobile the past few years and has an amazing workflow but can sometimes lack when it comes to quality of light used in scenes/levels. I want to build my knowledge of Unreal and use this small application to test and integrate as many of their new features released especially around AR and mobile deployment.