Analysis of 2D and 3D Game Engines

An Independent Study

Chloe Solis

Project Proposal

Analysis of 2D & 3D Game Engines:
Their Applications Inside and Outside of Games

Chloe Solis
May 5, 2020
Submitted to: Dr. Daryl Thomas

Abstract

The objective of this proposal is to learn about both 2D and 3D game engines. This will be done by doing a compare and contrast analysis study of multiple game engines. After this, an in depth analysis of one or two game engines of my choice would be conducted in order to get a deeper understanding of how the engines operate. No funding will be required as this is predominantly research oriented, and game engines are available to download online.

Statement of Need

As a computer science major and someone who is interested in video games and how they work, this would be a perfect opportunity to study how someone brand new to the software would be able to learn to use it on their own. With the advent of online gaming, and the spontaneous surge of online gaming activity in 2020 due to COVID-19, learning how to develop games and environments would allow for more self taught creators. The study would also include researching if game engines can have more day to day applications and if the existing methods can be improved on.

Methods

My approach to this study will be to accumulate multiple well known game engine titles into a list. This will be followed by an initial shallow researching process. Research material will come in the form of descriptions of the game engines as self described by the companies and owners, articles written by others, as well as online reviews and opinions.

Following this will be an analysis comparing and contrasting engines on their features, power, compatibility, and other relevant information. From this analysis, I will pick one or two engines that seem best for self learning and teaching based on the information gathered. I will do a deeper study on this/these engine(s) of choice, delving into documentation and instructional videos.

Once the research phase is complete, and if I think there is enough time available I will attempt a deliverable project that will be created using one of the game engines I studied. This project will likely be along the lines of creating a virtual environment or creating a small 2D demo game. If there is not enough time, I will compile all the information I learned, both shallow and in depth, along with the comparisons and documentation, and place it on a website so that others may use the information for future work and information gathering as needed.

Timeline

The shallow researching will most likely take one to two weeks, with the more in depth studies taking closer to three to four weeks depending on how many engines I decided to delve further into. The researching phase is subject to change, as some engines may have much more documentation and reviews than others. The remaining time will be spent teaching myself how to use the engine of my choice, and attempting to create a working environment. If the attempted environment cannot be completed before the course finishes, I will show the project's progress as is, and have the informational website up by course end.

Budget

There will be no budget required, as this is a self teaching study I am choosing to do, and many game engines are available for download online.

Game Engines List

- Unreal Engine https://www.unrealengine.com/
- Unity https://unity.com/
- Amazon Lumberyard https://aws.amazon.com/lumberyard/
- Cry Engine https://cryengine.com/
- GameMaker: Studio 2 https://yoyogames.com/gamemaker
- Godot https://godotengine.org/
- Cocos2d http://www.cocos2d.org/
- Solar 2D (Formerly Corona SDK) https://www.coronalabs.com/
- RPG Maker https://www.rpgmakerweg.com/

About the Engines (Skim)

- **Unreal Engine 4**
  - Owned by Epic Games
  - Multi-platform
  - 2D & 3D
  - Supports VR
  - Good documentation
  - Intuitive UI
  - Multiple templates available
  - Marketplace
  - Good learning support
  - Blueprint visual scripting
- **Unity 2020**
  - Multi-platform
  - 2D & 3D
  - Compatible with multiple OS
  - Can use C# or Javascript
  - Free for individual use
  - License for professionals cost varies from $1,500-$4,500/person
  - Supports VR
  - Good documentation
  - Professional
  - Marketplace
  - Good learning support
- **Amazon Lumberyard**
  - Owned by Amazon
  - 3D
  - Uses AWS and native support for C++

    - No 2D support
    - Supports VR
    - Only learning support is Documentation
- **CryEngine**
    - 2D & 3D
    - Has free learning resources
    - No licensing fees, but membership costs money
    - Stronger graphically than Unity
    - Higher learning curve
    - VR support
    - Documentation
    - Powerful
    - Less intuitive
- **GameMaker Studio 2**
    - 2D
    - “Point-and-click” (not a high learning curve)
    - Limited
    - Uses its own programming language
    - No 3D support
- **Godot**
    - Open source
    - No fees
    - 2D & 3D support
    - Multiplatform
    - C++, C#, GDscript
    - Game editor interface
    - Drag and drop interface
    - Cross platform
    - Good documentation
    - Comparable to Unity and Unreal Engine 4 in terms of power
- **Cocos2d**
    - Sprites based
    - 2D
    - Supports multiple OS
    - Does not support 3D
    - C++
    - Requires external graphics editors
    - Typically used for mobile games
    - Poor documentation
- **Solar 2D (Formerly Corona SDK)**
    - Bare bones/only IDE w game preview
    - Assets need to be made in other programs
    - Requires sprite sheets

    - Lua scripting language
    - Easy API integration if partnered
    - Easy to use simulator
    - Lack of templates that are plug and play
    - No 3D support
- **RPGMaker**
    - 2D
    - Good for making games like Fire Emblem and Pokemon
    - Multi-platform
    - Costs money, but isn't expensive
    - No 3D support
    - Javascript
    - HTML5 export
    - Difficult conversion to MV
    - No hardware accelerate

	2D or 3D	**VR Support**	**Multi-Platform**	**Price**
Unreal	Both	Yes	Yes	Free w/Licensing
Unity	Both	Yes	Yes	Free (Annual Turnover < $100K)
Amazon Lumberyard	3D	Yes	Yes	Free
CryEngine	Both	Yes	Yes	Free w/royalties
GameMaker: Studio 2	2D	Yes	Yes	Free for Windows/Mac
Godot	Both	Yes	Yes	Free
Cocos2d	2D	No	Yes	Free
Solar 2D (Formerly Corona SDK)	2D	No	Yes	Free
RPGMaker	2D	No	Yes	Payment Req.

Game Engine Video & Article Reviews

YouTube Game Engine Reviews

Best Game Engines in 2018 | Graphics, Prices and MORE! (Comparison)

- https://youtu.be/M7IzAFGqf7c?t=71
- Describes engines, rates based on different positives. List features and gives general gloss over of how the engines create games. LIke coding vs visual coding. Engines mentioned: unity, unreal, cryengine, godot.

Why Godot Over Unity or Unreal Engine.

- https://youtu.be/I7BrpcboJno
- Pros and Cons

Unity or Unreal Engine in 2020?

- https://youtu.be/DVcKdsrFj2k
- Unbiased
- Picking which engine is right for you
- Gives pros and Cons

Unity vs Unreal: Which Engine Should You Choose As A Beginner

- https://youtu.be/zsL6LYVYU5c?t=47
- Comparison

How to Pick a Game Engine

- https://youtu.be/ibK3Ds7nDyk?t=101
- Describes engines
- Lists basic features
- Lists pros and cons

Game Engines' Documentation & User Manuals

Unreal Engine 4

https://docs.unrealengine.com/en-US/index.html

Unity

https://docs.unity3d.com/Manual/index.html

Amazon Lumberyard

https://docs.aws.amazon.com/lumberyard/latest/userguide/lumberyard-intro.html

CryEngine

https://docs.cryengine.com/display/CEMANUAL/CRYENGINE+V+Manual

GameMaker: Studio 2

https://docs2.yoyogames.com/

Godot

https://docs.godotengine.org/en/stable/index.html

Cocos2d

https://docs.cocos2d-x.org/cocos2d-x/v4/en/

Solar 2D (Formerly Corona SDK)

https://docs.coronalabs.com/

RPGMaker

https://rmmv.neocities.org/

Game Engine Features, A Look at Unity and Unreal Engine 4

Unreal Engine

> *"Unreal Engine is a complete suite of development tools for anyone working with real-time technology. From design visualizations and cinematic experiences to high-quality games across PC, console, mobile, VR, and AR, Unreal Engine gives you everything you need to start, ship, grow, and stand out from the crowd".*

Unreal Engine's website contains a features list that is all very well described and thought out. However, if someone is not huge on reading, it is a lot to cover. Here I give the features list, with a short layman's terms description for most every listed feature

Pipeline Integration

- **FBX, USD, and Alembic support**
- **Python scripting**
- **Data conversion**
- **Visual Dataprep**
- **LiDAR**
- **Shotgun Integration**

These features allow users to work in parallel, at the same time, on the same project. They can create their own user interfaces and their own recipes for making effects. The use of LiDAR allows for true to life creation with the use of data clouds and map points. This is notably used in a Ubisoft game that features the Notre Dame cathedral. UE4 can read USD files without a complete import and write back changes that can be seen upon reloading. Datasmith allows for non destructive importing.

World Building

- **The Unreal Editor**
- **Scalable foliage**
- **Asset optimization**
- **Mesh editing tools**
- **Landscape and terrain tools**

UE4 is available for Windows, Linux, and MacOS. UE has automatic processes to make your life easier. It can auto generate levels of detail along with geometries and their meshes and textures. Meshes are editable and UE offers its own mesh editing tools. With scalable foliage and landscape tools, UE4 allows for large scale environment creation with layers of meshes,

textures, skyboxes, and more. Think of it as one big photoshop project, with layers and layers of different details, but 3D.

Animation

- **Character Animation Tools**
- **Animation Blueprints**
- **Live Link data streaming**
- **Take Recorder**
- **Sequencer**

Unreal Engine 4 has blueprints specifically made for animation that feature a Skeletal Mesh, along with state machines, kinematics, ragdoll effects, blend spaces, and more. All of this allows users to create a well animated character that is fully customizable to their liking within the engine. It also supports Live Link data streaming, which is a plugin that allows users to stream real time data directly to the engine from performance capture systems. Nonlinear cinematic editing and animation in real time lets the editing be fully collaborative and lets artists work in parallel on the same project, all at once.

Rendering, Lighting, and Materials

- **Forward Rendering**
- **Flexible Material Editor**
- **Photoreal rasterizing and ray tracing in real time**
- **Virtual Texturing**
- **Post-process and screen-space effects**
- **Color-accurate final output**
- **High-quality media output**
- **Advanced Shading models**

These features allow all lighting and shading to be as true to life as the creator is willing to make it. With material editors, texturing, and ray tracing, lighting effects can be taken to a whole new level, having subtleties that go far with making lighting look realistic. With two methods of Virtual Texturing and forward rendering, visuals can be rendered and loaded with optimal respects to the user's CPU and GPU, and all materials can be edited. Color accuracy, anti-aliasing, and post process effects act as a high quality end-of-project render and effects filter, to give your project whatever ambience or mood you desire.

Simulation and Effects

- **Niagara particles and visual effects**
- **Clothing tools**
- **Chaos physics and destruction system**
- **Strand-based hair and fur**

With its chaos physics and destruction system, users can shatter, demolish, and break large scale scenes in high quality, supporting many materials - cloth, stone, hair, metal - and integrates Niagara for particle/visual effects like smoke, dust, water, and all the complicated lighting that goes along. Give clothes realistic physics with varying degrees of application, and give hair and fur a whole new true to life physical performance in a combination of Niagara and Chaos.

Gameplay and Interactivity Authoring

- **Robust multiplayer framework**
- **Advanced AI**
- **Unreal Motion Graphics UI Designer**
- **Variant manager**
- **Blueprint visual scripting system**

UE displays mastery in decades of tested and tried multiplayer frameworks across multiple platforms. Advanced AI allows for characters to be controlled via blueprints or behavior trees and have increased spatial awareness and the ability to optimally and dynamically navigate real time mesh updates. Users can create their own in game user interfaces, and create and edit variants of assets. Blueprint visual scripting allows for prototyping content without actually having to alter the code yourself.

Integrated Media Support

- **Professional video I/O support and playback**
- **Unreal Audio Engine**
- **Media framework**

Unreal supports 4K UHD video and audio at high bit & frame rates, enabling integration of AR and CG into live broadcasts. It also offers a large number of audio tools to help capture that perfect sound. Videos can be played within the engine and can be controlled via C++ or blueprints, and supports Apple ProRes formats on windows.

Platform Support and On-set Tools

- **Multi-platform development**
- **VR, AR, and MR (XR) support**
- **Pixel Streaming**
- **Remote control protocols**
- **Efficient multi-display rendering**
- **Virtual Scouting for filmmakers**
- **Virtual Camera plugin**

From Windows to Xbox, Playstation to the Switch, VR to AR, Unreal Engine 4 supports them all. Unreal works with a large amount of platforms, as well as most every variant of VR/AR that is

available today. Pixel streaming allows users to host on cloud-based GPU, while multi-display rendering lets them render real time content at any resolution on more than one display. Virtual Scouting allows users to explore the world and locations through VR, and cinematographers can drive cameras in the engine using iPads.

Content

- **Quixel Megascans**
- **Industry-specific templates**
- **Marketplace ecosystem**
- **Sample projects**

Quixel Megascans is a library that is full of real-world scans of 3D and 2D assets that is available for free with every UE license. Templates are available depending on what style project you plan on creating, with industry specific options (like architecture). The marketplace offers huge quantities of templates, projects, assets, textures, shapes, and more that vary in price.

Developer Tools

- **Full access to C++ source code**
- **Seamless Perforce Integration**
- **C++ API**
- **Profiling and Performance**

Developers have complete access to UE's C++ source code, which is constantly maintained and updated through GitHub, and the C++ API extends UE functionality even further with live coding. UE has tools that optimize your project to prevent bottle-necking and improve real time performance.

Unity

Animation and Cinematic

- **Animation**
- **2D rigging IK**
- **2D animation**
- **Cinemachine**

Unity offers both 2D and 3D animation, with 2D rigging IK calculating for positions and rotations of chains of bones for your animations' skeletons. Cinemachine is an editing suite for animation that includes tools for dynamic, smart, codeless cameras and allows you to create and experiment with camera behaviors in real time.

Audio and Video 2D

- **Audio components/mixer**
- **Spatializer**
- **DSPGraph audio framework**
- **Video player**

Unity's audio has full 3D spatial sound, real time mixing with hierarchies, and more. The Audio Spatializer SDK is an extension of a plugin that lets you change the way audio is transmitting in the surrounding space, letting you alter what you hear from the left, right, center, back, etc. While the DSPGraph audio framework is technically listed under DOTS, I am including it here as it is a low end audio mixer that acts as a drag and drop audio graph. The component called Video player lets you attach a video file directly to a GameObject's texture.

Data-Oriented Technology Stack (DOTS)

- **C# Job System**
- **Burst compiler**
- **Entity component system**
- **Unity physics and Havok physics**

DOTS lets user create and iterate faster with C#. Both Unity and Havok physics are based off of DOTS framework, meaning they use the same data protocol. This means you can transition your projects physics from one to another without having to completely rebuild your project. Unity is a lightweight customizable physics while Havok offers high end physics with much more complexity.

Editor

- **Package Manager**
- **HUB**
- **Asset database**
- **Prefabs**

The Unity HUB is an application that allows users to manage both Unity projects and installations. The package is a window that contains both assets and tools. The asset database converts data from an asset's source file into a format that can be used in game or in real-time applications, while the Prefab system lets you create, configure, and store GameObjects along with their properties, child classes, components, and assets into what is then called a reusable asset.

Effects, Lighting, Rendering

- **Particle System**
- **Visual effect graph**
- **Global Illumination (GI)**
- **Progressive lightmapper**
- **Reflection probes**
- **Render pipelines**
- **Post processing**
- **Shaders**
- **Ray tracing**

Ray tracing allows for excellent lighting and pulls from information that the user cannot always see. This however, requires a powerful GPU, and thus is not an option for everyone, much less for anyone who has an average PC. Reflection proves are like cameras that capture a spherical view of its surroundings and is stored in a cube map that's available for objects with reflective surfaces. Unity's particle system component can simulate clouds, flame, liquid, smoke, etc. by generating and animating large amounts of small 2D images. Unity offers a wide variety of meshes, materials, shaders, and textures that can be applied to objects within a project. With options available for rendering, post processing is made easy by applying fullscreen filters and effects to a camera's image buffer.

Programming Tools and Optimization

- **C#**
- **APIs and IDE support**
- **Version control**
- **Unity test framework**

- **Unity profiler**
- **Frame Debugger**

Unity allows full access to the Unity Engine and Unity Editor APIs, both of which are in easy to read C#. Unity also has integrated development environment support, which is software that provides tools to easier develop other software. The Unity Test Framework allows users to test their code in Edit and Play mode alike, as well as on alternate platforms.

WorldBuilding and 2D

- **Tilemap and terrain**
- **Brushes**
- **Sprite Shape**
- **2D World building**

Tilemap is a component that handles tiel assets for making 2D levels. Sprite tools also allow for optimal 2D sprite animation and physics. Unity comes with a built-in set of terrain features that let users add landscapes to their game, as well as customizing their own terrains using tools and brushes.

Usage Experience, Creating a Room

Project

For my intro project I decided to recreate Scales Room 115, otherwise known as the Discrete Room. I will build the room in both Unity and Unreal Engine 4, and be comparing them from a beginner's/new user's perspective.

Unity

Initial Startup

As expected, Unity takes a while to start up, but it does not use a large portion of the CPU to do so. Google Chrome takes more computing power with three tabs and a fourth tab playing music. Unity also gives an approximate busy time/wait time for it to initialize or load a project.

Once the project is open, the usage of computing power varies based on what you are doing, but does not usually peak 15% unless loading a complex texture or mesh.

The user interface is a lot to look at. It seems like a professional interface more than a beginner's interface. Clicking and learning could get you only so far.

Creating a Project & General Usage

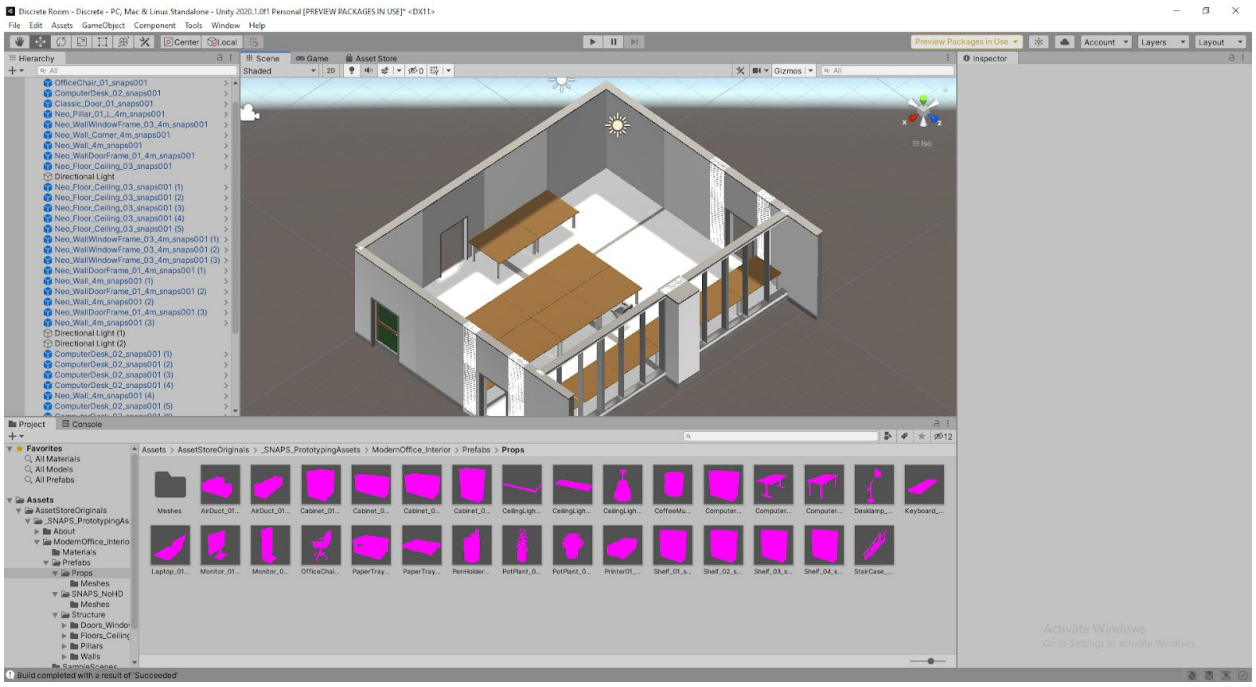
The Unity user interface is not very intuitive for a beginner, much less someone who has never used an editor of any kind before. It is professional looking, with the assumption that you've read the documentation and know what everything is and where it is located.

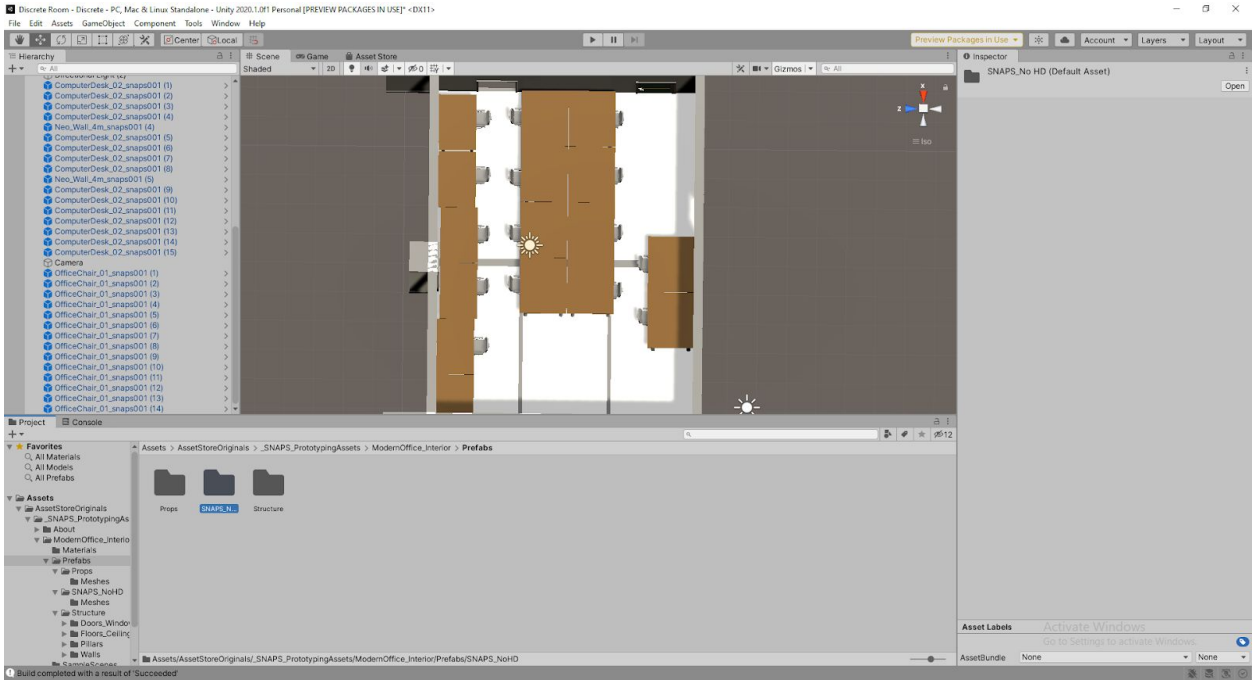
Camera navigation is difficult. Adding assets is a semi easy process. Unity offers an asset store tab, although they have moved the asset store to a website, so it is easier to just have the website saved as a shortcut so you can avoid having to click through the Unity menus only to open up a new browser instance. Compared to UE, there are much more free assets available on the assets store for Unity, but this could just be due to a more saturated market.

The actual creation of an environment is simple and resembles UE up to a point. Where UE has buttons available for different transformation tools, all of Unity's transformations of an object requires inputting values into a details section. Unity offers a more clean lines visual of the environment as well.

Building and Running the project is a bit more difficult. When run, the project automatically attaches the view to the main default camera with no movement capabilities. You have to add a FirstPersonCharacter asset to the project in order to have navigation. I ran into a problem here though, as the script that Unity provides comes with errors, and if used, will prevent the project from building and running. I was able to work around this by following a tutorial online that showed how to make a FPC out of a capsule and camera combination.

I would not recommend Unity to someone just starting out, but it requires a lot less computing power than Unreal Engine 4 does.





Unreal Editor 4

Initial Startup

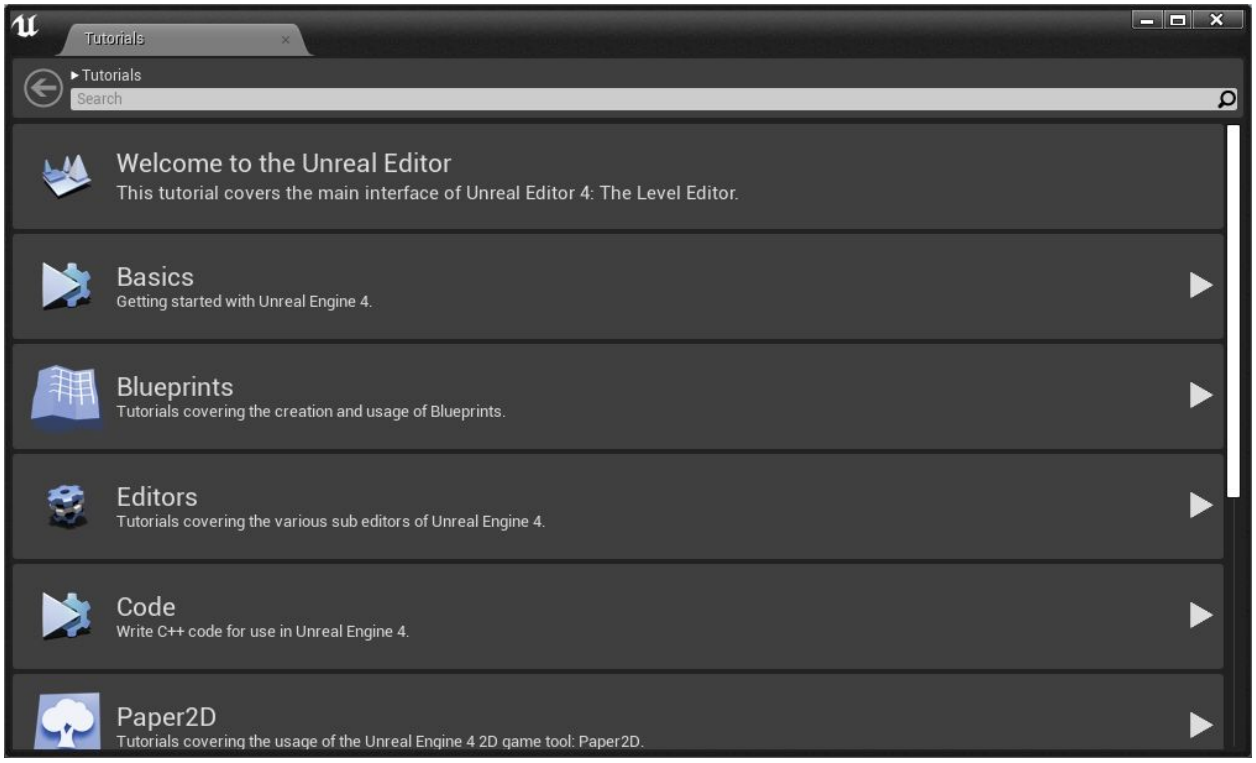
Unreal Engine offers different types of projects and templates when opened. Upon creation of a project, it immediately becomes apparent that UE requires more computing power than Unity does. The first attempt to create a project with real time ray tracing enabled resulted in the initialization process jamming at 45% and going no further. Attempt number 2 to create a project without ray tracing led to uncompromised initialization. The task managers report of CPU usage was quite high during initialization and the compiling of shaders (peaking at around 70%), but settled back down to a range between 0.8% and 5% once it all compiled.

UE immediately offered a tour of the tools it has, and the user interface is much more understandable and simplified in comparison to Unity. UE also explains how to navigate the viewport, whereas Unity assumed previous knowledge, or assumed users have read the documentation.

Creating a Project & General Usage

Unreal Engine 4 is much easier to use for beginners. It's easy to understand user interface makes locating the necessary tools very easy. The tutorial explains each tool and offers more

instruction if desired.



Changing views is made very simple in UE, with the use of a flying camera that is activated by holding right click, and navigating with WASD.as

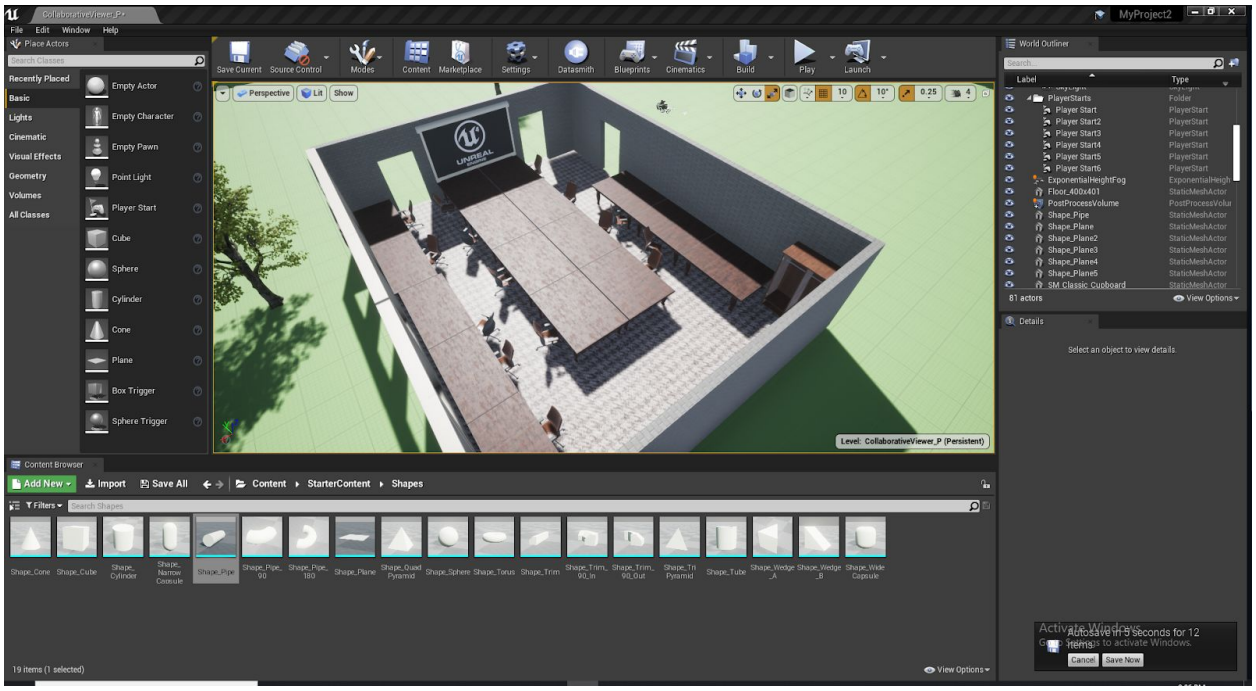
The starter package of materials did not offer the materials I wanted right off the bat, but additional items are available through either the Epic Games Unreal Editor website, or directly in the Epic Games Launcher.

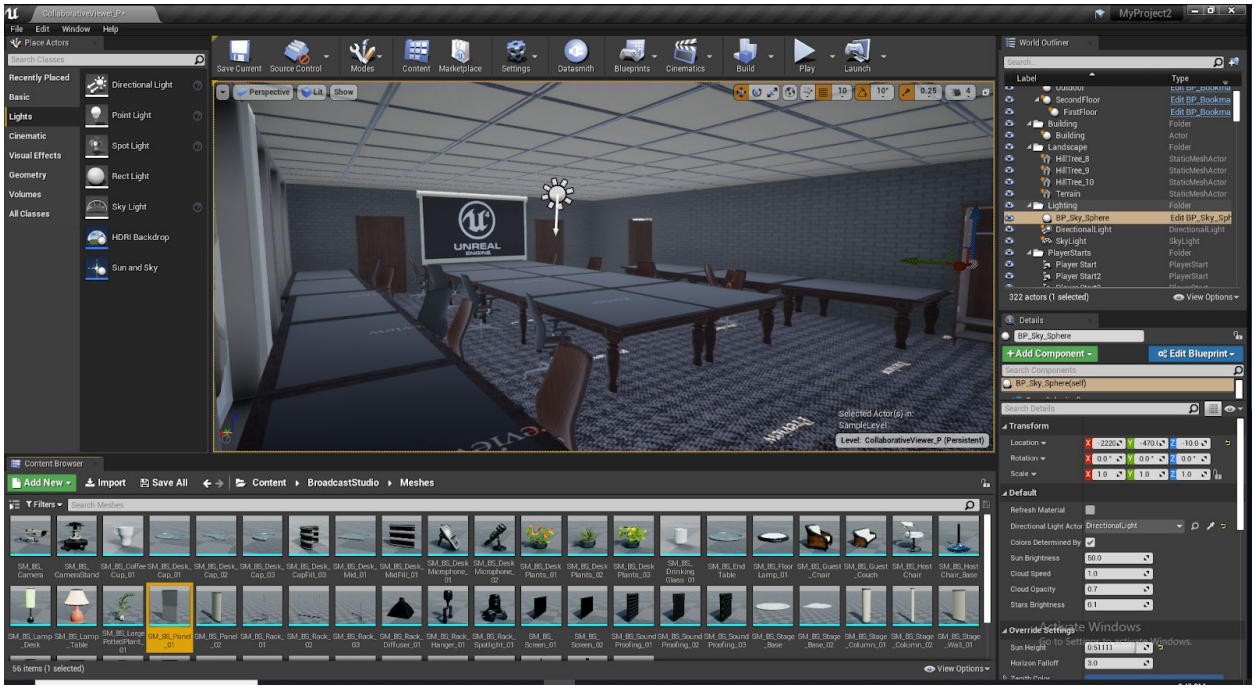
Acquiring free packages is simple and there are numerous options available for purchase as well. Dragging and dropping items and materials is a very easy process. I used the closest items available that resemble the items and furniture in the Discrete Room that were in the store for free. One could conceivably find closer matched textures and meshes within the store with monetary cost, but my goal was to run it entirely free.

Building and Running the creation is relatively simple in UE. You just press the very large build button, and then play. UE has built in methods of navigation already available.

- Paper airplane - basically a flying camera, same controls as viewing
- First Person Character - WASD navigation and camera panning remains the same, but your view is anchored to approximately eye level
- VR - same controls as FPC, but this allows for the materials physics to take effect

Overall Unreal Engine 4 is a very easy to use program if you are a beginner, as much of its UI is intuitive.





Utilization

About

Going into this study, one of the main objectives I had in mind was to research different ways people have and can use these engines outside of the game making environment, and this goal was only intensified during the processes of learning the basics for Unity and Unreal Engine 4. Even if a project is created with the intent of publishing it as a game, it can always be modified and turned into a learning or teaching experience.

A few areas I was interested in looking into were:

- Medical
- Architecture
- Educational
    - Creating remote VR classrooms/environments
    - Tours of historical sites
    - Using 2D games/text led games for interactive learning

Architecture

Notre Dame Cathedral

On April 15, 2019 the Notre Dame Cathedral, an iconic and historical site in Paris, France, went up in flames. It suffered massive structural damage from the fire, along with smoke and water damage. The roof burned, the spire collapsed, and the stone ceiling had numerous holes in it. The art and architecture world had suffered a large blow. While reconstruction efforts are being planned and scheduled, there was a lot of sadness among people who had regrettably never gotten the chance to see the cathedral how it was pre-fire.

However, soon after, a slew of articles were published, where the possibility of aiding the reconstruction could come from an unlikely source, namely, a video game. In Ubisoft's Assassins Creed line of games, a 2014 release called Assassins Creed Unity (referred to hereafter as AC Unity) was looked to. AC Unity takes place in Paris, France during the late 1700s. Players are able to travel and parkour around Paris as it was, but one of the most predominant features was the Notre Dame Cathedral.

An artist named Caroline Miousse was the main reason why the cathedral in the game appeared so true to life. She spent two years and 80% of her time working on the game developing and recreating Notre Dame brick by brick in the engine called AnvilNext2.0. While

the game takes a few artistic liberties, and changes the scale here and there, most of the cathedral is accurate. After the fire of Notre Dame, Ubisoft made the game available for free during the week following, to give people a chance to explore the cathedral.

Medical

Video games are a well known source of practical fine hand eye coordination development. My initial thoughts for medical purposes of games and game engines were on VR. Virtual Reality gear that we currently have often requires head movement, arm movement, height changes, and more advanced gear can also detect finger movement. One of the most obvious uses for this could be with physical therapy, or physical rehab. VR is a good development tool for larger movements such as crouching, walking, and swinging an arm, but it is also a good tool for fine motor control as well. VR controllers typically have buttons, toggles, triggers, and paddles for each hand. Practice with a combination of both fine and large motor control is a very typical trait of physical therapy.

While researching, another article came up (pharmaceutical-technology.com) with an approach I had not thought about. Using a game engine to help develop medicines. During the process of creating a drug, scientists often have to take molecular structures into account. This is often where we see the famous stick and ball models come into play, but these have limitations.

> *“By putting the molecule data into a game engine, and visualising that data in a 3D virtual space, chemists were able to easily get among the compounds, designing and testing at molecular level, running data on different variations efficiently and easily. According to Jones, when C4X CTO and founder Dr Charles Blundell first saw a molecule he’d been working on for years in VR, he immediately spotted a number of flaws in their designs, simply because he’d seen them from a different perspective.”*

Game engines are also being utilized for clinicals and surgical practice. In a physical environment, those doing practical learning, there are often rooms that are dedicated to practice surgeries, clinicals, etc. These rooms require real equipment and tools, which can often be expensive. By using game engines to create simulated environments, true to life scenarios can be created to scale, without risk of equipment maintenance or large amounts of space being required. While VR equipment can be pricey depending on which models you use, they are much easier to maintain than multiple rooms filled with medical equipment

Education

An easy use for education would be using a game engine to create a virtual classroom experience. I thought of this during the latter half of the Spring 2020 semester when schools were shut down in Texas and online learning became the norm. I was taking robots at the time, and I was constantly thinking to myself “Wow, this would be a lot easier if I was actually at

school in person, with the robot". The ability to create and utilize a model of the robot, and even the classroom would have been an excellent use of a game engine at the time. For my game engine learning project, I recreated a classroom I frequently used. The project could also easily be transferred and rebuilt in a VR environment if one has a strong enough GPU, and I believe a VR classroom would be much more conductive to learning than a conference call application could ever hope to be.

At a school I interned in, there was VR equipment that was accessible to the students under teacher supervision. The equipment was used as the lessons called for it. For example, one class was studying red and white blood cells for biology, and they were able to use VR to get an up close look at the cells and how they interacted in the bloodstream.

There are also very accurate scale models created in game engines of historical buildings and sites, museums and aquariums, and other such things. If a museum across the country has a section that is relevant to a lesson, it can be virtually toured from the comfort of a student's own school.

A more obvious use for game engines in education would be teaching students how to use them. They would be able to become familiar with the tools available and would develop the intuitive skill set to go with them that previous generations did not have.

Text based games and game engines are also helpful tools to create interactive text based adventures, stories, and lessons. My senior project was a text guided tour of what SWAU's Computer Science program has to offer. You can either click to button below to jump to the page, or go through the menu bar to Senior Project

References

Architecture

https://www.businessinsider.com/notre-dame-cathedral-assassins-creed-2019-4

https://news.artnet.com/market/how-technologies-old-and-new-will-be-needed-to-rebuild-notre-dame-1520689

https://mashable.com/article/notre-dame-assassins-creed-unity/#:~:text=The%20Notre%2DDame%20Cathedral%2C%20as,seen%20in%20'Assassin's%20Creed%20Unity.&text=The%20Notre%2DDame%20Cathedral%20was,Paris%20during%20the%20French%20Revolution.

https://www.nationalgeographic.com/news/2015/06/150622-andrew-tallon-notre-dame-cathedral-laser-scan-art-history-medieval-gothic/

Medical

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3833380/

https://www.pharmaceutical-technology.com/features/unreal-engine-gaming-ground-breaking-cures/

https://www.researchgate.net/publication/228368856_Evaluation_of_Game_Engines_for_Simulated_Clinical_Training

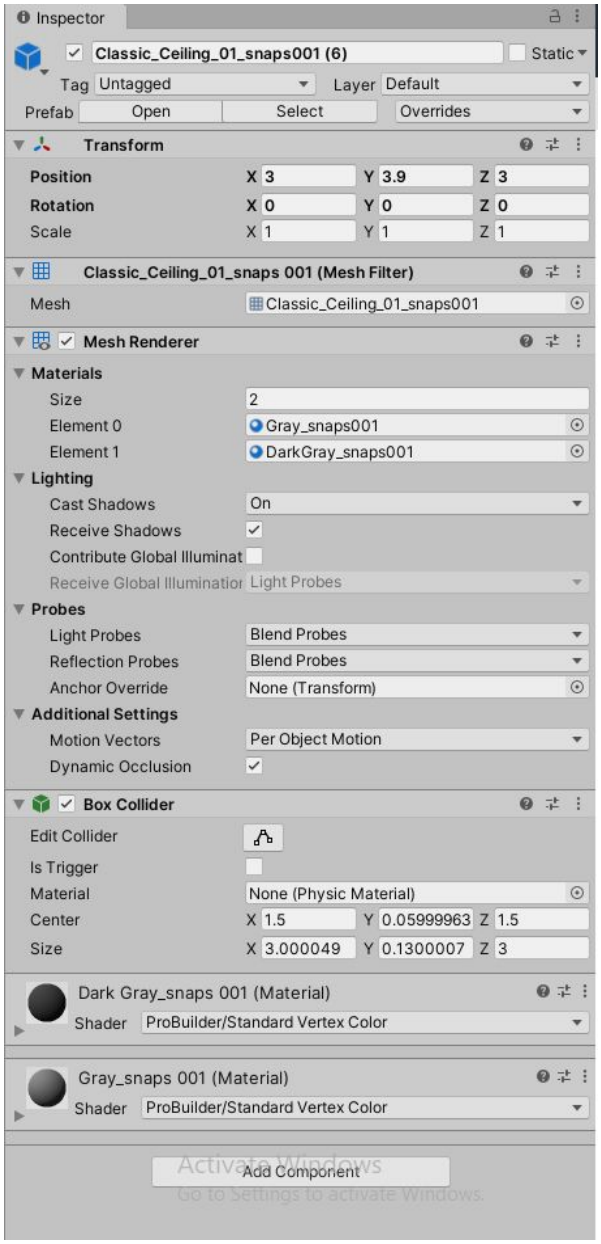
https://dl.acm.org/doi/10.1145/1321261.1321311

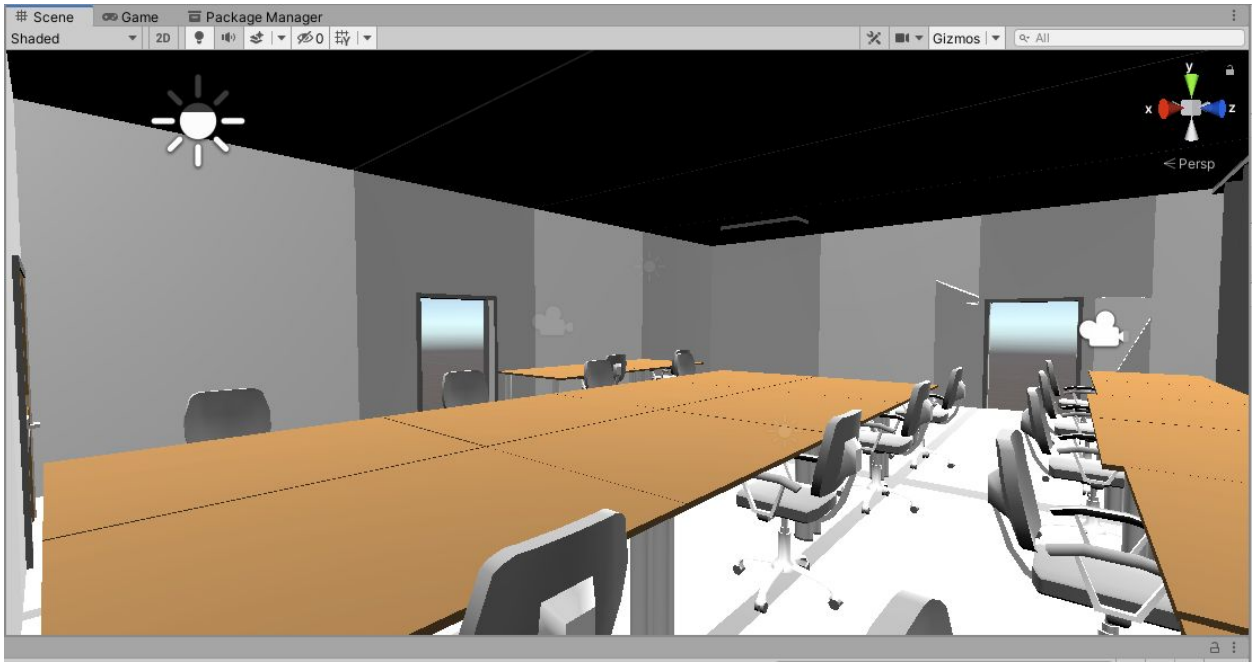
Education

https://www.researchgate.net/publication/319776155_A_Brief_Review_of_Game_Engines_for_Educational_and_Serious_Games_Development

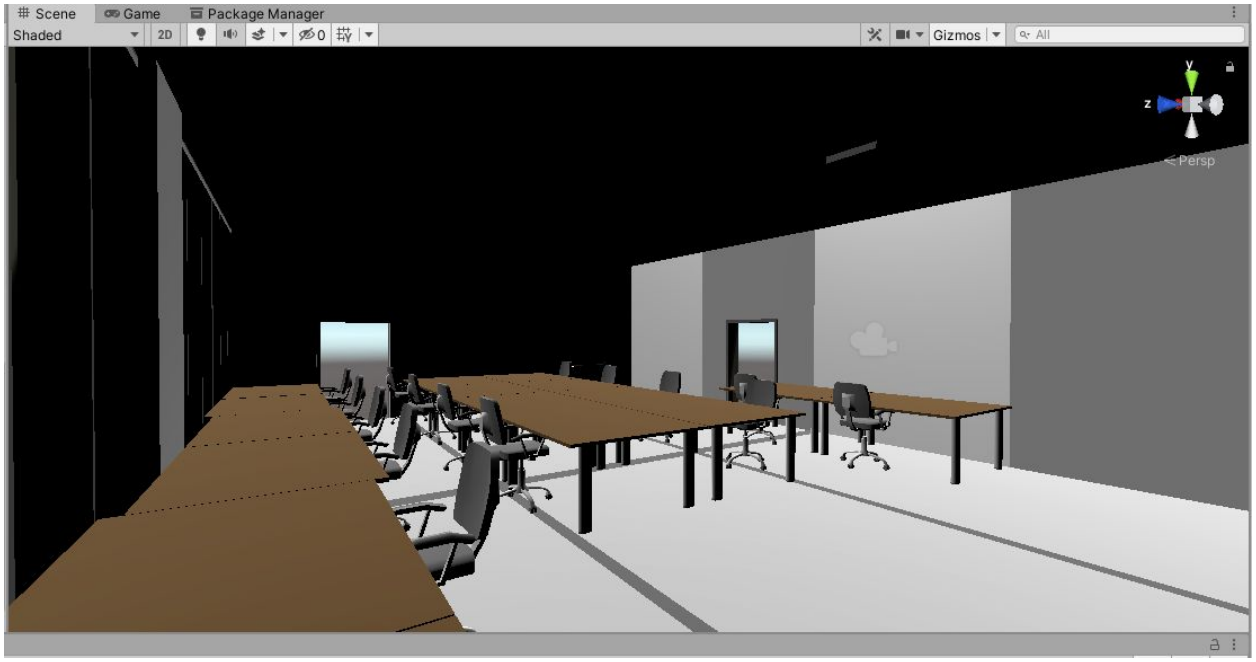
Image Gallery

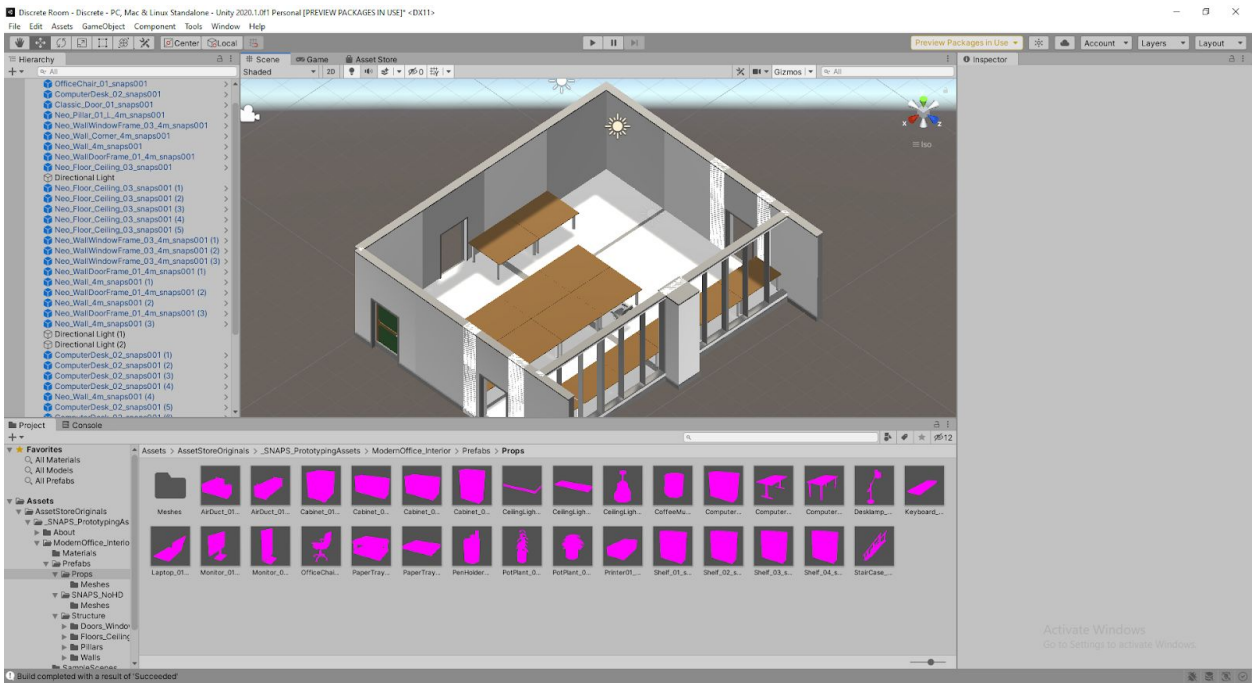
Unity

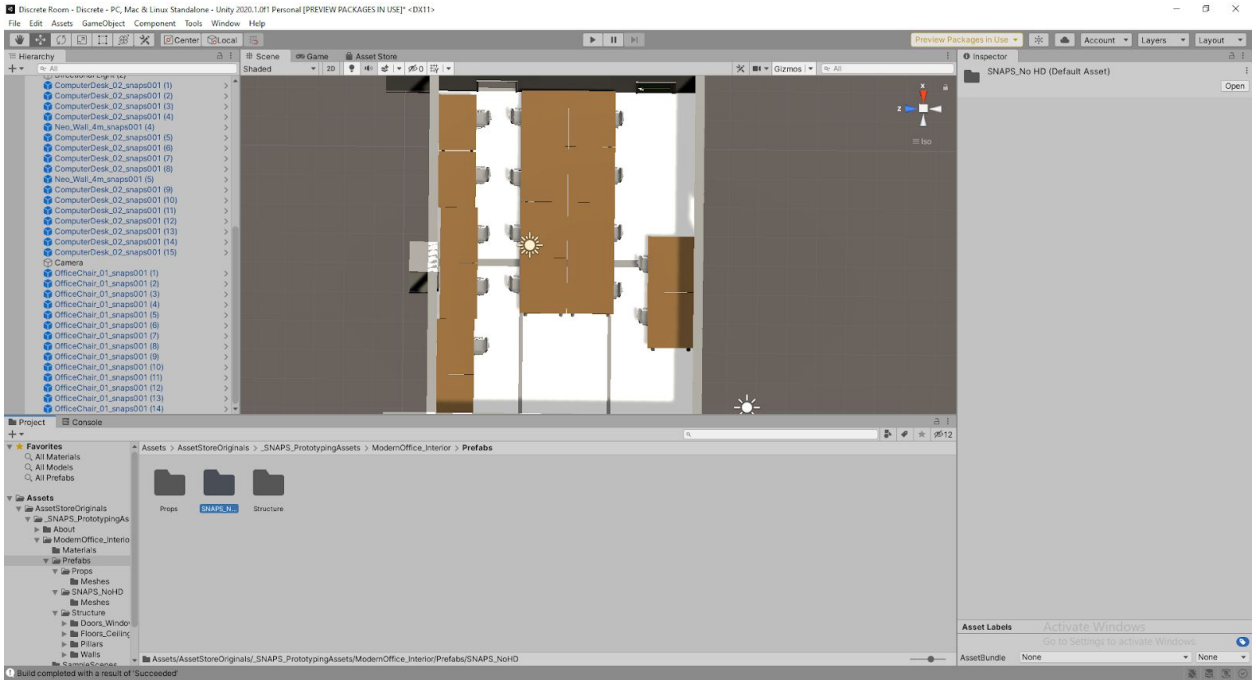



Scene
Game
Package Manager
Shaded
2D
Gizmos
All
Persp

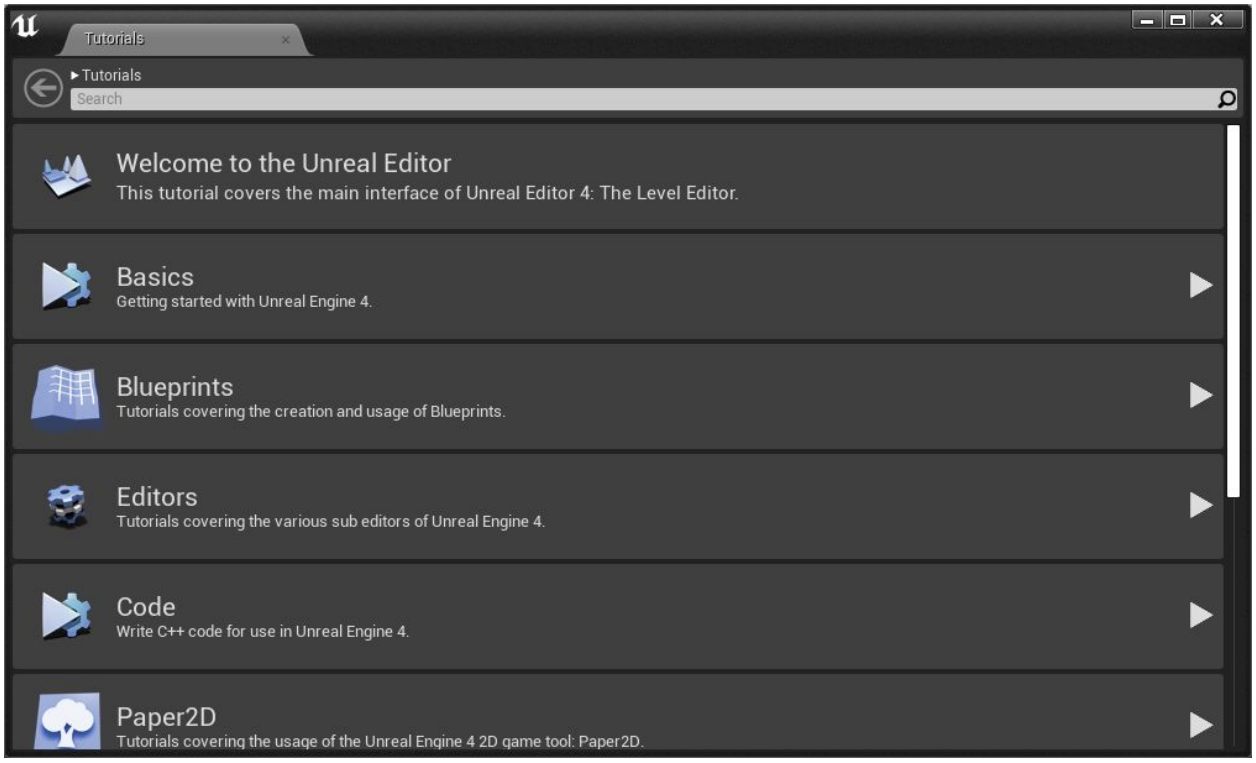

Scene
Game
Package Manager
Shaded
2D
Gizmos
All

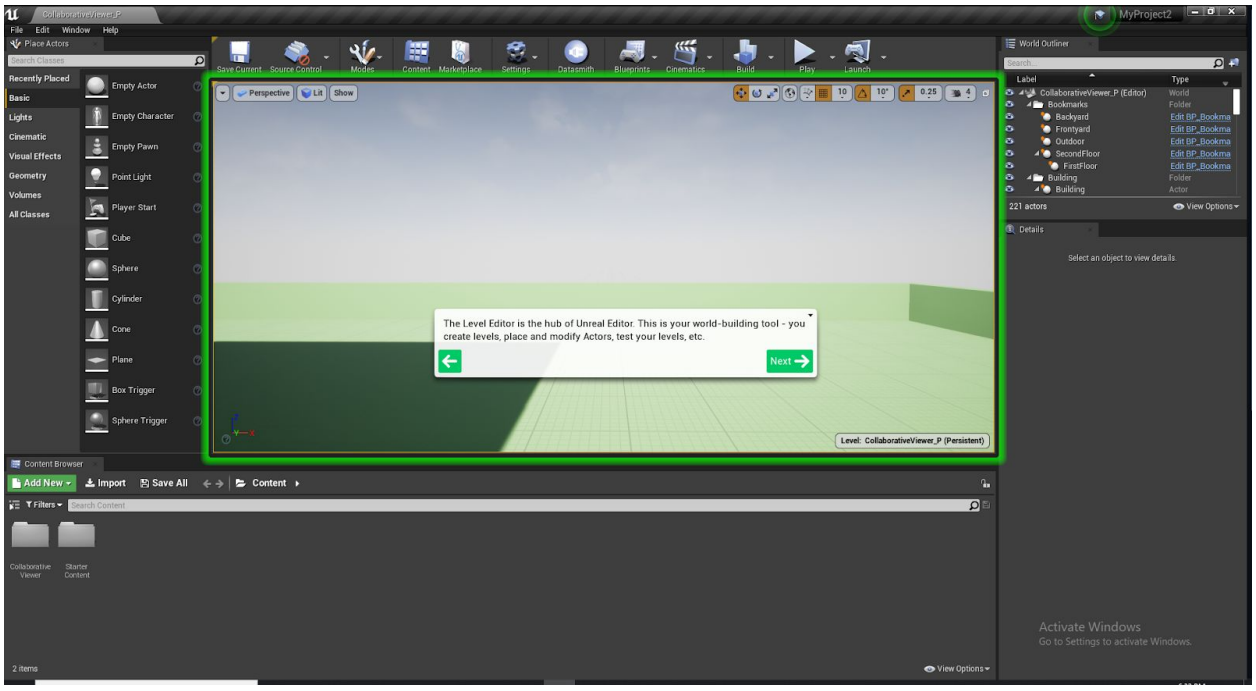

Scene
Game
Package Manager
Shaded
2D
Gizmos
All
Persp

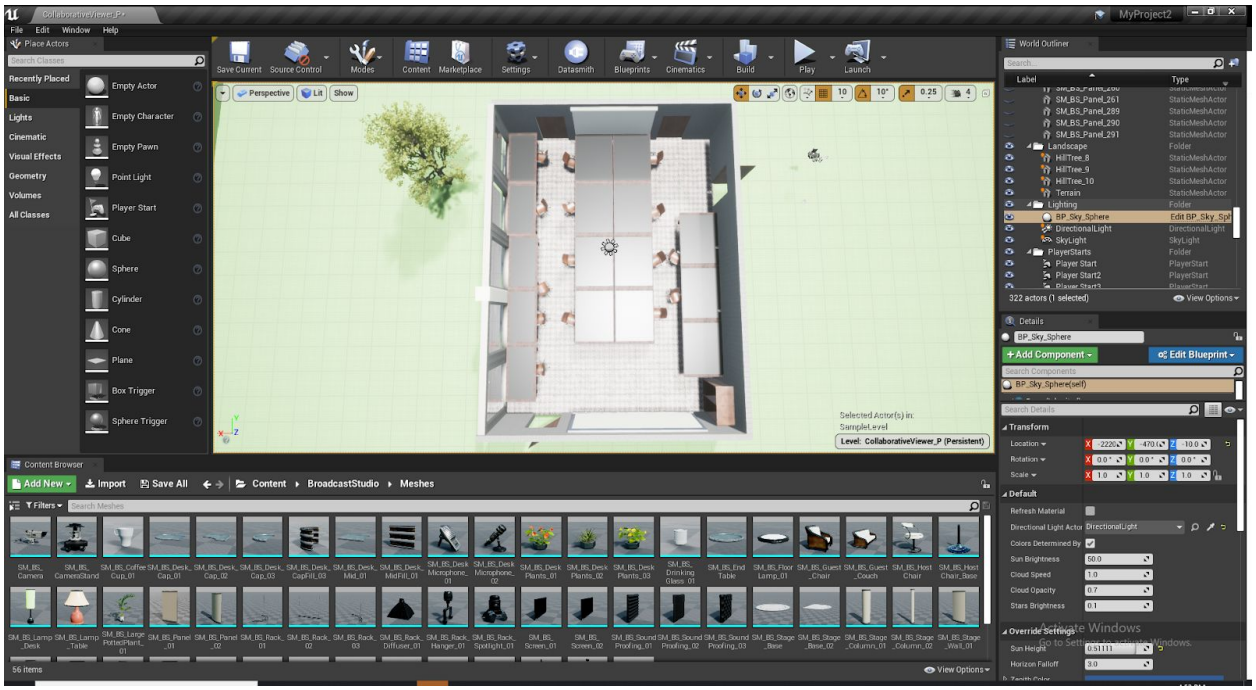

Discrete Room - Discrete - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE]* <DX11>
File Edit Assets GameObject Component Tools Window Help
Hierarchy
Scene
Game
Asset Store
Inspector
Project
Console
Favorites
Assets
Build completed with a result of 'Succeeded'

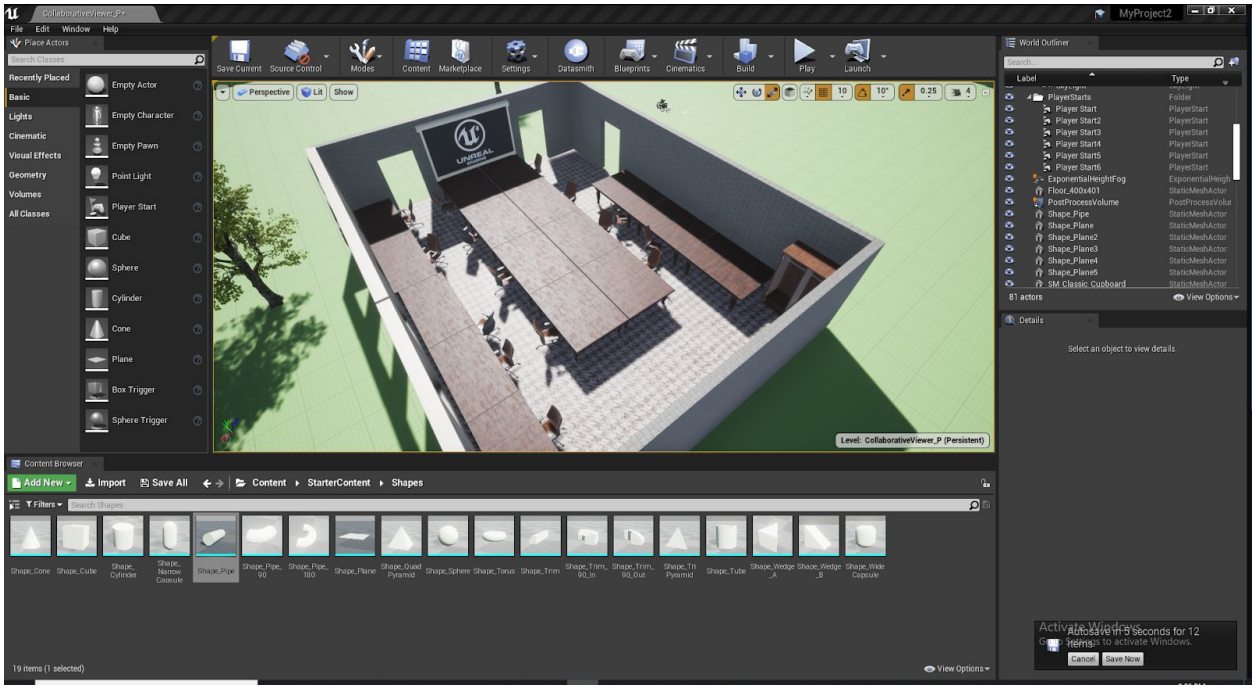

Discrete Room - Discrete - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal [PREVIEW PACKAGES IN USE]* <DX11>
File Edit Assets GameObject Component Tools Window Help
Hierarchy
Scene
Game
Asset Store
Inspector
SNAPS_No HD (Default Asset)
Open
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Assets > AssetStoreOriginals > _SNAPS_PrototypingAssets > ModernOffice_Interior > Prefabs
Props
SNAPS_N...
Structure
Asset Labels
AssetBundle
None
Build completed with a result of 'Succeeded'

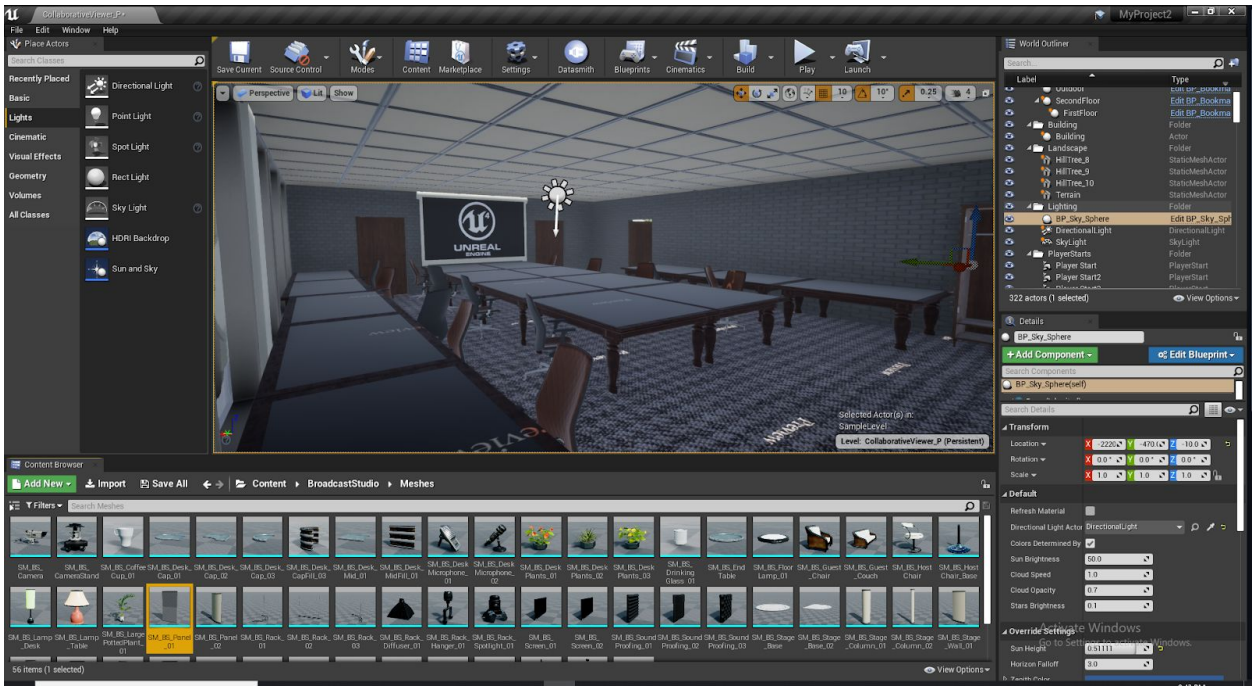
Unreal Engine 4






MyProject2
Place Actors
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty Actor
Empty Character
Empty Pawn
Point Light
Player Start
Cube
Sphere
Cylinder
Cone
Plane
Box Trigger
Sphere Trigger
Save Current
Source Control
Modes
Content
Marketplace
Settings
Datasmith
Blueprints
Cinematics
Build
Play
Launch
Perspective
Lit
Show
World Outliner
Label
Type
322 actors (1 selected)
Details
BP_Sky_Sphere
Add Component
Edit Blueprint
Transform
Default
Override Settings
Content Browser
Add New
Import
Save All
Content
BroadcastStudio
Meshes
56 items
Level: CollaborativeViewer_P (Persistent)


MyProject2
Place Actors
Recently Placed
Basic
Lights
Cinematic
Visual Effects
Geometry
Volumes
All Classes
Empty Actor
Empty Character
Empty Pawn
Point Light
Player Start
Cube
Sphere
Cylinder
Cone
Plane
Box Trigger
Sphere Trigger
Save Current
Source Control
Modes
Content
Marketplace
Settings
Datasmith
Blueprints
Cinematics
Build
Play
Launch
Perspective
Lit
Show
World Outliner
Label
Type
81 actors
Details
Select an object to view details.
Content Browser
Add New
Import
Save All
Content
StarterContent
Shapes
19 items (1 selected)
Level: CollaborativeViewer_P (Persistent)
Autosave in 9 seconds for 12 items
Cancel
Save Now

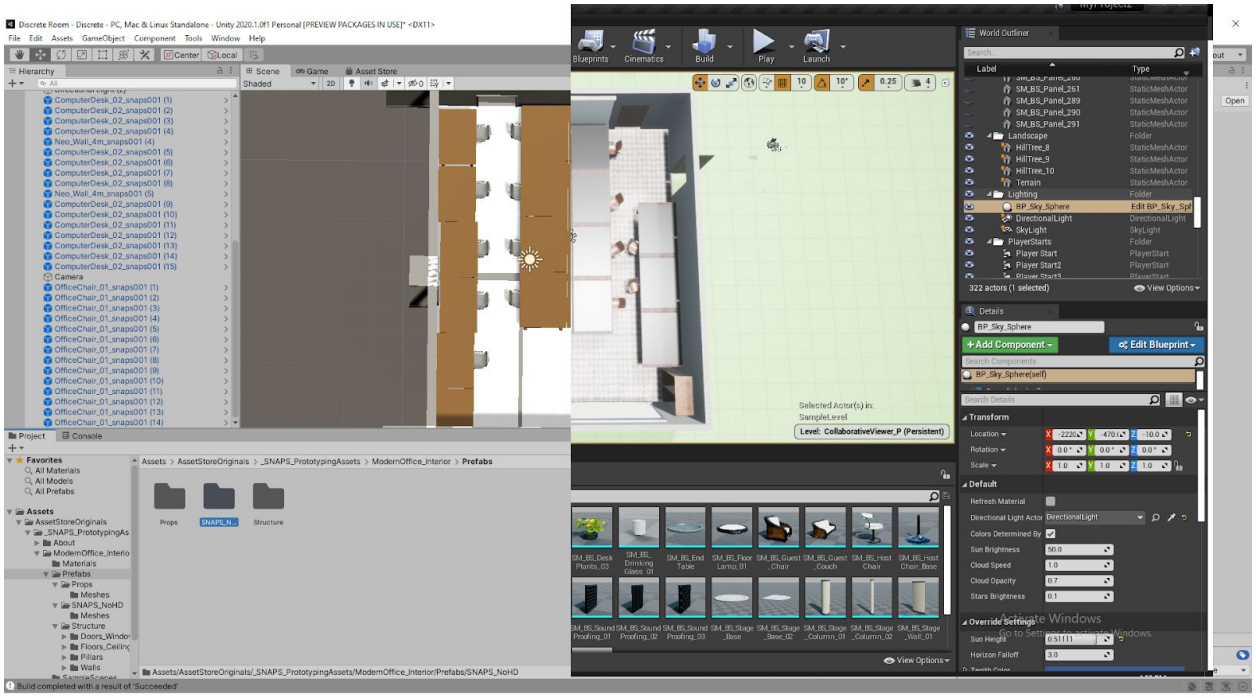


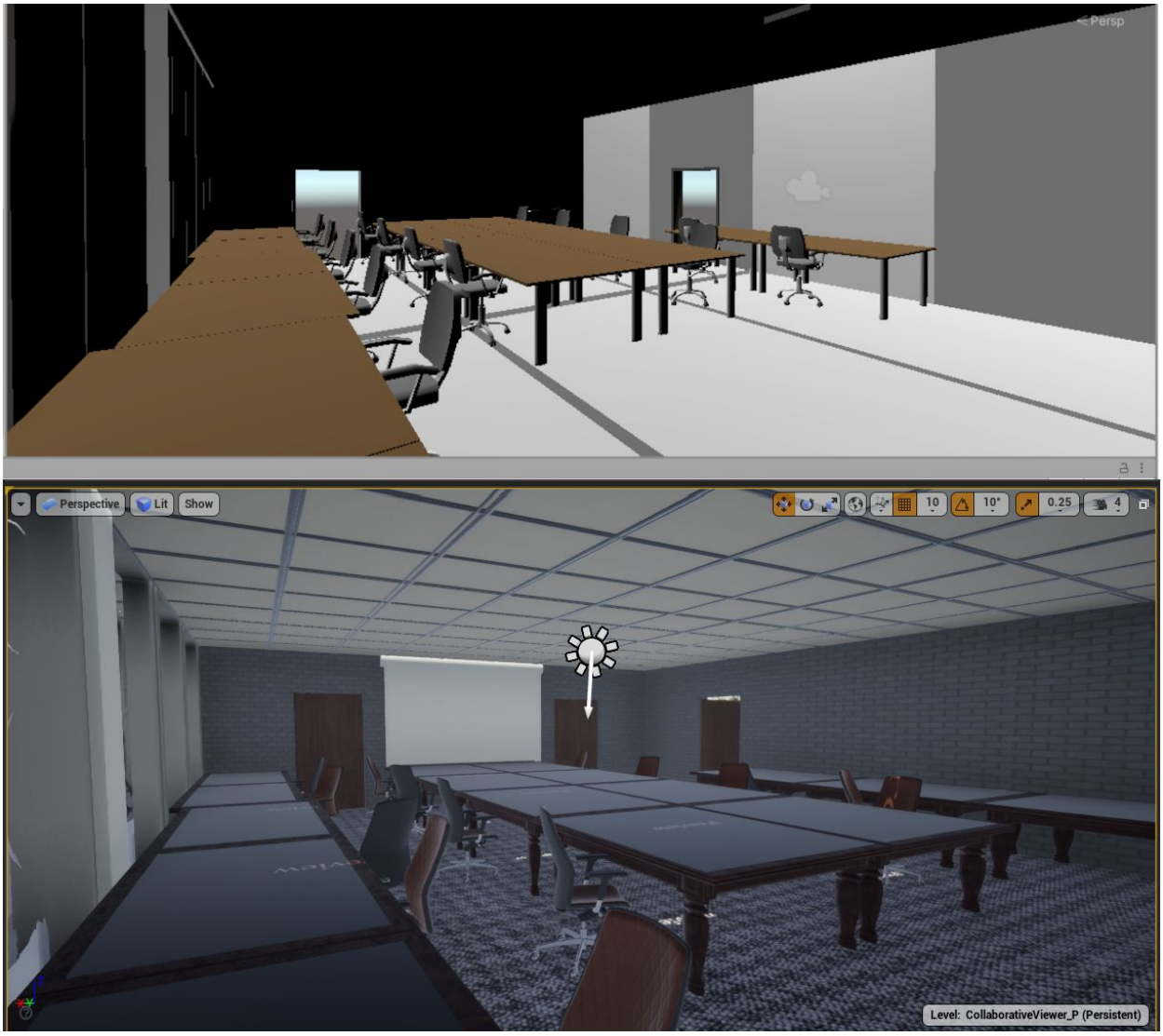

Perspective
Lit
Show
10
10°
0.25
4
Level: CollaborativeViewer_P (Persistent)


Perspective
Lit
Show
10
10°
0.25
4
Level: CollaborativeViewer_P (Persistent)


Perspective
Lit
Show
10
10°
0.25
4
Level: CollaborativeViewer_P (Persistent)

Comparison


Discrete Room - Discrete - PC, Mac & Linux Standalone - Unity 2020.1.0f1 Personal (PREVIEW PACKAGES IN USE)* <DX11>
File Edit Assets GameObject Component Tools Window Help
Hierarchy
Scene
Game
Asset Store
Project
Console
Favorites
Assets
Prefabs
Props
SNAPS_N...
Structure
Build completed with a result of 'Succeeded'
Blueprints
Cinematics
Build
Play
Launch
Selected Actor(s) in: SampleLevel
Level: CollaborativeViewer_P (Persistent)
World Outliner
322 actors (1 selected)
Details
BP_Sky_Sphere
+ Add Component
Edit Blueprint
Transform
Location
Rotation
Scale
Default
Refresh Material
Directional Light Actor
Colors Determined By
Sun Brightness
Cloud Speed
Cloud Opacity
Stars Brightness
Override Settings
Sun Height
Horizon Falloff
Activate Windows
Go to Settings to activate Windows.


Persp
Perspective
Lit
Show
10
10°
0.25
4
Level: CollaborativeViewer_P (Persistent)


Persp
Perspective
Lit
Show
10
10°
0.25
4
Level: CollaborativeViewer_P (Persistent)


Level: CollaborativeViewer_P (Persistent)