

ual:

Welcome Unreal Engine 5 Class



Epic's Lumen technology in Unreal Engine 5.1

Image Credit: Epic Games

Image by Skarletmotion

Who am I ?

Emily Sakai - CG Generalist

More than 15 years of experience in the following industries:
corporate, fashion, advertising and marketing, broadcast, graphic
& industrial design, architectural & medical visualisation and VFX.

Graphic Designer



Motion Graphics Artist/Editor



3D Artist/Animator



CG Generalist

Animation/Rigging



Modelling/Texturing/UV



Lighting/Rendering/Compositing



Effects

ual:

What is Unreal Engine?



UNREAL
ENGINE

Why Unreal Engine for VFX ?

ual:

1- Benefits:

- Real-time tools, Interactive 3d content and reasonably high-quality renders.
- High integration with other DCC (Digital Content Creator) via Datasmith, Nuke's Unreal Reader node, Houdini plugins, etc.
- Adaptable and flexible workflows.
- Remote collaboration and multi-user approach.

Why Unreal Engine for VFX ?

ual:

Usage:

1. Game Development
2. Architecture and Visualisation: immersive walkthrough experience
3. Training and Simulations: aviation, healthcare and military
4. Virtual Reality and Augmented Reality: immersive and interactive virtual environments
5. Films, Animation and TV: real-time content / virtual sets and environments creation

Why Unreal Engine for VFX ?

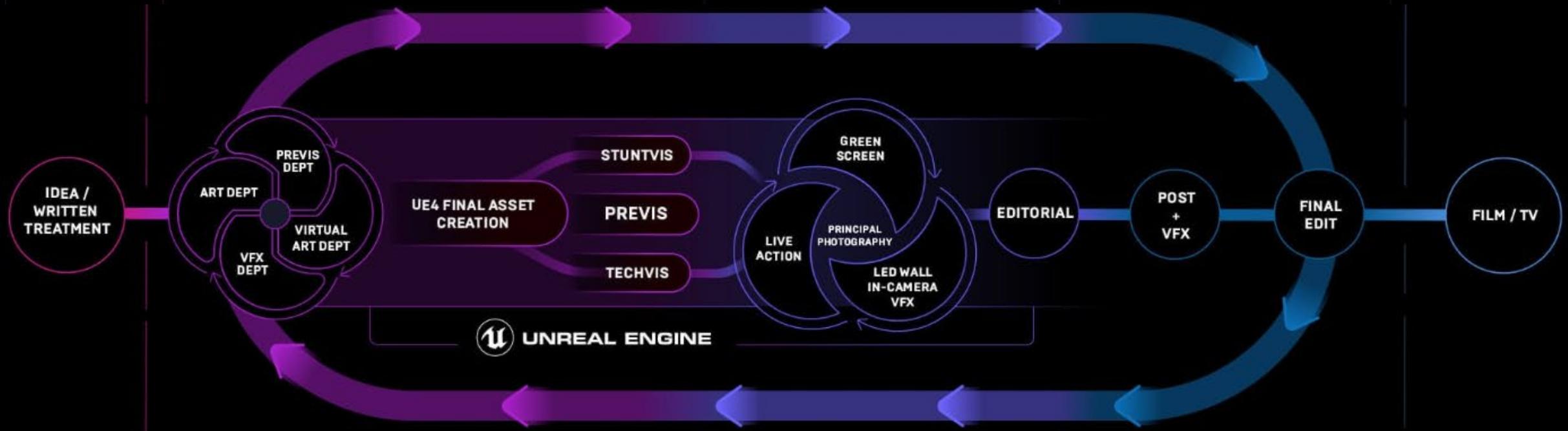
ual:

2- Pipeline Change:

DEVELOPMENT / PRE-PRODUCTION

PRODUCTION

POST-PRODUCTION



Why Unreal Engine for VFX ?

ual:

3- Adoption in Virtual Production (VP) by media organizations + content producers + VFX studios + software developers + hardware manufacturers + indie.

➤ ICVFX (in-camera visual effects) = 2 main types:

- a) LED Wall Cinematography: series of light emitting screens seamlessly integrating and synchronizing the virtual BG + foreground actors and physical assets. Freedom of camera movements with live camera tracking and real-time perspective calculation (parallax).
- b) Projection Screen Cinematography: Form of real-time vfx where tracked, live action green screen is live composited with CGI and physical elements matched.

LED Wall Cinematography Examples:

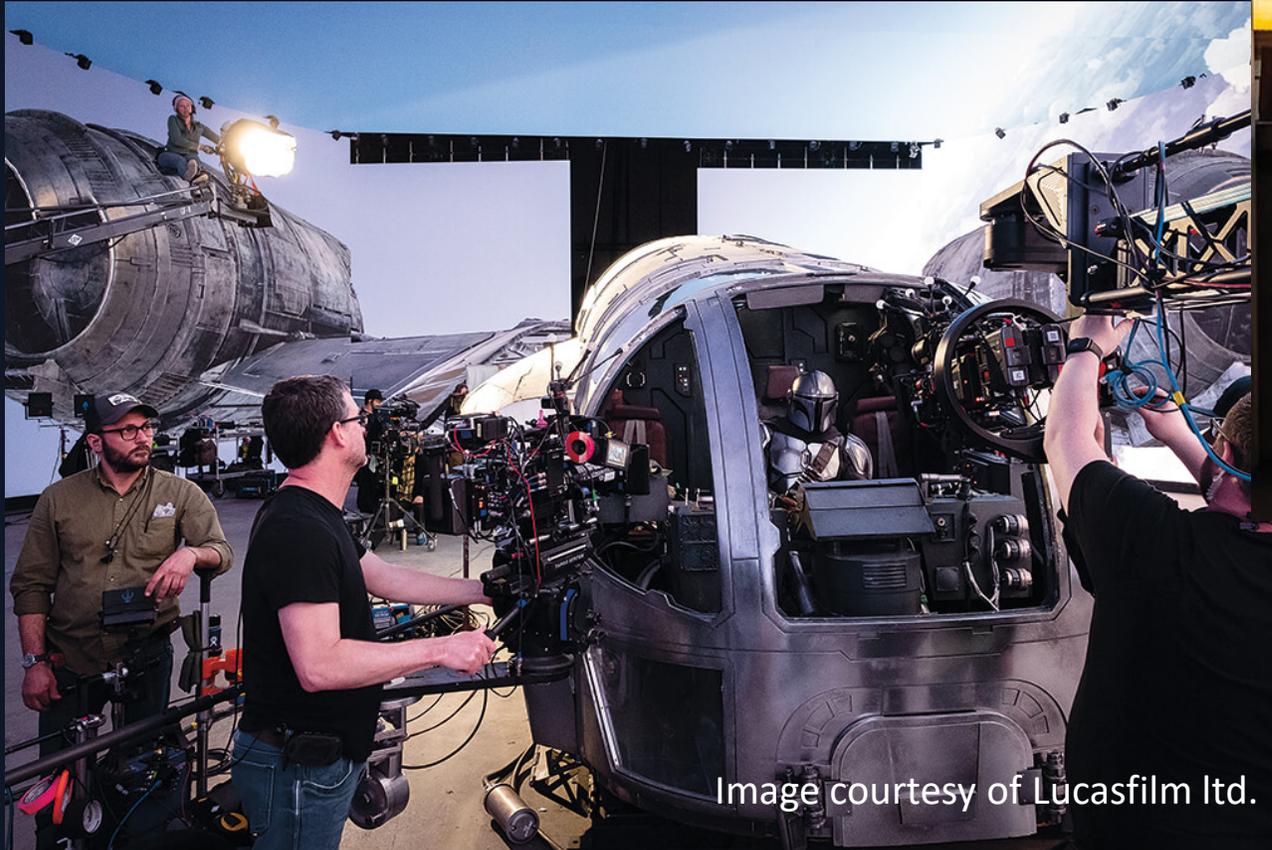


Image courtesy of Lucasfilm Ltd.



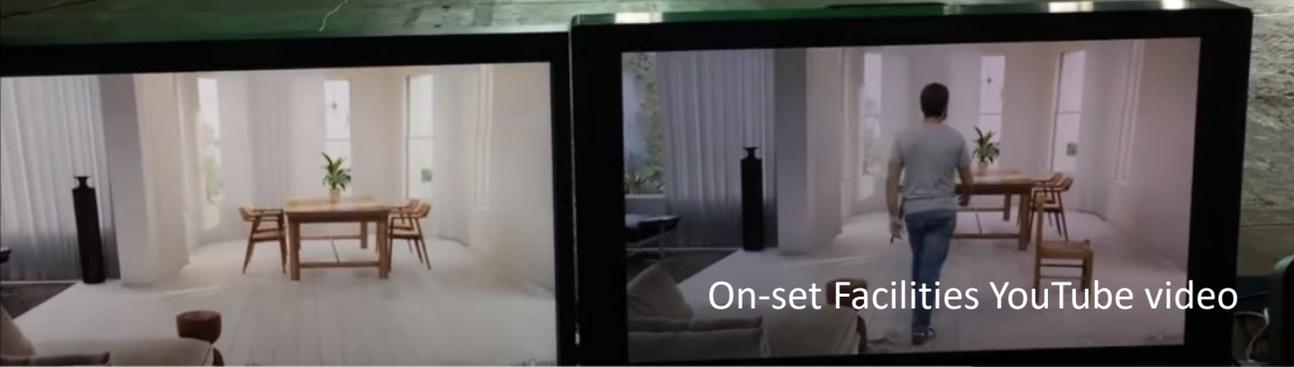
Image courtesy of MELS Studios

Projection Wall Cinematography

Examples:



Unreal Engine YouTube video



On-set Facilities YouTube video

Unreal Engine Application in VFX

ual:



<https://www.youtube.com/watch?v=Ufp8weYYDE8>

Image by Skarletmotion

ual:

Unreal Engine 5.1

Focus on Worldbuilding

UE5 Course Main Content

ual:

1	Introduction: Epic launcher, interface, terminology, navigation etc
2	Asset gathering and Quixel library
3	Creating and Editing Static Meshes
4	Materials, Textures and UV
5	Landscape, Foliage and Water Systems
6	Lighting, Atmospheric and Post Process Volume (PPV)
7	Cameras, PPV and Sequencer
8	Rendering and Movie Render Queue (MRQ)
9	Niagara Particles
10	Blueprints Visual Scripting

ual:

1- UE5 Introduction

UE5 Introduction Class Overview

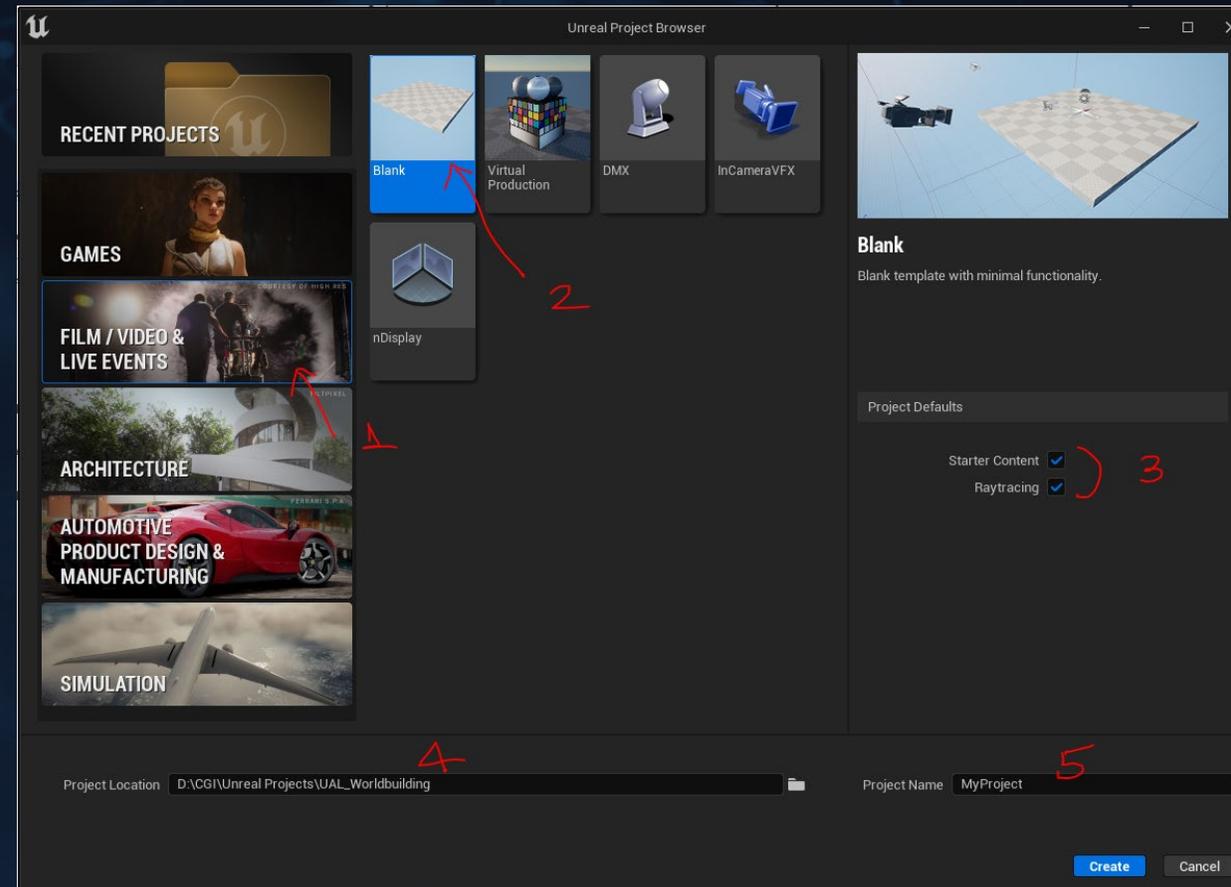
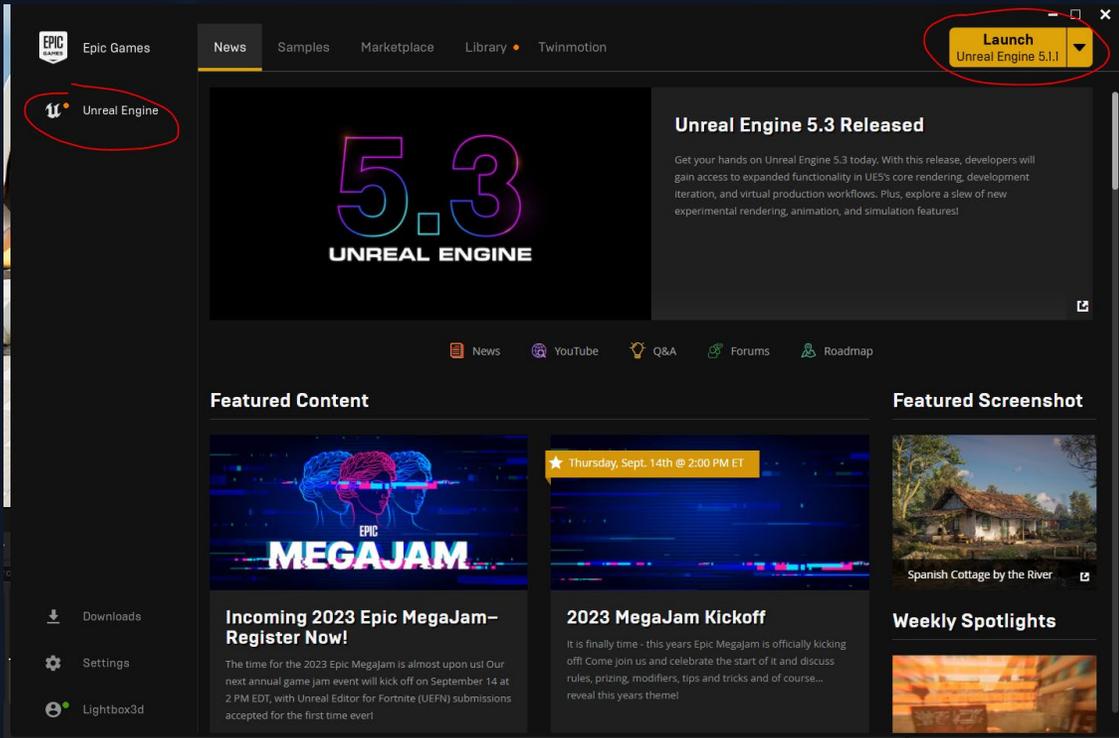
ual:

1. Epic Games Launcher and project creation
2. Unreal Engine interface and navigation
3. Files/directory Management
4. Terminology and naming conventions
5. Quixel Bridge and external libraries
6. Intellectual Property
7. In-Class Project: Log Cabin in the Woods
 - Real-time Project Setup
 - References
 - Blockout

1- Epic Games Launcher and Project Creation

ual:

You need Epic Launcher installed to have access to Unreal Engine 5.1 and the Epic Marketplace:
Unreal Engine Website



2- UE5 Interface and Navigation

ual:

Important Links:

[Foundational Knowledge](#)

[Level Editor](#)

[Interface](#)

[Viewport Controls](#)

[Content Browser Interface](#)



1-Menu

2- Top Toolbar

4- Viewport Navigation

3-Bottom Toolbar

ual:

2- UE5 Interface and Navig.

Editor Preferences: necessary settings to set as Default:

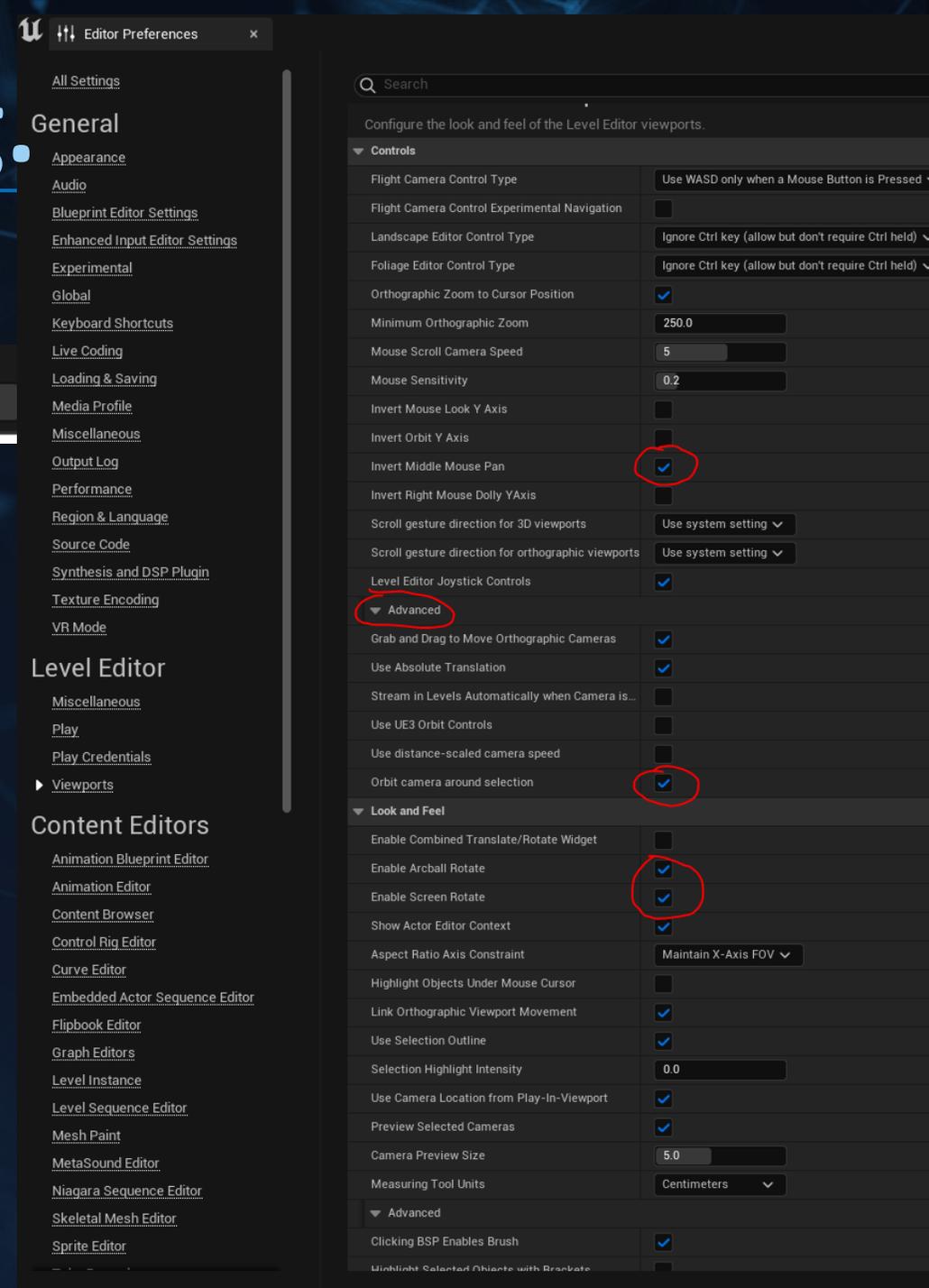
Set as Default

Export...

Import...

Reset to Defaults

- Invert Middle Mouse Pan
- Orbit camera around selection
- Enable Arcball Rotate
- Enable Screen Rotate
- Left Mouse Drag Does Marquee in Level Sequence Editor (search in the bar for 'marquee')



3- UE5 Files/Directory Management

ual:

UE Project: .uproject

All UE Assets (materials, geometry, textures etc: .uasset

The screenshot shows a Windows File Explorer window with the following structure:

- Config
- Content
- DerivedDataCache
- Intermediate
- Platforms
- Saved
- ZIPPED
- CabinWoods.uproject

The breadcrumb path is: CGI > Unreal Projects > UAL_Worldbuilding > CabinWoods > Content > EuropeanHornbeam > Textures >. The 'Textures' folder is circled in red. Below the breadcrumb, a table lists three UASSET files:

Name	Date modified	Type	Size
T_EuropeanHornbeam_Tileable_Albedo.uasset	20/09/2023 02:02	UASSET File	133,107 KB
T_EuropeanHornbeam_Tileable_Mask.uasset	18/09/2023 14:02	UASSET File	157,029 KB
T_EuropeanHornbeam_Tileable_Normal.uasset	18/09/2023 14:02	UASSET File	157,447 KB

The breadcrumb path is: CGI > Unreal Projects > UAL_Worldbuilding > CabinWoods > Content > EuropeanHornbeam > Geometry > SimpleWind. The 'Geometry' folder is circled in red. Below the breadcrumb, a table lists two UASSET files:

Name	Date modified	Type	Size
SM_EuropeanHornbeam_Forest_04.uasset	18/09/2023 14:02	UASSET File	31,947 KB
SM_EuropeanHornbeam_Forest_05.uasset	18/09/2023 14:02	UASSET File	37,905 KB

The screenshot shows the Unreal Engine Content Browser with the following structure:

- Content Browser
- Favorites
- CabinWoods
 - All
 - Content
 - A_UAL_Training
 - Characters
 - Developers
 - EF_FirstFloor
 - EuropeanHornbeam
 - FL_DomeSwamp
 - Geometry
 - Megascans
 - MS_BrushesTradV1
 - MS_ForestFloorV1
 - MS_NorForest2
 - MSPresets
 - StarterContent
 - StylizedWinterCity

- Engine

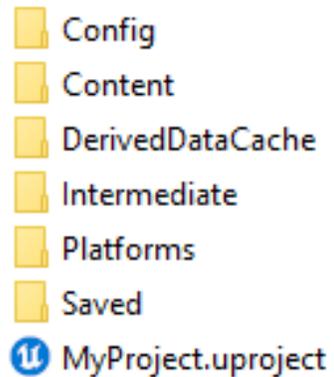
The screenshot shows the Unreal Engine Content Browser with the following structure:

- Content Browser
- Unreal Projects > OneHour_5.1.1 > OneHourProject > Content
- Name
- Assets:
 - _ExternalActors_
 - _ExternalObjects_
 - A_UAL_Training
 - Characters
 - Collections
 - Developers
 - EF_FirstFloor
 - EuropeanHornbeam
 - FL_DomeSwamp
 - FL_Waterbody
 - ForestLandscape
 - Geometry
 - GrassLandscape
 - LevelPrototyping
 - Megascans
 - MS_BrushesTradV1
 - MS_ForestFloorV1
 - MS_NorForest2
 - MS_Valley
 - MSPresets
 - StarterContent
 - ThirdPerson
 - Track

3- UE5 Files/Directory Management

ual:

Project Structure: it holds all the contents of your project: blueprints, materials, textures, meshes etc. File format is .uproject



- **Config:** Project and UE settings. If deleted, you will reset everything you have setup so far, so be mindful.
- **Content:** Most Important folder containing all assets. Don't delete it.
- **Derived Data Cache (DDC):** used to compile and store shaders. If deleted, your project will take time to open. It will be recreated on restart.
- **Intermediate:** Temporary files used when building project or using the engine. If deleted, it will be regenerated on the next restart but will take a while.
- **Saved:** Important files such as autosaves and logs. You can delete except for the Config folder and it won't be recreated.

* If deleting folders, always make a backup copy.

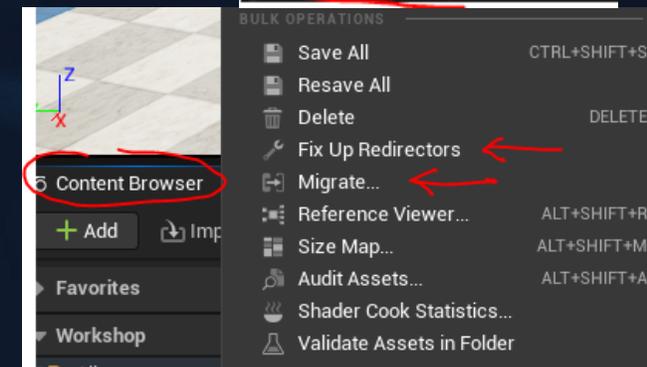
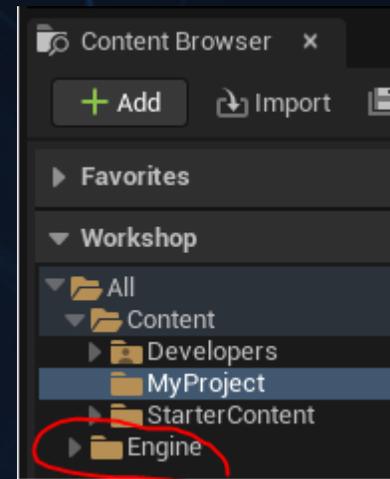
3- UE5 Files/Directory Management

ual:

Link: [Content Directory Structure](#)

Best Practices:

- Management should be done within Content Browser inside UE5 (and not Windows Explorer) to avoid breaking dependencies. Therefore, tasks like renaming, moving assets, importing, deleting should be done within Unreal Engine.
- In case of broken dependencies: 'Fix up Redirectors'
- Exporting assets from projects: use 'migrate' in the Content Browser.
- Engine Content: don't modify assets here.



TASK 1

- 1- Get used to Interface and Navigation
- 2- Familiarise with files and directory

4- Terminology and Naming Convention

Links:

[Terminology in UE5](#)

[Glossary in UE5](#)

[Style Guide used in the industry](#)

[Naming Convention in UE5](#)

4- Terminology and Naming Convention

Beware of proper naming of assets imported to UE5:



4- Terminology and Naming Convention

Main Naming Convention:

[Asset Type Prefix]_[Base Asset Name]_[Descriptor or Property]_[Suffix]

Example: M_wood_castle_01 (material)

SM_rock_garden_V1 (static mesh or geometry)

T_concrete_N (texture normal)

A- Never use the following unless absolutely forced to:

- White space of any kind
- Backward slashes '\'
- Symbols i.e. '#!@\$%'
- Any Unicode character like:
ø ñ ë æ

B- Strive to only have the following characters when possible:

- ABCDEFGHIJKLMNOPQRSTUVWXYZ
- abcdefghijklmnopqrstuvwxyz
- 1234567890
- _ (sparingly)

4- Terminology and Naming Convention

There are a few different ways you can name your assets. Here are some common casing types:

PascalCase = Capitalize every word and remove all spaces, e.g. DesertEagle, StyleGuide, ASeriesOfWords.

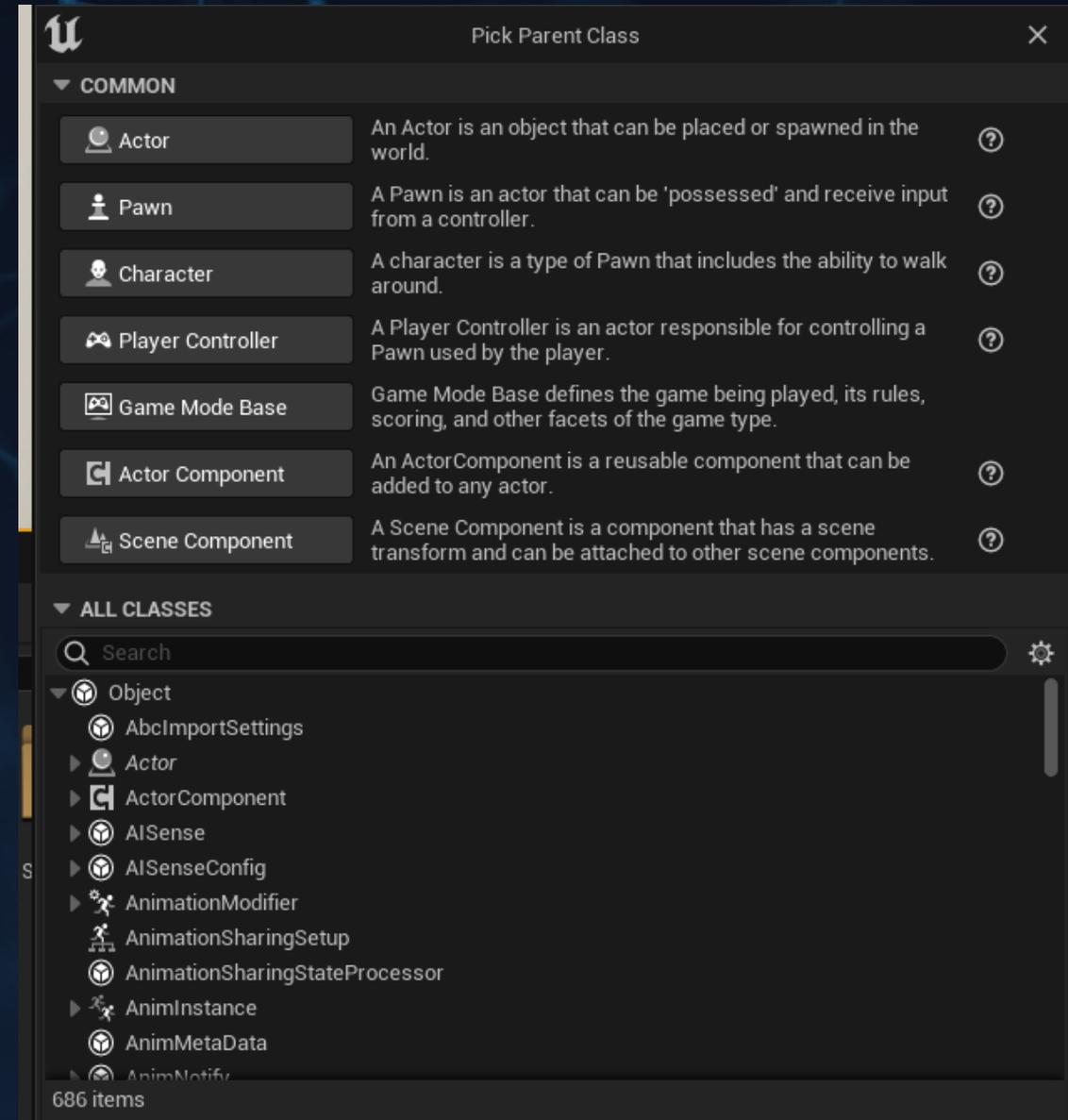
camelCase = The first letter is always lowercase but every following word starts with uppercase, e.g. desertEagle, styleGuide, aSeriesOfWords.

Snake_case = Words can arbitrarily start upper or lowercase but words are separated by an underscore, e.g. desert_Eagle, Style_Guide, a_Series_of_Words.

4- Terminology and Naming Convention

CLASSES:

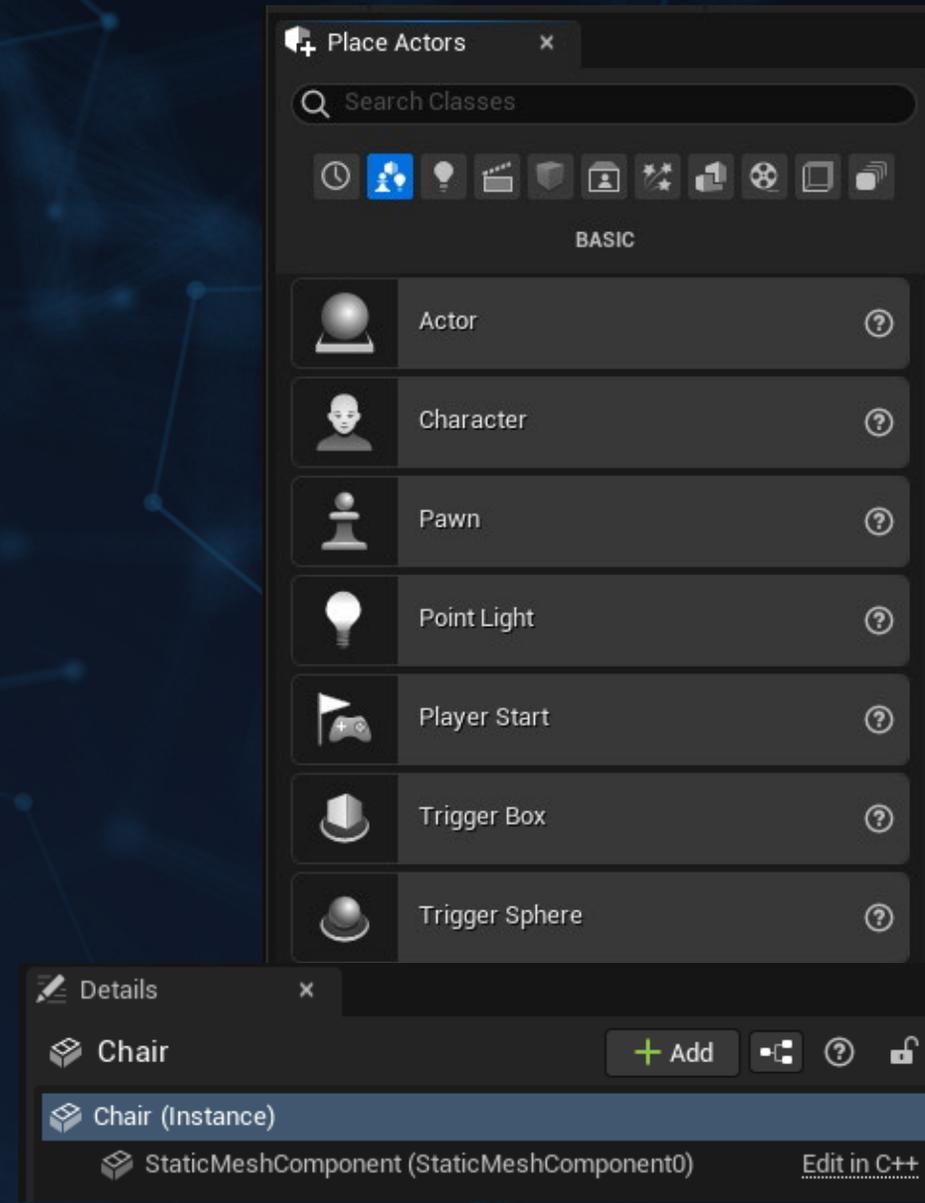
- Defines the behaviours and properties.
- They are Hierarchical and all subclasses will inherit information from its parent class.
- 'Objects' are the most basic class, act like building blocks and have essential functionality for your assets.



4- Terminology and Naming Convention

ACTORS:

- Any object that can be placed into a level, such as camera, static mesh, skeleton mesh, lights etc.
- They support 3d transformations such as translation, rotation, and scaling.
- By attaching 'Component', you can add functionality to the Actors such as: skeletons, emit lights, rotate etc
- Components must be attached to an Actor and can't exist by themselves.



4- Terminology and Naming Convention

PAWN:

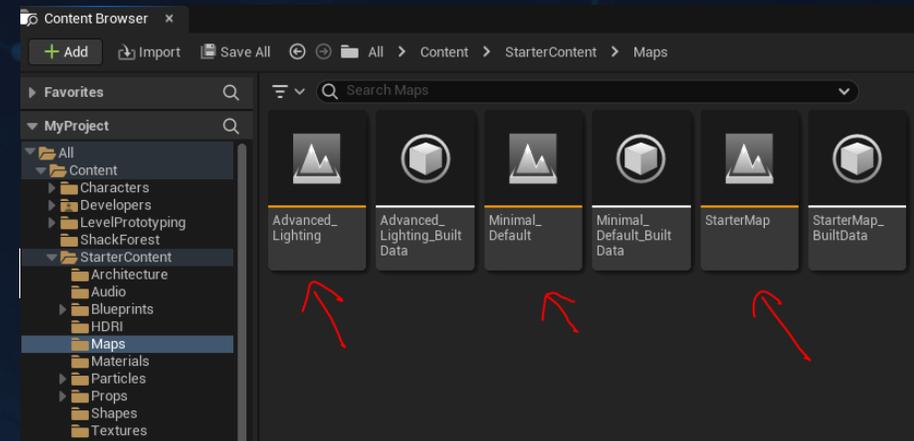
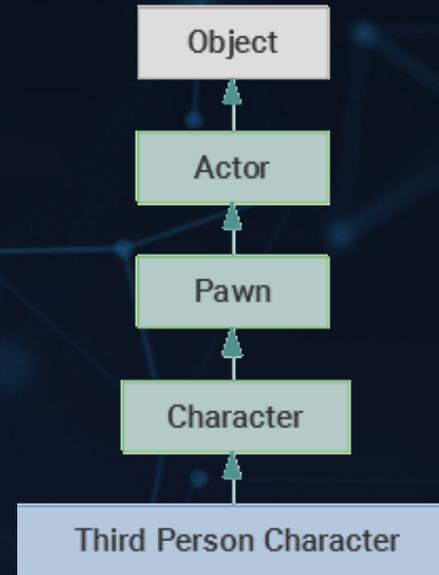
- Subclass of Actor and serve as an in-game avatar or persona. It can be controlled ('possessed') by a human or AI player

CHARACTER:

- Subclass of a Pawn Actor used as a player character

LEVEL OR MAPS:

- Levels contain everything a player can see and interact with, such as geometry, pawns, and actors.
- Also called Maps because Unreal saves each separate .umaps



5- Quixel Bridge and external libraries

ual:

Optimized for UE:

- Epic Marketplace, Quixel Megascan and Sketchfab

Other good libraries with free assets:

- CGTrader, Cubebrush, Turbosquid and Kitbash3d

5- Quixel Bridge and external libraries

ual:

- Find a good balance between using them fully, 'kitbashing'* or build models from scratch depending on:
 - Your career goals
 - Time availability
 - Labour availability
 - Project scope
- The more you create yourself in other DCC and importing into UE5, the more skilled you will become in modelling skills.

** 'Kitbashing' is combining different assets to create something original and new.*

5- Quixel Bridge and external libraries

ual:

Guidelines for using libraries:

- UE5 still don't have the necessary tools 'out of the box' to build models from scratch with high quality, so libraries are very useful.
- Assets brought from libraries can be used in different ways:
 - Edited with UE5 basic modelling tools
 - Modular building
 - Kitbashing
 - Mixed with other assets imported from DCC
 - Mixed with Blueprint procedural modeling

6- Intellectual Property

- Always read the EULA (End User License Agreement) and Intellectual Property Rights of the website you are getting the assets from.
- Creative Commons gives the main guidelines when using other peoples' work: <https://creativecommons.org/about/cclicenses/>
- You can find pages explaining the EULA:

<https://help.cgtrader.com/hc/en-us>

<https://sketchfab.com/search?type=models>

<https://kitbash3d.com/pages/cargo-eula>

<https://help.artstation.com/hc/en-us/articles/360056658232-General-FAQ-For-Buyers>

6- Intellectual Property

General Terms to be safe (but not limited to):

- If using an asset with no or little modification, credit the creator. Most assets can be modified but not all. If in doubt, read the Licenses and Copyrights and/or contact the creator.
- Don't re-sell, don't redistribute your assets. Keep it for your own use.
- Use assets for your personal educational reason and state what is your own creation and what is not. How? In your showreel credits and in interviews.
- Use as part of a rendered image and don't make the asset accessible to public. Don't share your files or projects.
- Only use it within the educational environment of the university (don't use it for commercial reasons, for example in a paid project).

*Assets from Quixel Library are ok

TASK 2

- 1- Familiarise with Quixel Bridge**
- 2- Explore other libraries and marketplace**

8- In-Class Project

ual:

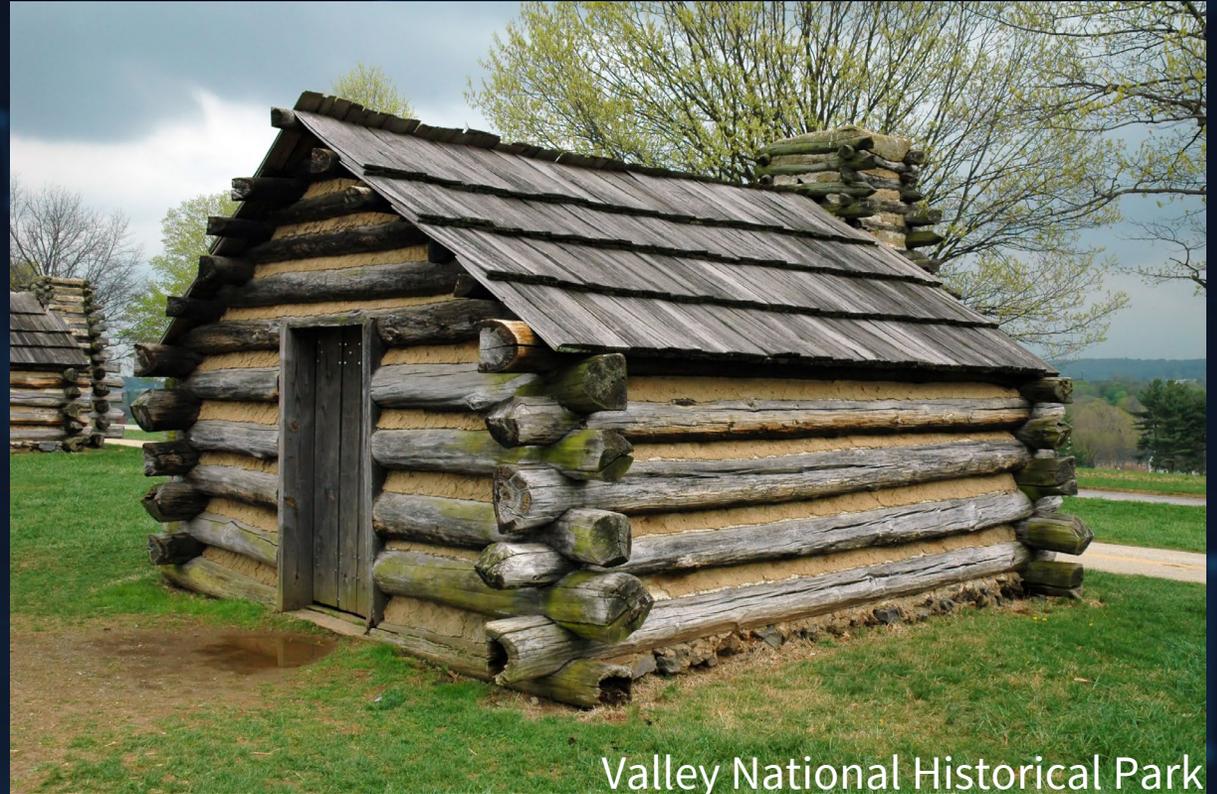
Log Cabin in the Woods



8- In-Class Project

1- Gather References

ual:

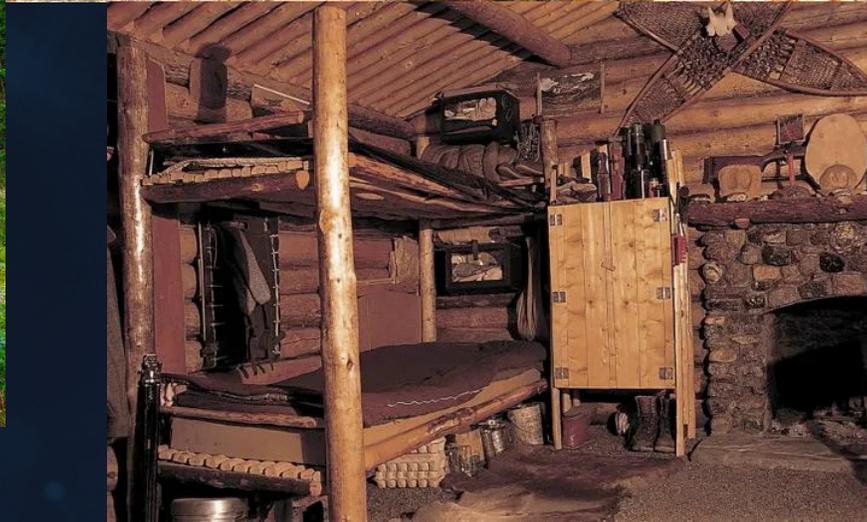
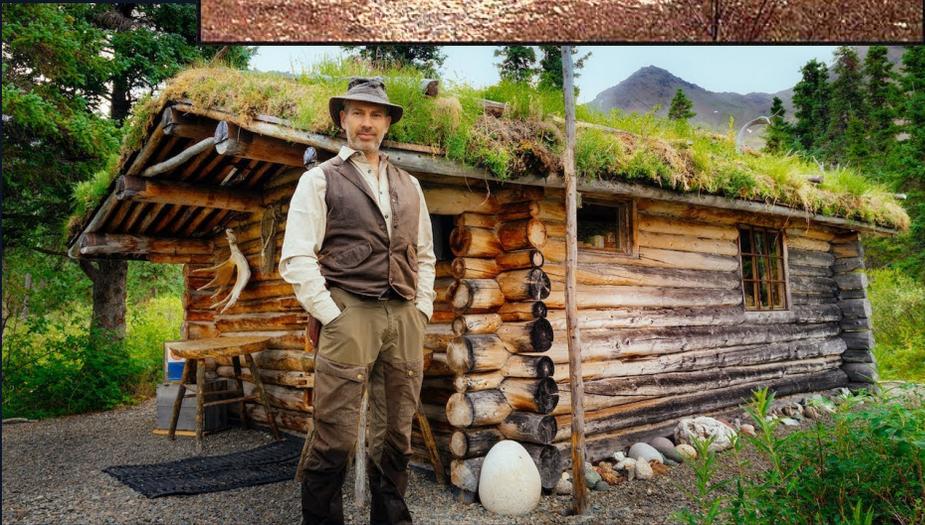
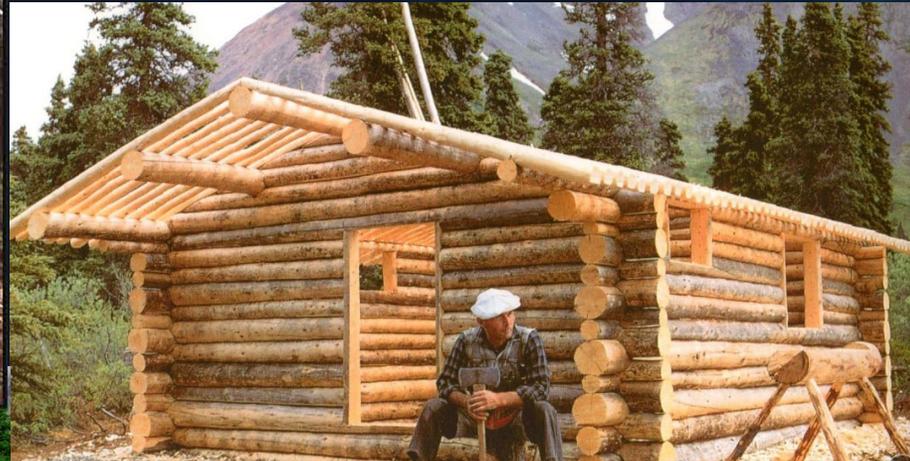


Valley National Historical Park

8- In-Class Project

ual:

1- Gather References



Documentary: Alone in the Wilderness

8- In-Class Project

2- Mood Board

ual:



8- In-Class Project

ual:

Links for references:

Unsplash

Pexels

<https://unsplash.com/photos/FKJgBUDoVC0>

<https://unsplash.com/photos/Fx-vbdsezWs>

<https://unsplash.com/photos/4S6FmLPEP6A>

<https://unsplash.com/photos/5K9Jj5Cm4bQ>

<https://unsplash.com/photos/Y3vv3dZ9zKU>

<https://unsplash.com/photos/65qc9pfDSyY>

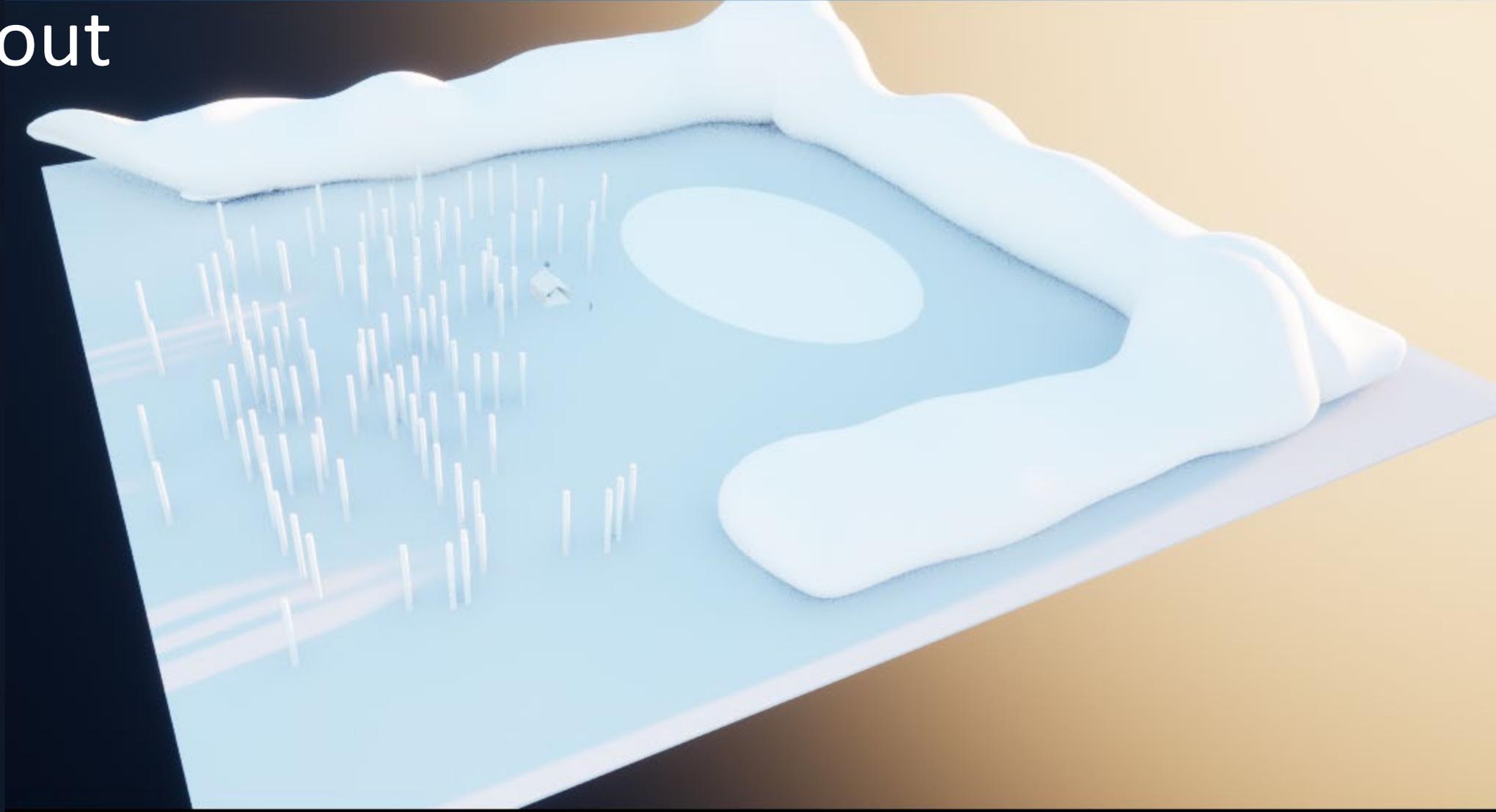
<https://unsplash.com/photos/q2ibJ3yCAb0>

<https://unsplash.com/photos/9oCmXZS7qpo>

8- In-Class Project

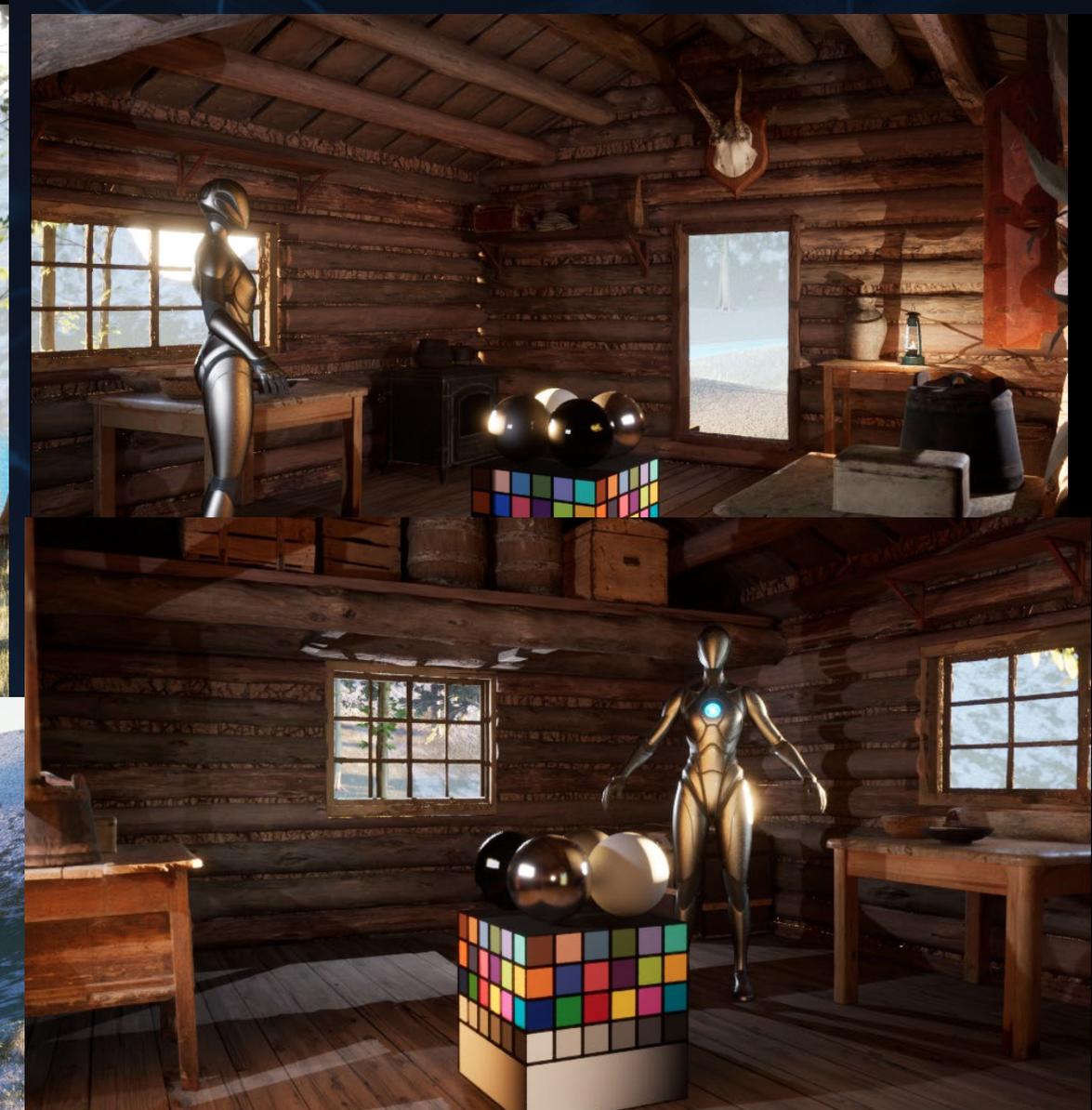
ual:

3- Blockout



8- In-Class Project

ual:



8- In-Class Project

ual:

Real-time Project Setup using Lumen and Nanite:

- Occlusion Culling = turn off
- Enable Virtual Texture Support = on
- Global Illumination = Lumen
- Reflection Method = Lumen
- Software Raytracing Mode = Global Tracing
- Use hardware Raytracing when available = on
- Support Hardware Raytracing = on
- Allow Static Lighting = off (if you don't have any baked lighting)
- Separate Translucency = off (better Depth of Field)
- SM6 = on
- Default RHI = DirectX12
- Shadow Map Method = Virtual Shadow Maps (Beta)

Unreal Engine Learning Links

ual:

[Create photoreal cinematics in UE4: Rebirth tutorial](#)

[Learning Library](#)

[Unreal Engine Documentation](#)