

# Mordhau level creation guide (WIP)

***Most of this guide was included into the uSDK guide. If you're new to Modding in Mordhau, you should start with the uSDK guide***

***[https://docs.google.com/document/d/17A2JBqtCwxjYIFttLe\\_2HU1UJaeaEv4-7veKMXgWOtg/edit#](https://docs.google.com/document/d/17A2JBqtCwxjYIFttLe_2HU1UJaeaEv4-7veKMXgWOtg/edit#)***

***Use this guide only if you're still using the PlouMasterContent pack.***

<b>1. Introduction</b>	<b>2</b>
<b>2. How to start your project</b>	<b>3</b>
2.1 Creating the project in UE4	3
2.2 Before starting: unsolved restrictions	3
2.3 Basics to start your map and test it	4
2.4 Tips	7
<b>3. How to make a landscape with material and textures</b>	<b>12</b>
<b>4. Elwebbaro's best practices guide for your project.</b>	<b>13</b>
4.1 General Workflow and Info	13
4.2 Lighting	14
4.3 Post Process	16
4.4 Optimization	17
4.5 Custom Content	18
<b>5. Blueprints &amp; c++ classes</b>	<b>20</b>
5.1 Ammo boxes	20
5.2 Adding multiple random spawns and team spawns	21
<b>6. Adding custom audio to your map.</b>	<b>24</b>
6.1 Music	24
6.2 Walking collision audio	24

Last update : 18th september 2019

Created by the [Mordhau Mapping & Modding community](#).

## 1. Introduction

The objective of this guide is to provide you the tools and knowledge about pre-sdk mapping in Mordhau. It is compiled from the initial guide and the community progress made since the release of our method. Mordhau devs such as Elwebbaro and Geach3d also contributed to this guide.

### Disclaimer :

Keep in mind that this is an unofficial, tricky way of creating maps in Mordhau before the release of the official modding tools. This method is by no means comparable to an official SDK and the results can't be expected to have the same compatibility level.

You should be aware that there is no assurance that your current projects will be fully compatible with the future versions of the game, nor that you will be able to fully import your project in the official SDK when it comes out (even if it would be surprising that the devs take a direction that would make the community work done before SDK not compatible).

### Requirements

- **Unreal Engine 4.20.3.** Make sure to use this precise version. This is extremely vital. ([Download here](#)). *It has to be done through an epic games account.*
- Converted assets from Mordhau (PlouMasterContent.rar in the #starting-packs channel)

The PlouMastercontent.rar include most of Mordhau's assets. This will allow you to have enough materials to work with and create new maps for this game.

You can import more assets in addition to this content. Note that you should not use materials that are copyrighted without the authorisation of their owners.

## 2. How to start your project

### 2.1 Creating the project in UE4

- Create a blank project (without starter content)
- Name and save your project (Name it Mordhau for better compatibility with the community work on blueprints). Make sure to place your unreal project as close as possible to the disk root or you may run into a *"Couldn't save package, filename is too long"* when you'll cook your maps.

#### Then quit Unreal 4.

- Go to the location where your Unreal project is, then extract the compressed files of PlouMasterContent.rar into the "content" folder. Make sure to not create a Content/Content folder but instead merge them.
- Launch your project again, then look at your content browser : you should see the exact same folder structure as Mordhau's one. Within some of them static meshes with applied materials ready to be dropped into your map. It is normal to see some empty folders.

### 2.2 Before starting: unsolved restrictions

- Even if you have a NavMeshBoundsVolume in your map, bots will ignore it and won't be able to move. **Essentially as of now, bots will not work no matter what you do.** Don't be alarmed. You can spawn bots in your map, but they will not move.
- The Official spawning system used in Mordhau is not accessible to us. There are workarounds provided by the community (see Community Blueprints below) for multiple spawns and team spawns. You can use the default PlayerStart actor from UE, but it will allow only one unique spawn in your map even if you place several of them, which is enough to test your map in the first place.

## 2.3 Basics to start your map and test it

It is important to set a proper work environment before starting.

Your level must be in a "**project name\Content\Mordhau\Maps\Mapname\**" folder.  
**(important)**

### - Using custom assets

Any additional custom assets (IE assets that are not in our pack) will have to be placed in an additional folder in this *Mapname\* folder.

When you download a pack from the UE store, it will be placed in the Content folder : for convenience you can move the whole pack to your custom asset folder. You don't have to move/copy assets from our Mordhau pack unless you modify them.

The name of this folder has no importance, just make sure to move all your custom assets into it using UE move function. Never move them using windows file browser since UE has to manage the changes itself.

For instance here is how your folders should look like for a map named "Waterfall". Here I simply named my custom assets folder "assets" :

The image shows two screenshots of a Windows File Explorer window. The top screenshot shows the path: Disque local (C:) > Unreal projects > Mordhau > Content > Mordhau > Maps > Waterfall. The bottom screenshot shows the path: Disque local (C:) > Unreal projects > Mordhau > Content > Mordhau > Maps > Waterfall > assets.

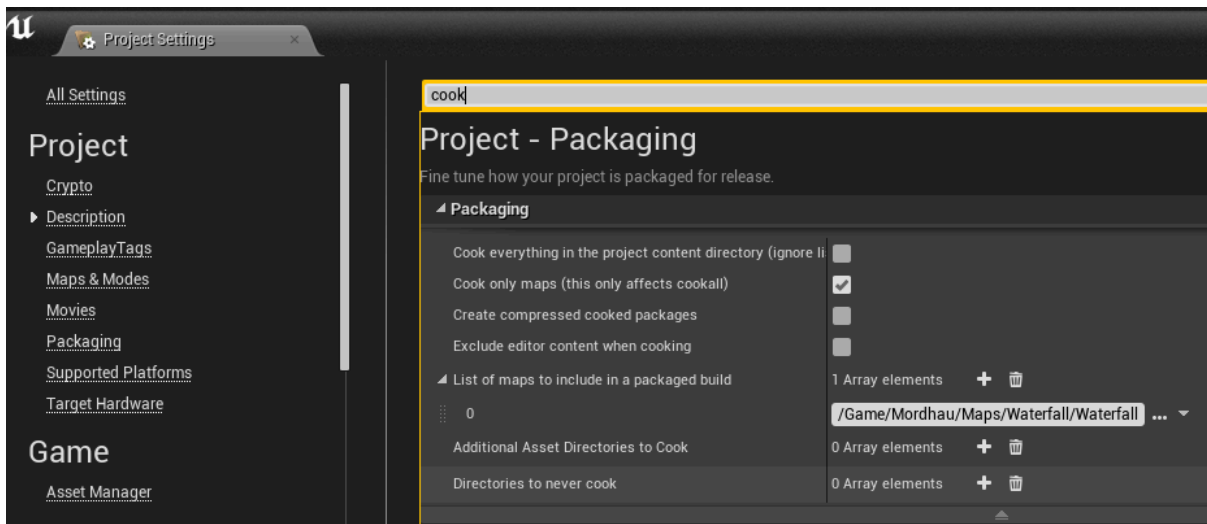
Nom	Modifié le	Type	Taille
assets	18/08/2019 20:07	Dossier de fichiers	
Waterfall.umap	19/08/2019 02:18	Fichier UMAP	3 323 Ko
Waterfall_BuiltData.uasset	19/08/2019 02:18	Fichier UASSET	1 108 Ko

Nom	Modifié le	Type	Taille
DesertOasis	17/08/2019 10:48	Dossier de fichiers	
Environment_Set	17/08/2019 08:02	Dossier de fichiers	
FootstepSound	17/08/2019 14:32	Dossier de fichiers	
ForestCollection	16/08/2019 11:39	Dossier de fichiers	
Landscape	17/08/2019 14:07	Dossier de fichiers	
NecrosUtilityMatPack	17/08/2019 11:25	Dossier de fichiers	

Our next step is to make sure that you will only cook your map and the assets that it uses and not all the assets + the showcase maps that are in those custom packs.

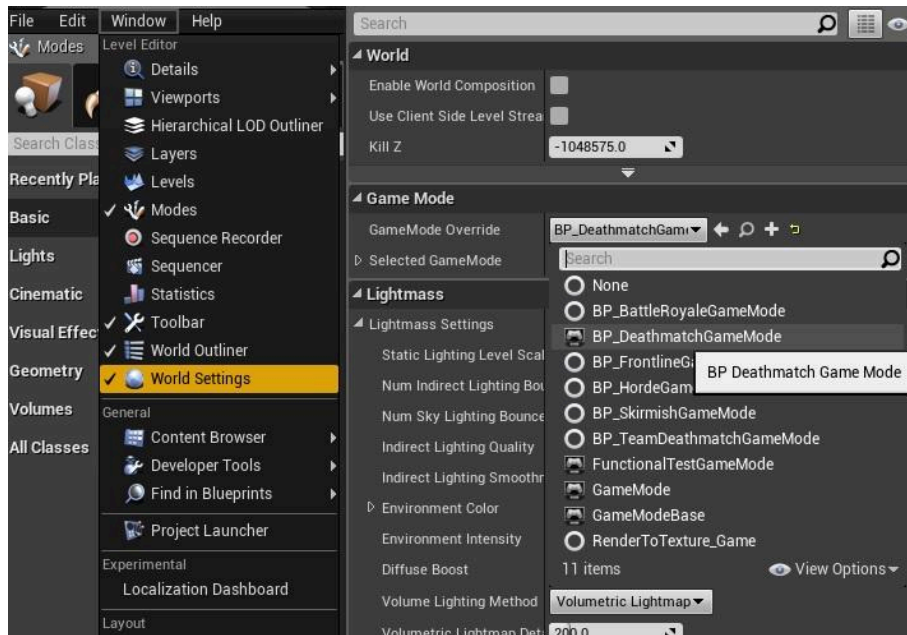
- Open the Project Settings and search for “cook”.
- Make sure that “Cook only maps” is enabled.
- In “List of maps to include in a packaged build”, search for your level and select it.



We highly recommend to create a simple test map first, drop in some of the meshes from our pack + eventual custom assets of your choice and make sure everything works fine before really starting your project.

### - **Setting a game mode**

To be able to spawn in your map, you need to set a proper game mode in Unreal 4. Go in your World Settings and select for instance BP\_DeathmatchGameMod as GameMod Override:



For Game State Class select BP\_DeathmatchGameState. The others like HUD class, Player Controller class, Player State Class and spectator don't matter, you can leave them as default since the game will assign them automatically when your map will start.

#### - **Setting spawns**

The Mordhau devs are using a custom iteration of spawn locations and we don't have access to it. This means that when you use the playerstart from UE4, even if you place several of them, only one will work and players will all spawn at this location. This can still allow you to test your map. To include several spawn locations and team spawns, see the *community blueprints* category.

#### - **Cooking files**

Always click on "Save all" before you cook your level. For a quick test, you may not want to build your lights (a process that can take some time).

#### **Manually :**

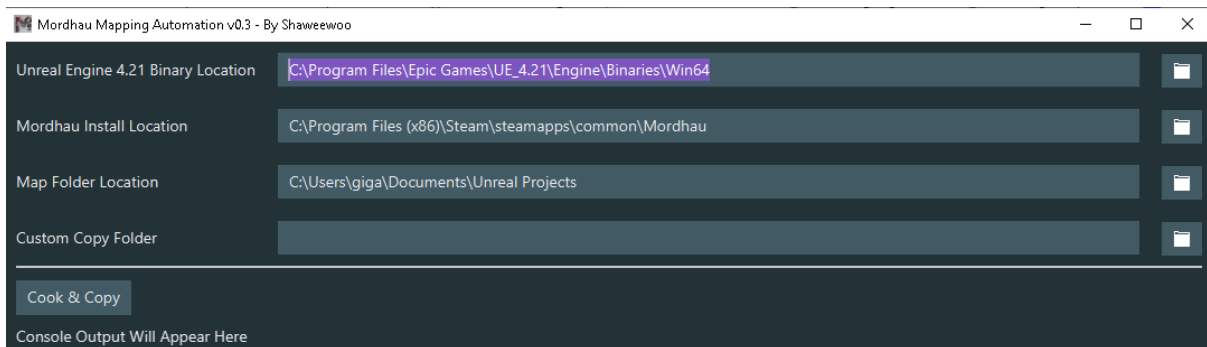
Go in the file menu and select "Cook Content for Windows".

Go in "project name\Saved\Cooked\WindowsNoEditor\Mordhau\Content\Mordhau\Maps\". You will find the cooked files of your map here in a *mapname* folder .

Copy this folder to the maps folder of Mordhau  
 (Steam\steamapps\common\Mordhau\Mordhau\Content\Mordhau\Maps)  
 You don't need to copy the other folders that have been cooked.

## Auto cooking app made by Quinn :

Will cook your files and place them automatically in the Maps folder of Mordhau



Download : <https://github.com/Shaweewoo/mordhau-mapping-automation/releases> (click on "Assets" then mordhau-mapping-automation.exe)

### - Testing your map

Launch Mordhau, open the console and type "open *mapname*"

### - Setting a server

Upload your cooked files in the maps folder of your server (same path as the local Mordhau folder), you should have a file manager in the main menu on your provider's interface, or you can also use FileZilla to access via FTP and drop the files in the maps folder directly.

Restart your server, log in as a admin, open the console and type `changelevel mapname`

### - Share your files with friends/community

Only players who also have your map files in local will be able to join your server when your custom map is up. Auto-download from servers is not yet a thing and players will have to download the maps in a separate location in order to play on the server. Map versions aren't checked when joining a server so if a user as a different version he could run the map but not see the additions made.

**To share your project with more people in order to test it and have feedback, or submit it for the Community Map Installer, ask for the Mapper role to Admins or Moderators.**

## 2.4 Tips

### - New to UE4 ?

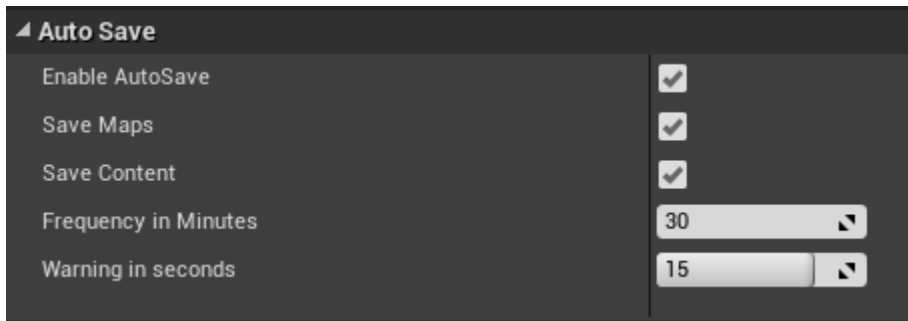
Start by reading the basics guides : <https://docs.unrealengine.com/en-US/index.html>

You can also find tutorial series on youtube (research “unreal beginner tutorial”)

### - **Backup your work**

Make sure to create backups of your level by saving it under different names in the same folder as your main level. Do so for every important step/modifications of your project.

There’s an autosave feature in UE, that you can configure in the Editor Preferences. (search for “auto save”). I’d recommend to set the frequency to a larger number than the default one, since you may take time to realise some mistakes you’ve done and it will avoid the autosaves to be stacked across the last hour.



You can find the autosaves in Saved\Autosaves\Game\Mordhau\Maps\

### - **Submitting a map for the map installer**

There is no other requirements for your map to be in the Map Installer than being approved by the community through a vote.

Still, I’d recommend you to build your project around the idea that your map could be played for years in its actual state. The first impression being primordial, submitting a rushed, non-polished map could be detrimental for its reputation. Take time on your project, there is nothing better than playing on a well made map.

It takes a lot of work to polish a map in every aspect. Obviously the more time you invest in it, the better your map will be.

I’d also suggest to submit a map including all game modes available (TDM, SKM, DM, ...) unless it is specifically designed for only one of them.

It is important that you follow the format used by the community when sharing your map.

More info here :

<https://docs.google.com/document/d/1jNN3GgzNbPUh5gQnRdBHnaASz788wL9tlj63FMH5Z6M/edit>

## - **Map files size**

For convenience to players, server owners, and the Map Installer, your cooked map folder shouldn't be too heavy in terms of files size. For instance the size of the actual maps in the installer does not exceed 50Mb (zipped) for the large majority of them.

When you use Mordhau assets the cost for the file size is reduced since these assets aren't in your custom folder and won't be cooked.

When using custom assets, the more you use in your level, the more your cooked map folder size will increase.

Keep an eye on that when creating your level since it will help for your map to be shared and played easily. A server owner might not upload your map to his server if he has to upload a 2GB folder for instance. When auto-download from servers will be implemented, players without a good connection probably won't wait for it to be downloaded either.

You can detect which assets are using the most space by browsing your cooked folder and for instance replace them with a Mordhau one in your level.

## - **Level design**

Before starting your level you should have a precise idea of its purpose and the general level design that will fit it.

In a TDM/SKM context, Your level should be balanced for both teams, for instance reaching a crucial place from the spawns should be done in an equal time for both. One team should not have a big advantage such as having a perfect archery/ballista position if the enemies have no counter for it. Having places to take cover and paths to move forward to main areas should be balanced for both teams. For an SKM level, think about what strategy could be used to exploit your level design and make the round unfair and/or boring (one way entrance places that can be camped for instance)

The spawns should be protected from being easily spammed with projectiles or firepots.

For a Duel/FFA level, players should not have to run too long from their spawn to get action and spawns should be equally distributed around the level.

Be careful about places that players could exploit to troll/ruin the match. Setting player collisions to avoid players getting out of bounds is important. There will always be a way to "explore" your level using the toolbox, but you can reduce this possibility.

## - **Optimization**

Optimization is a very important aspect for your level. it allows players with low configs to have a good experience playing your map. If you want your map to be played in a

competitive context, it must allow max FPS to everyone since it's something primordial for comp players.

There's a lot of things that you can do to optimize your map, this topic is expanded in this guide.

- **Lighting**

Another primordial aspect of your map that you should take care of and invest time on.

For a start you should read the tutorials from UE :

<https://docs.unrealengine.com/en-US/Engine/Rendering/LightingAndShadows/index.html>

- **Finding Mordhau assets names while in-game**

- If you want to find the name of a specific Mordhau asset in game type the following in console when playing a map :

- 1) ShowDebugForReticleTargetToggle
- 2) ShowDebug Physics

When you look at an asset, the name will appear in the top left. It will allow you to find it easily in UE.

## - UE4 Shortcuts/Hotkeys

### Unreal Engine 4 Hotkeys

#### Viewport Navigation

Movement (Standard)	MBs + Drag
Movement (Game-style)	RMB + WASDEQ
Movement (Maya)	Alt + MBs+ Drag
Zoom	Scroll-wheel

#### System-wide

Find in Content Browser	Ctrl + B
Open Console	`
Open Documentation	F1

#### Transformation Selection

Translate	W
Rotate	E
Scale	R
Next Transform	Space Bar

#### Common Commands

Copy	Ctrl + C
Cut	Ctrl + X
Paste	Ctrl + V
Duplicate	Ctrl + W
Delete	Delete
Undo	Ctrl + Z
Redo	Ctrl + Y
Select All	Ctrl + A
Rename	F2

#### Viewport Commands

Lit View Mode	Alt + 4
Unlit View Mode	Alt + 3
Wireframe View Mode	Alt + 2
Focus on Selection	F
Grid Size Decrement	Shift + [
Grid Size Increment	Shift + ]
Perspective View	Alt + G
Front View	Alt + H
Side View	Alt + K
Top View	Alt + J

#### Modes

Placement	Shift + 1
Paint	Shift + 2
Landscape	Shift + 3
Foliage	Shift + 4
Geometry Editing	Shift + 5

#### Play World

Play	Alt + P
Poses or Eject Player	F8
Pause	Pause

MB = Mouse Buttons  
LMB = Left Mouse Button

#### Level Editor

Select All Adjacent Floor Surfaces	Shift + U
Select All Adjacent Slant Surfaces	Shift + Y
Select All Adjacent Surfaces	Shift + J
Select All Adjacent Wall Surfaces	Shift + W
Select All Coplanar Surfaces	Shift + C
Snap Bounds to Floor	Shift + End
Snap Origin to Grid	Ctrl + End
Snap Pivot to Floor	Alt + End
Snap to Floor	End
Find in Blueprint	Ctrl + K
Game View	G
Full Screen	Shift + F11
Immersive Mode	F11
Jump to Bookmark	0-9
Set Bookmark	Ctrl + 0-9
Show Navigaton Mesh	P
Save All	Ctrl + S
Measure Tool (Orthographic Only)	MMB + Drag
Context Menu	RMB
Duplicate & Transform	Alt + Transform
Hide Selected Object	H
Unhide All Hidden Objects	Ctrl + H

MMB = Middle Mouse Button  
RMB = Right Mouse Button

### **3. How to make a landscape with material and textures**

<https://www.youtube.com/watch?v=T984hgQiCiQ>

## 4. Elwebbaro's best practices guide for your project.

### 'Best' Practices

In quotes because a lot of the settings and workflows are based on your actual scene, this is just a collection of style decisions, observations and common pitfalls. So think of this as a rough guide to achieve results that run well, look decent and somewhat match the style of existing Mordhau maps, if that is your goal.

When I find the time I will add screenshots and examples to this to make it easier to understand.

### 4.1 General Workflow and Info

Check out the Unreal Engine documentation for starters, because a lot of resources out there, especially forums and youtube videos teach bad practices.

Study the maps of Mordhau and other games to get a good idea of what limitations and tricks there are.

Use Static Meshes with Static mobility and set up your lightmaps, don't be lazy. Unreal does most of the work for you anyway.

We do not use Distance Fields due to the higher memory footprint

Aim for at around 1500 draw calls before HLOD, lower is always better

Landscapes are expensive, try to minimize component size and their resolution as much as you can. Try to only cover as little area as possible with it and use proxy meshes to fill up the backdrop.

Build your level in a modular fashion, but merge and/or instance your Meshes when you're done. Unreal does not instance Static Meshes by itself. More on that further down

You can use BSP to whitebox your level, but try to replace it with meshes eventually

#### Level structure and Gamemodes:

Mordhau derives the gamemode from the level name, the way our levels are constructed are as follows:

Your Persistent level at the top is your gamemode level which only holds gamemode specific actors (spawns, bounds)

The Level that holds your actual level geometry is a sub-level of that, set to be always active. This allows for an easy way to add multiple gamemodes to the base map without having to worry about parity since they all reference the same base.

NOTE: When building lights/HLOD etc. make sure you are on the base level and not a gamemode level, and watch out which level is active when you place new actors to avoid placing them on the gamemode level.

## 4.2 Lighting

### General Tips:

Less is more. Start with only a directional light and a skylight.

Start on a clean slate, remove or disable your Post Process volumes and Fog actors.

When adding additional lights, use static mobility whenever you can.

Work in a set exposure when you start to light your scene. This will give you a better idea of the values and contrast. Find good median settings so the level is lit and well exposed in most areas and then move on to tweaking the Auto Exposure.

Check the Lighting Only and Detail Light viewmodes regularly to get a better idea of the values of your level or debug your light build for artifacts.

Keep your Bake quality at Preview or Medium for quick iteration times

### **-World Settings-**

#### Lightmass during Iteration:

- Static Lighting Level Scale: 1.0
- Indirect Lighting Bounces: 10
- Skylight Indirect Lighting Bounces: 5-10
- Indirect Lighting Quality: 1
- Indirect Lighting Smoothness: 1
- NOTE: These Settings are fairly high quality and still quick to bake. You can get great results on Production quality builds, as well as quick bakes on Preview.

#### Lightmass for final bake:

- Static Lighting Level Scale: 0.1
- Indirect Lighting Bounces: 20
- Skylight Indirect Lighting Bounces: 10
- Indirect Lighting Quality: 1.5
- Indirect Lighting Smoothness: 0.75
- NOTE: These settings are absolutely overkill and really only for final production bakes. On mid to high end CPUs these bakes can finish roughly over night. I'd recommend using these settings if you have a level with lots of interior spaces that needs a little bit more bounce light. Otherwise you can just use the settings above instead and won't notice much of a difference.

#### Ambient Occlusion:

- Not always needed because a high quality bake will have indirect shadows accurate enough to do the same thing, helpful if you're still iterating or need a little more contrast
- Direct Illumination Occlusion: 0.5-1 Higher values give more AO even in direct light as an artistic choice, can help to make the scene a little grittier
- Rest of the settings usually on default

### **-Directional Light-**

Mobility: Stationary (dynamic lighting on moveables/skeletal meshes, baked lighting on static geometry)

#### Light:

- Indirect Lighting Intensity is best left alone in early stages of iteration, but if your level has a lot of switches between exterior and interiors it can be handy to turn up the intensity slightly for increased bounce lighting

#### Light Shafts:

- Use at will and if it fits the scene

#### Cascaded Shadow Maps Settings:

- Dynamic Shadow Distance StationaryLight: 1000
- Num Dynamic Shadow Cascades: 1
- Distribution Exponent: 3
- Transition Fraction: 0
- Distance Fadeout Fraction: 1
- These are crisp Shadows that draw for a very short distance. Since we bake the majority of the scene, we can get away with setting this at a very low distance. Only used on Ultra Shadow Settings, the biggest impact of this setting is that Landscape Grass/Foliage cast nice Shadows which you would otherwise not get

#### Lightmass:

- Light Source Angle: Determines how crisp your baked shadow penumbras will be, Higher values result in a softer look which is useful for overcast scenes with lots of diffuse light.
- Use Area Shadows for Stationary Light: Enabling this flag will bake out shadows with penumbras, which gives them a nicer and softer look, but can lead to more artifacts on very low lightmap densities. Generally recommended though.

### **-Skylight-**

Mobility: Stationary (dynamic lighting on moveables/skeletal meshes, baked lighting on static geometry)

#### Light:

- SLS Captured Scene, this gives you the easiest to manage option, make sure to update the skylight whenever you tweak your skydome
- Intensity: Depends on the scene and your directional light. Higher contrast lighting = strong directional light and weaker skylight (sunny evening), lower contrast lighting = weaker directional light and stronger skylight (overcast noon)
- Indirect Lighting Intensity is best left alone in early stages of iteration, but if your level has a lot of switches between exterior and interiors it can be handy to turn up the intensity slightly for increased bounce lighting
- Lower Hemisphere is Solid Color: Leave this checked on and leave at black, or set to an average of your most prevalent ground color. This mimics the bounce light from the ground back up and mostly affects vertical surfaces. In cases when your foliage turns too dark, you can increase the luminance a bit, or vice versa if your foliage is too washed out and can use some contrast

### 4.3 Post Process

#### Post Process Volumes:

- Use a single unbound volume for the base settings on your entire map
- Use smaller local ones to tweak settings if necessary e.g. increasing exposure in indoor areas or creating a fade to a different mood

#### General Settings on Volumes:

#### Color Grading:

- Temperature & Tint: Adjust this in the latter stages for final tweaks, use light colors and different color grading first to set the general mood
- Global: Tweak these color wheels to your liking to get the intended look, always keep team color readability in mind though
- Shadows/Midtones/Highlights: Localized adjustments depending on the values in a range. NOTE: the Saturation and Offset values of these are overwritten by the game due to some effects (Taking damage, dying, low stamina)
- LUT: You can use an external tool to color grade your scene as well

#### Film:

- Slope/Toe/Shoulder: Vanilla Mordhau levels keep close to the defaults and tweak the Global contrast color wheel on Y instead for consistency

#### Lens:

- Chromatic Aberration: This is overwritten by the game when you take damage
- Bloom: Convolution Bloom is very expensive, so don't use it
- Exposure: We recommend using the Basic Auto Exposure instead of the Histogram based one. This generally gives more predictable results, Set your Min and Max Brightness to cover a nice exposure range that doesn't switch too wildy but keeps clarity in dark areas. Often times it's better to clamp the exposure at e.g. 0.5 - 1.5 to avoid crazy jumps when looking at the ground vs. the sky and causing them to

over/underexpose too much. If your interiors become too underexposed, pop another Volume in there that has a higher baseline exposure. Speed Up is usually left at default and Speed down set to 3.0 to match that

- Image Effects: Vignette and Grain are both overwritten by the game, if you wish to add a stronger default Vignette, do so with a post process material
- Depth of Field: Overwritten by the Armory, the only place in the game that uses Depth of Field currently, you can use it localized for effects like going under water and such

#### Rendering Features:

- Ambient Cubemap: Not needed when using a reflection environment and baked GI, you can achieve the same effects with those at no extra overhead. Only use this if you cannot bake your lights.
- Ambient Occlusion: Vanilla Mordhau maps all have the same settings here: Intensity: 0.6, Radius: 50, Power: 3.0, rest is left default
- Global Illumination: Higher Indirect Lighting Intensity values can be used to even out lighting a bit in darker areas, keep in mind that it also decreases the overall directionality of the light and can cause characters to appear flat
- Motion Blur: Mostly Overwritten by the game Settings. Mordhau levels use, Max: 2.0 Per Object Size: 0.5
- Screen Space Reflections: Mordhau levels use Intensity: 50, Quality: 50, Max Roughness: 0.8

## 4.4 Optimization

#### General:

- Use the stat commands provided by the Engine: stat unit, stat scenerendering, stat initviews etc.
- Avoid long sight lines wherever possible
- Keep an Eye on your Actor count in your World Outliner. Our FL sized maps use around 5k actors
- The Statistic tab is useful to search for problems, especially if you need to get the VRAM usage down a bit
- Use the Shader Complexity view to look for expensive material instructions and overdraw. Lots of overlapping foliage, fast emitting particles, and decals can quickly ramp up the cost in a small area.
- Most optimization tools and workflows are highly dependant on your scene, there is no one size fits all solution, so keep that in mind. You can easily make things worse if you are not careful

#### Merging:

- Use the Merge Actors feature to combine lots of small meshes into a single one. Ideally you want to merge meshes with matching materials to keep the slot count of

the resulting mesh low. Merge in smart chunks so you can reuse these meshes elsewhere.

- Don't merge too much into a single mesh. Keep in mind that you will still need a reasonable lightmap resolution for your resulting mesh. The bigger mesh can also not be culled as efficiently either

#### Instancing:

- If you want to reduce draw calls but merging is not applicable e.g. if you need a higher lightmap resolution, use the Instancing tab in the Merge Actors tool instead. This is useful for foliage like Trees and bushes that repeat a lot
- You can only instance the same exact Static Meshes and you cannot use negative scaling
- Instanced meshes are a bit awkward to work with, so only start instancing once you know your meshes placement is somewhat final
- Keep in mind that an instanced group will not be culled if one of the instances is visible, so try to lay out your clusters in a way that you don't have one instance on one end of the map, and another on the other side

#### HLOD:

- HLOD greatly reduces overall draw calls by merging and baking clusters of meshes into a single one. (In case of our levels we have savings from 30-50% roughly) This is the final step before shipping and has to be redone whenever you change the geometry of your map but is a huge tool to get your framerates up.
- We use Simplygon and a global HLOD settings class for all our maps. I will post some image eventually.

## **4.5 Custom Content**

#### Static Meshes:

- If you use vertex normals make sure to export them from your 3D package and set your Unreal import settings to use them as well otherwise you will get a single averaged smoothing group, otherwise there will be mismatches with your baked normal maps or
- Watch your UV channels and remove all the unnecessary ones
- Keep material slots as low as possible
- Use Unreal's built in LOD reduction. Debug your LODs with the Level of Detail Coloration. For most props 3 levels of LODs is a good baseline

#### Lightmap Settings:

- Unreal's Generated Lightmap packer usually does a solid job. It only breaks apart if you use overlapping UVs, create your own Lightmap UVs then.
- Check the Lightmap Density View every once in a while to check for errors.

- Keep the resolution as low as possible. Light blue is a good point to start from, tweak green and red meshes if you can afford to lose the detail.
- If the lightmap UVs of a small mesh are very inefficient, like a long thin pole for example. Keep in mind that you can always merge that pole with other meshes in the future which will even out the distribution.

#### Materials/Textures:

- Create master materials with a certain featureset and derive from them with Instances to increase your iteration speed, keep it reasonable though and don't cram too many features into one huge Master Material
- We use default compressions for Albedo and Normal maps
- Linear textures are most often packed into a single texture to save on Samplers, usually RMA (Roughness, Metalness, AO) or RHAO (Roughness, Height, AO)  
Compression is usually set to Mask



Method by Dead Goat to spawn ammo boxes. Just copy and paste this blueprint into the level blueprint editor and attach it to the, "begin play" event. This will swap all "bp\_ammobox" assets with a working ammo box in game.

<https://blueprintue.com/blueprint/4a559a8i/>

## 5.2 Adding multiple random spawns and team spawns

### Cswic's video guide

<https://www.youtube.com/watch?v=fgxo6bqYvkQ&feature=youtu.be>

#### Steps in text :

Prerequisite: Install Visual Studio

Step 1: Create a new project and title it Mordhau

- It is important that the project is titled Mordhau or your player starts will not be recognized

Step 2: Navigate to file and select create new C++ Class

- This class will be used as a placeholder for your player start

Step 3: Check the box that indicates show all classes and search player start

- The player start class will not appear if you do not check this box

Step 4: Title your player start as "MordhauPlayerStart" and create class

- Make sure the directory says "/Unreal Projects/Mordhau/Source/Mordhau/" when you are creating the class

Step 5: Visual Studio should automatically launch after UE4 finishes compiling the new class

- If Visual Studio does not automatically launch, navigate to "A:\Unreal Projects\Mordhau\Source\Mordhau" and open the MordhauPlayerStart.h file with Visual Studio
- If your project fails to compile the C++ class, please contact the moderation team or ping Battlefield Recapitator; you may have an incompatible version of Visual Studio or need to verify the integrity of your UE4 files

Step 6: In Visual Studio, you will need to replace your code in your MordhauPlayerStart.h file with:

```
#pragma once
```

```
#include "CoreMinimal.h"
```

```
#include "GameFramework/PlayerStart.h"
```

```
#include "MordhauPlayerStart.generated.h"
```

```

UCLASS()
class MORDHAU_API AMordhauPlayerStart : public APlayerStart
{
    GENERATED_BODY()
public:
    UPROPERTY(EditAnywhere, BlueprintReadWrite)
        int Team = 5;
};

```

- It is important to name Team equal to some number, the number can be what you like EXCEPT for 0 or 1, as these are the default team values and will create conflict when you try to use your player start

Step 7: In Visual Studio, with your player start file open, navigate to build and select build source

- This is a crucial step; your player start will appear without doing this, but will be missing the team component

Step 8: If you had not already, restart UE4 and reload your project

- UE4 will not be able to use the new player start until you finish this step

Step 9: In the content browser, click on the folder icon next to Content and look for C++ classes there should be a child folder titled Mordhau that contains your new Player Start class

Step 10: Drag in your player start as you would a normal object

Step 11: Navigate to the details panel and find the category Mordhau Player Start

- There should be a component that reads Team with the number you picked earlier assigned to it

Step 12: Replace the currently defined number with a 0 for red team or a 1 for blue team (0 if it's a FFA level)

- Do not leave the team with the original number as this will likely cause erratic player spawning behavior

Step 13: Place more player starts and assign the team values to each one

- Make sure to check that each one is set, especially if you notice incorrect spawning behavior

Step 14: Compile your code in UE4

- The compile button only appears after you have successfully created your C++ class and is located next to the build option

Step 15: Ensure your game mode is set to team deathmatch in the world settings panel

Step 16: Cook your map and place the map files in their regular location

- You do not have any new file that you have to place in a specific location, it should be handled automatically by the game's engine

Step 17: Load into your map and add a considerable number of bots to test that the spawns never behave unexpectedly

- You should be finished at this point, please contact the moderation team or ping Battlefield Recapitator for assistance if anything should go wrong

Note: As I understand it, you must cook any map with this player start system in the mordhau project. Your map will say that the class is missing (in the server log) if you cook the map in another project.

Credit for the discovery of this method goes to Cswic

## 6. Adding custom audio to your map.

### 6.1 Music

- Squackety's guide to adding custom audio.

**part 1**

→ This PC > SSD (H:) > MyProject > Content > Adamant

Name	Date modified	Type	Size
Adamant.umap	7/26/2019 2:37 PM	UMAP File	2,369 KB
Adamant_BuildData.asset	7/26/2019 2:18 PM	UASSET File	17,181 KB

this is what your map should look like uncooked

→ This PC > SSD (H:) > MyProject > Content > Mordhau > Maps

to add audio, go here and create a folder with the same name as your map. (make sure umap and folder are named the same)

→ This PC > SSD (H:) > MyProject > Content > Mordhau > Maps > Adamant

Name	Date modified	Type	Size
Adamant	7/26/2019 1:48 PM	File folder	

open the folder you made and then make another folder inside of it with any name that you like.

→ This PC > SSD (H:) > MyProject > Content > Mordhau > Maps > Adamant > Adamant

Name	Date modified	Type	Size
Forest and Nature S...	Forest and Nature Sounds...	scrapper900	YouTube
Forest_and_Nature_...			
Wind.asset			
Wind.wav	10 HOURS Desert Wind So...	The Relaxed Guy	YouTube

put your wav files in here and then press import asset ( should appear in the bottom right corner in unreal), then place your sound in your map.

**part 2**

→ This PC > SSD (H:) > MyProject > Saved > Cooked > WindowedEditor > MyProject > Content > Adamant

Name	Date modified	Type	Size
Adamant.umap	7/27/2019 10:05 PM	Windows Media P...	1,387 KB
Adamant.umap	7/27/2019 10:05 PM	UMAP File	306 KB
Adamant_BuildData.asset	7/27/2019 10:05 PM	UASSET File	8 KB
Adamant_BuildData.ubulk	7/27/2019 10:05 PM	UBULK File	18,656 KB
Adamant_BuildData.umap	7/27/2019 10:05 PM	Windows Media P...	2,410 KB

this is what your map will look like after being cooked. notice that your folder, and audio files aren't there

→ This PC > SSD (H:) > MyProject > Saved > Cooked > WindowedEditor > MyProject > Content > Mordhau > Maps

Name	Date modified	Type	Size
Adamant	7/27/2019 10:05 PM	File folder	

navigate to here in your files, and look for the folder that you made.

open it

→ This PC > SSD (H:) > MyProject > Saved > Cooked > WindowedEditor > MyProject > Content > Mordhau > Maps > Adamant

Name	Date modified	Type	Size
Adamant	7/27/2019 10:05 PM	File folder	

move this folder into here

→ This PC > SSD (H:) > MyProject > Saved > Cooked > WindowedEditor > MyProject > Content > Adamant

Name	Date modified	Type	Size
Adamant.umap	7/27/2019 10:05 PM	Windows Media P...	1,387 KB
Adamant.umap	7/27/2019 10:05 PM	UMAP File	306 KB
Adamant_BuildData.asset	7/27/2019 10:05 PM	UASSET File	8 KB
Adamant_BuildData.ubulk	7/27/2019 10:05 PM	UBULK File	18,656 KB
Adamant_BuildData.umap	7/27/2019 10:05 PM	Windows Media P...	2,410 KB

this is how it should look after you are complete.

→ This PC > SSD (H:) > MyProject > Saved > Cooked > WindowedEditor > MyProject > Content > Adamant > Adamant

Name	Date modified	Type	Size
Forest_and_Nature_Sounds.asset	7/27/2019 10:05 PM	UASSET File	1 KB
Forest_and_Nature_Sounds.umap	7/27/2019 10:05 PM	Windows Media P...	852 KB
Wind.asset	7/27/2019 10:05 PM	UASSET File	1 KB
Wind.umap	7/27/2019 10:05 PM	Windows Media P...	1,291 KB

On an additional note, add a dud "MusicSoundClass"(Sound Class) to "UE4Project\Content\Mordhau\Audio\Other" and apply that sound class to the sound cue you placed in the map (Sound Class Override) in the Details Tab. This should let you adjust the music volume and eliminates the bug of turning down master to 0 and then losing the music in game.

### 6.2 Walking collision audio

- Teabag's guide on how to replace the walking collision of a floor audio surface :

<https://cdn.discordapp.com/attachments/603238467274539010/604826062840397864/unknown.png>

### 6.3 How to add triggered audio



## Misc & Oldies

### 5.2 Adding multiple random spawns (Deathmatch)

Taki's blueprint : <https://github.com/WaGi-Coding/TakiRandomSpawnForMordhau>

Dead Goat's blueprint (Simply copy the text and paste into your level blueprint. You will need to follow the very short highlighted instructions inside the blueprint.)

[https://blueprintue.com/blueprint/-3fs\\_81-/](https://blueprintue.com/blueprint/-3fs_81-/)

Battlefield Recapitor's guide to create a multiple player spawns blueprint (if you want to create your own and try stuff):

<https://docs.google.com/document/d/e/2PACX-1vSTH9Vwm4-cOdvchCIY8WBZkVVQoCqTWtGAr2B9G0C8ZYqYicceCVt10ADI5iQIDB-9FLRj8FhPD7hC/pub>

Geach3d advices on lighting :

<https://cdn.discordapp.com/attachments/601197922213953544/602682216425848838/unknown.png>

<https://cdn.discordapp.com/attachments/601197922213953544/602721295716712462/unknown.png>

About Occlusion Culling. Can cause performance loss and improve it on the other side.

Aswell you can compensate some of the Flickering/Light-through issues :

<https://www.youtube.com/watch?v=6MhsdwTW3SQ>