

MMO Starter Kit 1.10 Documentation

For questions not covered in this guide, please use the Q & A website: <http://mmokit.net/>

[unofficial japanese translation by かなた](#)

Table of Contents

[MMO Starter Kit 1.10 Documentation](#)

[Table of Contents](#)

[Credits](#)

[Getting Started](#)

[The editor](#)

[Opening the project](#)

[Setting up the database](#)

[Launching for the first time](#)

[Testing with 2 players in the editor](#)

[Full configuration](#)

[New to UE4?](#)

[Orienting in blueprints](#)

[Using your own map](#)

[Transferring assets from another project](#)

[Customizing the player character](#)

[Mesh and animations](#)

[Setting up the inventory doll](#)

[Materials and highlight](#)

[Character sounds](#)

[Customizing the NPCs](#)

[Creating items, quests and abilities](#)

[Creating your own item type](#)

[Customizing item quality tiers](#)

[Editing multiple assets](#)

[Adding your own persistent stats](#)

[Creating your own draggable windows](#)

[Running the server on your local computer](#)

[Dedicated Server and Packaging](#)

[Server IP and clients](#)

[Removing the players limit](#)

[Setting up dedicated server on a VPS](#)

[Ordering a VPS](#)

[Transferring the server to a VPS](#)

[Launcher/autoupdater](#)

[Removing demo content](#)

[Troubleshooting and FAQ](#)

[FAQ \(not issues\)](#)

[Problems with entering world](#)

[Crashes, game not launching or not compiling](#)

[Other issues](#)

[Debugging PHP](#)

Credits

I would love to thank and give credit to the following:

[VaRest Plugin](#) - Vladimir Alyamkin (Ufna)

[LoadingScreen Plugin](#) - Nick Darnell

The authors of the free assets that are used in the demo:

Sword Girl model (used only on the demo server, not distributed) - [Bunt Games](#)

Inventory icons - [Ravenmore](#), [Flare icons](#)

Icons for abilities and empty slots - [game-icons.net](#)

Inventory sounds - [qubodup](#), [Akeroyc](#)

Healing spell sound - [qubodup](#)

Spell particles, fire spell sounds - official UE4 demos (ContentExamples and FeaturesTour2014)

Female voices - [AderuMoro](#), [MadamVicious](#)

Male voices - [Michel Baradari](#)

Getting Started

The editor

You will need Visual Studio 2015 and a source-built version of UE4. If you don't have the source-built engine yet, follow the official instructions here: <https://www.unrealengine.com/ue4-on-github>

Important: Download the “release” branch since it's the most stable one:

<https://github.com/EpicGames/UnrealEngine/tree/release>

This guide is for UE4 version 4.16

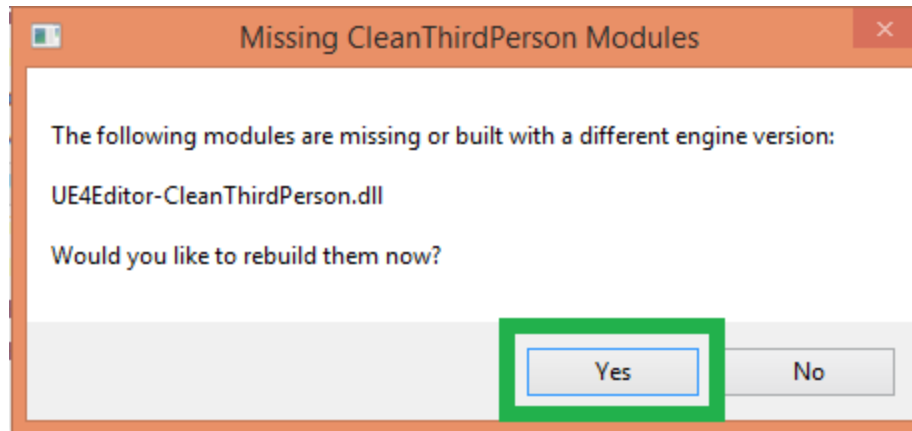
Opening the project

Once you have built the editor, you can open the project by clicking on the blue .uproject file in MMOSTarterKit-1.10/ folder:

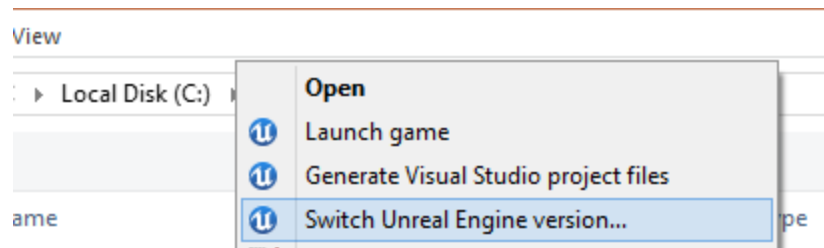
 CleanThirdPerson.uproject

NOTE: Don't place the project on the desktop or inside a deep folder structure, since it has been known to cause build errors. A good path example: D:/MyGame/

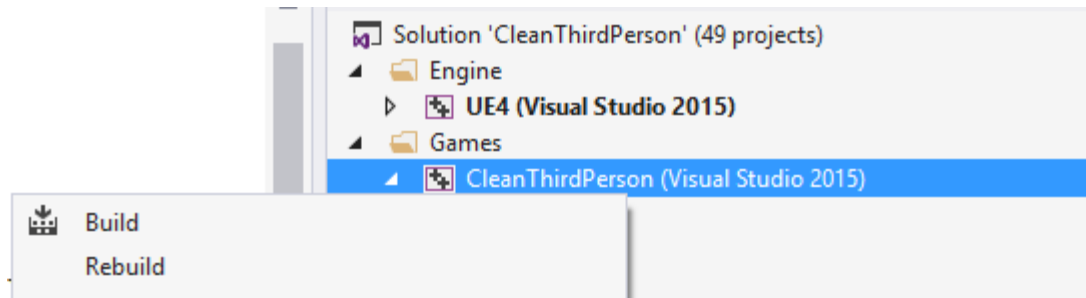
The editor will tell you that the project was built with another version, click "Yes":



If it fails to convert, right-click on the blue .uproject icon in the folder with the project and choose "Switch Unreal Engine version", choose your source-built engine, then right-click again and choose "Generate Visual Studio project files".



Then open ThirdPersonClean.sln in Visual Studio, Right Click on CleanThirdPerson under Games and hit “Build”, then wait a few minutes for a line that will mention that the project has been built successfully:



Setting up the database

You will need a webserver with PHP 5.5+ and mysql, it can be the same server where you host your website. If you already have a hosting provider, you can check its documentation or control panel to see if you have PHP 5.5+, or contact their support.

These hosting providers have been reported to have PHP 5.5+, however this list is by no means exhaustive:

one.com

binero.se (Swedish)

spaceweb.ru (Russian)

Not recommended: godaddy.com - because of the php time limit, this hosting provider doesn't work well with the uploader/launcher that comes with MMO Kit

You **don't** need a paid domain name to work with MMO Kit.

NOTE: PHP 5.5+ is required because of the new encryption methods, if you really want to use an older version of PHP, you can modify the password hashing in mmoregistration.php and mmologin.php. However, it is not recommended.

The following instructions assume you already have a hosting provider:

1. Create a new mysql database and import the dbstructure.sql file to create the structure. Usually this is done via phpMyAdmin.
2. In mmoconnection.php script in the PHP folder, fill your MySQL connection settings:

```
<?php
$servername = 'localhost';
$username = 'your username';
$password = 'your password';
$dbname = 'your db name';
```

NOTE: Most hosting providers use the same server for the database and the php so don't replace 'localhost' unless you are sure it doesn't work with it.

3. Upload the contents of the PHP folder to your web server.
4. In your web browser, go to

<http://yoursite/somefolder/myphpversion.php>

If the reported version is **5.5+**, proceed to the next step.

5. Now test the php scripts by going to

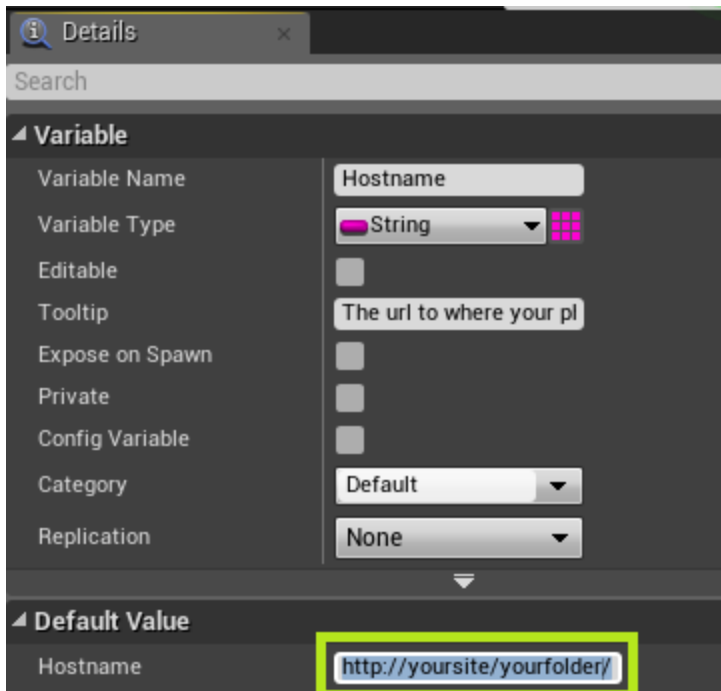
`http://yoursite/somefolder/mmologin.php`

If it says `{"status":"Login information is incorrect. Check your username and password."}` this means the script is working fine, good job!

If it says `Connection failed` or `Access denied`, double-check the database connection info that you filled in step 2

If instead it displays raw php code (with `<?` and such), this means that the php scripts don't execute properly on your webserver.

6. In **MMOInstance** blueprint (located in `Content/MMO/Blueprints/GameModes/`), set the **Hostname** variable's default value to the url path where you've put the scripts, ending with a slash. For example: <http://mysite.com/mmo/>



7. Open hostname.ini file located in Chat\ folder with Notepad or other text editor and place the same url as in p.6 there

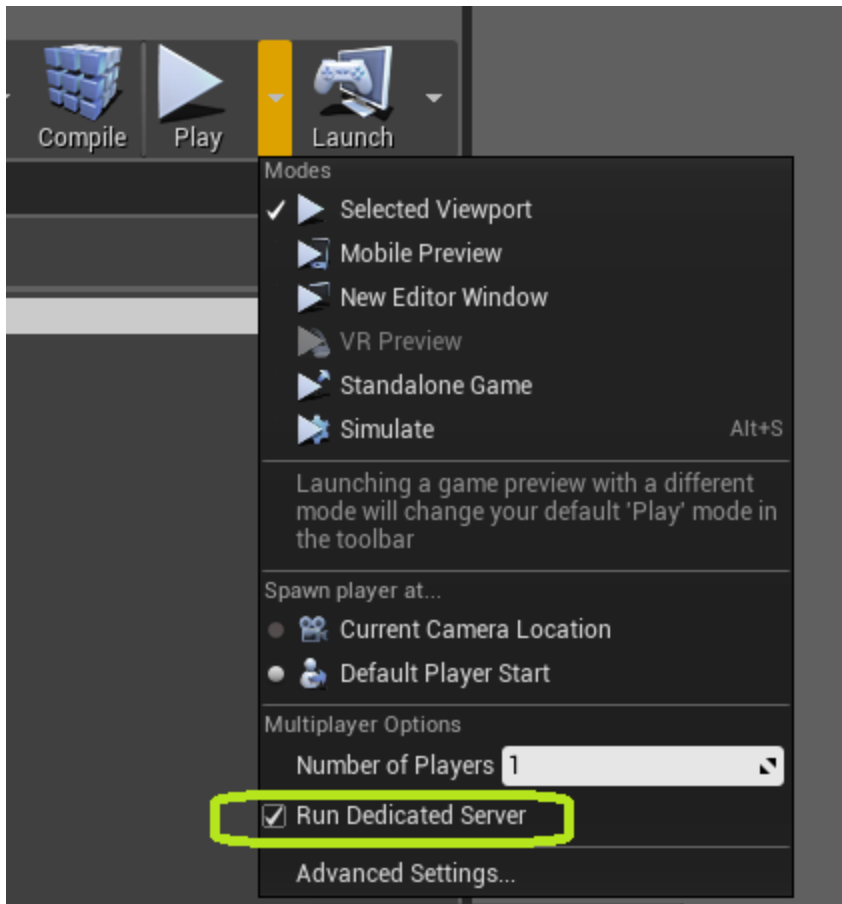
Launching for the first time

1. (optional) Run the chat server (Chat\Server.exe). This is only needed whenever you want to use chat while you test the game. It has to be launched before the game server.
2. Open the **Start** map located in Game\Maps
3. Click Play, then register and create a character. Don't try to enter the world - it's not possible in editor to go from start map to the game map.

NOTE: Don't delete this first character. It will have the "id" 0 in the database and when playing in editor, the game assumes you have logged in as this character with id 0. This allows you to fully test everything in PIE - saving/loading of position and stats, inventory items, etc.

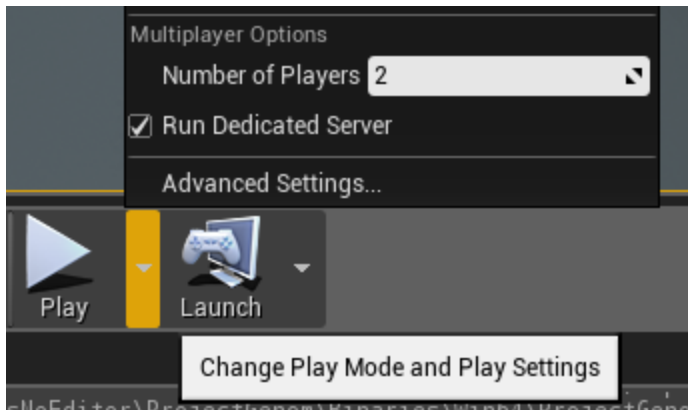
If you do need to delete it, set the "id" of some other character to 0 - it will be the new character that's used when playing in editor.

4. Open **DesertRallyRace** map. In Play options check **Dedicated server** to have a server run in the background. Click Play. You should be able to play as the character with id 0 in your database. Make sure that the name of your character ingame is displayed correctly and is not "CharacterName".



Testing with 2 players in the editor

You can play the game in the editor as 2 players. Make sure that there are characters with ids 0 and 1 in your MySQL database and set the number of players to 2:



Full configuration

If you want to concentrate on customization and gameplay first, you can skip this for now.

Follow the steps below if you want to configure the full MMO Starter Kit demo setup (launcher/patcher, dedicated server on Softlayer or other VPS):

1. Build the [dedicated server](#) and test it locally
2. [Order and set up a virtual server](#)
3. Configure the [patch system](#) and distribute the launcher to your users

New to UE4?

If you are new to UE4, you can follow multiple tutorials that are available on the official website:

<https://docs.unrealengine.com/latest/INT/>

Here are the tutorials that deal with networking and UMG, parts of UE4 that are heavily used in MMO Kit:

Blueprint Networking: <https://www.unrealengine.com/blog/blueprint-networking-tutorials>

UMG (creating user interface): <https://docs.unrealengine.com/latest/INT/Engine/UMG/QuickStart/index.html>

Orienting in blueprints

Here's a quick map of some of the more important blueprints. The color corresponds to the node block color.

character blueprints:

MyCharacter	movement, camera
MMOCharacter	attack calculations
	attack cosmetics on client - animation, sounds, logging
	highlight on mouse over
MMOPlayerCharacter	looting NPCs, picking up items from the ground, dropping items
	health regen
	accepting and turning in quests
	displaying question/exclamation mark above NPC
	saving character information on disconnect
ModularPlayerCharacter	equipping and unequipping modular armor on server and on client

other blueprints:

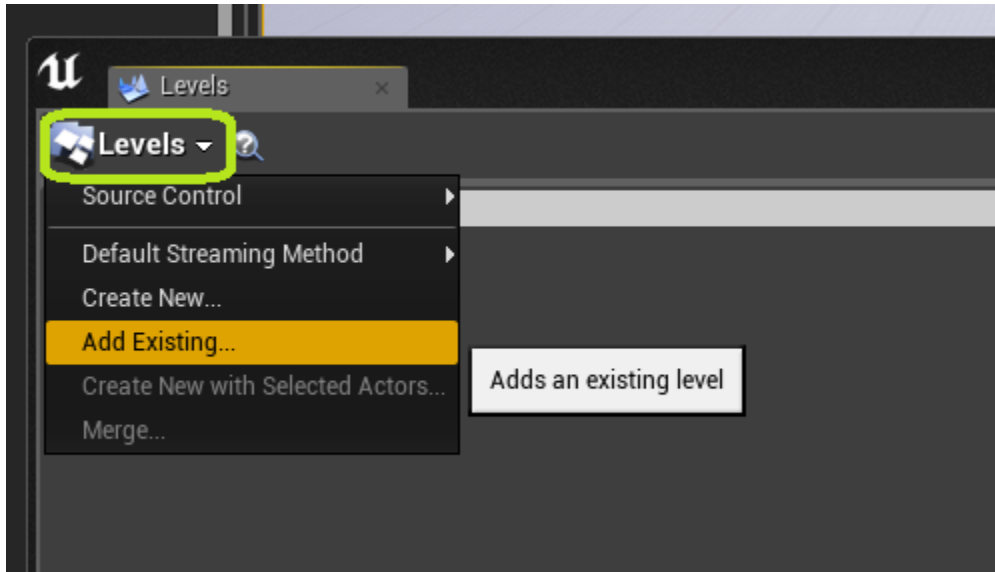
MMOPlayerState	saving and receiving information from MySQL database
MMOPlayerController	opening different game windows (inventory, settings...)
	clicking on characters/ground to set/clear target
	right-clicking on characters to attack/loot
	when a player joins, initiate getting their character data

Using your own map

1. *A. If you want to use the map from another project:* In the project with the map, right-click on the map that you wish to use and choose Migrate, then choose the Content folder of the MMO project. A list of dependencies will appear, click OK.

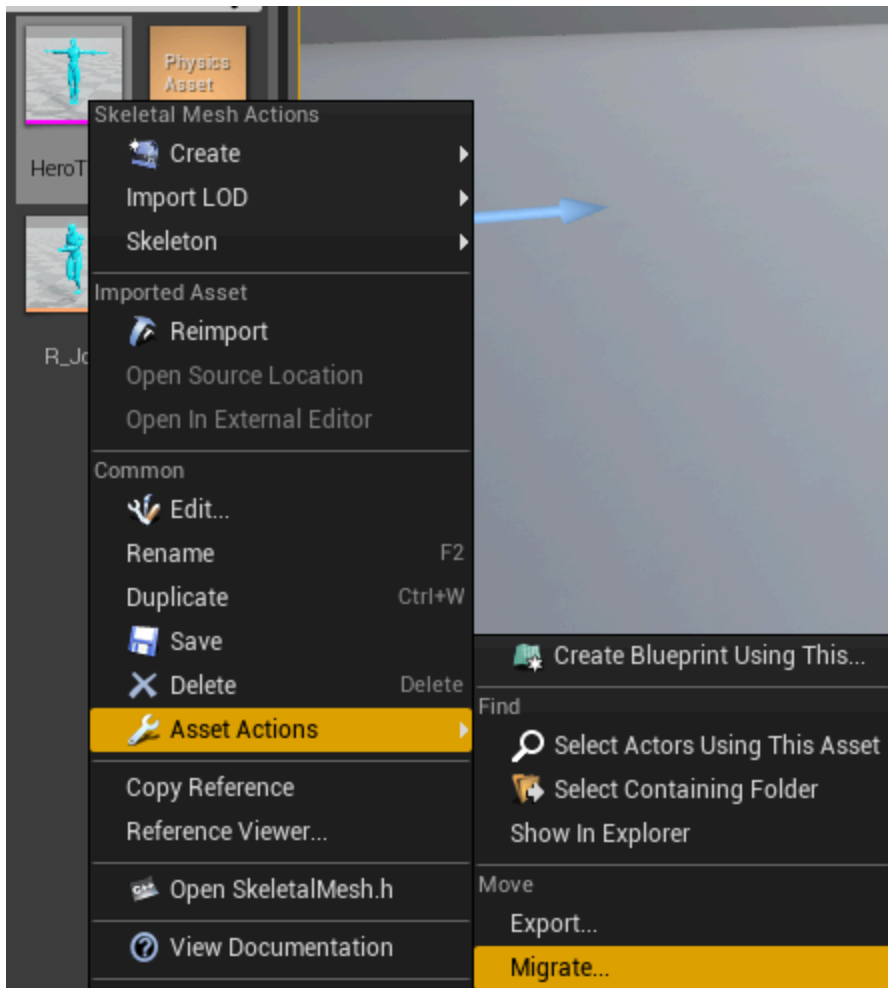
B. If you want to create a map from scratch: Create the map as you would normally.
2. Go to **Settings > Project Settings > Maps & Modes** and change **Editor Startup Map** and **Server Default Map** to your new map. Don't change Game Default Map - the game should still start in the main menu.
3. Make sure that there is a **Player Start** actor somewhere in the map since it will be used for the location where players respawn after death.
4. Go to **Window - Levels**, click **Levels - Add Existing** and choose "DollMap". This is needed for the "character doll" to work correctly in the equipment window.
5. *If this is the map where new characters should start* - in mmocreatecharacter.php set the correct starting coordinates and rotation (yaw).
6. *If there are going to be NPCs with pathfinding on this map* - make sure that you have a **Nav Mesh Bounds Volume** for them to use.

Note: If when you play in editor, your character (the one with id 0) falls through the ground, use /unstuck chat command.



Transferring assets from another project

1. Right-click on the asset that you wish to migrate and choose **Asset Actions > Migrate**.
2. A list of dependencies will appear, click OK, then choose the Content folder of the MMO project.



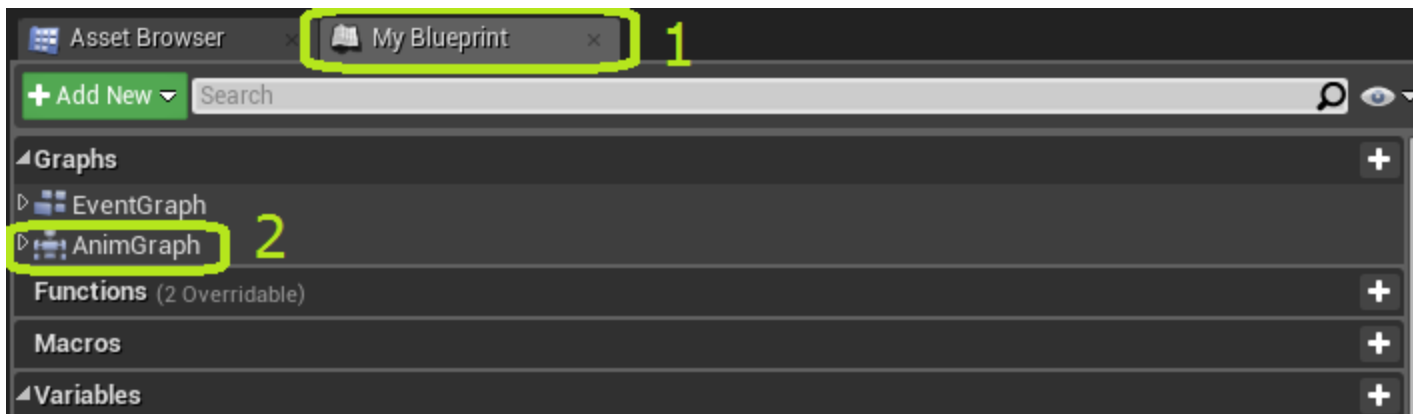
Customizing the player character

Mesh and animations

This guide assumes you have the following animations for your model: idle, walk, jumpstart, jumpend, jumploop, combat idle, attack, and death. If you don't have them all, everything will still work but the t-pose will be displayed in place of the clip that you haven't specified.

NOTE: The animations should be without root motion.

1. Import the skeletal mesh and desired animations, or you could [migrate the assets from another Unreal project](#).
2. Right-click on the **ModularPlayerCharacter** blueprint located in MMO/Blueprints and choose **Create Child Blueprint Class**, name the new blueprint appropriately. You will work with this new blueprint for the rest of the steps.
3. Find the **GuyAnim** animation blueprint (MMO/NewCharacter/). Right-click on it and choose **Retarget Anim Blueprints > Duplicate Anim Blueprints and retarget**, then choose your character's skeleton. You will probably have to uncheck **Show only compatible skeletons**.
4. Find the newly created animation blueprint - it should be in the same folder as your new character's skeleton and it has a **dark orange** line on the bottom of its icon. Open its **Anim Graph** and click on the first node (State Machine). In the state machine open the Idle/Run state, choose the central Blendspace node (it will be named like IdleRun_TPP_Copy) and double-click on it to open the Blendspace.



5. Drag and drop your run, walk and idle animations to the three circles, top to bottom.
6. Repeat the steps 4 and 5 for CombatIdleRun node.
7. Back in state machine, double-click on JumpStart node, click on the Play there and change the animation sequence to your model's jump start animation.
8. Repeat the previous step for JumpEnd, JumpLoop and Dead nodes.
9. When you created the new animation blueprint, several retargeted animations were created in the folder with your character's skeleton, with names ending with "_Copy" and possibly a thumbnail looking black. You can delete them now, just make sure you don't delete anything that has a reference left.
10. Open the character blueprint that you've created, and set the default skeletal mesh and animation blueprint to the new mesh and animation blueprint that



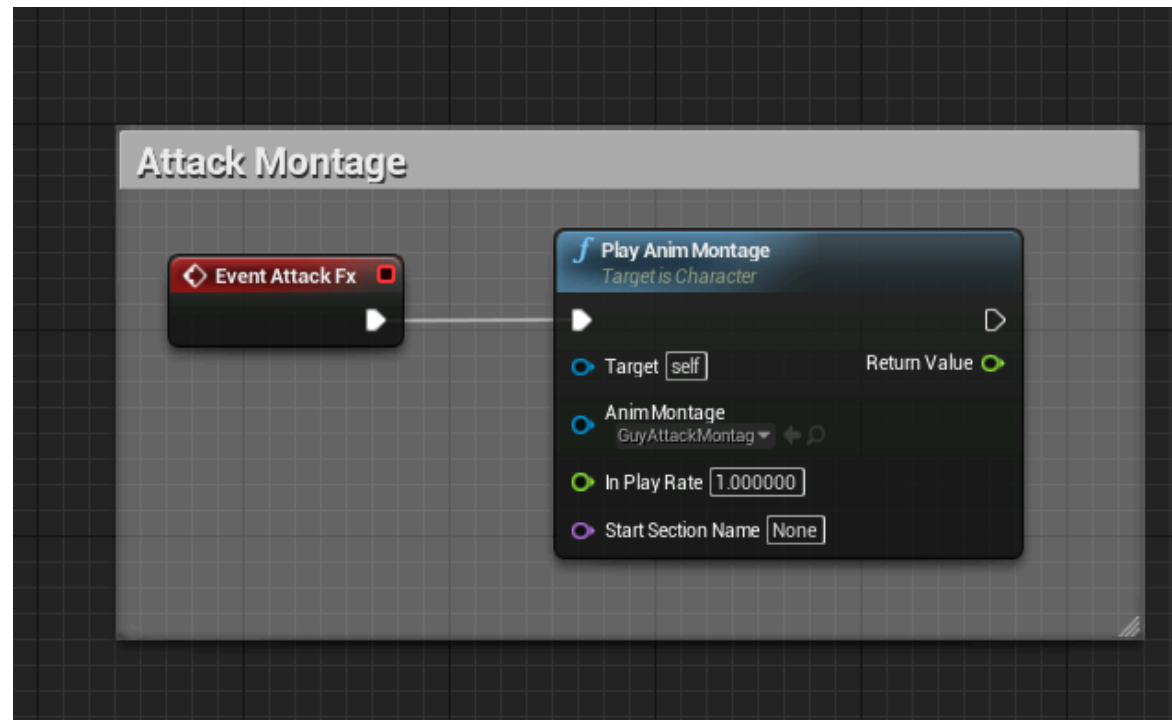
node

you

have created. If the character is modular you will have to set the skeletal mesh of each body part, but you only have to set the animation blueprint for the chest mesh - the other parts will follow it automatically.

11. Right-click on your attack animation, **Create > Create AnimMontage**, save the created montage.
12. In your new character blueprint find this Play Anim Montage node and change the AnimMontage in it to your new montage:
13. Open the GameMode "MMOGame", located in MMO\Blueprints\GameModes, go to Defaults and set **Default Pawn Class** to your new character.

NOTE: make sure your skeletal mesh has a Physics Asset specified. It is required to be able to click on dead characters. You can create a Physics Assets by right-clicking on a Skeletal Mesh and choosing **Create > Physics Asset**.



Setting up the inventory doll

This guide assumes you have changed your player character successfully and now want to change the inventory doll accordingly.

1. Open the **ModularDoll** blueprint (MMO/Blueprints) and change the skeletal mesh for components attached to the Body (Legs, Head, Hands...).
2. In the Event graph of the same blueprint find the node **Set Skeletal Mesh** and set the mesh for the chest there

3. If you've added additional body parts to the ModularPlayerCharacter, plug in the the new body parts to the **Set Master Pose Component** node in the Event graph

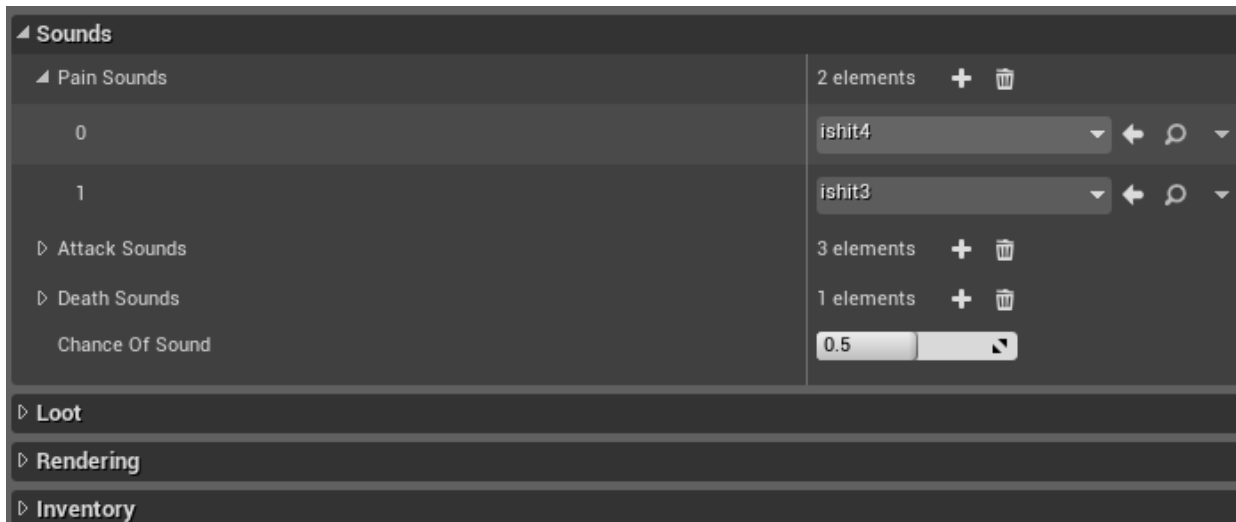
Materials and highlight

If you want the characters be highlighted on mouse hover, the material of your character's mesh needs to have a scalar parameter "Highlight Amount" - the highlight effect is up to you, for example you can add an emissive color. If you would like to know more about this you can check out the [GuyMaterial](#) located in MMO\NewCharacter. Note that player's own character is not highlighted on mouseover, only other players and NPCs are.

Character sounds

1. For character sound navigate to the character blueprint that you've created in [Mesh and animation](#) tutorial
2. In the Defaults tab under the category "Sounds" you can find three arrays: Pain, Attack and Death sounds. Each of them can have as many sounds as you wish - they are chosen at random.

Chance Of Sound variable determines how often the pain and attack sounds will be played (death sound is always played).



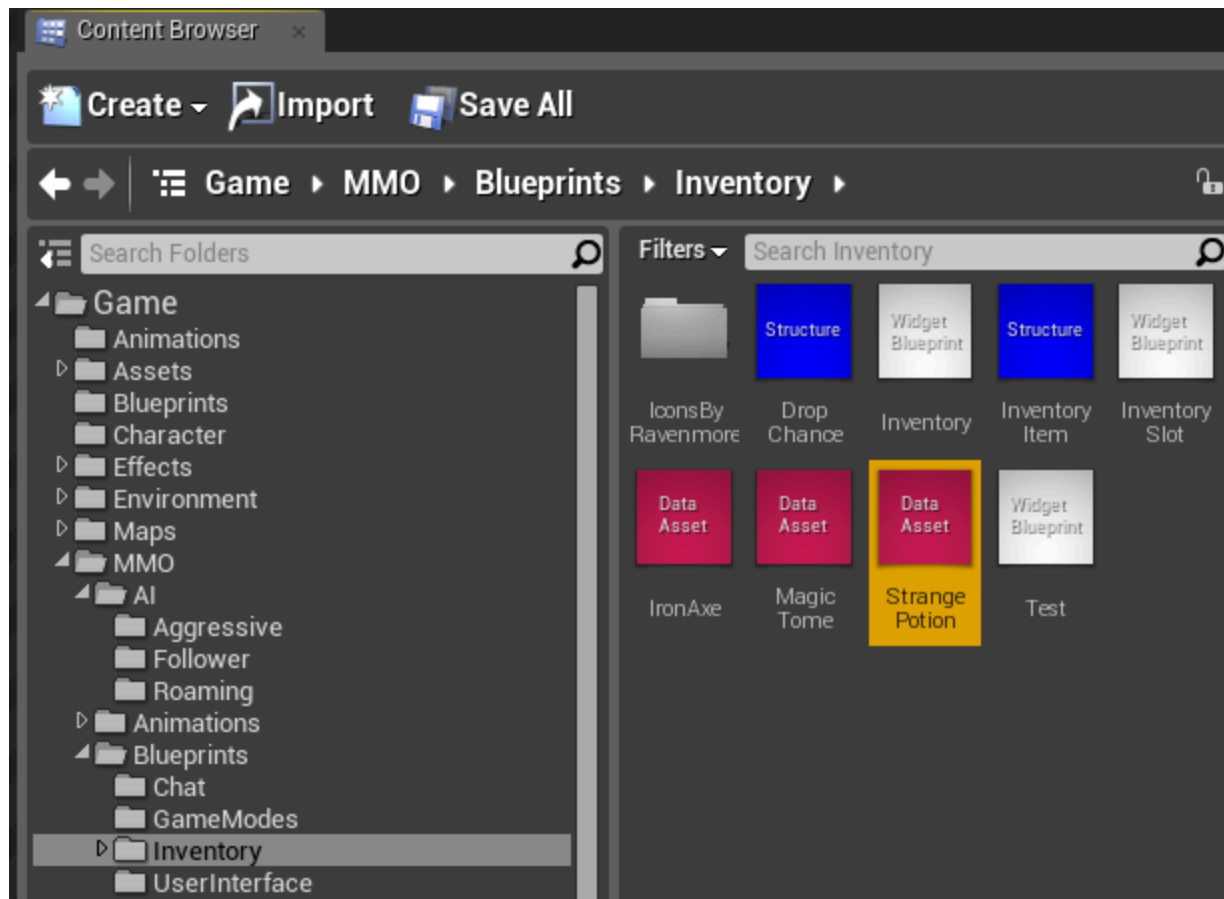
Customizing the NPCs

To customize the NPCs you will need to follow the same steps used earlier when learning how to customize the player character. The only difference will be instead of inheriting from **ModularPlayerCharacter** blueprint you would inherit from **RoamingCharacter** blueprint located in MMO/AI/Roaming.

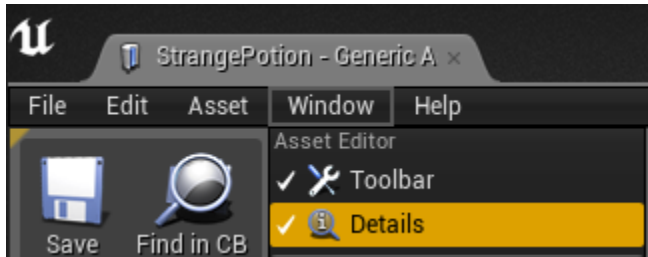
After you've created your new NPC, you can place it on the map and it will appear in game.

Creating items, quests and abilities

1. In MMO/Blueprints/Inventory, MMO/Quests and MMO/Abilities folders you can find some demo items, quests and abilities (**magenta assets**). You can duplicate one of them, or create an asset from scratch by doing the following: **right-click > Miscellaneous > Data Asset > ItemData / Quest / Ability**



2. Open the new asset and customize its details. If you open the asset and its details are not visible, click on **Window** menu and put a check next to **Details**.



Creating your own item type

In this example I will create a new item type called Book that derives from regular item (ItemData) class. It will have one additional field used to store the book's text.

1. Save your project.
2. In the editor, click **File - Add Code to Project**
3. Check **Show all classes**
4. Choose ItemData, click Next
5. Enter your new class name (for example Book), click Create Class
6. Wait for Visual Studio to load. Open your new Book.h file and the ItemData.h file
7. Copy a field from ItemData.h, for example I'm going to copy this FText field):

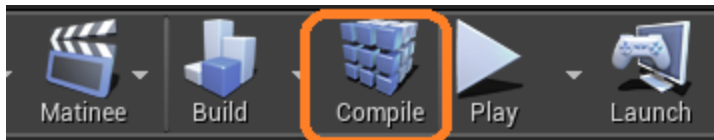
```
UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Item)
    FText Description;
```

8. Paste it to Book.h after GENERATED_BODY()
9. Right after the GENERATED_BODY() add a new line:
public:
10. Then you just have to rename the new field, change its type (if needed) and category

I renamed it like this:

```
UPROPERTY(EditAnywhere, BlueprintReadOnly, Category = Book)
FText BookText;
```

11. In UE4 click **Compile**, wait until it says compile success, close UE4 right away (don't save anything if it prompts you) and open again. Done! :)

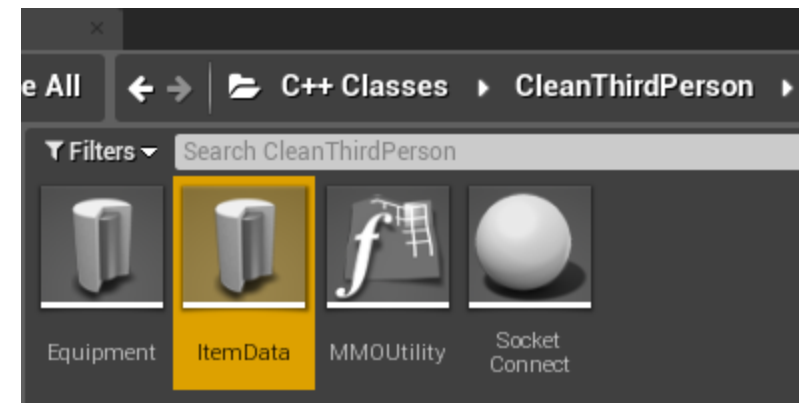


After that you will be able to create a Book asset with 1 additional Text field.

Customizing item quality tiers

1. In the Content Browser, navigate to **C++ Classes/CleanThirdPerson/ItemData** and double-click on it to open Visual Studio
2. Add or remove the item quality tiers.

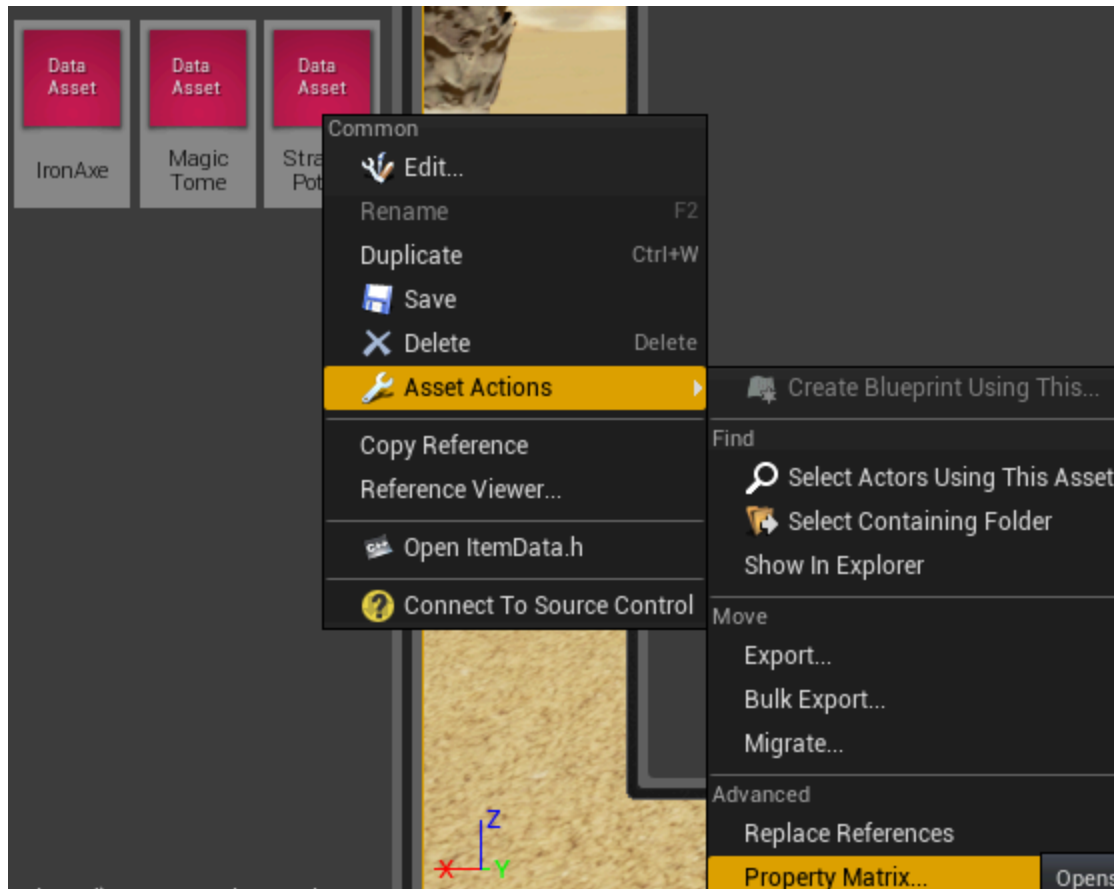
```
ENUM(BlueprintType)          // "BlueprintType" is essential to include
enum class ItemQuality : uint8
{
    Poor        UMETA(DisplayName = "Poor"),
    Common       UMETA(DisplayName = "Common"),
    Uncommon     UMETA(DisplayName = "Uncommon"),
    Rare         UMETA(DisplayName = "Rare"),
    Epic         UMETA(DisplayName = "Epic")
}
```



Editing multiple assets

If you want to edit multiple NPCs or items at once or if you want to edit them individually but there is a lot of them, use the property matrix:

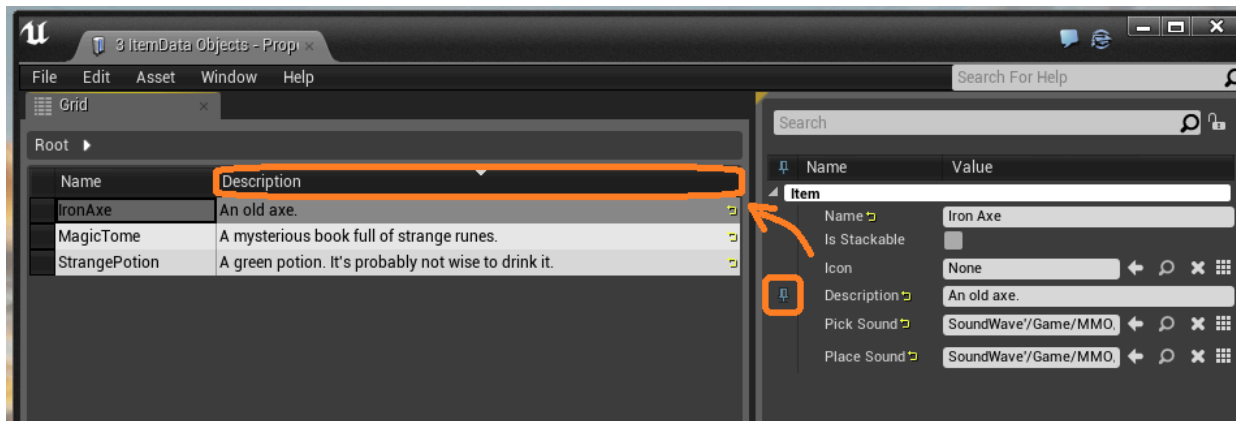
1. Select the assets that you want to edit, **right-click > Asset Actions > Property Matrix**



2. In the resulting window, you can select a single asset and edit its properties (to the right).

3. To edit multiple assets at once, select all the assets that you want to modify with shift and modify the properties that you want just like you would to with a single item.

NOTE: in property matrix you can sort the assets by any property by pinning the property so it shows up in the table and clicking on the column name.



Adding your own persistent stats

Persistent stats are saved to the MySQL database when the character logs out and retrieved by server when the character logs in. Examples of persistent stats already in the demo are: character name, health, position, equipped helmet. You could add your own stats, such as: character haircut, skin color, money, etc. Here's how to do this:

1. Add your new fields to the 'characters' table in the database, preferably at the end of the row. If you add them in the middle, make sure the order of all fields is correct in step 4.

Server: localhost » Database: secondhand_mmo » Table: characters

Structure

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)			No	None	AUTO_INCREMENT	
2	user_id	int(11)			No	None		
3	name	varchar(20)	utf8_general_ci		No	None		
4	class	int(11)			No	None		
5	gender	int(11)			No	None		
6	health	int(11)			No	None		
7	mana	int(11)			No	None		
8	level	int(11)			No	None		
9	experience	int(11)			No	None		
10	posx	int(11)			Yes	NULL		
11	posy	int(11)			Yes	NULL		
12	posz	int(11)			Yes	NULL		
13	rotation_yaw	float			No	None		
14	equip_head	varchar(100)	utf8_general_ci		Yes	NULL		
15	equip_chest	varchar(100)	utf8_general_ci		Yes	NULL		
16	equip_hands	varchar(100)	utf8_general_ci		Yes	NULL		
17	equip_legs	varchar(100)	utf8_general_ci		Yes	NULL		
18	equip_feet	varchar(100)	utf8_general_ci		Yes	NULL		

Check All With selected:

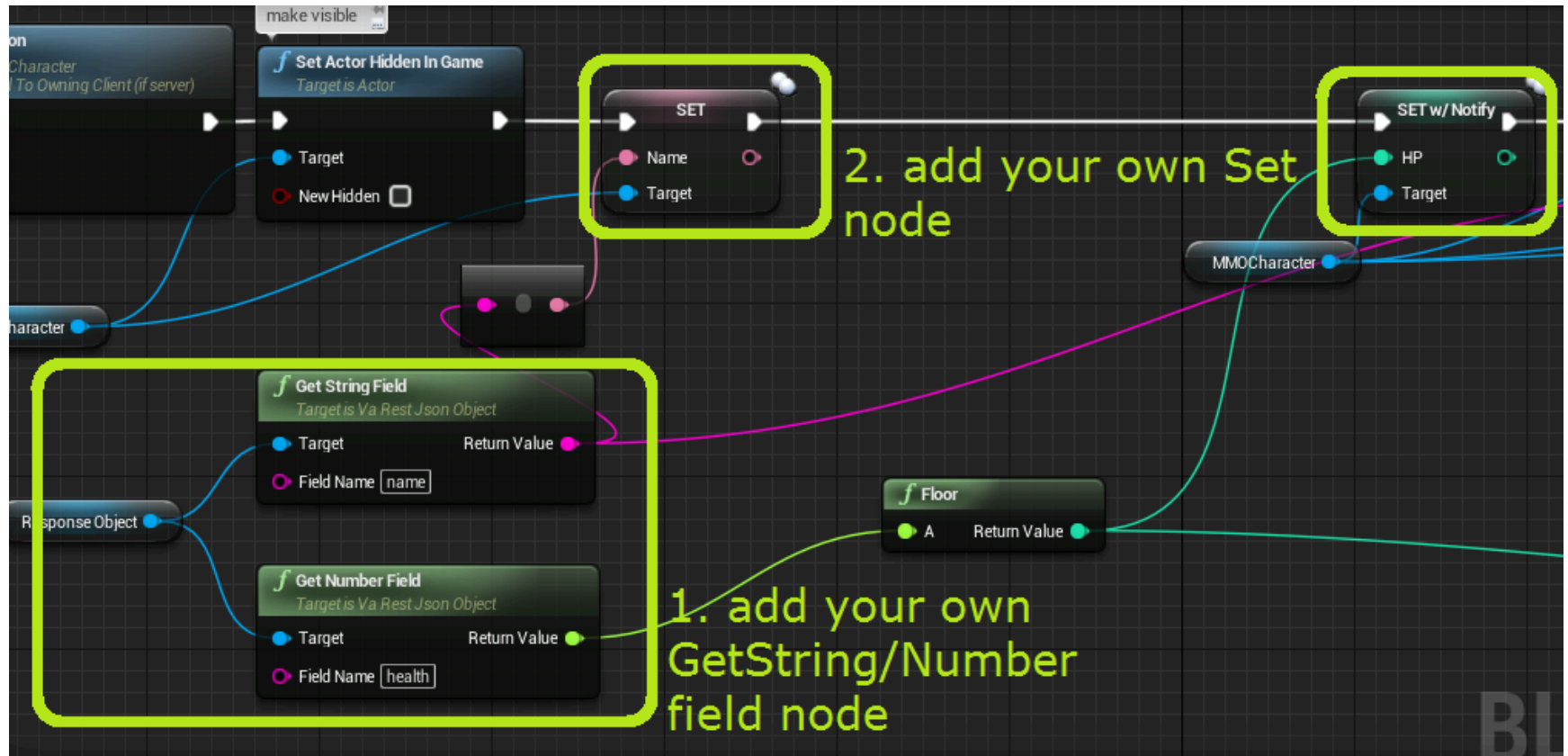
Print view Propose table structure Move columns

Add 1 column(s) At End of Table At Beginning of Table After id Go

+ Indexes

2. Add the new fields to your player character blueprint (or to MMOCharacter if the enemies/NPCs also need to have them).

3. In **MMOPlayerState** blueprint add the new fields to **CreateSaveCharacterRequest** and **OnResponseGetCharacter** (don't confuse the latter with **OnResponseGetCharacterSI**). Look how other fields are handled there and do the same:



4. Add your new fields to **mmosavecharacter.php**, **mmogetcharacter.php** and **mmocreatecharacter.php** scripts. Look how other fields are handled there and do the same. For example do a search for "health", see how the "health" stat is handled, and do the same for your stat.

NOTE: when editing php scripts, make sure your prepared statements' syntax is correct. The number of '?' in the query and the number of "s" or "i" letters should be the same as the number of variables after them. Example: `$stmt->bind_param("ssss", $health, $posx, $posy, $posz);`

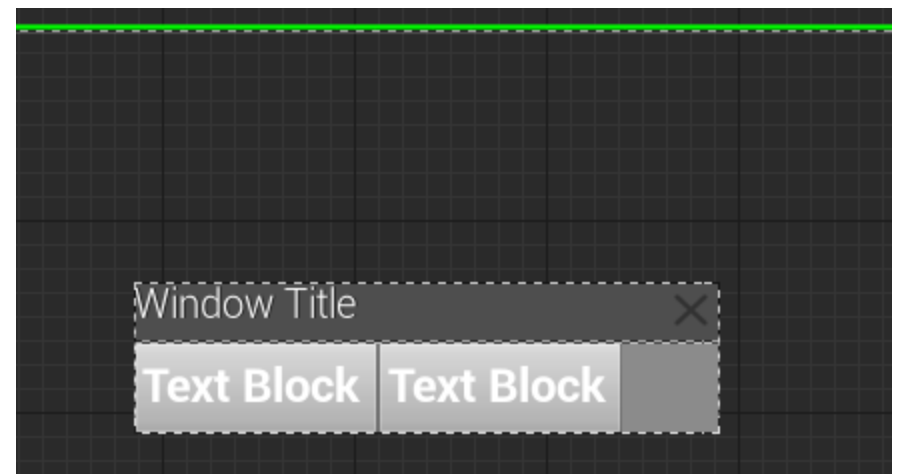
NOTE: When you modify the stats in game, make sure you're doing it **on the server**, since it's the server that will send them to the database.

Creating your own draggable windows

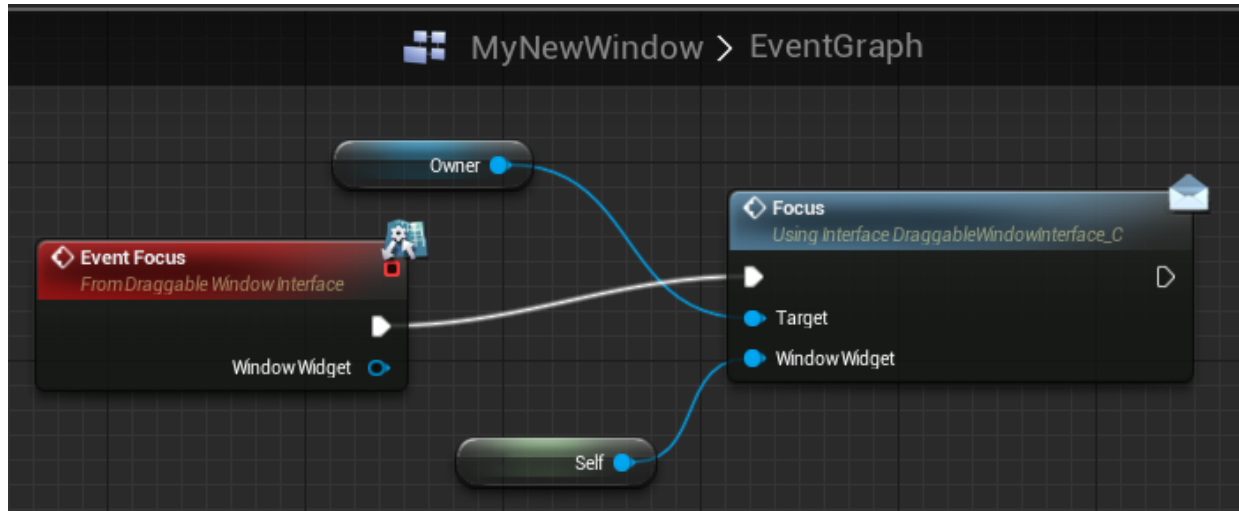
This guide will show you how to create your own draggable windows that can gain focus and appear on top of others just like the Inventory/Equipment windows in the demo.

NOTE: I intend to make this a bit easier in the future, but decided to provide the guide for the current version nevertheless:

1. Create a new UMG widget (Right-click in any folder - User Interface - Widget Blueprint) and open it
2. In the User Created category find a **Draggable Window** widget and drag it onto CanvasPanel in the Hierarchy. Rename it to delete the C_1 text (to just "DraggableWindow").
3. Choose the new window in the Designer view and resize it to the desired size
4. In the Details tab enter the desired name into **Default - Window Title** field. (The change won't be visible in the Designer, it will still be called "Window Title", it's normal).
5. Drag any content you want into ContentSlot of the window in the Hierarchy. Here I dragged a Horizontal Box and then two buttons:
6. Go to the Graph view and create a new Variable of type UserWidget named Owner. Check Editable for it.



7. Go to Class Settings and under Interfaces add a **Draggable Window Interface**. Compile.
8. Add a new Event Focus event like this:



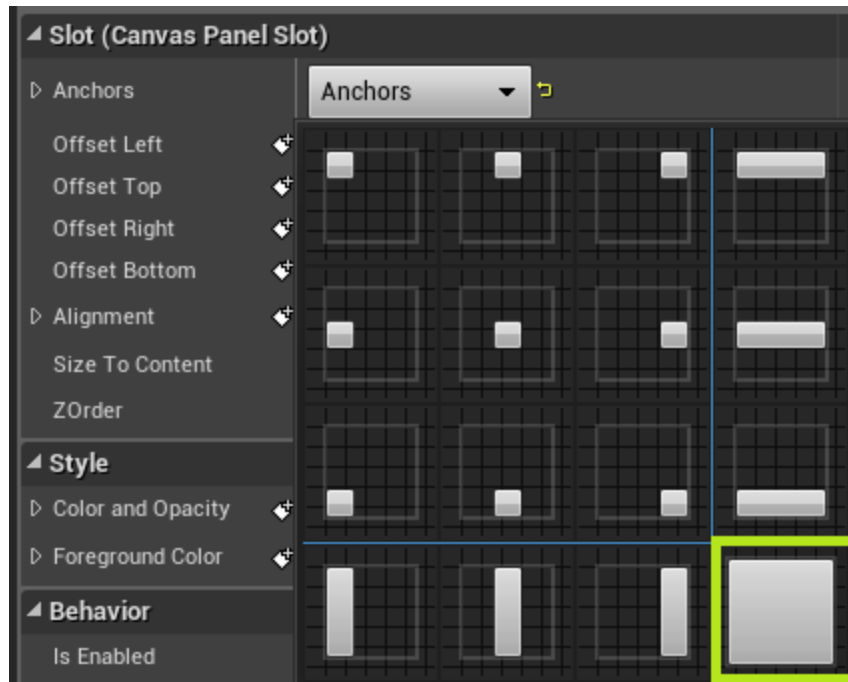
9. And an Event Construct like this:



10. Save the widget and close.
11. Now open the **FullUI** widget, located in MMO/Blueprints/UserInterface

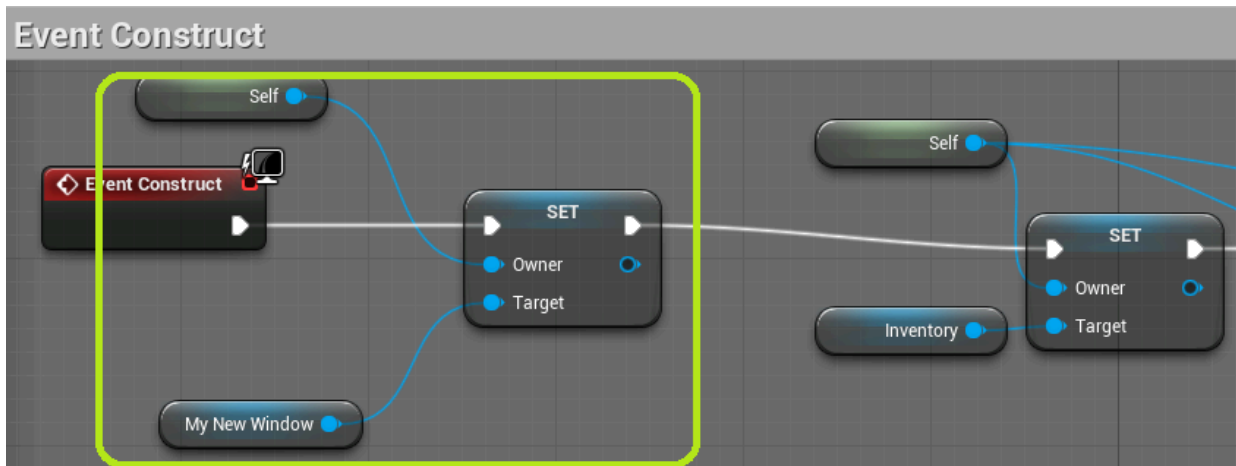
12. In the User Created category find your newly created widget and drag it onto the Canvas panel in the Hierarchy. Rename it to delete the C_1 text.

13. In Details, choose the anchor in the lower right corner:



14. Set all Offsets to 0. The widget should take all the available space now.

15. Go to the Graph view. In the beginning of Event Construct, set the Owner variable on your new window to Self, just like it's done for Inventory/Equipment windows:



16. At the end of Event Construct, add a pin to the Make Array node and add connect your new window there.

17. (skip this if your window doesn't need use a hotkey to open it) Now we need to assign some key to toggle our new window on and off. Open the **MMOPlayerController** in MMO/Blueprints and find the yellow comment block "C key: toggle equipment". Make a similar one with the key of your choice.

18. (optional) If the window doesn't need to be open when the game starts, in the Designer tab of your new window set **DraggableWindow** widget's visibility to **Hidden**.

Q: Why do we have a Canvas panel as the root for the draggable window widget and not the window itself?

A: This way the window can have popups that exceed the window's own borders.

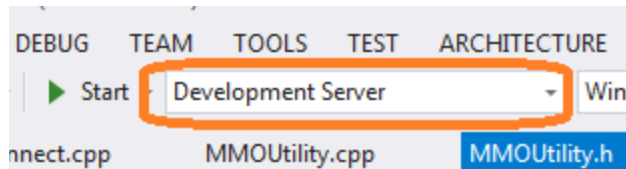
Running the server on your local computer

NOTE: if you want to set up a server that will run 24/7, you could look into [ordering a virtual server](#).

Dedicated Server and Packaging

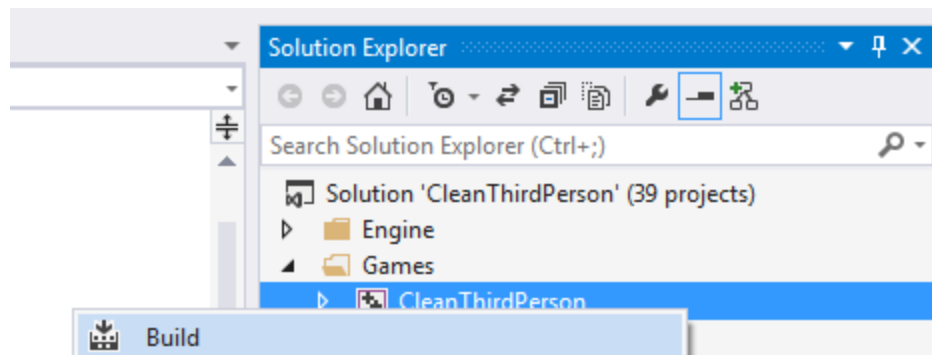
A game has to be packaged in order to be able to run without UE4 editor. A console dedicated server is needed in order to run your game on a virtual/bare metal server (for example, Softlayer or Amazon) which normally have no video cards and to maximize performance.

1. Open VS (**File > Open Visual Studio**), choose **Development Server** configuration.



NOTE: If you don't have this option, make sure you're using a [source-built version of Unreal](#). If you have a source-built engine, but there's still no "Development Server" option in VS, check that the project uses the right engine. You can do that by right-clicking on the .uproject file and selecting "Switch Unreal Engine version"

2. In Solution Explorer right-click on your project (Games/CleanThirdPerson) and choose **Build**:

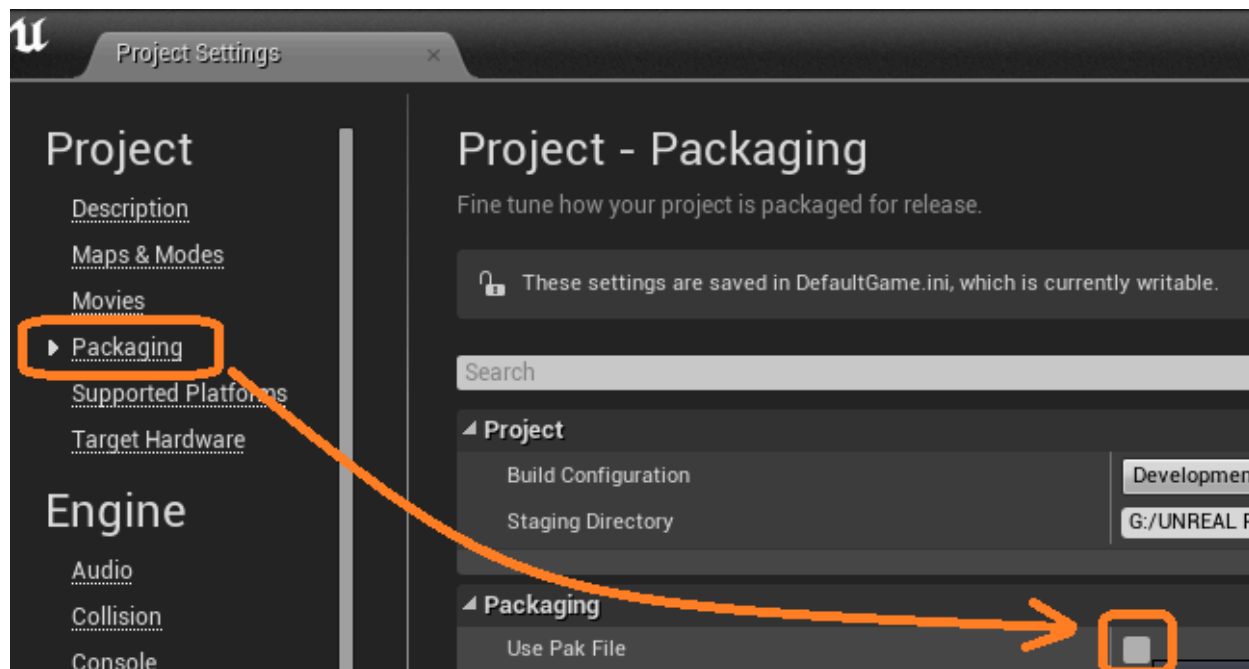


3. Depending on your machine, this can about 5-15 minutes. You will only have to do it once per the version of Unreal or until you modify something in the engine or game code.

4. After it's done, switch the configuration from Development Server back to **Development** and close the Visual Studio.
5. A dedicated server can only run from a packaged game folder, so let's package the game now. First, go into **File > Package Project > Packaging settings** and check **Full Rebuild**. Next time you package the game, you can uncheck this setting and the packaging will be much faster - around 2 minutes.

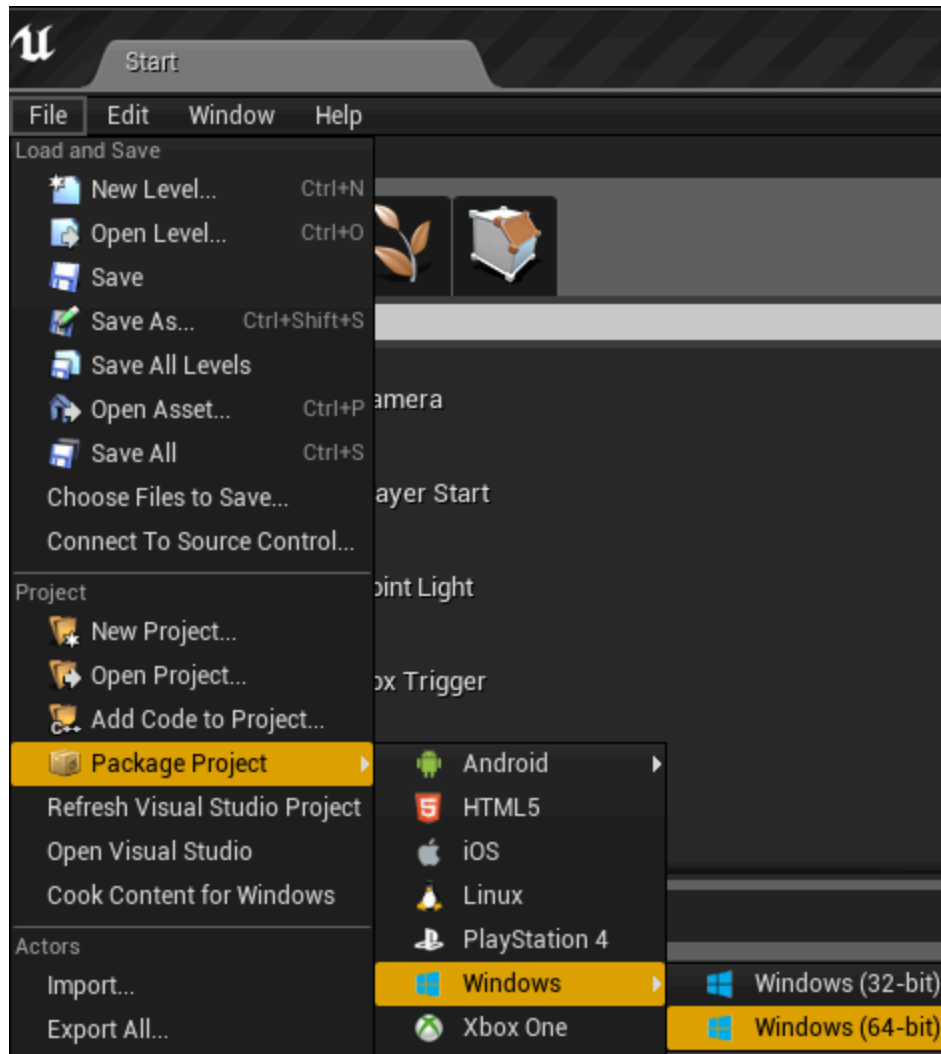
NOTE: normally you only have to do a Full Rebuild the first time and if you port the project to a new UE4 version.

6. Make sure **Build Configuration** is set to **Development** (otherwise the game won't generate any logs) and **Use Pak file** is unchecked:



7.

8. Go to **File > Package Project** and choose the target hardware and an empty folder for the packaged game, for example: C:\MmoPackaged\. Don't use the project or engine folder for that! The packaging process will take about an hour.



9. After the packaging stops, go to the folder that you chose for the packaged game and check if it has something in it. If it is empty, this means there was an error during the packaging. You can check **Window > Developer Tools > Output Log** to see what the error was.
10. If the packaging was successful, go to your project folder, and copy the Binaries\Win64\<ProjectName>Server.exe to **<ProjectName>\Binaries\Win64** folder of the packaged game.
11. (optional) run the chat server
12. Create a shortcut for the file you copied in the previous step, and add parameters “-log -local”. Run the shortcut. A console window should appear - this is the dedicated server.
13. Create a shortcut for the client .exe located in the same folder and add a “-local” parameter. Run the shortcut and test if you can connect to the dedicated server and play.
14. If you want other people to connect to this server, follow the next part of the guide.

Server IP and clients

In order for the clients to be able to connect to your dedicated server, you need to specify the server IP address in mmogetserver.php. Find this text “address’=>” , and specify your ip address there.

NOTE: when connecting remotely, there should be no **-local** parameter in the client or server shortcut (so the address from mmogetserver.php will be used).

Removing the players limit

To remove the default players limit, in your project folder open file Config\DefaultGame.ini and add these lines on top (where 100 is the desired number):

```
[/Script/Engine.GameSession]  
MaxPlayers=100
```

Setting up dedicated server on a VPS

Ordering a VPS

At some point you will want to have a server that is online 24/7 and doesn't interfere with your other activities on your computer. You can maintain this server yourself on a dedicated machine or order it from a hosting provider of your choosing. This guide explains how to order and set up a virtual server on [Softlayer](#), where MMO Starter Kit demo server is being hosted (E5520 Xeon @ 2.27 GHz, 1GB RAM). You can safely skip it if you already have a VPS or want to use a different provider.

NOTE: while MMO Starter Kit demo is currently being hosted on Softlayer, it's not the cheapest hosting provider. I recommended it to new users because of 24/7 chat support and very fast deploy times. For most of my servers I'm using a different provider, [ipserver](#), i7 4790k for \$79/month.

1. Archive and upload to your favorite cloud (for example, Dropbox) the following :

- the packaged game (in a .zip)
- the chat server (in a .zip)

2. Order a Softlayer virtual server with Windows Server OS through www.softlayer.com . You can order an hourly server first to test things out, and if everything is fine, order a monthly server.

The lowest virtual server configuration is enough for small-scale testing, though you might want to increase the disk space. With a 25 GB hard drive you will have only around 2.5 gb free space with the fresh Windows install.

When determining which data center to use - if your users are from Europe you could use Amsterdam, and you could choose Dallas for the US. MMO Starter Kit demo server is hosted in the data center in Washington to provide decent ping for Europe and the US both.

You could check the stats at <https://wondernetwork.com/pings> to have an estimate at what your users' ping will be.

If you have PayPal, you can pay with it instead of a credit card, then your order will be processed instantly. Credit card orders can take longer or even fail if your bank requires a special verification. The website says your order will be processed within 15 minutes. If you don't receive an email from Softlayer in 15-30 minutes, you can contact their live chat directly - usually they answer immediately 24/7.

3. Once your order has been approved, you'll receive your login and a temporary password, log in and change the password to a permanent one.

4. Wait until you get the email saying the server has been provisioned and is ready to use.

Transferring the server to a VPS

NOTE: This guide is for Windows Server VPS. While UE4 dedicated server can work on Linux, I don't provide support for Linux servers currently. If you choose to go with Linux for the dedicated server, keep in mind that you will have to package the game twice each time you package.

1. Upload your packaged game and the chat server to a cloud of your choice (for example, Dropbox)

2. In Windows, Run > Remote Desktop Connection. Use the ip address, username and password of your VPS for the remote connection. If using Softlayer, to get the ip of your VPS go to control panel, then to **Devices**, and double click on your server.

3. On your VPS, open up Internet Explorer and go to the cloud where you've uploaded the game files. Download all the files that you've put there.

4. On your VPS, install the prerequisites located in your game's folder **Engine\Extras\Redist\en-us**.

5. Start the chat server and the dedicated server on the VPS.

6. Copy the ip of your VPS to **mmogetserver.php** script.

Done! Try to run the game as a client (using the .exe from your packaged game folder) and see if it connects to your new server.

NOTE: after ordering a VPS server, I recommend to always make sure that the plan in **Control Panel - System and Security - Power Options** is set to **High Performance**. Otherwise with some processors (for example i7 4790k) it may result in the CPU being underclocked.

Launcher/autoupdater

There's a separate documentation for the Launcher, you can find it [here](#).

Removing demo content

Important! Make sure you have backed up your work or are using version control before proceeding.

If you decide that you don't need the demo "desert map" in your project, you can remove it to reduce the project's size:

- 1) Delete the map itself, Maps/DesertRallyRace
- 2) Remove all the meshes and effects from the Start Map
- 3) Delete the DesertRallyAssets/ and StarterContent/ folders.

Troubleshooting and FAQ

FAQ (not issues)

Q: Why does the engine have to be built from the source?

A: You can only build a dedicated server when using the source-built version of UE4. It's not specific to MMO Starter Kit, it's like that for all UE4 projects

Q: Which ports does the server use?

A: The game server uses the default Unreal 7777 port (UDP), and the chat server uses the port 3457 (TCP).

Q: Are you planning to release the girl model used in the demo?

A: The sword girl model is not by me, it's by Bunt Games and can be found on Unity Asset Store

Q: Are the future updates going to be free? How can I get them?

A: Yes, all updates will be free and available for download automatically to all customers.

Q: What version control system can you recommend?

A: I'm using Git without the integrated features, just regular Git, SourceTree and GitLab that I host on Linode. Levels that are bigger than 100 MB are shared via Dropbox. I believe that's the way to go for teams that have 1-4 people working with the editor.

Q: How can I rename the project?

A: I haven't tried renaming the project yet, it's not trivial for code projects in UE4. Here are the instructions posted by UE4 staff member:

<https://answers.unrealengine.com/questions/242407/renaming-a-c-project.html>

Q: How can I have different armor meshes and/or animation for different gender/class/race?

A: You could add a bool variable isMale to the character blueprint and in variable settings set **Replication - RepNotify**. In its OnRep function it can go over all equipment and change it to the correct version with **SetSkeletalMesh** node. **CanExecuteCosmeticEvents**

node can be used to only execute that on the client. The Equipment class should have an additional field for FemaleMesh. When an item is equipped, it uses a Select node + IsMale to determine whether to set female mesh or male mesh.

For different character classes/races you can use the same principle, just a enum instead of a bool. You can also change the animation blueprint by using **SetAnimInstanceClass** node.

Q: How to ban players?

A: You can handle bans entirely in php - in the 'users' table create a new column - bool "isBanned". Change mmologin.php so it doesn't allow users with isBanned == true to login.

Note that you will also have to change mmregistration.php, the INSERT query that creates the new user, since the amount of columns in 'users' has changed

Q: How to only allow certain accounts into the game (closed beta testers, customers who've purchased the game...)?

You can use one of these two options:

A) distribute game keys, let the user enter the key at registration along with username/email, and check if it's valid. For example you can make a new "keys" table in your MySQL database, with fields: id, key(varchar) and activated(bool). Add some keys to it manually or using import. Mmoregistration.php should check if the key provided by the user is in the "keys" table and not activated yet. Once a user registers using the key, set "activated" = 'true' for that key.

B) you can get Greenlit on Steam and distribute/authorize through Steam. If you're Greenlit, it is possible to distribute the game through Steam to closed beta testers while not in Early Access yet.

Q: Why do you have several "character" blueprints?

A: *MMOCharacter* is the base blueprint that both AI characters and players derive from. Modular character blueprint (*ModularPlayerCharacter*) is not needed if your game doesn't need modular equipment. And *MyCharacter* is a base movement and camera blueprint almost without any mmo functionality.

Q: How can I persistently store world data, for example player structures?

A: You can save the world data to MySQL database in the same way the character data is saved. You would have to create a new php script similar to savecharacter.php, and handle sending the request in MMOPlayerState blueprint. In MMOGame or some other server blueprint you can set use a **Set Timer Delegate** node that will call the request every 5 mins or so.

Q: I don't want the player's character to destroy immediately on logout. How would I achieve that?

A:

1. Add a new C++ class that derives from APlayerController
2. In this new class, override PawnLeavingGame(). The original looks like that:

```
if (GetPawn() != NULL)
{
    GetPawn()->Destroy();
    SetPawn(NULL);
}
```

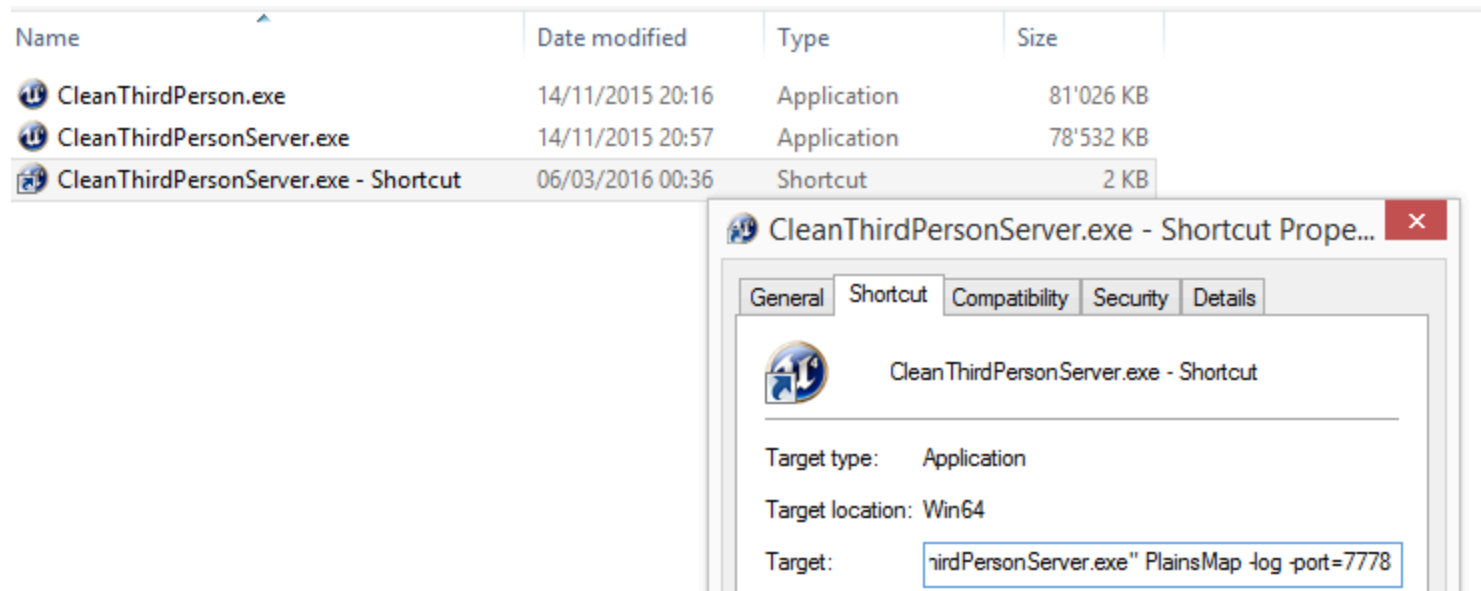
so you can just copy it and comment out the destroy line

3. In the editor, change the parent of MMOPlayerController to your new class

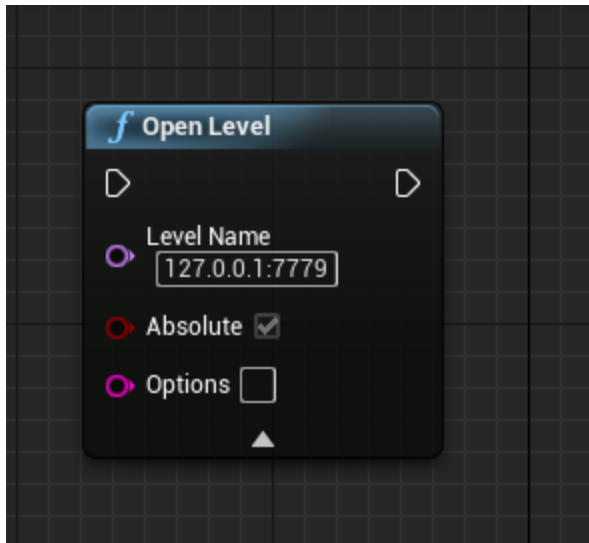
Q: How can I implement travel between zones?

A: In the game I'm working on we use this system: each physical server has several mmo kit dedicated server processes running, one for each zone, each uses a certain fixed port: 7777, 7778, 7779, etc.

For example you could have server shortcuts with these parameters: "PlainsMap -log -port=7777", "ForestMap -log -port=7778":



To transfer between server processes, when a player walks into a trigger volume, **OpenLevel** blueprint node is called on the client with the correct zone's ip and port. Example:



To travel between levels locally (while not connected to a server), you can call Open Level node with the name of your level instead of ip:port, for example "MyNewMap".

When the client enters world, it should connect to the server process with the zone they last logged off. To do that, you could add a "map" column to the "characters" table in the database and save the map along with other character info (in MMOPlayerState blueprint, CreateSaveCharacterRequest). When the client gets the list of its characters with name, level, etc, it should also receive the map and use it to connect to the right server.

After the connection, when the server retrieves the character data from the database (MMOPlayerState blueprint, OnResponseGetCharacter), it should also receive the map and compare it to the map this server is running. If the map is the same, it means the player has returned from being offline, so the location from the database should be used (the current default behaviour). If the map is different, it means the character has travelled from another map and should be placed at some portal or gate location.

Q: What about instanced zones (instanced dungeons, battlegrounds)?

A: For instanced zones, you would use the same principle but launch the server instances dynamically (you could extend C# chat socket server source code to do that). The game server could send a command to the socket server and the latter would launch the

server process with required map parameter and the next available port parameter and send back the port number. The port number is then sent to the players via blueprint so they can connect to the new instance.

Q: How can I have different PvP rules per zone?

A: It is definitely possible. I can see two main features that you would have to add (in blueprints):

1. *Different PvP rules per map*: the easiest way is to create an bool or enum "PvPMode" in your main GameMode and set the default value, for example "Safe Zone", then create a GameMode derived from your main GameMode and set PvPMode in it to something different, for example "Enabled". Note that GameMode only exists on the server, but you probably need that info on the client as well for some visual effects - for that you can send the map's PvPMode value from the server to the client when the client joins the map.
2. *Zone travel* - see zone travel question above

Q: How can I implement a no-target combat system?

A: If you know how to make a no-target combat system in a single player UE4 project, it shouldn't be hard to add that to the MMO Kit in place of the default system. You could use [Line Trace For Objects](#) for weapons that use crosshair (the line would start at the center of the screen and go forward in the camera direction, its length would be the gun's range) and [Sphere Overlap Actors](#) for weapons that don't.

There are tutorials available on the web/YouTube on how to make a simple FPS setup in UE4. I recommend making your system work in a clean TPP/FPP project first, and then transfer it to MMO Kit.

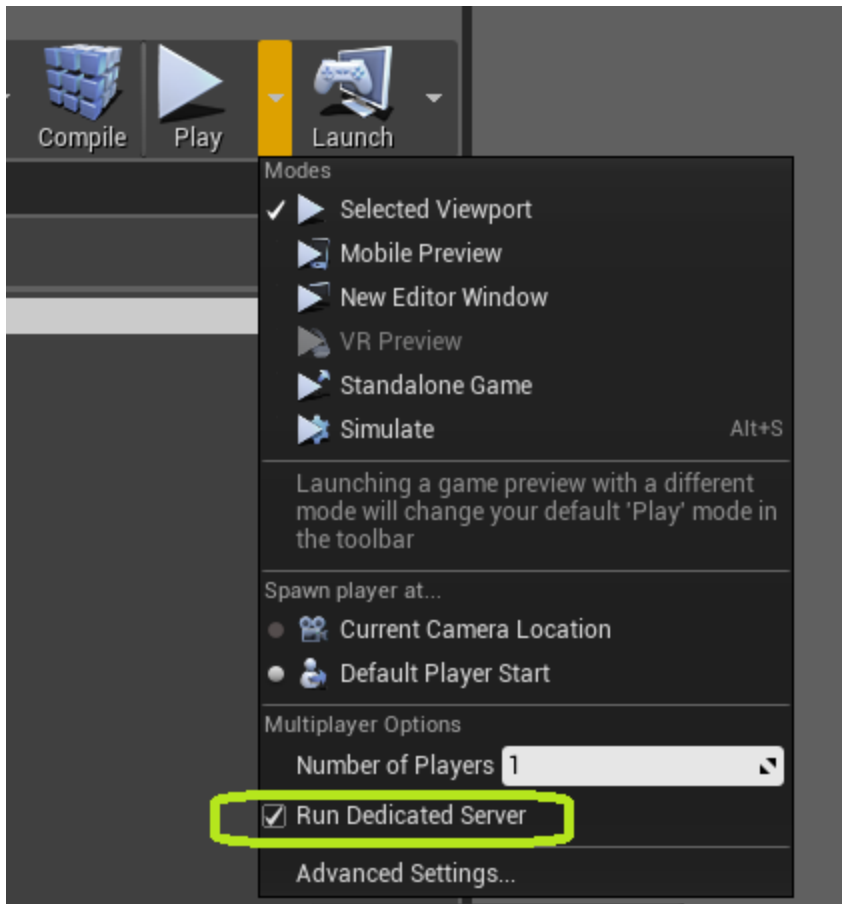
The blueprints that you'd want to change are:

MMOPlayerController - Right-Mouse Button event starts auto-attacking
MMOCharacter - Auto-Attack event

Problems with entering world

Q: When running the game in editor, the loading screen doesn't go away

A: Make sure that **Run Dedicated Server** is checked in the play settings:



If it is checked and the loading screen still doesn't go away, most likely the error is with your php/mysql, you can check the Output Log (Window - Developer tools - Output log) for the exact error.

If you have a completely black screen (not the one that says "loading"), check that there are no UE4 server processes running in the background.

Q: I can't connect to my remote dedicated server. The IP address in mmogetserver.php is correct. I can connect just fine when I play in editor or when I run a local dedicated server and use -local parameter on the client.

A: Open your latest server log located in Saved/Logs and search for "Join request".

If you've found a "Join request" in the log:

Most likely the server and client are running different versions of the game. Reupload the packaged game folder to the server and client.

If there was no "Join request" in the log:

Check your firewall settings on your local machine and on the dedicated server machine. You can temporary disable the firewall altogether to quickly see if it was indeed the issue (just remember to turn it back on after that!) If disabling it helps, check that the 7777 and 7778 UDP ports and the client and server UE4 applications are not blocked. If disabling the firewall doesn't help, try pinging your server ip from the command line - see if you can ping your server at all.

If you're hosting the dedicated server yourself, check the IP in the server console. If the IP is private (starts with 192.168.), most likely you are behind a router.

Crashes, game not launching or not compiling

Q: When running the game in editor, I get a crash after I choose a character and try to enter world

A: You can't enter world from start menu when running in editor. You can package the game to fully test it locally but normally I only do packaging before releasing a new patch to the end users.

Q: When I try to launch the server .exe, I get multiple errors that start with "Error: CDO Constructor"

A: Make sure you've packaged the game and copied the server.exe to Binaries folder of the packaged game (where the client .exe is). You should launch the server from that folder, not from the project folder.

Q: I tried to run the packaged game (or server) on another machine and it doesn't launch at all

A: First of all, make sure you've transferred the whole packaged game's folder (WindowsNoEditor/).

To be able to run UE4 games you have to install the UE4 prerequisites first, located in the packaged game's Engine/Extras/ folder.

Q: When compiling the project, the compile fails with this error: *Cannot open include file: 'windows.h': No such file or directory*

A:

- 1) Make sure that your Windows user name doesn't contain any special characters.
- 2) Make sure you've included C++ in your installation of visual studio 2015. You can check this by going to **Control Panel -> Programs and Features**. Select "Microsoft visual studio community 2015", press change , press modify in the installer and look for C++ and hit install.
- 3) If the steps above didn't work, try reinstalling Windows SDK: <https://msdn.microsoft.com/en-us/windows/desktop/bg162891.aspx>
- 4) Another possible solution: go to **Control Panel\System and security \ System -> Advanced system settings -> Environment Variables**. Under "System variables" check if "Path" variable has "C:\Windows\System32;C:\Windows\SysWOW64" , if it doesn't, add it

Q: When I open your project in Unreal version *.xx, it crashes

A: Are you using the latest stable version? The "sneak peek" versions and branches can have unexpected behaviour, I recommend you use the latest stable version of Unreal Engine.

Q:When I switch levels, the game crashes with this error: "X not cleaned up by garbage collection!"

A: If you haven't modified anything in the demo project, check the first question in this section instead. Make sure your persistent classes such as GameInstance(MMOInstance) have no references to the actors in the previous level. Check the log for the information on which classes to look for. You can clear the references by setting them and not attaching anything to the set node. Also, when you create UMG widgets, add to them to an array and destroy the widgets in that array before loading a new level.

Q: The game crashes with BSOD/system reboot, what to do?

A: This is a hardware error, most probably your video card overheats. It can be fixed by running the game with shortcut parameter **-VSync** (Unreal doesn't use VSync by default so your frame rate can go very high which can cause the video card to overheat). If it still crashes, but you can play for a few minutes, try entering in console in game: **t.maxfps 30**, that will set the max fps even lower. If the crash happens right when you start the game, you will have to add **t.maxfps=30** to an .ini file instead.

Q: Most people can run the game fine, but a certain customer/tester always gets a crash when trying to launch the game.

A: Let this person try to launch some of UE4 demos:

<http://www.techpowerup.com/downloads/2369/unreal-engine-4-five-tech-demos/>

If the demos launch fine, then probably the antivirus is blocking the game. In this case, usually no error will be thrown, no logs will be created. Try disabling the antivirus.

If the problem persists with the demos, this means it's not related to your game, and most likely has to do with the videocard / videocard drivers. Known causes:

- 1) This [Windows 7 Service Pack 1 update](#) has helped some people who couldn't launch any UE4 game.
- 2) Another possible cause is having both integrated Intel video card and another video card. In case of Nvidia cards, to fix this the user has to make a custom profile for this game in Nvidia Control Panel, and there is a setting to force it to use Nvidia card

Other issues

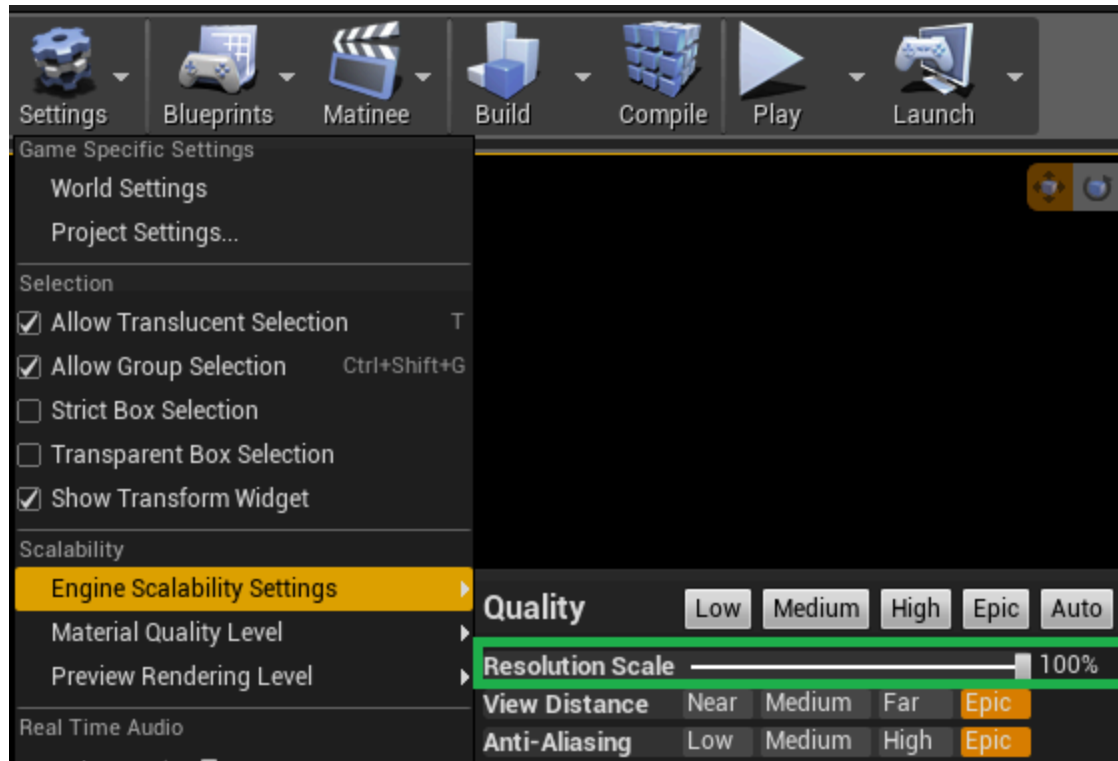
Q: How do I check if my PHP server is 5.5 or higher?

A: Upload the `myphpversion.php` found in PHP/ folder of the kit to your webserver to the same folder where you've put the other php scripts. Run the script from browser, it should output something like:

```
Current PHP version: 5.5.16-pl0-gentoo2.0
```

Q: Everything looks low quality/like the resolution is very low, how to fix this?

A: Setting **Resolution Scale** to 100% should help:



Q: Where is <some functionality> done in blueprints? I can't find it.

A: You can ask me on PM/Skype/forum thread, however I recommend you use **Edit > Find in blueprints** feature to save some time. For example you could search for "Create Widget" to find where the UI is created. :) Note that this search doesn't include level blueprints, except for the currently open level.

Q: When a player character dies, it falls through the terrain

A: Check [Using your own map](#) step 6.

Q: I've made some changes and now Unreal crashes on play/startup. How do I fix this?

A: First of all, you really should use version control in the future to avoid situations like that. I use regular non-integrated Git and SourceTree.

If you don't have version control, there are autosaves in Saved/Backup and Saved/AutoSaves folders - if you know what blueprint causes the crash, you could look there to restore an older version of it.

If you don't know which blueprint causes the error, you could move all of the blueprints from the project folder to a separate folder and check if the error persists. Then move half of them back, etc until you narrow it down.

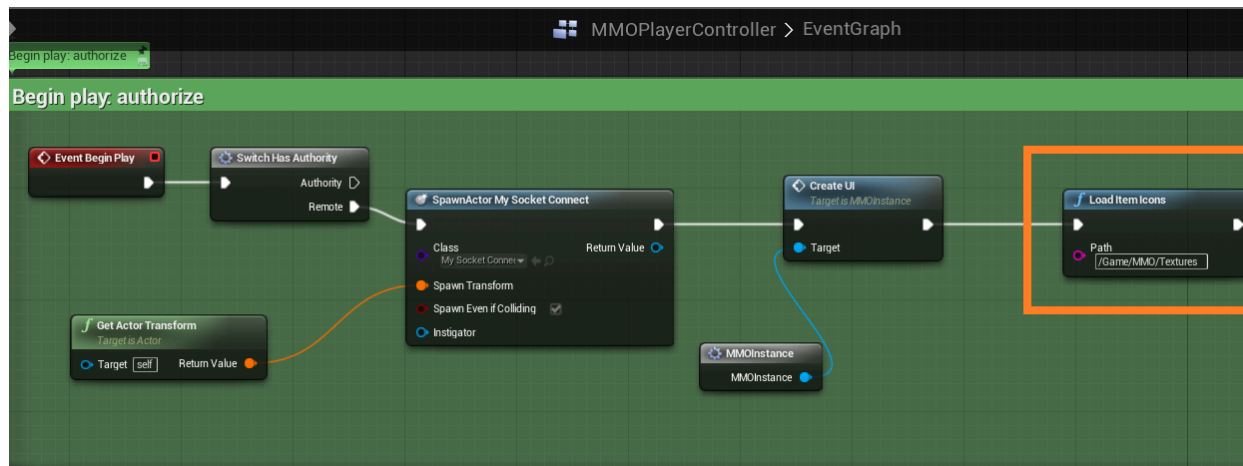
Q: I walked off the edge of the map and now my character falls when I play in editor

A: In your mysql database, in 'characters' table find the character with id 0 and set their posx, posy and posz to some appropriate coordinates in your map.

Q: My inventory item/ability textures show as white sometimes

A: Hi, all item icons should be placed in the same folder as the demo item icons (/Game/MMO/Textures), since they are loaded from that folder when the game starts. (UE4 doesn't load textures which are not referenced automatically, and item icons are not referenced when the game starts)

If you want to change that folder, change this node in MMOPlayerController:



Q: When testing things in editor, I can't seem to block the controller movement/mouse look in Start map (the one with the login menu). SetIgnoreMouseInput and SetIgnoreMovementInput don't work.

A: Make sure you are running the starting map with Dedicated server turned off and 1 player.

Debugging PHP

Q: One of my PHP scripts is not working, how do I find the error?

A: After each VaREST request the response is written to the UE4's **Output Log** (Window - Developer tools - Output log). There can be three possible situations here:

a) If the response code is **200**, this means the php script is working fine:

```
LogVaRest: Response (200): {"status":"You are not logged in."}
```

b) If the response code is **not 200**, it will usually also list your php version. If the php version is lower than 5.5, you need to upgrade your php server.

c) If the response code is **not 200 and the php version is 5.5+**, you have to look for your **php error log** - it's location can vary depending on your hosting, some of the hosting providers allow you to access this log from the control panel, others allow ftp access.

If you've written the new php script yourself, you can also use the online php checker to check it for syntax errors.