

# MMO RPG Camera & Controller Manual

## Version 3.7.1

### Table of contents:

- 1 Getting started
  - 1.1 Attaching the scripts
  - 1.2 Creating the needed Inputs
  - 1.3 Adding animations
  - 1.4 Assigning the right layer (optional)
  - 1.5 Tagging objects right (optional)
  - 1.6 Using the correct shader (optional)
- 2 Variable overview and explanation
- 3 Version history
- 4 Got questions or problems?

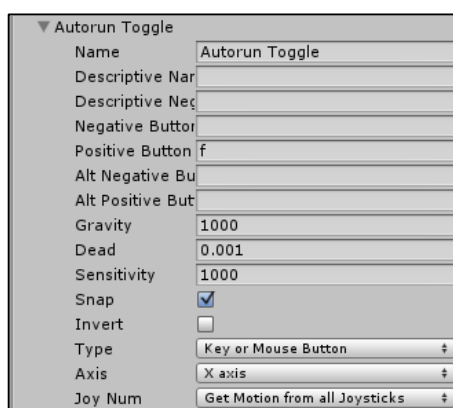
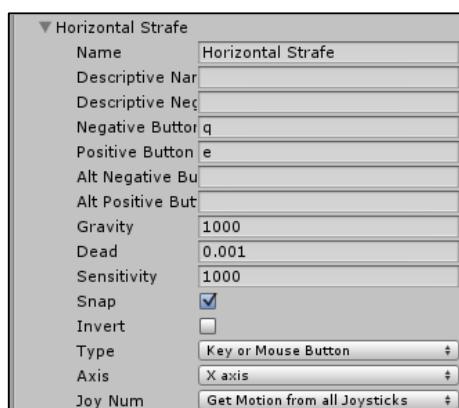
## 1 Getting started

### 1.1 Attaching the scripts

- Attach the script “RPGCamera.cs” to your character game object if you would like to have the camera controls
- Attach the script “RPGController.cs” to your character game object if you would like to have the character controls
- Attach the script “RPGMotor.cs” alone if you would like to use your own controller script but use my motor script. Do not forget to start the motor via method StartMotor() after every input was set

### 1.2 Creating the needed Inputs

Set up the following Inputs in the Project Input Settings (Edit > Project Settings > Input). The used buttons shown in the pictures are the same as in the demo – of course you can use whatever buttons you want. Please refer to [johnstairs.com](http://johnstairs.com) for an input mapping overview.



▼ Walk Toggle

Name: Walk Toggle

Descriptive Name:

Descriptive Negative:

Negative Button:

Positive Button: x

Alt Negative Button:

Alt Positive Button:

Gravity: 1000

Dead: 0.001

Sensitivity: 1000

Snap: ☒

Invert: ☐

Type: Key or Mouse Button

Axis: X axis

Joy Num: Get Motion from all Joysticks

▼ Sprint

Name: Sprint

Descriptive Name:

Descriptive Negative:

Negative Button:

Positive Button: left shift

Alt Negative Button:

Alt Positive Button:

Gravity: 3

Dead: 0.001

Sensitivity: 3

Snap: ☒

Invert: ☐

Type: Key or Mouse Button

Axis: X axis

Joy Num: Get Motion from all Joysticks

▼ First Person Zoom

Name: First Person Zoom

Descriptive Name:

Descriptive Negative:

Negative Button:

Positive Button: page up

Alt Negative Button:

Alt Positive Button:

Gravity: 1000

Dead: 0.001

Sensitivity: 1000

Snap: ☒

Invert: ☐

Type: Key or Mouse Button

Axis: X axis

Joy Num: Get Motion from all Joysticks

▼ Maximum Distance Zoom

Name: Maximum Distance Zoom

Descriptive Name:

Descriptive Negative:

Negative Button:

Positive Button: page down

Alt Negative Button:

Alt Positive Button:

Gravity: 1000

Dead: 0.001

Sensitivity: 1000

Snap: ☒

Invert: ☐

Type: Key or Mouse Button

Axis: X axis

Joy Num: Get Motion from all Joysticks

### 1.3 Adding animations

Just add the provided Animator component (Animator Controller > Animator Controller) to your character game object and assign the desired animations to their corresponding states in the Animator tab. Now you should be good to go! If you want to use the animations from [Mixamo](#) I used for the demo, here are their names with the corresponding motion:

Motion	Mixamo Animation
Fall	Falling Idle
Idle	Idle
Run Backward Strafe	Jog Backward Diagonal
Run Backward	Running Backward
Run Forward Strafe	Jog Forward Diagonal
Run Strafe	Jog Strafe Left
Run	Jog Forward
Running Jump	Running Jump
Sprint	Sprint
Stunned	Dizzy Idle
Swim Idle	Treading Water
Swim	Swimming
Turn	Left Turn
Walk Backward Strafe	Walking Backwards
Walk Backward	Walking Backwards
Walk Forward Strafe	Right Strafe Walking
Walk Strafe	Right Strafe Walking
Walk	Walking

## 1.4 Assigning the right layer (optional)

The default value for the variable “Occluding Layers” of the RPG View Frustum is the “Default” layer mask. Therefore, if your character game object has the layer “Default” assigned, it will be faded out slightly when its materials has a transparent shader set. If you want your character to be set up like the one in the demo, you have to set its layer to “Ignore Raycast” or any layer which is not part of the “Occluding Layers” layer mask (e.g. a new one called “Player”).

## 1.5 Tagging objects right (optional)

By default the variable “Occlusion Handling” of the RPG View Frustum is set to “Tag Dependent”. So every game object with a tag contained in the “Affecting Tags” list of the RPG View Frustum forces the camera to zoom in. For example, the tag “AffectCameraZoom” is an affecting tag by default and if you assign it to a wall object, the camera will zoom in when the wall enters the view frustum.

## 1.6 Using the correct shader (optional)

If you want to use the fading feature of the RPG View Frustum, the objects you would like to see fade have to have a material with a transparent shader attached. You can use the shader I used for the demo (Resources > Shaders > Bumped Diffuse with Z) if you want. It is a simple bumped diffuse shader with transparency and Z-buffering enabled.

## 2 Variable overview and explanation

In this chapter, I am going to explain some public variables and what they do as their names may not be self-explanatory. For a complete variable overview please go to [johnstairs.com](http://johnstairs.com).



### RPG Camera

#### Used Camera

References the camera object used by the script. If no camera is assigned to this variable, a new one will be generated automatically when entering the play mode.

#### Used Skybox

This is the current skybox used by the camera referenced by UsedCamera. The skybox can be changed at runtime by calling the RPGCamera script's method “SetUsedSkybox(*material*)”. Direct assignments to this variable **will not** take effect.

#### Camera Pivot Local Position

This is the local position of the camera pivot, the “anchor” of the camera. Turn on the script's Gizmos to visualize it as a small cyan sphere.

#### Activate Camera Control

If set to false, all camera controls are disabled. Can be used to



turn off the camera when interacting with a GUI (e.g. see the demo GUI interaction).

### **Always Rotate Camera**

If set to true, you do not have to press “Fire1” or “Fire2” to control the camera.

### **Align Character**

Determines when the character’s view/walking direction should be aligned to the camera’s view direction. By default, the character is only aligned with the camera when “Fire2” is pressed.

### **Align Camera When Moving**

If set to true, the camera view direction aligns with the character’s view/walking direction when the character starts to move (forward/backwards/strafe). If additionally **Support Walking Backwards** is set to true, the camera faces the front of the character when walking backwards.

### **Underwater Threshold Tuning**

Gives the possibility to fine-tune the threshold where the camera is seen as underwater. The higher the value, the earlier the underwater effects kick in. The effects are enabled/disabled via methods “EnableUnderwaterEffects()”/“DisableUnderwaterEffects()”. Use these two methods to implement your own visual effects.

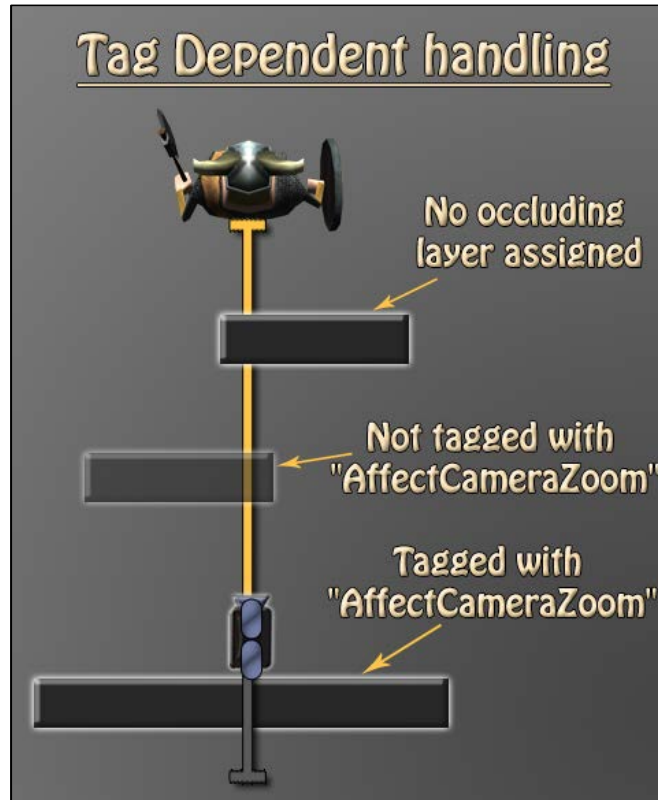
## **RPG View Frustum**

### **Occlusion Handling, Occluding Layers and Affecting Tags**

These three variables might be the most difficult ones to understand. Therefore, a larger explanation follows:

- \* When a game object enters the camera’s view frustum and has an *occluding layer* assigned, it gets affected (faded out) or affects the camera zoom (lets the camera zoom in) depending on the “Occlusion Handling”
- \* “Occlusion Handling” controls how the camera reacts on objects inside its view frustum:
  - “Do Nothing” - every object is ignored and has no effect
  - “Always Zoom In” - every object forces the camera to zoom in to retrieve a clear sight on the camera’s pivot

- “Tag Dependent” – every object with a tag contained in the “Affecting Tags” list forces the camera to zoom in.  
Every object which has an occluding layer and **no** affecting tag assigned gets faded out when entering the frustum and faded back in when leaving the frustum



### Fade Out Alpha

The alpha to which objects fade out when they enter the view frustum.

### Fade In Alpha

The alpha to which objects fade back in after they left the view frustum.

### Enable Character Fading

If set to true, the character starts to fade when the camera’s distance to its pivot is smaller than the “Character Fade Start Distance”.

### Character Fade Start Distance

This is the distance between the camera and its pivot where the character fading starts.

### Character Fade End Distance

This is the distance between the camera and its pivot where the character is completely faded out to “Character Fade Out Alpha”.

### RPG Motor

#### Allowed Airborne Moves

The character is allowed to move slightly while airborne after performing a standing jump. This variable saves the maximum number of allowed airborne moves.

### Swimming Start Height

Defines the local water height at which the character should start to swim. Enable Gizmos for easier setup.

### Dive Only When Swimming Forward

If set to true, the character starts to dive only when moving forward (or backwards). Otherwise, the character remains in an upright stance.

### Sliding Threshold

The value of this variable represents the terrain/ground angle at which the character starts to slide down. Therefore, a value of “40” means that the character object starts to slide off every hill or surface which is steeper than 40 degrees.

### Falling Threshold

A value representing the degree at which the character starts to fall. The default value is “6” to let the character be grounded when walking down small hills.

### \_autorunning

Private variable which indicates if the character is currently automatically moving forward. It is toggled via method ToggleAutorun() and stopped via method StopAutorun(). With the RPGController, autorunning is automatically stopped if a manual “move forward” input was given.

### \_stunned

Private variable which indicates if the character is currently stunned. A stun can be triggered by calling method “Stun(duration)” which stuns the character for “duration” seconds. The corresponding getter method is “IsStunned()”.

## 3 Version history

### v3.7.1

- \* Created dummy animations and assigned them to the Animator Controller so that the demo settings are not lost. Note that the “Mirror animation” checkbox has to be set manually after replacing the dummies.
- \* Added an animation mapping table to the manual to easier find out which Mixamo animation was used for which motion in the demo.

### v3.7

- \* Introduced new feature: Swimming
- \* Introduced new [RPGMotor](#) variable “Swim Speed Multiplier”. Defines the multiplier which is applied to the character movement speed when the character is swimming
- \* Introduced new [RPGMotor](#) variable “Swimming Start Height”. Defines the local water height at which the character should start to swim. Gizmo available for easier setup
- \* Introduced new [RPGMotor](#) variable “Dive Only When Swimming Forward”. If set to true, the character starts to dive only when moving forward (or backwards). Otherwise, the character remains in an upright stance
- \* Introduced new [RPGCamera](#) variable “Underwater Fog Color”. Defines the fog color which should be applied when the camera below the water surface



- \* Introduced new [RPGCamera](#) variable “Underwater Fog Density”. Defines the fog density which should be applied when the camera below the water surface
- \* Introduced new [RPGCamera](#) methods “EnableUnderwaterEffects()” and its counterpart “DisableUnderwaterEffects()”. Enables/disables the visual underwater effects which should be applied once the camera is below water surface. Use these two methods to implement your own visual effects.
- \* Introduced new [RPGCamera](#) variable “Underwater Threshold Tuning”. Gives the possibility to fine-tune the threshold where the camera is seen as underwater. The higher the value, the earlier the underwater effects kick in
- \* Complete overhaul and simplification of the provided [Animator Controller](#)
- \* New demo with character model and new animations
- \* Fixed camera jittering which sometimes occurred when the character was aligning with the camera
- \* If the character is stunned, the camera can now continue to orbit around the character with RM or the alignment input pressed
- \* Added the possibility to stun the character during the demo (button ‘T’)
- \* Minor code optimizations
- \* Updated to the newest Unity stable release

### v3.6

- \* Introduced new [RPGMotor](#) variable “Unlimited Airborne Moves”. If set to true, the number of allowed moves while airborne is unlimited
- \* Introduced new [RPGMotor](#) method “Stun(duration)” which stuns the character for “duration” seconds, disabling movement and transitioning into the new “Stunned” animator state
- \* Introduced new [RPGCamera](#) variable “Align Character Speed” which controls the character-to-camera turning speed
- \* Character fading now correctly depends on the distance between the camera and the character
- \* Small logical adjustments

### v3.5

- \* Introduced new [RPGMotor](#) variable “Move With Moving Ground”. If set to true, the character will stay on moving objects like moving platforms
- \* Introduced new [RPGMotor](#) variable “Rotate With Rotating Ground”. If set to true, the character will rotate with rotating objects while standing on them
- \* Introduced new [RPGMotor](#) variable “Ground Object Affects Jump”. If set to true, the jumping direction gets affected by the (moving) object the character was standing on
- \* Introduced new [RPGCamera](#) variable “Rotation Stopping Input” for setting which input should be pressed to pause automatic camera rotation with the character (only usable if “Rotate With Character” is set to “Rotation Stopping Input”)
- \* Introduced new [RPGCamera](#) variable “Alignment Input” for setting which input should be pressed to align the character’s view direction with the camera view direction (only usable if “Align Character” is set to “On Alignment Input”)
- \* Updated the demo scenes for the new features
- \* Updated the GUI layout
- \* Updated the manual for using the [RPGCamera](#) with Unity’s ThirdPersonController standard asset

- \* Minor bugfixes

#### v3.4

- \* Fixed a bug when the [RPGController](#) and [RPGCamera](#) were used independently
- \* Added instructions to the Documentation folder for using the [RPGCamera](#) with Unity's ThirdPersonController standard asset
- \* New [RPGCamera](#) variable "Rotate With Character" which lets you set if the camera should rotate as well when the character turns
- \* Minor code additions and changes

#### v3.3

- \* When LM is pressed, the camera does not rotate with the character anymore
- \* When jumping into objects from below, the character bounces off immediately
- \* Introduced new [RPGCamera](#) variable "Support Walking Backwards" for better character alignment when walking backwards
- \* Improved the provided demo shader
- \* Fixed a bug where it was possible to turn the camera upside down in first-person mode

#### v3.2

- \* Fixed a bug in first person view which caused the camera to behave incorrectly
- \* Fixed a bug where it was possible to force the camera under terrain for a few frames
- \* Added the possibility to group the [RPGCamera](#) script variables in the inspector

#### v3.1

- \* Fixed an animation bug in the demo caused by Unity's new editor version 5.1.1f1
- \* Improved interleaving of scripts
- \* Improved [RPGController](#) and [RPGMotor](#) scripts
- \* Implemented a better approach for camera rotation
- \* Introduced new [RPGCamera](#) variables:
  - \* Used Skybox
  - \* Constrain Mouse X
  - \* Mouse X Min
  - \* Mouse X Max
- \* Moved the variable "Align Character With Camera" to [RPGCamera](#) and renamed it to "Align Character" to improve readability and efficiency
- \* Updated manual instructions

#### v3.0

- \* Added animations controlled by Mecanim
- \* Added a walking toggle
- \* Object fading between the camera and its pivot
- \* Different occlusion handling modes selectable
- \* Tag dependent camera zoom enforcement (easy setup in the inspector)
- \* Slower backwards walking if desired
- \* 21 new public variables for easier and more customization
- \* New GUI for the demo



- \* Replaced the RPGClipPlane by the [RPGViewFrustum](#) and moved view frustum computations there
- \* Removed a bug where you could jump while sliding resulting in greater jumps

#### v2.1

- \* Update for Unity 5

#### v2.0

- \* Complete revision
- \* Huge amount of new controller and camera features added
- \* UML diagram included
- \* Let out the animation scripting

#### v1.1

- \* Added the "RPG\_Animation" script
- \* Added a capsule model with animation clips attached to it
- \* Changed object hierarchy of the "PlayerChar" object
- \* The cursor is not locked anymore if you press the left mouse
- \* New public variable "fallingThreshold" in "RPG\_Controller" defining the terrain height at which the character starts to stumble
- \* Setting the layer of PlayerChar to "Ignore Raycast" is not necessary anymore
- \* Some code and names improved/edited for clearer understanding

### 4 Got questions or problems?

Feel free to send me an email to [mail@johnstairs.com](mailto:mail@johnstairs.com) if your question is not covered by the FAQs on [johnstairs.com](http://johnstairs.com)! Please attach a screenshot showing the used variable values and occurring error messages.

Best regards,

*John Stairs*