Realistic Head Controller V1.0

General Documentation

Support: & https://www.inanevin.com/realistic-head-controller

Video Tutorial of This Documentation:

Intro

Welcome, thanks for purchasing Realistic Head Controller! This documentation will guide you through the basic usage of the system. In order to see how to call the necessary methods in the system, and also how to extend the system for more capabilities, please check the Scripting Documentation. Let’s get started!

What is Realistic Head Controller?

It’s an asset that includes scripts that would manage your camera movement, preferably in an FPS game, in order to create the most realistic feeling possible. It uses 5 gameobjects to receive input from any player controller and act according to those inputs. All variables are customizable to fit your needs, and there are also included presets, to provide an easy of use while implementing.

What kind of movements are included in Realistic Head Controller?

- **Camera Shakes:** Both custom shakes and preset explosions, earthquake shakes that are easy to call!
- **Camera Recoil Effect:** An effect to simulate any camera recoil, whether for player being hit, or firing a weapon. You can customize the amounts really easily and call one line of code to hit the camera!
- **Camera State Positioning:** RHC includes a system to adjust the position & rotation of the camera, meaning the orientation, according to the player's current state. So, in your player controller, you would just notify the system that player has crouched, proned, jumped etc. And the Camera State Positioner will place the camera to the orientation accordingly. You can customize the orientations for different states with few button clicks to suit your needs!
- **Camera Head Bobbing:** RHC includes an advanced head bobbing system, which is fully customizable. You can simulate any kind of movement bobbing effects on the camera, completely dependent on the player's state. So you'd have options to select in which state of the player, what kind of a head bobbing shall be performed. There are included presets, to simulate: walking light, walking medium, walking heavy, walking injured, walking carrying an item, walking drunk,

running and crawling, and many more!

- **Camera Controller:** An industry level camera controller that provides a smooth mouse look that would fit in any FPS game!

What is the Main Workflow of Realistic Head Controller?

All the components of RHC works accordingly to your custom player controller! It has 2 different player states in *RHC_EventManager* script. The components in RHC reads those 2 player states, and acts accordingly. *RHC_EventManager* also includes custom methods to change states easily, so the only thing you have to do when your player changes a state is to call a method in *RHC_EventManager* using a single line of code to ensure camera fits perfectly to the state! Let's look at those 2 states.

- **PlayerStanceStates:** Player's stance states. Includes -> Standing, Crouching, Crawling and OnJump.
- **PlayerMovementStates:** Player's movement states. Includes -> Idling, Walking and Running.

RHC Working Scehema

RHCStatePositioner object reads the states in RHCEventManager and orients the camera according to which stance state the player is in. As the stance states change, RHCStatePositioner goes to standing, crouching, proning or jumping orientation, and ensures that all transitions are smoothed out correctly!

RHCBobController object reads the states in RHCEventManager, and decides what kind of bobbing should be applied. There are 6 default bobbing methods:

- Walk Bob
- Run Bob
- Crouch Walk Bob
- Crouch Run Bob
- Crawl Slow Bob
- Crawl Fast Bob

And all of these bobbing methods are easily customizable! Moreover, it’s pretty easy to add new methods depending on your custom player controller. Check the Scripting Documentation to extend the capabilities of RHC!

RHCShakeController object waits for the events in *RHC_EventManager*. You can call a custom shake event, or a generic shake event, or a recoil event. All the event calls are single line of code, so they are pretty easy to implement. Once an event is called, *RHCShakeController* object shakes the camera based on the settings, which are easy to customize!

Usage

Go to *RealisticHeadController -> Prefabs* and drag & drop the *RHCCamera* prefab to your scene. Make it the child of your player controller, and delete your camera(or replace it with the one inside the

prefab). Now you can configure the settings on the objects, select presets and customize. After that, you only need to call the necessary methods to let the system work. To see how to call methods, check the Scripting Documentation!

Inspector Variables

In this section, the inspector usage of all the objects in RHC are given in detail. For more information, check the video tutorials!

> *The word "camera" in the descriptions refers to game object that the script is attached to.*

Shake Controller

Variable	Description
Use Recoil Shake	Toggles whether to use recoil shakes or not.
Min Rotation	Minimum rotation values. (The rotation value to be used will be randomized)
Max Rotation	Maximum rotation values for randomization.(The rotation value to be used will be randomized)
Up Speed	Speed to use while interpolating camera TO the randomly selected rotation.
Down Speed	Speed to use while interpolating camera FROM randomly selected rotation to original rotation.
Use Increasing	If selected, the power multiplier to multiply with the randomly selected rotation will increase on each

Power	recoil hit, from a min value until a max value with a given acceleration and deceleration.
Start Power	Starting value for multiplier’s power.
End Power Maximum	Value that the multiplier’s power can go to.
Acceleration	How fast will the multiplier’s power will go to “end power” from “start power”.
Deceleration	How fast will the multiplier’s power gill go to “start power” from “end power”.
Add New Shake Type	Adds a new generic shake.
Remove Last Shake Type	Removes the last added generic shake.
Reset List	Removes all shake types.
Generic Shake Types	Generic Shake Types
Name	Name of the shake type.
Position Amounts	How much shake will be applied in the terms of position?
Position Speeds	How fast the positional amounts will be applied to shake?
Rotation Amounts	How much shake will be applied in the terms of rotation?
Rotation Speeds	How fast the rotational amounts will be applied to shake?
Shake Time	For how many seconds will this shake, shake the

	camera?
Apply Distance	If the this value is bigger than the distance to the source of a generic shake, apply this shake. (If there is another shake which has still bigger distance, but is much closer to the source distance, that shake will be applied. E.G Shake A : Distance 15, Shake B: Distance 20, Source Distance: 5, Shake A will be applied.)

> *When recoil event is fired, a random rotation value will be chosen between these Min & Max Rotations.*

Camera State Positioner

Variable	Description
Air Position	Position of the camera(state positioner object) when player is on air.
Air Rotation	Rotation of the camera when player is on air.
Air Down Position	Position of the camera when player has landed from air.
Air Down Rotation	Rotation of the camera when player has landed from air.
Go Down Position Speed	The speed to apply the positional change from Air Position TO Air Down Position.
Go Down Rotation Speed	The speed to apply the rotational change from Air Rotation TO Air Down Rotation.

Crouch Position	Position of the camera when player has crouched.
Crouch Rotation	Rotation of the camera when player has crouched.
Crawl Position	Position of the camera when player is in crawl state.(proned)
Crawl Rotation	Rotation of the camera when player is in crawl state.
Position Interpolation(Under Speeds)	How fast will all the positional changes be applied? (Except interpolation from Air Position to Air Down Position)
Rotation Interpolation(Under Speeds)	How fast will all the rotational changes be applied? (Except interpolation from Air Rotation to Air Down Rotation)
Air/AirDown/Crouch/Crawl/Default Buttons	Saves the current position & rotation of the gameobject as Air/AirDown/Crouch/Crawl/Default position & rotation. / Loads the Air/AirDown/Crouch/Crawl/Default positional & rotational values and applies them to the gameobject.

Bob Controller

Variable	Description
Reset Speed	How fast will the camera reset to it’s original orientation if no bobbing is present?
Select	Select which settings you’d like to edit.

To Edit	
Preset	Preset to use.
Bob Style	Positional -> Only changes the camera’s position while applying bob. Rotational -> Only changes the camera’s rotation while applying bob. Both -> Changes both the camera’s position & rotation while applying bob.
Amounts	Positional/rotation amounts to apply while bobbing.
Speeds	How fast will the position/rotation amounts be applied while bobbing?
Smooth	The value used to smooth out head bobbing. (Bigger -> slower, Smaller -> faster)

Camera Controller

Variable	**Description**
Rotate Player	If check, then a field will appear to assign player transform. Then the camera will rotate the player transform from the horizontal input of the mouse, and rotate itself from vertical input. (Moving the mouse right & left will turn the player around, up & down will turn the make the camera look up & down.)
Sensitivity	Horizontal(X) & Vertical(Y) sensitivity.
Smooth	Smoothing values for applying sensitivities. (Bigger -> slower, Smaller -> faster)
Minimum Y	What is the maximum DOWNWARDS angle can the camera go to? (Looking down limitation)
Maximum Y	What is the maximum UPWARDS angle can the camera go to? (Looking up limitation)