Clonefactor

Camera Bot

Version 1.2

Canis
5-21-2015

Table of Contents

Overview

Have you ever always need to change the view of the camera in your games, but found it's hard to develop? Or even feel headache after playing the game for a long period of time? Here you can find the Antidote.

"CameraBot" design as the dynamic orbit camera controller, based your own setting in editor, to support {first person, 3rd person, shoulder camera, product display…etc.}

For designer

No script knowledge needed, you can use all these features by drag and drop, and then tune the parameters in the inspector panel.

For programmer

All source code provided. Welcome to discuss and report bugs that you found

Features

The CameraBot supports the following features:

- Orbit control manager, handling the relative position between chasing target and camera position & angle
- Double orbit tracking system, instead of facing the chasing target itself, camera are able to face the virtual relative position for chasing target.
- Preset camera in the inspector, predefine the camera position and chasing target in editor mode, during playing 3D game can dynamic change the position and focus target by user's input.
- Motion sickness, adjustable setting for acceleration and deceleration for camera movement & rotation, reducing player camera sickness effect.
- Camera seamlessly transitions during switching camera mode.

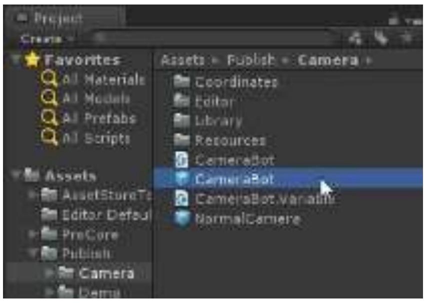
- The full editor interface and gizmo shows different variety of coordinates, developer can visualize the reference of camera position and their facing direction of selected camera, reduce the time for level design.
- Object-oriented method & No dependence of hierarchy structure, developer can place script component anywhere they want. All script object are encapsulation and able to control by giving method.

Installation guide line

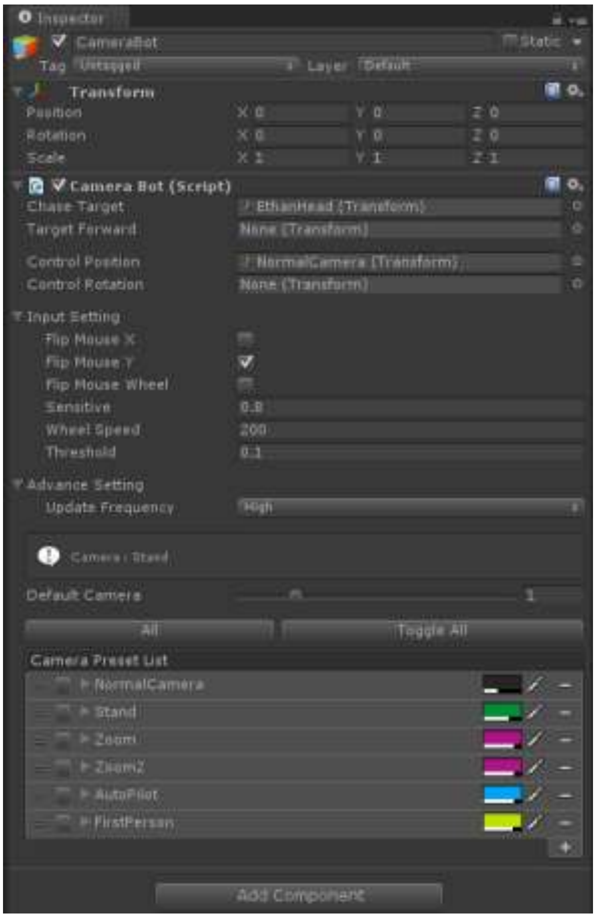
I. Download from asset store

II. Import asset into your project

III. Add controller prefab into scene, In Unity3D editor's Project tab, Search for "CameraBot.prefab" and drag it into hierarchy



IV. Select the CameraBot game object you just drag into hierarchy, you'll find the "Chase Target" and "Control Position" are empty, drag the target you want to chase into "Chase Target" field and then drag the camera you want to control into "Control Position" field (If Control Position field are empty, script will auto search for main camera in the game)



☑ CameraBot

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Camera Bot (Script)

Chase Target: EthanHead (Transform)

Target Forward: None (Transform)

Control Position: NormalCamera (Transform)

Control Rotation: None (Transform)

Input Setting

- Flip Mouse X ☐
- Flip Mouse Y ☑
- Flip Mouse Wheel ☐
- Sensitive 0.8
- Wheel Speed 200
- Threshold 0.1

Advance Setting

- Update Frequency High

Camera : Stand

Default Camera 1

All | Toggle All

Camera Preset List

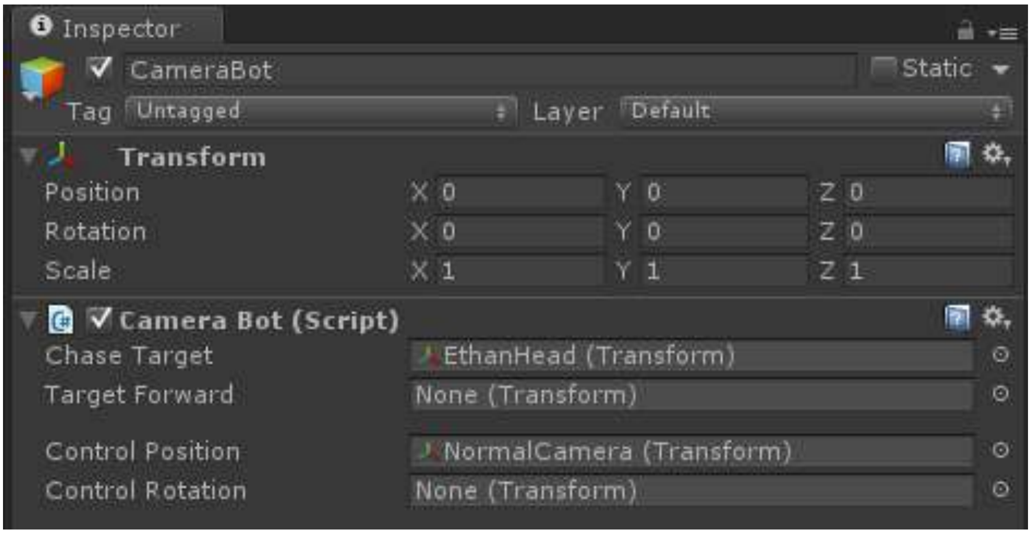
- NormalCamera
- Stand
- Zoom
- Zoom2
- AutoPilot
- FirstPerson

V. And go to "Camera Setting" create your preset camera by camera setup guide line.

VI. Now you basically finish the setup of "CameraBot". Enjoy!

Setup guide line (Detail)

An example of provided demo scene.



Inspector

☑ CameraBot □ Static

Tag Untagged Layer Default

Transform

Position X 0 Y 0 Z 0

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Camera Bot (Script)

Chase Target EthanHead (Transform)

Target Forward None (Transform)

Control Position NormalCamera (Transform)

Control Rotation None (Transform)

Chase Target

The target game object you want CameraBot to chase.

Control Position

The game object usually (Camera) that you want CameraBot to control and stick to orbit position as you preset in editor, leave it blank will auto search for the main camera in the scene.

Target Forward

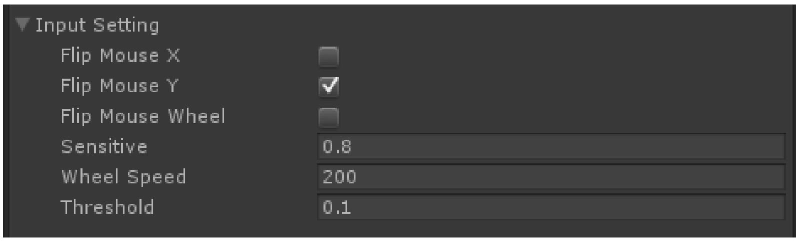
When “Chase Target” was not always point to same point, e.g. you target on a animate object, use this(Target Forward) to select the transform to represent the forward direction, CameraBot will use that as the direction reference to calculate the relative angle.

Control Look At

The game object usually (Camera) that you want CameraBot to control and rotate that angle toward the *chase target*. Leave it blank will automatic use the same object of *Control Position*

Input Setting

The input section for CameraBot.



▼ Input Setting

Flip Mouse X	☐
Flip Mouse Y	☑
Flip Mouse Wheel	☐
Sensitive	0.8
Wheel Speed	200
Threshold	0.1

Flip Mouse X

Opposite direction of horizontal input value. E.g. move left with become move right

Flip Mouse Y

Opposite direction of vertical input value. E.g. move up with become move down

Sensitive & Wheel Speed

The enhancement for the input value. Usually used for amplify mouse input value

E.g. set to 2, move 1 unit will become move 2 unit.

Threshold

The threshold designed for gamepad input device, the input value within this unit *WILL NOT* trigger the system response.

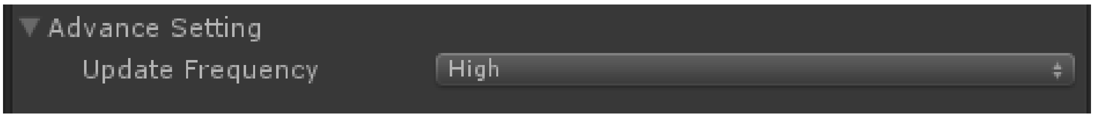
Input Script (For developer)

For using the CameraBot script you need to provide the input value to CameraBot.UpdatePosition() as following

```
void FixedUpdate (){
    CameraBot.UpdatePosition(
        Input.GetAxis("Horizontal"),
        Input.GetAxis("Vertical"),
        Input.GetAxis("Mouse X"),
        Input.GetAxis("Mouse Y"),
        Input.GetAxis("Mouse ScrollWheel"));
}
```

Advance Setting

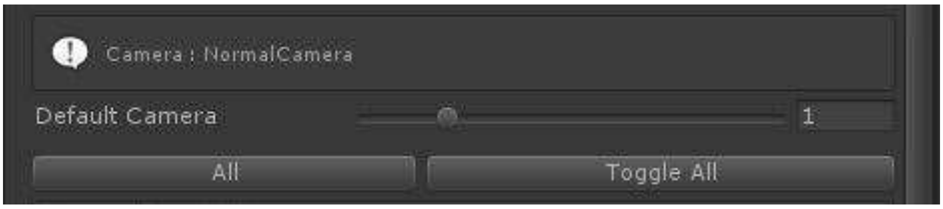
It about the performance and update frame



Update Frequency: High

Camera Preset List

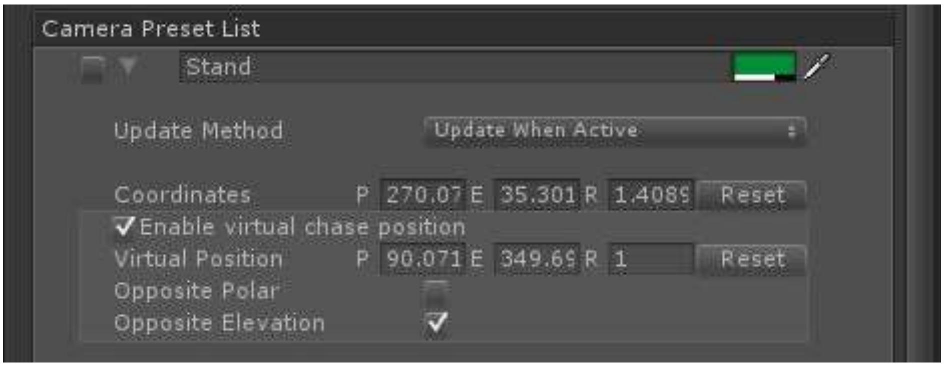
The section to preset your camera position and movement method.


Camera : NormalCamera

Default Camera: 1

Default Camera

The default camera method when the game launch.



Camera Preset List

Stand

Update Method: Update When Active

Coordinates: P 270.07 E 35.301 R 1.4089 Reset

☑ Enable virtual chase position

Virtual Position: P 90.071 E 349.69 R 1 Reset

Opposite Polar □

Opposite Elevation ☑

Camera Preset List – Elements

[ ] Debug Mode

Debug GUI, to display the camera position & angle HUD in editor scene view. (Debug)

Name

The name that you want to call for this camera method. E.g. “Stand”

You may use it for the function call in script to switch the preset camera

E.g.

```
public CameraBot CameraBot = null;
void Update()
{
    if (Input.GetKeyUp(KeyCode.Alpha1))
        CameraBot.SwitchCamera("Stand");
}
```

Color

The color only for developer easily identify the camera on the scene. (Debug)

Coordinates

The camera coordinates, is a spherical coordinates which relative to *chase target*, {P,E,R} are representing:

- P = Polar angle to chase target
- E = Elevation angle to chase target
- R = Radial distance to chase target
- Reset = reset the position to (0,0,0) the position of *chase target*.

Usually developer are not a required to set this parameters manually, in order to set this value selected the move tool (Hotkey: W) of unity3d editor to enable move handler.

Enable virtual chase position

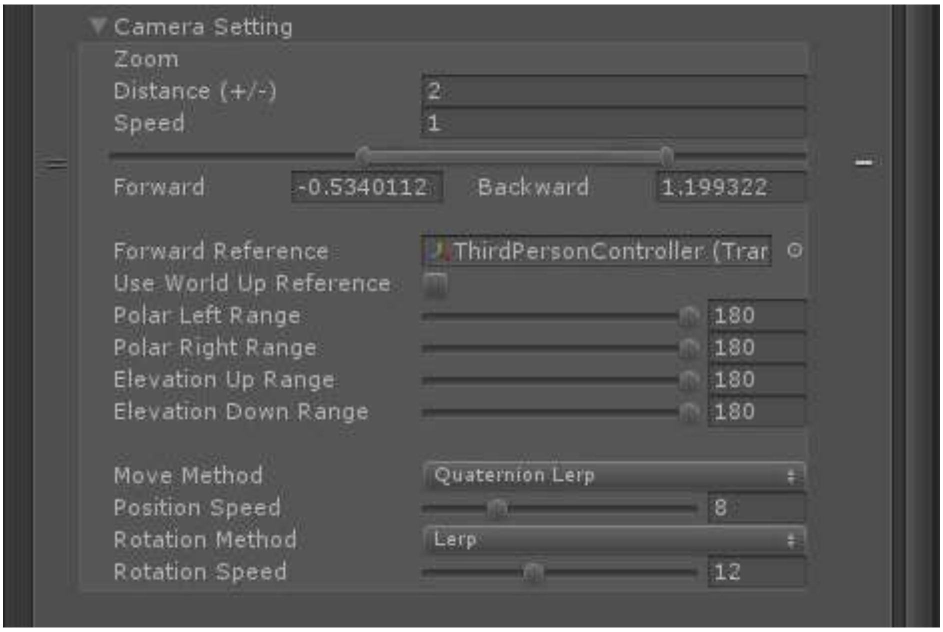
When enable, the camera will chase the virtual position relative to *chase target*, instead of facing the *chase target* directly.

Virtual position

The virtual position relative to *chase target*, is a spherical coordinates which relative to *chase target*. As same as camera "*coordinates*"

Opposite Polar & Opposite Elevation

When enable, *virtual position* will move to the opposite direction of the camera



Camera Setting

Zoom

Distance (+/-) 2

Speed 1

Forward -0.5340112 Backward 1.199322

Forward Reference ThirdPersonController (Trar

Use World Up Reference

Polar Left Range 180

Polar Right Range 180

Elevation Up Range 180

Elevation Down Range 180

Move Method Quaternion Lerp

Position Speed 8

Rotation Method Lerp

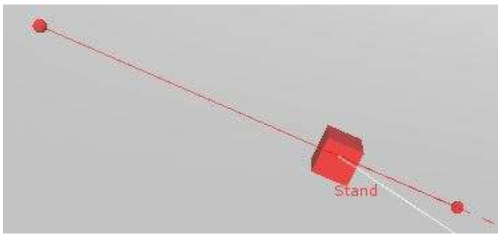
Rotation Speed 12

Camera Setting

Zoom

Distance (+/-)

Maximum zoom in & out range, when zero represent cannot zoom



Speed

Zooming speed

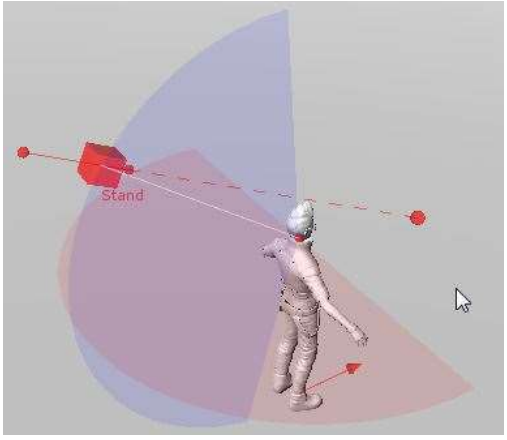
(Forward/Backward)

Relative position between Zoom distances.

Clamp Angle

Forward Reference

The forward reference for this camera calculation, leave it brank will using global forward setting



Use world up reference

When camera pitch angle pass over the chase target, which way is the upward of camera. E.g. always upward?

Polar & Elevation Ranges

The angle limitation will represent by color. (Debug)

Move & Rotation Method

The camera move method

- Snap – Snap to target position, without any acceleration and deceleration
- Lerp – linear move to target orbit position
- Quaternion Lerp – linear but movement will keep the distance to the target, which mean move by arc.

Position & Rotation Speed

The amount of Move & Rotation Method, that will translate and rotation

Support

If you have any comments, suggestion, questions or issues, please don’t hesitate to contact me. I’ll try my best to help you.

Thanks for your attention.

Canis

Asset store: https://www.assetstore.unity3d.com/#!/content/34655

Author website: http://www.clonefactor.com/

Author Email: canistk@gmail.com