Camera Pack
Thanks for purchasing this tool!

1 Description

The camera Pack includes the simple script which is very easy for using. The script allow to work the camera in several modes RTS, RPG(work in progress), MTP(move to point). Each of the modes often happens useful. There is full support of touch screens on Win tablets, mouse and keyboards. There are the most necessary parameters in the inspector allowing to quickly and easily setup the camera especially for your project.

[View Web Demo](#)

Features:
- RTS, RPG(wip), MTP cameras.
- Completely adjusted camera modes.
- Smooth movement.
- Move, rotate, tilt, zoom view.
- Rotation around target.
- Demo included.
- Full source code.

New in version 1.1:
- Multitouch zoom for unity5 and windows touch tablets/monitors
- Mouse Edge Screen Move mode (like in realtime rts game)
- Now separate axis RTS camera move inversion
- More comfortable follow mode in RPGMode (see in demo)
- RTS/MTP modes working on mobile devices
- Update in documentation
- Different RTS camera prefabs included
- Update demo scene

New in version 1.2:
- Move to object mode in RTS camera
- Y axis rotation unlocked in RTS camera
- More smoothnes in RPG camera
- Roll InOut outside the perimeter effect in RTS camera
- Speed slow damping in RTS camera

New in version 1.2.1:
- Orbit rotations in RTS camera

New in version 1.2.5:
- Bugs fixed for axis inversion, zoom, rollout effects.
- Better working on mobile.

New in version 1.3:
- Rotation around point by button event. (RTS Camera)
- Camera speed correction by altitude.(RTS Camera)
- Orbit rotation around the point (MTP Camera).
- Moving point by point (MTP Camera).
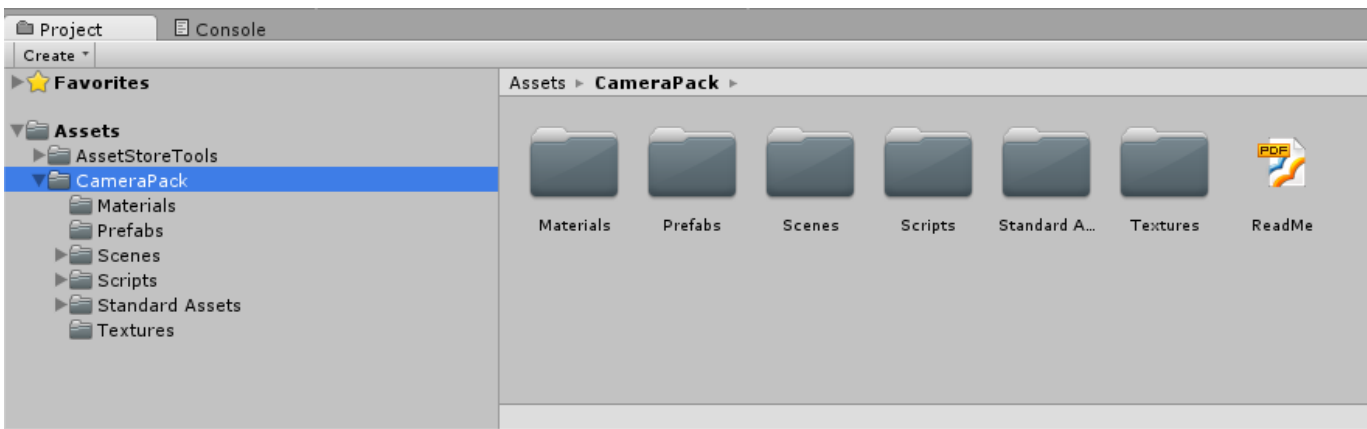- Demo scene update.(switch on uGUI)

2 Usage

2.1 Import the Unity Package "Camera Pack"

Assets > Import Package > Select **CameraPack.unitypackage**

Or double click on the **CameraPack.unitypackage** in your browser.

Or import asset from Asset Store account.

2.2 Script preparation

You can open demo scene: "**Demo.unity**" and look how does it work.

Or use **CameraMove.cs** script in own scene. Simply drag **CameraMove.cs** on the **MainCamera** object. If you add script in another object, and in this object does not have a camera component, script try to create camera component in this object in game.

2.3 Set camera mode.

Set the camera mode with **Select Camera parameter** in drop down menu. Now you can configure the camera as you need in the **Inspector window** of Main Camera object. How does the camera, you can look in the special demo

scene. In the Inspector window, when you add script to the object, you can see comments to the parameters for more information.

2.3.1 RTS_Camera
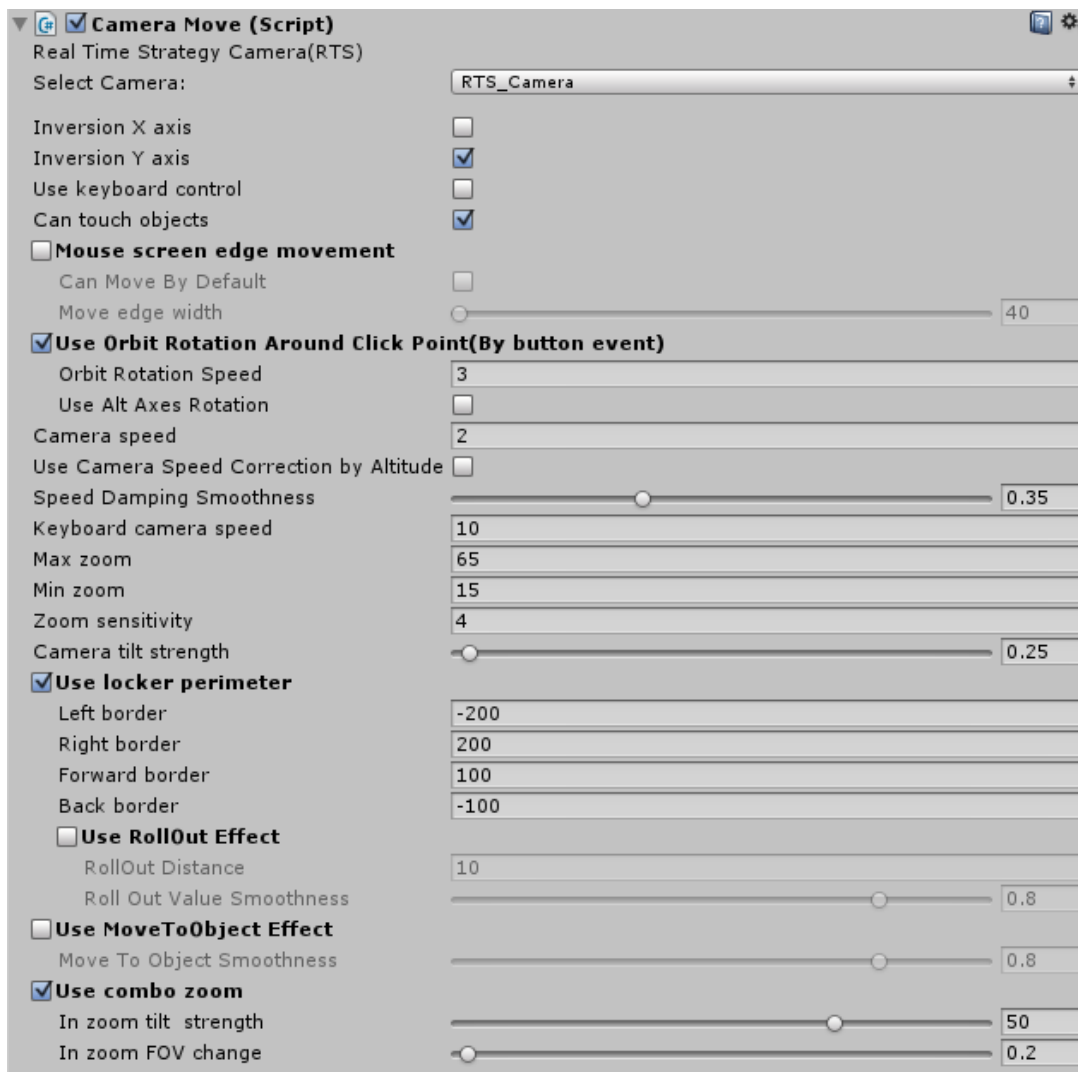
Inversion X axis - Invert moving direction by X axis.

Inversion Y axis - Invert moving direction by Y axis.

Use keyboard control - WASD for moving and QE for zooming.

Can touch objects - Activate touch event methods by touching special object with tag "Touch".

Mouse screen edge movement - Camera is moving when the mouse position on the edge of screen.

> **Can Move By Default** – additionally user can move camera by mouse/finger.

> **Move edge width –** zone width which allow to move camera

Use Orbit Rotation Around Click – you can rotate camera around point view

Orbit Rotation speed – orbit speed rotation

Use Alt Axes Rotation – camera rates around own axe

Camera Speed - Speed of the movement of the camera.

Use Camera Speed Correction by Altitude -

Speed Damping Smoothness - how fast camera is stopped.

Keyboard camera speed - Camera speed WASD keys moving.

Max zoom - Max zoom distance.

Min zoom - Min zoom distance.

Zoom sensitivity - For RTS and RPG. No zoom if Zoom Sensitivity = 0.

Camera tilt strength - Camera lateral inclination strength. 0 - without lateral inclinations.

Use locker perimeter - Use lock perimeter for camera moving.

Left/Right/Back/Forward borders - distances on which the camera from the initial point can be displaced.

Use Roll Out Effect - roll in back into perimeter.

RollOutDistance - how far the camera can move outside the perimeter.

Roll Out Value Smoothness - cameras move smoothness outside the perimeter.

Use MoveToObject Effect - camera moving to the object by user touch/click.

Move To Object Smoothness - how fast camera moving to the selected object.

Use combo zoom - Change FOV in zooming process.

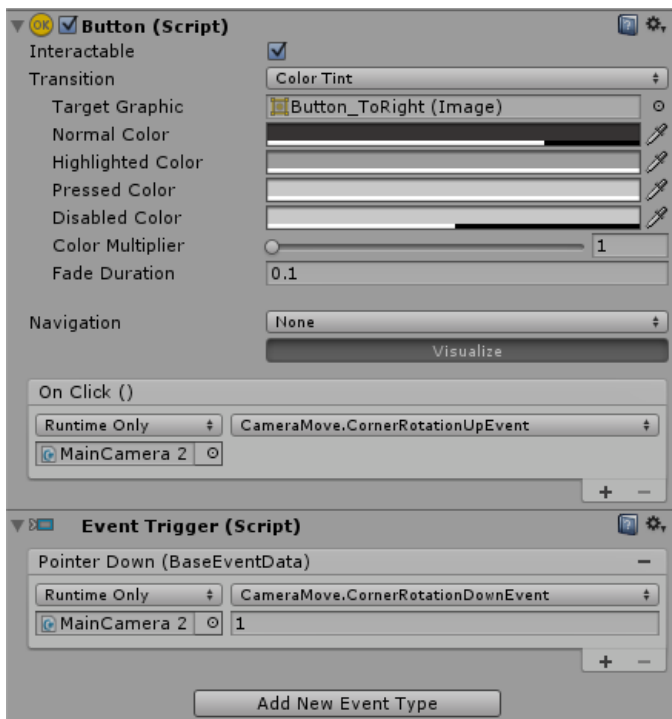**In zoom tilt strength** - Camera tilt in zooming process.

In zoom FOV change - changes FOV in zooming process.

2.3.1.1 RTS Camera - Orbit Rotation Set Up

You should create two UI buttons and set up them.

Set up button for rotation to the right:

You should add in **OnClick** event trigger the Camera gameobject and set CameraMove > **CornerRotationUpEvent**

Now add new component to the button - **EventTrigger.** Add new event type: **Pointer Down**. Add in the **Point Down** the Camera gameobject and set Camera > **CornerRotationDownEvent.** Set value = 1 – this is for right direction rotation.

Set up similar way left button, but at the end set -1 instead 1. Done.

2.3.2 RPG_Camera
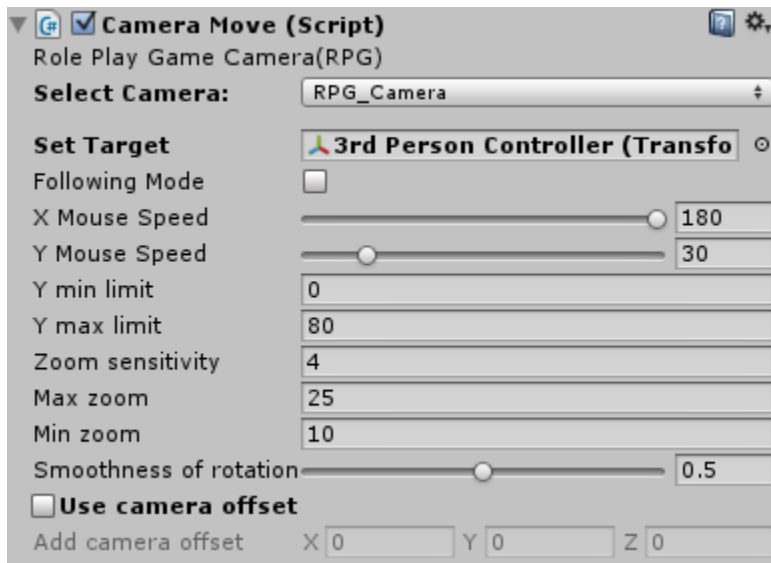
> **Attention!**
> In RPG Mode set up the **Target** transform is necessarily! In the Demo scene Target is the object with "**Player**" tag.

Set Tatget - The camera will be rotate around that object.

Following mode - The camera smooth following of the object.

X mouse speed - Camera speed on X axis.

Y mouse speed - Camera speed on y axis.

Y min limit - Camera limits on Y axis.

Y max limit - Camera limits on Y axis.

Zoom sensitivity - For RTS and RPG modes. No zoom if Zoom Sensitivity = 0.

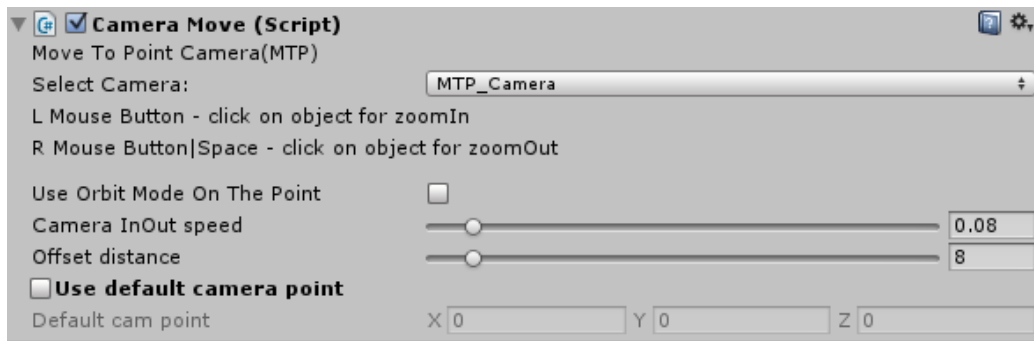**Max zoom** - Max zoom distance.

Min zoom - Max zoom distance.

Smoothness of rotation - Smoothness of turn of the camera

Use Camera offset - Use Stable offset for camera position.

    Add camera offset - Stable offset for camera position.

2.3.3 MTP_Camera

Use Orbit Mode On The Point -

Camera InOut speed - Smoothness of the movement of the camera.

Offset Distance - Offset distance.

Use Default Camera Point - Use own start camera point.

> **Default Cam Point** - set your initial camera world position.

2.4 Main scripts in pack

CameraMove.cs

In the script, there are some several methods for example, like that:

void **CheckClickPoint_RTS**(vector3 vect)

void **CheckClickPoint_MTP**(vector3 vect)

This methods using for make action on click at the objects event. In the demo scene objects marked with special "Tag": "**TouchPerson**" and "**Touch**".

> **Attention!**
> Create and Use "**TouchPerson**" or "**Touch**" tags for RTS_Camera(Move to object mode) and MTP_Camera

BaseGUI.cs

Special UI script for demo scene.

---

3 Additional Info

Contact details below:

vadakuma@gmail.com

Vadakuma 2017.