Safire 2D Camera Documentation

version

TwoBitMachines

August 18, 2022

Contents

Safire 2D Camera Documentation

Introduction

If reading the auto generated pdf version, go to https://twobitmachines.github.io/Safire2DCameraDocs/ for a more readable experience. If you have any questions or feedback, please contact us at TwoBitMachinesDev@gmail.com.

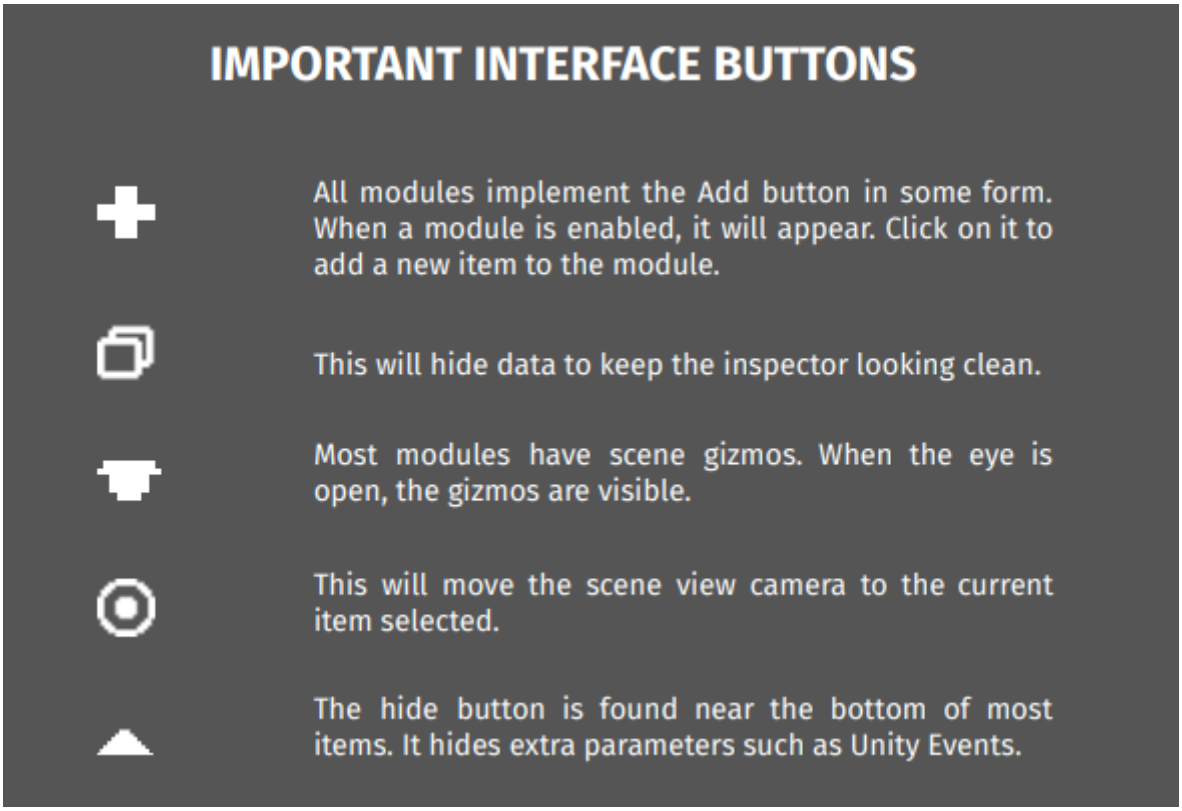
Welcome! This is the documentation for Safire 2D Camera This tool contains modules that are fundamental to any 2D camera system. It has a friendly interface that makes it fast and easy to setup these systems. Simply put the **SafireCamera** component on the camera gameobject to get started.

Next, choose the target to follow. This is typically the player, and you will need to set the Transform reference. If you choose User, the system is expecting the camera to be controlled with the pan, zoom, and rotate modules. If the camera is a perspective camera, choose the zooming mode. Either zoom by FOV or by distance.

Once setup, you will need to add modules. Click the “Module+” button to open the module menu. Select what you need, then press the left button to enable the module. If you want to remove a module, click the close or minus button on the module bar. The Follow and Pixel Perfect module will always be visible by default.

Important

The camera will execute in Late Update only.



Important Methods

Method	
ResetAll()	Call this when there is a game reset or when the player changes position abruptly, like a teleportation. If using Flare Engine, this will be called automatically on a game reset.

ModuleEnable(string moduleName)	Enable the specified Module.
ModuleDisable(string moduleName)	Disable the specified Module.
PauseFollowMechanics (bool enable)	This pauses all the modules that modify the camera's position in a meaningful way but leaves modules like shake, parallax, etc, still active if enabled. This can be useful in case another system needs to take control of the camera.

Follow

Implement standard camera mechanics for following the target. Mix and match these properties where it makes sense to do so.

Property	
Speed	How fast the camera should follow the target.
Smooth	The camera will smooth towards the target. A value of zero halts the camera. A value of one instantly moves the camera to the target's position.
Offset	Offset the target's follow position in the x or y axis.
Dead Zone	The area surrounding the target in which no target movement is detected by the camera.
Screen Zone	Prevent the target from escaping the camera view. This occurs when the target is moving too fast or when the camera is moving too slow.
Auto Scroll	Scroll the camera with the specified speed. Setting any axis here will override the other mechanics in that axis
Detect Walls	If None is enabled, the system will ignore this setting. If Ignore Gravity is enabled, the camera will not follow the target when it's jumping. If Detect Walls is enabled, the camera will only follow the target if it's touching a surface. Specify the layer that represents hard surfaces in your world. This mechanic assumes the target contains a BoxCollider2D. Enable this by method. 0 == Node, 1 == IgnoreGravity, 2 == DetectWalls.
Push Zone	Force the camera to follow the target only in the desired direction. For example, if Push Zone X is set to Push Left, the camera will ignore the target if it's moving to the right. The value represents where in the screen the camera will begin to follow the target. Do note, the camera will ease into the push zone to prevent abrupt target changes.

Method	
ChangeTargetTransform (Transform newTransform)	
CenterCameraOnTarget ( )	
SetCameraPosition (Vector3 position)	
SetFollowSpeed (float speed)	
SetFollowSmooth (Vector2 smooth)	
SetAutoScrollSpeed (Vector2 speed)	
PauseAutoScroll (bool enable)	
DetectWallsSet (int key)	

Speed Zoom

The camera will zoom based on the target's speed. Add as many threshold speeds as necessary. Arrange them from the lowest threshold to the highest threshold.

Property	

Threshold	The speed the target must exceed or equal to trigger a zoom.
Zoom	The camera zoom level.
Speed	How quickly the camera should zoom.
Axis	The axis in which to detect the target's speed.

Look Ahead

The camera looks ahead in the direction of target movement. This requires the Follow module to be enabled.

Property	
Normal	The camera is always looking ahead.
Recenter	The camera will stop looking ahead and revert once the target stops moving.
Incremental	The camera will continue to look ahead only if the target is moving. If the target is not moving, the camera will retain its position.
Threshold	The camera will only look ahead if the target has reached the speed threshold.
Mouse	The look ahead direction will depend on the position of the mouse and not the target's direction.
Shape	If Square is enabled, the look ahead boundaries will be rectangular, defined by the Horizontal and Vertical values. If Circle is enabled, the look ahead boundary will be a circle defined by the Radius value.
Smooth	How quickly the camera should look ahead.

Rooms

Section your game level with rooms and let the target traverse them. When the target enters a room, it causes the camera to transition into the room. The room boundaries will then clamp the camera's position.

Property	
Name	The name of the room for identification purposes (optional).
Boundary	Each room has four boundaries, corresponding to each room wall (left, right, top, bottom). If enabled, the camera will be clamped by the corresponding boundary. If disabled, the camera will freely move through the boundary. The last button is to exit the room. If enabled, when the target exits the room, the camera will remain clamped to the room. Otherwise, the camera will exit the room.
Transition	Once the target enters a room, it will trigger a room transition causing the camera to automatically move inside the room. Specify the duration of the transition and if there should be a delay before starting the transition.
Tween	Transition tween type.
Zoom	Zoom to the desired level during the transition period. There are three types. Zoom: specify the zoom level. Match height: zoom to match the room height. Match width: zoom to match the room width.
Restrict	The target will not be able to physically move outside the room boundaries.
Hold Target	If enabled, during a transition the system will prevent the target from moving.

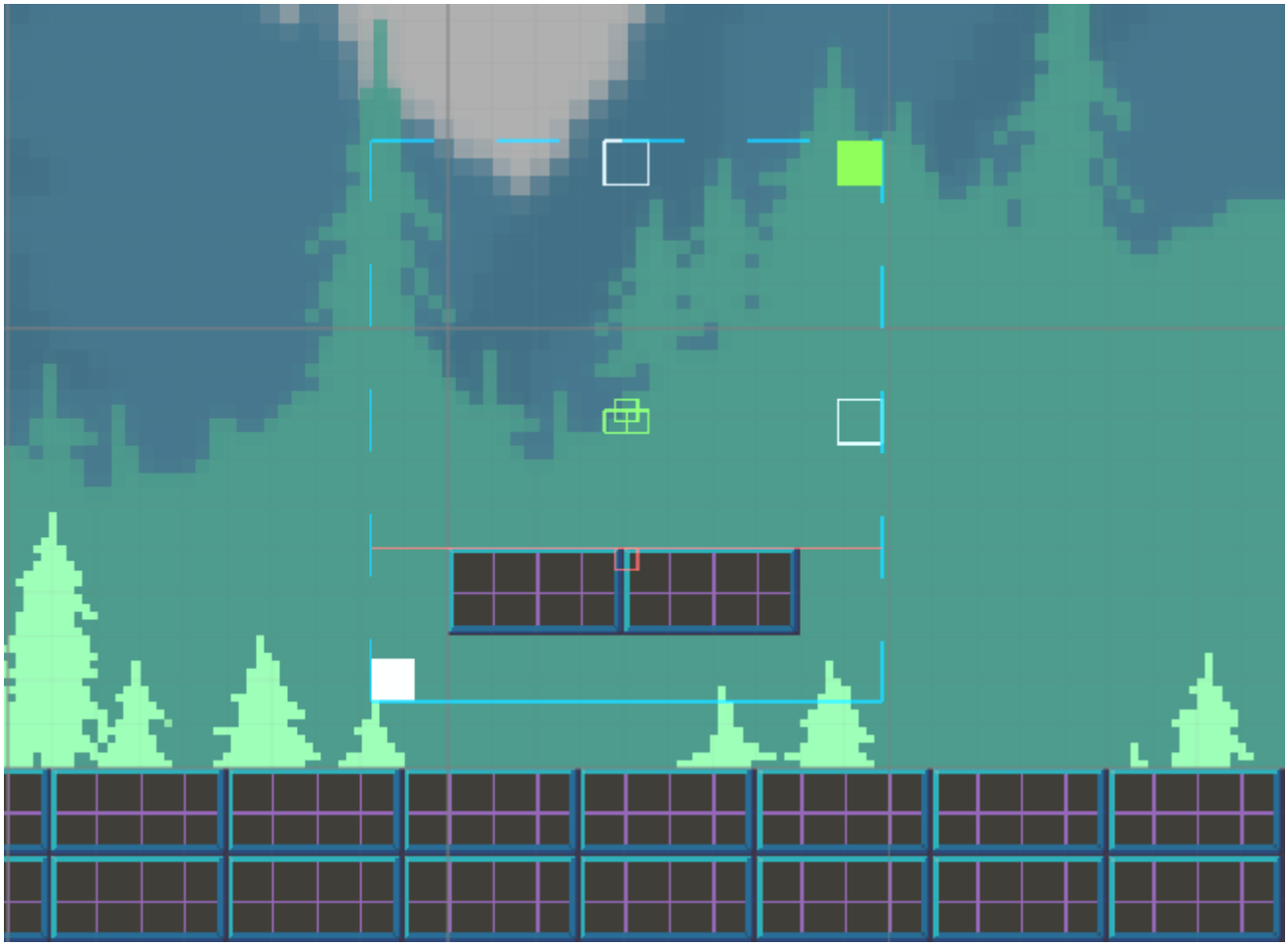
Method	
RoomRestrict (string roomName)	Enable a room's restriction option.
RoomUnrestrict (string roomName)	Disable a room's restriction option.
RoomAddMultipleTarget (string roomName, Transform transform)	Add a transform to the multiple target's list.

RoomRemoveMultipleTarget (string roomName, Transform transform)	Remove a transform from the multiple target's list.

Note

While editing the room in the scene, pressing the upper right corner will resize the room to the size of the current orthographic camera. Make sure the main camera has the tag MainCamera.

The room is extremely versatile. It is possible to overlap two rooms adjacent to each other to facilitate transitions so the target doesn't go off screen. You can also modify the trigger area of the room. The trigger area detects the target, which in turn triggers a room transition. By default, it's the same size as the room. To modify the trigger area (the red line), drag the small buttons found in the middle of the room. You should only modify these values during editor time, and as a reminder, the trigger area is only there to detect the target, not to clamp the camera's position.



This becomes useful for creating room platforms. In the image above, the room boundaries are all disabled except for the bottom one. And the bottom trigger has been moved up. The benefit is that the only way to enter this room is by jumping on top of the platform. If the target hits the bottom of the platform, it will not cause an unnecessary camera transition.

Multiple Targets

Each room comes with the ability to follow multiple targets including the main target. The camera will try to keep the specified targets in view by moving towards the average center and by zooming (if enabled). This feature is useful in case the main target is interacting with dynamic game elements that need to stay inside the camera view. This requires the Follow module to be enabled.

Property	

Early Detect	When a target is near the camera edge, the camera will begin to zoom. Change the size of this detection area to detect targets sooner. A value between 1 and 5 should be a good starting point.
Zoom To Fit	If enabled, the camera will zoom to encapsulate moving targets and keep them inside the screen. The zoom level will not grow bigger than the room.
Target Transforms	Specify the targets and their positional offset for the system to keep track of.

Cinematics

Create cinematic sequences to highlight special moments in your game.

Property	
Name	Name the cinematic for identification (optional).
Trigger	If Trigger Once is enabled, the cinematic can only be activated once by the target. If Trigger is enabled, the cinematic can be activated any number of times. If By Script is enabled, the cinematic can only be activated by a method call.
LetterBox	Set the reference for the letterbox that will be activated to give the sequence a more cinematic feel. This prefab can be found in the Assets/Prefab folder. Drag and drop it into the scene and set the reference.
On Begin	The Unity Event invoked when the cinematic begins.
On Complete	The Unity Event invoked when the cinematic completes.

Targets	
Position	The position the camera will move to in sequence. This value can be modified in the scene.
Duration	The time it takes for the camera to get to the target's position.
Wait	Once the camera arrives at its target, specify how long it waits there. If the toggle is enabled, the wait time is indefinite. To exit out of this state, call the following method: CinematicNextTarget();
Zoom	Zoom the camera to a specified zoom level.
On Arrival	The Unity Event invoked when the camera arrives at the target.

Method	
CinematicTrigger (string cinematicName)	Trigger this cinematic.
CinematicNextTarget ();	Move to the next cinematic target if the wait time at the current one is indefinite.

LetterBox

The letterbox will activate black bars that will tween into the camera view during a cinematic. The tweening values can be modified in the Let's Wiggle components attached to this gameobject. This is a tweening library that can be configured in the inspector. Simply change the time and tween settings for each MoveToY child tween.

Rails

The camera moves in a segmented path as it follows the target. A segmented path consists of targets that you must place in the scene. The first and last targets will be the triggers. Once the target enters a trigger, the camera begins to follow the rail path. This requires the Follow module to be enabled.

Property	
Name	The name of the rail for identification purposes (optional).

Horizontal	The camera follows the target's horizontal position. This rail shouldn't have vertical segments.
Vertical	The camera follows the target's vertical position. This rail shouldn't have horizontal segments.
Auto	The camera completely ignores the target's position and follows the segmented path at a specified speed. The automatic rail contains a Unity Event that is invoked when it completes the path.
RailPause (bool value)	Pause or unpause an active auto rail.

Shakes

This is an easy to use shake creator. The idea is to create shakes with predefined settings so they can be activated at runtime by name. Each shake can be called by a method or by using the shake trigger. Each shake that is created is actually linked to a Scriptable Object, which means you don't have to keep recreating the same shakes for each scene. As long as the Scriptable Object is referenced, the shakes will be remembered. In fact, the module won't work without this Scriptable Object. To create one, right click Create/FlareEngine/ShakesSO.

Any shake implemented will run simultaneously with all other shakes. The system will pool and recycle shakes to keep things nice an efficient.

Press the play button to test any shake while in Editor Mode.

Property	
Shake SO	The reference to the Scriptable Object that will save the list of shakes.
Name	Name the shake for identification purposes.
TimeScale	If the shake uses time-scale, the shake can be affected by slow motion.
Random	The camera abruptly shakes according to a random pattern.
Perlin	Shake the camera with the perlin noise algorithm, resulting in a smoother and less chaotic shake.
Sine	Shake the camera with a sinusoidal algorithm.
One shot	The camera shakes in a single direction. This is a recoil. This shake will need to be called by script since the shake direction can't be known ahead of time.
Is Single Shake	These shakes are meant for high volumes short duration shakes such as bullet impacts. These shakes are grouped together and handled in a single state. The advantage is that these shakes are not added into the shake pool.
Amplitude	This is the camera displacement in the horizontal and vertical axis. **The value Z is the angle displacement.**
Speed	How fast the shake is implemented. Typically, a lower value is smoother, while a higher value if sharper.
Strength	Shake intensity.
Duration	How long the shake will last. The toggle button, if enabled, will make the shake last indefinitely. This is a constant shake and it must be turned off manually. Only one constant shake can run at a time.

Method	
Shake()	A convenient method for calling a normal sized shake.
ShakeSmall()	A convenient method for calling a small shake.
Shake(string shakeName)	Call the specified shake.
ShakeType(…)	Create a shake through script. Call any of the different types.
TurnOffConstantShake ( )	

TurnOffAllShakes ( )	

Parallax

Create the illusion of depth by using several backgrounds to move with the camera. The parallax will scroll infinitely, even with a perspective camera.

Each background transform must contain a single sprite that is set to pivot center. Also, don't forget to set the layer order manually for each sprite in order to render the background images in correct order. At runtime these backgrounds will be duplicated to achieve infinite parallax scrolling. Click the add button to add as many backgrounds as necessary.

Note

If using a perspective camera, parallax comes for free! But you can still use this module to get infinite scrolling.

Property	
Transform Layer	The Transform that will scroll with the camera.
Parallax Rate	The rate at which the transform scrolls with the camera. A value approaching 1 is for the farthest layers. A value approaching 0 is for the layers nearest to the player. Values less than -1 are for foregrounds layers.
Auto Scroll	If this value is greater than zero, the transform will automatically move on its own. This could be a layer that contains clouds. Try many parallax rates until you get one that gives a correct motion with the auto scroll.

Warning

Infinite scrolling limitations: the camera size must always be smaller than the size of the image. If it isn't, there will be noticeable gaps in each layer during scrolling. Things that can cause this are zooming the camera to a large value, a perspective camera that is too far away from the image layer, or a camera size that is simply too large. These issues may present themselves concurrently if not careful. Choose zoom levels wisely.

Peek

This allows the user to quickly peek the camera in the desired direction. This is similar to look ahead, but with this module, the direction of the camera is controlled by the user and not by the target's direction.

Property	
Distance	The peek distance.
Easing Time	The time it takes to reach the maximum peek distance.
Ignore Clamps	If enabled, peek will ignore room clamps and the world clamp. Otherwise, it is clamped. Be aware that sometimes the peek effect won't be noticeable if the camera itself is being clamped if this option is disabled.
Direction	If any of the buttons are pressed, the camera will peek in the desired direction. Set to None if not required.

Method	Call these methods continuously to peek in the respective direction.
PeekUp()	
PeekDown()	
PeekLeft()	

PeekRight()	
PeekDirection(Vector2 direction)	

Follow Blocks

Create and arrange a series of follow blocks to guide the camera in horizontal or vertical pathways. Once the target enters one of these blocks, the camera is locked and can only move in that axis. This is helpful if the game level requires only certain pathways for the camera to travel in. Typically you put a horizontal block next to a vertical block and so on.

Property	
Horizontal	If the target enters this block, the camera can only move horizontally as it follows the target.
Vertical	If the target enters this block, the camera can only move vertically as it follows the target.
Exit	This is the exit. When the target enters this block, the camera returns to normal operation.

Highlight Target

On occasion there is a need to highlight an important game element by having the camera follow it. This element is not the main target. It is also possible to follow the new target indefinitely and allow a certain range in which the main target can still influence the camera.

Use this module to create predefined targets that can be triggered by some event and have the camera follow them. This requires the Follow module to be enabled.

Property	
Target	The target can either be a transform or a position. Specify the follow speed.
Duration	The total time the camera will follow the target. If the toggle is enabled, the camera will follow the target indefinitely, or else it will revert to following the main target automatically.
Main Target Range	If the camera is following the target indefinitely, this value will specify the range in which the main target can still influence camera movement.
Ignore Clamps	If enabled, the camera will be able to move through rooms while following the target.

Method	
HighlightTarget (string name)	
HighlightTargetTerminate ( )	
HighlightTarget (…)	If the highlight target hasn't been created in the inspector, you can still create one by code.

Zoom Trigger

Once the target enters this trigger, the camera will begin to zoom.

Property	
Maintain	If the target exits the trigger, the zoom level will be maintained until changed by another module.
Revert On Exit	The zoom level will return to the original zoom level when the target exits the trigger.
Scale	The zoom level.
Smooth	The smoothing effect applied to the zoom.
On Enter	The Unity Event invoked when the target enters the trigger.

On Exit	The Unity Event invoked when the target exits the trigger.

Method	
Zoom (float scale, float duration)	
ZoomReset ( )	This will immediately reset the zoom to 1.
ZoomSetToOne ( )	This will zoom the camera to 1, taking 1 second.

Slow Motion Trigger

Once the target enters this trigger, time scale will be slowed down.

Property	
Maintain	If the target exits the trigger, the slow motion effect will remain.
Revert On Exit	The slow motion effect will be turned off when the target exits the trigger.
Intensity	The value Time.timeScale will be set to.
On Enter	The Unity Event invoked when the target enters the trigger.
On Exit	The Unity Event invoked when the target exits the trigger.

Method	
SlowMotion (float intensity, float duration, bool constant = false)	
TurnOffSlowMotion ()	

Basic Trigger

The basic trigger can be used for any miscellaneous events that require the OnEnter() and OnExit() Unity Events.

Property	
Trigger	Set to Trigger Once if you want this trigger to only activate once. With Trigger, the target can activate the trigger any number of times.
On Enter	The Unity Event invoked when the target enters the trigger.
On Exit	The Unity Event invoked when the target exits the trigger.

Regions

When the target enters a region, it will cause the camera to move towards the center of the region. The farther the target moves inside the region, the less influence it has on the camera. This requires the Follow module to be enabled.

Property	
Position	The region’s center can be a position in world space or a transform’s position.
Outer Radius	If the target is inside the outer region, both target and region will have influence over the camera.
Inner Radius	If the target is inside the inner region, the region will have complete control over the camera.
Zoom	Zoom the camera as it moves closer to the region center with a specified smooth value. If no zoom is desired, set to zero.
Smooth	The smoothing effect applied to the zoom.
On Enter	The Unity Event invoked when the target enters the region.
On Exit	The Unity Event invoked when the target exits the region.

World Clamp

The world clamp is a rectangle that encapsulates the entire game level and prevents the camera from moving outside of it. This is similar to rooms. However, while rooms can overlap, they can't exist inside each other. This gets around that limitation.

Property	
Destination	The transform where the gameobject will be teleported to.
LayerMask	Only gameobjects on this layer mask will be teleported.
Delay	Add a time delay before teleporting.
On Delay Start	The event invoked when the time delay begins. This only executes if delay is greater than zero.
On Teleport	The event invoked when a gameobject is teleported.

User

Instead of having the camera follow the target, the camera is controlled directly by the user using touch, the keyboard, and mouse input. Switch the follow type from Target to User.

When the follow type is User, the system is using the center of the camera as the target. Thus, this mode isn't compatible with the following modules: Dead Zone, Look Ahead, Follow Blocks, Multiple Targets, Regions, Screen Zone, Push Zone, and Peek.

Pan

This controls the camera movement. Use Touch, Mouse, or Keyboard input to move the camera.

If a touch or mouse position is at the edge of the camera, Pan Edge will move the camera automatically in that direction. Choose how big the area will be to detect a touch or mouse position near the edge of the camera. The Left, Right, Top, Bottom values are percent values. A value of one will result in a distance that is half the width of the camera. A value of zero will result in no edge detection. Use the Scale value to damp or boost the pan edge speed.

Zoom

This controls the camera zoom level. Zoom with Touch, Mouse, or Keyboard. The Range will determine the minimum and maximum zoom levels. Speed and Smooth will control the rate of the zoom.

Rotate

This controls the camera rotation. Rotate with Touch, Mouse, or Keyboard. Speed and Smooth will control the rate of rotation.

Method	
PauseUser (bool pausePan, bool pauseZoom, bool pauseRotate)	Pause a user module.
UserPanKeyboard (KeyCode left, KeyCode right, KeyCode up, KeyCode down)	Change the user keyboard settings for pan movement.

UserPanMouse (Pan panType, MouseButton mouseButton, MouseButton holdButton)	Change the user mouse settings for pan movement.
UserZoomKeyboard (KeyCode zoomInKey, KeyCode zoomOutKey)	Change the user keyboard settings for zooming.
UserRotateKeyboard (KeyCode rotateLeftKey, KeyCode rotateRightKey)	Change the user keyboard settings for camera rotation.

Pixel Perfect

Event though Unity already has a Pixel Perfect Camera, this module will be kept for the completeness of the engine.

This module is intended for use with orthographic cameras. Before starting your game, choose the resolution the game will be designed for. By default it's set to 320x180 since it can multiply nicely into common device resolutions. This will determine how much of the game world the user can view as determined by the PPU of your game art; changing this resolution later might change the look and feel of the game, so choose wisely.

The camera will execute in Edit Mode and create black bars automatically for any screen resolution that is not pixel perfect.

It is recommended to use a separate camera for UI. Add the PixelPerfectUI component to the UI camera. Make sure the UI camera is only rendering the UI layer. On the flip side, the main camera should not render the UI layer. Match the PPU and resolution settings. On any canvas in your game world, set the render mode to Screen Space - Camera and set the UI camera as a reference. Set Canvas Scaler to scale with screen size and set the target resolution. Set the UI camera depth to a larger value than the main camera so it renders on top of it. And finally, set the clear flags to Don't Clear.

Typically you shouldn't zoom with a pixel perfect camera as it will break pixel perfectness. But if you do, zoom by integer multiples if possible to retain a pixel perfect screen.

All your sprites should strive to have the same PPU settings and bet set to Point filter with no compression.

Property	
PPU	This is the PPU (Pixel Per Unit) of the game art. How many pixels fit in one world unit of the game? The Wrench Button, if clicked, will snap all scene objects to the PPU grid. The color will change the color of the black bars.
Resolution	The target size of your game world.