

**SimpleSelect
Documentation**

Contents

1 Introduction	3
1.1 Adding SimpleSelect to a Project.....	3
1.2 Using SimpleSelect.....	3
2 SimpleSelection Settings	5
3 Simple Camera Mover	8
4 Access SimpleSelect by Script.....	10

1 Introduction

SimpleSelect is a simple to use object selection script package for the **Unity Game Development Tool**. It adds a powerful, fast and customizable selection tool to games created with Unity.

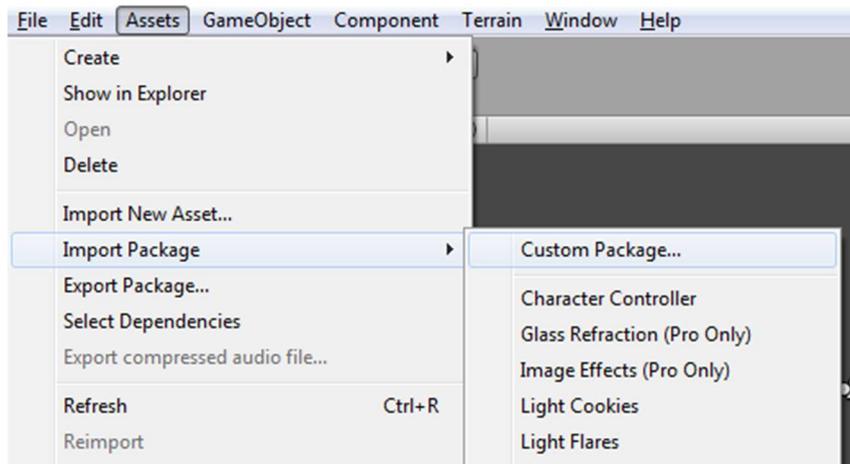
SimpleSelect is available in **C#** and supports **mouse and touch input**, spawning a selection cursor (prefab), **single/multi** unit selection, **drag selection**, etc.

SimpleSelect is compatible with **Unity 5.4** and officially supports publishing to the **Web Player, PC and Mac Standalone, Android** and **iOS** platforms. Although it might run on other platforms (e.g. Xbox360, PlayStation 3, Wii), they are yet untested and not officially supported.

1.1 Adding SimpleSelect to a Project

To add **SimpleSelect** to a **Unity** project, you have to import the **simpleSelect_cs.unitypackage**.

You can do this by using the Assets menu:

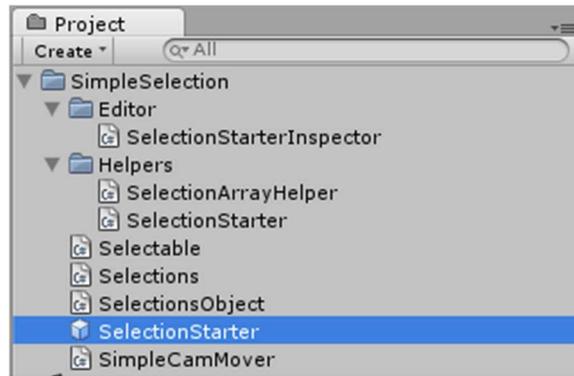


Assets > Import Package > Custom Package...

Select the **simpleSelect_cs.unitypackage** in the file manager and import it into your project. After **Unity** finished compiling the script files you can start using **SimpleSelect**.

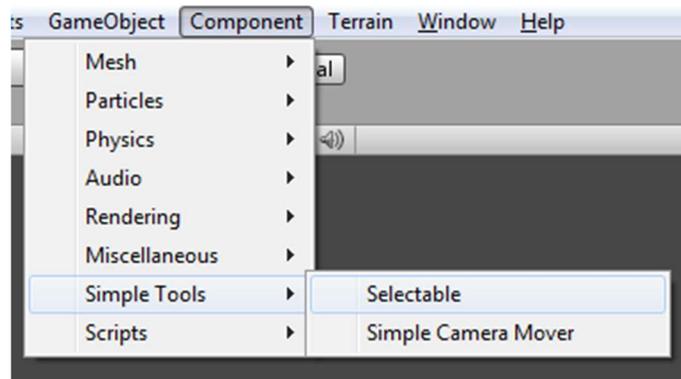
1.2 Using SimpleSelect

Now that you have **SimpleSelect** imported into your project, all you have to do is add the **SelectionStarter** prefab to the first scene in your game (or the first scene you want to use **SimpleSelect**).



To add the prefab to your scene, simply drag it from the project view into the scene view. You can change various settings on the **SelectionStarter** prefab in your scene – read more about the settings in **chapter 2**.

If you want an object in your scene to be selectable, you need to add the **Selectable** component to it. To do this, select the object in the scene view or project hierarchy and add the component by using the *Component* menu:



Component > Simple Tools > Selectable

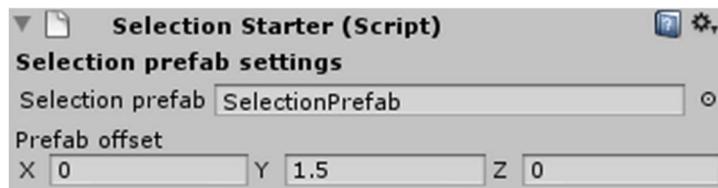
Your object also needs a **Collider** component attached (e.g. box collider, sphere collider, capsule collider, etc.) to support selection by clicking on the object (drag selection doesn't require a collider).

2 SimpleSelection Settings

SimpleSelection enables you to customize settings to your needs. The settings are made at the **Selection Starter** prefab *Inspector* view in the scene you added it to.

Selection prefab settings

You can use a prefab to mark selected objects in your scene. The prefab will automatically be spawned on your objects when they are selected, and destroyed when the selection is removed.



Selection prefab

The prefab that will be used to mark selected objects. Drag a prefab (game object) from your project hierarchy onto the object field.

Prefab offset

The offset the prefab spawns when added to the selected objects in the scene. E.g. Y=3 will spawn the object 3 world units above the selected object.

Use Pooling

The prefab instances will be pooled, i.e. they'll be disabled instead of destroyed and reused instead of spawning a new instance.

Multi selection settings

You can use multi selection and define the key that needs to be held down for multi selection. Multi selection allows selecting multiple objects by clicking on them (or dragging a selection box over them) and toggle selected objects.



Auto multi selection

Multi selection is automatically enabled, you don't need to hold down a key.

Multi key

The key used to enable multi selection.

Is button

The multi key specifies the name of a Unity input axis. If not selected, the multi key will be treated as key (e.g. 's', 'left shift', ...).

Mouse/touch settings

Set how to control selections here.

Mouse button

The mouse button used for clicking/dragging. Use 0 for left mouse button, 1 for right mouse button, 2 for middle mouse button.

Touch count

The number of fingers (touches) used for clicking/dragging. Use 1 for one finger, 2 for two fingers, etc.

Drag selection settings

Drag selection enables you to drag the mouse in the game and select objects with a selection box.

Use drag

Drag selection is enabled.

Minimum drag distance

The minimum distance the mouse/touch has to move to be recognized as a drag.

Immediate select

The objects in the selection box will be selected immediately. If not selected, the objects will be selected once the mouse/touch was released.

Use color

A color is used for the selection box. If not selected, a texture will be used.

Color

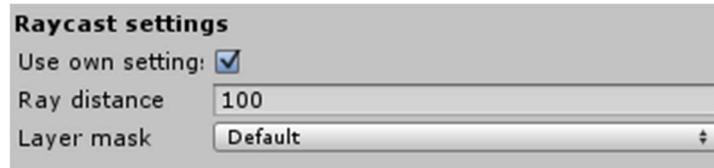
The color to be used for the selection box.

Texture

The texture to be used for the selection box. Drag a texture from the project hierarchy onto the texture field.

Raycast settings

SimpleSelect uses a raycast to select objects by clicking/touching. If you need special settings for raycasting, you can set them here.



Raycast settings

Use own setting:

Ray distance: 100

Layer mask: Default

Use own settings

Use your own raycast settings. If not selected, a simple raycast without distance is used and will collide with all layers.

Ray distance

The distance the raycast uses. If nothing is hit within this distance, the ray stops.

Layer mask

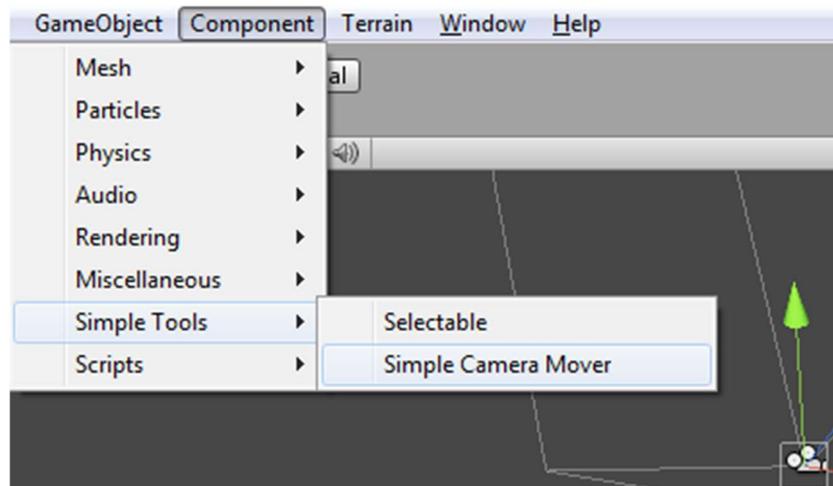
The layer mask used for raycasting. Select the layer you want the raycast to collide with.

3 Simple Camera Mover

SimpleSelect also includes a camera control script for mouse and touch input. It's designed for top-down view and enables you to move the camera by either moving the mouse to one of the borders or by dragging the view around using touch.

The **Simple Camera Mover** also enables you to set a limit for the camera movement along the X and Z axis.

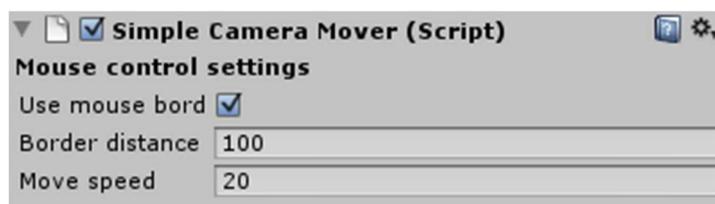
You can use the **Simple Camera Mover** by adding it to the camera in your scene with the *Component* menu:



Component > Simple Tools > Simple Camera Mover

Mouse control settings

Settings for controlling the camera by mouse. The camera will move into the direction you move the mouse near the border.



Use mouse border

Mouse control is enabled.

Border distance

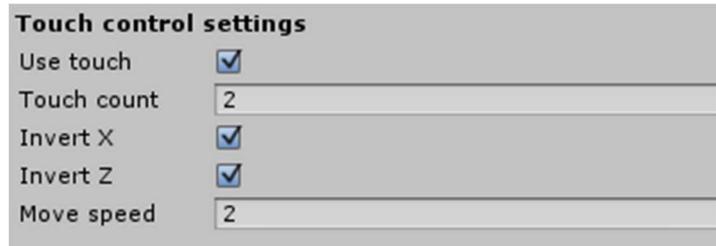
The distance to the screen borders which will start the camera move. The move will start slow and be at full speed when the mouse is at the border of the screen.

Move speed

The full move speed (per second).

Touch control settings

Settings for controlling the camera by touch. The camera will move when you drag the scene around by moving the touch.



The screenshot shows a dialog box titled "Touch control settings" with the following options:

Use touch	<input checked="" type="checkbox"/>
Touch count	<input type="text" value="2"/>
Invert X	<input checked="" type="checkbox"/>
Invert Z	<input checked="" type="checkbox"/>
Move speed	<input type="text" value="2"/>

Use touch

Touch control is enabled.

Touch count

The number of fingers (touches) used to trigger touch control.

Invert X

The movement along the X axis will be inverted.

Invert Z

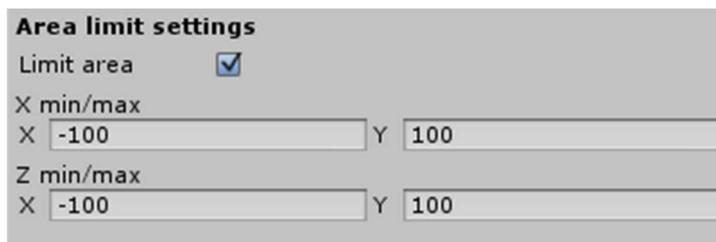
The movement along the Z axis will be inverted.

Move speed

The speed the touch will move the camera at.

Area limit settings

You can set the camera to stop moving into a direction once a certain position is reached.



The screenshot shows a dialog box titled "Area limit settings" with the following options:

Limit area	<input checked="" type="checkbox"/>
X min/max	X <input type="text" value="-100"/> Y <input type="text" value="100"/>
Z min/max	X <input type="text" value="-100"/> Y <input type="text" value="100"/>

Limit area

The camera will stop at the set limits.

X min/max

The minimum and maximum position along the X axis of the scene. X defines the minimum, Y the maximum.

Z min/max

The minimum and maximum position along the Z axis of the scene. X defines the minimum, Y the maximum.

4 Access SimpleSelect by Script

SimpleSelect only handles selection of objects, but it doesn't do anything on its own with them – to make use of it, you need to be able to access the selected objects by script.

This is done by using the **Selections** class – it offers static functions to access selected objects easily.

Since all source code of **SimpleSelect** is in the **SimpleSelect** namespace, you need to include the namespace in your code by putting this at the top of your scripts:

using SimpleSelect;

Selections.GetAllSceneSelectable()

Returns an array of all **Selectable** components in the scene.

Selections.GetAllSceneSelectable(ref List<Selectable> list)

Adds all **Selectable** components in the scene to the list.

Selections.GetAllSceneObjects()

Returns an array of all **GameObjects** with a **Selectable** component attached.

Selections.GetAllSceneObjects(ref List<GameObject> list)

Adds all selectable **GameObjects** in the scene to the list.

Selections.GetAllSceneTransforms()

Returns an array of all **Transforms** with a **Selectable** component attached.

Selections.GetAllSceneTransforms(ref List<Transform> list)

Adds all selectable **Transforms** in the scene to the list.

Selections.GetSelectableCount()

Returns an integer number representing the number of **Selectable** components in the scene.

Selections.ClearSelection()

All selections are removed. No selectable object is selected any more.

Selections.GetSelected()

Returns an array of all currently selected **Selectable** components.

Selections.GetSelected(ref List<Selectable> list)

Adds all selected **Selectable** components to the list.

Selections.GetSelectedObjects()

Returns an array of all currently selected **GameObjects** with a **Selectable** component attached.

Selections.GetSelectedObjects(ref List<GameObject> list)

Adds all selected **GameObjects** to the list.

Selections.GetSelectedTransforms()

Returns an array of all currently selected **Transforms** with a **Selectable** component attached.

Selections.GetSelectedTransforms(ref List<Transform> list)

Adds all selected **Transforms** to the list.

Selections.GetSelectionCount()

Returns an integer number representing the number of currently selected **Selectable** components.

Selections.SetBlockChange(bool b)

Set true to block the selection from changing (or selecting at all), false to allow selection changes. Use this function to prevent selection changes when interacting with a GUI – e.g. block the selection changes when the mouse is positioned over a GUI.

Selections.IsChangeBlocked()

Returns true if changing the selection is blocked.

Selections.TagLimit = "tagname"

Sets the selection tag limit to the defined tag. Only **Selectable** components with matching tags can be selected.

Selections.ClearTagLimit()

Removes the selection tag limit, allowing selection of all **Selectable** components.