Ilumisoft

Visual State Machine

Documentation

Table of Contents

Welcome

Thank you for purchasing Visual State Machine!

This package has been designed to speed up your development, by providing to you a plug & play solution to create state machines visually in the editor.

On the next pages, you will find descriptions about how to use this package. We recommend to also take look at the example scenes, coming with this package. They showcase different use cases and are a valuable learning resource.

If you like the project, we would be grateful if you would take a minute and give us a rating in the Asset Store. This really helps us in order to create and improve our Unity Assets.

If you encounter any problems or errors or if you have any questions, please get in touch with us via email:

support@ilumisoft.de

Setup

1. Make sure you are using the latest release of Unity 2018.4 or higher.
2. Open the project you want to use with Visual State Machine.
3. Import Visual State Machine from the Asset Store.

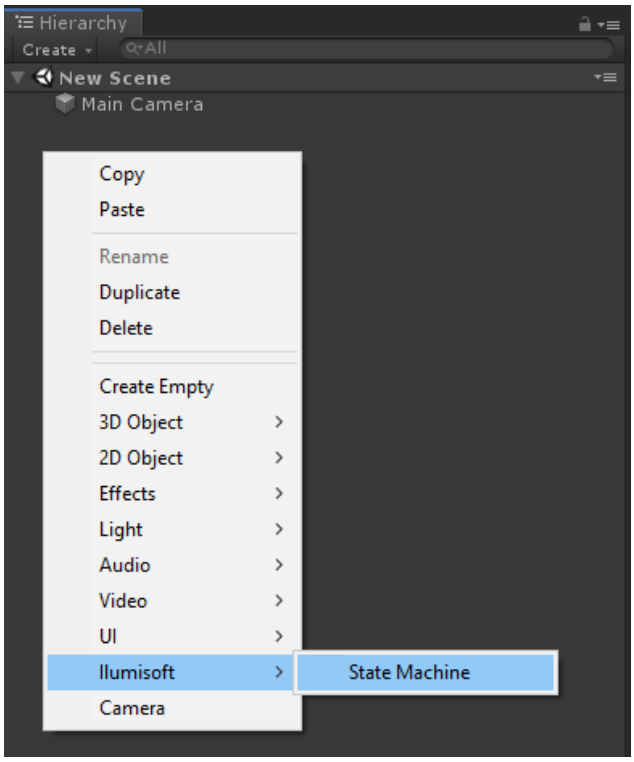
Since **Visual State Machine** has been designed to work out of the box, no further setup is required. If you still have problems importing the project or getting shown any error messages, please contact us.

Basics

How to Create a State Machine

There two ways creating a State Machine with VSM:

1. Right click in the **Scene** and select **Ilumisoft->State Machine**



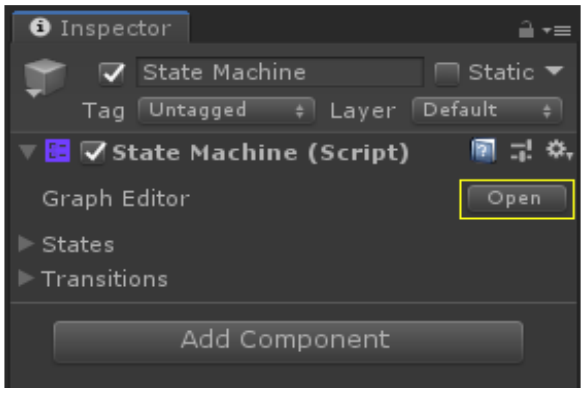
2. Select the **GameObject** you want to add a State Machine to in the inspector and add the **StateMachine** component to it via **Add Component**

How to Open the State Machine Graph Window

To open the **State Machine Graph** window, you can

- Click **Window->Visual State Machine-> State Machine Graph**
  This will focus any currently State Machine Graph window or open a new one otherwise.

- Select the State Machine GameObject you want to edit and click the open button in the inspector



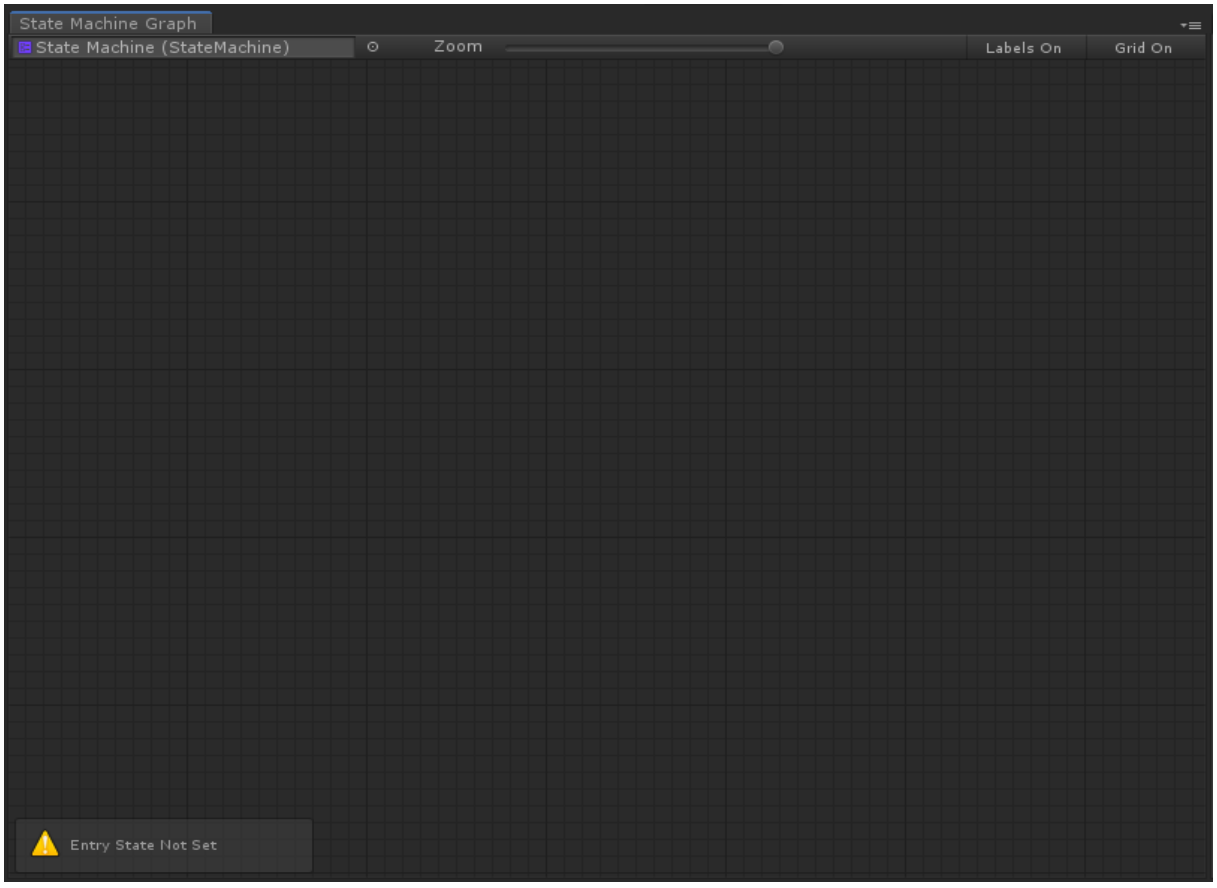
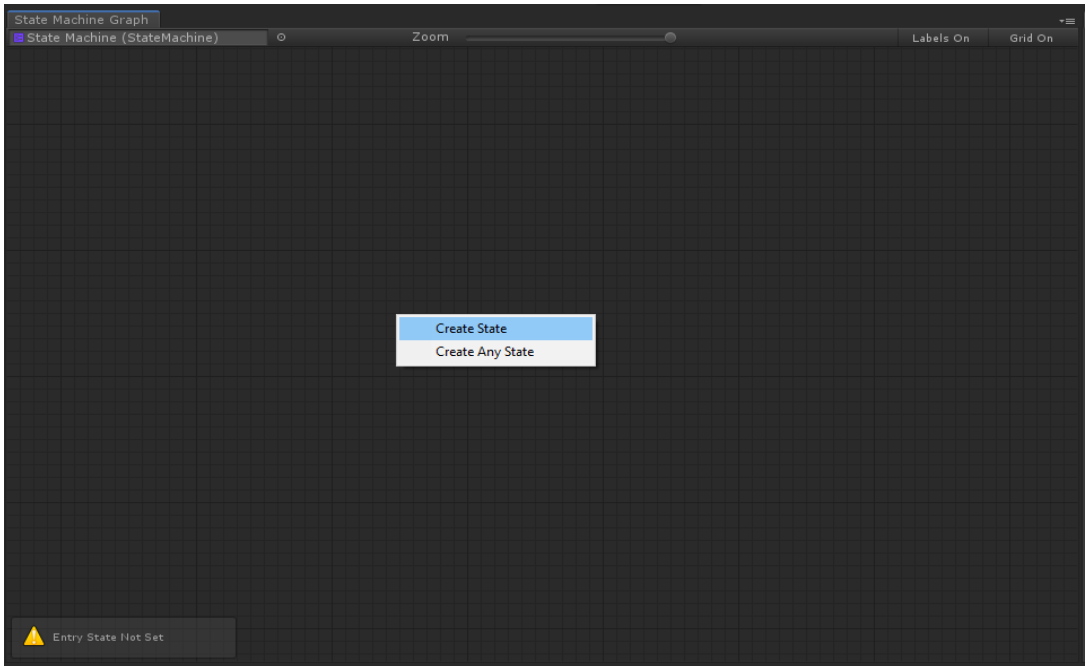


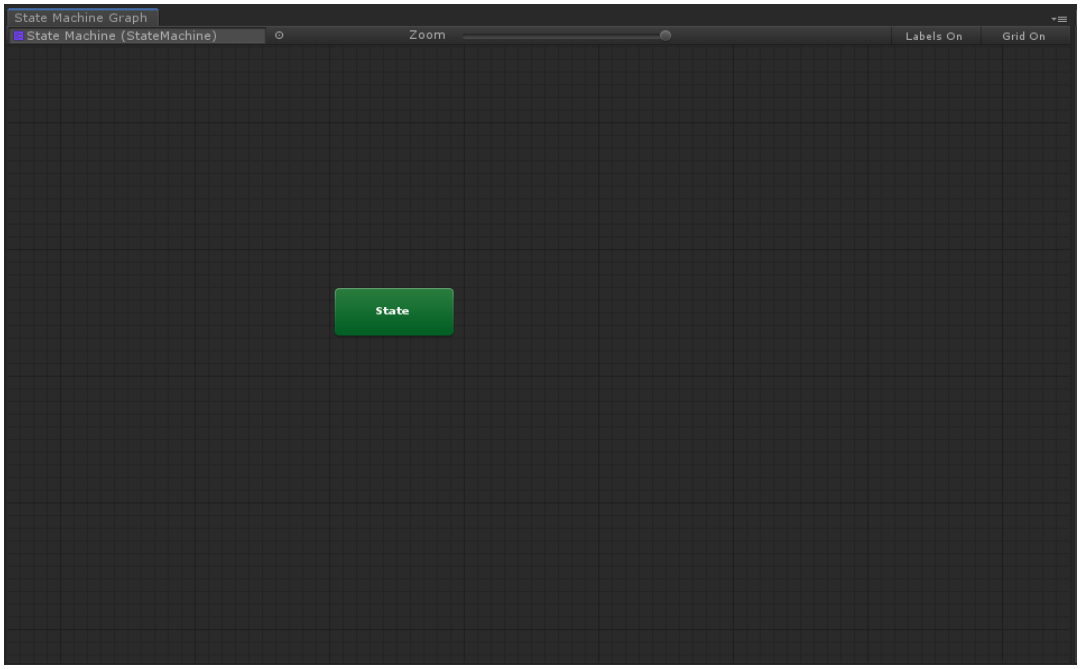
Figure 1: State Machine Graph Window

How to Create a State

1. Right click in the graph view and select **Create State**



Result:

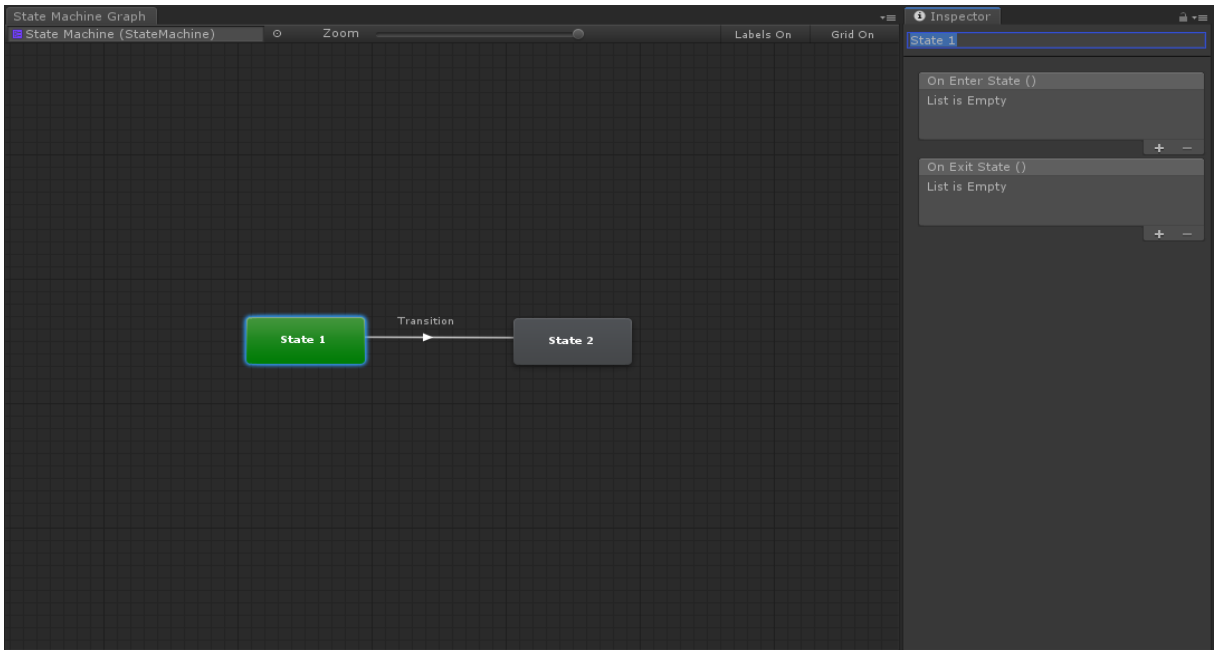


State IDs

Each state inside of a State Machine needs to have a unique ID assigned to it, which allows to identify it. You can use the ID of a state to access it via code in your scripts.

How to Change the ID of a State

1. You can assign your own IDs to states. First, select a state in the State Machine Graph window. In the Inspector select the ID field at the top and enter the ID you want to use.
   Please note: Since each state needs to be identifiable inside of the State Machine by its ID, you cannot assign the same ID to multiple states.

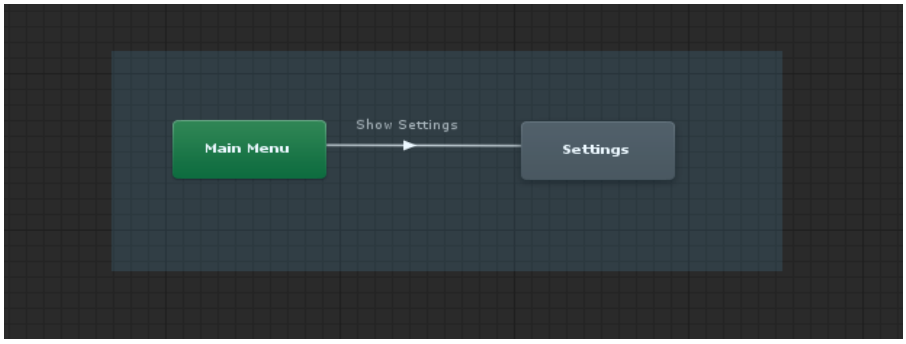


Info

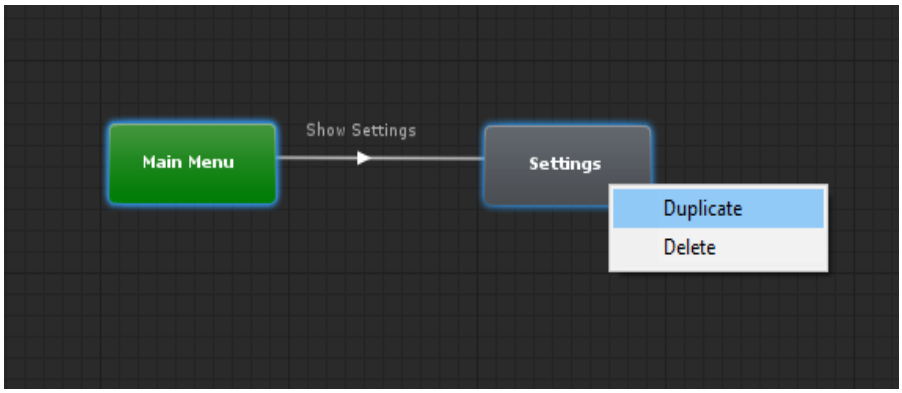
State IDs need to be unique inside of a state machine.

How to Duplicate States

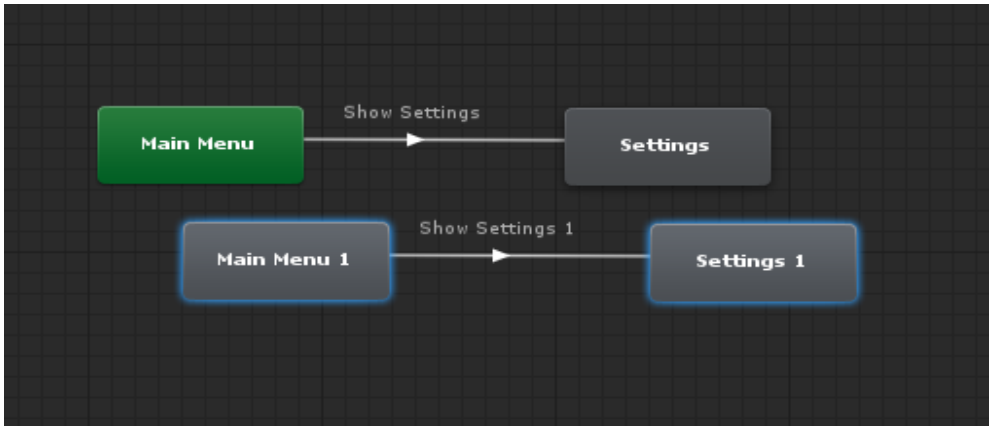
1. Select the states you want to duplicate



2. Right click on one of the selected states and select **Duplicate**



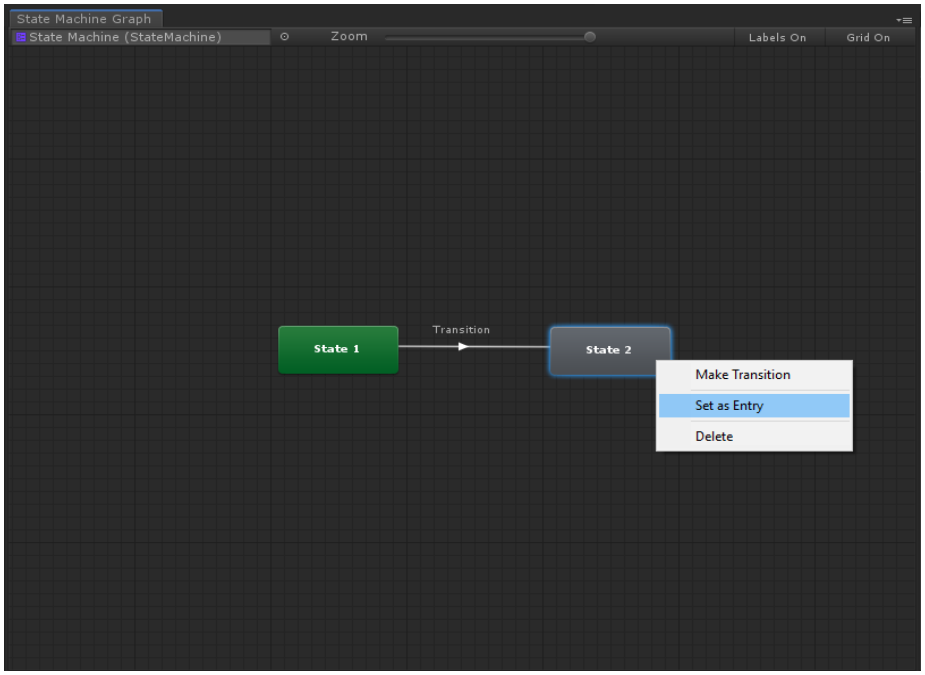
Result:



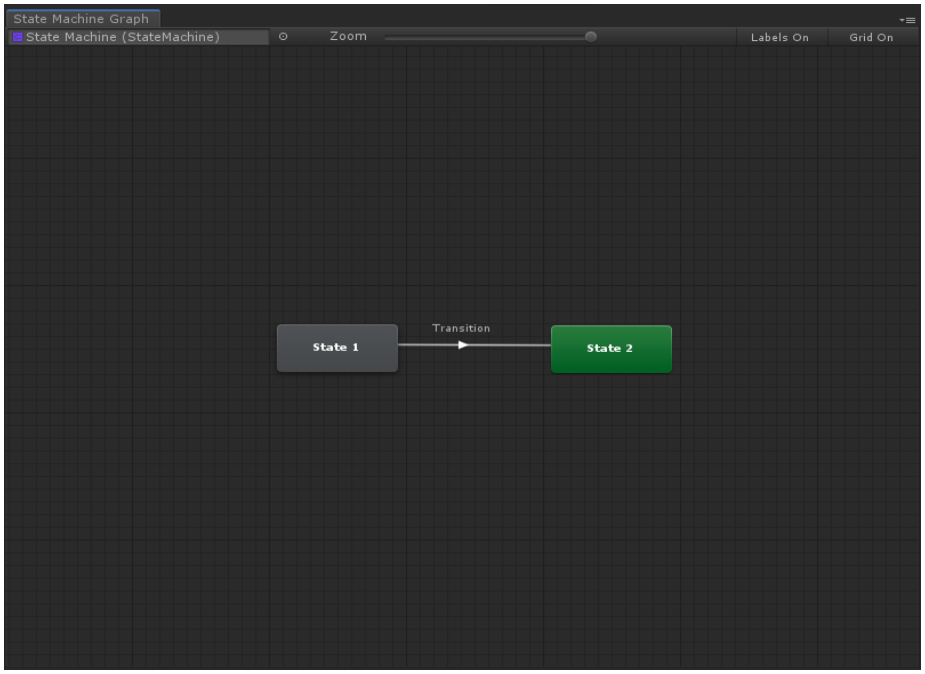
How to Set the Entry State

The entry state is the state, which is automatically entered when the state machine is started.

1. Right click on the desired state and select **Set as Entry**.



Result:

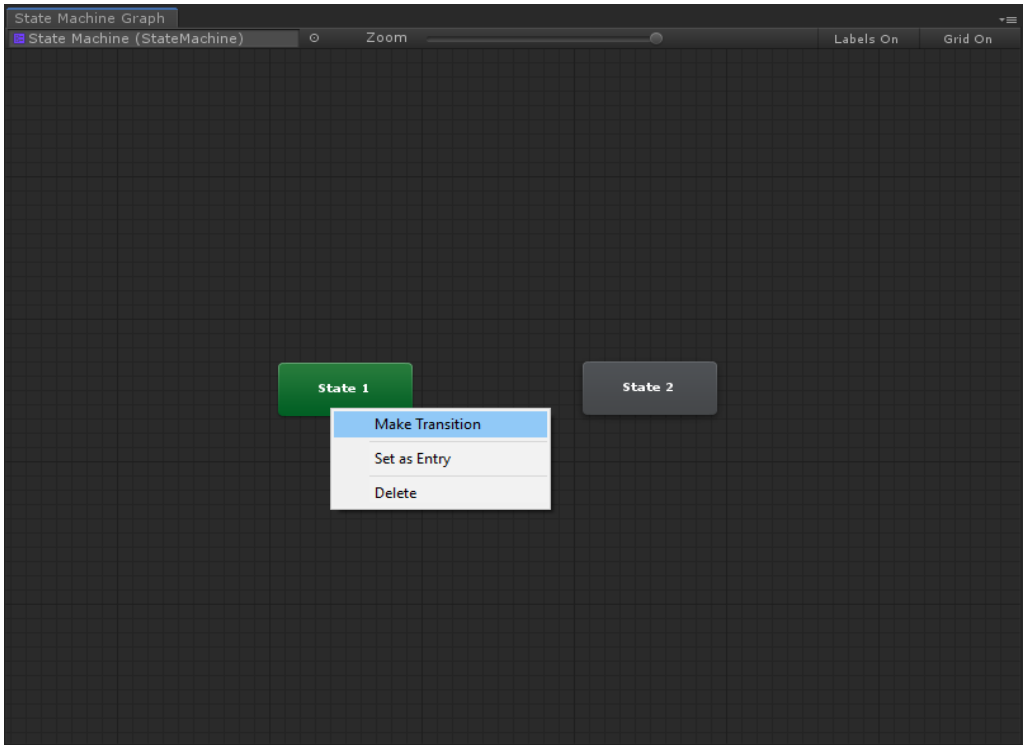


Info

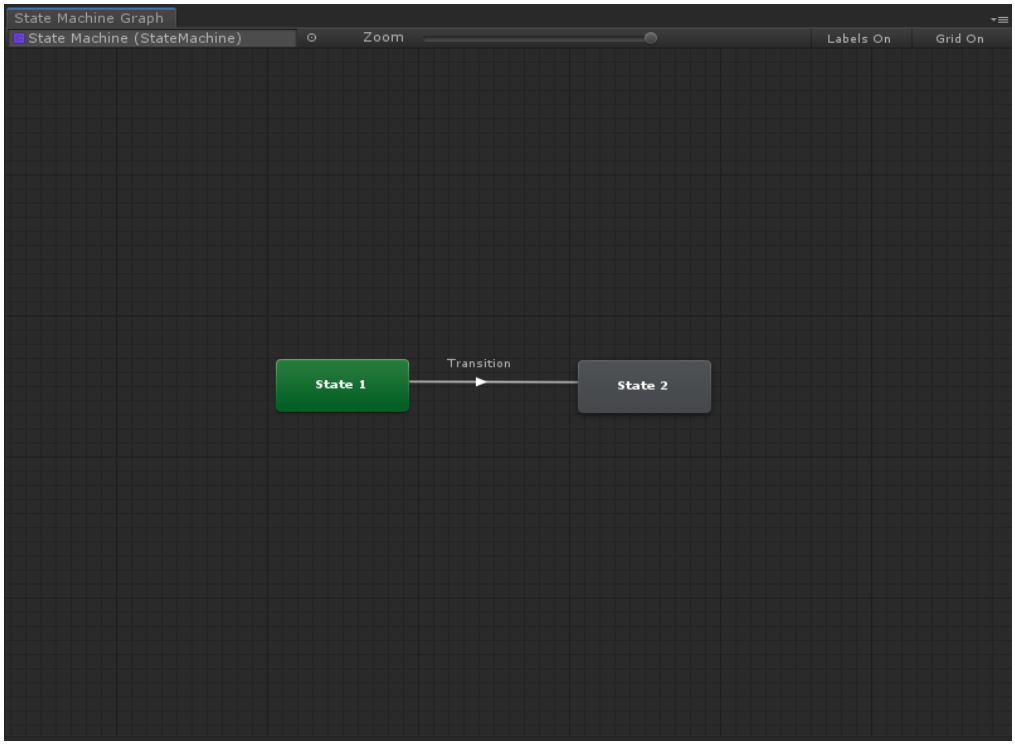
The first state you create on a state machine becomes automatically the entry state.

How to Create a Transition

1. Right click on the origin state and select Make Transition.



2. Left click on the target state to create the transition.

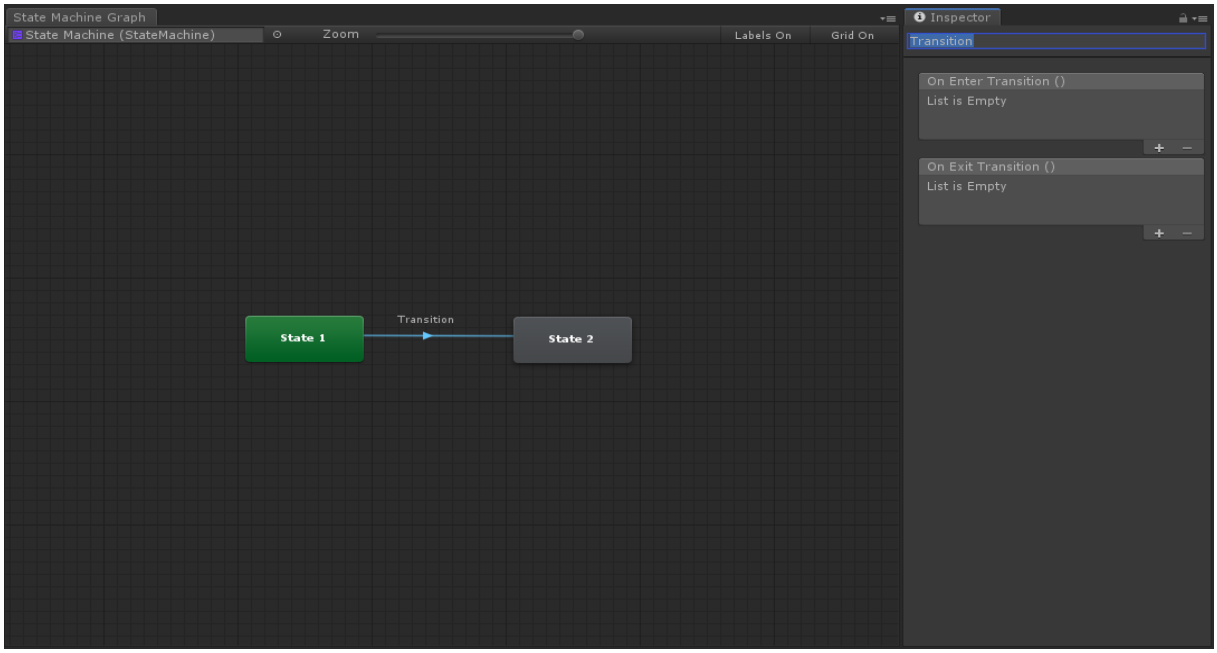


Transition IDs

Each transition inside of a State Machine needs to have a unique ID assigned to it, which allows to identify it. You can use the ID to trigger a specific transition or to access it via code in your scripts.

How to Change the ID of a Transition

1. Select the transition you want to assign a new ID to, by clicking on it in the State Machine Graph window. In the Inspector select the ID text field at the top and enter the ID you want to use.



Info

Transition IDs need to be unique inside of a state machine.

Transition Labels

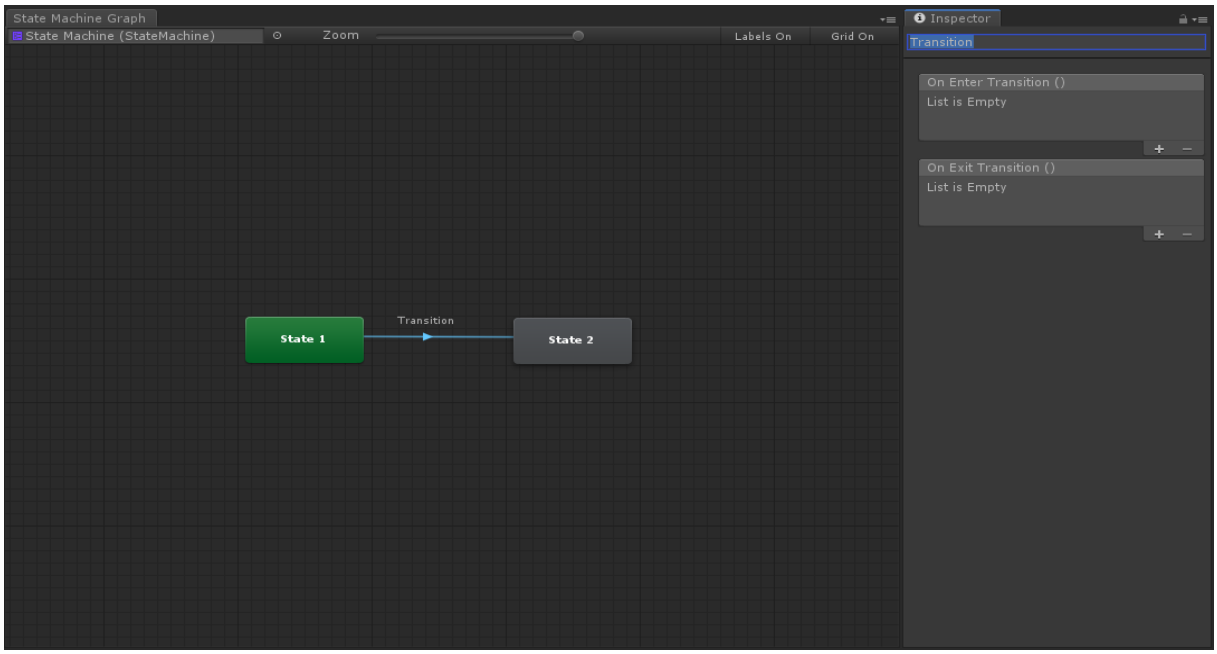
You can assign a label to a transition, which in contrast to the ID, does not need to be unique.

Example

Imagine you create a State Machine controlling your UI and have a return button on each panel. You could now assign the label “Return” to all transitions which should perform a return and call the **TriggerByLabel** method when the button is clicked.

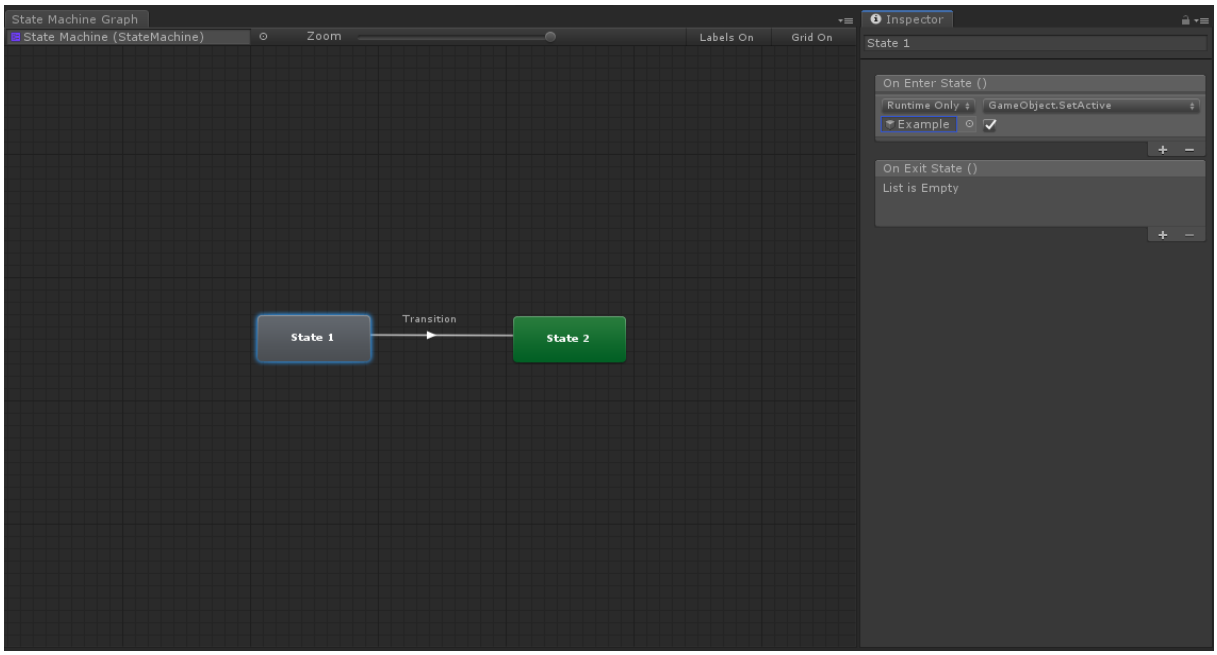
How to Assign a Label to a Transition

Select the transition you want to assign a label to, by clicking on it in the State Machine Graph window. In the Inspector select the **Label** text field at the top and enter the label you want to use.



How to Listen to Graph Events

States and transitions provide some events which are invoked when they are entered or exited. To add listeners to these events in the editor simply select the state or transition in the State Machine Graph and edit the **OnEnterState**, **OnExitState** or **OnEnterTransition**, **OnExit** Transition fields respectively.

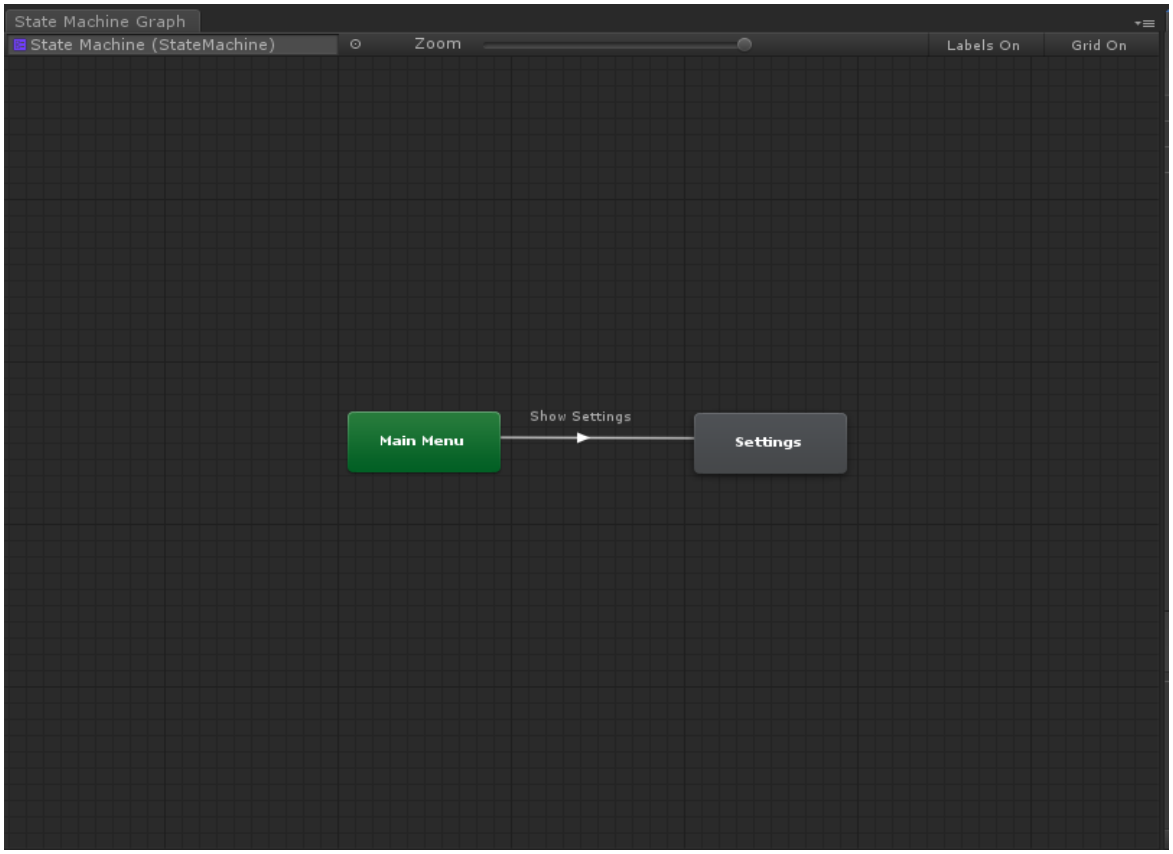


How to Trigger a Transition

To trigger a transition in a State Machine, call the **Trigger** method of the **State Machine** and pass the ID of the transition you want to trigger.

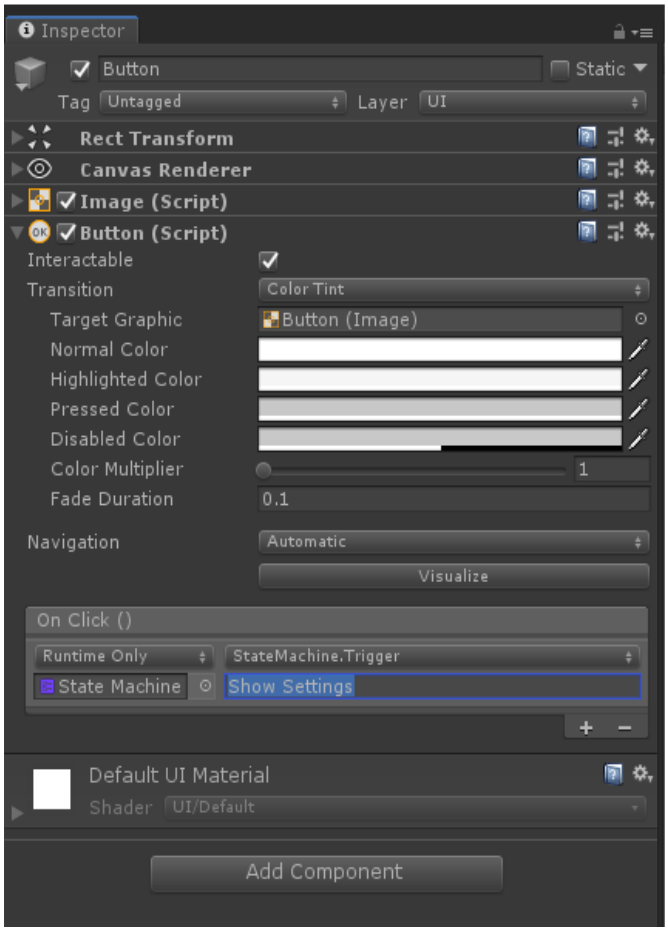
Example:

Imagine you have the following **State Machine** and want to trigger the transition with the ID **Show Settings**.



The following two examples will show how you could achieve this with an UI Button in the inspector or from inside a script.

UI Button OnClick Example



Script Example

This script will trigger the transition with name **Show Settings** on the **State Machine** set in the inspector when the "***S***" key is down.

```
//The namespace used by Visual State Machine
using Ilumisoft.VisualStateMachine;
using UnityEngine;

public class TriggerExample : MonoBehaviour
{
    //Reference to the state machine set via the inspector
    [SerializeField]
    StateMachine stateMachine = null;

    private void Update()
    {
        if(Input.GetKeyDown(KeyCode.S))
        {
            stateMachine.Trigger("Show Settings");
        }
    }
}
```

How to Trigger a Transition by Label

You can also trigger a transition by the label you assigned to it. Labels do not need to be unique, which means that you can have multiple transitions with the same label. When triggering a transition by its label, the State Machine will trigger the first transition it finds, which has the given label and can be triggered from the currently active state. This behaviour ca be useful when you want to implement a return button for your UI or e.g. a restart option after the game is won or lost.

To trigger a transition by label you can simply call:

- TriggerByLabel(string label)

or

- TryTriggerByLabel(string label)

Advanced

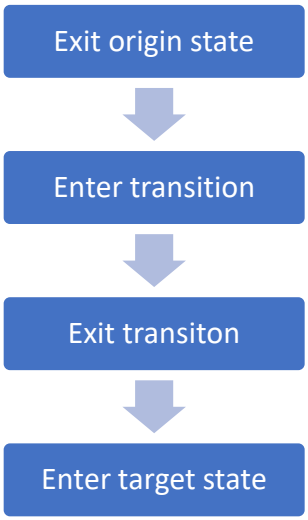
State Machine Insights

Lifecycle

The **State Machine** component is executed before all normal components (execution order(-1)) and automatically enters the entry state on Start.

Trigger Execution Order

When triggering a transition on a state machine, the state machine processes the following operations in the given order:



Support

If you encounter any problems, errors or got any ideas how the asset could be improved, please contact us via email:

support@ilumisoft.de