

Quick Start Guide

Introduction

Naninovel is an extension for Unity game engine. It consists of a C# framework and editor utilities to support development of [visual novel games](https://naninovel.com).

Online Documentation

For the extensive up-to-date guides and API reference, please visit naninovel.com.

Installation

Import Naninovel package from the Asset Store to your Unity project and wait for the initial scripts compilation and assets import process.

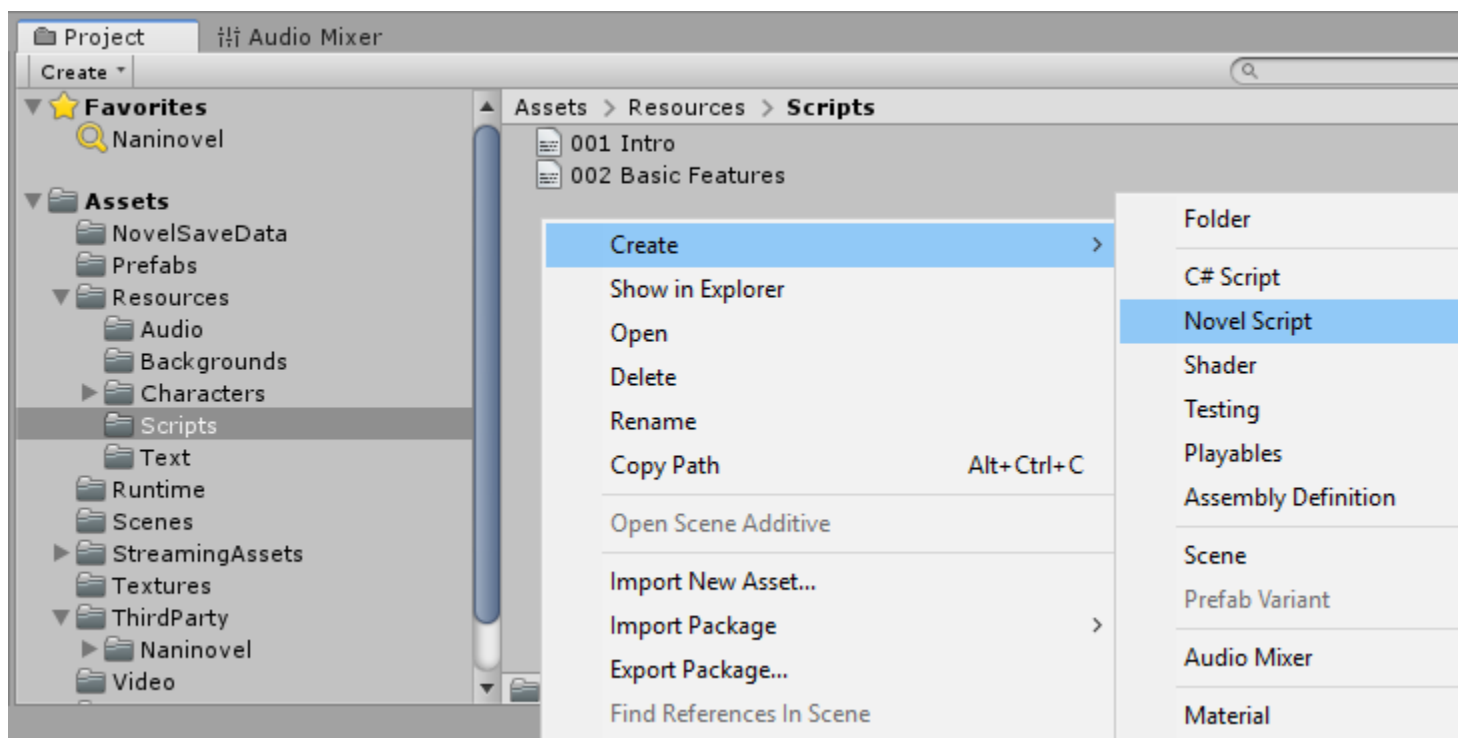
During the first-time import a number of configuration assets will be automatically generated inside the package folder; when the process finishes, an information window will be shown with the currently installed engine version and links to the online resources.

You are free to move the package folder anywhere inside your project assets directory.

In case you delete the package folder and then re-import it again in the same project, the configuration assets won't be automatically re-generated; open Naninovel -> Configuration menu to generate them before running the editor or building the player.

Add Novel Script

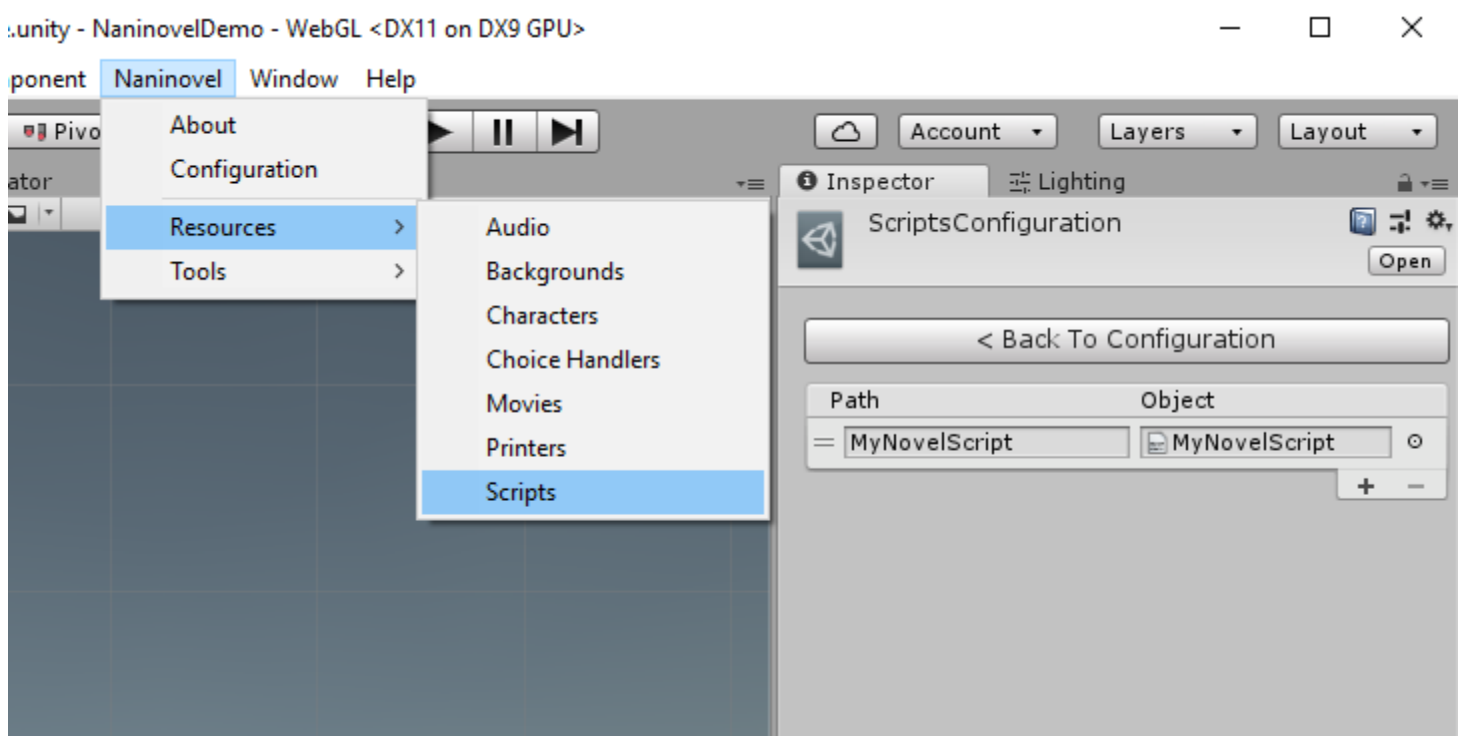
Use Create -> Novel Script asset context menu to create a novel script asset.



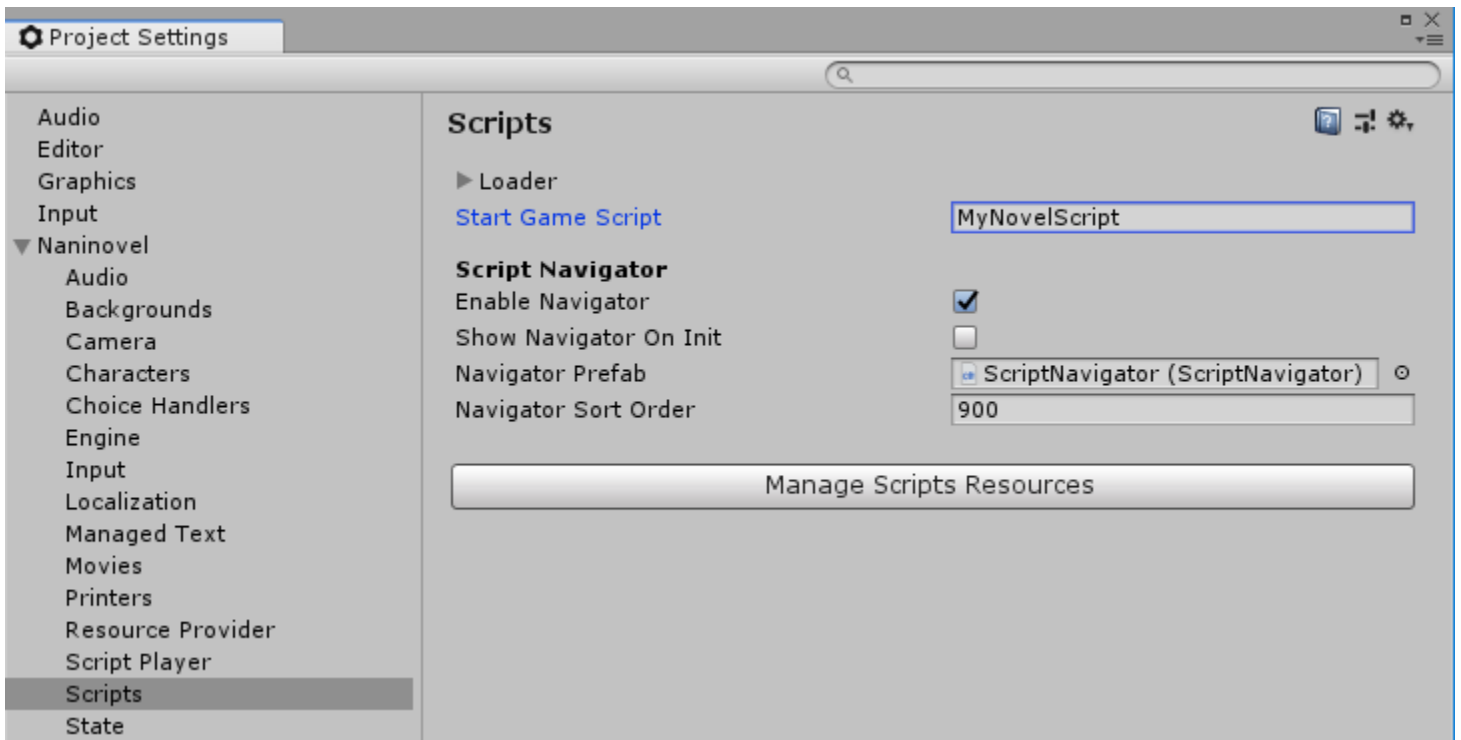
Novel scripts are text documents (.txt extension) where you control what happens on scenes. You can open and edit them with a text editor of your choice, like Notepad, Word or Sublime.

```
002 Basic Features.txt 001 Intro.txt x
1 >n Nanikun
2 >b Bubble
3
4 @char $n pos:0.5,-0.1 time:0
5
6 n: So, imagine you've imported Naninovel package to your Unity project.[i]
  For starters, create a novel script.[char $b.CreateScript pos:0.5,0.45,1
  wait:false][char $n pos:0.8 wait:false][i] Novel scripts are text
  documents where you control what happens on scenes.[i] Open them with a
  text editor of your choice, like Notepad, Word or Sublime.
7
8 n: For example, here is the contents of a novel script used for this demo
9 so far.[char $b.OpenScript]
```

Add the created novel script to the project resources. For this, open scripts manager using Naninovel -> Resources -> Scripts context menu, press + (plus sign) button in the list to add a new record and drag-drop script asset to the list. Alternatively, you can just keep the scripts in a Resources/Scripts folder and they'll be automatically exposed to project resources.



In case you've chosen to not use the script manager, you'll have to manually set Start Game Script value in the Naninovel -> Configuration -> Scripts menu; it will be set automatically when you add first novel script in the manager otherwise.



Open the created script in a text editor and add the following text:

```
Hello World!  
@stop
```

The first line will print the text "Hello World!" when the game is run and the second is required to gracefully stop script execution.

Enter play mode and start new game to see the result.

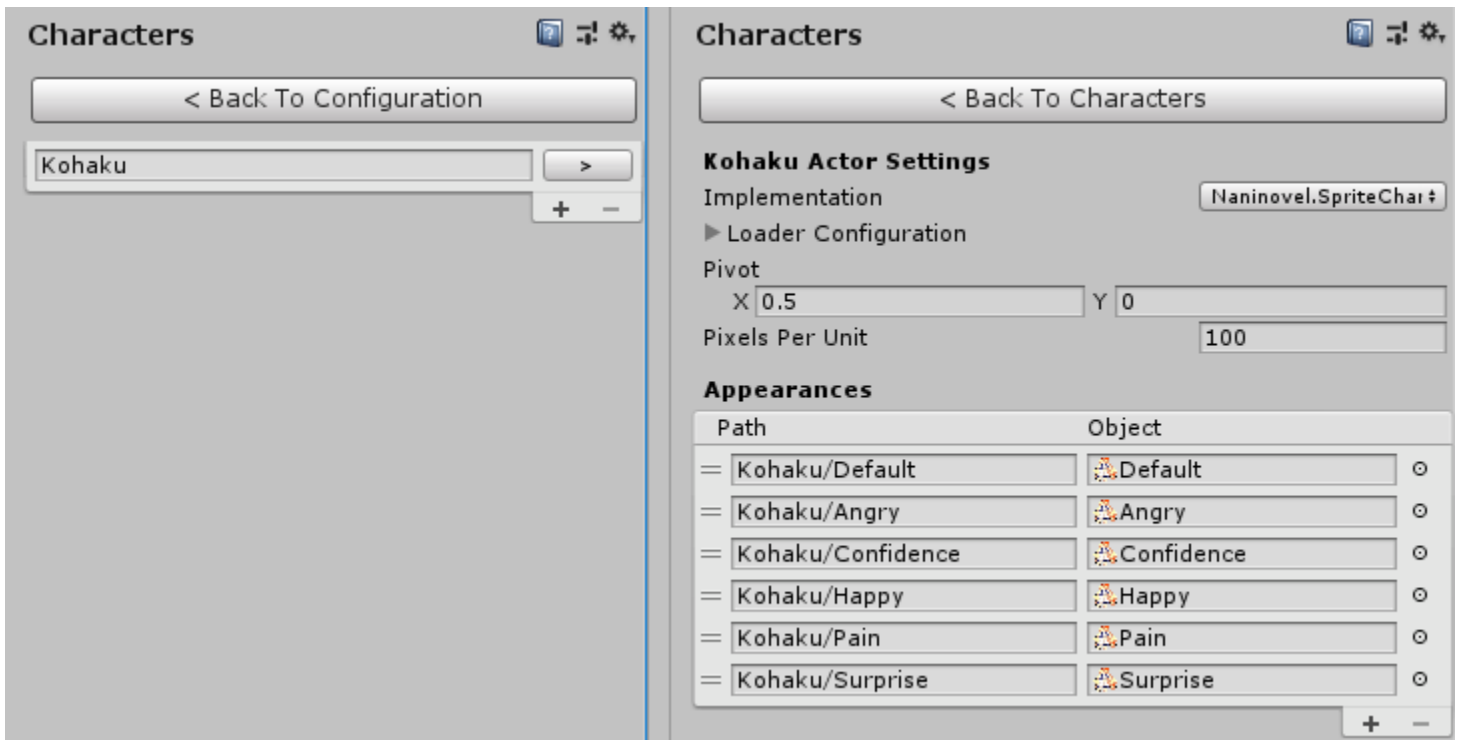
If the "NEW GAME" button is not interactable, make sure you've set a correct novel script name for the Start Game Script option in the scripts configuration.

Add Character

Characters in Naninovel can be based on regular and diced sprites, animated Live2D models and 3D meshes; you can add your own implementations as well. For the purpose of this tutorial, we'll use a sprite implementation.

Each character is represented by a name and a set of appearances. To add a sprite character, you can either use the character manager GUI or place the character's appearance assets in a `Resources/Characters/CharacterName` folder, where `CharacterName` is the name of the character.

If you choose to use the manager GUI, access it with `Naninovel -> Resources -> Characters` menu, add new character record specifying its name, then double click the name record (or press the `>` button) and add all the appearance assets (sprites) to the `Appearances` list.



Let's assume the added character name is "Kohaku". Edit novel script to show the added character:

```
@char Kohaku
Hello World!
@stop
```

Run the game and you'll see one of the character appearance sprites at the center of the screen. When you don't specify an appearance, either the one with "Default" name or a random one will be chosen by default. To select a specific appearance, add its name after the character name separated by a dot like this:

```
@char Kohaku.Happy
Hello World!
@stop
```

Given there is an appearance with the name "Happy" added for the character "Kohaku", the corresponding sprite will now be shown instead of the default one.

You can now also associate the printed text with the character by adding its name followed by a colon before the text:

```
@char Kohaku.Happy
Kohaku: Hello World!
@stop
```

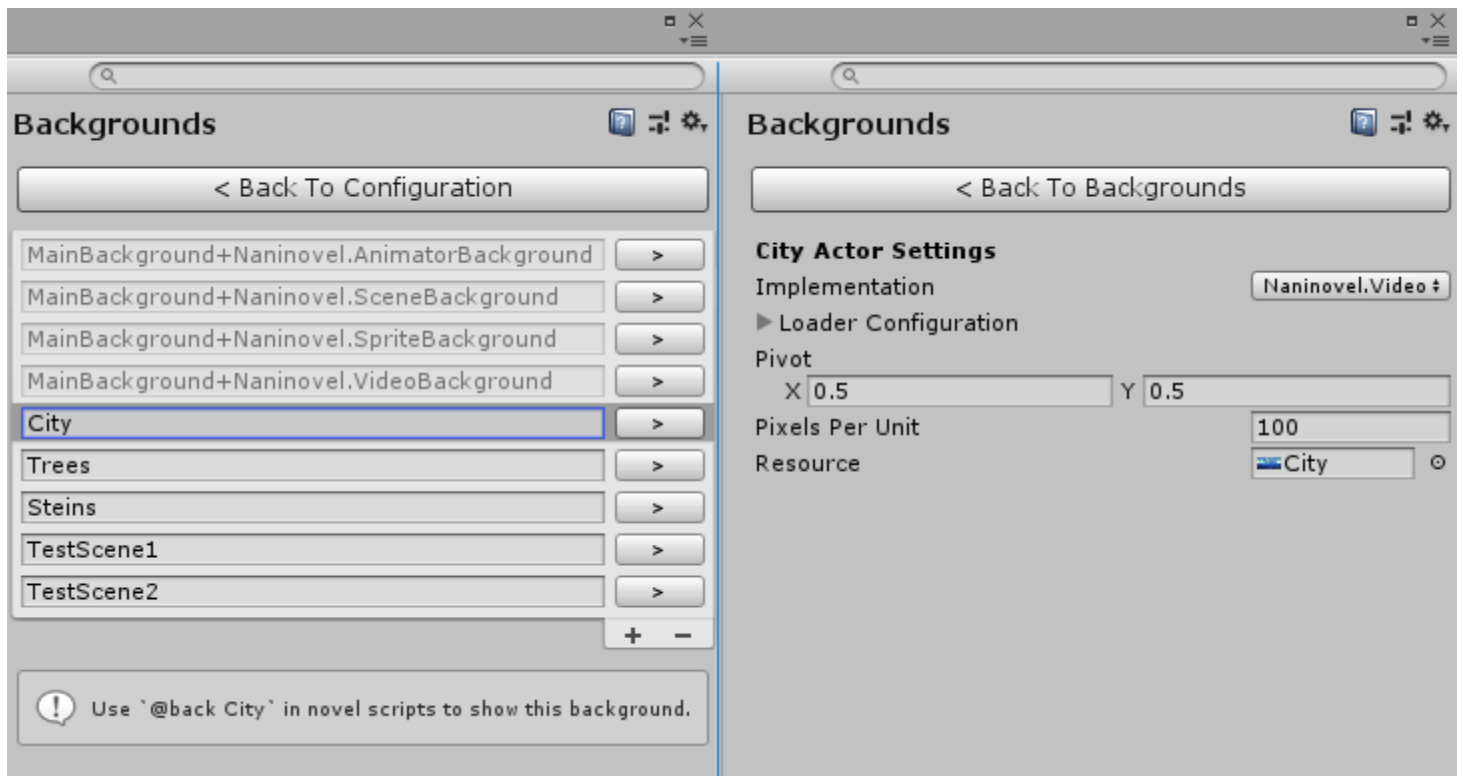
To hide (remove from scene) a character (or any other actor), use @hide action followed by actor name:

```
@char Kohaku.Happy
Kohaku: Hello World!
@hide Kohaku
@stop
```

Add Background

Similar to characters, a background can be represented in multiple ways in Naninovel: sprite, animated object, video and scene; custom user implementations are also possible.

Sprite background is represented by a name and a single sprite asset. Add sprite background in the same way you've added a character: using editor GUI `Naninovel -> Resources -> Backgrounds` or place it in a `Resources/Backgrounds` folder.



Let's assume the added background name is "City". To show a background, use a `@back` action followed by the background name:

```
@back City
```

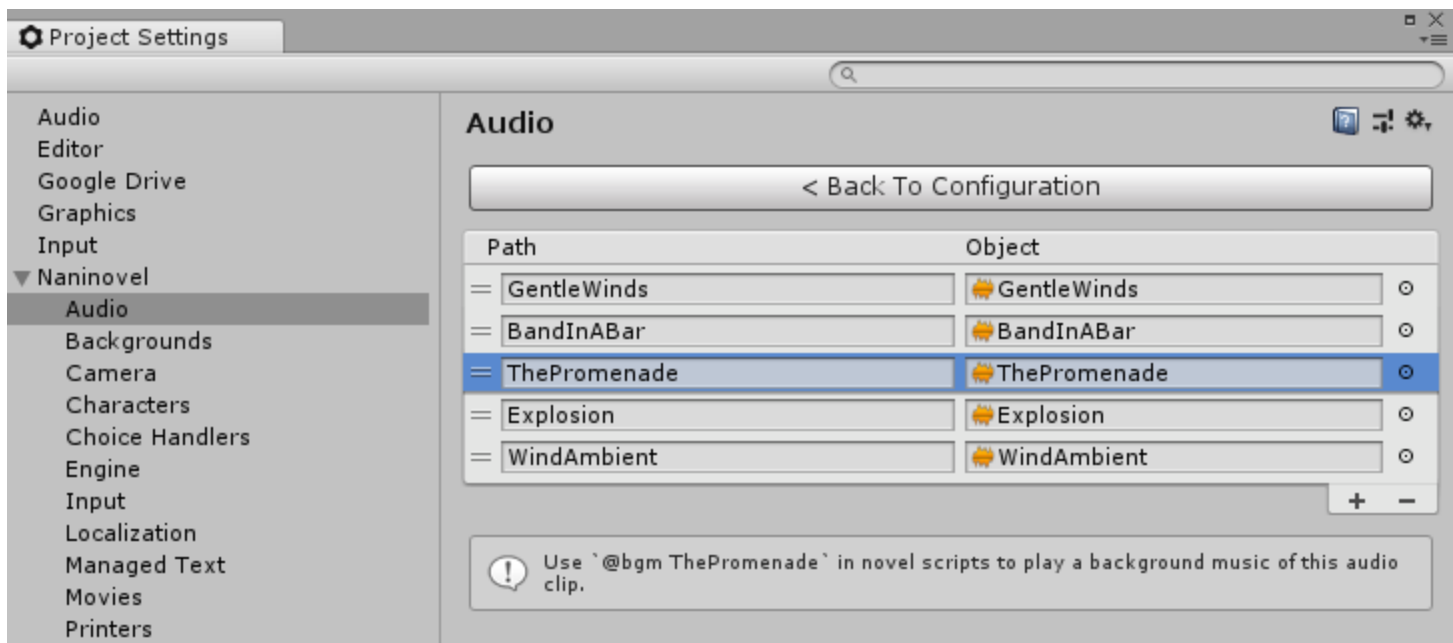
When switching between backgrounds a cross-fade transition effect will be used by default. To change the effect, specify transition after the background name:

```
@back City  
@back School.RadialBlur
```

This will transition "City" to "School" using "RadialBlur" transition effect.

Add Music and Sound Effects

To add a BGM (background music) or SFX (sound effect) asset, either use `Naninovel -> Resources -> Audio` editor menu or place the audio assets inside `Resources/Audio` folder. You can use any audio formats [supported by Unity](#).



Let's assume the added BGM file name is "ThePromenade". To play this track as a background music use @bgm action followed by the name of the track:

```
@bgm ThePromenade
```

Only one background music track can be played simultaneously and a cross-fade effect will be automatically applied when switching the music tracks. The music will also loop by default, though you can change this, as well as volume and fade duration.

On the contrary, you can play multiple sound effects at the same time and they won't loop by default. Assuming you've added a "Explosion" SFX, use an @sfx action to play it back:

```
@sfx Explosion
```

Support

If you have issues with the engine, make sure you're using the latest available version. You can update the package using Unity Asset Store interface: Window -> Asset Store.

If updating didn't help, try to re-install the package by deleting Naninovel folder from the project and re-importing it from the Asset Store. Be aware that by deleting the package folder you'll lose all the engine configuration data; you can keep the configuration by migrating Naninovel/Resources/Naninovel/Configuration folder to the new package installation folder.

Issue Tracker

In case the above steps didn't help to resolve the issue, check the [issue tracker](#) — chances are the problem is already being worked on. You can also create new issues there to ask a question or report a bug. When reporting a bug, please use bug template and fill in the required information.

Unity Forum

Not comfortable using GitHub? Check out [Unity forum thread](#). To discuss the issue in private, [send a private message](#).