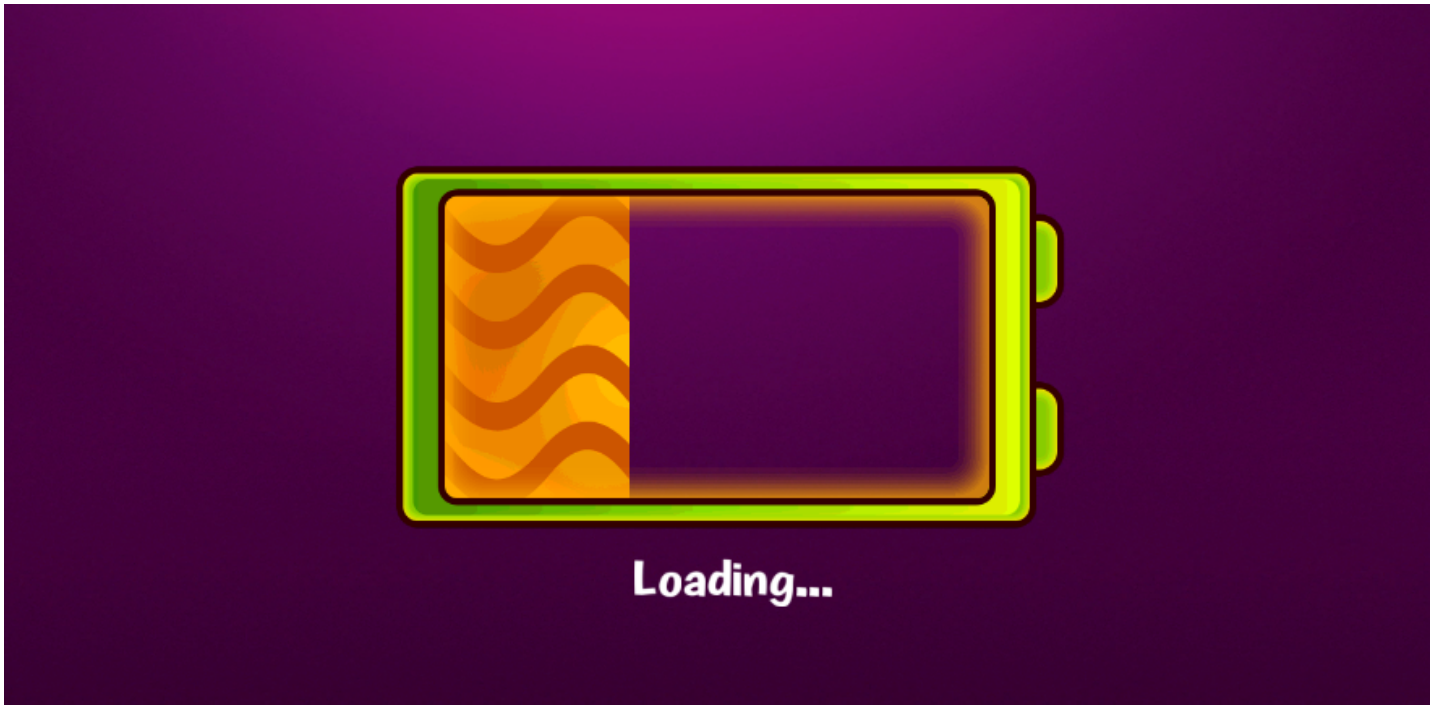


This is an offline copy of documentation available online. You won't be able to follow some links. Please see the README.txt file for an URL to online version of this documentation.

## Loading Screens Add-on



Loading Screens Add-on is an extension for Mad Level Manager. It allows you to easily create beautiful level loading screens taking advantage of the powerful features of Mad Level Manager.

Mad Level Manager loading screens can display loading progress bars even if you don't own Unity Pro. At the end of loading process you can display "press anywhere to start" message. Pressing on it will immediately replace visible contents with just loaded level.

You can prepare loading screen graphics theme by yourself or use one of three beautiful themes that are included in the package.

**Mad Level Manager Loading Screens is an extension that can be acquired separately only from the Unity Asset Store.**

## Getting started

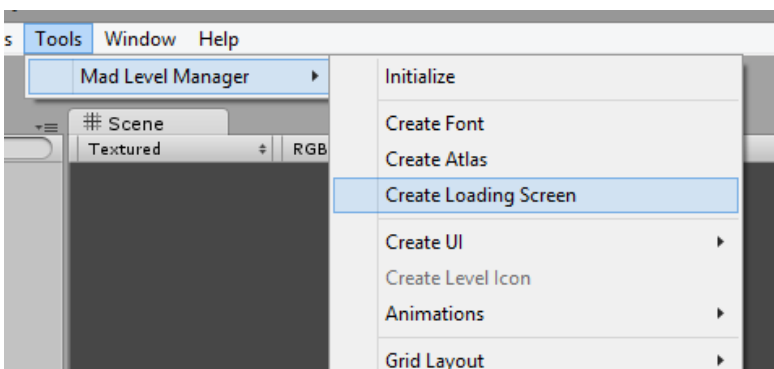
Please make sure that the latest **Mad Level Manager** (base package) and **Mad Level Manager Loading Screens** are imported into your project. There should be these two directories visible in your *Project* window:

- Mad Level Manager
- MLM Loading Screens Addon

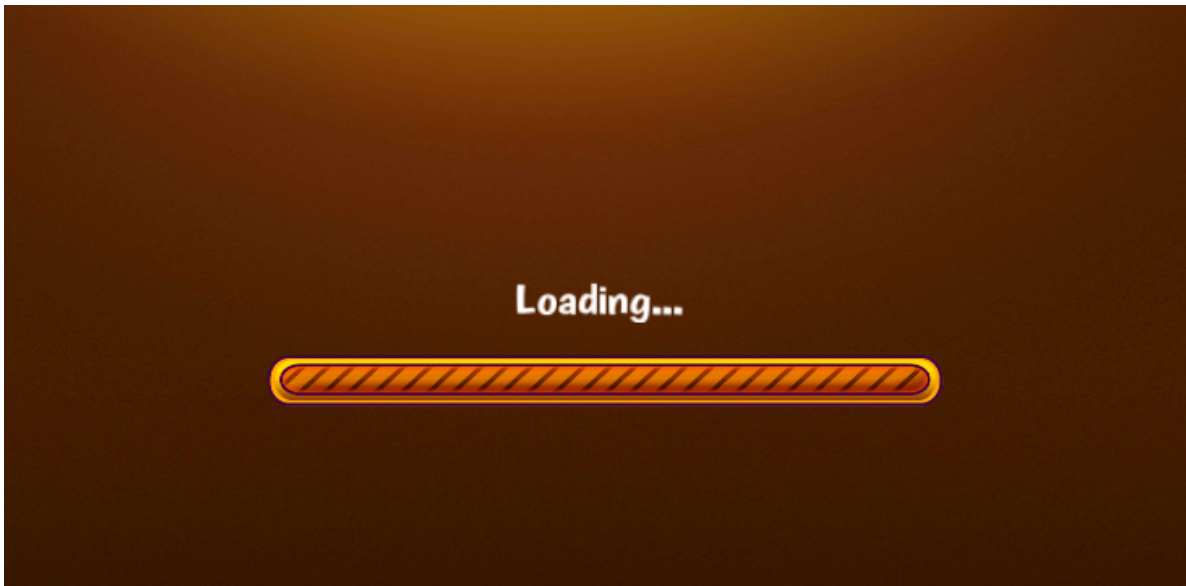
Always be sure to use the latest version. Having the outdated version of Mad Level Manager may result in unexpected issues and error messages.

## Creating a level loading screen

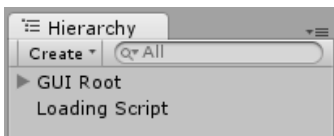
Creating a level loading screen is really simple. On the empty scene, please execute: *Tools* → *Mad Level Manager* → *Create Loading Screen*.



Your scene will be initialized with an example theme.



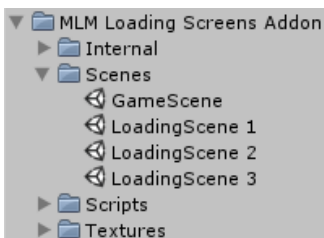
In the *Hierarchy* you can notice two important objects.



- **GUI Root** is the root object of the **Mad2D** UI system. You can create UI elements from the *Tools* → *Mad Level Manager* → *Create UI* menu.
- **Loading Script** is the object that contains the most important **Loading Screen** script. We will talk about it in a moment.

## Using existing themes

If you don't want to waste your time on creating your own level loading screen, or if you're happy with the one of provided themes, you can find these in the **Scenes** directory. There's also a test level called **GameScene**. You can use it to check how your playable scene will be loaded and paused before it is actually visible.



## The Loading Screen script

After your scene is set up, you need to configure your **Loading Screen** script. The Loading Screen script is a pretty simple to comprehend. It may be attached to any game object on the scene, but we're attaching it to a *Loading Script* empty game object to make clear that this is the place where it can be found. This is what its inspector looks like:

Let's explain it step by step.

## Test Mode

The Test Mode is most probably the first thing you will want to set up. Normally loading screens should be set up as [Mad Level Manager Extension](#) **before the target level**. Mad Level Manager will be confused if you run your game from the extension, so the test mode will allow you to test the loading screen behavior without any particular setup.

- **Enabled** – Set to true if you want to enable the test mode.
- **Load Level** – The name of the level you want to load.

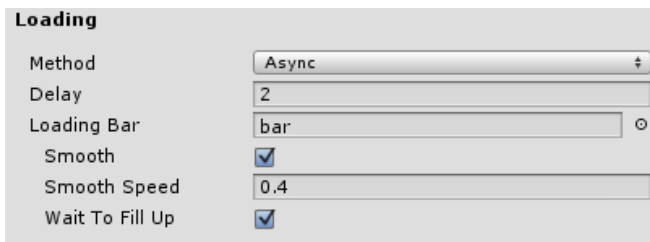
Load Level takes a level name as an argument. It can be the level name from [Mad Level Manager Level Configuration](#) or the name of scene from [Build Settings](#) (what comes first).

Test mode will create a game object called `_mlm_ignore` in your level loading scene. Do not delete it. It will be removed when you disable the **Test Mode**. This object tells Mad Level Manager to skip the synchronization check.

Loading screen is configured to work as [Mad Level Manager Extension](#) **before the target level**. Any other way of using it will cause errors, except the test mode.

## Loading

Here is the place how the actual loading should behave.



**Loading**

Method: Async

Delay: 2

Loading Bar: bar

Smooth: ☒

Smooth Speed: 0.4

Wait To Fill Up: ☒

- **Method** – The loading method.
  - **Regular** – Will load the level using the regular (blocking) loading method. This is the only method that is working for the free version of Unity.
  - **Async** – Will load the level asynchronously. If the **Loading Bar** is attached, current loading progress will be displayed. **work only with Unity Pro.**
- **Loading Bar** – The game object that has a fill value kind variable or property. This can be:
  - None – no loading bar will be displayed.
  - MadSprite – Mad Level Manager UI Sprite component.
  - UISprite (NGUI) – Sprite from NGUI.
  - Image (uGUI) – Unity 4.6 uGUI image.

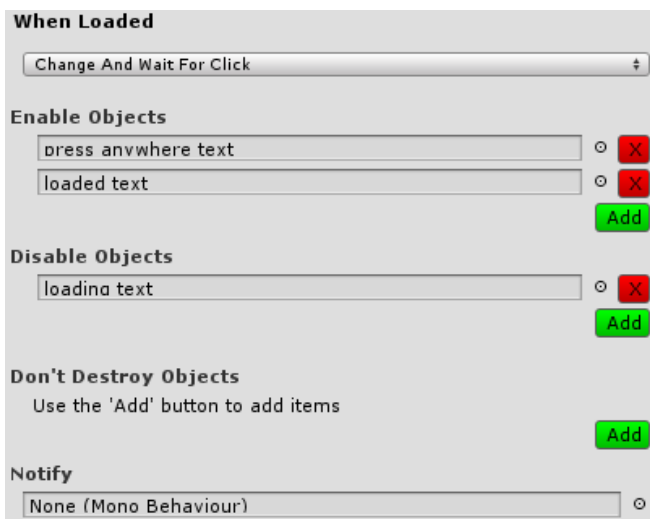
### Loading bar settings

- **Smooth** – Bar filling will be smoothed.
- **Smooth Speed** – Filling-up factor. For example 0.5 means 50% fill up per second.
- **Wait To Fill Up** – When the level is loaded before the bar is 100% filled, the loading screen will wait for the **Smooth** animation to finish.

Note that if you are using **Regular** loading method, you will be able to see the bar filling up when **Smooth** and **Wait To Fill Up** options are enabled. This little trick allows you to indicate loading progress without Unity Pro features.

### When Loaded

The last section is responsible to what will happen when the level has been loaded.



**When Loaded**

Change And Wait For Click

**Enable Objects**

press anywhere text

loaded text

**Disable Objects**

loading text

**Don't Destroy Objects**

Use the 'Add' button to add items

**Notify**

None (Mono Behaviour)

### When Loaded

You have three options here:

- **Show Immediately** – Current scene will be replaced at once when the target scene loading has been finished. This is not entirely true when **Wait To Fill Up** is enabled because the bar filling-up animation will delay this action.
- **Wait And Show** – Will pause for X seconds (new configurable field appears) before the level is shown. As before, the **Wait To Fill Up** option may take its time.
- **Change And Wait For Click** – Changes the loading screen appearance (see **Enable/Disable Objects**) and waits for a single click or touch anywhere on the screen. This is a good option for “Press anywhere to start” type notification.

**Wait And Show**, **Change And Wait For Click**, or any option with **Wait To Fill Up** enabled will actually load and launch your scene in the background, but the [Time.timeScale](#) will be set to 0 for the time being. Make sure that your scene scripts are not trying to ignore the time scale.

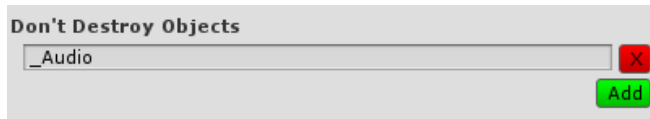
### Enable/Disable Objects

In these two lists you can define which loading scene objects should be enabled or disabled when the level has been loaded. This is useful when the loading screen is set to be visible for a while after. For instance you can disable the “Loading” label and replace it with the “Loaded” label. Also you can show “Press anywhere to start” label if you want your player to press anywhere on the screen (**Change And Wait For Click** option).

## Don't Destroy Objects

When level is loaded, **Loading Screen** scripts destroys all scene objects found at the moment when the loading has been started. Sometimes you may want it to ignore some kind of objects. For example you may have an “\_Audio” object that is playing your music and you don't want it to be destroyed in the process.

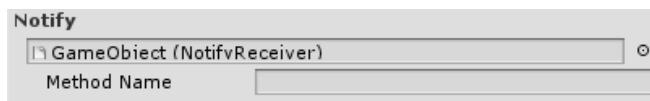
To add any number of objects that should be preserved, click on the **Add** button.



Then in new text field enter the object name exactly as it is displayed in the *Hierarchy* view.

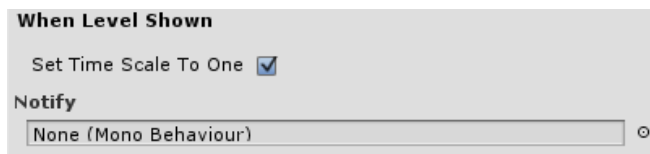
## Notify

You can get a notification when the level is loaded. In order to do that, assign any script to the **Notify** field.



A field called **Method Name** will appear. Be sure to enter a method name that you want to be called when your level is loaded. This function uses a [SendMessage](#) under the hood.

## When Level Is Shown



Here you can configure what should be done when the level is visible to the user.

- **Set Time Scale To One** – By default until the level is shown, the [Time.timeScale](#) value is set to 0 (this is common game pausing method). Keep this option enabled to change *Time.timeScale* to 1 when the level is shown.
- **Notify** – As [before](#) you can receive a notification when the level is shown.

## Testing your loading screen

1. Make sure your level described in **Test Mode** → **Load Level** is added to your [Build Settings](#)
2. Run your scene

## When you're done

1. Disable the **Test Mode**.
2. Save your scene
3. Create a new **extension** and add your new loading scene as **Load Before Level**, so it will be displayed before the actual level is loaded. For more information please [take a look here](#).
4. Test it by entering the level from the level select screen.

The loading screen won't work if entered directly. It may be fixed in the future.

## Troubleshooting

### I hear my level sounds/music too soon.

It's because the music is ignoring the [Time.timeScale](#) value. Please adjust your scripts so nothing will play/start until Time.timeScale is equal to 0. It may look like this:

```
bool musicStared = false;
```

```
void Update() {  
    if (!musicStarted && Time.timeScale != 0) {  
        StartMusic();  
        musicStarted = true;  
    }  
}
```

### **My music object (or any other object set as DontDestroyOnLoad()) is destroyed**

Please read [this](#).

### **I see some GUI elements of my level on the level loading screen.**

There can be several reasons for that:

#### **You're using OnGUI() functions for GUI rendering.**

I'm sorry to say that but using OnGUI() for the production GUI is just wrong. There are so many bad things about OnGUI() functions that I don't know where to start. If you're seeing your debug GUI, don't worry about that, or consider switching to uGUI or other GUI systems available in the Asset Store.

#### **You have a camera with depth of 100.**

The trick behind the loading screen displaying even when the game level is loaded is behind the camera with the depth of 100. In most games, camera depths are 1 or 2-digit values, and 100 is the biggest possible camera depth. If this happen to you, please adjust your game cameras depth to be lower than 100.

#### **I have an issue that is not listed here.**

Then don't hesitate and please write to [support@madpixelmachine.com](mailto:support@madpixelmachine.com). We will assist you as best as we can!