



# Looader<sup>1.2</sup>

## Documentation

## Package includes

All necessary scripts,  
Ready to use prefabs,  
4 different templates,  
Animations,  
Trigger events,  
and more...

## So what is Looader?

Looader is a loading screen solution for your project.  
With Looader, you can create good looking loading  
screen within minutes. And it's totally free!

# First to do

You've imported Looader to your project. So, what now?

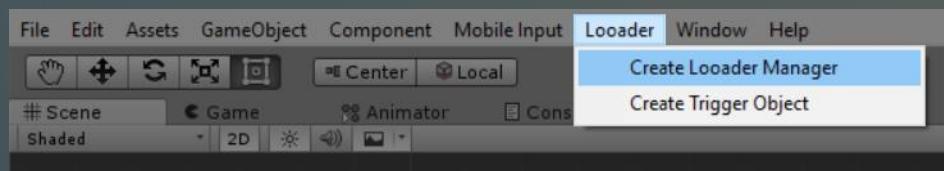
First of all, let's decide what you're gonna do.

# Licence

Thank you for using Looader.  
Kit contains all necessary scripts and sprites  
for ready to use in commercial purposes.

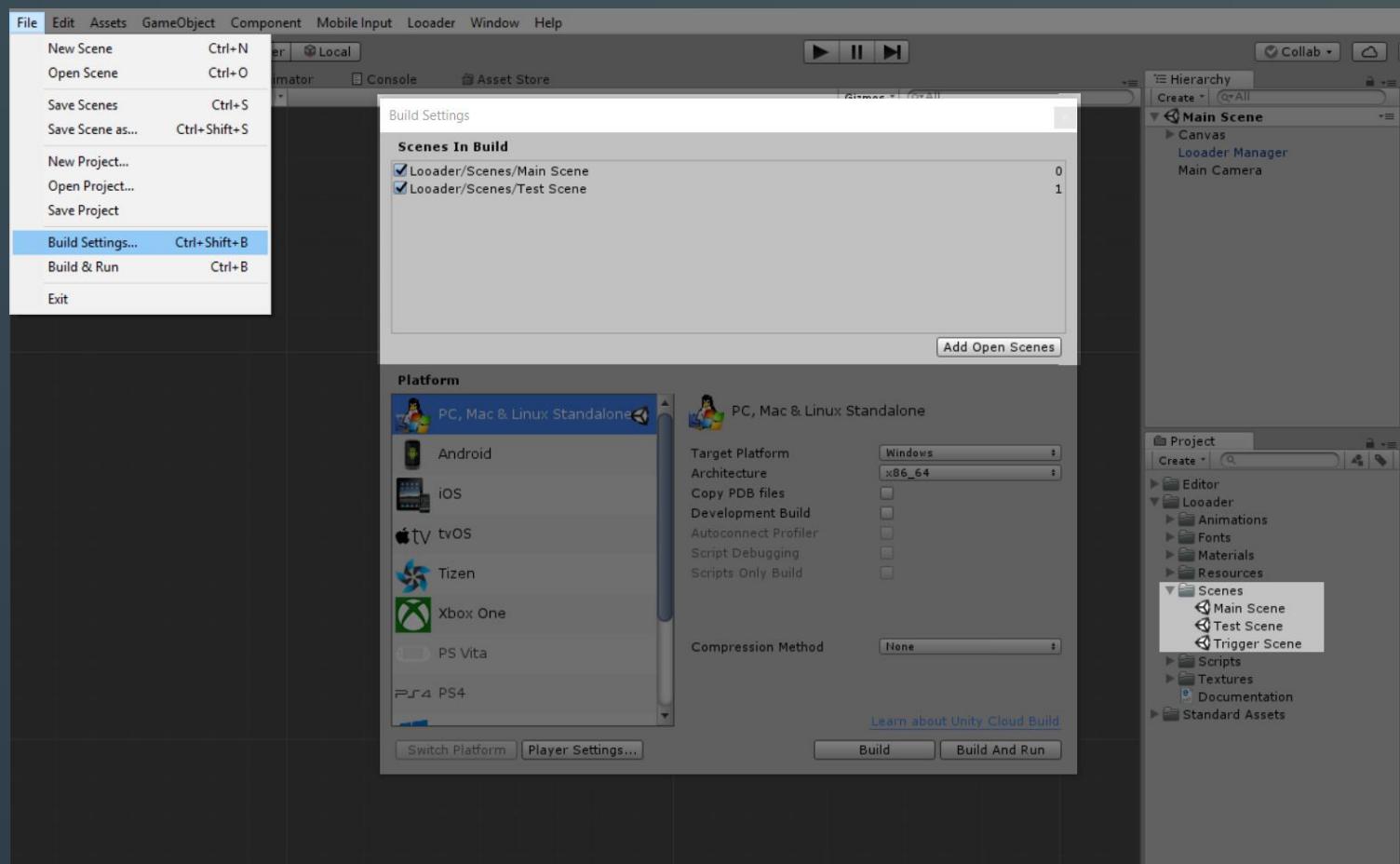
# Ready to use templates

First, we need to create Looader Manager object.  
You can easily create it in the editor window.



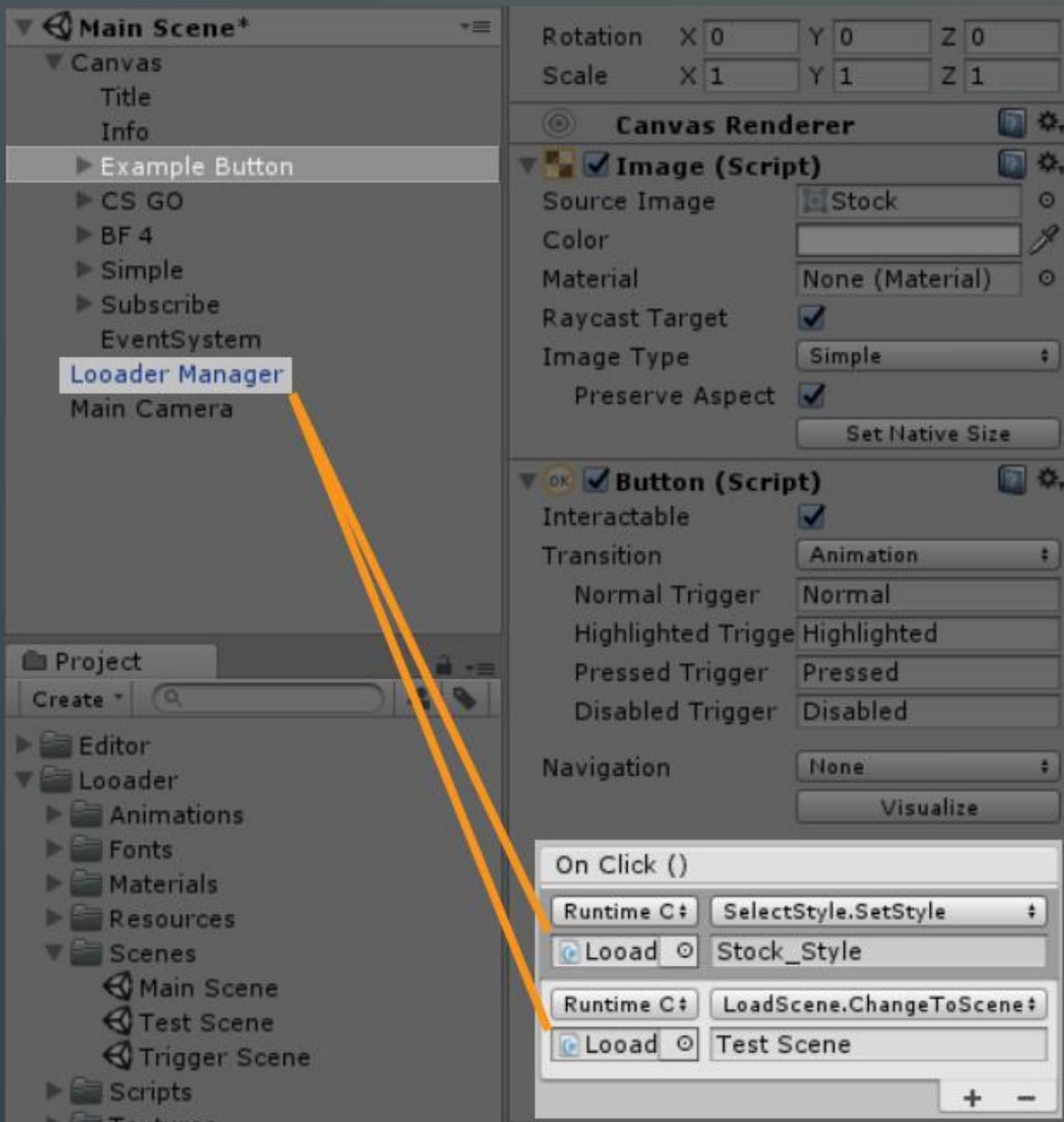
That's all for loading screen! Quite easy, isn't it?

Now, we need to prepare scenes and buttons for loading process. Go to Build Settings, and drag your scenes to Scenes In Build section.



And now, let's create a button.

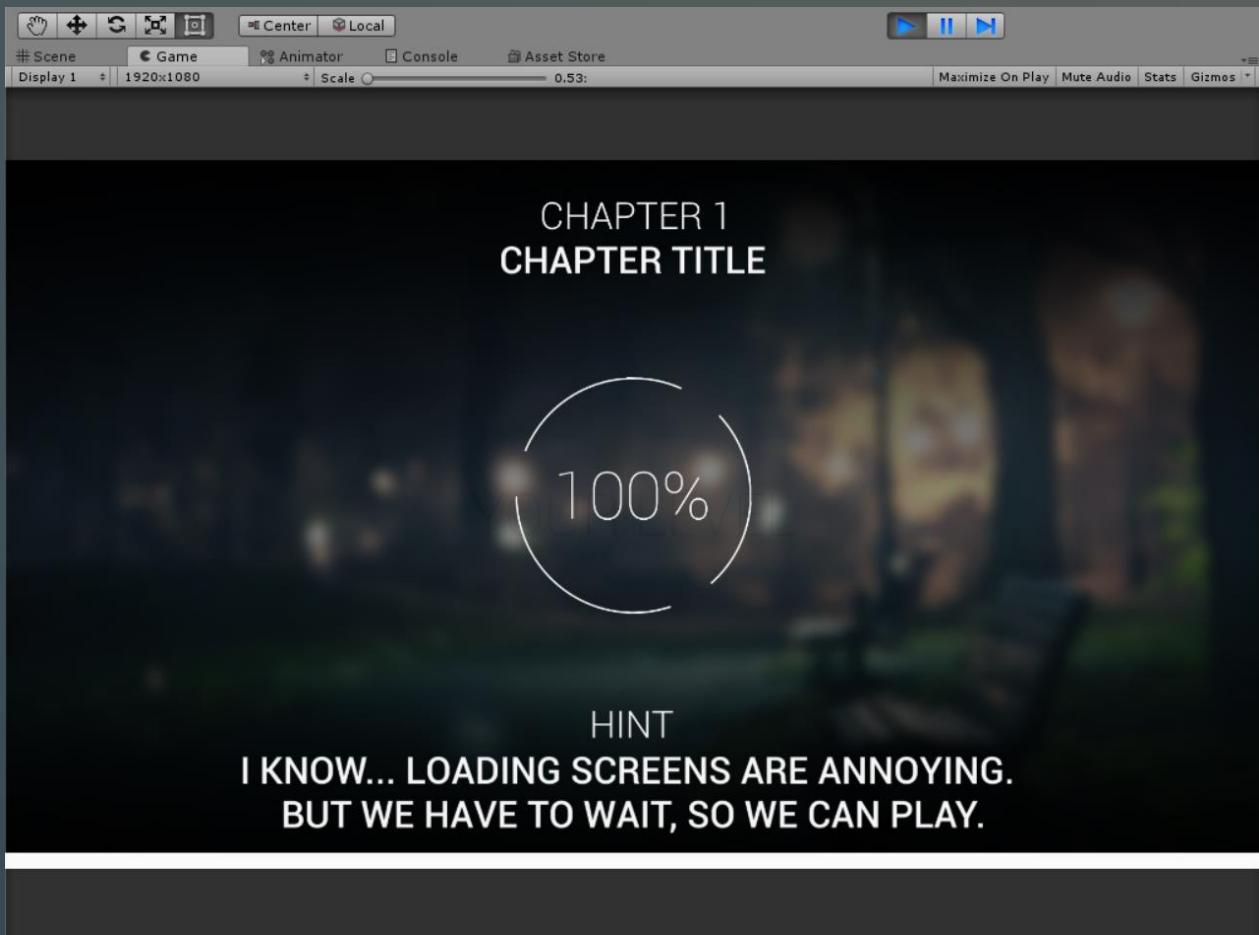
Add 2 **OnClick** events to the button, and fill in as shown.



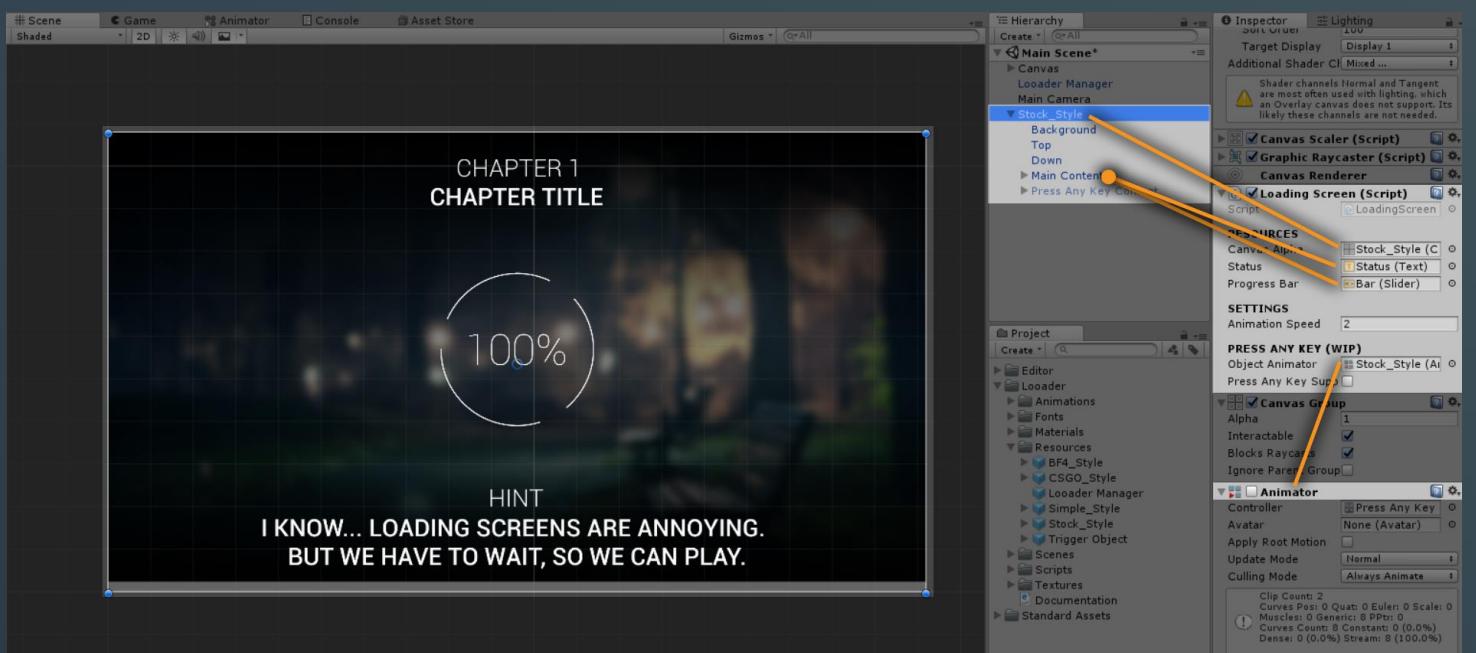
**SetStyle** = Your loading screen prefab name.  
You can type any prefab as long as it's in  
Loader\Resources folder.

**ChangeToScene** = Name of the scene to be loaded.  
And don't forget to add your scenes to the build window.

That's it! You can click and enjoy your loading screen (or not).

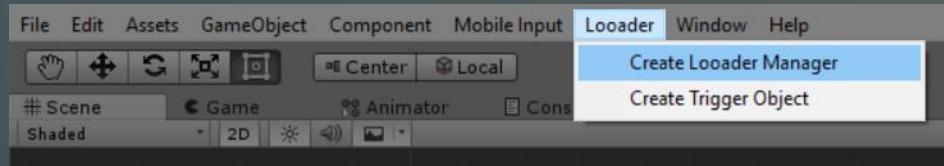


Now, you can customize your loading screen if you want. Prefabs are in [Looader\Resources](#) folder. Drag it to the scene, change as you like, [Apply](#) and you're good to go!

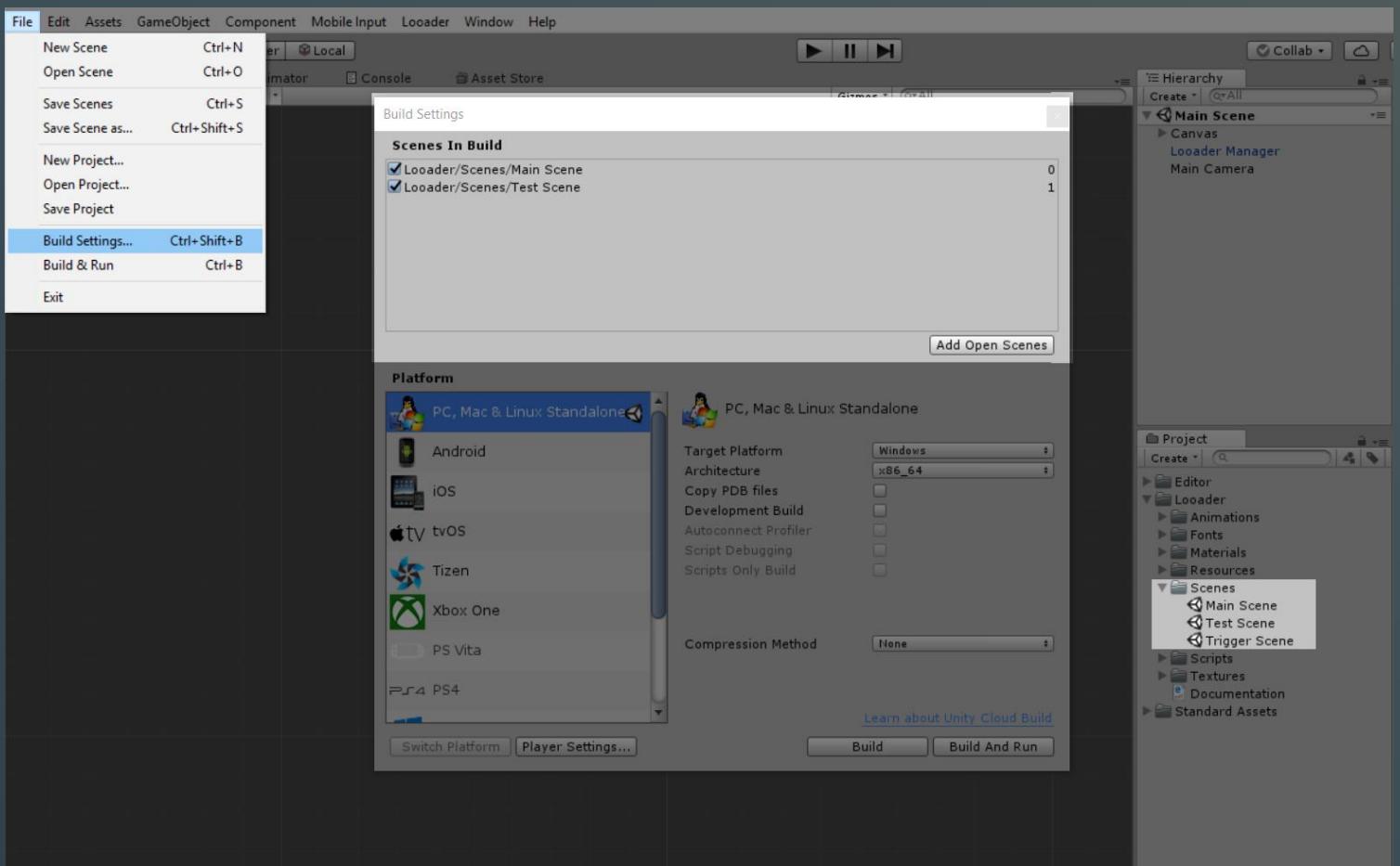


# Preparing the loading screen

First, we need to create Looader Manager object.  
You can easily create it in the editor window.



Now, go to Build Settings and drag your scenes to Scenes In Build section.



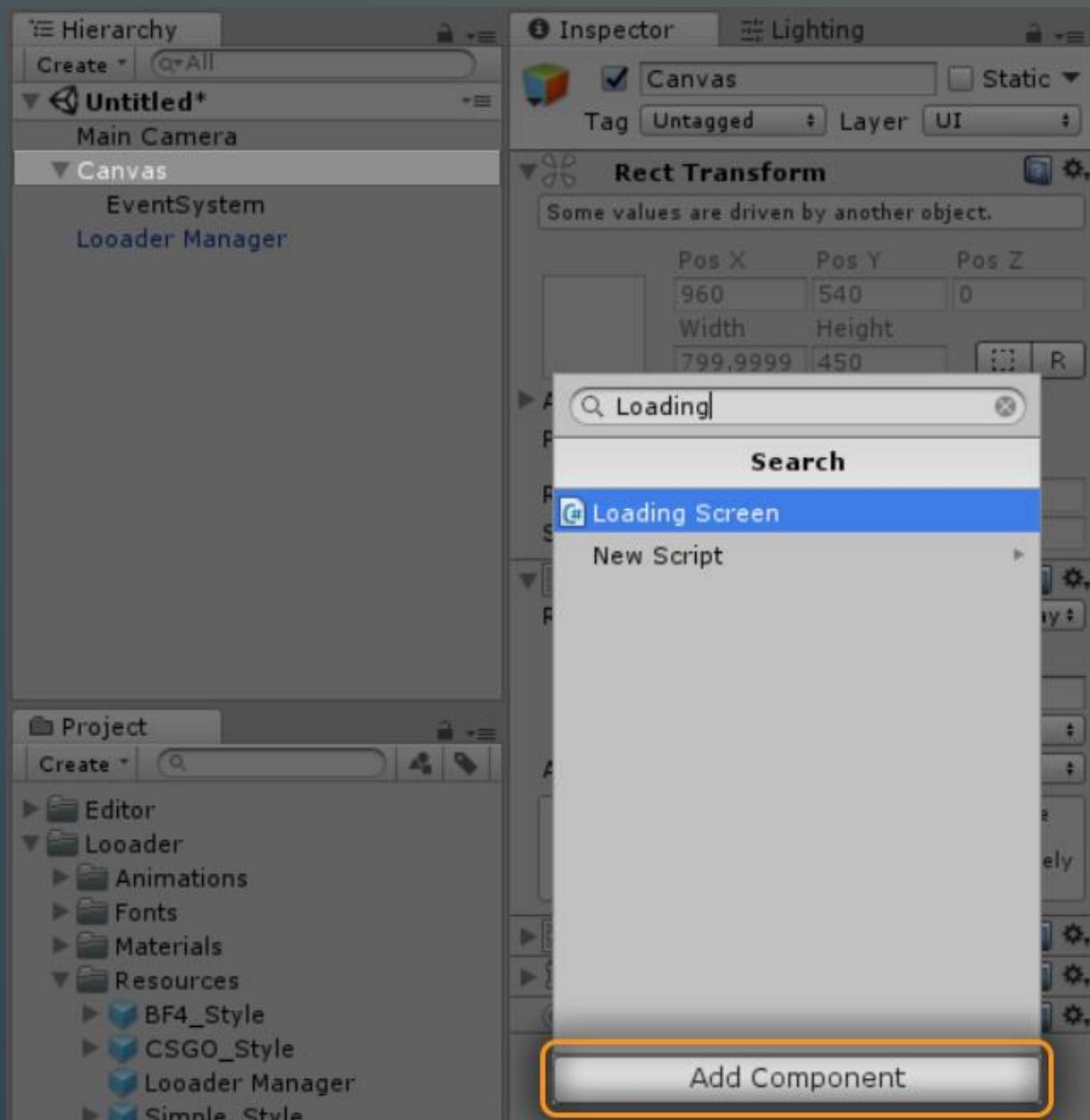
Then let's create a canvas.

Hierarchy > Create > UI > Canvas

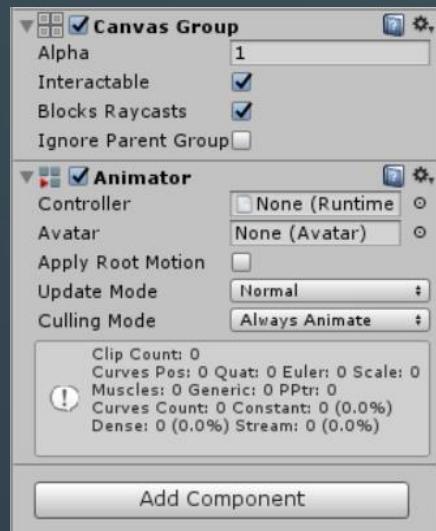
or

Editor Window > GameObject > UI > Canvas

Add LoadingScreen script to your canvas.

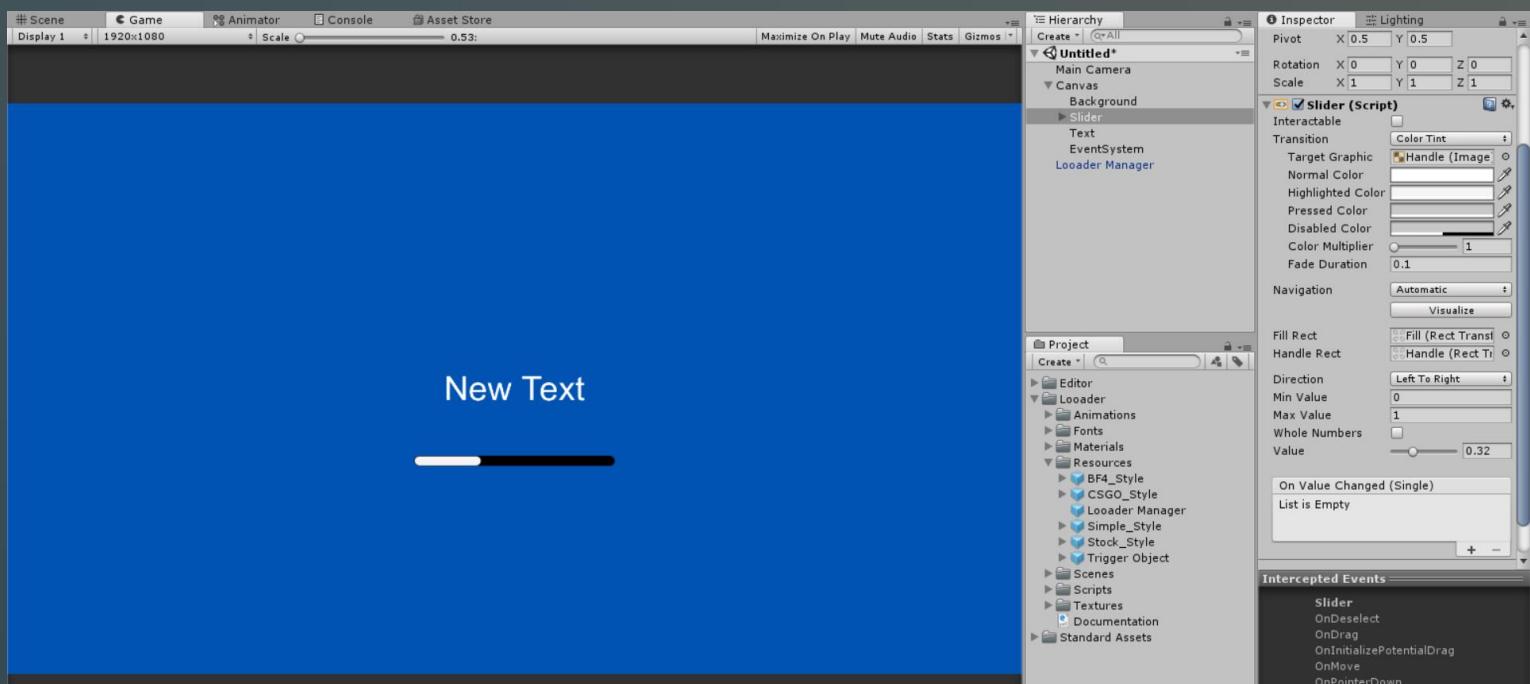


After that, add **Canvas Group** and **Animator** components.



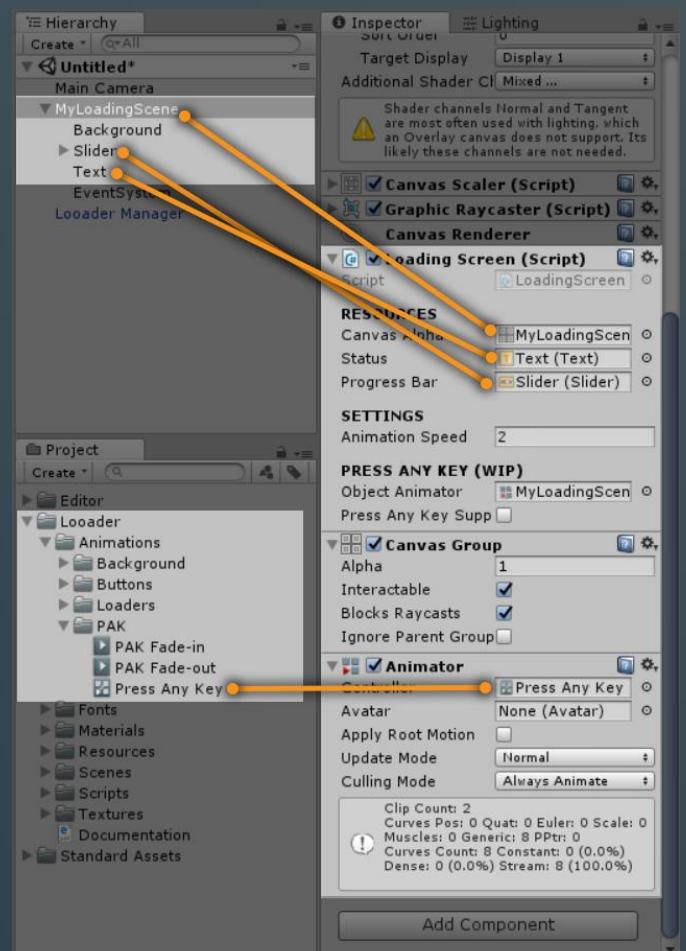
Now, let's create a **slider** and a **text** object.

Customize the **slider** and **text** as you like. And change **Interactable** value of slider to **false**. And add a **image** object for background. It should look like this after all.



Then, select and rename **Canvas** as **MyLoadingScreen**. Fill in **LoadingScreen** variables as shown.

You can change **Animation Speed** or variables as you like.

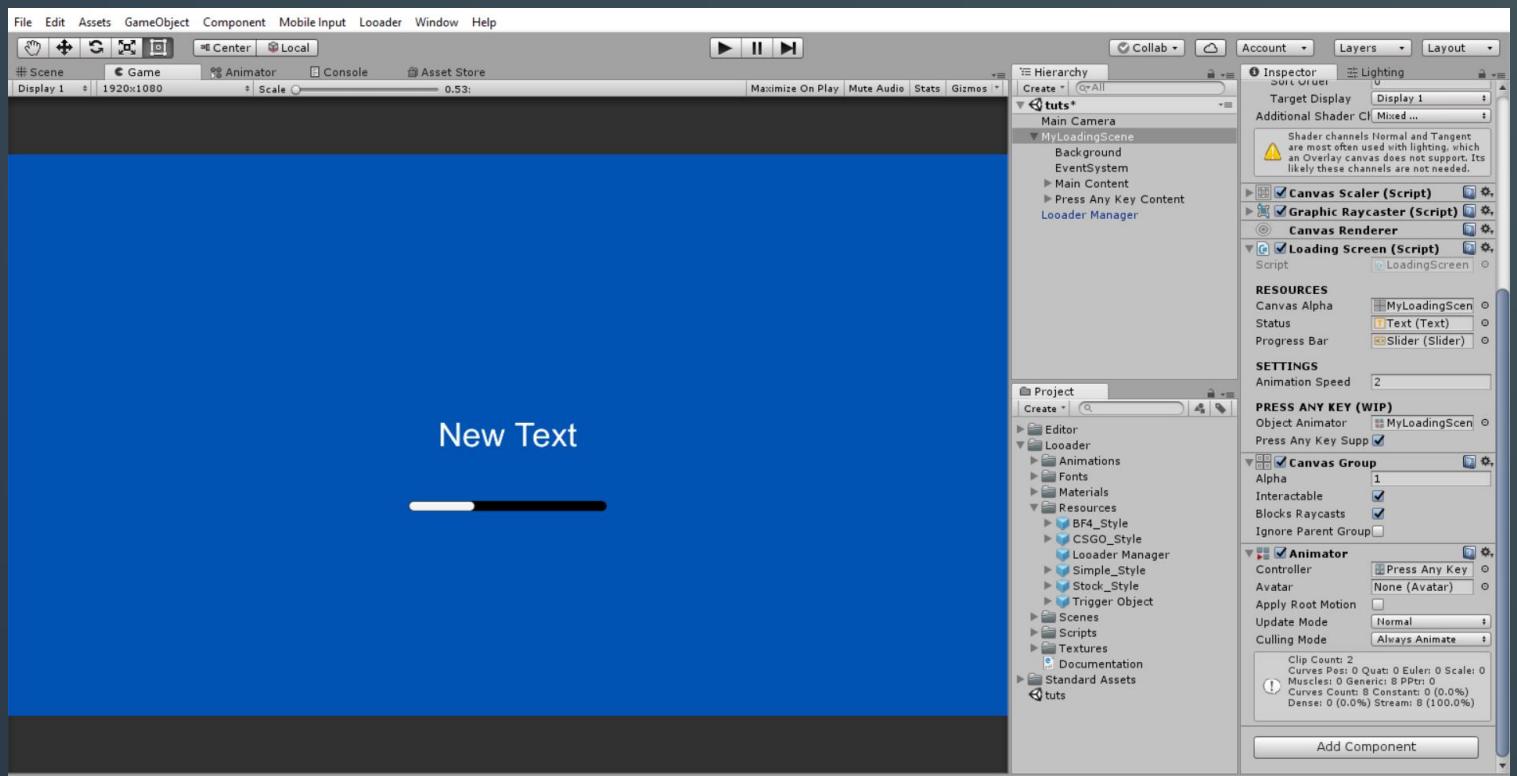


If you want Press Any Key function, just check Press Any Key Support checkbox from MyLoadingScreen. Then create a panel, delete image component from the panel, add canvas group to it, and duplicate it. Rename as Main Content and Press Any Key Content.

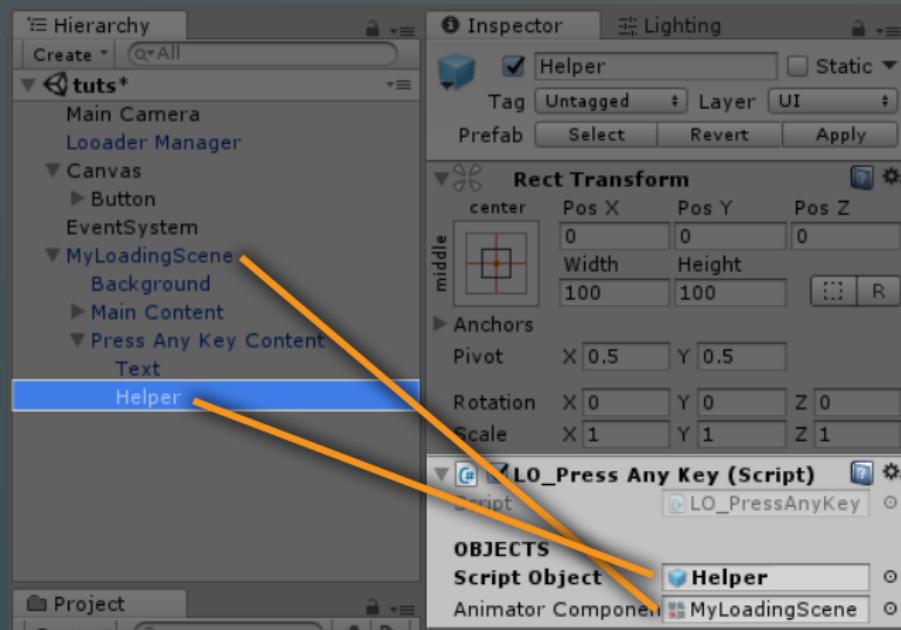
If you want, you can add some content to Press Any Key Content panel, such as "Press Any Key" text. Drag all objects -except Background- to Main Content panel (or drag it too, your choice). And disable Animator component.

Lastly, disable Press Any Key Content object.

After all, it should look like this:

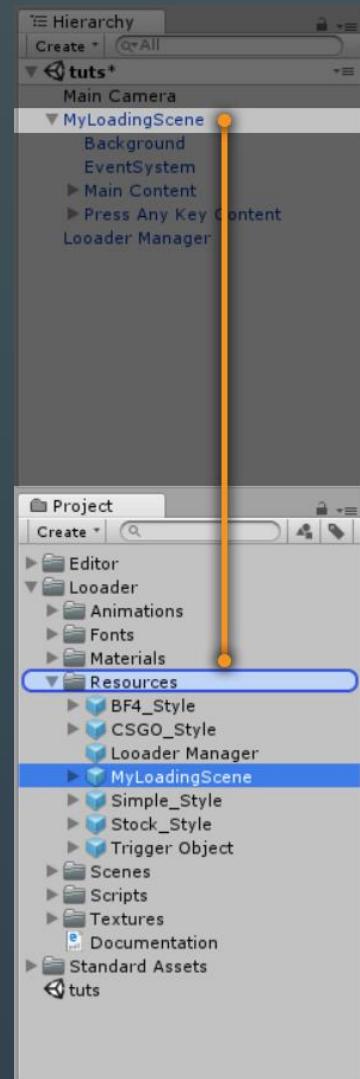


Let's finish PAK feature.  
Create an empty object and rename it as Helper. Then add Press Any Key component to it. Fill the blanks as shown.

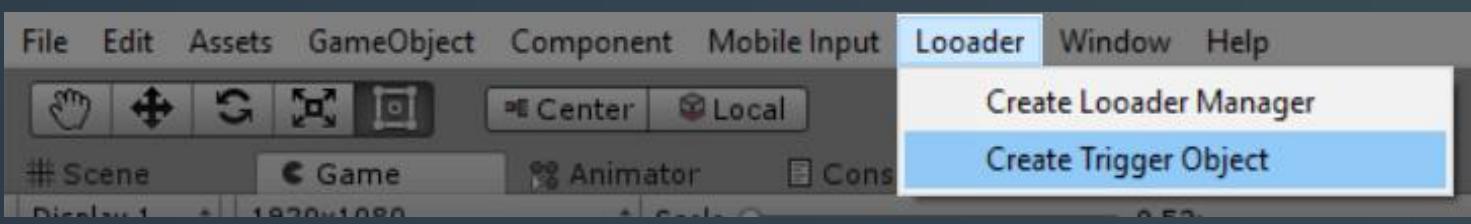
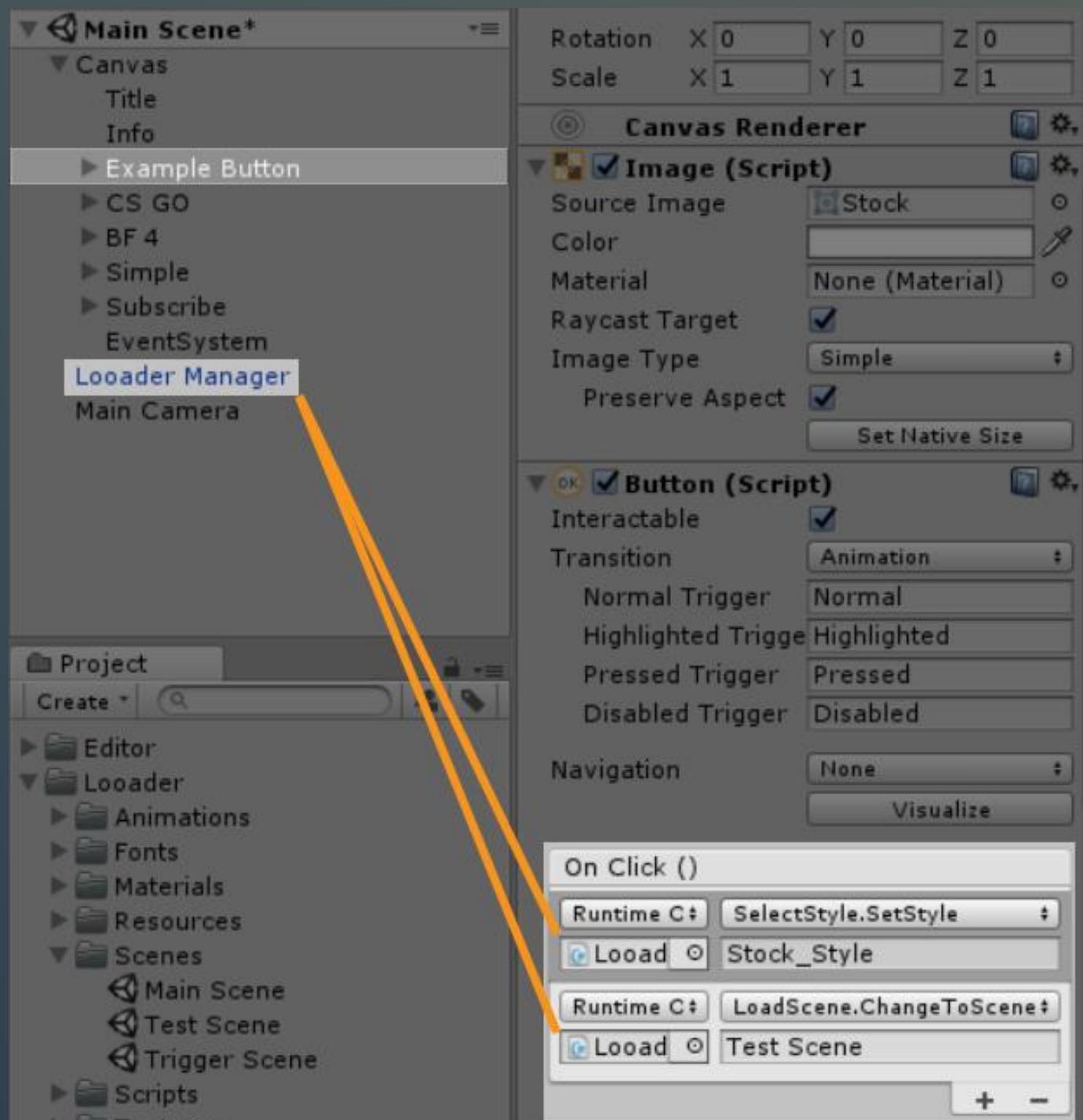


Now, let's make this a prefab.

Drag MyLoadingScreen object to Loader\Resources.



Okay, Loading Screen is done.  
You can call it with a button, or a trigger.  
It's up to you.



More information and tutorials coming soon! I don't have a lot free time, so it's hard to make these things :)

You can contact me via