

Level Menu 2D



Ever wondered how beautiful Episode/Level Select menus are developed in Awesome Mobile Games?

Or how Character/Weapon Selection Sliding Menus are developed?

Or you want to make next big hit Indie Game with a beautiful Level/Theme/Episode/Character or custom selection menu in your game?

Welcome to Level Menu 2D, a Unity asset bundle for creating level selection menu in your game. As a developer, you can spend your time on being creative in making the next big hit Indie game, while Level Menu 2D works out the boring parts like managing selection variables and codes, keeping track of the currently selected, next, and previous levels, animate objects while navigation, and many more.

THIS DOCUMENT COVERS THE BASICS OF HOW TO USE LEVEL MENU 2D TO CREATE AWESOME MENUS FOR YOUR NEXT BIG HIT INDIE GAME.

YOU CAN SEE MORE DETAILED DOCUMENTATION, WEB DEMO, VIDEO TUTORIALS ON FOLLOWING LINK:

<http://wajahatkarim.com/portfolio/level-menu-2d/>

Table of Contents

Introduction	3
Installation and General Usage Consideration	3
Terminology	3
Creating Menu from Level Menu 2D using Unity Sprites.....	4
Putting Next/Back Buttons in the Menu	4
Adjust Menu for Offset, Spacing, and other settings.....	4
Calling any method on Item Click using LevelMenu2D	5
Manually navigation to next/previous or to any specific item in LevelMenu2D using Swipe Gestures	5

Introduction

Level Menu 2D is a Unity asset bundle for creating level selection menus in your game. This reference documentation is a detailed descriptions of all features of Level Menu 2D. If you just want to get a quick start, we recommend you to watch the video tutorial at [video tutorial section](#). You should also check the [API docs](#) for full information about the available classes and members.

Installation and General Usage Consideration

The Level Menu 2D package is installed from the [Unity Asset Store](#) using the Asset Store window of unity. After you have bought the Level Menu 2D package at the Unity Asset Store you can install the package into your project by clicking the Import button next to the Level Menu 2D package. This will create a folder in your project: `LevelMenu2D`. This folder holds all the resources required for using `LevelMenu2D`. All classes are being delivered as source code and will be compiled by Unity.

We opted for providing the source code instead of some precompiled DLL, so you can make the most of `LevelMenu2D` and integrate it smoothly into your game. You are free to extend any of the provided classes to make them work according to your specific needs. In general we tried to anticipate the most common situations where you would want more control and provided means to change the behavior of `LevelMenu2D` without the need to modify existing classes.

Terminology

Within this documentation we use the following terminology:

- **Item** is a Unity Game Object. It can be Unity Sprite or game object with nested objects in it. Level Menu 2D uses items for creation of menus.
- **LevelMenu2D** is a singleton class and Prefab object, and the whole menu can be accessed from it. It contains methods like `gotoNextItem()`, `gotoBackItem()`, `indexOf(GameObject itemObject)`, and so on.

Creating Menu from Level Menu 2D using Unity Sprites

You begin by installing the Level Menu 2D package from the Unity Asset Store into your project. The package contains a folder named `LevelMenu2D` which holds all required code and assets.

*It must be noted that `LevelMenu2D` uses [*iTween*](#) for animation purposes. You must import [*iTween*](#) from the Asset Store to run the `LevelMenu2D` built menu. You can find *iTween* on Asset Store at following [link](#).*

Also, you must have to import [Singleton Perfect](#) from Asset Store at following [link](#).

Drag `LevelMenu2D` prefab into your scene. The property inspector of `LevelMenu2D` will look like the image on right:

- Insert number of items in `Size` attribute of `Items List`.
- Drag item game objects in the `Items List` array.
- Set `Orientation` to `Horizontal` or `Vertical`.
- Run the game, and your menu will be working.

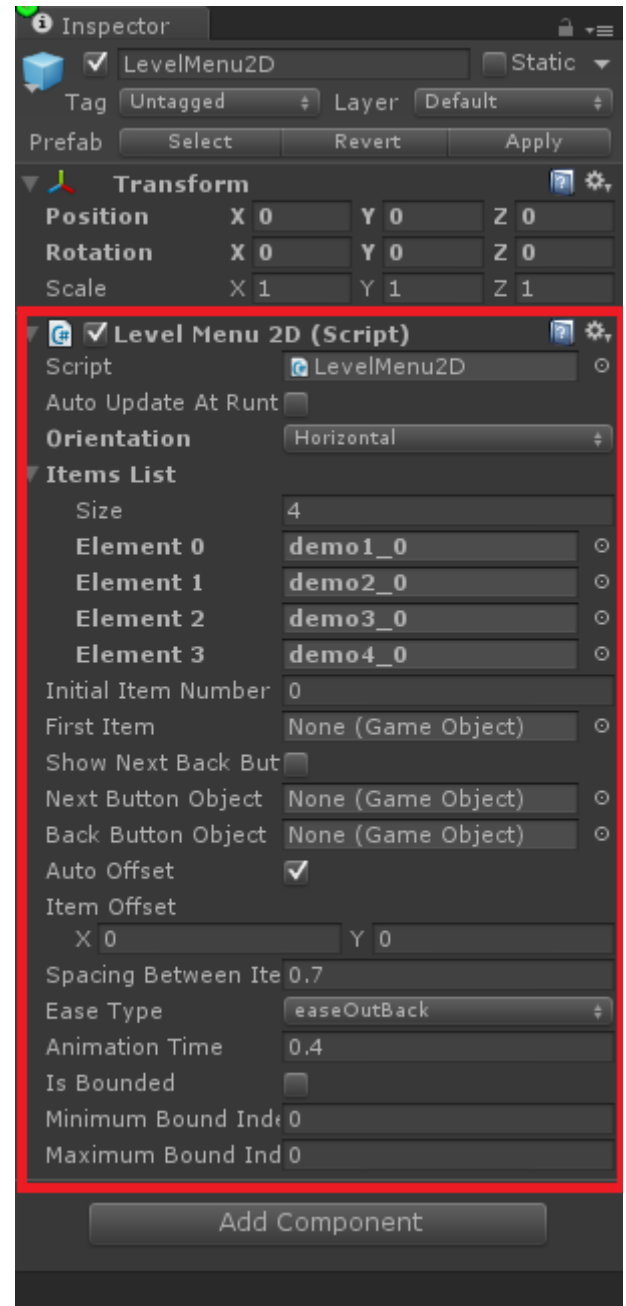
Putting Next/Back Buttons in the Menu

Drag `LevelMenu2D` prefab into your scene. The property inspector of `LevelMenu2D` will look like the image on right:

- Insert number of items in `Size` attribute of `Items List`.
- Drag item game objects in the `Items List` array.
- Set `Orientation` to `Horizontal` or `Vertical`.
- Check `Show Next Back Buttons` to `True`.
- Drag next and back button objects into game objects in the `Next Button Object` and `Back Button Object`.
- You are set to go. These buttons will start working on click. Run the game, and enjoy.

Adjust Menu for Offset, Spacing, and other settings.

Drag `LevelMenu2D` prefab into your scene. The property inspector of `LevelMenu2D` will look like the image on right:



- Insert number of items in `Size` attribute of `Items List`.
- Drag item game objects in the `Items List` array.
- Set `Orientation` to `Horizontal` or `Vertical`.
- Check `Auto Update at Runtime` to `True`.
- Run the game, and your menu will be working.

Now, when you will update any settings in prefab inspector of `LevelMenu2D`, you will see those changes updated at runtime. Adjust your settings accordingly, and copy those settings. Stop the game, and paste those values in `LevelMenu2D` inspector settings.

Note: It must be noted that Auto Updating affects the performance very badly and you should check it to false when building release version of your game.

Calling any method on Item Click using `LevelMenu2D`

Drag `LevelMenu2D` prefab into your scene. The property inspector of `LevelMenu2D` will look like the image on right:

- Insert number of items in `Size` attribute of `Items List`.
- Drag item game objects in the `Items List` array.
- Set `Orientation` to `Horizontal` or `Vertical`.
- Run the game, and your menu will be working.
- Create any empty game object in the scene, and add a script in it (let's call it `ClickerScript.cs`).
- To get Item Click, we will have to listen for `OnItemClick` delegate event.

Following code shows `ClickerScript.cs`:

```
public class ClickerScript : MonoBehaviour {

    void Awake()
    {
        LevelMenu2D.I.OnItemClicked += OnItemClicked;
    }

    void OnItemClicked (int itemIndex, GameObject itemObject)
    {
        Debug.Log("Item Clicked: " + itemIndex + " Name: " + itemObject.name);
    }
}
```

Manually navigation to next/previous or to any specific item in `LevelMenu2D` using `Swipe Gestures`

`LevelMenu2D` comes with built-in `SwipeDetector`. To get started, first drag `LevelMenu2D` prefab into your scene. The property inspector of `LevelMenu2D` will look like the image on right:

- Insert number of items in `Size` attribute of `Items List`.
- Drag item game objects in the `Items List` array.

- Set Orientation to Horizontal or Vertical.
- Run the game, and your menu will be working.

Now whenever or wherever you want to navigate to the menu, all you have to do is to access the singleton instance of LevelMenu2D and call methods from it. **(It must be noted that there should be only instance of LevelMenu2D in one scene.)** LevelMenu2D will automatically detect scene's menu and navigate it.

The built-in Swipe Detector delegates events when any swipe is detected, and we have to listen for those delegate events in our code.

Following code snippet shows finger gesture swiping navigation in file `NavigatorScript.cs`:

```
public class NavigatorScript : MonoBehaviour {

    void Awake()
    {
        LevelMenu2D.I.OnItemClicked += HandleOnItemClicked;
        SwipeDetector.OnSwipeLeft += HandleOnSwipeLeft;
        SwipeDetector.OnSwipeRight += HandleOnSwipeRight;
    }

    void HandleOnItemClicked (int itemIndex, GameObject itemObject) {
        Debug.Log("Item Clicked: " + itemIndex + " Name: " +
itemObject.name);
    }

    void HandleOnSwipeRight ()
    {
        LevelMenu2D.I.gotoBackItem();
    }

    void HandleOnSwipeLeft ()
    {
        LevelMenu2D.I.gotoNextItem();
    }
}
```