

# Overview UI Sprite Manager

## Introduction

Thanks for downloading this package stay tuned this is just the beginning or you may call it proof of concept more features are in the queue.

After spending some years in Game Industry, I realize managing sprites in large projects is a headache, as artists have to iterate images a lot to make them better, cool and according to demand. While developers have to integrate those images and to show the feel of those images in the game.

Also, due to this hassle, artists and developers choose a messy way around, and instead of working in spritesheet, they start working in individual sprites.

## Purpose

In big Unity projects, artists have to modify images frequently to make it better, and developers have to reflect that change, as the old reference becomes Missing (if the name of the updated sprite is not the same)

## Unity Asset Store Link

<https://www.assetstore.unity3d.com/#!/content/89387>

## GitHub Link

<https://github.com/mohsinkhan26/UISpriteManager>

## BitBucket Link

[https://bitbucket.org/unbounded-eagle/ui\\_sprite\\_manager/](https://bitbucket.org/unbounded-eagle/ui_sprite_manager/)

## Blog Link

<http://cslearners.blogspot.com/2017/04/unity3d-ui-sprite-manager.html>

## Features

- Easily make reskin of your game, if the game is made using UI Sprite Manager
- Promotes the use of spritesheet/atlas
- Get full use of spritesheet and packaging tag
- Place images anywhere in Assets/, as if the images are placed in Assets/Resources packaging tag don't work

- Still you can change the sprite on runtime with ImageExtension public methods
- Easy to use, Plug n play
- Open Source code without any DLL
- Done by extending Image component

## **Setup**

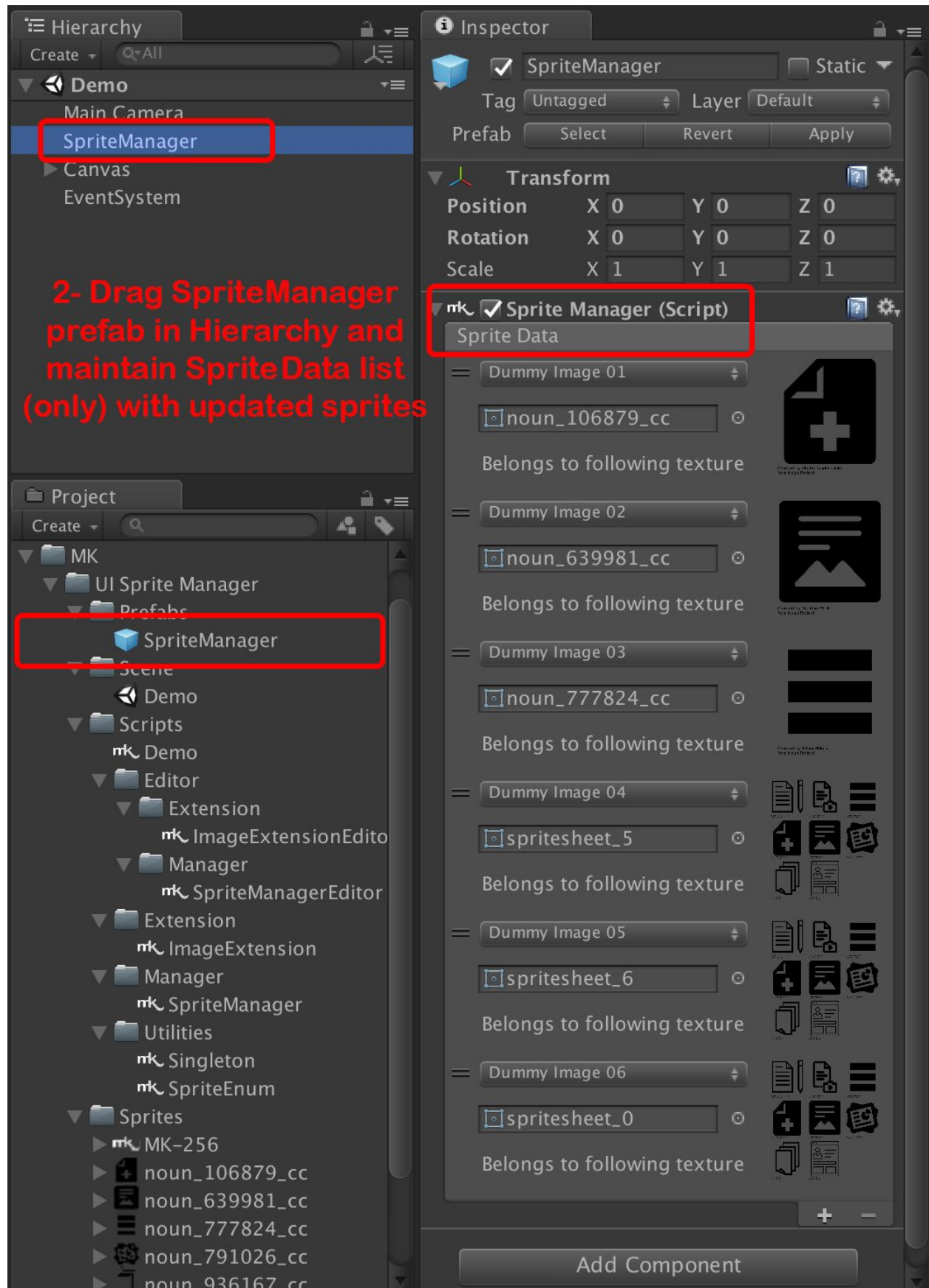
- Import plugin
- Add enum against your sprite in SpriteEnum.cs
- Add respective sprite in SpriteManager.prefab
- Now, just add component ImageExtension.cs instead of Image in Unity UI and use it as is

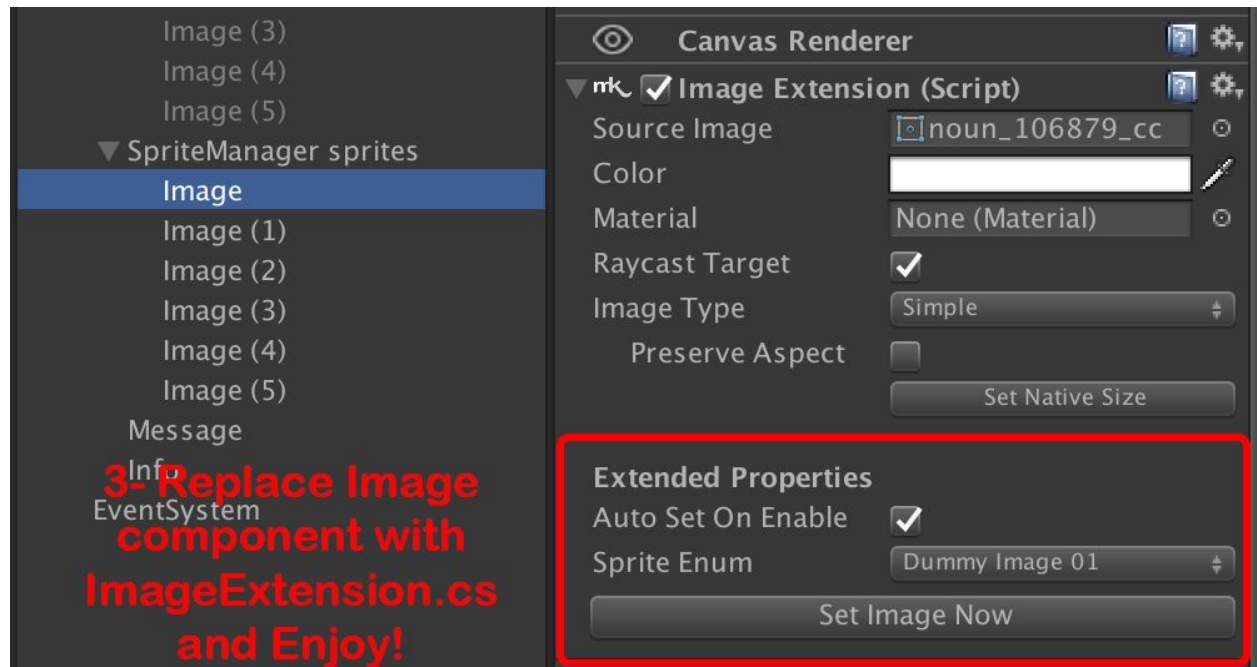
SpriteEnum.cs

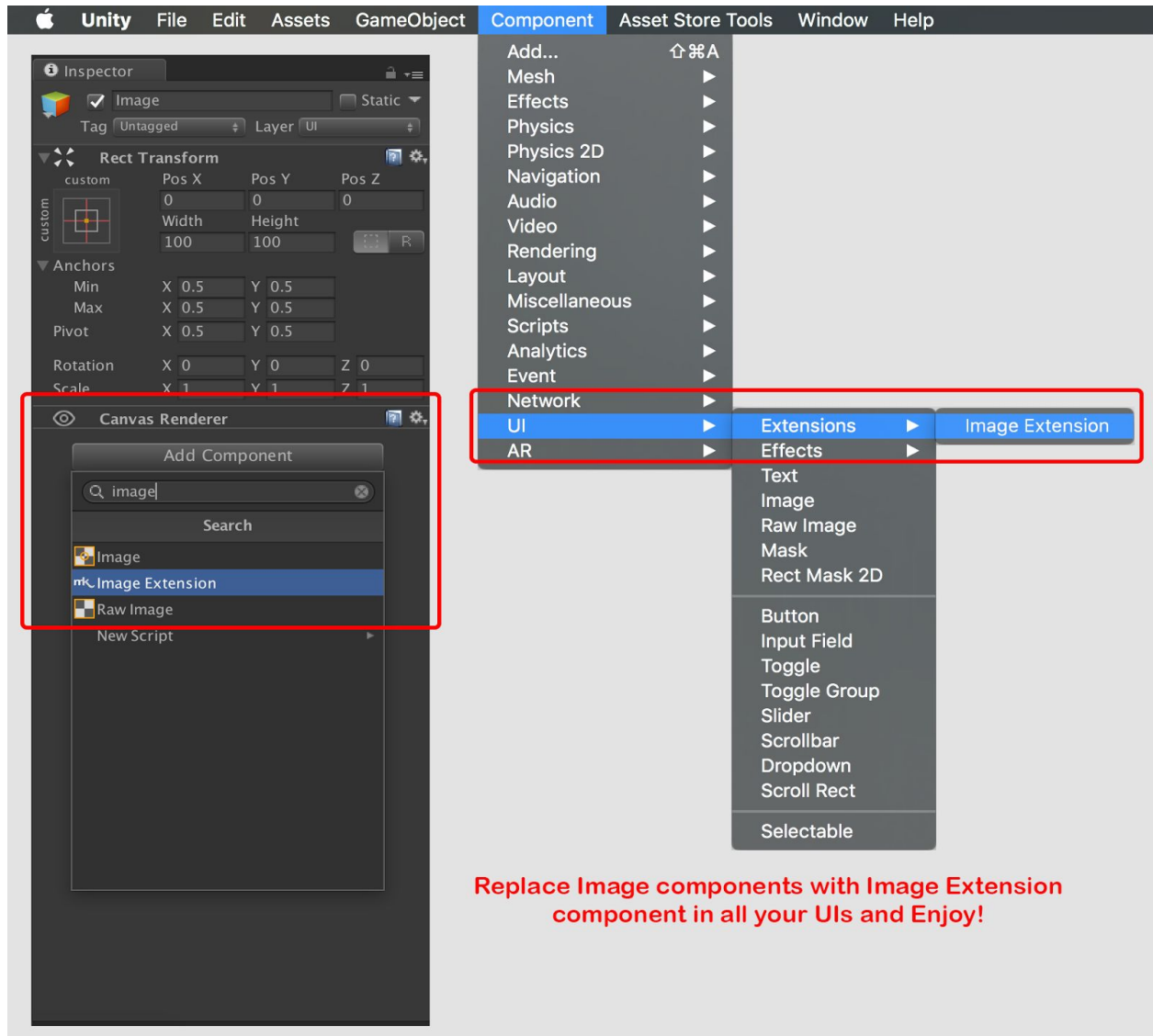
SpriteEnum ▶ No selection

```
1 namespace MK.Utilities
2 {
3     public enum SpriteEnum
4     {
5         None = 0,
6         Menu_Background,
7         Levels_Background,
8         Shop_Background,
9         Play_Button,
10        Back_Button,
11        Forward_Button,
12        dummyImage01,
13        dummyImage02,
14        dummyImage03,
15        dummyImage04,
16        dummyImage05,
17        dummyImage06,
18    }
19 }
20
```

1- Add sprite  
respective  
enum name  
here







## Remember

- Drag SpriteManager.prefab in the first scene and it will persist

## Special Thanks

- All the users who provide feedback and suggestions to improve
- All the users who gave reviews on Asset store

**Thanks for your support!**