HYPER BYTE

STUDIOS

UI THEME SETUP GUIDE

TABLE OF CONTENT

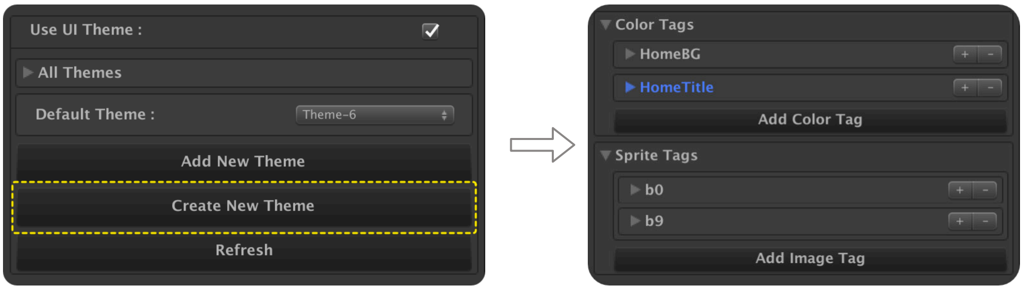
INTRODUCTION

Hyperbyte Game template comes with pre-installed ui theme integration and it can be configured with very few simple steps. Please follow this guide to setup UI Themes. Setting UI Theme consist of 2 parts, Creating UI Theme and Configuring a UI Theme.

Creating a Theme

To get started with configuration, please open UI Theme Settings from **Hyperbyte > UI Theme Settings** menu item inside the unity editor. It will look like the image attached. Check the Use UI Theme Toggle Button and Press the **Create New Theme** button.

Upon Pressing a Create New Theme Button, a new Scriptable object editor will popup, UI Theme consists of 2 parts. It will look like attached image.



Use UI Theme : ☑

Default Theme : Theme-6

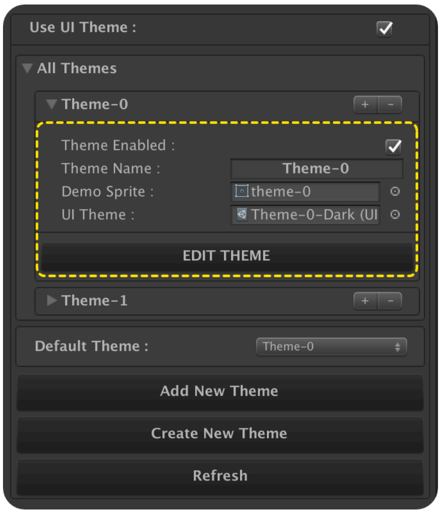
- **Color Tags** are simply a color configuration. It works like key and value. Tag is a key and color is the color associated with the tag. Please configure all required color tags and its color.
- **Sprite Tags** contains a sprite image with its tag. Please configure as many as needed.

Note: *Please make sure all the UI Themes have the same color tags and sprite tags to avoid errors.*

Configuring UI Theme

Once UI Themes are created, please configure those themes on UI Theme Setting. For each created UI Theme, please add configuration.These configuration includes,

- **Theme Enabled :** Checkbox to toggle theme activation.
- **Theme Name :** Name of the theme configuration.
- **Demo Sprite :** Assign a demo sprite. Demo sprite will show in the theme selection screen. Theme selection screen can be further configured from **MainScene ~> Canvas-Popups ~> SelectTheme popup.**
- **UI Theme :** Assign the UI Theme created with Color Tags and Sprite Tags.



Use UI Theme : ☑

Theme Enabled : ☑

Theme Name : Theme-0

Demo Sprite : theme-0

UI Theme : Theme-0-Dark (UI

Default Theme : Theme-0

Using UI Themes

UI Theme can be used with different component types like, image, text, sprite etc. UI Theme can be used by adding component of target type or by using different methods to access associated color tags or sprite tags. Please follow below details for using UI Themes.

- **Image Color Tag** : If any UI Image needs to override its color with change of different UI Theme please Add **ImageColorTag** script component and assign target color tag which is added in UI Theme.

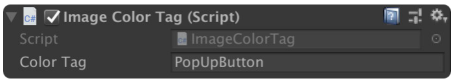
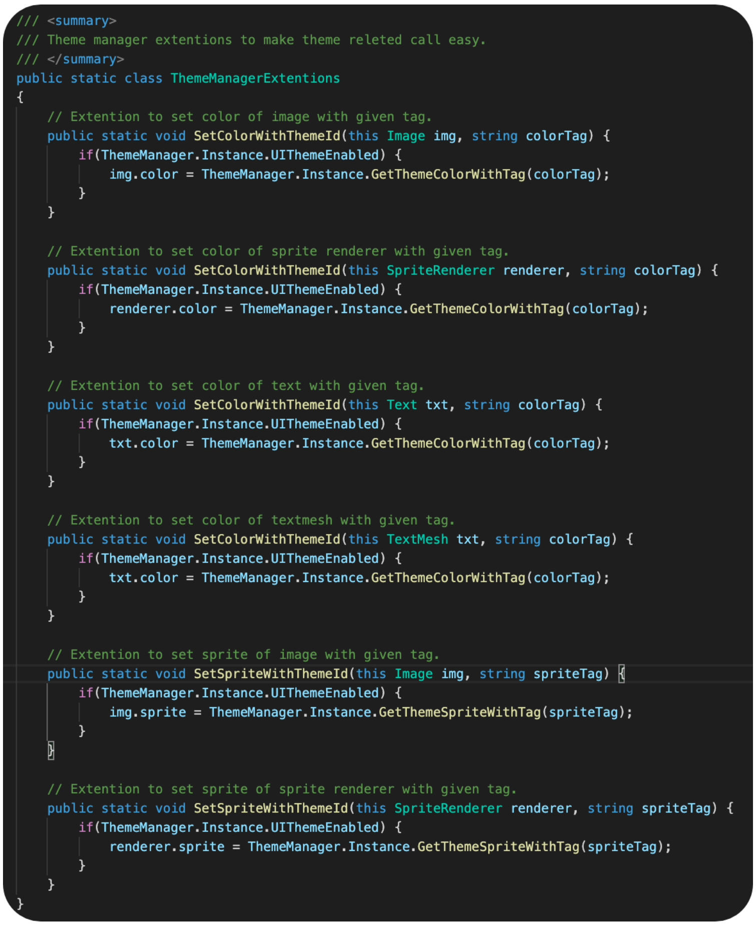
  

  Image Color Tag (Script)
  Script ImageColorTag
  Color Tag PopUpButton

  In the similar way to **ImageColorTag,** there are few more components can be used in the exact same way. There are few pre-built component included like ImageSpriteTag, SpriteRendererColorTag, SpriteRendererSpriteTag, TextColorTag, TextMeshColorTag. It can be used with relative component based on requirements.

- **Technical Info:** UI Theme tool contains all the methods and classes to handle and use UI Themes, fetching associated color or sprite with given tag and assign to the object.

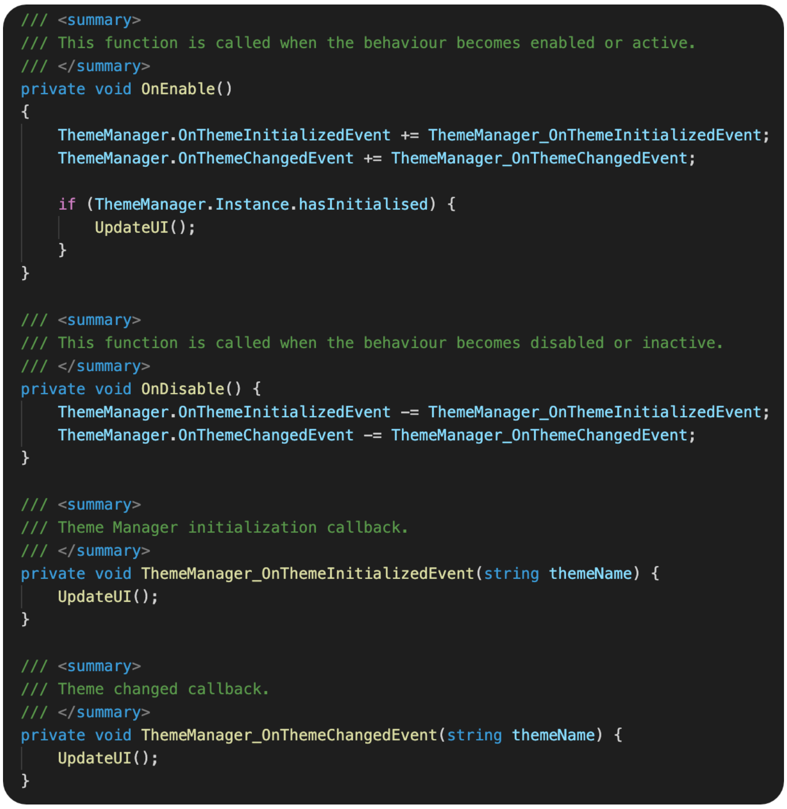
  Please see **ThemeManager.cs** further customization and also see below code snippet to see supported extensions with UI Themes.



```
/// <summary>
/// Theme manager extentions to make theme releted call easy.
/// </summary>
public static class ThemeManagerExtentions
{
    // Extention to set color of image with given tag.
    public static void SetColorWithThemeId(this Image img, string colorTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            img.color = ThemeManager.Instance.GetThemeColorWithTag(colorTag);
        }
    }

    // Extention to set color of sprite renderer with given tag.
    public static void SetColorWithThemeId(this SpriteRenderer renderer, string colorTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            renderer.color = ThemeManager.Instance.GetThemeColorWithTag(colorTag);
        }
    }

    // Extention to set color of text with given tag.
    public static void SetColorWithThemeId(this Text txt, string colorTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            txt.color = ThemeManager.Instance.GetThemeColorWithTag(colorTag);
        }
    }

    // Extention to set color of textmesh with given tag.
    public static void SetColorWithThemeId(this TextMesh txt, string colorTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            txt.color = ThemeManager.Instance.GetThemeColorWithTag(colorTag);
        }
    }

    // Extention to set sprite of image with given tag.
    public static void SetSpriteWithThemeId(this Image img, string spriteTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            img.sprite = ThemeManager.Instance.GetThemeSpriteWithTag(spriteTag);
        }
    }

    // Extention to set sprite of sprite renderer with given tag.
    public static void SetSpriteWithThemeId(this SpriteRenderer renderer, string spriteTag) {
        if(ThemeManager.Instance.UIThemeEnabled) {
            renderer.sprite = ThemeManager.Instance.GetThemeSpriteWithTag(spriteTag);
        }
    }
}
```

If case there is requirements for further customization with extensions, please add those to **ThemeManagerExtentions.cs**

- **Callbacks** : Its important to detect theme initialization, change etc callbacks to handle code based on the changes. Please see the below code snippet for attached to for implementing and using ui theme related event callbacks.



```
/// <summary>
/// This function is called when the behaviour becomes enabled or active.
/// </summary>
private void OnEnable()
{
    ThemeManager.OnThemeInitializedEvent += ThemeManager_OnThemeInitializedEvent;
    ThemeManager.OnThemeChangedEvent += ThemeManager_OnThemeChangedEvent;

    if (ThemeManager.Instance.hasInitialised) {
        UpdateUI();
    }
}

/// <summary>
/// This function is called when the behaviour becomes disabled or inactive.
/// </summary>
private void OnDisable() {
    ThemeManager.OnThemeInitializedEvent -= ThemeManager_OnThemeInitializedEvent;
    ThemeManager.OnThemeChangedEvent -= ThemeManager_OnThemeChangedEvent;
}

/// <summary>
/// Theme Manager initialization callback.
/// </summary>
private void ThemeManager_OnThemeInitializedEvent(string themeName) {
    UpdateUI();
}

/// <summary>
/// Theme changed callback.
/// </summary>
private void ThemeManager_OnThemeChangedEvent(string themeName) {
    UpdateUI();
}
```

WRAP UP

- While we tried to involve most stuff in the documentation, though, in case if you have any further trouble, be sure to get in touch and will assist on the issue.
- If you liked asset content and quality, be sure to share your feedback on the asset store. Also, be sure to get in touch for any further queries, suggestions, bugs, or feature requests.
- Website : https://www.hyperbytestudios.com
  Email : support@hyperbytestudios.com

THANKS