

```
//
////////////////////////////////////
Author: Chinh Nguyen
Email: chinh.dinang@gmail.com
Social Networ: https://twitter.com/DucChinhNguyen2
//
```

2. Initialize a scene

1. Introduction

VKSDK will help you implement smooth UI/UX flows like no other tool on the asset store has.

VKLayerController vs **VKLayer**. **VKLayerController** controls the layers and the order in which they appear. **VKLayer** is a base layer that allows you to inherit and customize the parameters of a layer.

VKLayer is a base layer that allows you to inherit and customize the parameters of a layer.

- Go to File/New Scene

- Go to File/New Scene

- Go to GameObject/Create Empty. Change the name **VKLayerController**

- Add component **VKLayerController**

-Config VKLayerController following :

- + Ui Camera: when you select camera rendermode for canvas,

you drag an orthographic camera to here.

+ VKLoading: Create a GameObject/AddComponent VKLoading. You drag the game object to here.

- + firstLayer: the name of the first layer you want to show.

+ **currentLayer**: the name of the highest layer display on screen.

+ path: the path to get prefabs in Resources Folder. Here, I save

- + deep order: the space between the 2 layer by sorting order.

Example: deep order=5, you show layer1 with sorting order 1, the sorting order of next layer2 $1+5=6$

+ deep plane distance: the space between the 2 layer by distance to camera or z.

Example: deep plane distance=50, you show layer 1 with distance to camera 590,
the distance of the next layer 2: 540

+ deep order starts: the framework group layers in 3 type bottom, middle, top.

If you want to a layer hide a part of another layer.

+ plane distance starts: the framework group layers in 3 type bottom, middle, top.

These are distances from layer to camera of 3 type of layer bottom, middle, top.

+ Landscape: true/false. Your layers are landscape or portrait

+ Ready Layers: the layers which have been already included in scene

+ Prefabs Layers: the layers which have saved in Asset folder

- The main base class is VKLayer. You only create a script override VKLayer for each a layer.

```
public class DemoClass : VKLayer{
```

}

- You show a layer using API ShowLayer():

```
VKLayerController.Instance.ShowLayer("DemoClass");
```

let to make above, you create a prefab which add component DemoClass

Drag DemoClass prefab in the path to get prefabs in Resource

- You want to convert VKLayer to DemoClass? You must use:

```
DemoClass demoClass=(DemoClass)VKLayerCon
```

- You can add event layer enable in API EnableLayer()

- You can add event layer load the first in API FirstLoad

- You can add event reload layer in API ReloadLa

- You can add event hide layer in API HideLayer()

- You can add event `OnLayerOpenDone()`, `OnLayerOpenPopupDone()` when you use `Ar`

