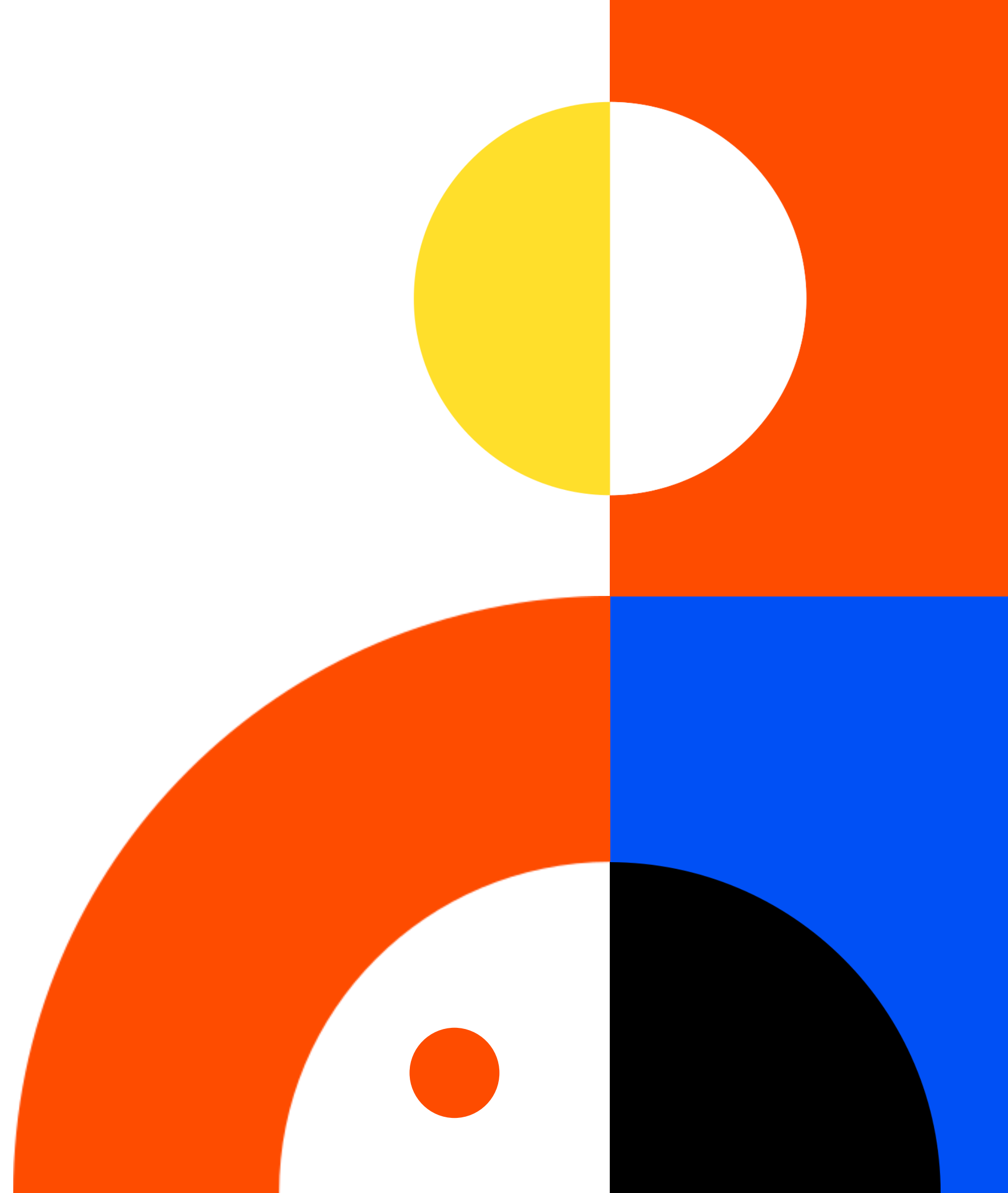
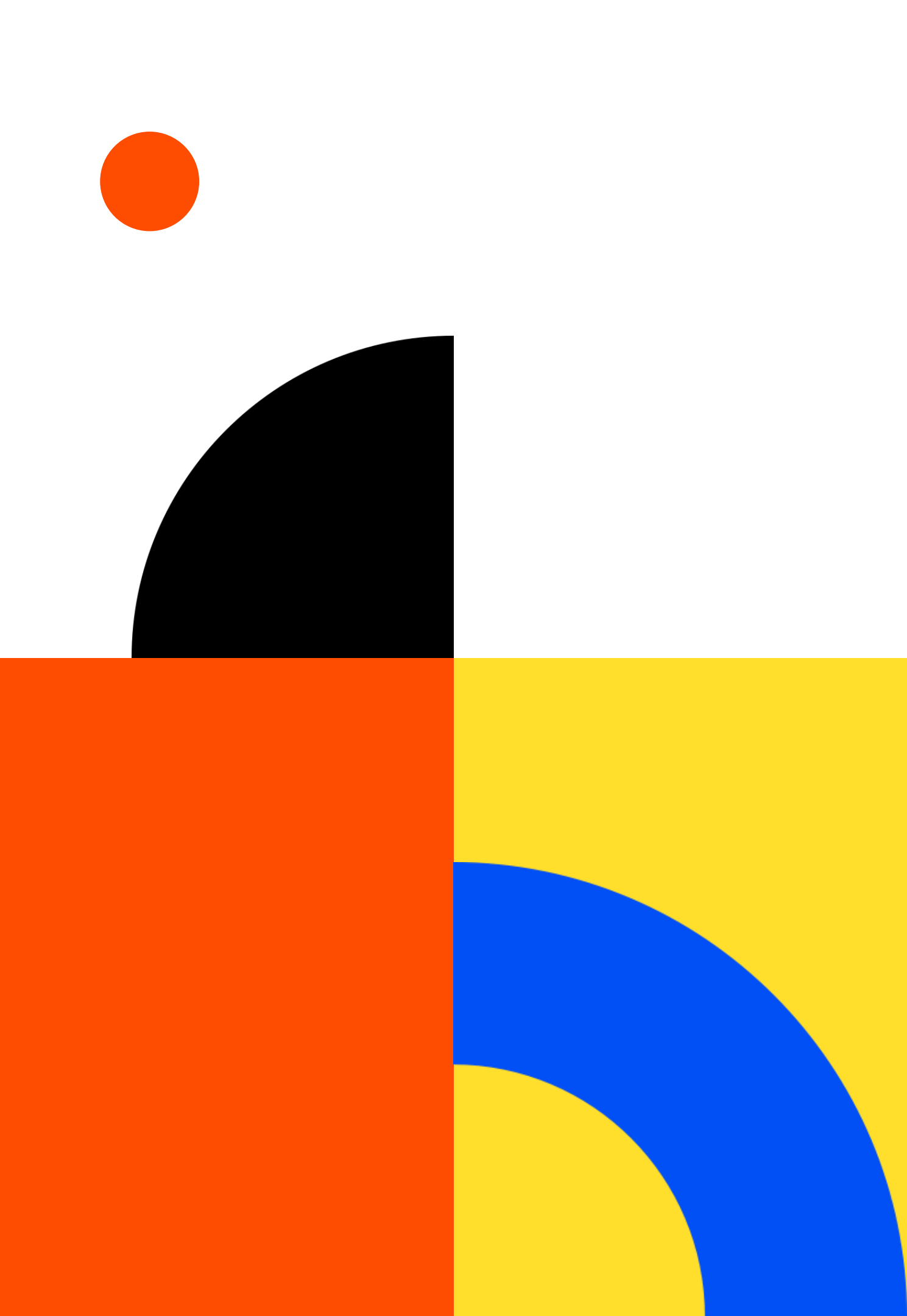


Лекція №11

Custom Editors

2025





План

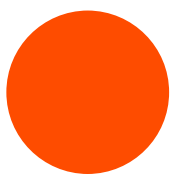
1. Custom Editors
2. EditorWindow
3. Gizmos



Custom Editors

Unity дозволяє розширювати редактор і створювати власні інструменти через **Editor API**.

Це дає можливість:

- створювати зручні інспектори під свої компоненти;
 - будувати складні інструменти для дизайнерів;
 - робити однокнопочні тулзи (генератори карт, розрахунки, кнопки “Build prefab”, тощо);
 - пришивидшувати workflow команди.
- 

Editor — головний клас

Щоб змінити вигляд інспектора, створюється клас, що успадковується від:

```
using UnityEditor;  
using UnityEngine;
```

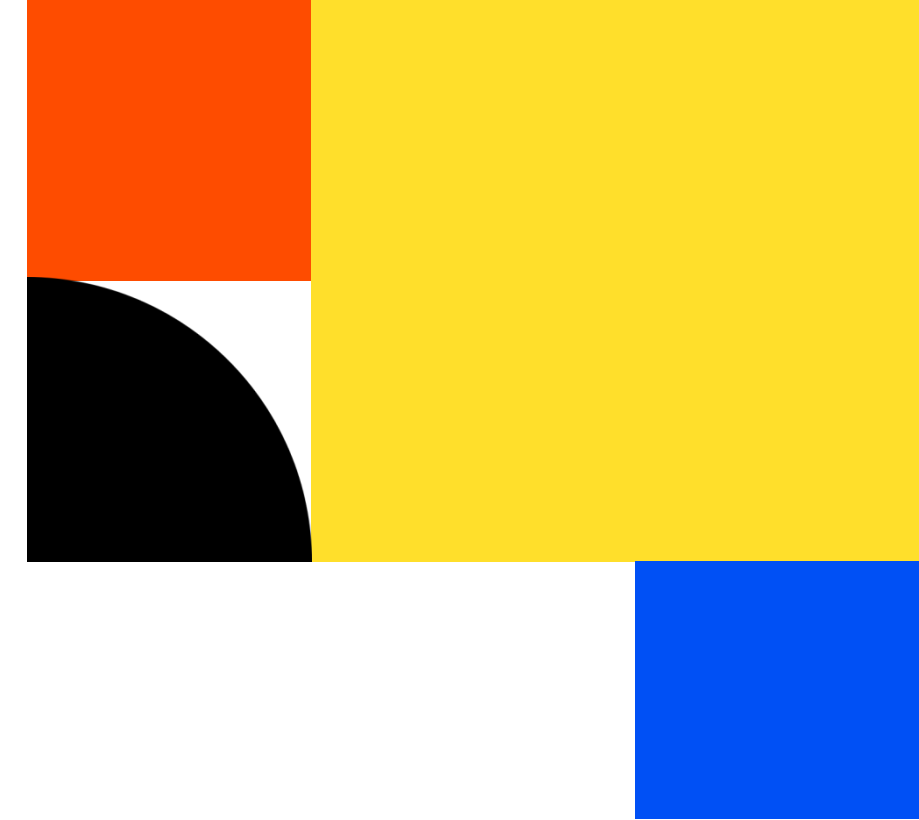
```
[CustomEditor(typeof(MyComponent))]  
public class MyComponentEditor : Editor  
{  
    public override void OnInspectorGUI()  
    {  
        DrawDefaultInspector();  
    }  
}
```

Де зберігати кастомні редактори

Усі редакторські скрипти повинні лежати у папці

- **Editor/**
або
- будь-яка папка з назвою **.../Editor/**.

Unity автоматично виключає їх із білду.



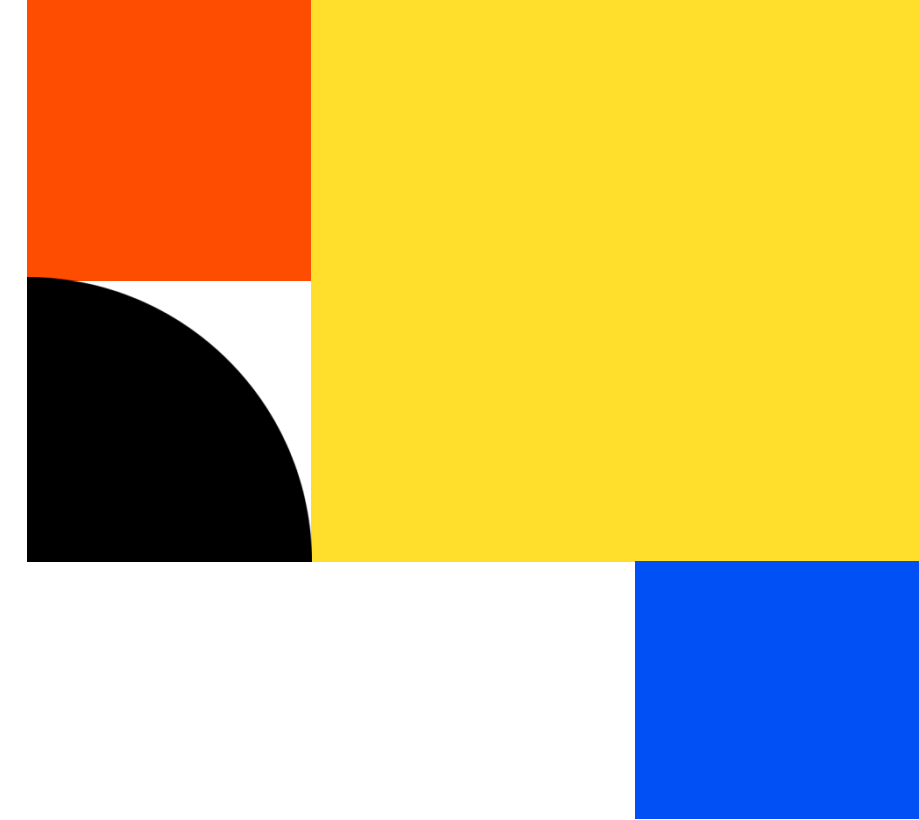
```
[CustomEditor(typeof(StatComponent))]  
public class StatComponentEditor : Editor  
{  
    public override void OnInspectorGUI()  
    {  
        StatComponent comp = (StatComponent)target;  
  
        // Draw fields manually  
        comp.health = EditorGUILayout.IntSlider("Health", comp.health, 0, 200);  
        comp.damage = EditorGUILayout.IntField("Damage", comp.damage);  
  
        EditorGUILayout.Space();  
  
        if (GUILayout.Button("Reset Stats"))  
        {  
            comp.ResetStats();  
        }  
  
        // Save changes  
        if (GUI.changed)  
            EditorUtility.SetDirty(comp);  
    }  
}
```

Результат:

Слайдер для Health

Поле для Damage

Кнопка "Reset Stats"

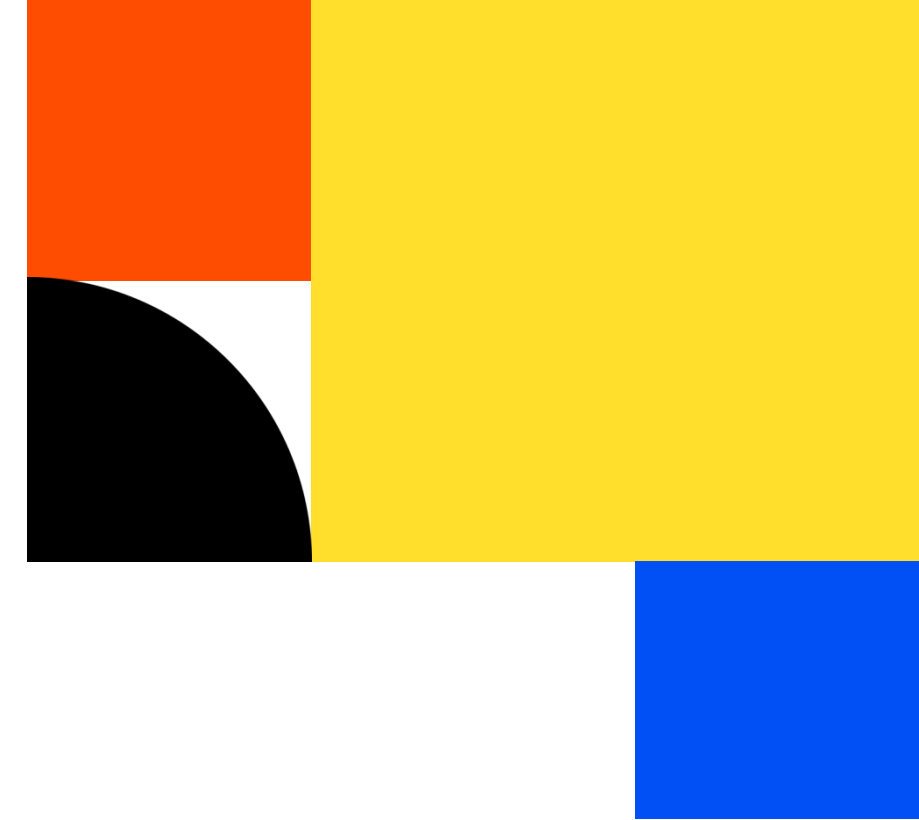


EditorWindow

```
public class MyToolWindow : EditorWindow
{
    [MenuItem("Tools/My Window")]
    static void Open()
    {
        GetWindow<MyToolWindow>("My Tool");
    }

    void OnGUI()
    {
        GUILayout.Label("My Custom Tool", EditorStyles.boldLabel);

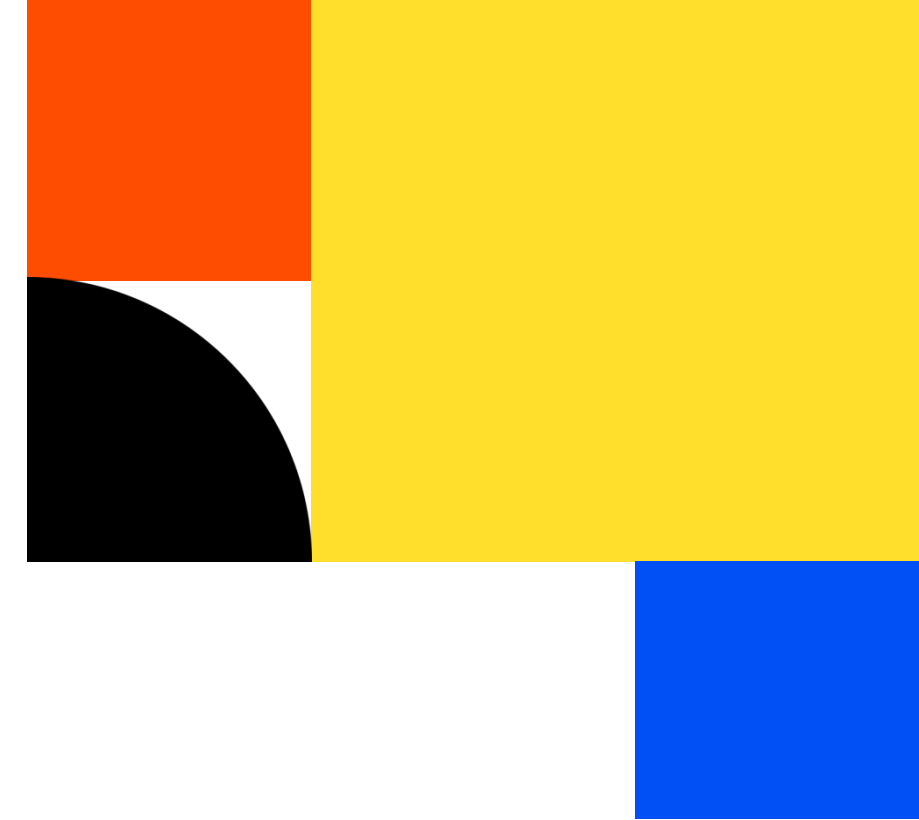
        if (GUILayout.Button("Click me"))
            Debug.Log("Pressed!");
    }
}
```



Gizmos

Gizmos — графічні підказки, які відображаються **лише в редакторі**, щоб допомагати орієнтуватися в сцені.

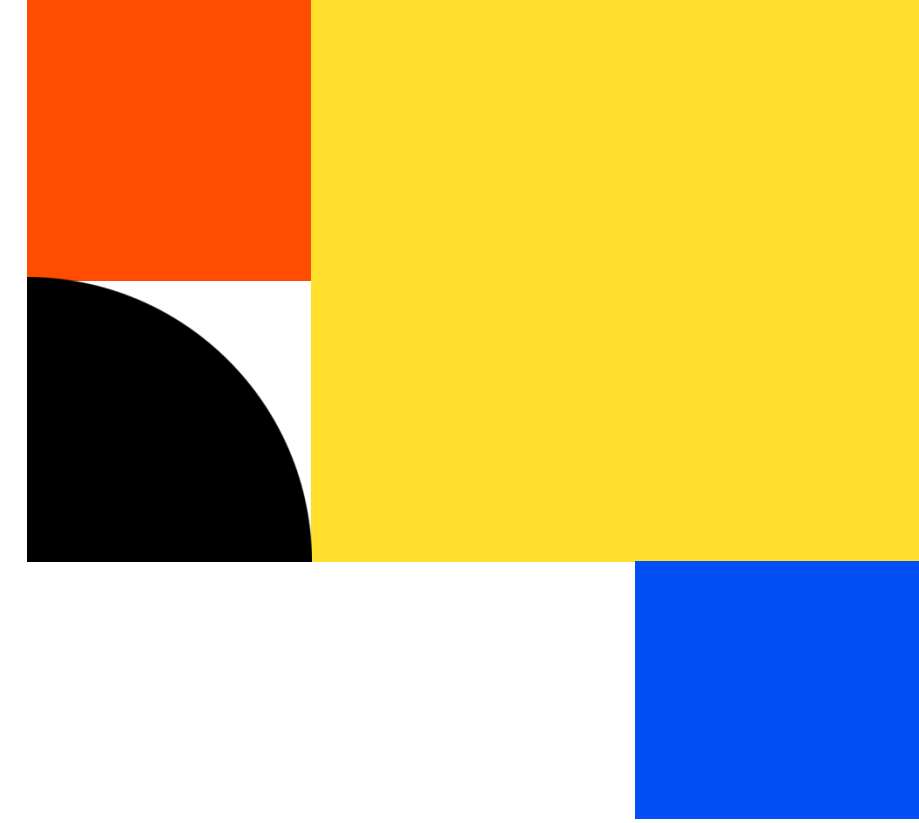
- Вони не впливають на гру, не рендеряться в білд і використовуються для:
- візуалізації радіусів;
- показу точок спавну;
- відображення трас, зон агро, областей патрулювання;
- дебагу навігації та тригерів.



В компонент додається метод:

```
void OnDrawGizmos()  
{  
    Gizmos.color = Color.green;  
    Gizmos.DrawSphere(transform.position, 0.3f);  
}
```

Це малює зелений сферичний маркер у сцені.



OnDrawGizmos()

- Показує гізмо **завжди**, навіть коли об'єкт не вибрано.

OnDrawGizmosSelected()

- Показує гізмо тільки коли об'єкт виділений у інспекторі.

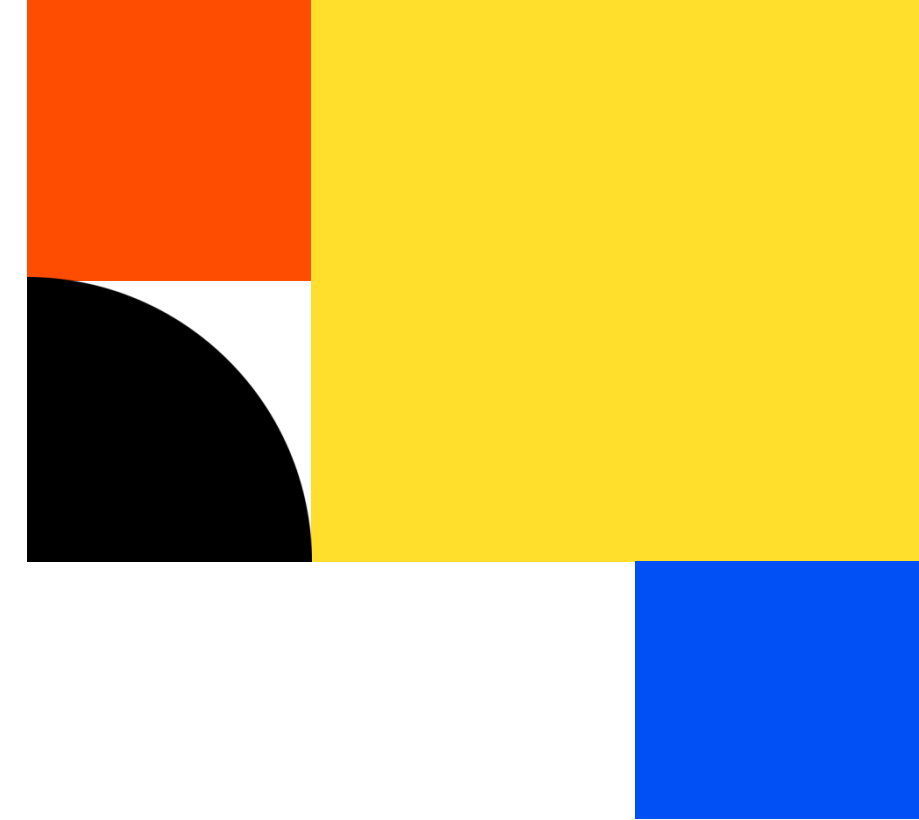
```
void OnDrawGizmosSelected()
```

```
{
```

```
    Gizmos.color = Color.yellow;
```

```
    Gizmos.DrawWireCube(transform.position, Vector3.one);
```

```
}
```



Малювання різних фігур

`Gizmos.DrawSphere(position, radius);`

`Gizmos.DrawWireSphere(position, radius);`

`Gizmos.DrawCube(position, size);`

`Gizmos.DrawLine(start, end);`

*`Gizmos.DrawLine(transform.position, transform.position +
transform.forward * 2);`*

