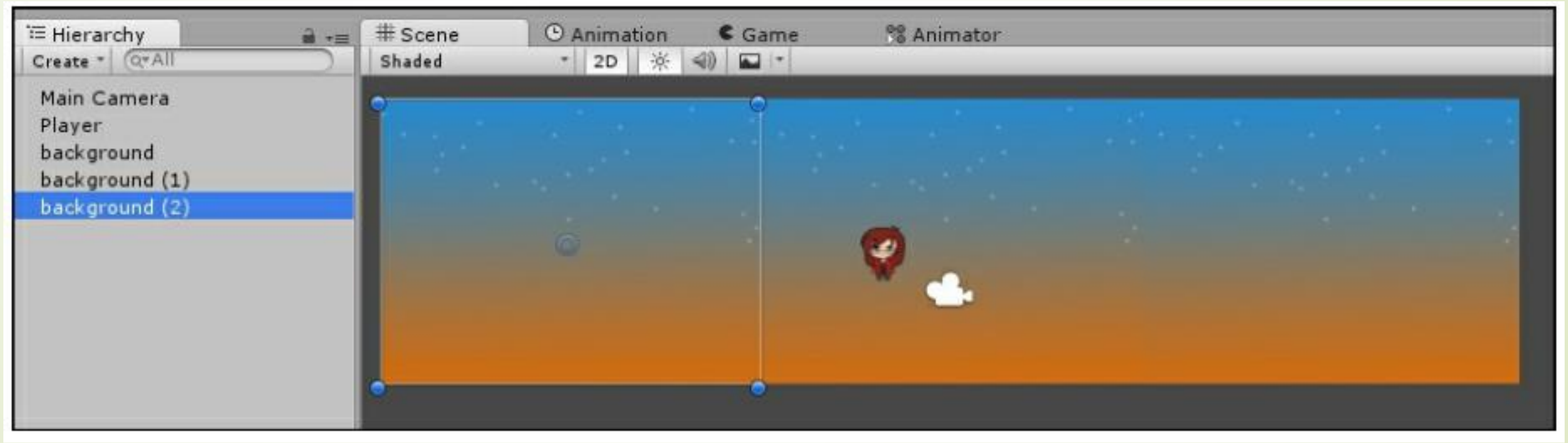
# Thiết kế background

# Thiết kế background

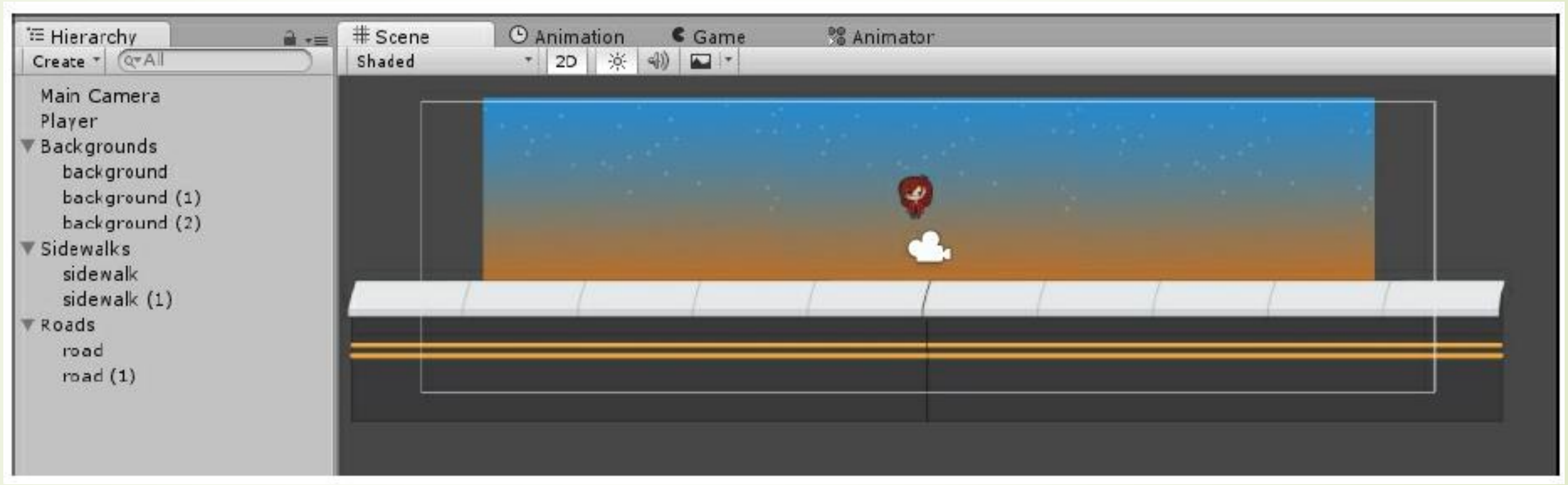
- Thêm sprite background vào scene
- Sao chép nhiều background
- Tạo một GameObject, rename thành background, set position (0,0)
- Kéo các background vào gameobject vừa tạo
- Ghép background bằng phím V K



Main Camera
Player
background
background (1)
background (2)

# Thiết kế background

- Tương tự cho Roads, Sidewalks



## Follow Camera

- Tạo file script followcamera gắn vào gameobject camera

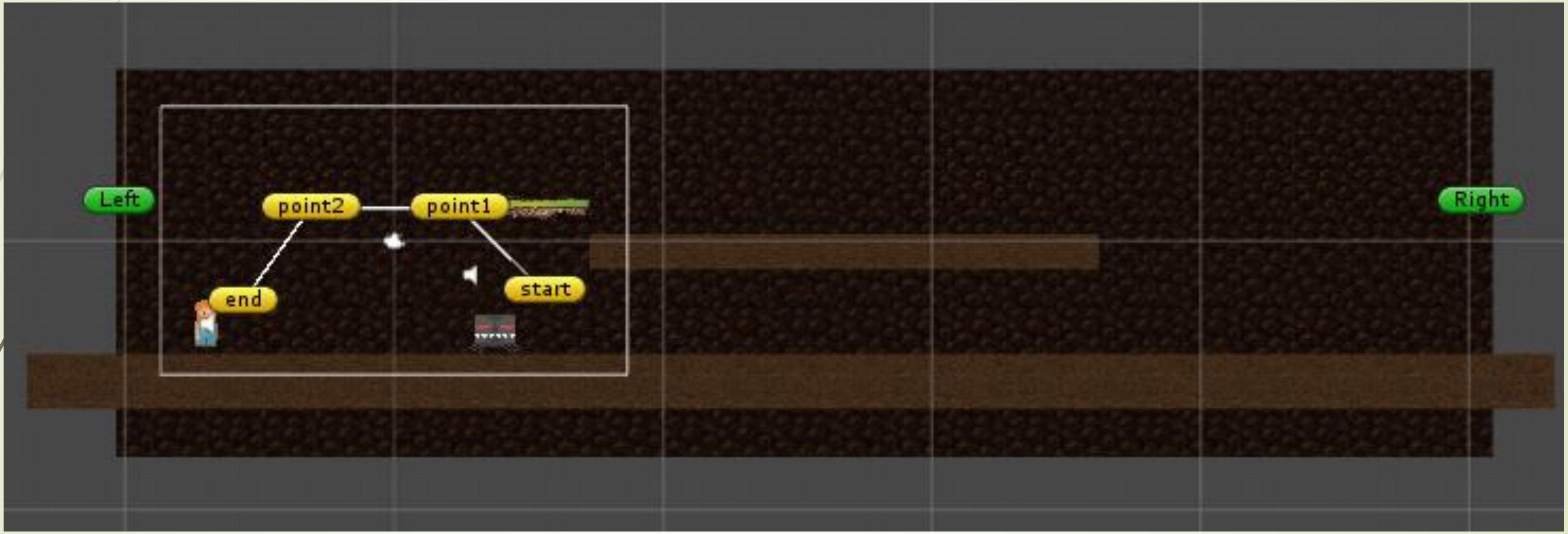
```
public Transform player;
void Start () {
        player = GameObject.Find(“Player").transform;
}
void LateUpdate () {
        if (player != null)
        {
            Vector3 temp = transform.position;
            temp.x = player.position.x;
            transform.position = temp;
        }
}
```

## Comparing Update, FixedUpdate, and LateUpdate

Cập nhật cho mỗi frame hình

- Update()
- FixedUpdate() : cho gameObject có tính vật lý
- LateUpdate() : cho camera

# Demo SpiderCave



## Giới hạn đường biên Camera

```
public class CameraControl : MonoBehaviour {

    Transform player;
    // Use this for initialization
    void Start () {
        player = GameObject.Find("Player").transform;
    }

    // Update is called once per frame
    void Update () {
        if (player != null)
        {
            Vector3 temp = transform.position;
            temp.x = player.position.x;
            if (temp.x < -1.4f) temp.x = -1.4f;
            if (temp.x > 31f) temp.x = 31f;
            transform.position = temp;
        }
    }
}
```

## Background chạy ngang

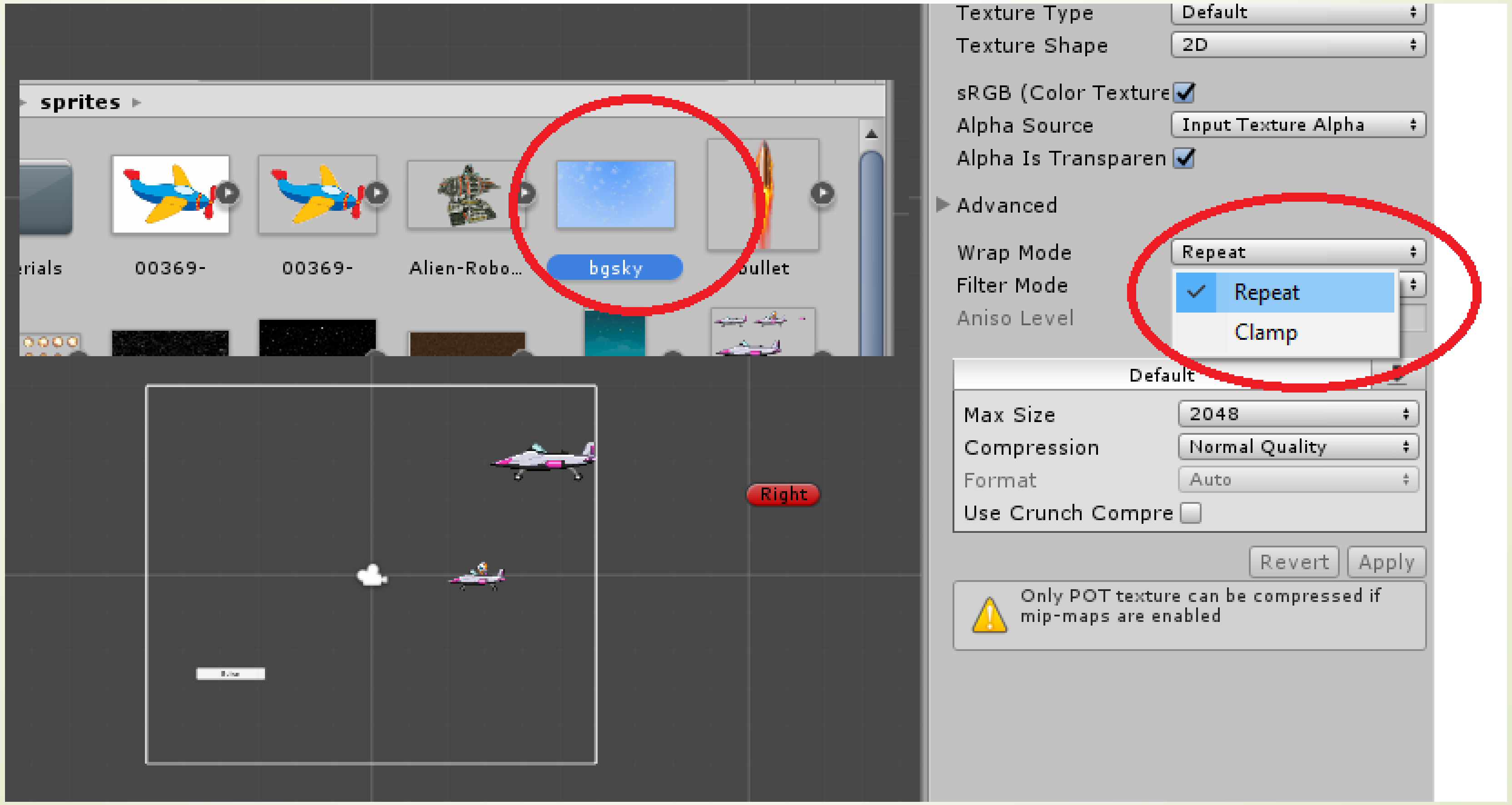
```
 public float speed = 0.5f;
   private Vector3 startPos;
// Use this for initialization
void Start () {
        startPos = transform.position;
}

// Update is called once per frame
void Update () {

        transform.Translate(new Vector3(-1, 0) * speed * Time.deltaTime);
        if (transform.position.x < -6) transform.position = startPos;
}
```

## Background chạy dọc

- transform.Translate(new Vector3(0, -1) * speed * Time.deltaTime);

# Dùng Quad

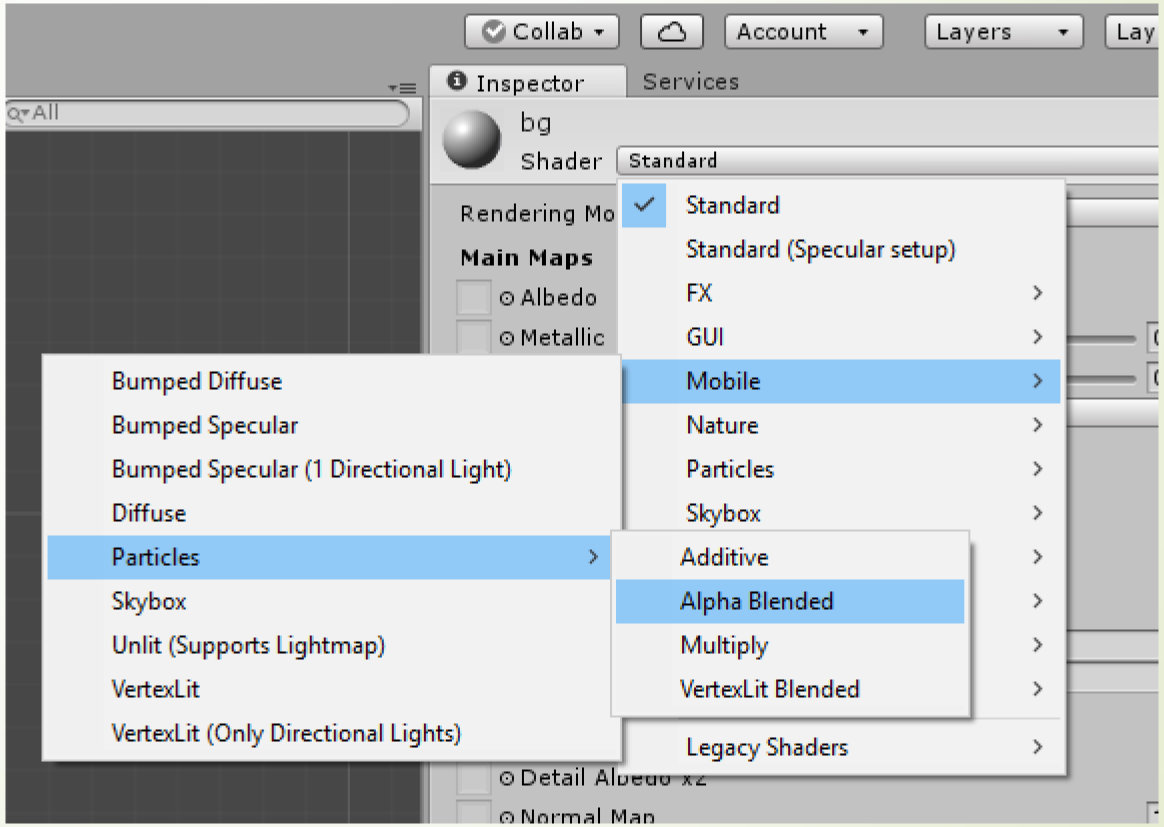
- Chỉnh sprite là Repeat



Texture Type: Default
Texture Shape: 2D
sRGB (Color Texture ✓
Alpha Source: Input Texture Alpha
Alpha Is Transparen ✓
Advanced
Wrap Mode: Repeat
Filter Mode
Aniso Level
Max Size: 2048
Compression: Normal Quality
Format: Auto
Use Crunch Compre ☐
Only POT texture can be compressed if mip-maps are enabled

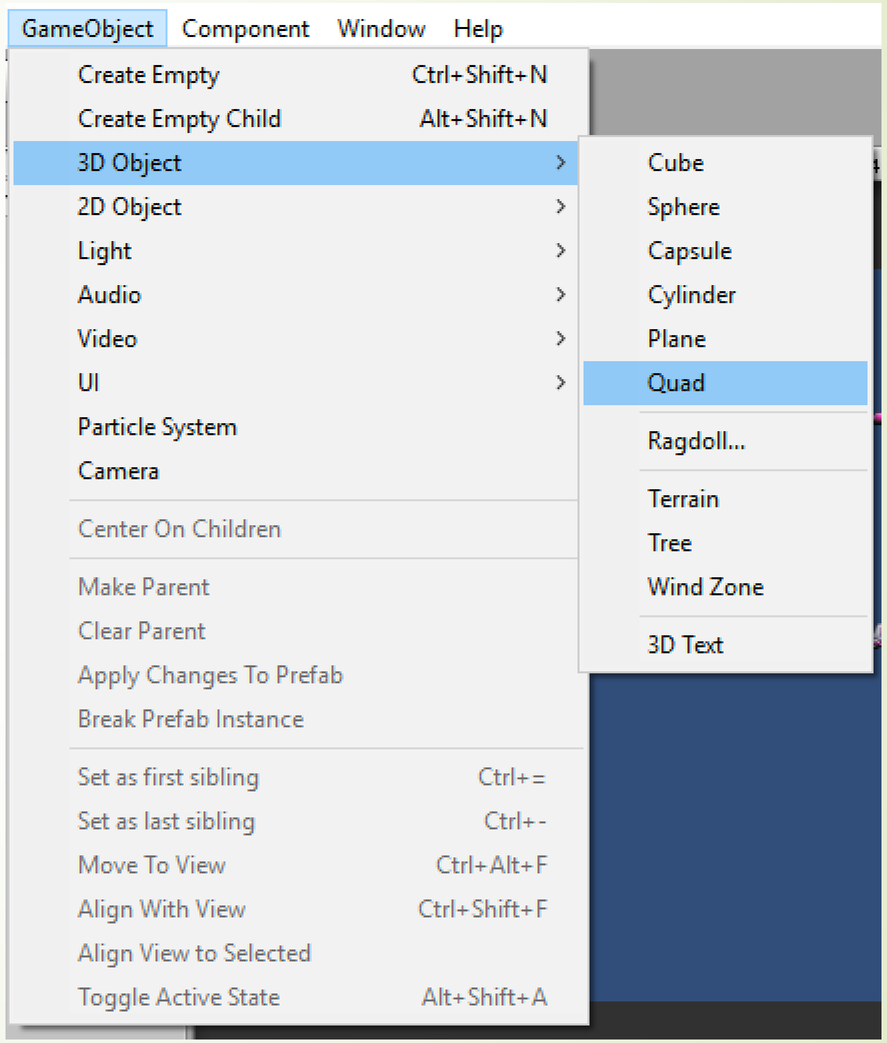
# Dùng Quad

- Tạo material



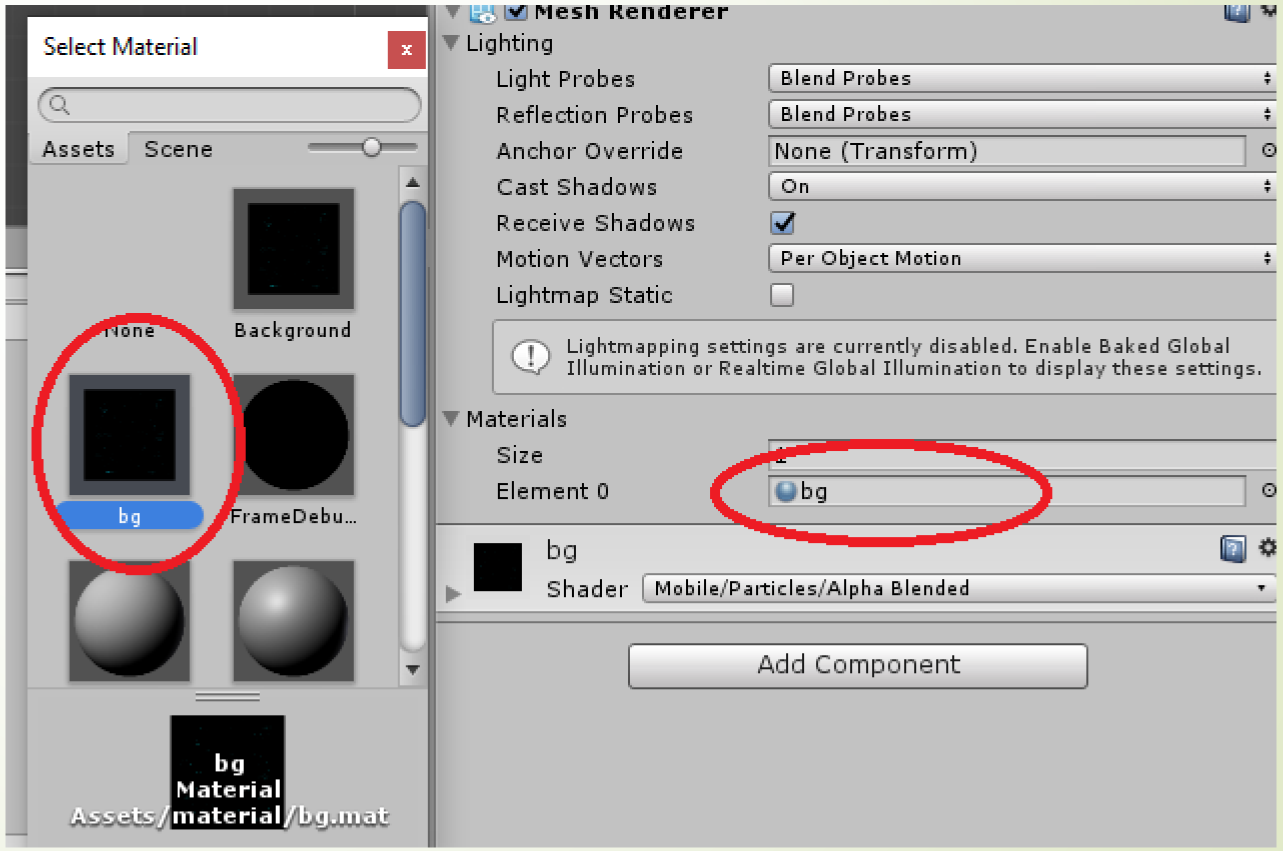
# Dùng Quad

- Thêm gameobject Quad



# Dùng Quad

- Chọn Material cho Quad



Select Material

Assets Scene

None | Background
bg | FrameDebu...

bg
Material
Assets/material/bg.mat

Mesh Renderer

Lighting

| | |
|---|---|
| Light Probes | Blend Probes |
| Reflection Probes | Blend Probes |
| Anchor Override | None (Transform) |
| Cast Shadows | On |
| Receive Shadows | ✓ |
| Motion Vectors | Per Object Motion |
| Lightmap Static | |

Lightmapping settings are currently disabled. Enable Baked Global Illumination or Realtime Global Illumination to display these settings.

Materials

| | |
|---|---|
| Size | 1 |
| Element 0 | bg |

bg
Shader Mobile/Particles/Alpha Blended

Add Component

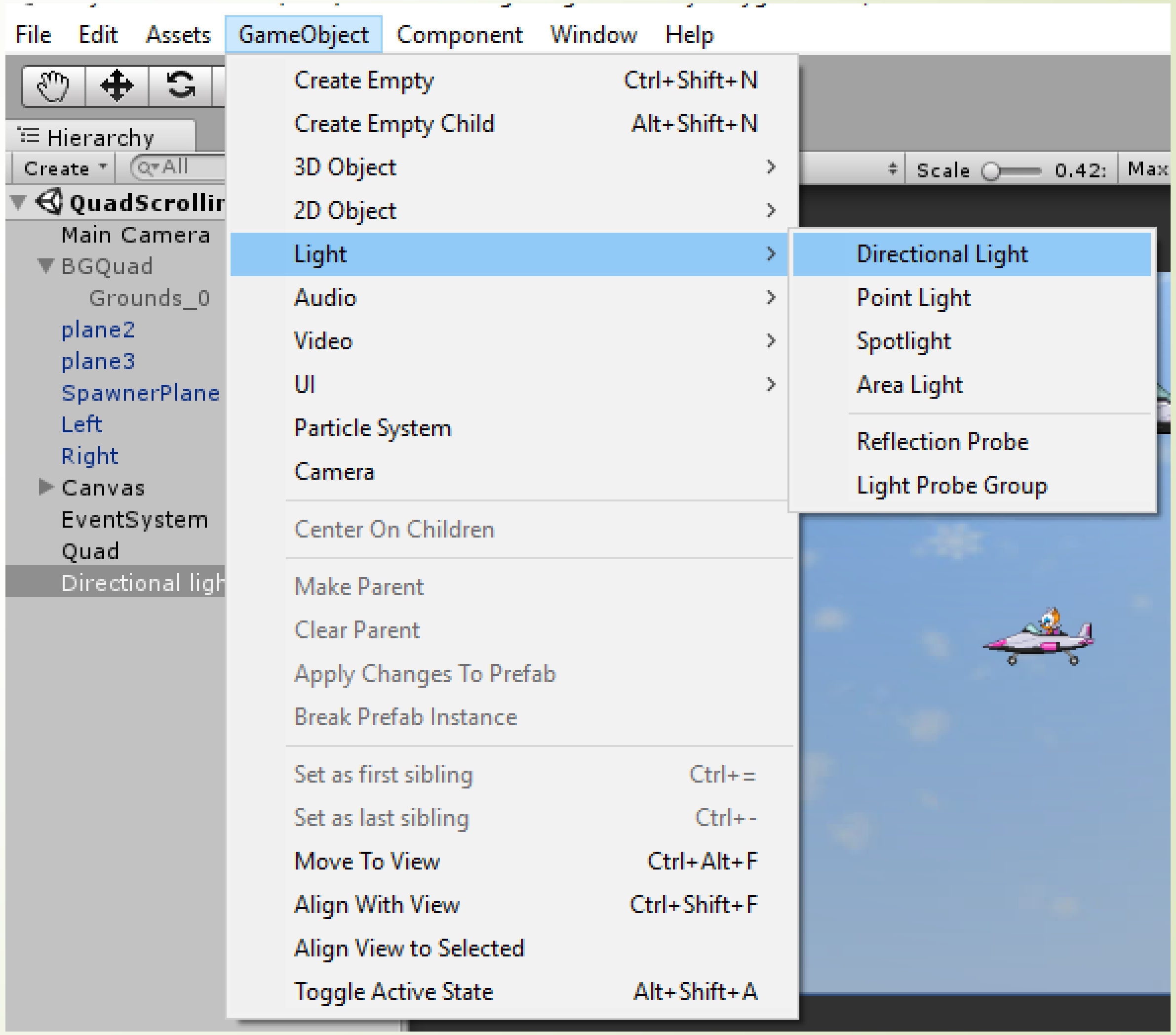
## Dùng Quad

- Viết BGScaler

```
public class BGScaler : MonoBehaviour {



void Start () {
        float worldHeight = Camera.main.orthographicSize * 2f;

        float worldWitdh = worldHeight * Screen.width / Screen.height;

        transform.localScale = new Vector3(worldWitdh, worldHeight, 0f);
}
}
```

## Dùng Quad

- Thêm Directional Light để điều chỉnh độ sang cho background

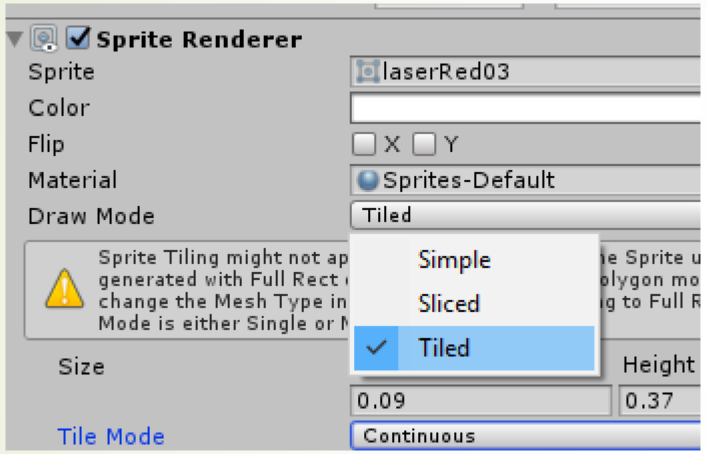


## Dùng Quad

- Viết script bgQuadScrolling

```
public class bgQuadScrolling : MonoBehaviour {


    public float speed =1f;
    public Material mat;
    private Vector2 offset = Vector2.zero;
    private void Awake()
    {
        mat = GetComponent<Renderer>().material;
    }
    void Start () {
        offset = mat.GetTextureOffset("_MainTex");
    }

    // Update is called once per frame
    void Update () {
        offset.x -= speed * Time.deltaTime;
        mat.SetTextureOffset("_MainTex", offset);
    }
}
```

# Tile

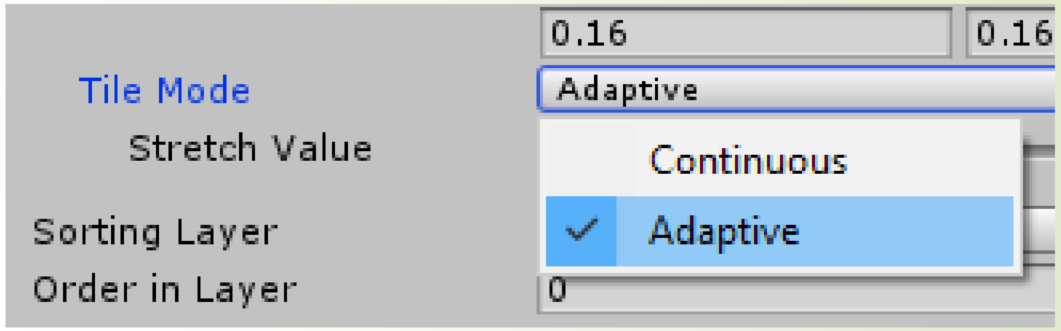

Sprite Renderer

| | |
|---|---|
| Sprite | laserRed03 |
| Color | |
| Flip | X Y |
| Material | Sprites-Default |
| Draw Mode | Tiled |

Sprite Tiling might not ap
generated with Full Rect
change the Mesh Type in
Mode is either Single or M

e Sprite u
olygon mo
g to Full R

Size Height
0.09 0.37
Tile Mode Continuous

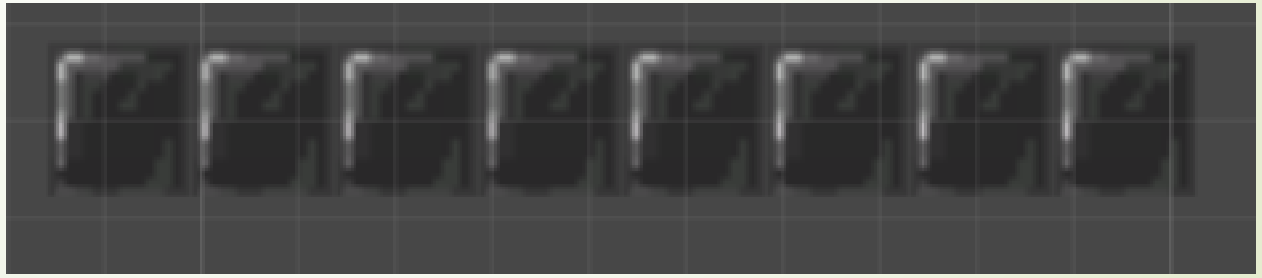

0.16 0.16

Tile Mode Adaptive

Stretch Value

Sorting Layer

Order in Layer 0



## Parralax

```
public class Parralax : MonoBehaviour
{
    public Transform[] backgrounds;
    private float[] parallaxScales;
    public float smoothing = 1f;

    private Transform cam;
    private Vector3 previousCamPos;

    void Awake()
    {
        cam = Camera.main.transform;
    }
```

```
 void Start()
    {
        previousCamPos = cam.position;
        parallaxScales = new
float[backgrounds.Length];

        for (int i = 0; i < backgrounds.Length; i++)
        {
            parallaxScales[i] = backgrounds[i].position.z *
1;
        }
    }
```

## Parralax

```
  // Update is called once per frame
    void LateUpdate()
    {
        for (int i = 0; i < backgrounds.Length; i++)
        {
            float parallax = (previousCamPos.x - cam.position.x) * parallaxScales[i];
            float backgroundTargetPosX = backgrounds[i].position.x + parallax;
            Vector3 backgroundTargetPos = new Vector3(backgroundTargetPosX,
backgrounds[i].position.y, backgrounds[i].position.z);
            backgrounds[i].position = Vector3.Lerp(backgrounds[i].position,
backgroundTargetPos, smoothing * Time.deltaTime);
        }
        previousCamPos = cam.position;
    }
}
```