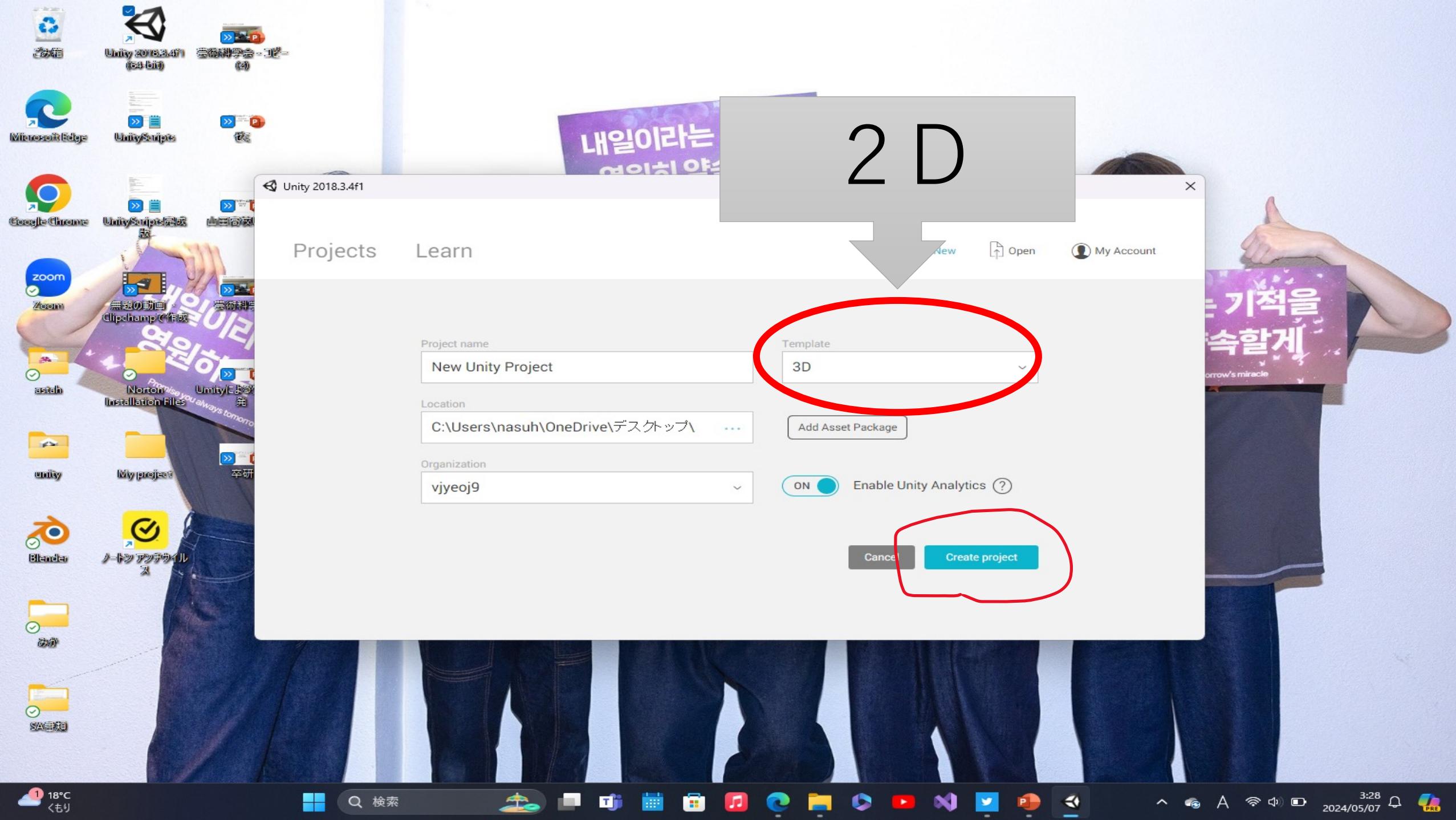


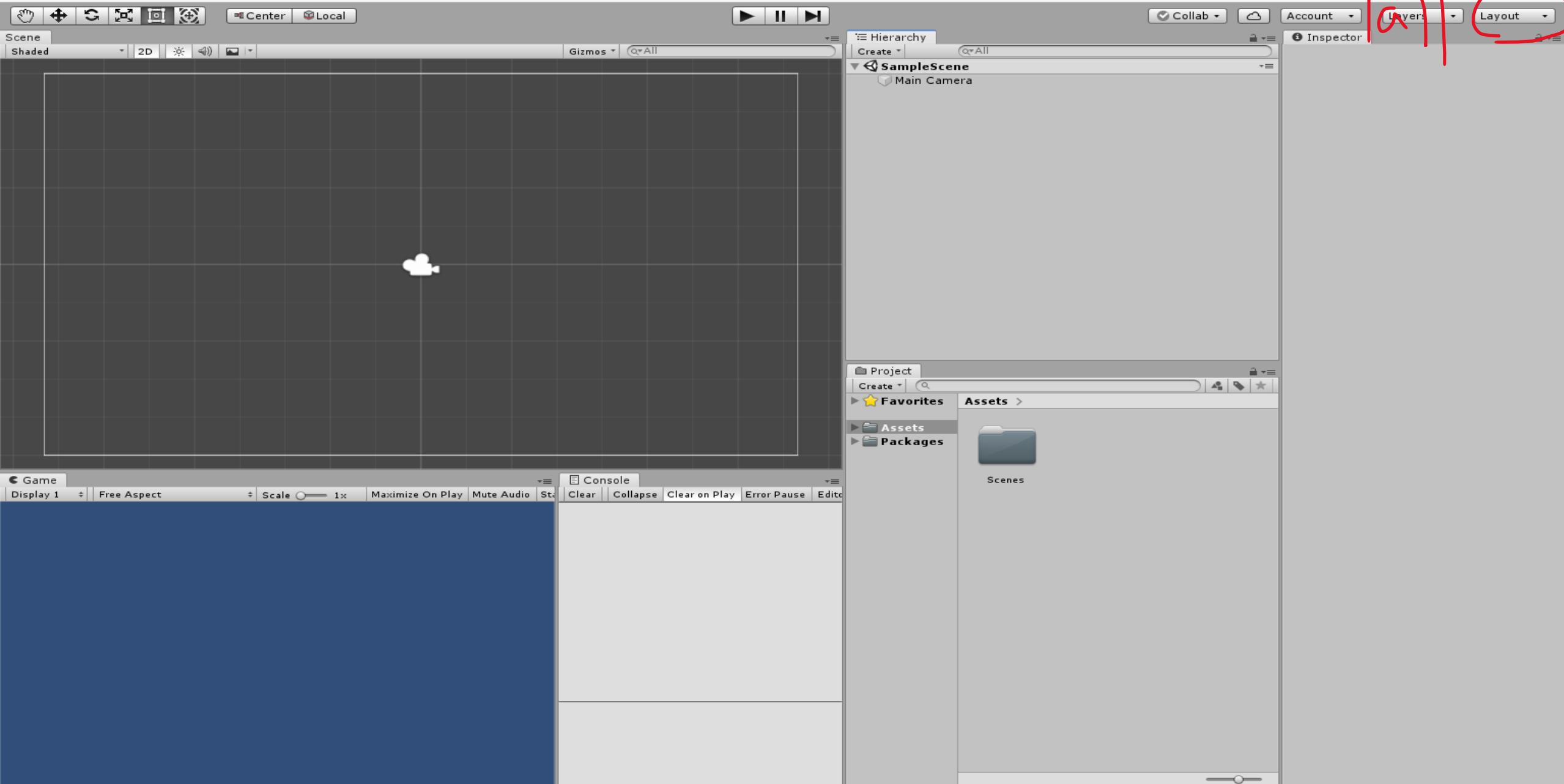
2Dアクションゲーム

角田研究室

奥崎 弥



File Edit Assets GameObject Component Window Help





Center Local



Collab



Account



Layers



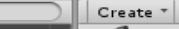
Scene

Shaded 2D



Gizmos All

Hierarchy



All



SampleScene



Inspector



シーンビュー

ヒエラルキー

インスペクター

プロジェクト

ゲームビュー

Game

Display 1

Free Aspect

Scale 1x

Maximize On Play

Mute Audio

St

Console

Clear

Collapse

Clear on Play

Error Pause

Edito

Project

Create



Favorites



Assets



Packages

Assets >



Scenes



1

18°C

くもり



検索



3:29

2024/05/07



マップの作成

File Edit Assets GameObject Component Window Help



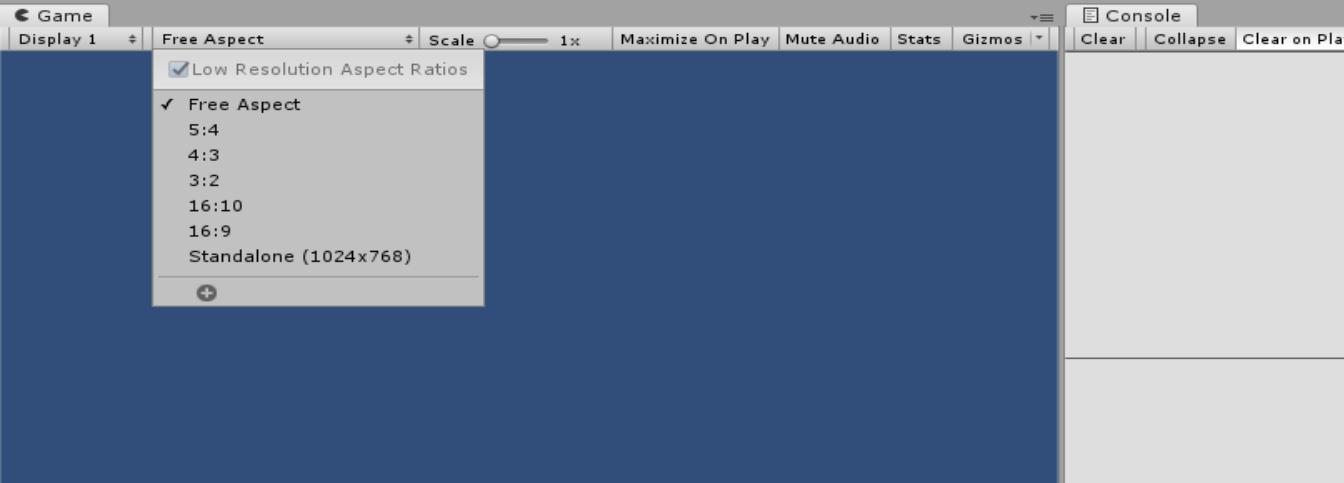
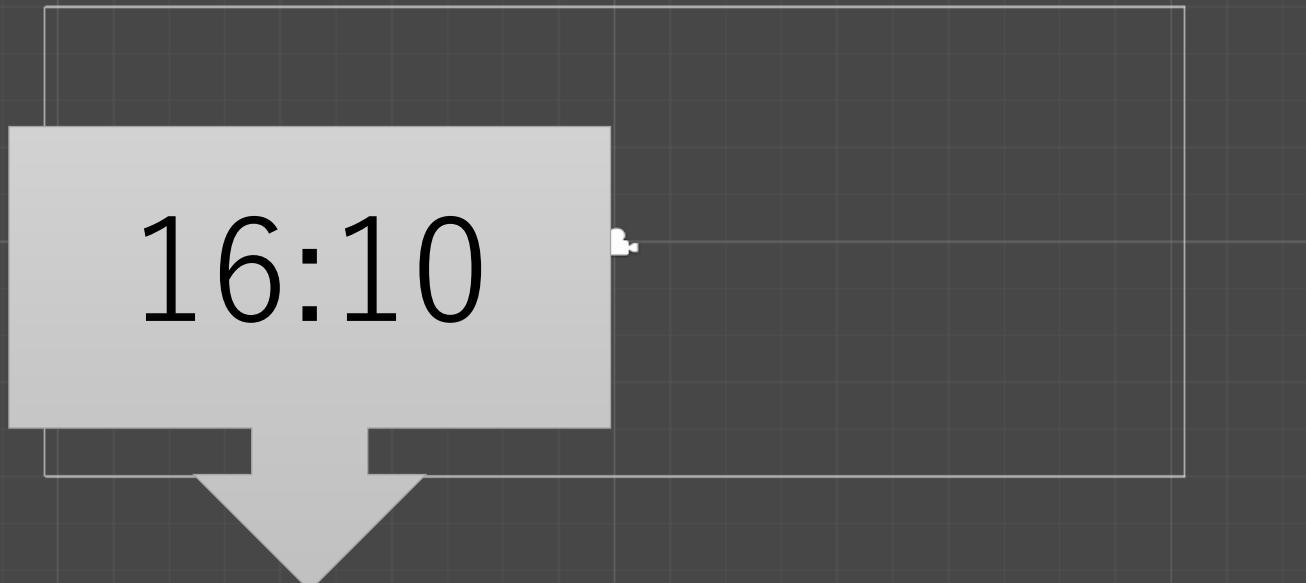
Center Local



Collab Account Layers Layout

Scene
Shaded 2D

Gizmos All



Hierarchy

SampleScene

Main Camera

Project

Create

Favorites

Assets

Packages

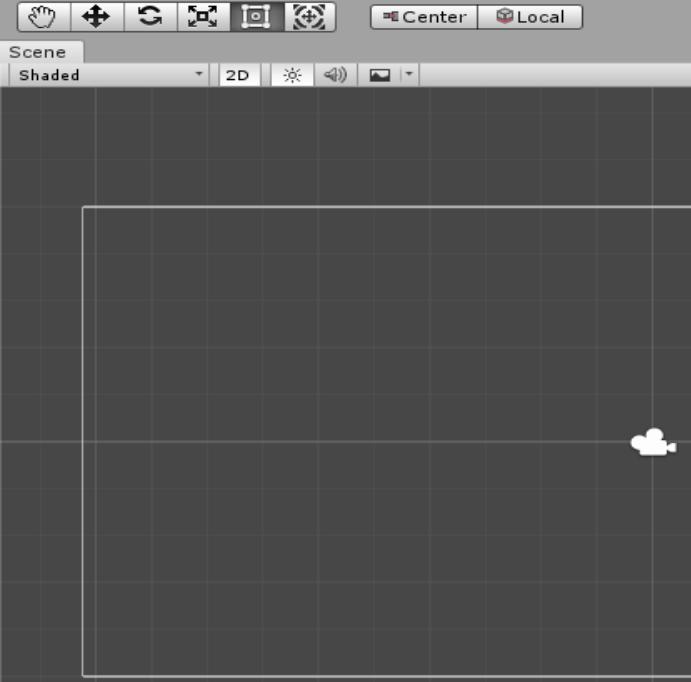
Scenes



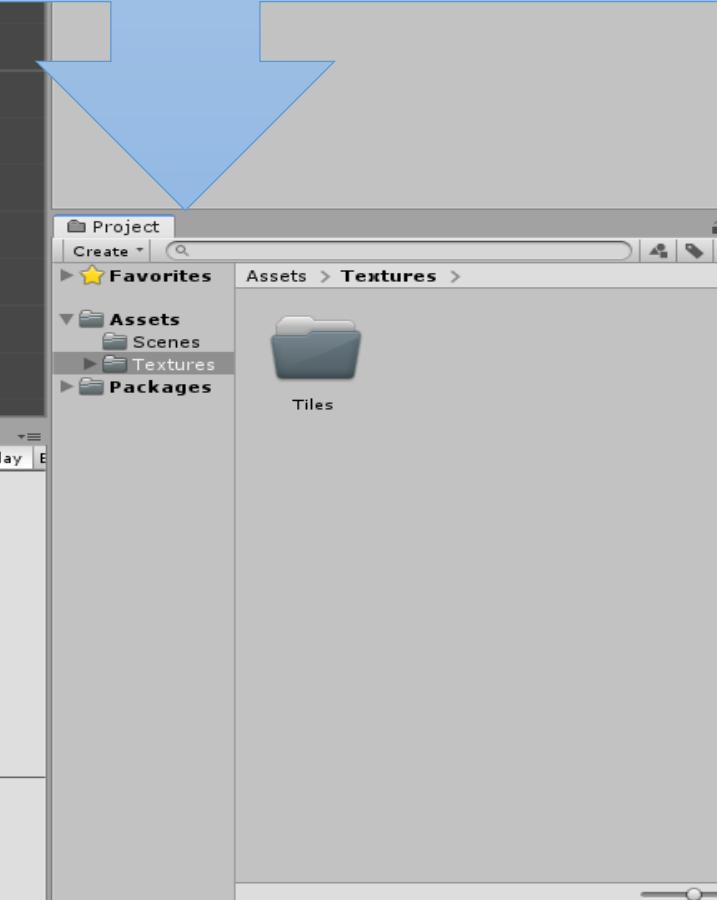
検索



3:52 2024/05/07 PRE



【Project】の中にtexturesフォルダ作成
Assets→Create→folder
名前：Textures



Scene
Shaded
2D

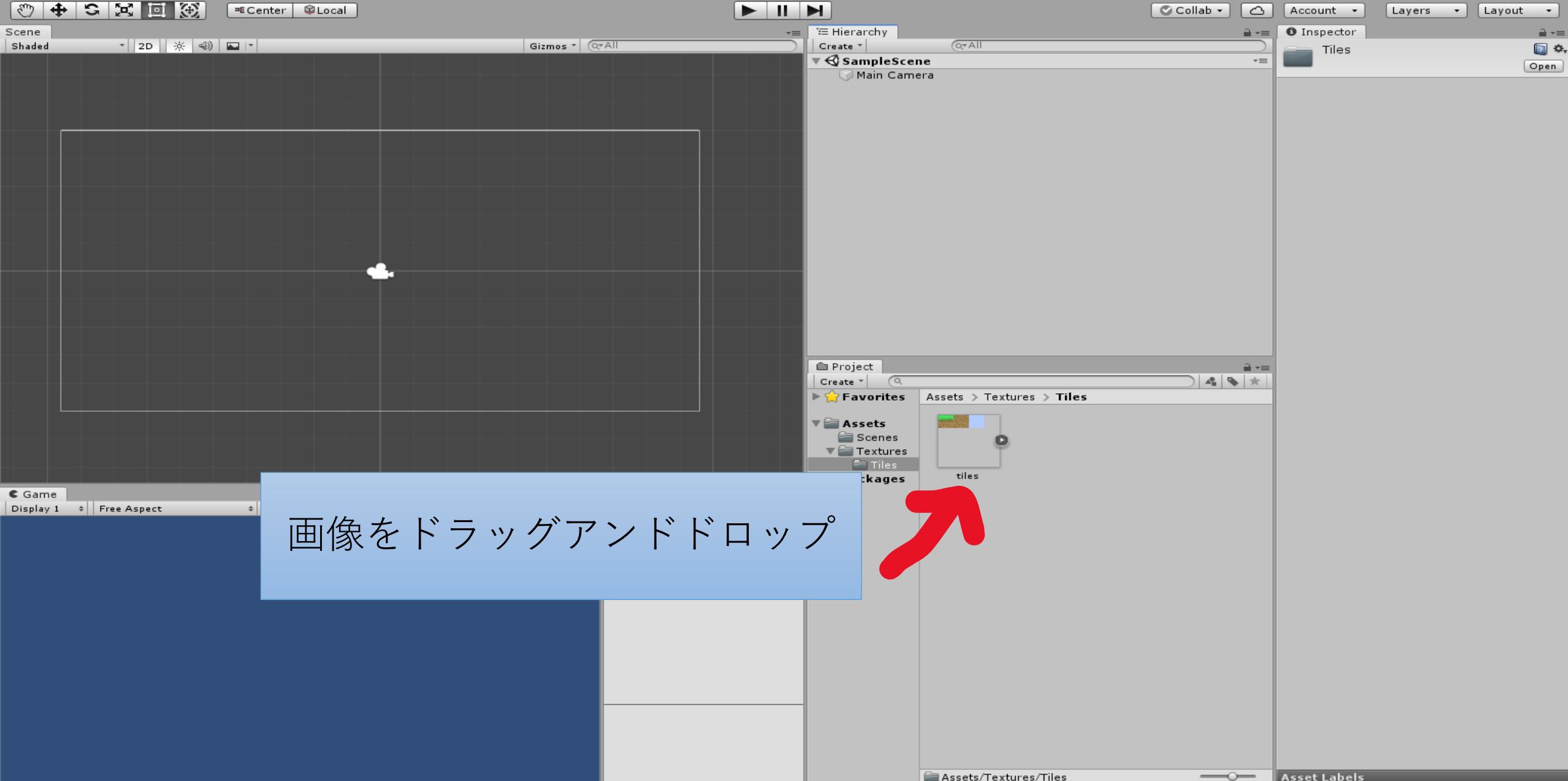
Gizmos All

Texturesの中にTiles フォルダ作成

Textures → Create
→ folder
名前 : Tiles

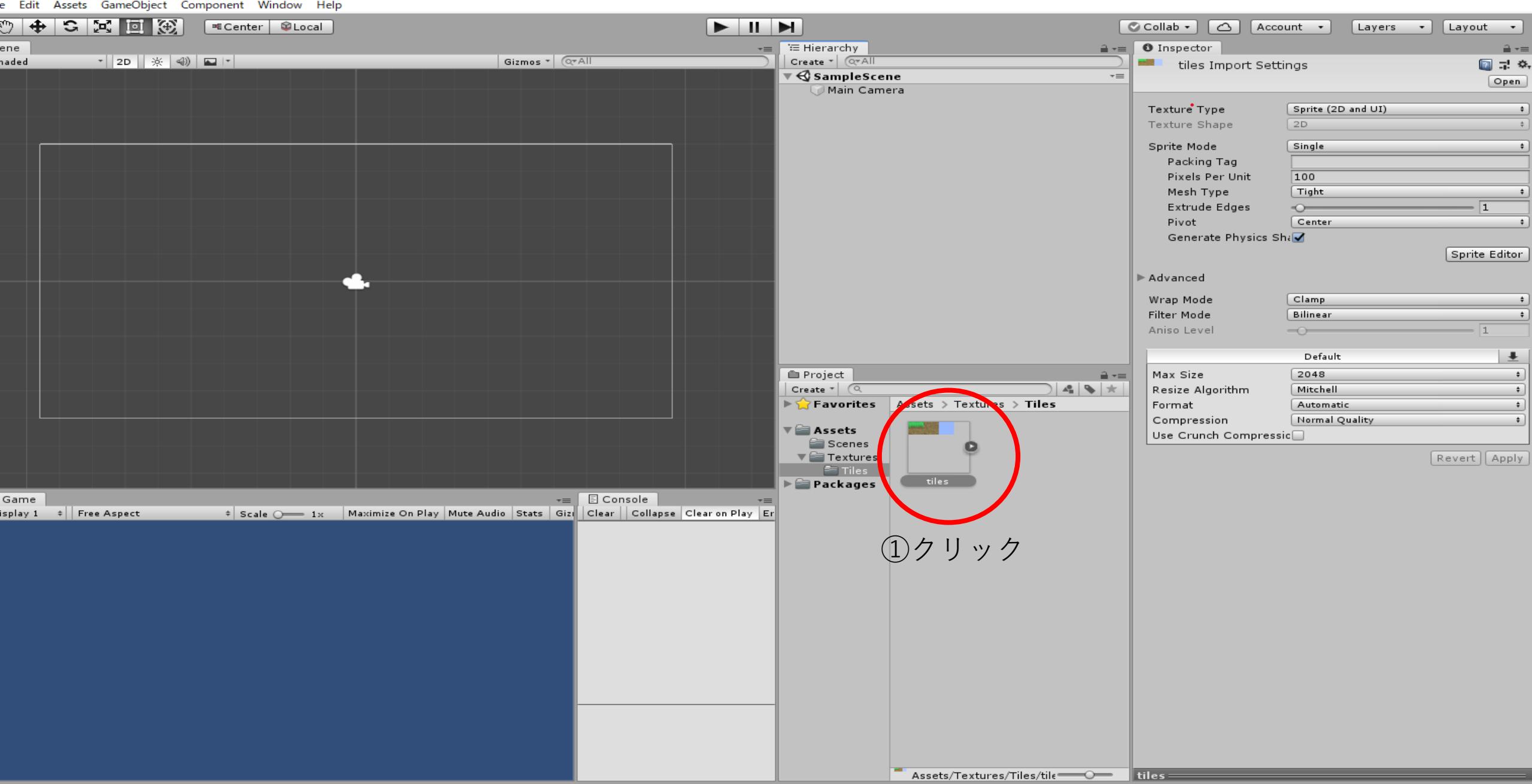
The screenshot shows the Unity Editor interface with the 'Create' menu open. The 'Folder' option is highlighted with a red box. Other options visible in the menu include C# Script, Shader, Testing, Playables, Assembly Definition, TextMeshPro, Scene, Prefab Variant, Audio Mixer, Material, Lens Flare, Render Texture, Lightmap Parameters, Custom Render Texture, Sprite Atlas, Sprites, Tile, Animator Controller, Animation, Animator Override Controller, Avatar Mask, Timeline, Physic Material, Physics Material 2D, GUI Skin, Custom Font, Legacy, Brush, Terrain Layer, and UIElements Editor Window.

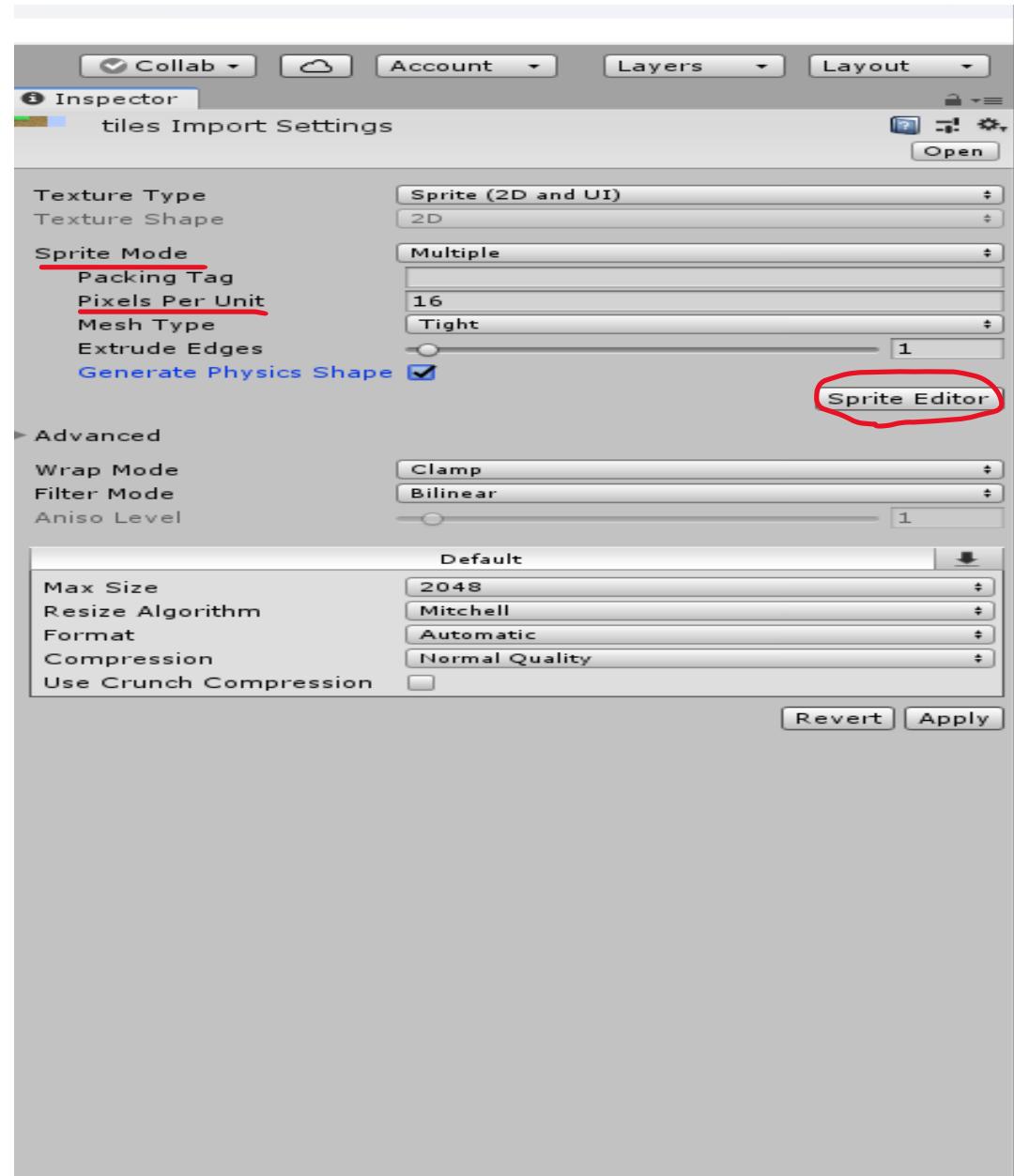
File Edit Assets GameObject Component Window Help



画像をドラッグアンドドロップ

テクスチャの設定





① Sprite Mode → Multiple(複数)

: 1つの画像から複数の
スプライトを作成

② Pixels Per Unit → 16

: スプライトのサイズ調整

③ Sprite Editor クリック

Scene
Shaded
2DGizmos
Create
All

Hierarchy Tile Palette

Create All

SampleScene

Main Camera



Account

Layers

Layout

tiles Import Settings



Texture Type

Sprite (2D and UI)

Texture Shape

2D

Sprite Mode

Multiple

Packing Tag

Pixels Per Unit

16

Mesh Type

Tight

Extrude Edges

Generate Physics Shape

Sprite Editor

Advanced

Wrap Mode

Clamp

Filter Mode

Bilinear

Aniso Level

1

Default

Max Size

2048

Resize Algorithm

Mitchell

Format

Automatic

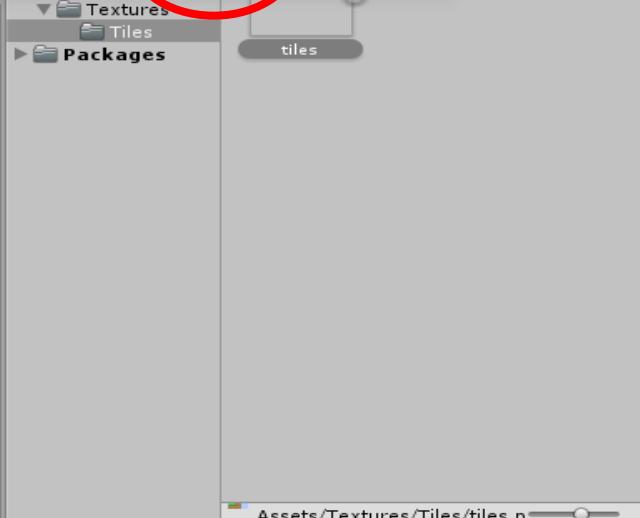
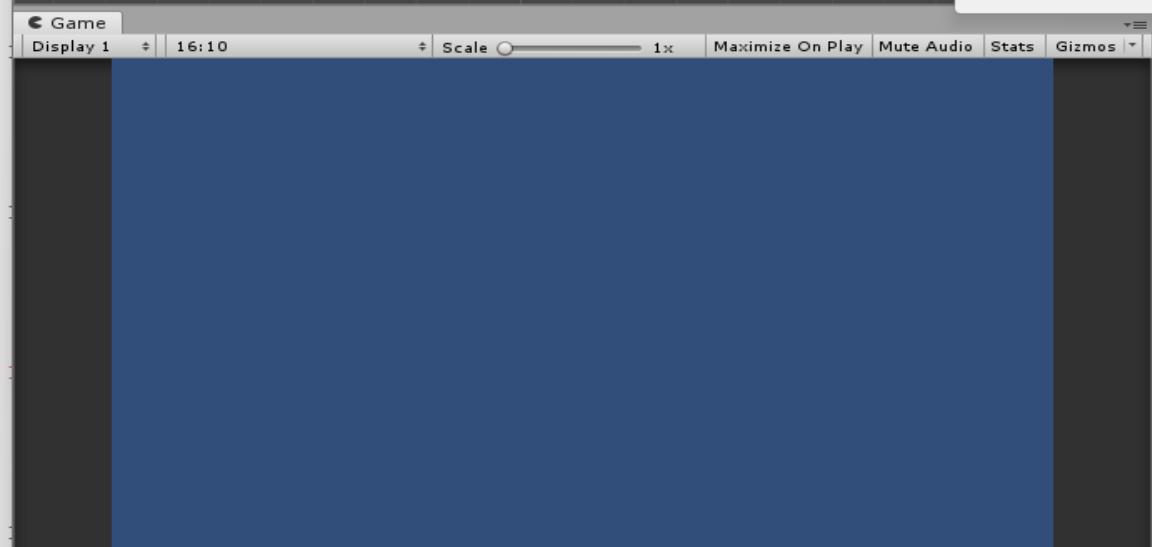
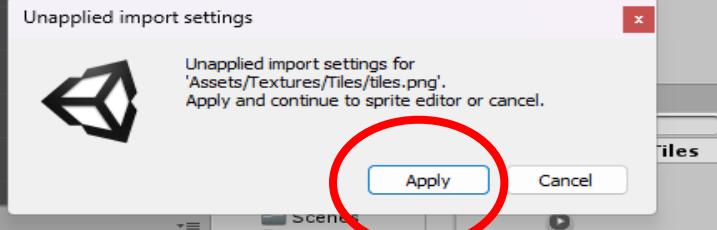
Compression

Normal Quality

Use Crunch Compression

Revert

Apply



File Edit Assets GameObject Component Window Help



Center Local



Collab Account Layers Layout

Scene

Shaded 2D

Gizmos

Hierarchy Tile Palette

Create All

SampleScene

Inspector

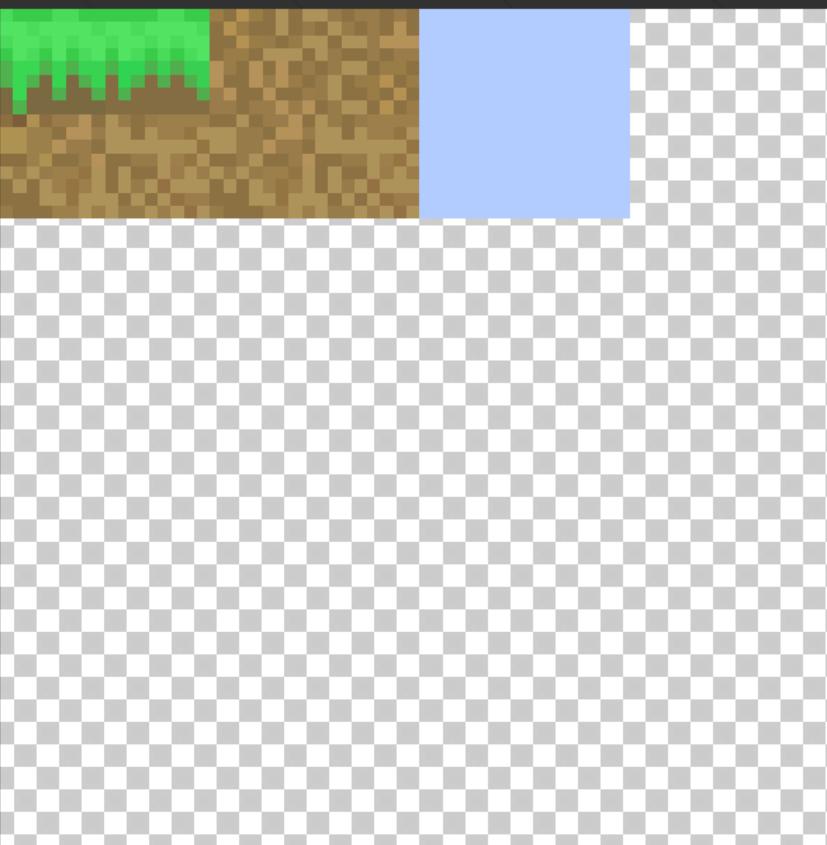
tiles Import Settings

Open

Sprite Editor

Sprite Editor Slice Trim

Revert Apply



Game

Display 1



16:10

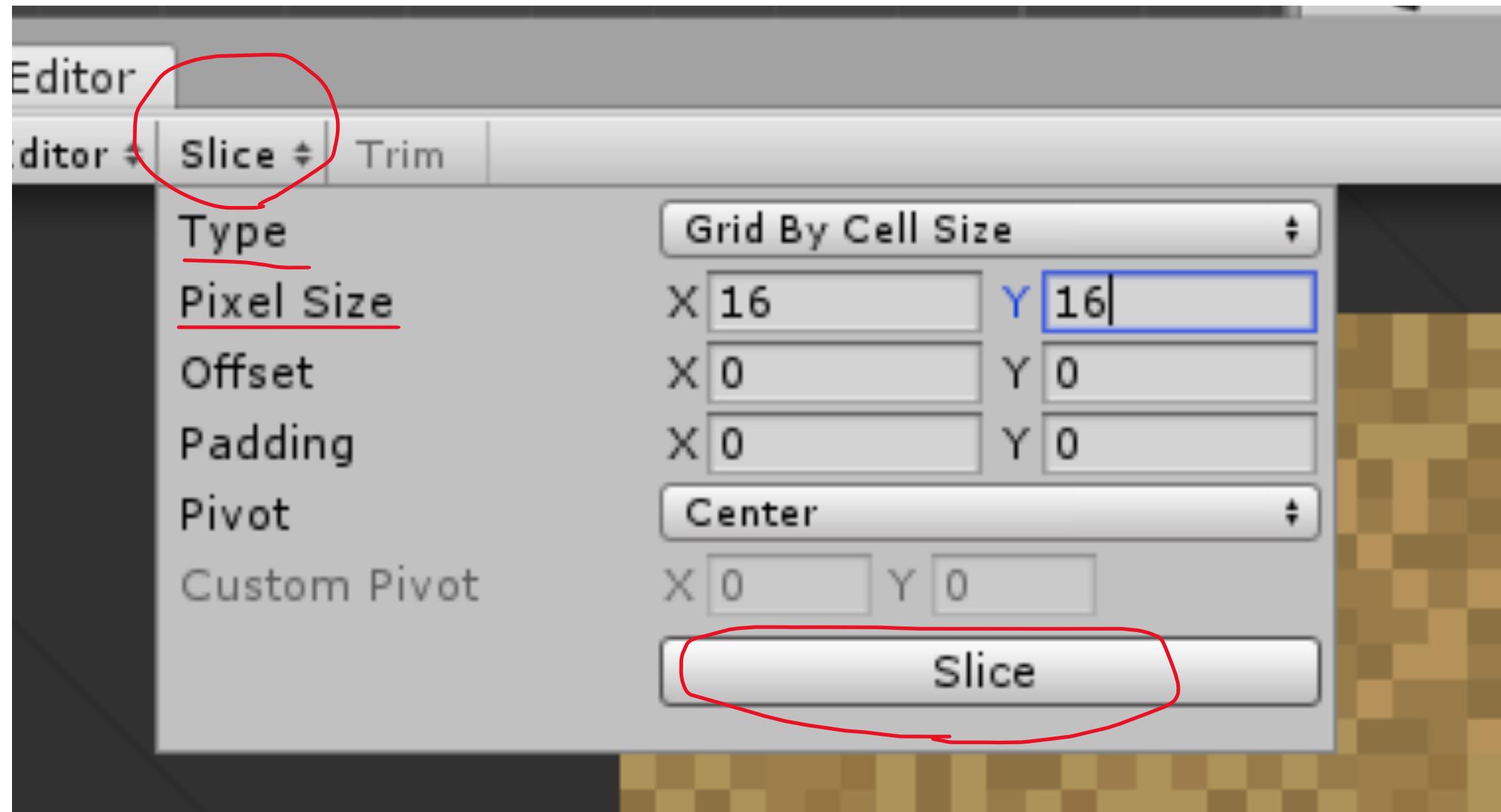


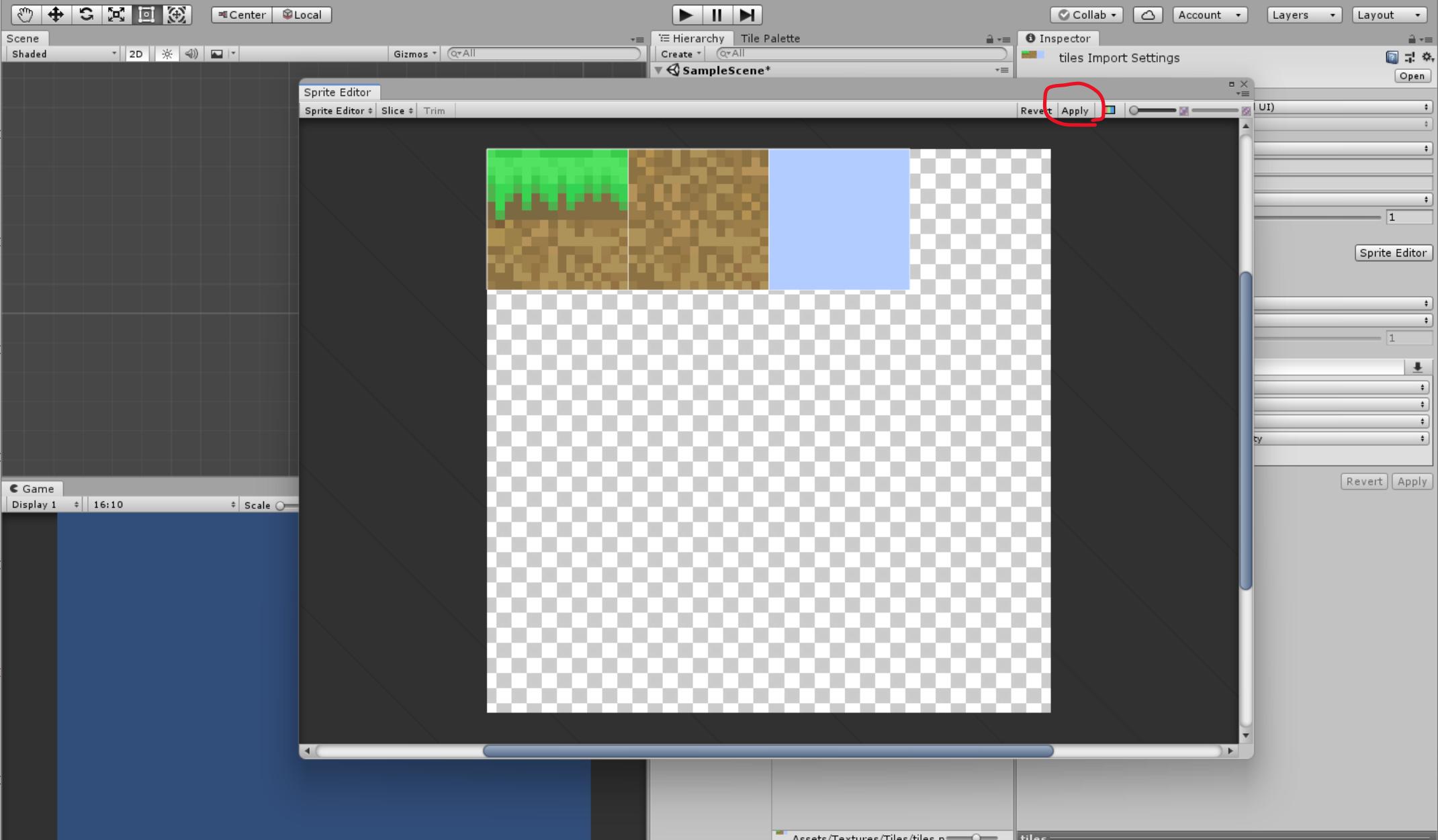
Scale

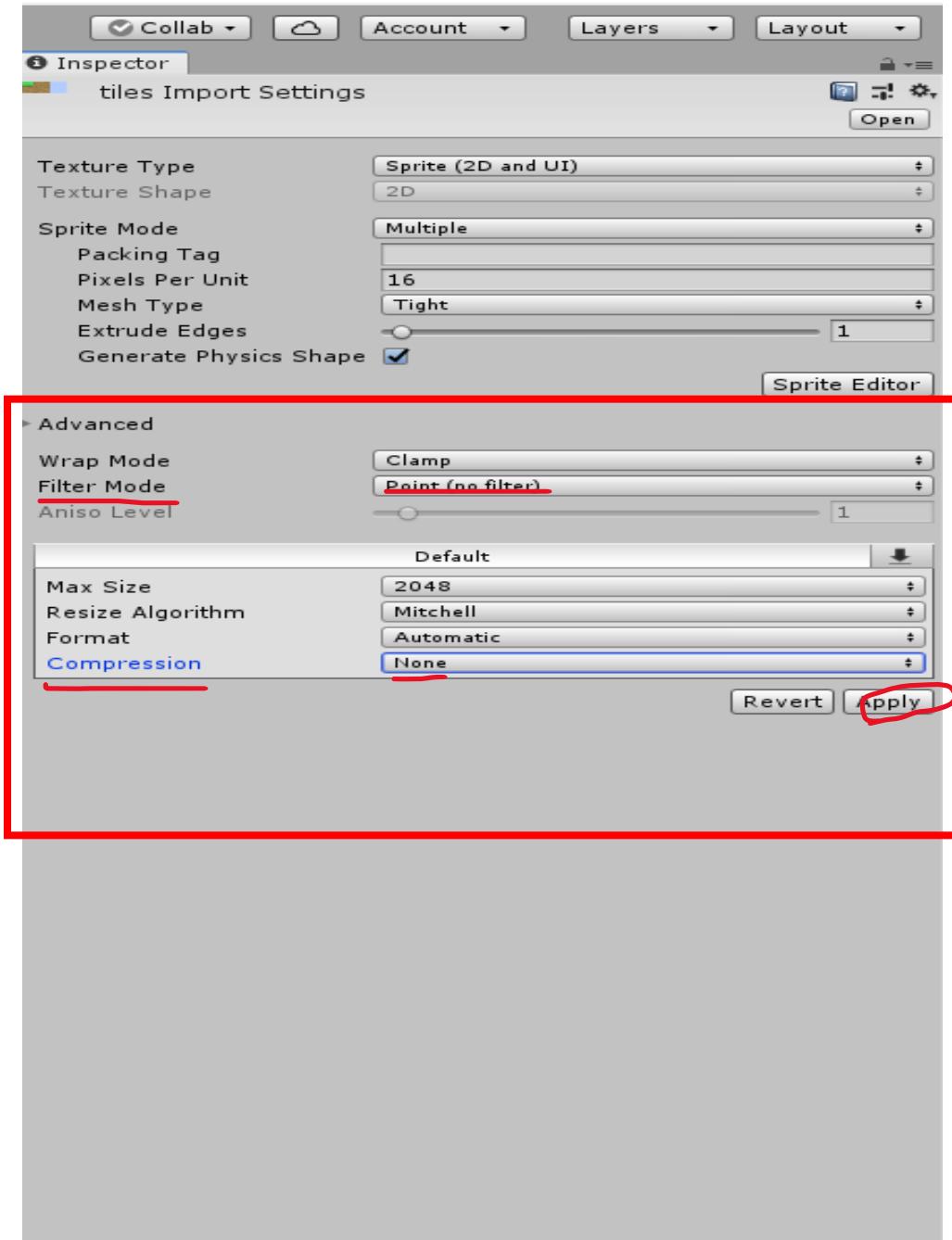
Revert Apply

Assets/Textures/Tiles/tiles.p

tiles





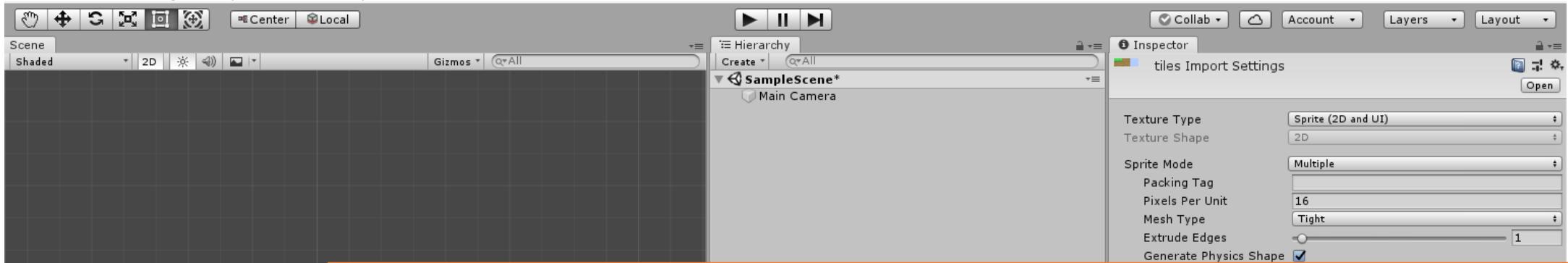


画質が悪い状態を直す

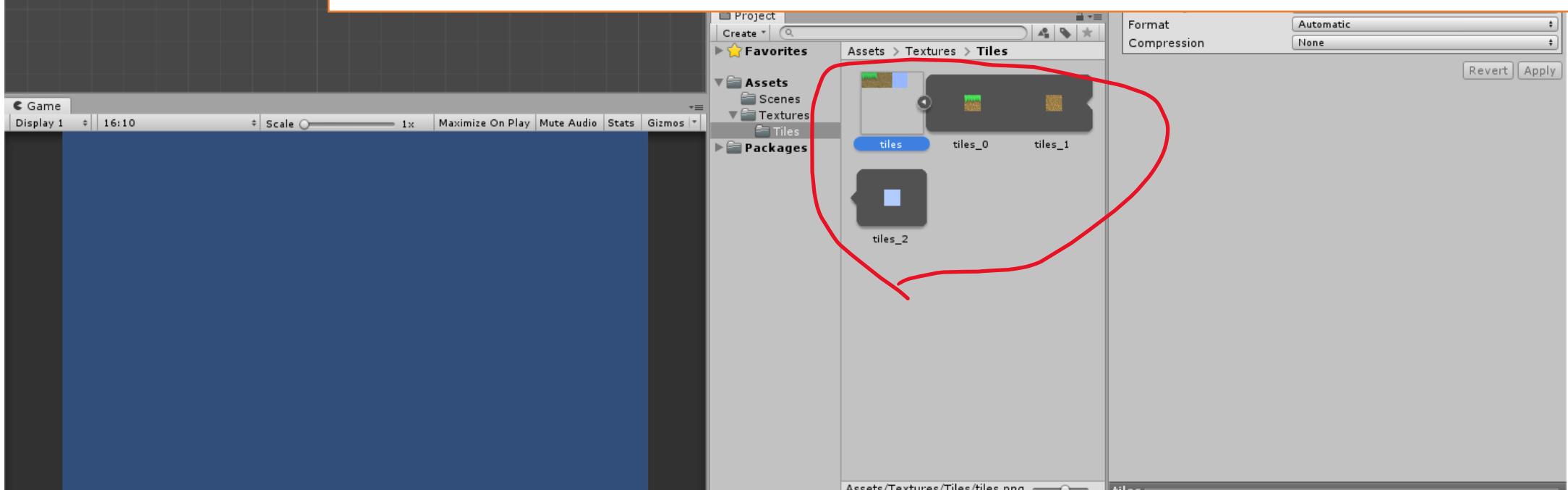
①Filter Mode→point (no filter)

②compression→None

③Applyクリック

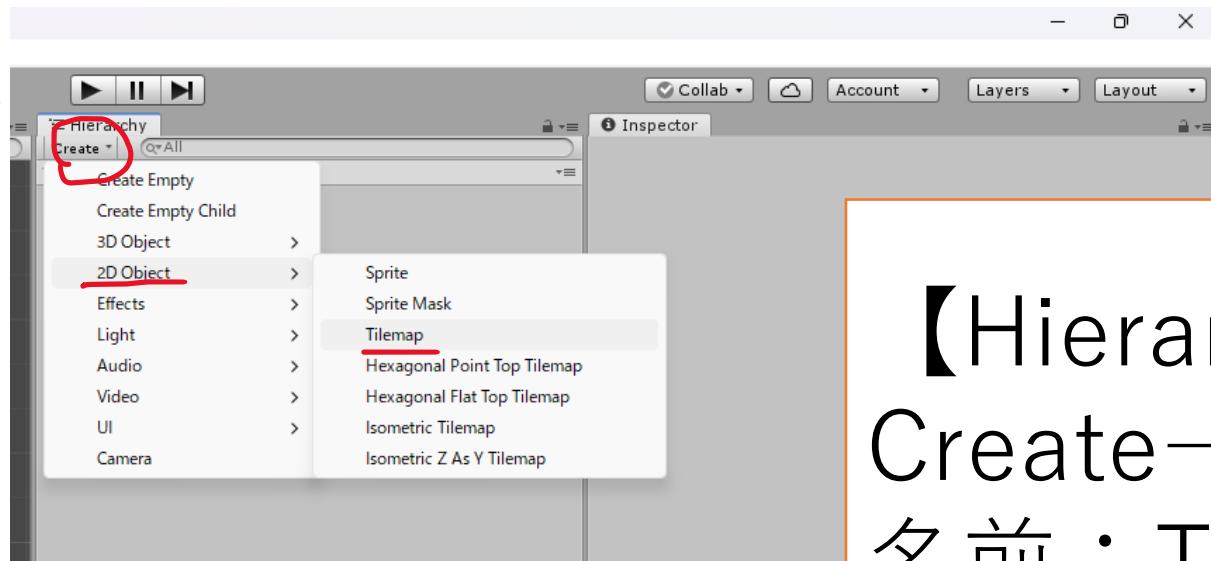


1枚の画像から3枚の画像を切り取り
することができた



タイルマップの作成

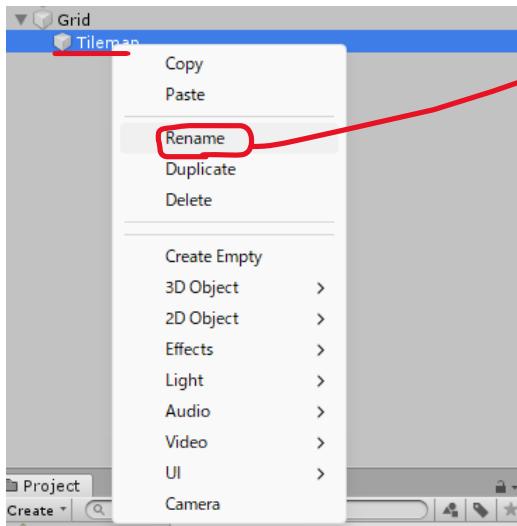
1

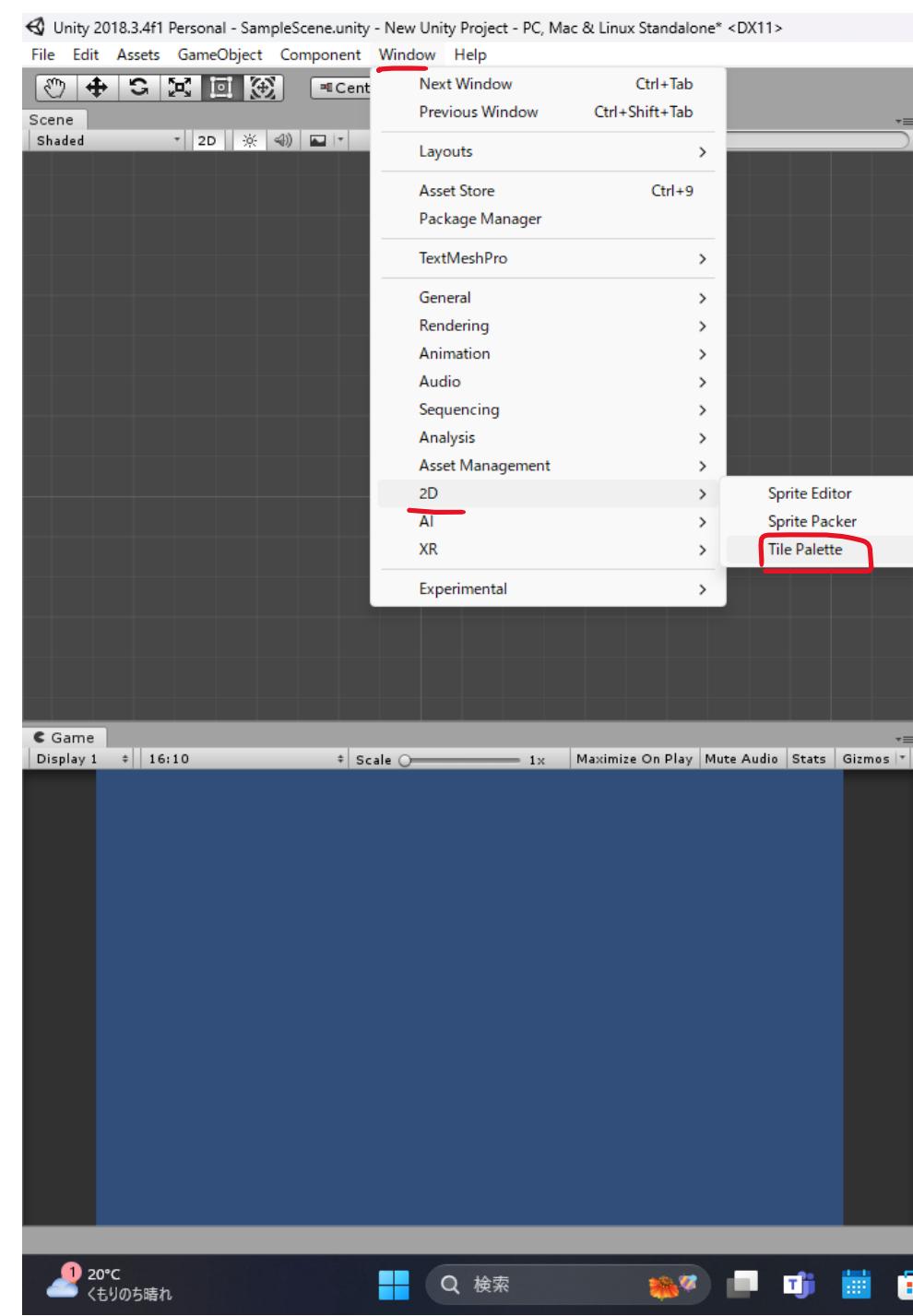


【Hierarchy】
Create → 2DObject → Tilemap
名前：Terrain (地形)

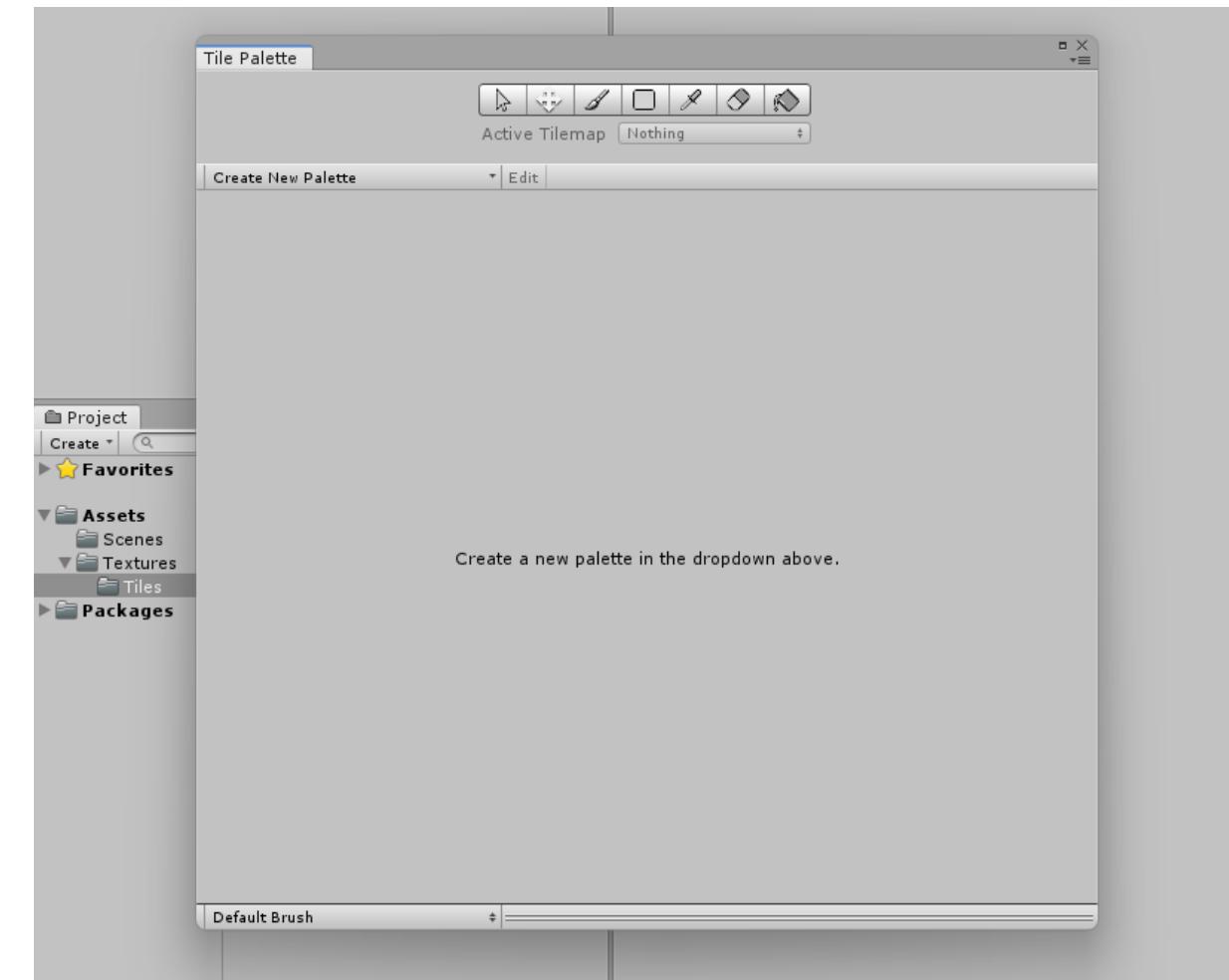
2

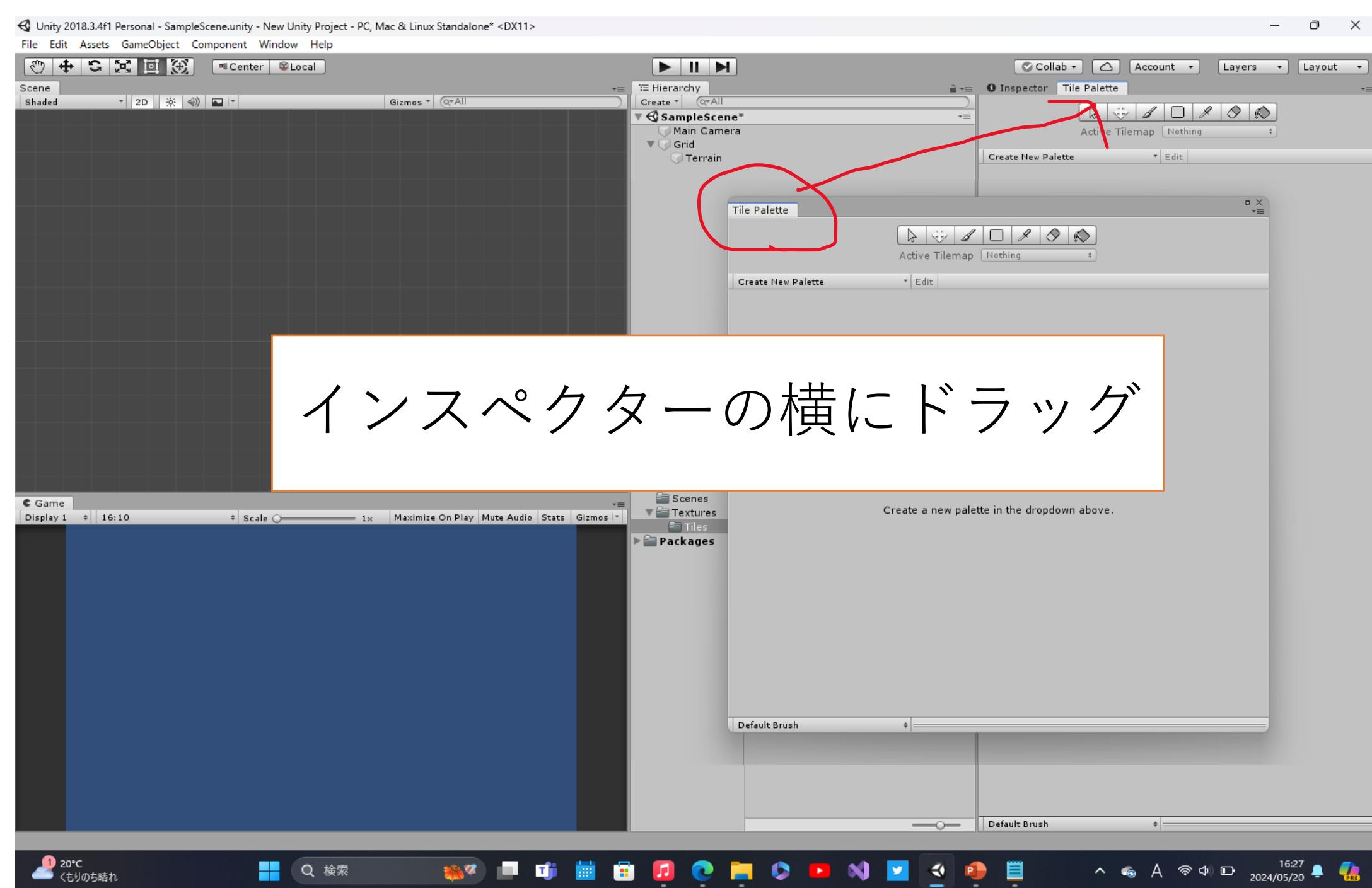
名前変更





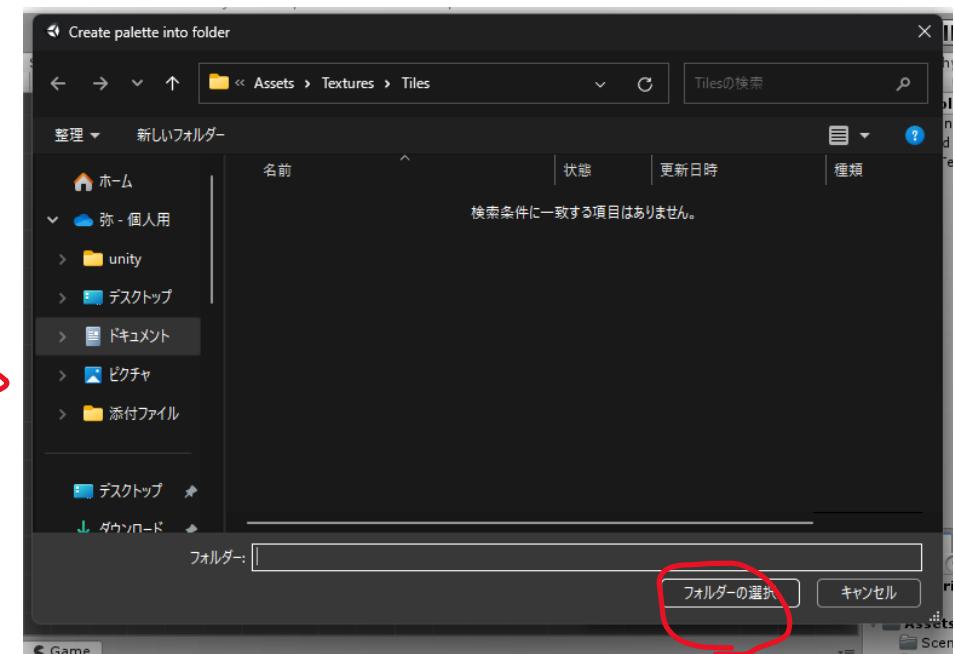
タイルパレットのウィンドウを開く
Window → 2D → Tile Palette

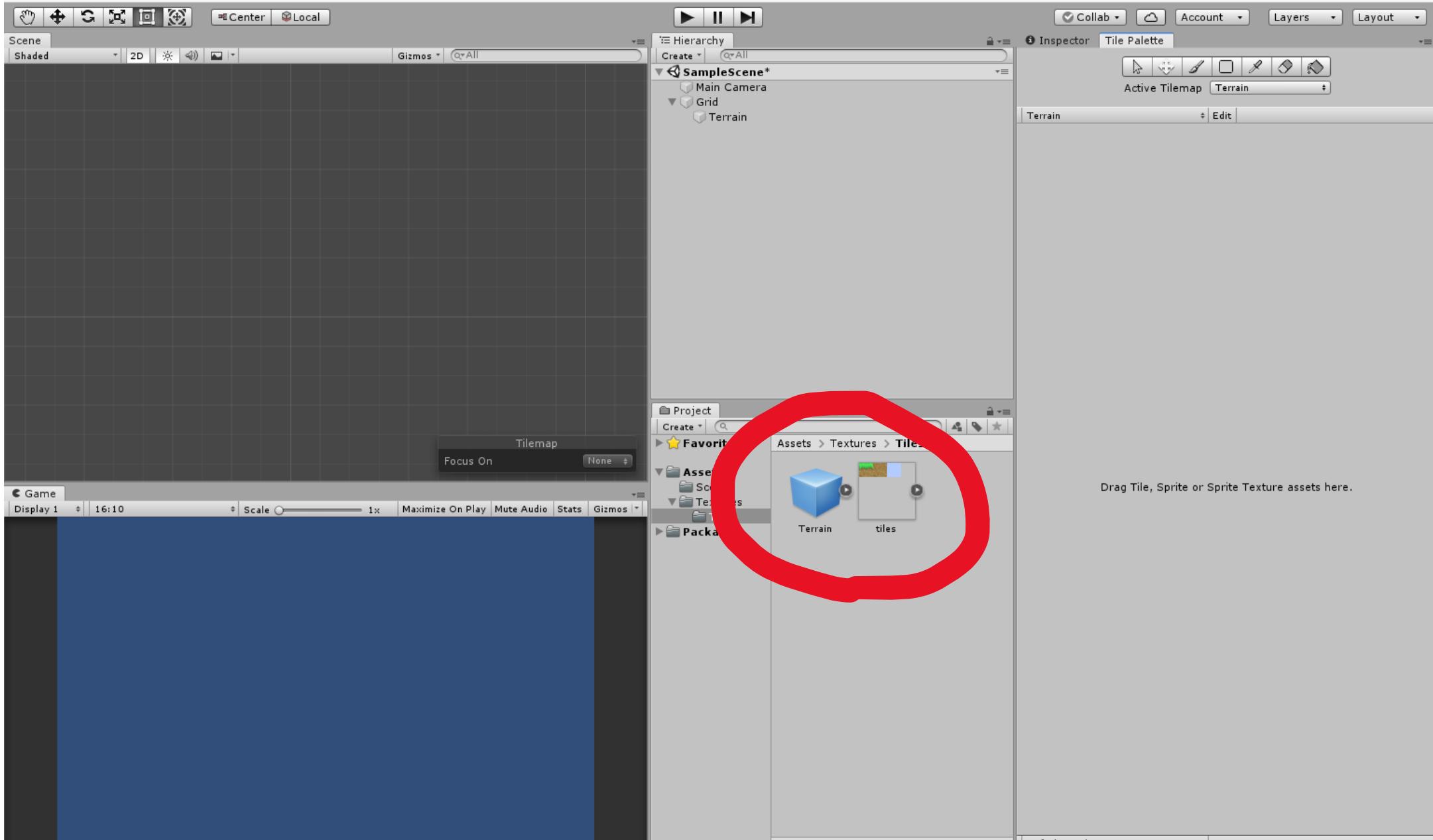


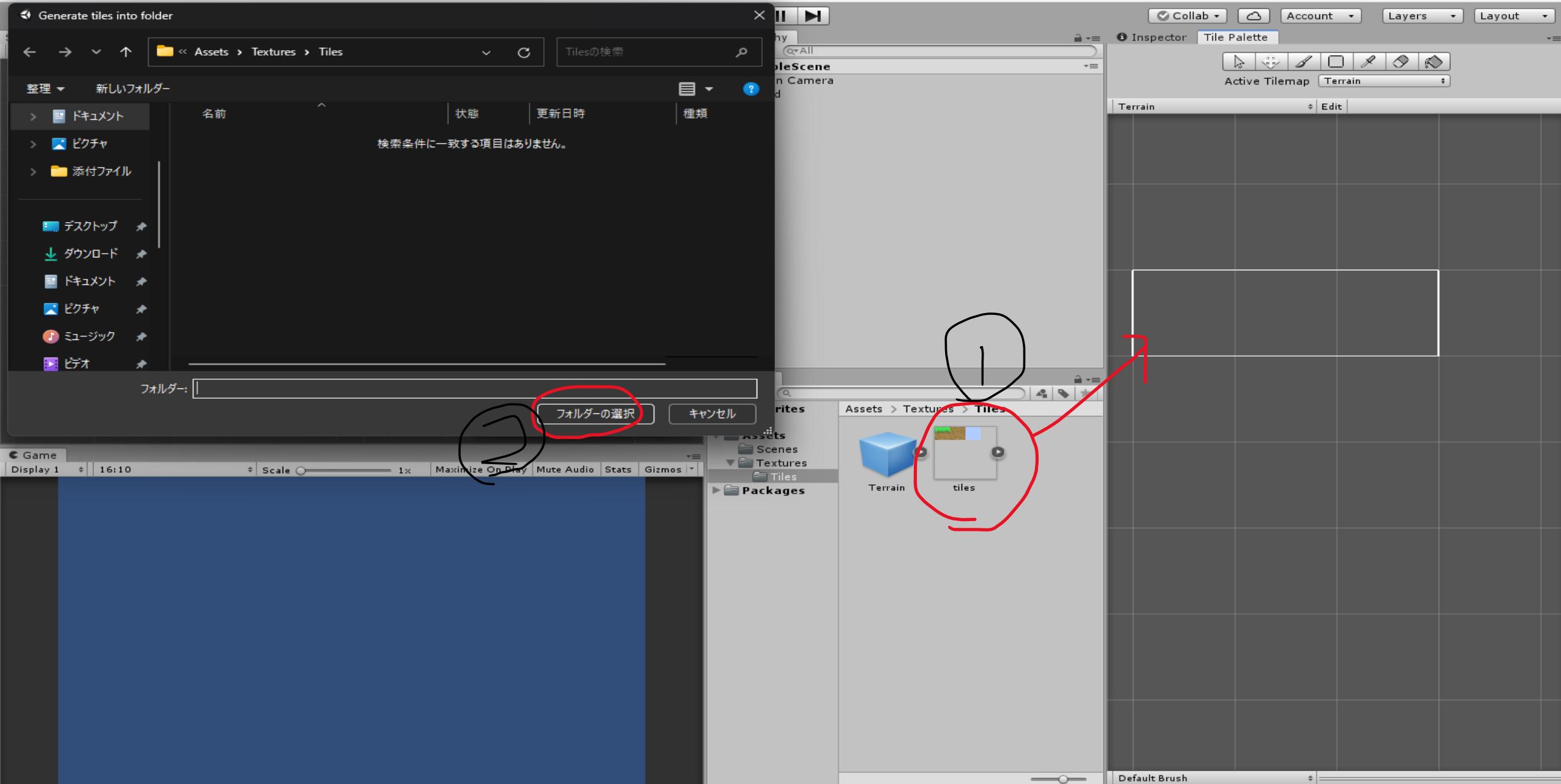




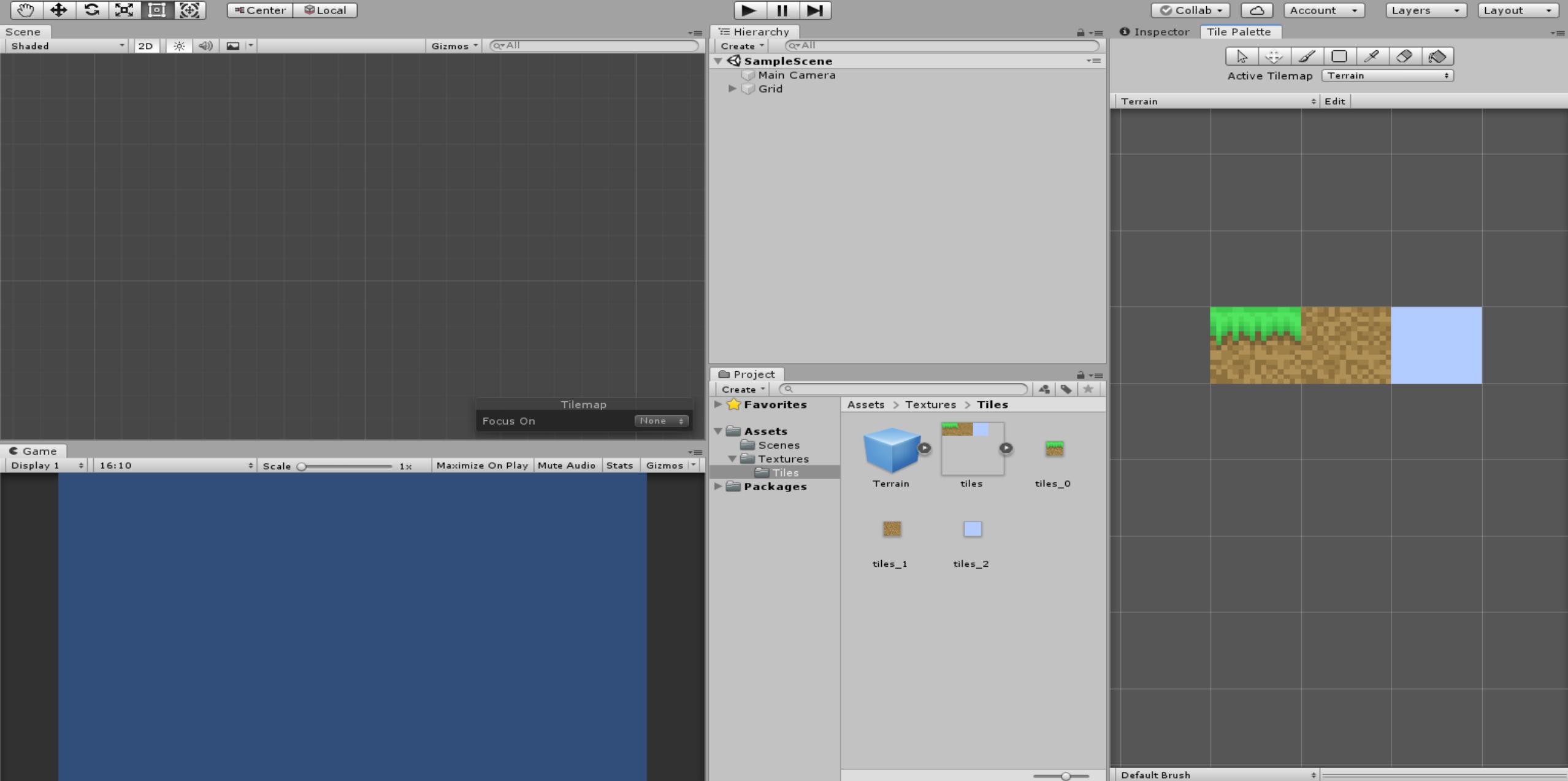
名前 : Terrain

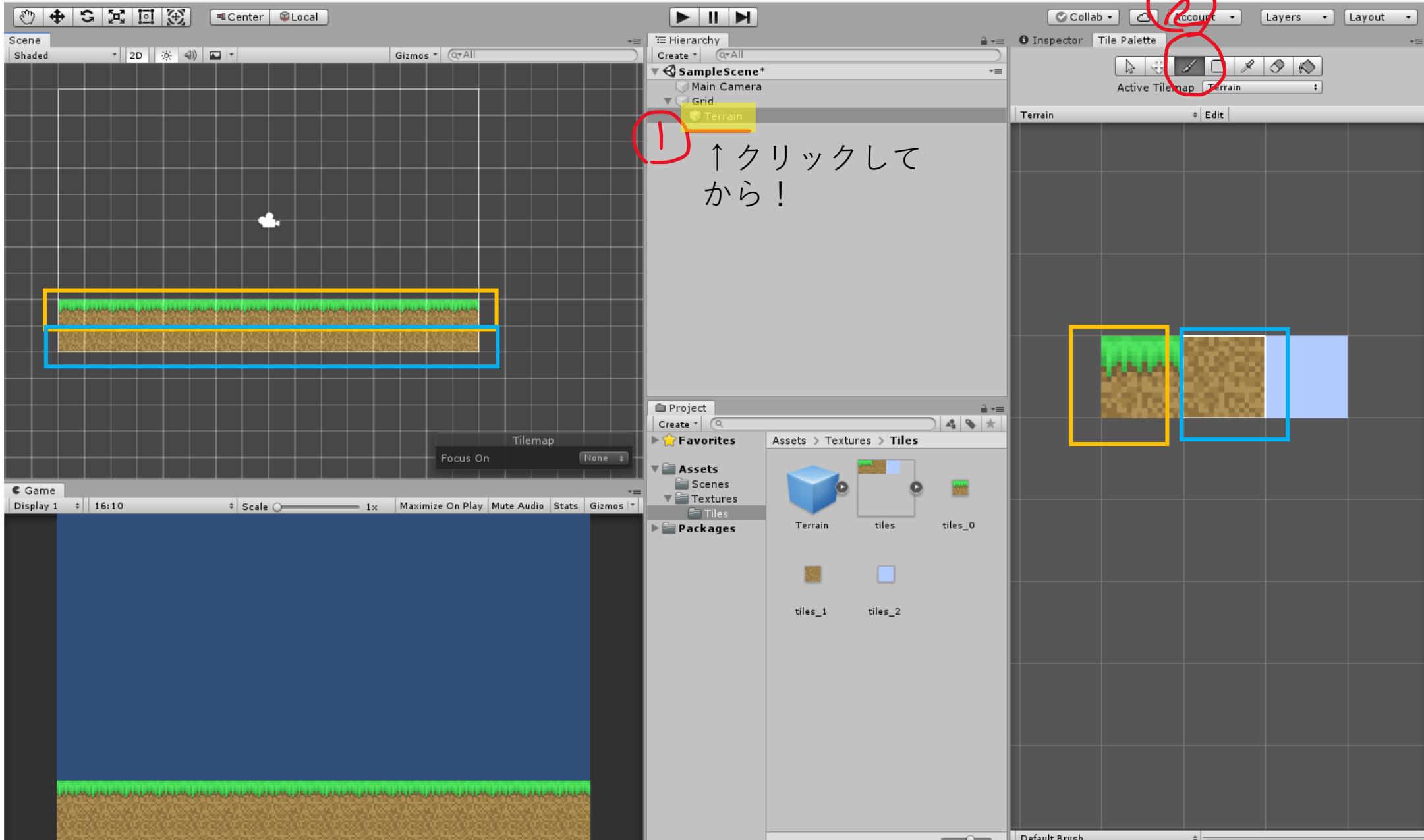




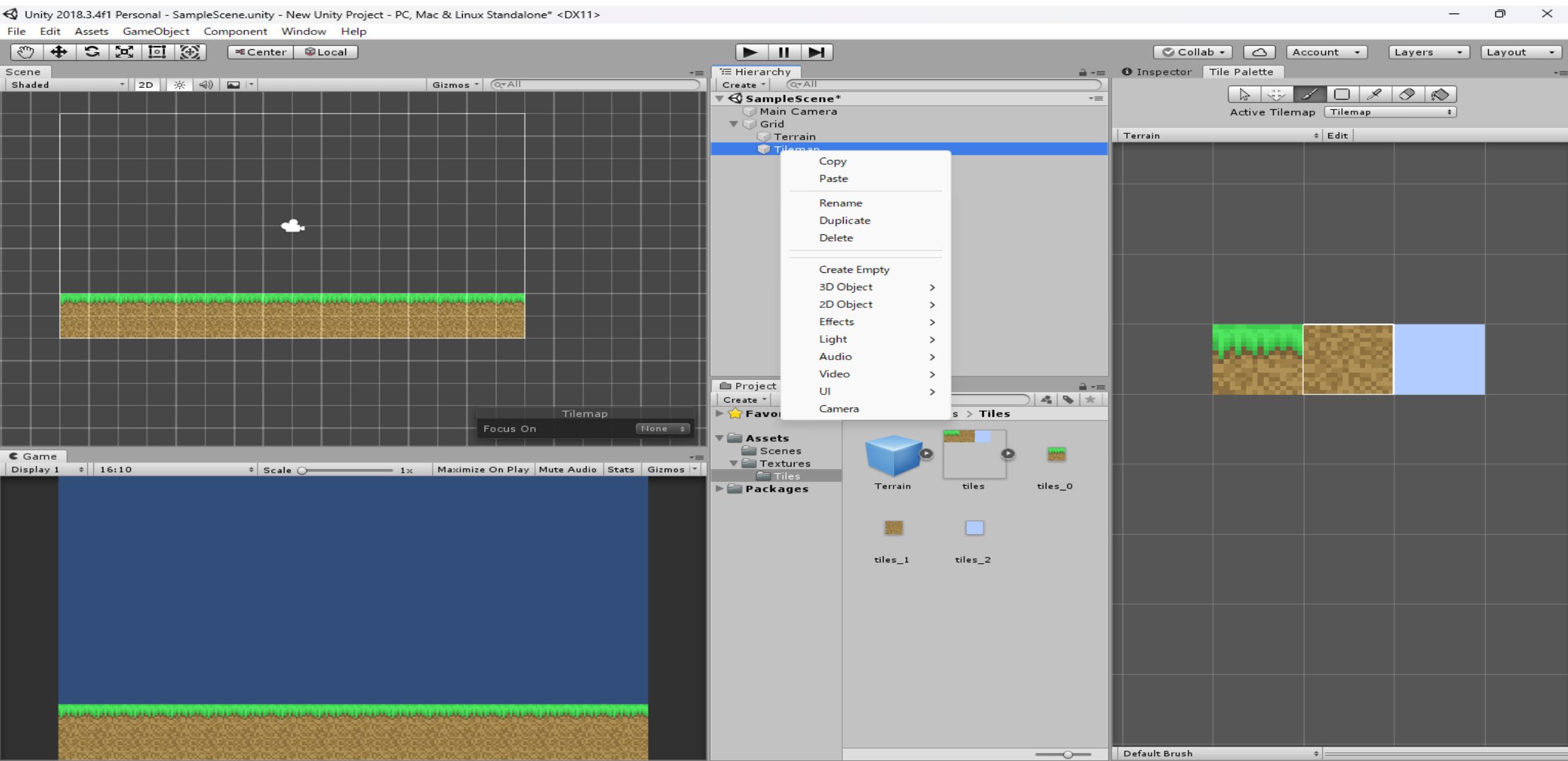


File Edit Assets GameObject Component Window Help





背景用のタイルマップを作成





Scene

Shaded

2D

3D

Audio

Gizmos

Gizmos

Center

Local

Hierarchy

Create

All

SampleScene*

- Main Camera
- Grid
- Terrain
- Background

Collab Account Layers Layout

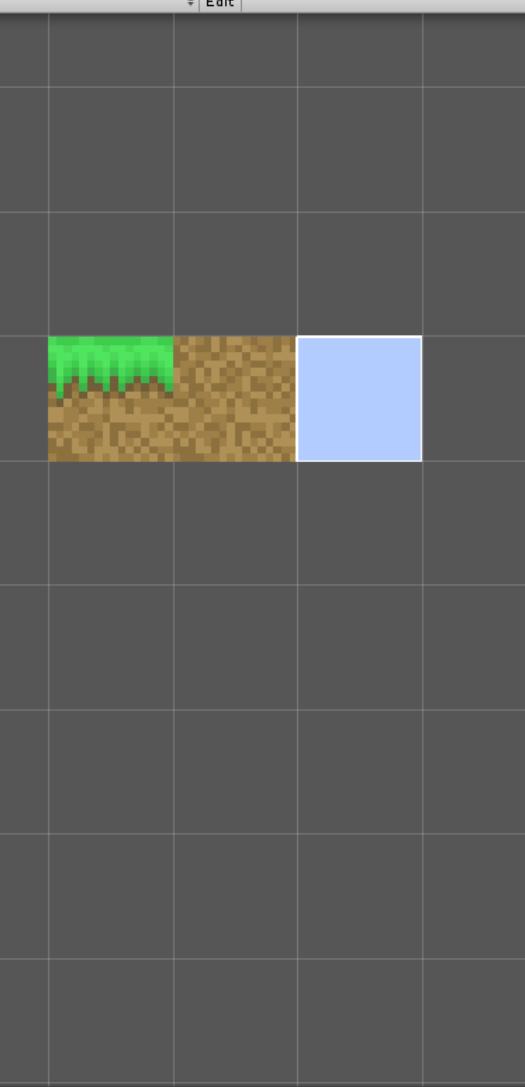
Inspector Tile Palette



Active Tilemap Background

Terrain

Edit

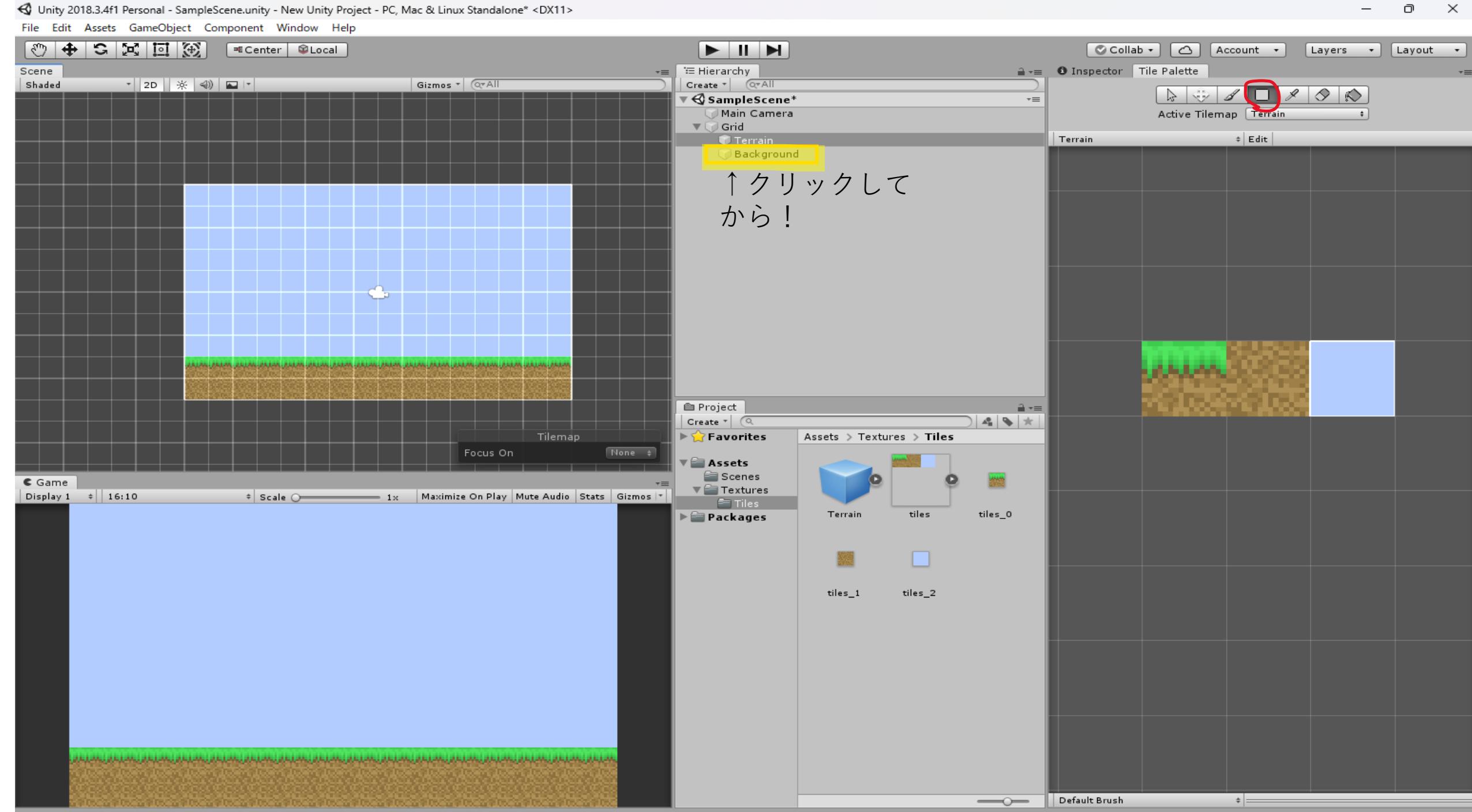


【Hierarchy】

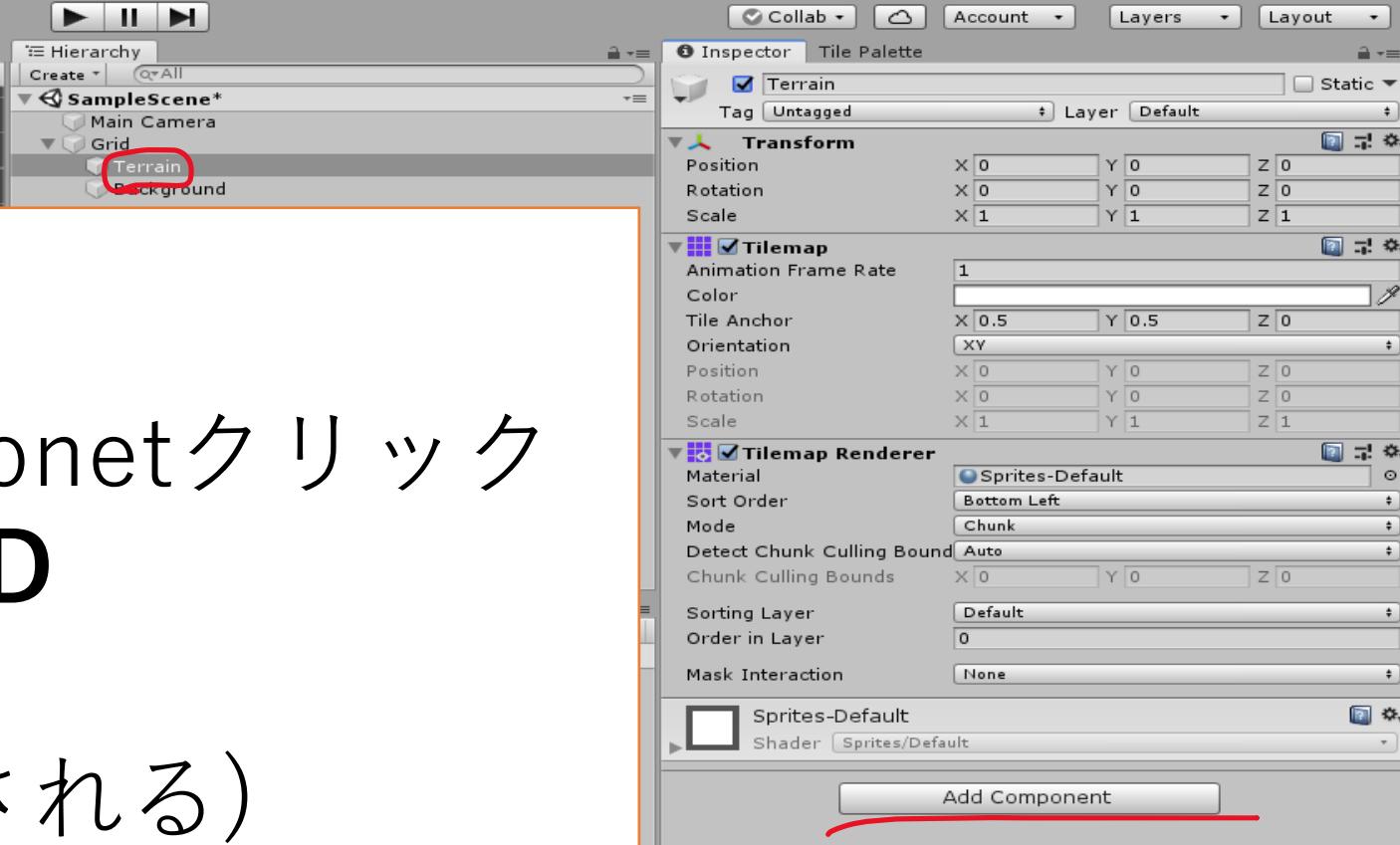
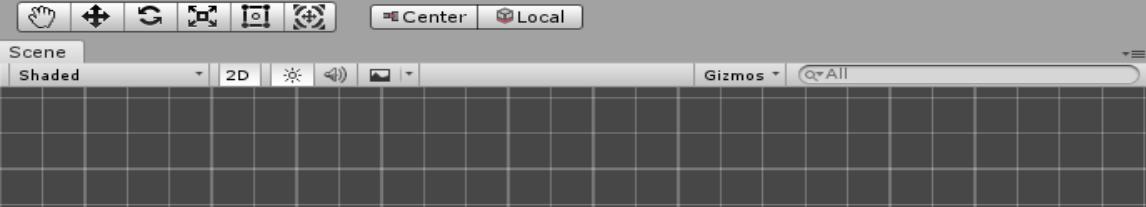
Create → 2DObject → Tilemap

名前 : Background





当たり判定



【Inspector】

Terrain選択 → Add Componentクリック

- **Composite Collider 2D**



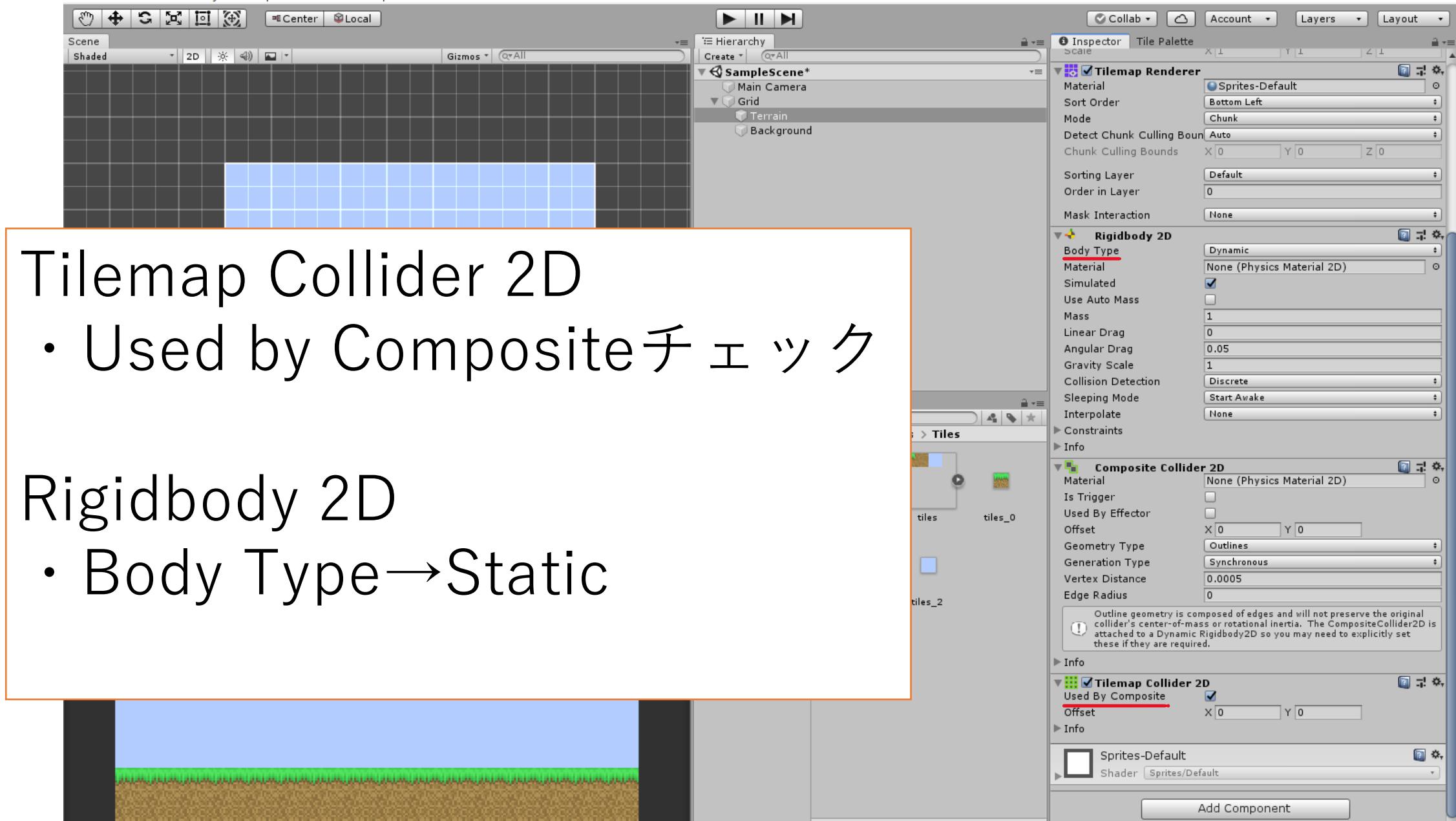
(Rigidbody 2Dも追加される)



- **Tilemap Collider 2D**

: タイルマップ用の当たり判定を作る





Tilemap Collider 2D

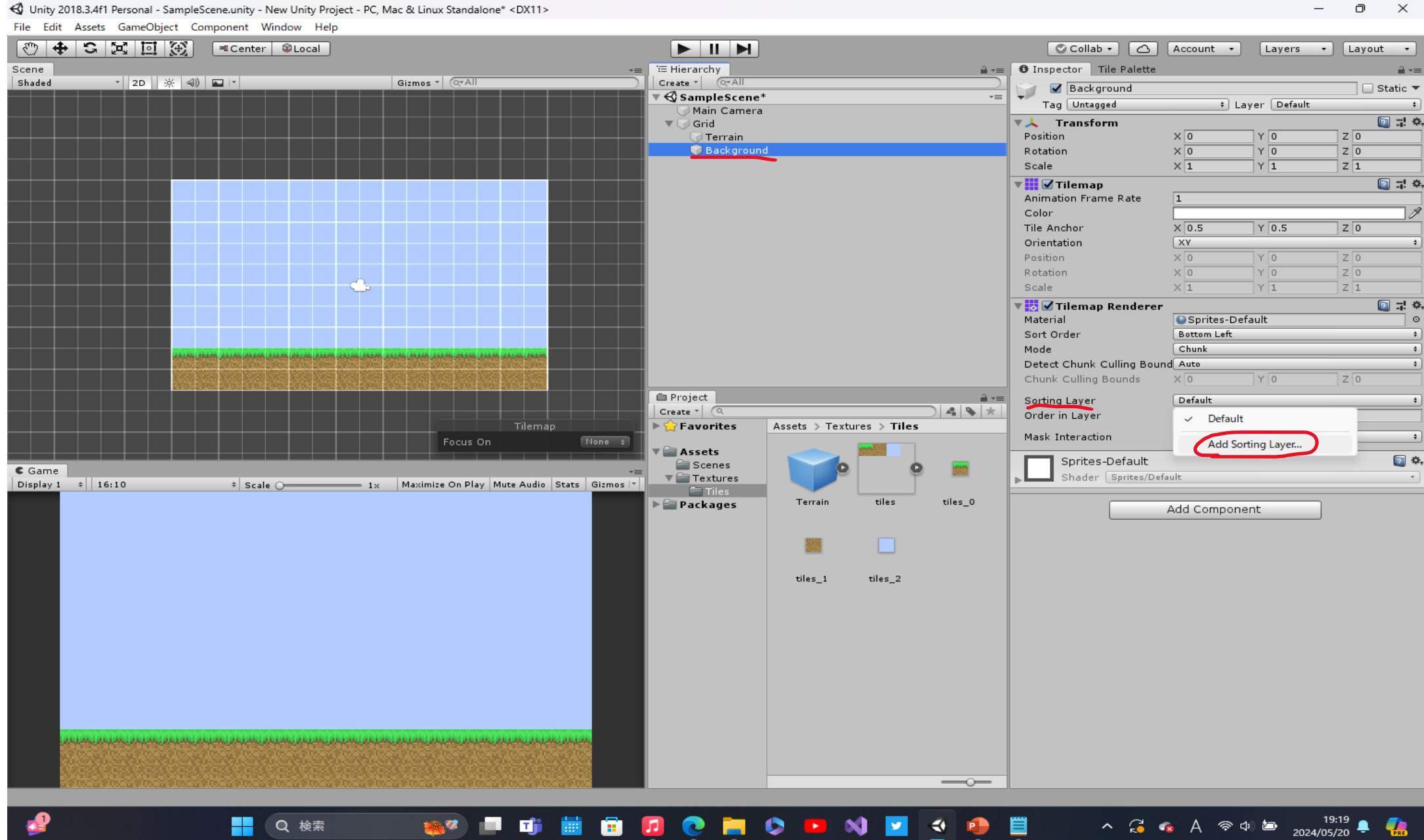
- Used by Composite チェック

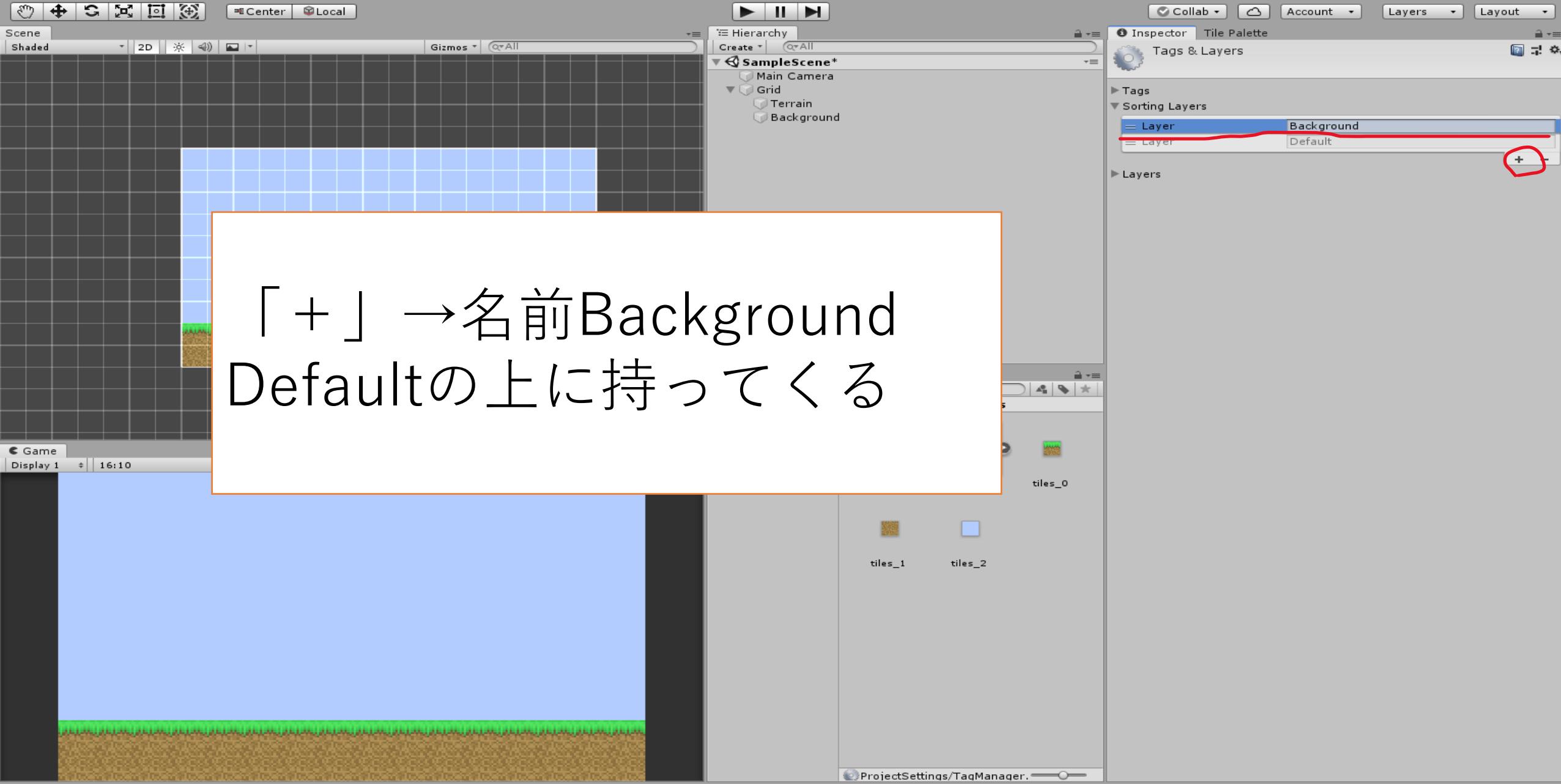
Rigidbody 2D

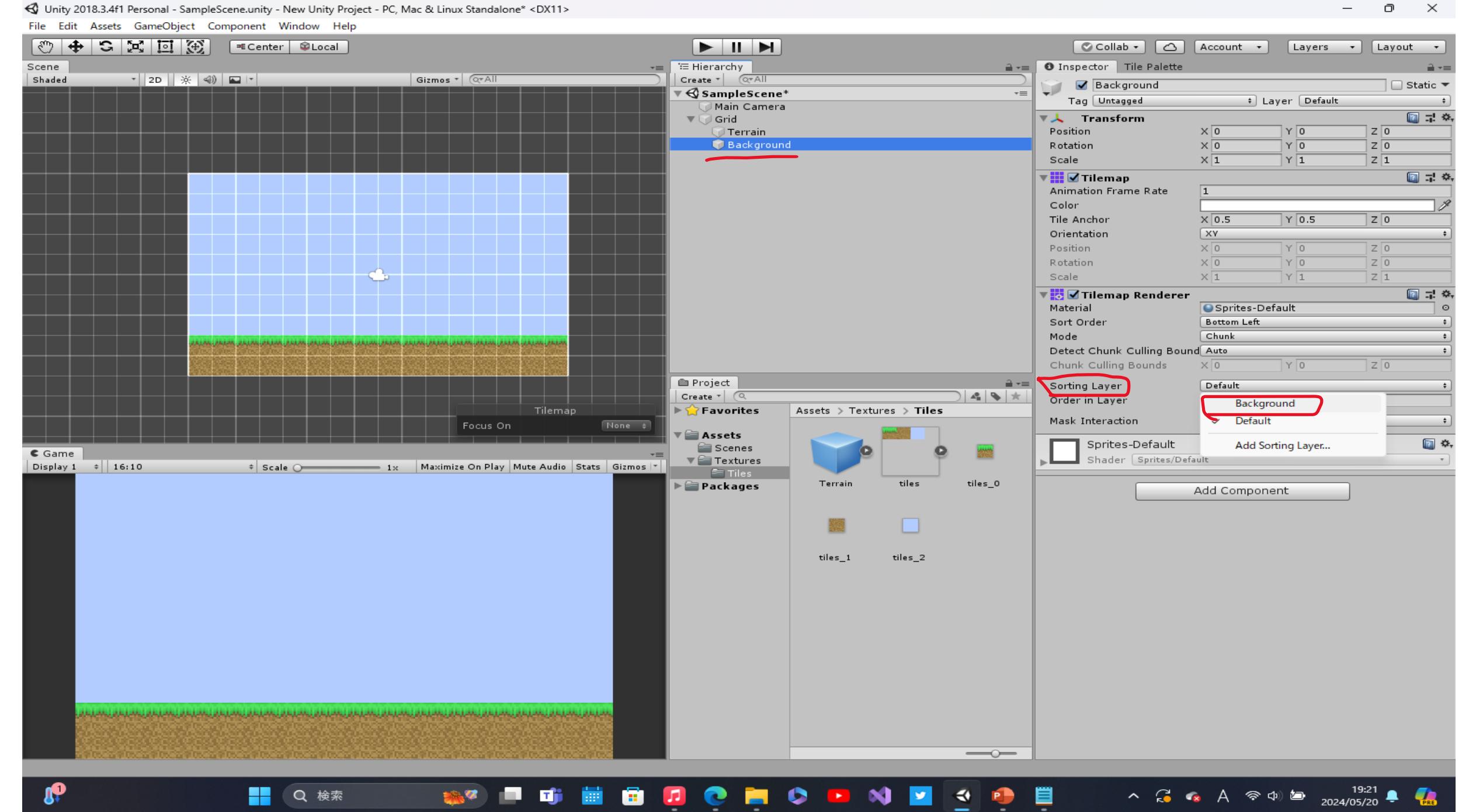
- Body Type → Static



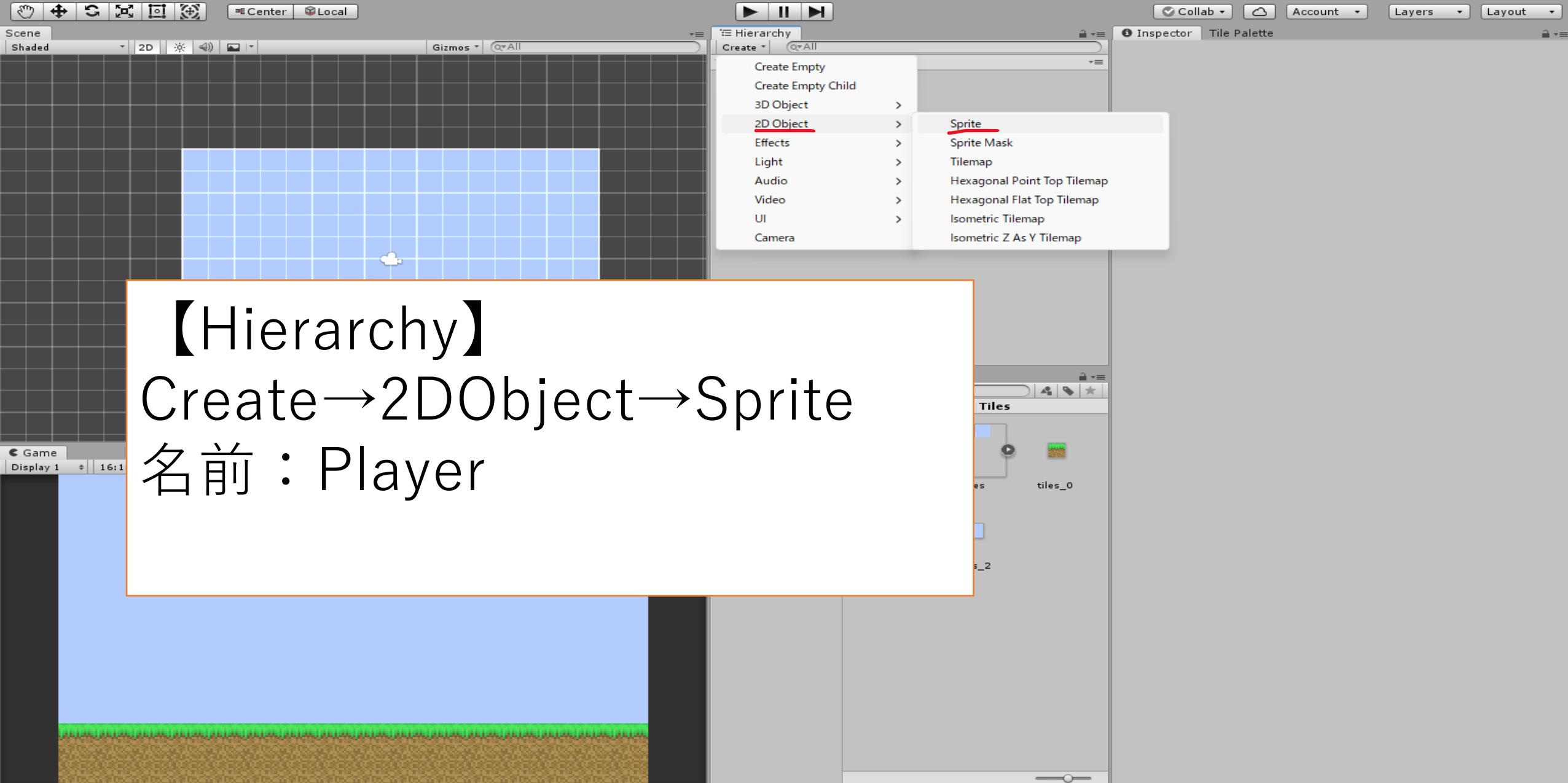
Playerが背景の後ろに隠れているのを修正する







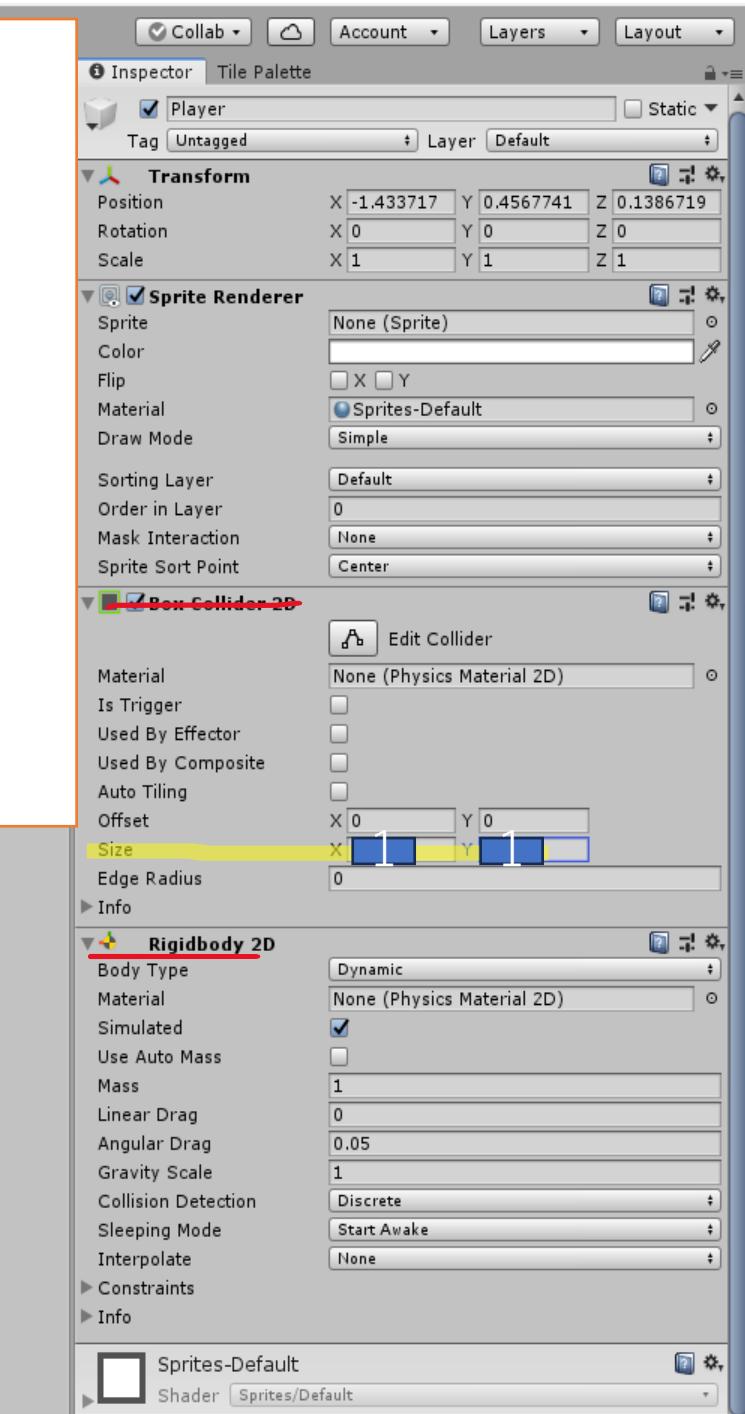
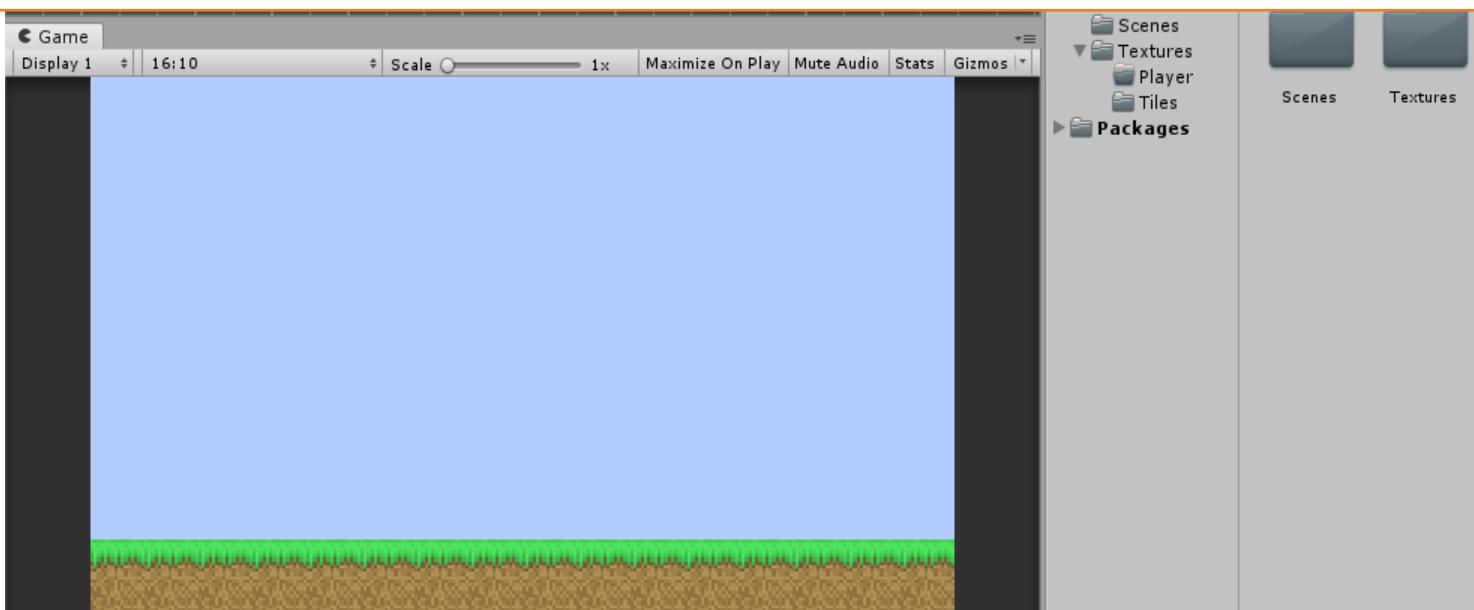
プレイヤーの作成



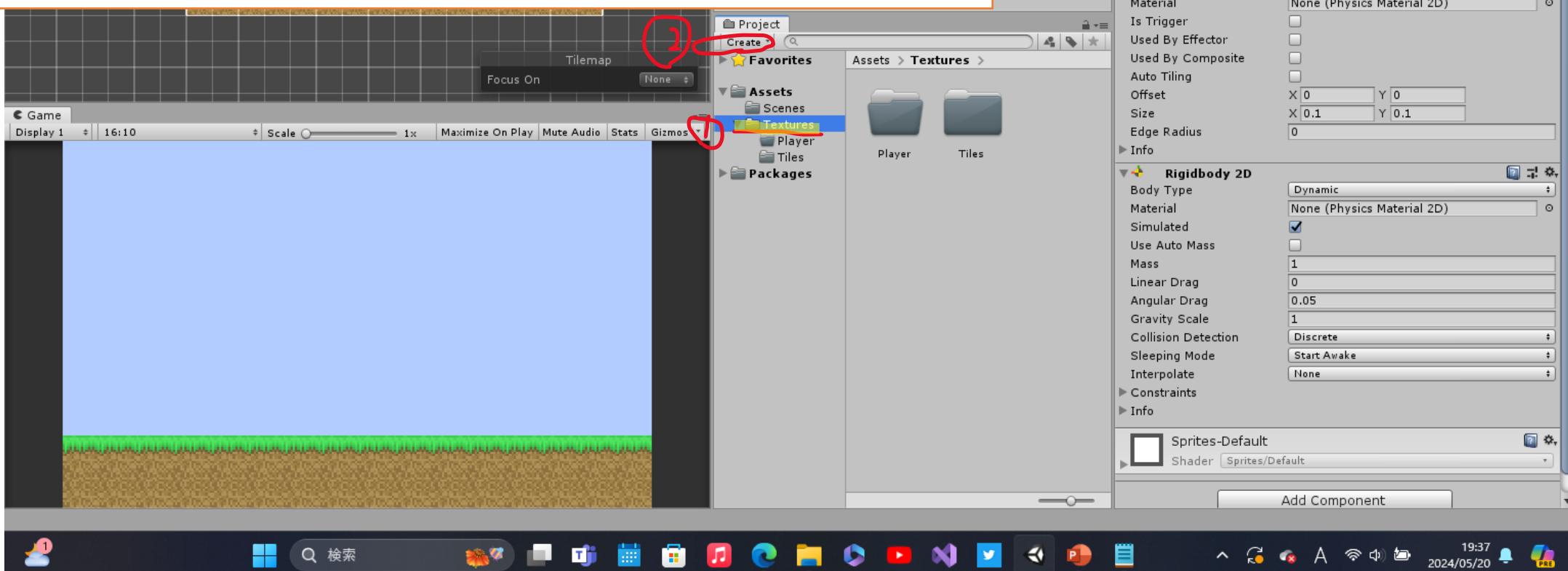
【Inspector】

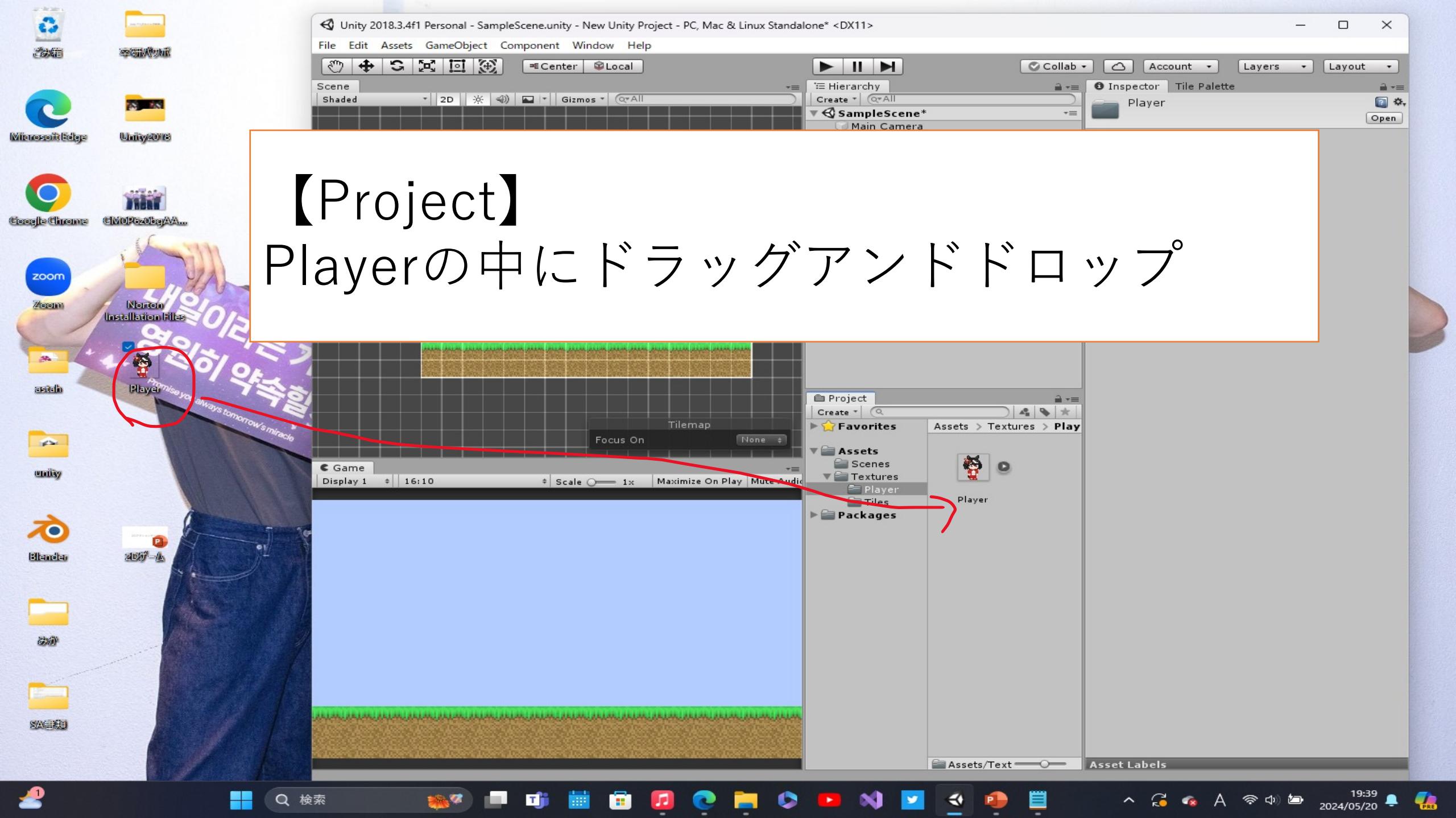
Player選択→Add Componetクリック

- Rigidbody 2D 追加
 - Box Collider 2D 追加
- * Size→ x 1、 y 1



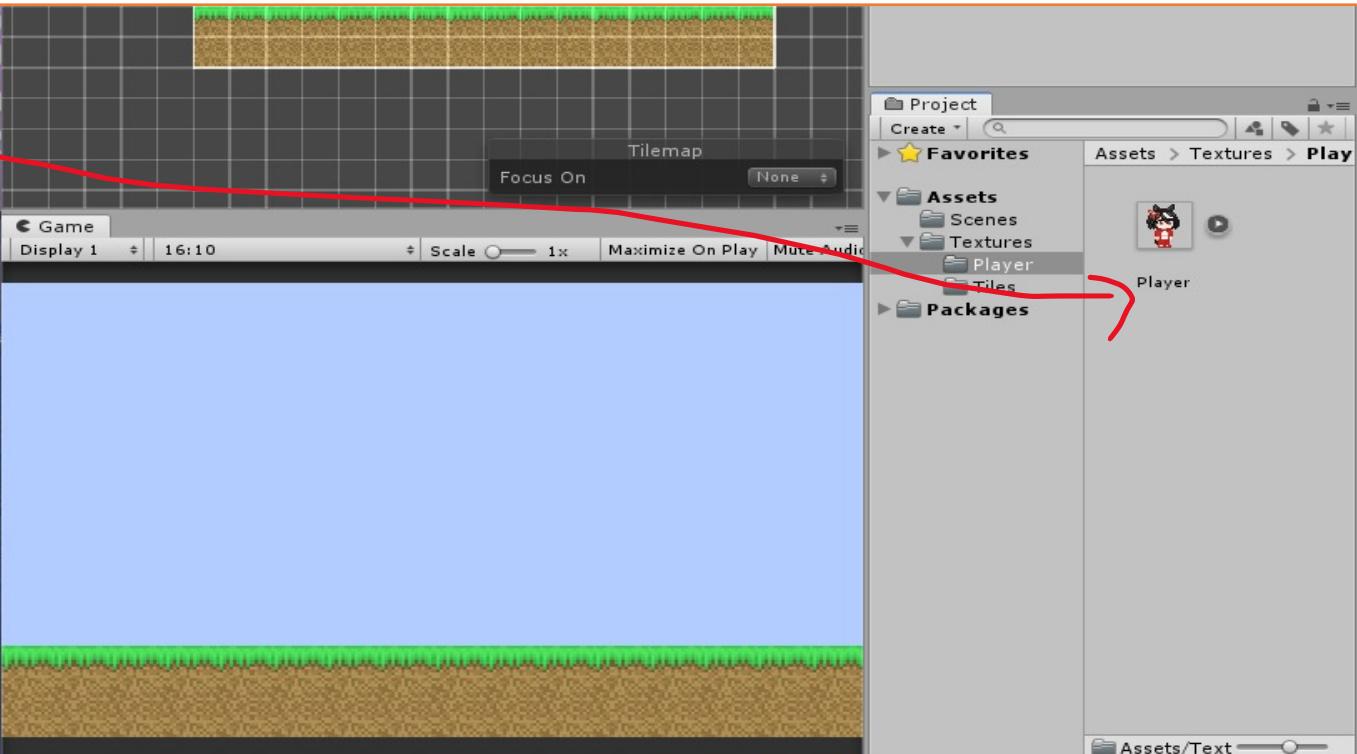
【Project】
Assetsの中のTexturesを選択
Create→folder
名前：Player

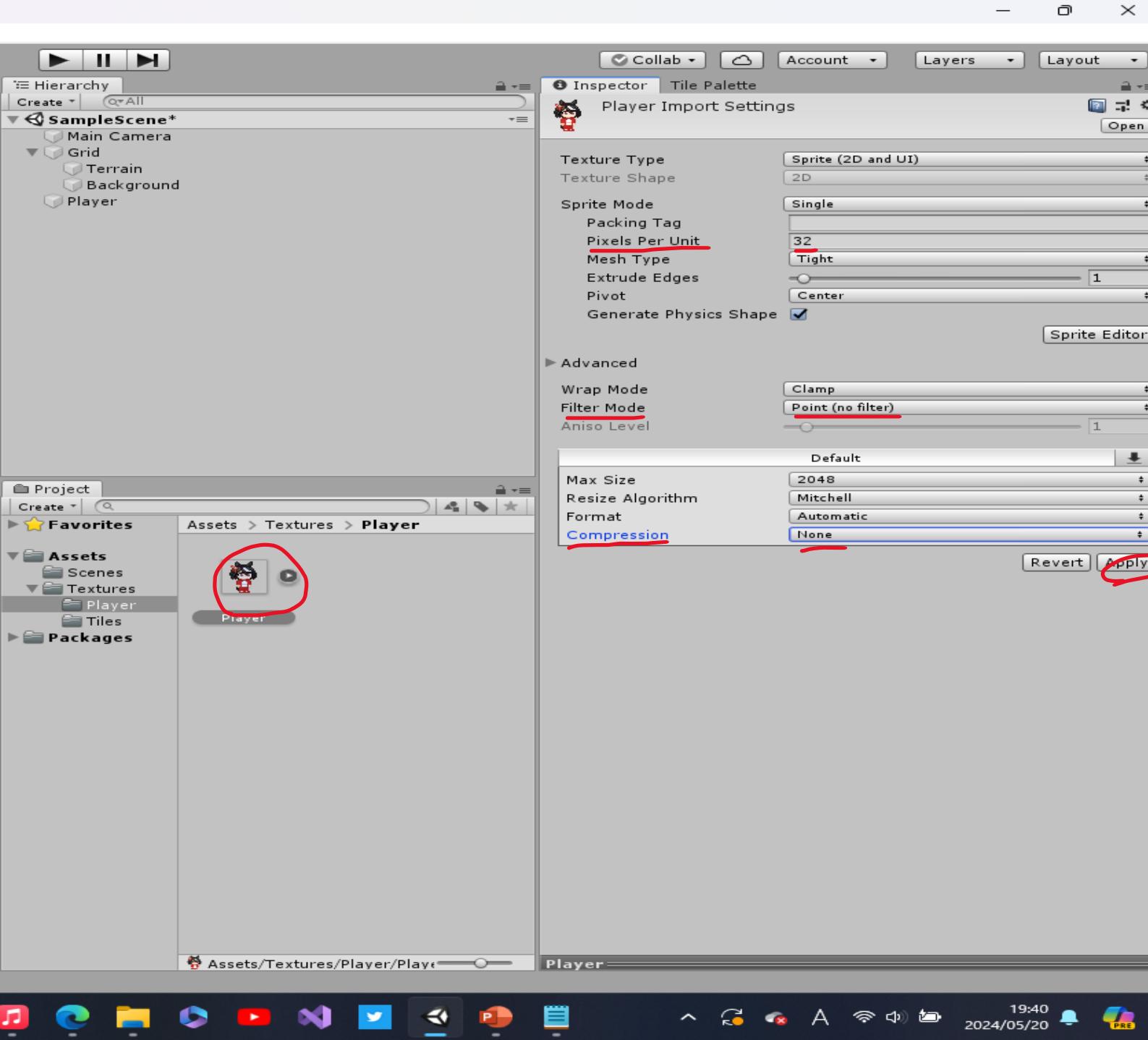




【Project】

Playerの中にドラッグアンドドロップ





① Pixels Per Unit → 32

② Filter Mode
→ point (no filter)

③ compression → None

④ Apply クリック

The screenshot shows the Unity Editor interface with a 2D game scene open. The Scene View on the left displays a blue sky with a small white cloud and a green ground layer. The Hierarchy View shows the scene structure:

- SampleScene*
- Main Camera
- Grid
- Terrain
- Background
- Player

The Player object is selected in the Hierarchy View. The Inspector View on the right provides detailed settings for the Player object:

- Transform**: Position X: -1.433717, Y: 0.4567741, Z: 0.1386719; Rotation X: 0, Y: 0, Z: 0; Scale X: 1, Y: 1, Z: 1.
- Sprite Renderer**: Enabled, using the "Player" sprite from the Assets/Textures/Player folder. The material is set to "Sprites-Default" and draw mode to "Simple". Sorting Layer is "Default" (order 0), and Mask Interaction is "None".
- Box Collider 2D**: Enabled, using "None (Physics Material 2D)". Offset is (0, 0). Size is (1, 1). Edge Radius is 0.
- Rigidbody 2D**: Enabled, using "Dynamic" body type. Material is "None (Physics Material 2D)". Simulated is checked. Use Auto Mass is checked. Mass is 1. Linear Drag is 0, Angular Drag is 0.05, Gravity Scale is 1. Collision Detection is "Discrete". Sleeping Mode is "Start Awake". Interpolate is checked. Constraints are "None".

A red arrow points from the "Player" sprite in the Project View to the "Sprite" field in the Inspector View. A red circle highlights the "Player" sprite in the Project View.

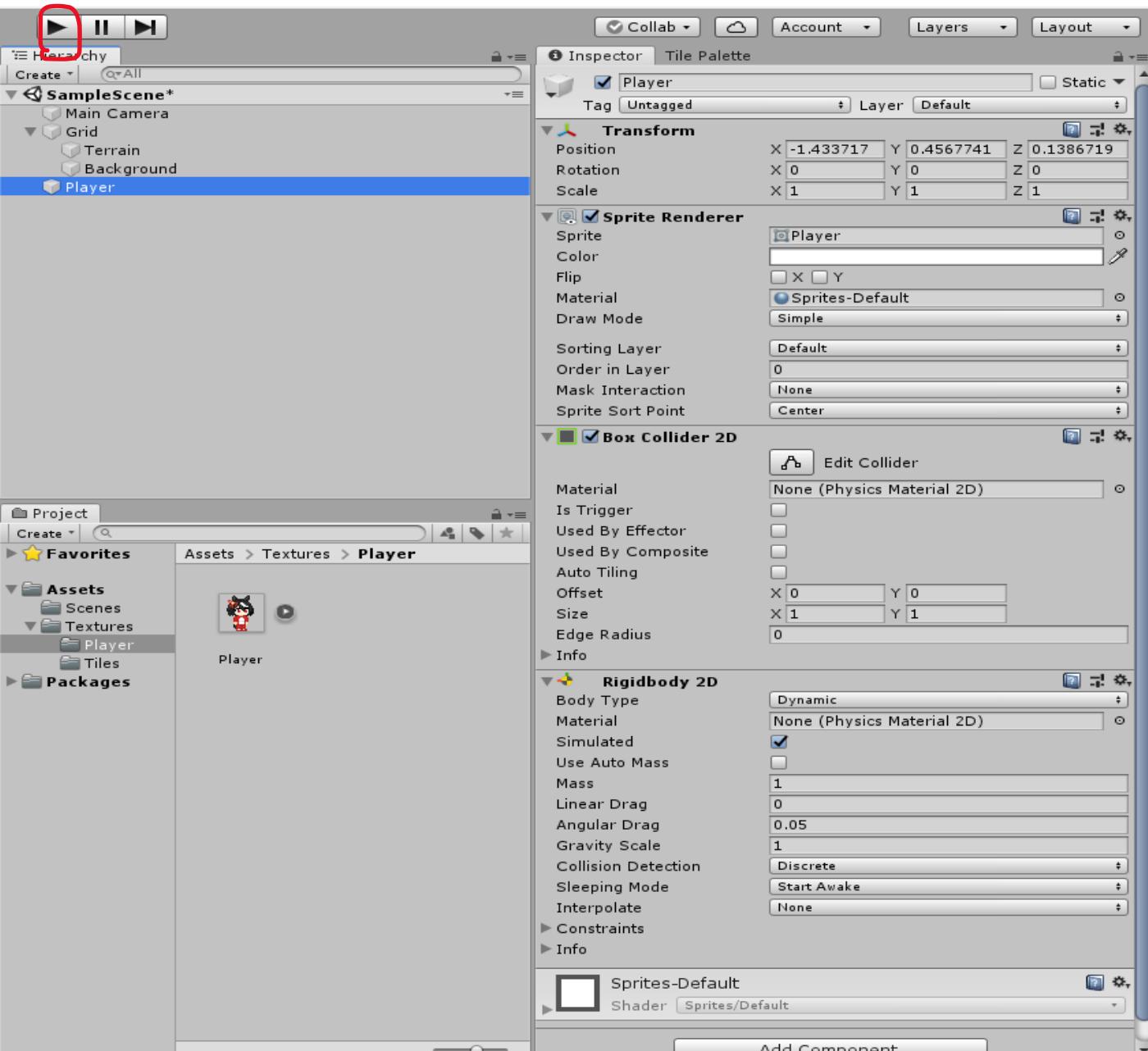
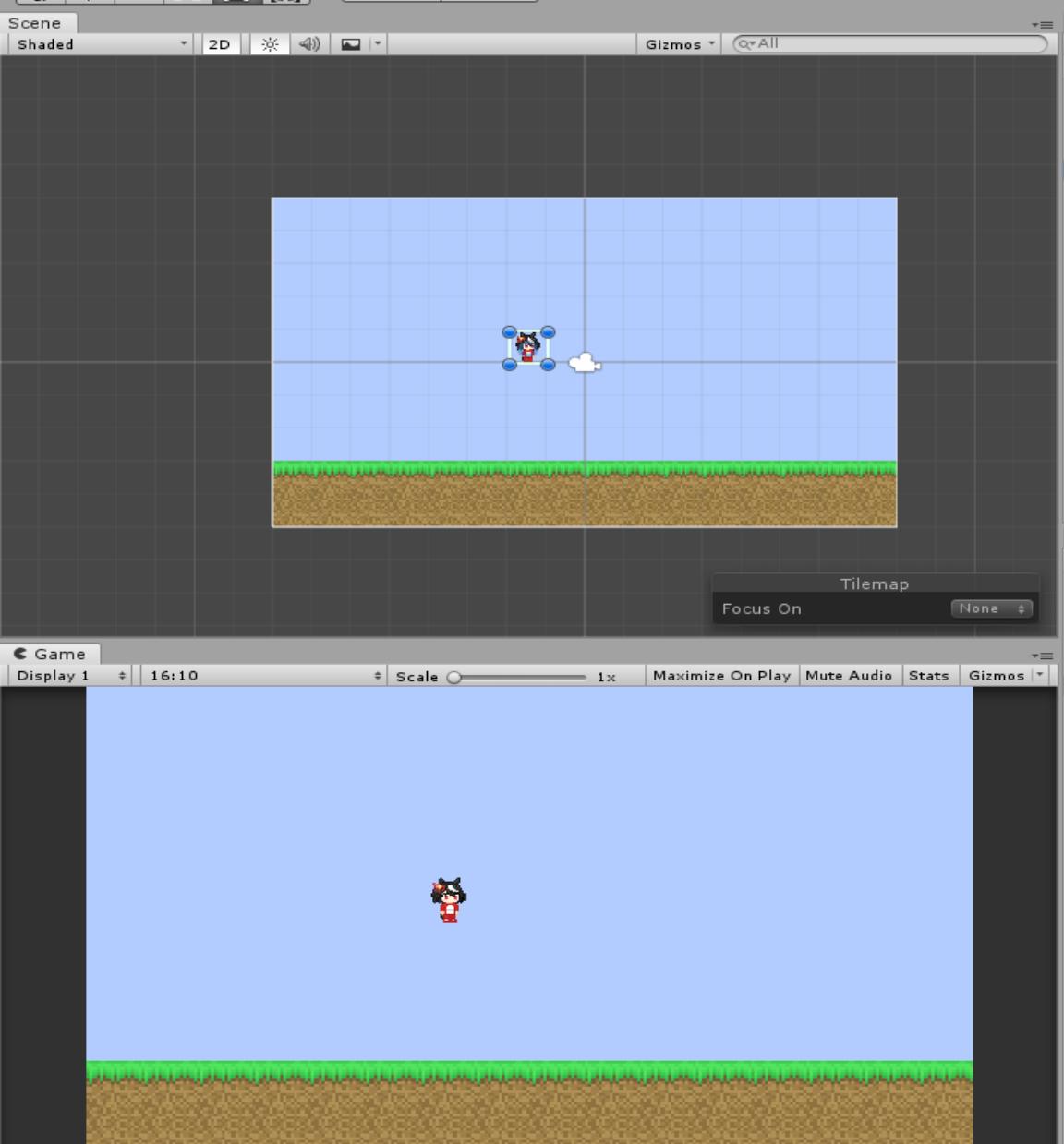


検索

19:44
2024/05/20



Center Local



検索

19:48
2024/05/20

The Unity Editor interface showing a 2D game scene titled "SampleScene".

Scene View: Displays a 2D tilemap with a grid, a small white cloud, and a player character standing on a green grass tile. A red arrow points from the bottom left towards the player character.

Hierarchy: Shows the scene structure:

- SampleScene*
- Main Camera
- Grid
- Terrain
- Background
- Player

Inspector: Provides detailed settings for the selected "Player" object:

- Transform:** Position: X: -1.433722, Y: -2.485271, Z: 0.1386719; Rotation: X: 0, Y: 0, Z: 0.002; Scale: X: 1, Y: 1, Z: 1.
- Sprite Renderer:** Enabled, using "Player" sprite, color white, material "Sprites-Default", draw mode Simple, sorting layer Default (order 0), mask interaction None, sort point Center.
- Box Collider 2D:** Enabled, using "Edit Collider" (None), offset X: 0, Y: 0, size X: 1, Y: 1, edge radius 0.
- Rigidbody 2D:** Body type Dynamic, material "None (Physics Material 2D)", simulated checked, mass 1, linear drag 0, angular drag 0.05, gravity scale 1, collision detection Discrete, sleeping mode Start Awake, interpolate checked, constraints None.

Project View: Shows the "Assets" folder structure:

- Favorites
- Assets
 - Scenes
 - Textures
 - Player
 - Tiles
 - Packages

The "Player" texture is selected in the Project View.



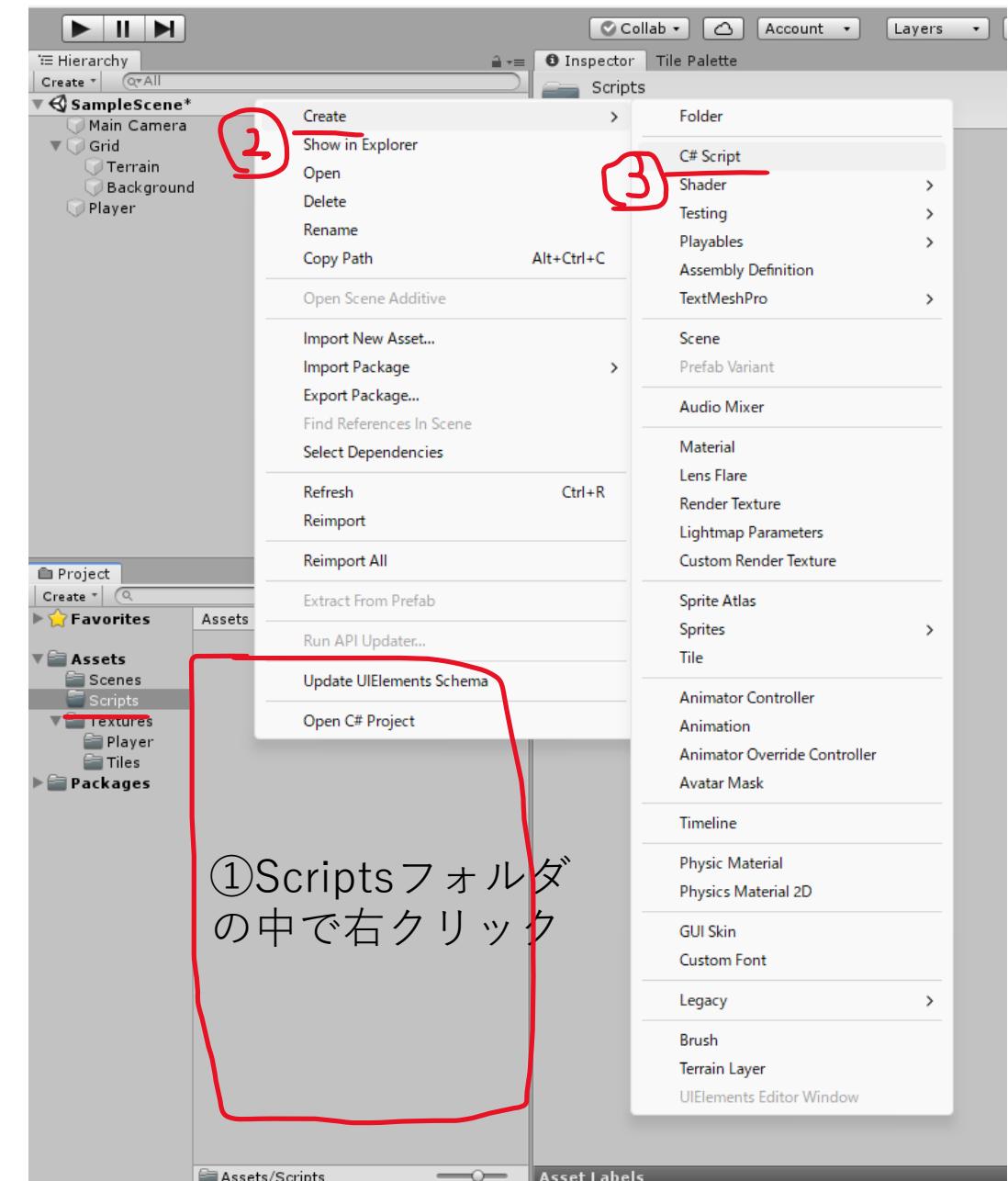
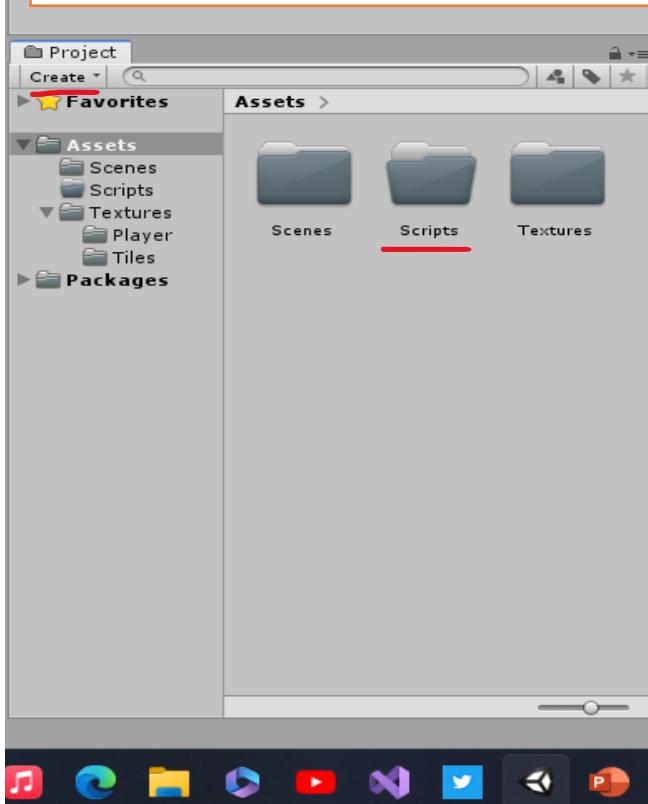
検索

19:49
2024/05/20

スクリプト作成

- ・左右移動

(Project)
Assetsを選択
Create→folder
名前：Scripts





Center Local

Scene Shaded 2D Gizmos Q>All



Hierarchy Create Q>All

SampleScene*
Main Camera
Grid
Terrain
Background
Player

Account

Layers

Layout



Tilemap

Focus On

None



Game

Display 1

16:10

+

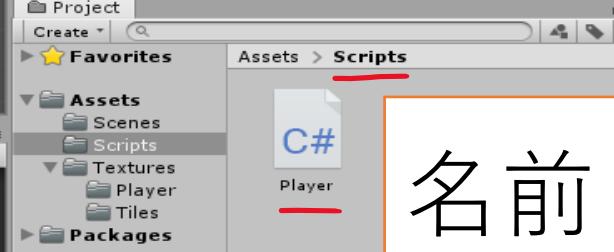
+

Scale

1x

Maximize On Play Mute Audio Stats

Gizmos



名前：Player



検索

20:30
2024/05/20

Player.cs

```
Assembly-CSharp
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {
        // Start()はこのスクリプトを適用したオブジェクトがシーン内に現れたときに呼び出される
    }

    // Update is called once per frame
    void Update()
    {
        // Update()は毎フレーム呼び出される
    }
}
```



エラー一覧... 出力

```
//Player.cs
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Player : MonoBehaviour
{
    private Rigidbody2D rb; // (a)

    void Start()
    {
        rb = GetComponent<Rigidbody2D>(); // (b)
    }

    void Update()
    {
        //Player Movement
        rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal"), rb.velocity.y); // (c)
        //この一文で、プレイヤーの左右移動を制御している
    }
}
```

スクリプトの適用

The screenshot shows the Unity Editor interface with a 2D game scene, Hierarchy panel, Project panel, and Inspector panel.

Scene View: Displays a 2D game world with a blue sky, green grass, and brown ground. A small character model is positioned on the ground. The Game View shows the same scene from a different perspective.

Hierarchy Panel: Shows the scene structure:

- SampleScene*
- Main Camera
- Grid
- Terrain
- Background
- Player

The "Player" object is circled in red.

Project Panel: Shows the project structure:

- Assets
- Scenes
- Scripts
- Textures
- Packages

The "Player" script file is highlighted and circled in red. It has a C# icon and the name "Player".

Inspector Panel: Provides detailed settings for the selected "Player" object:

- Transform:** Position X: -1.433717, Y: 0.4567741, Z: 0.1386719; Rotation X: 0, Y: 0, Z: 0; Scale X: 1, Y: 1, Z: 1.
- Sprite Renderer:** Enabled, Sprite: Player, Color: White, Material: Sprites-Default, Draw Mode: Simple, Sorting Layer: Default, Order in Layer: 0, Mask Interaction: None, Sprite Sort Point: Center.
- Box Collider 2D:** Enabled, Material: None (Physics Material 2D), Is Trigger: False, Used By Effector: False, Used By Composite: False, Auto Tiling: False, Offset: X: 0, Y: 0, Size: X: 1, Y: 1, Edge Radius: 0.
- Rigidbody 2D:** Body Type: Dynamic, Material: None (Physics Material 2D), Simulated: True, Use Auto Mass: False, Mass: 1, Linear Drag: 0, Angular Drag: 0.05, Gravity Scale: 1, Collision Detection: Discrete, Sleeping Mode: Start Awake, Interpolate: False, Constraints: None, Info: Info.
- Player (Script):** Script: Player, Player (Script) checkbox is checked.

Scene Shaded 2D Gizmos Hierarchy Create All

Player

Position X -1.433717 Y 0.4567741 Z 0.1386719
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Sprite Renderer

Sprite Player
Color X Y
Material Sprites-Default Simple
Draw Mode Default
Sorting Layer 0
Order in Layer None
Mask Interaction None
Sprite Sort Point Center

Box Collider 2D

Material None (Physics Material 2D)
Is Trigger
Used By Effector
Used By Composite
Auto Tiling
Offset X 0 Y 0
Size X 1 Y 1
Edge Radius 0

Rigidbody 2D

Body Type Dynamic
Material None (Physics Material 2D)

Player (Script)

Interpolate None
Constraints
Freeze Position
Freeze Rotation Z
Info

Project

Favorites

Assets

Scenes

Scripts

Textures

Player Tiles

Packages

C# Player

Freeze Rotation: Z



移動速度変更

```
//Player.cs
using UnityEngine;

public class Player : MonoBehaviour
{
    public float MoveSpeed = 5f; // (d)

    private Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        //Player Movement
        rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal") * MoveSpeed, rb.velocity.y); // (e)
    }
}
```

Unity Editor Screenshot showing a 2D game scene setup.

Scene View: Displays a 2D tile-based environment with a character sprite and a small cloud. A grid is overlaid on the scene.

Hierarchy: Shows the scene structure:

- SampleScene*
- Main Camera
- Grid
- Terrain
- Background
- Player

Inspector: Details for the selected Player object.

- Sprite Renderer:**
 - Sprite: Player
 - Color: X 0 Y 0
 - Material: Sprites-Default
 - Draw Mode: Simple
 - Sorting Layer: Default
 - Order in Layer: 0
 - Mask Interaction: None
 - Sprite Sort Point: Center
- Box Collider 2D:**
 - Material: None (Physics Material 2D)
 - Is Trigger: Unchecked
 - Used By Effector: Unchecked
 - Used By Composite: Unchecked
 - Auto Tiling: Unchecked
 - Offset: X 0 Y 0
 - Size: X 1 Y 1
 - Edge Radius: 0
- Rigidbody 2D:**
 - Body Type: Dynamic
 - Material: None (Physics Material 2D)
 - Simulated: Checked
 - Use Auto Mass: Unchecked
 - Mass: 1
 - Linear Drag: 0
 - Angular Drag: 0.05
 - Gravity Scale: 1
 - Collision Detection: Discrete
 - Sleeping Mode: Start Awake
 - Interpolate: None
 - Constraints:
 - Freeze Position: X Y Unchecked
 - Freeze Rotation: Z Checked
- Player (Script):**
 - Script: Player
 - Move Speed: 5

Project View: Shows the project structure with Assets, Scenes, Scripts, Textures, and Packages. A C# script named Player is selected.

Game View: Shows the character sprite in the game world.

Text Overlay: "ここでもスピード調整ができる" (You can also adjust the speed here) is displayed in Japanese text at the bottom right of the Inspector panel.

ジャンプ実装

```
//Player.cs
using UnityEngine;

public class Player : MonoBehaviour
{
    public float MoveSpeed = 3f;
    public float JumpForce = 15f; // (f)

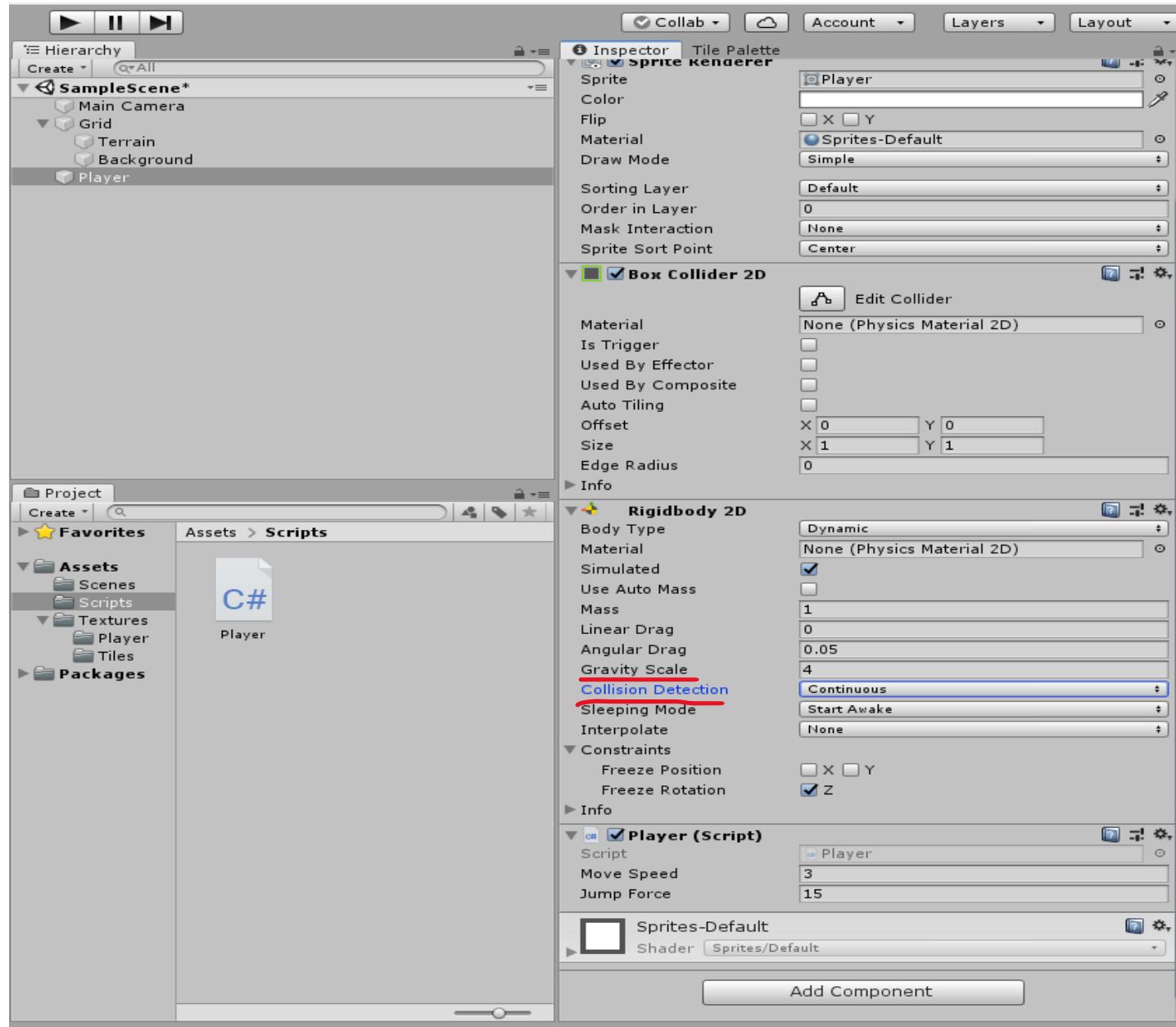
    private Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        //Player Movement
        rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal") * MoveSpeed, rb.velocity.y);

        if (Input.GetButtonDown("Jump")) // (g)
        {
            rb.velocity = new Vector2(rb.velocity.x, JumpForce);
        }
    }
}
```

値を設定

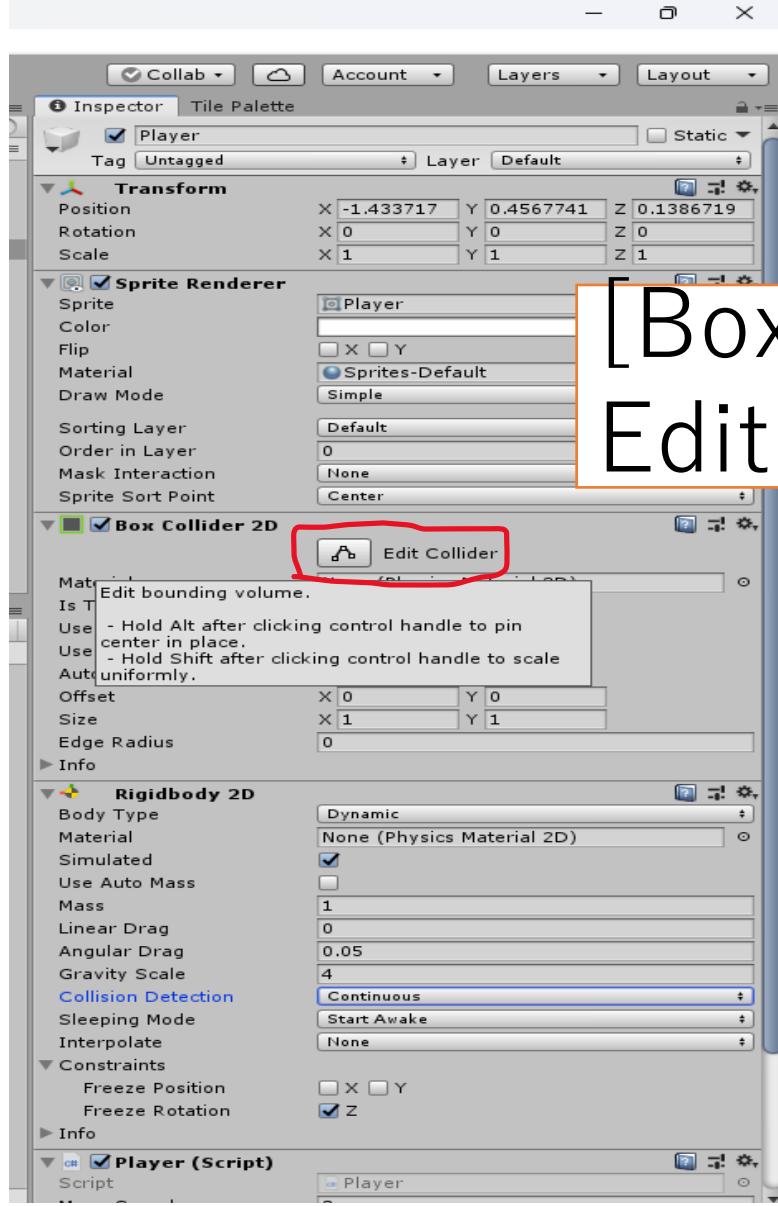


[Rigidbody 2D]
Gravity Scale → 4
: 重力スケール

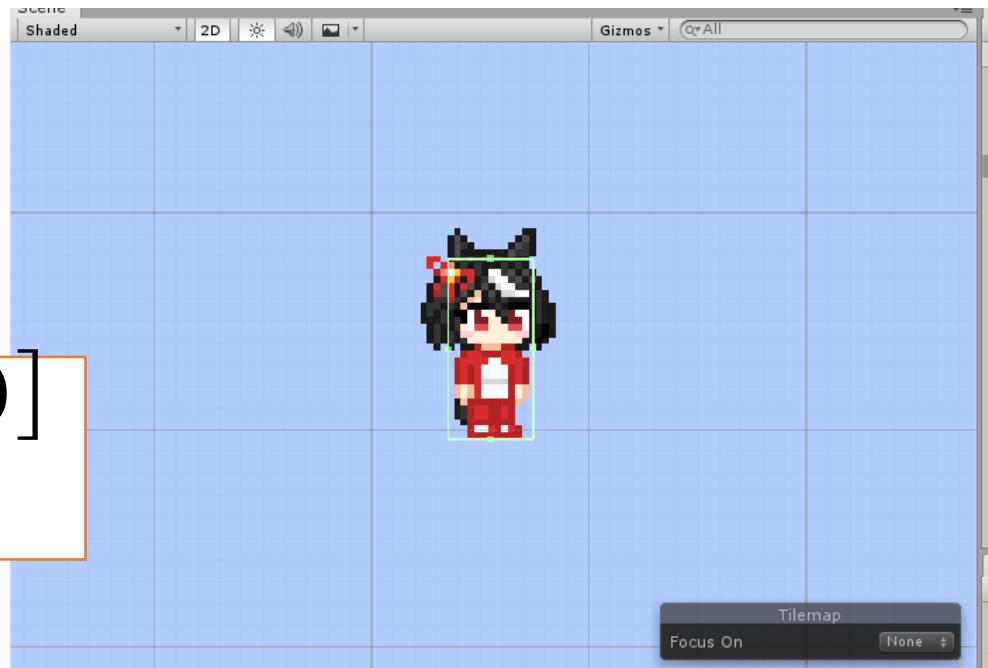
collision Detection → Continuous
: 衝突判定を連続的
→ 着地したときにめり込む
のを防ぐ

[Player (Script)]
Move Speed → 3
Jump Force → 15

キャラクターの当たり判定を変更



[Box Collider 2D]
Edit Collider



キャラクターの大きさに合うよう
に緑の枠の大きさを変更する

向き変更

//前のコード省略

```
void Update()
{
    //Player Movement
    rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal") * MoveSpeed, rb.velocity.y);

    if (Input.GetButtonDown("Jump"))
    {
        rb.velocity = new Vector2(rb.velocity.x, JumpForce);
    }
    //Sprite Flip //追加
    if (rb.velocity.x > 0)//(h)
    {
        GetComponent<SpriteRenderer>().flipX = false;
    }
    else if (rb.velocity.x < 0)
    {
        GetComponent<SpriteRenderer>().flipX = true;
    }
}
```

着地判定

- ・地面に足がついている状態でのみジャンプ

Scene Shaded 2D Gizmos Hierarchy Create All SampleScene* Main Camera Grid Terrain Background Player

Terrain Tag Untagged Layer Default

Transform Position X 0 Y 0 Z 0 Rotation X 0 Y 0 Z 0 Scale X 1 Y 1 Z 1

Tilemap Animation Frame Rate 1 Color X 0.5 Y 0.5 Tile Anchor XY Orientation Position X 0 Y 0 Z 0 Rotation X 0 Y 0 Z 0 Scale X 1 Y 1 Z 1

Add Layer...

Tilemap Renderer Material Sprites-Default Sort Order Bottom Left Mode Chunk Detect Chunk Culling Bound Auto Chunk Culling Bounds X 0 Y 0 Z 0 Sorting Layer Default Order in Layer 0 Mask Interaction None

Rigidbody 2D Body Type Static Material None (Physics Material 2D) Simulated Info

Composite Collider 2D Material None (Physics Material 2D) Is Trigger Used By Effector Offset X 0 Y 0 Geometry Type Outlines Generation Type Synchronous Vertex Distance 0.0005 Edge Radius 0 Info

Tilemap Collider 2D Used By Composite Offset X 0 Y 0 Info

Sprites-Default Shader Sprites/Default

Terrain

Focus On None

Game Display 1 16:10 Scale 1x Maximize On Play Mute Audio Stats Gizmos

Favorites Assets Scenes Scripts Textures Player Tiles Packages

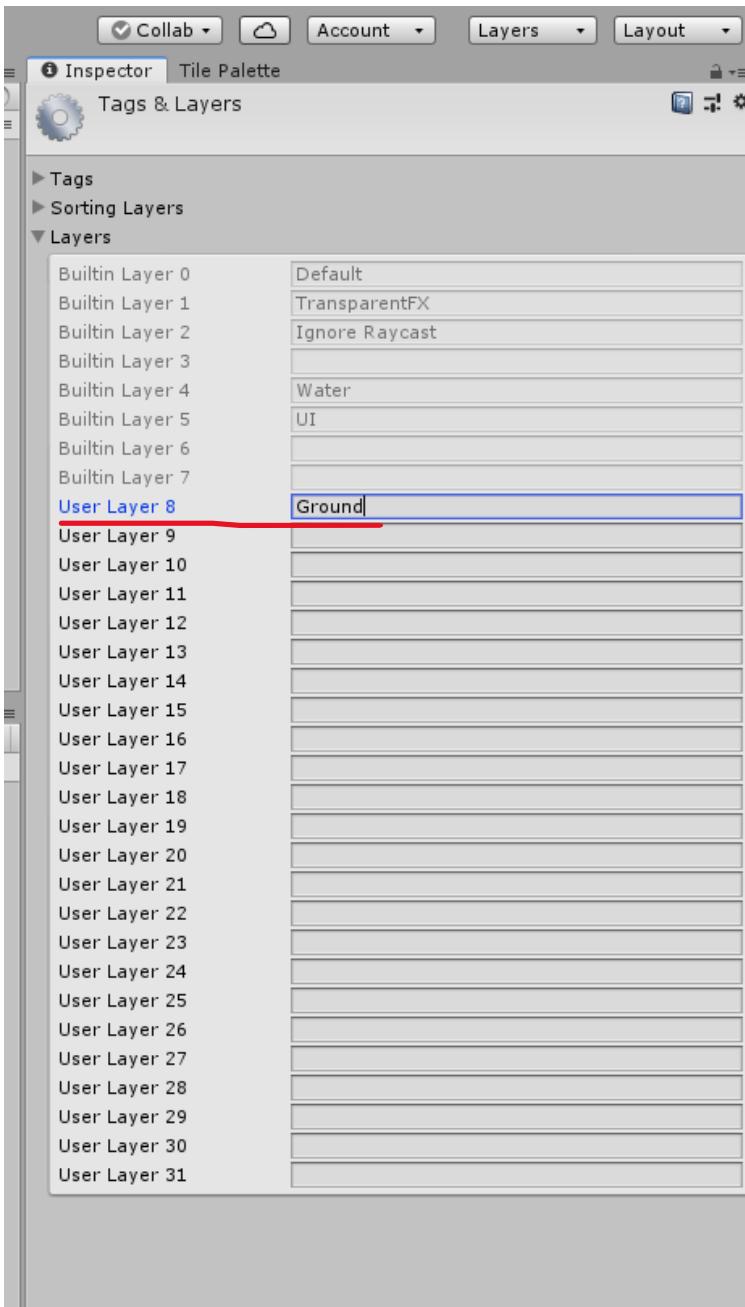
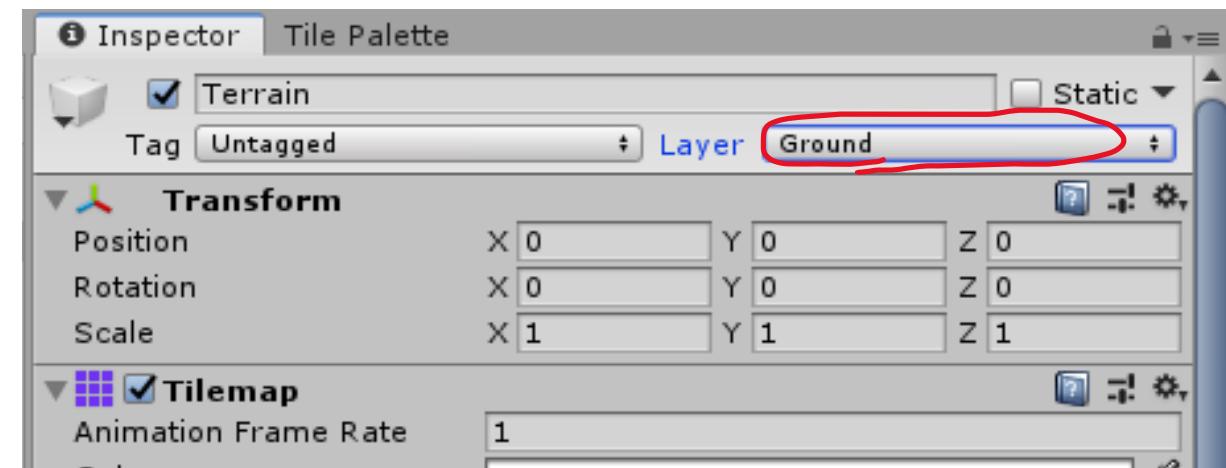
C# Player

Terrainを選択
Layer→Add Layer...

Groundという名前のレイヤーを追加する User Layer → Ground

※ 「ソートレイヤー」は表示上の設定を使うが、
こちらの「レイヤー」は主に衝突判定の設定で使う。

設定画面でのレイヤーの上下関係は関係はない。



```
//Player.cs
using UnityEngine;

public class Player : MonoBehaviour
{
    public float MoveSpeed = 3f;
    public float JumpForce = 15f;
    public LayerMask GroundLayer; // (i)

    private Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

}
```

```
void Update()
{
    //Player Movement
    rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal") * MoveSpeed, rb.velocity.y);

    if (Input.GetButtonDown("Jump") && isGrounded())//(j)
    {
        rb.velocity = new Vector2(rb.velocity.x, JumpForce);
    }
    //Sprite Flip
    //省略
}
private bool isGrounded()//(k)
{
    BoxCollider2D c = GetComponent<BoxCollider2D>();
    return Physics2D.BoxCast(c.bounds.center, c.bounds.size, 0f, Vector2.down, .1f, GroundLayer);
}
```

The Unity Editor interface is shown, featuring the Hierarchy, Inspector, and Scene windows. A central text box contains Japanese text. The Inspector window shows components like Box Collider 2D and Rigidbody 2D, with the Ground Layer highlighted.

Playerを選択
[Player(Script)]
Ground Layer→Ground

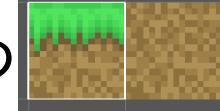
Inspector Window Components:

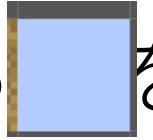
- Box Collider 2D:
 - Material: None (Physics Material 2D)
 - Is Trigger:
 - Used By Effector:
 - Used By Composite:
 - Auto Tiling:
 - Offset: X: -0.019261, Y: -0.0851547
 - Size: X: 0.3937776, Y: 0.8296905
 - Edge Radius: 0
- Rigidbody 2D:
 - Body Type: Dynamic
 - Material: None (Physics Material 2D)
 - Simulated:
 - Use Auto Mass:
 - Mass: 1
 - Linear Drag: 0
 - Angular Drag: 0.05
 - Gravity Scale: 4
 - Collision Detection: Continuous
 - Sleeping Mode:
 - Interpolate:
 - Constraints:
 - Freeze Position:
 - Freeze Rotation:
- Player (Script):
 - Move Speed:
 - Jump Force:
 - Ground Layer: (highlighted with red underline)

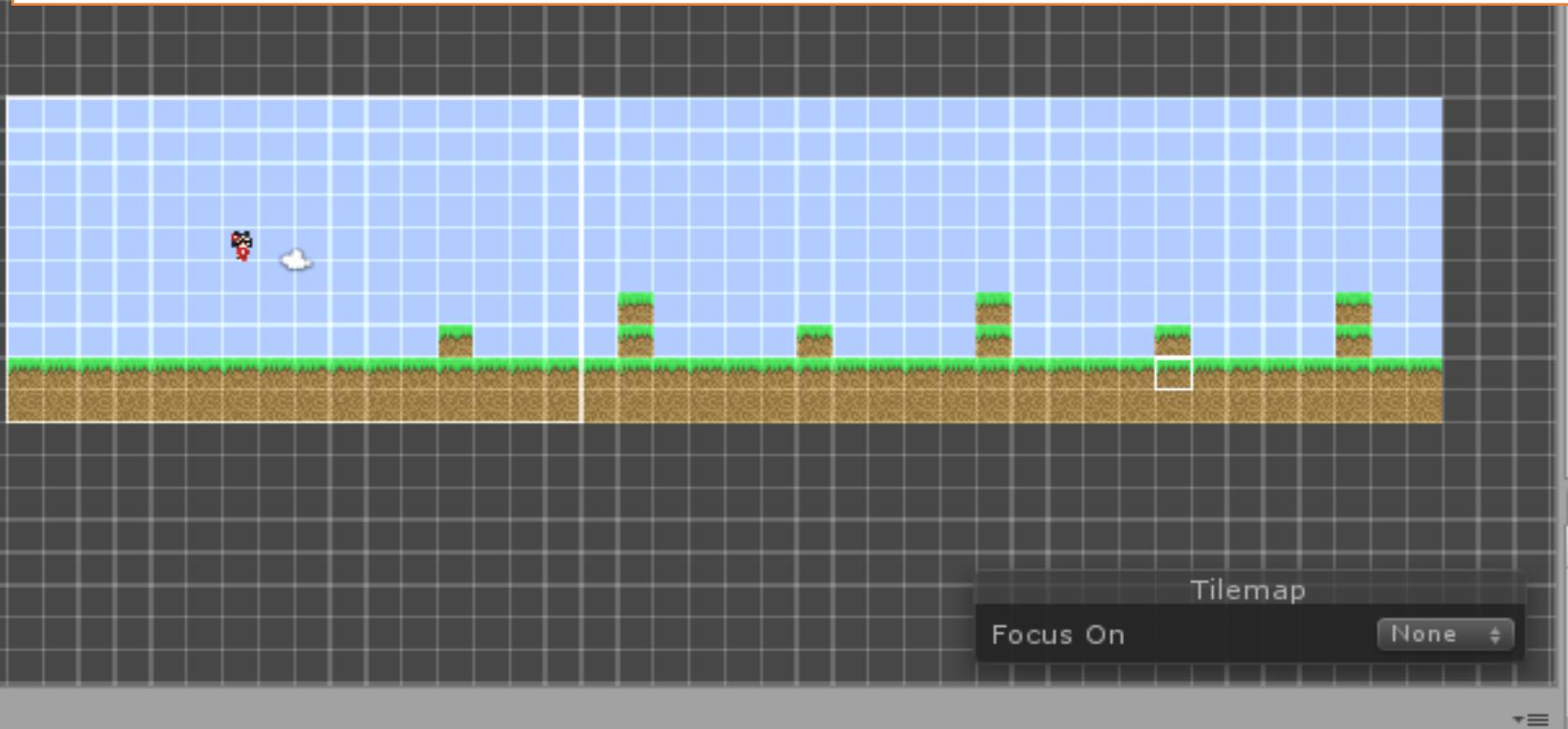


マップ拡張・カメラ移動

マップ拡張

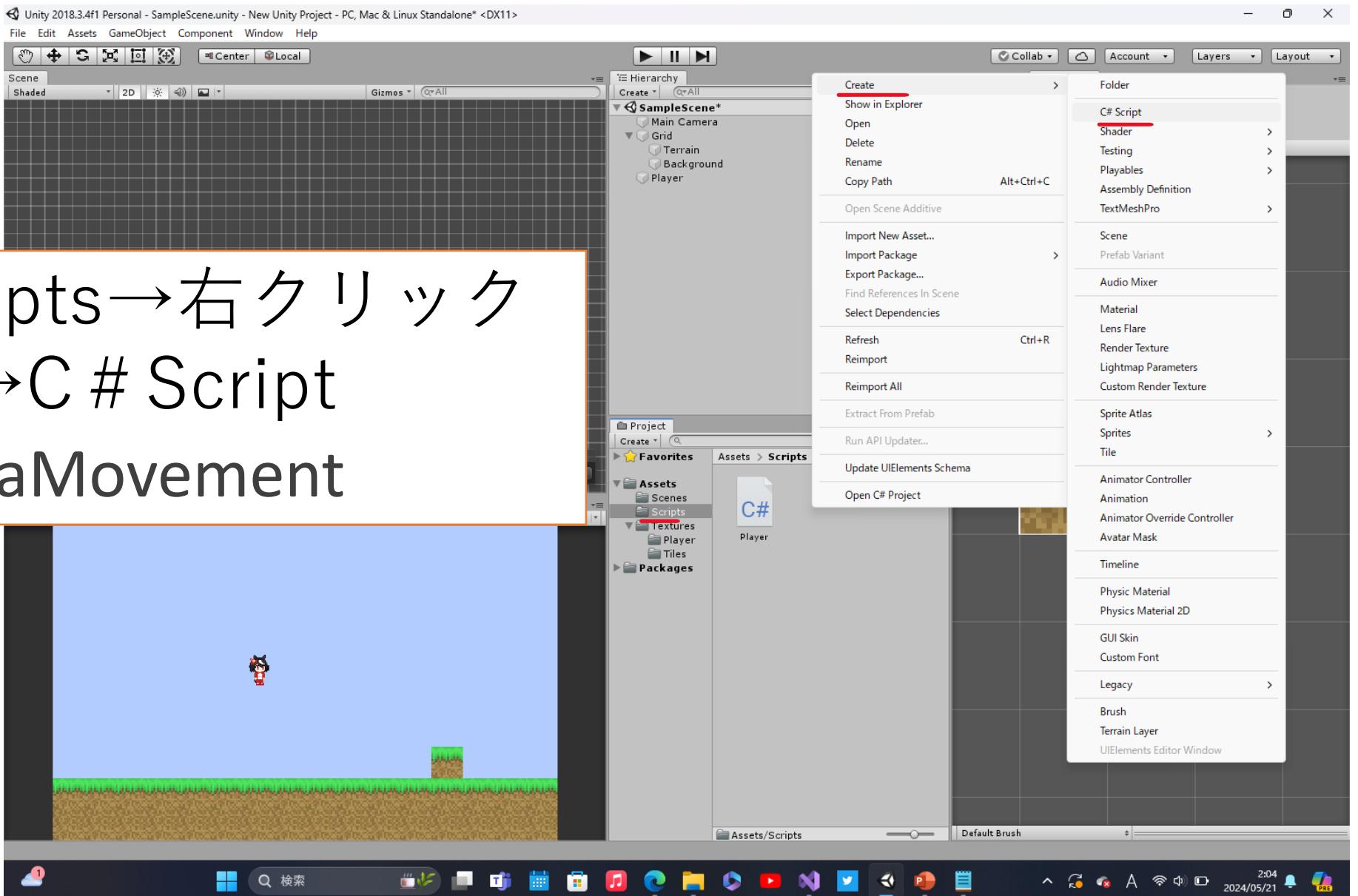
※ • Terrainを選択してから  を挿入

• Backgroundを選択してから  を挿入



カメラ移動：カメラの移動を制御するスクリプトを作成

Assets→scripts→右クリック
→Create→C# Script
名前：CameraMovement



```
//CameraMovement.cs
using UnityEngine; //()

public class CameraMovement : MonoBehaviour
{
    public GameObject Target;
    public GameObject LeftEdge;
    public GameObject RightEdge;

    void Start()
    {

    }
}
```

```
void Update()
{
    this.transform.position = new Vector3(Target.transform.position.x,
this.transform.position.y, this.transform.position.z);

    if (this.transform.position.x <= LeftEdge.transform.position.x)
    {
        this.transform.position = new Vector3(LeftEdge.transform.position.x,
this.transform.position.y, this.transform.position.z);
    }
    else if (this.transform.position.x >= RightEdge.transform.position.x)
    {
        this.transform.position = new Vector3(RightEdge.transform.position.x,
this.transform.position.y, this.transform.position.z);
    }
}
```

```
//CameraMovement.cs //すべてのコード
using UnityEngine;

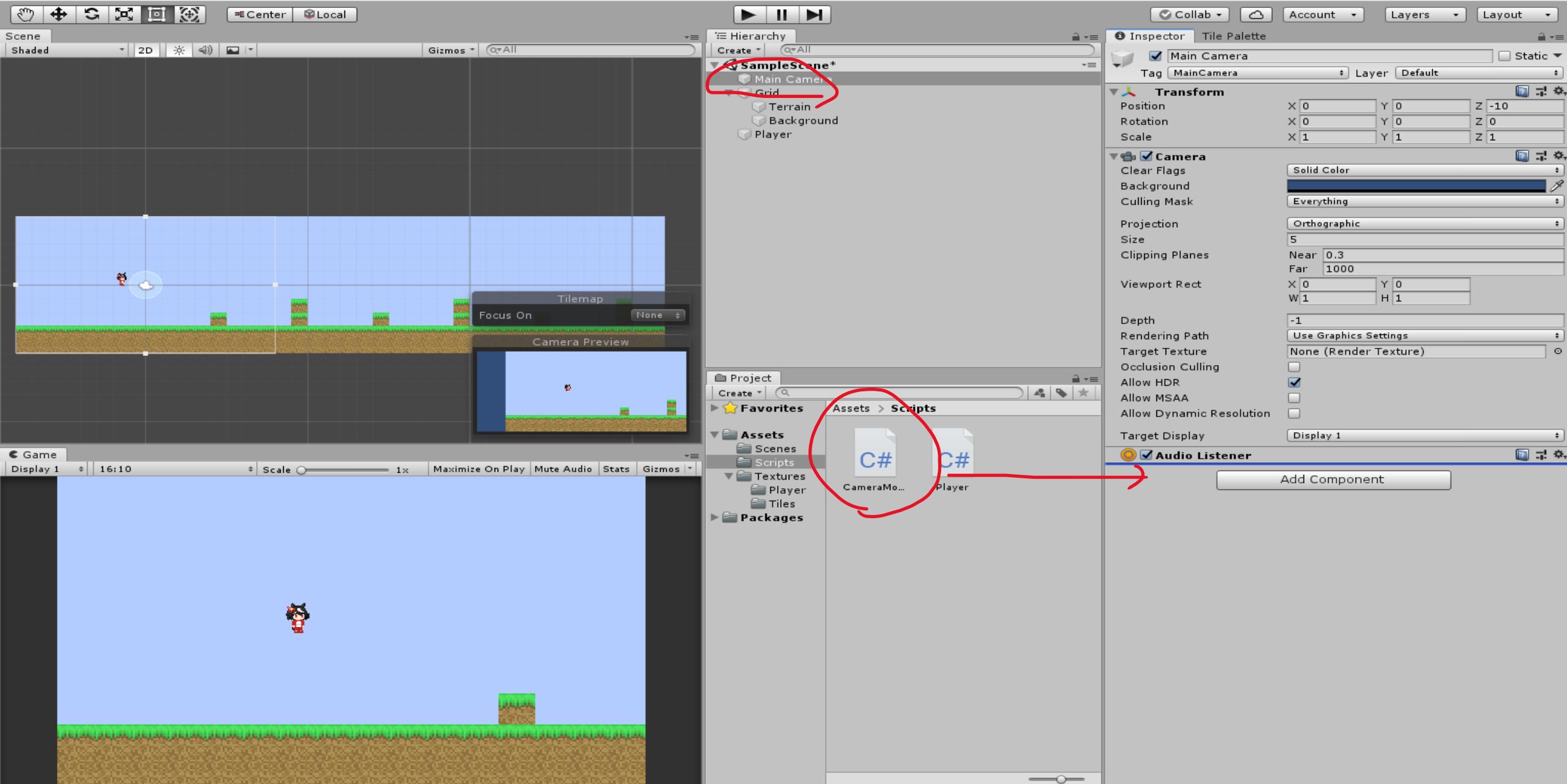
public class CameraMovement : MonoBehaviour
{
    public GameObject Target;
    public GameObject LeftEdge;
    public GameObject RightEdge;

    void Start()
    {

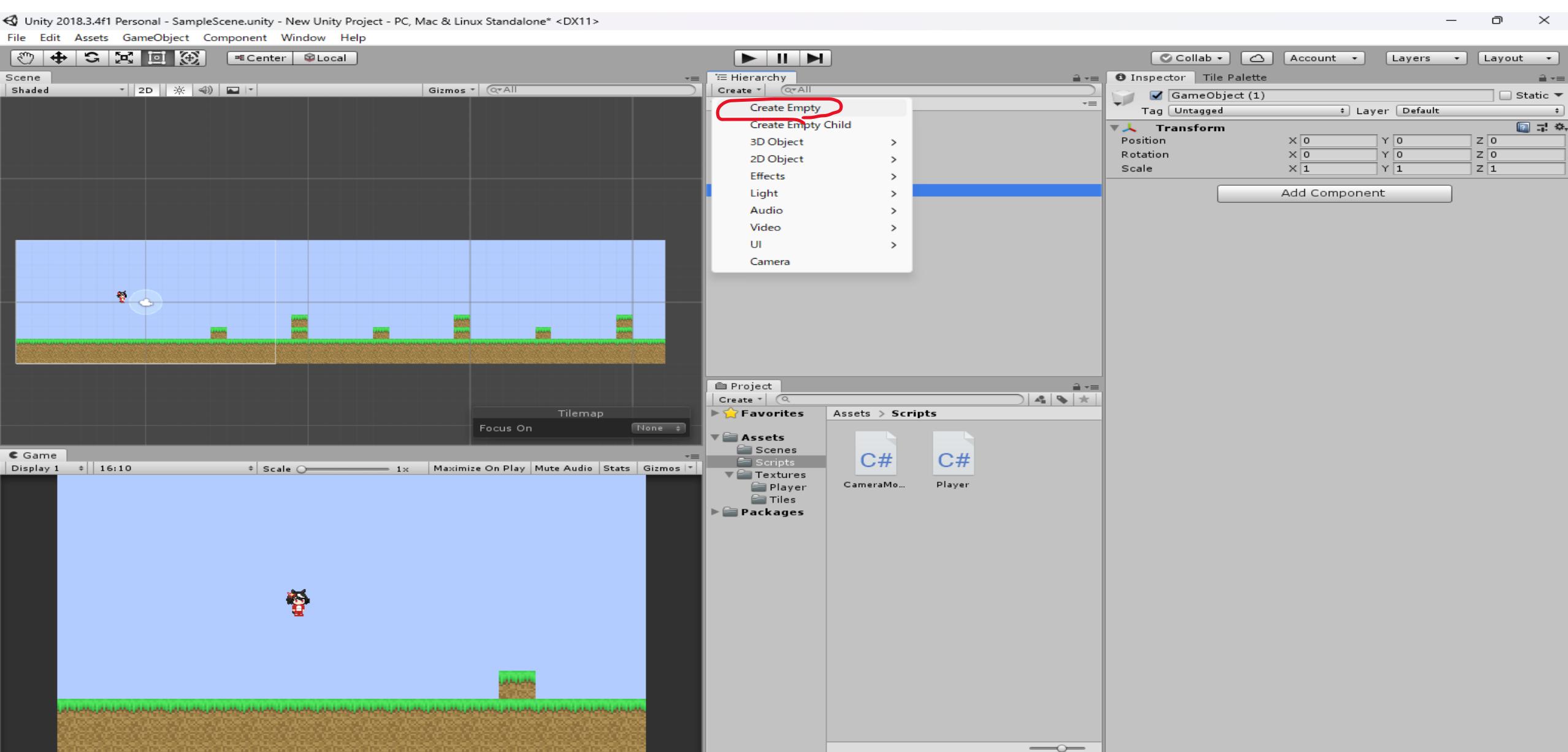
    }

    void Update()
    {
        this.transform.position = new Vector3(Target.transform.position.x, this.transform.position.y, this.transform.position.z);

        if (this.transform.position.x <= LeftEdge.transform.position.x)
        {
            this.transform.position = new Vector3(LeftEdge.transform.position.x, this.transform.position.y,
this.transform.position.z);
        }
        else if (this.transform.position.x >= RightEdge.transform.position.x)
        {
            this.transform.position = new Vector3(RightEdge.transform.position.x, this.transform.position.y,
this.transform.position.z);
        }
    }
}
```

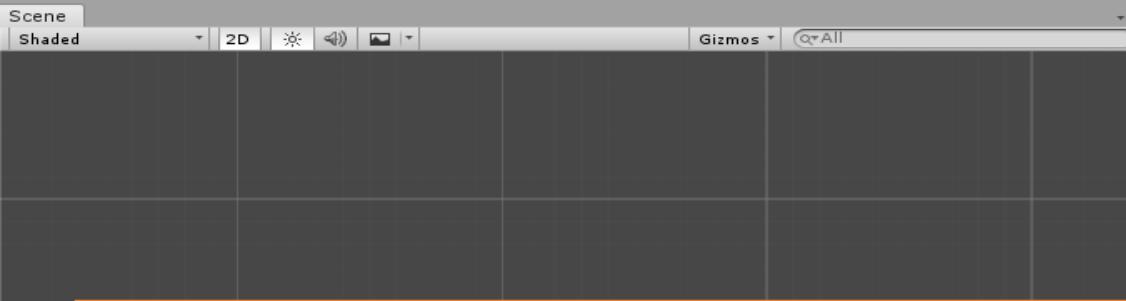


端検知用オブジェクト

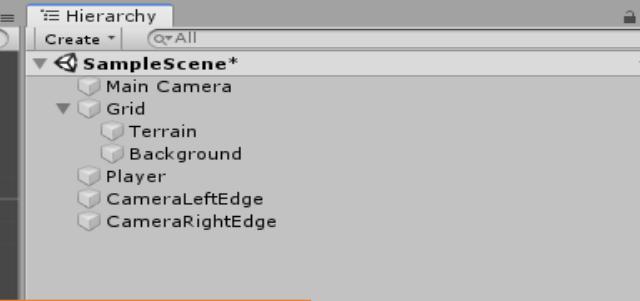
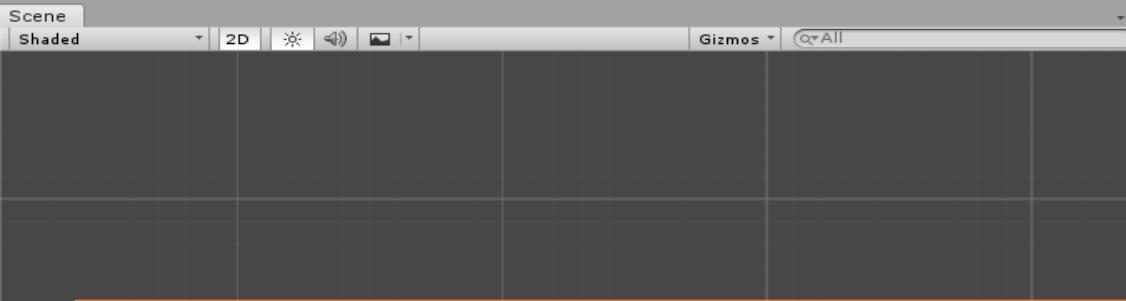




Center Local



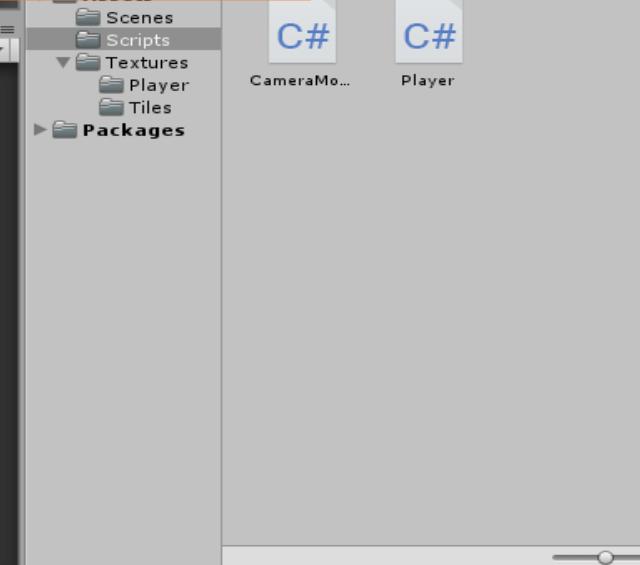
Collab Account Layers Layout



Inspector Tile Palette

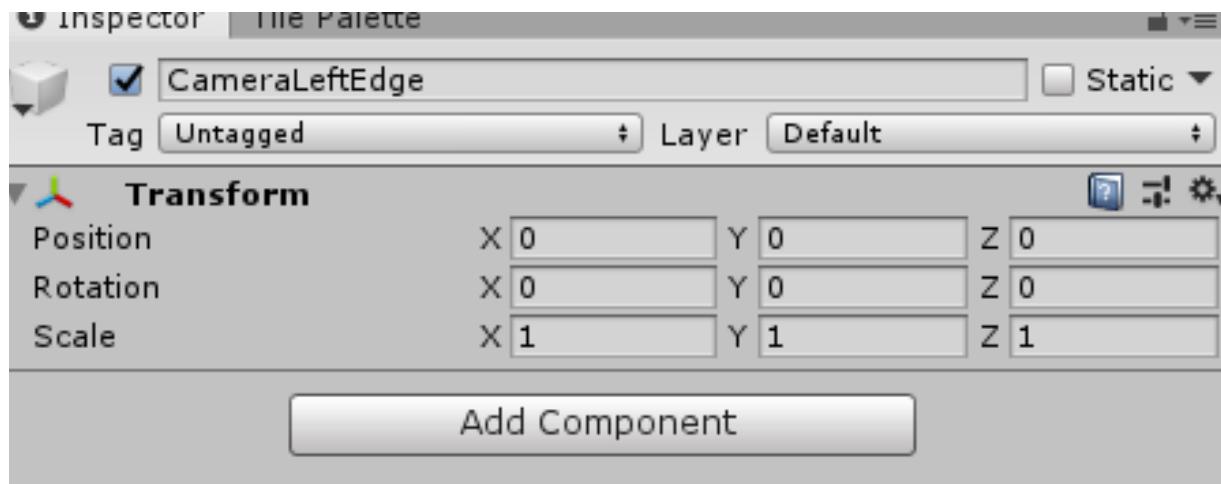
名前

- CameraLeftEdge
- CameraRightEdge

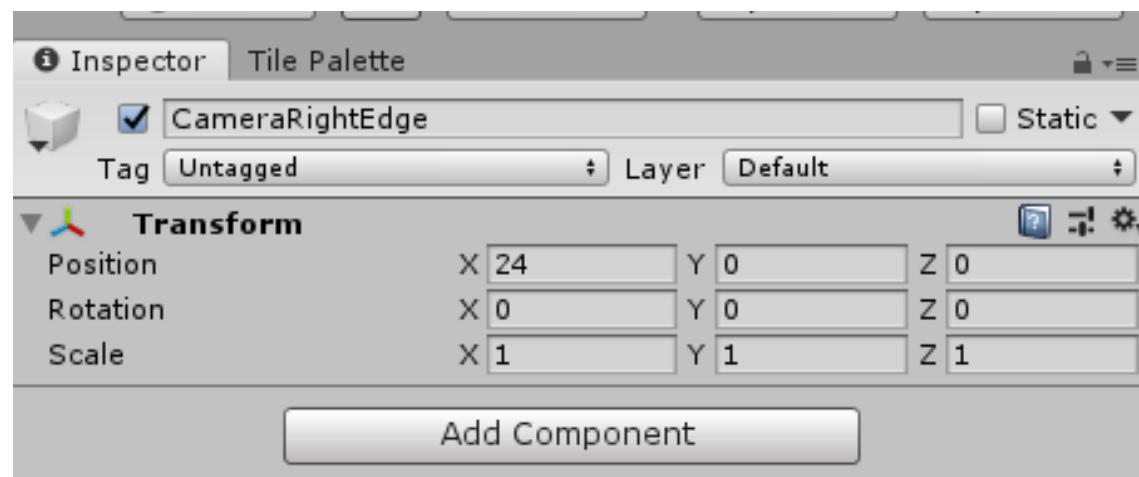


検索

2:18
2024/05/21

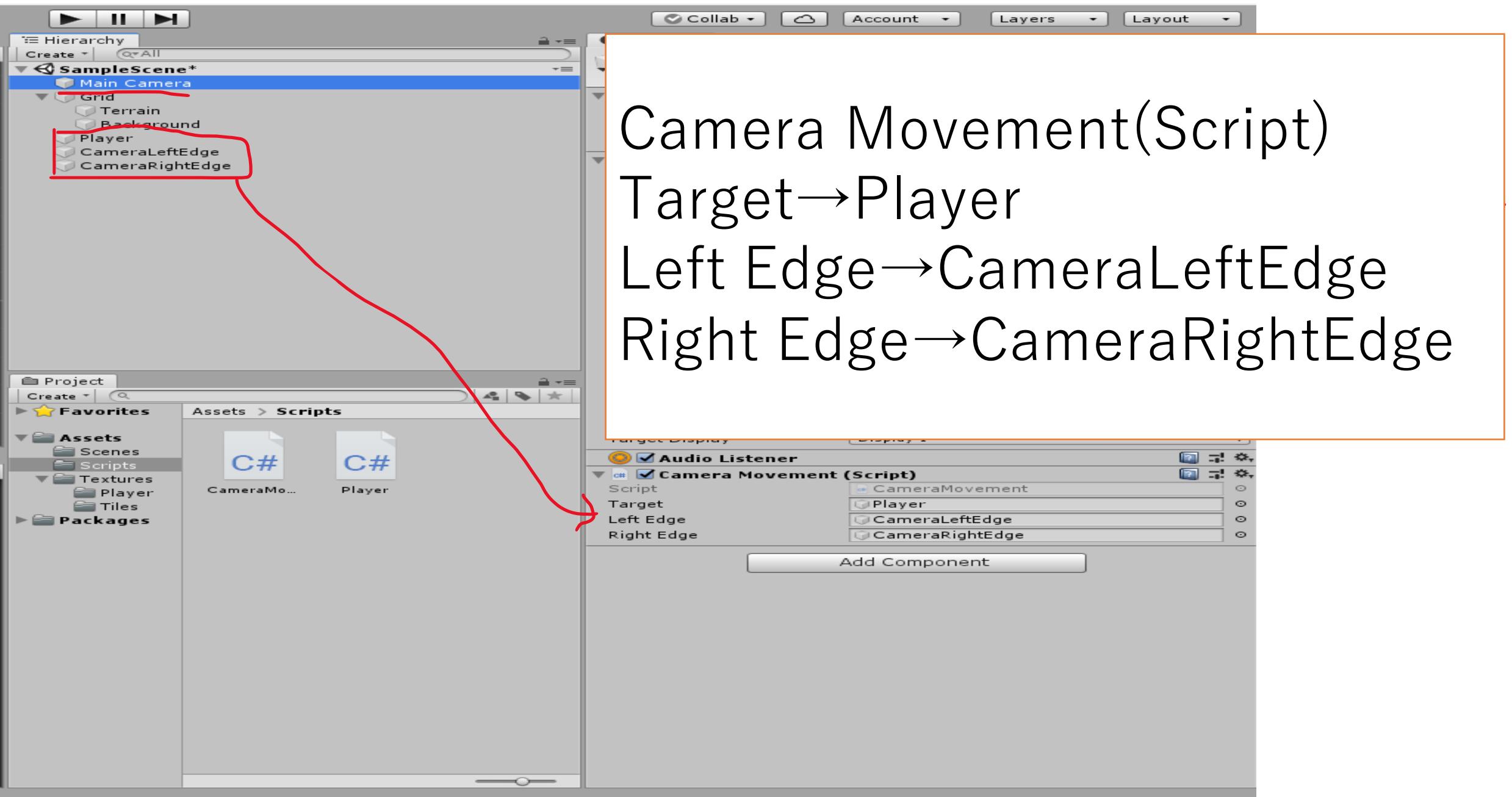


- CameraLeftEdge



- CameraRightEdge

右端から 8 マスのところ



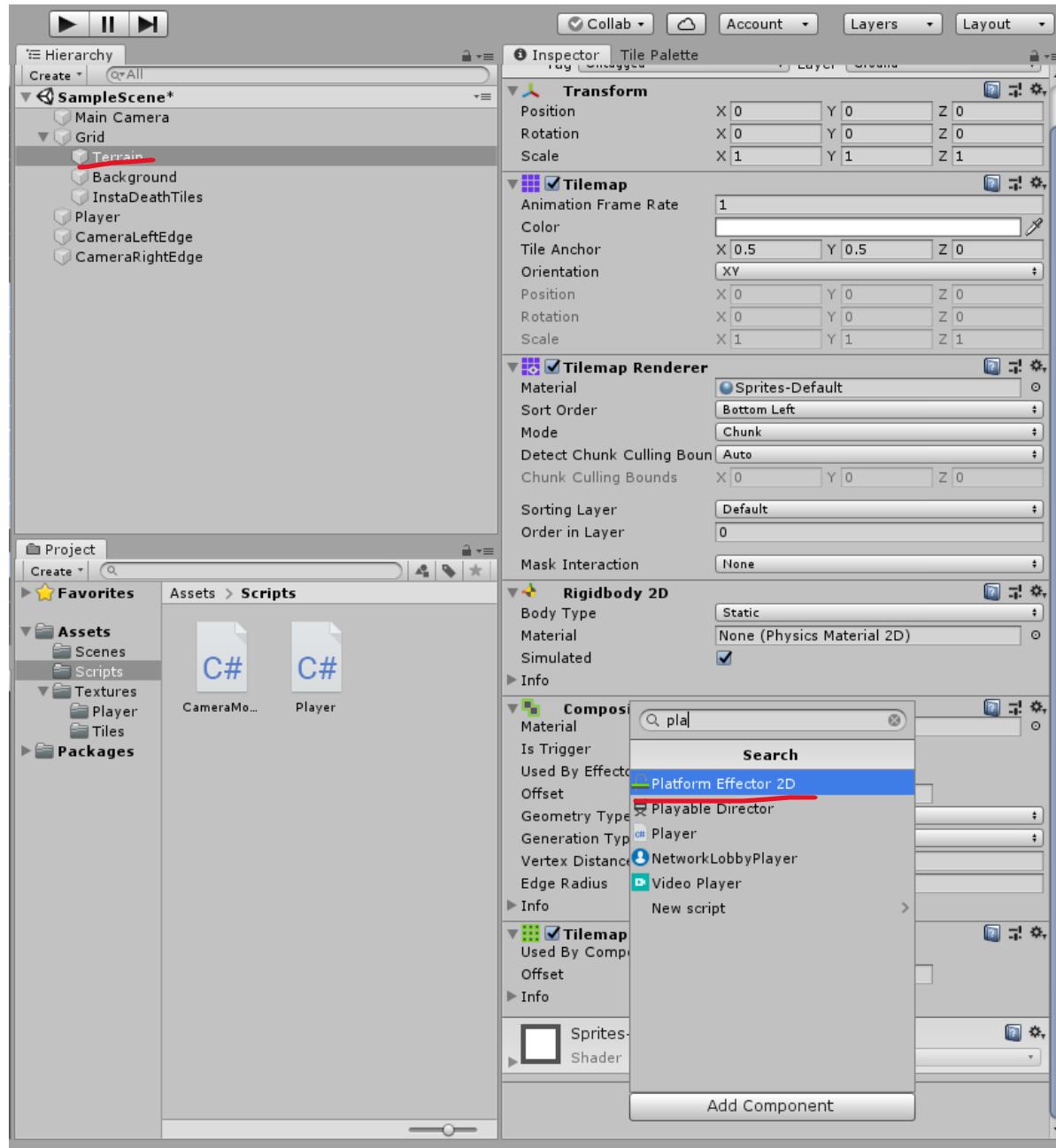
Camera Movement(Script)

Target→Player

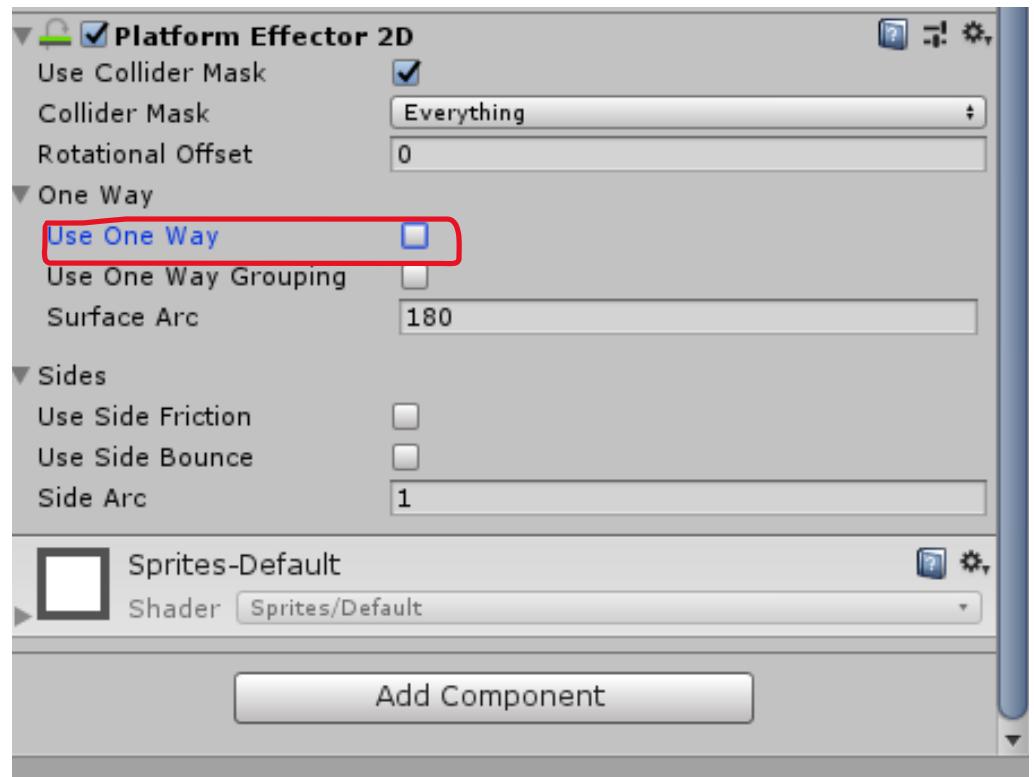
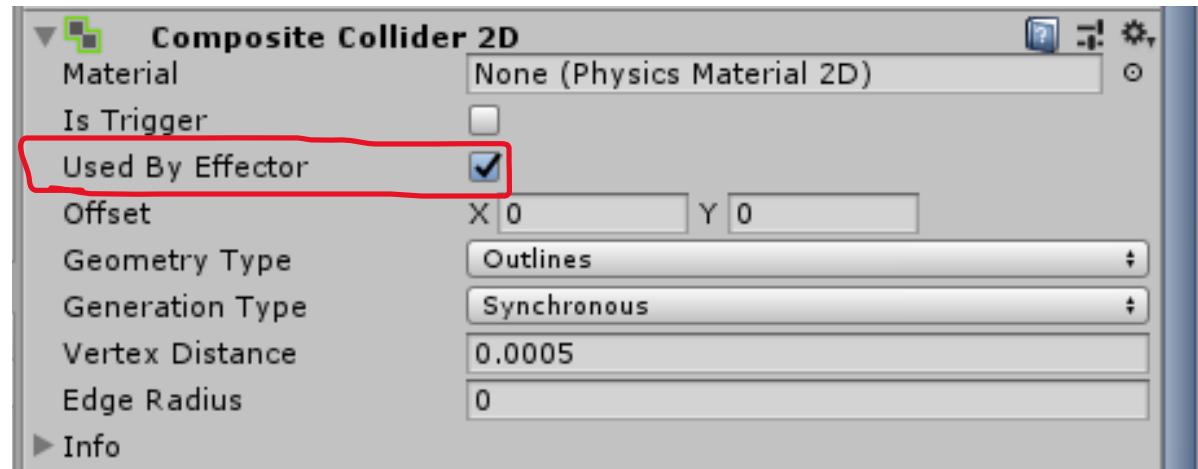
Left Edge→CameraLeftEdge

Right Edge→CameraRightEdge

ジャンプ修正【移動とジャンプを同時に使うと壁に接触する】



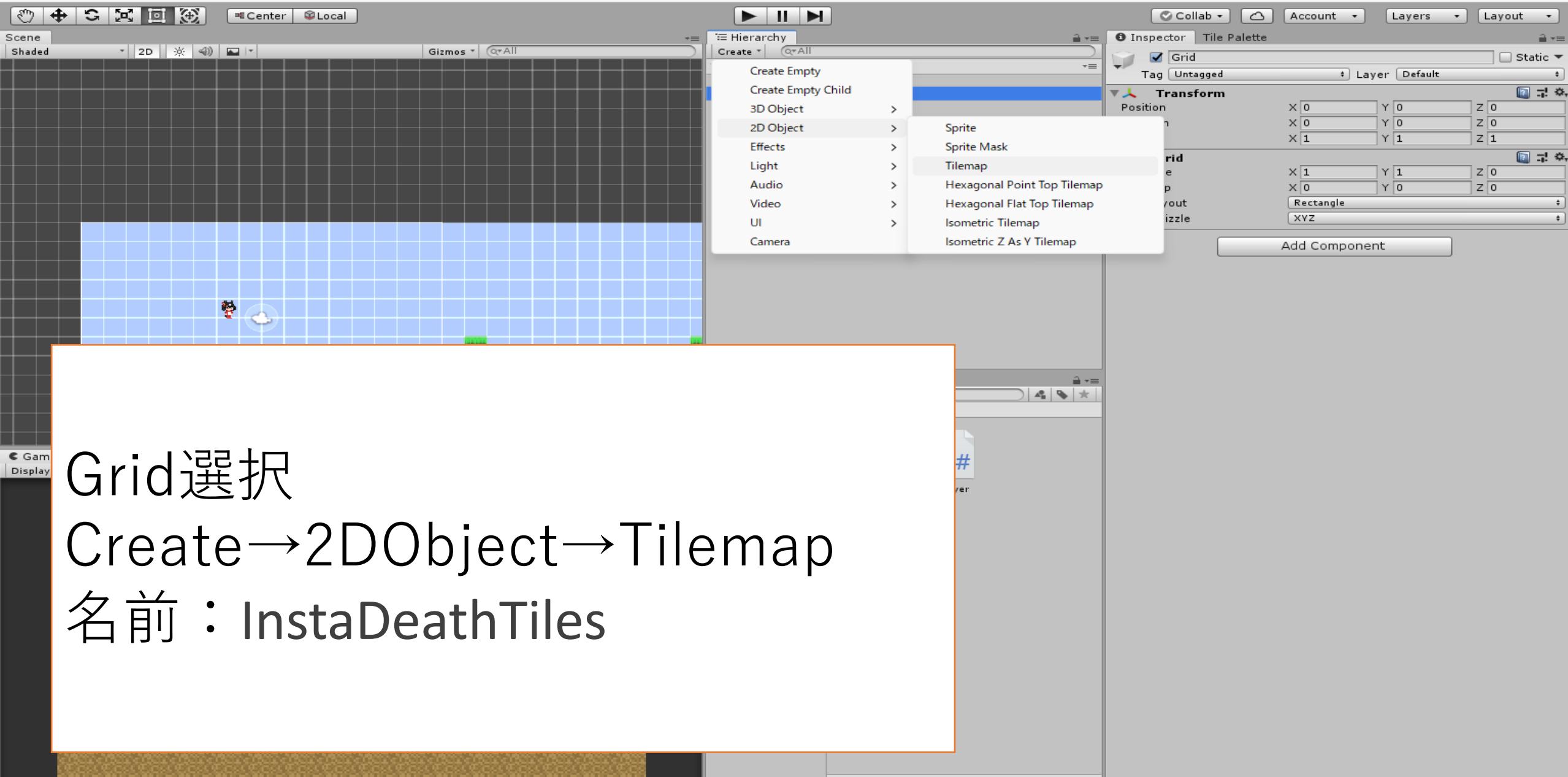
Terrain選択→Add Component
• Platform Effector 2D追加



【 Composite Collider 2D 】
Used By Effector

【 Platform Effector 2D 】
Use One Way

デス判定





Center Local

Gizmos

All



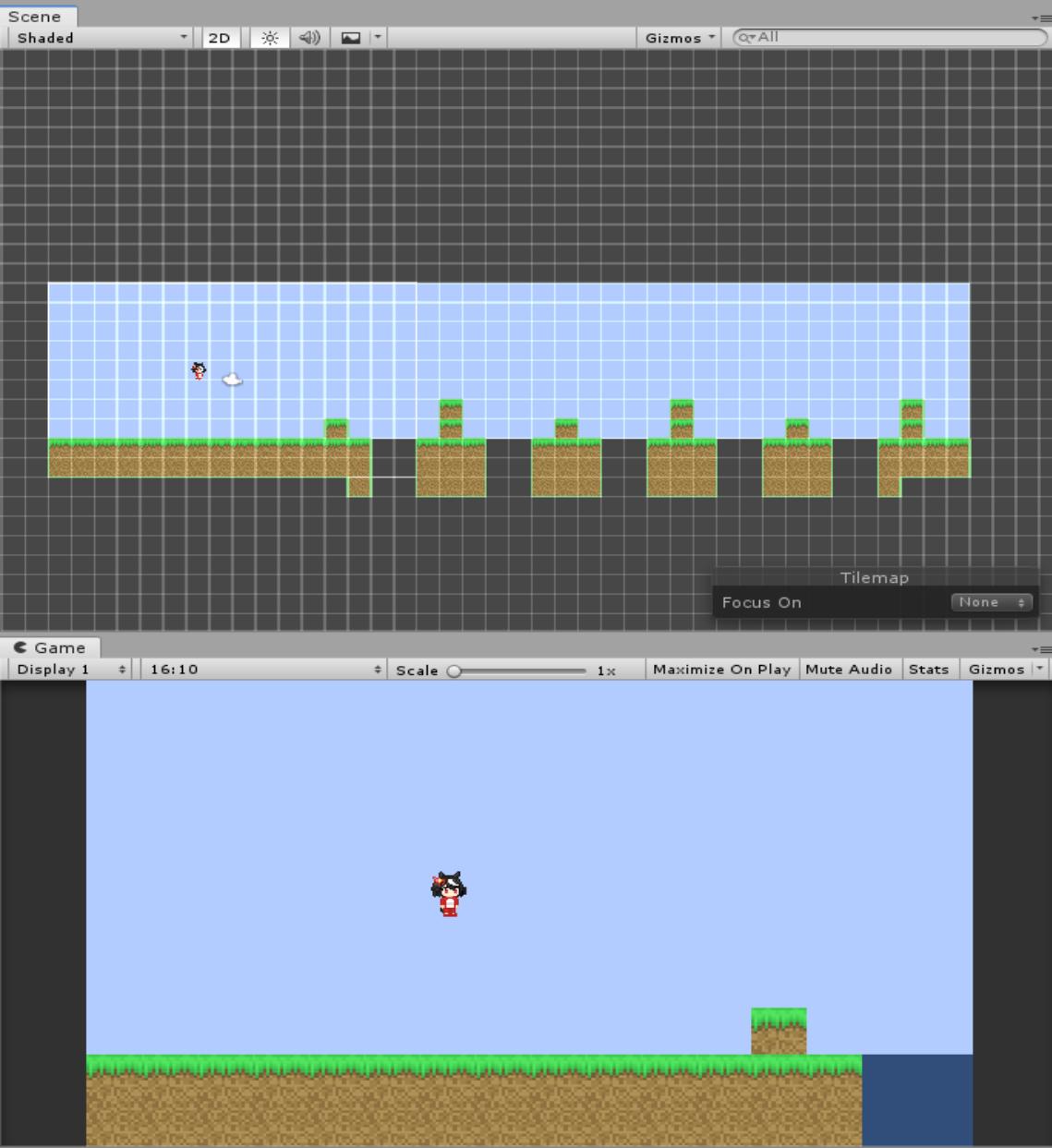
Collab



Account

Layers

Layout



Terrain選択

Project

Create



Favorites



Assets



Scenes



Scripts



Textures



Player



Tiles



Packages

Assets



Scenes



Scripts



Textures



Player



Tiles



Packages



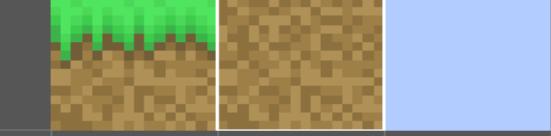
C#



C#

CameraMo...

Player



Default Brush



検索

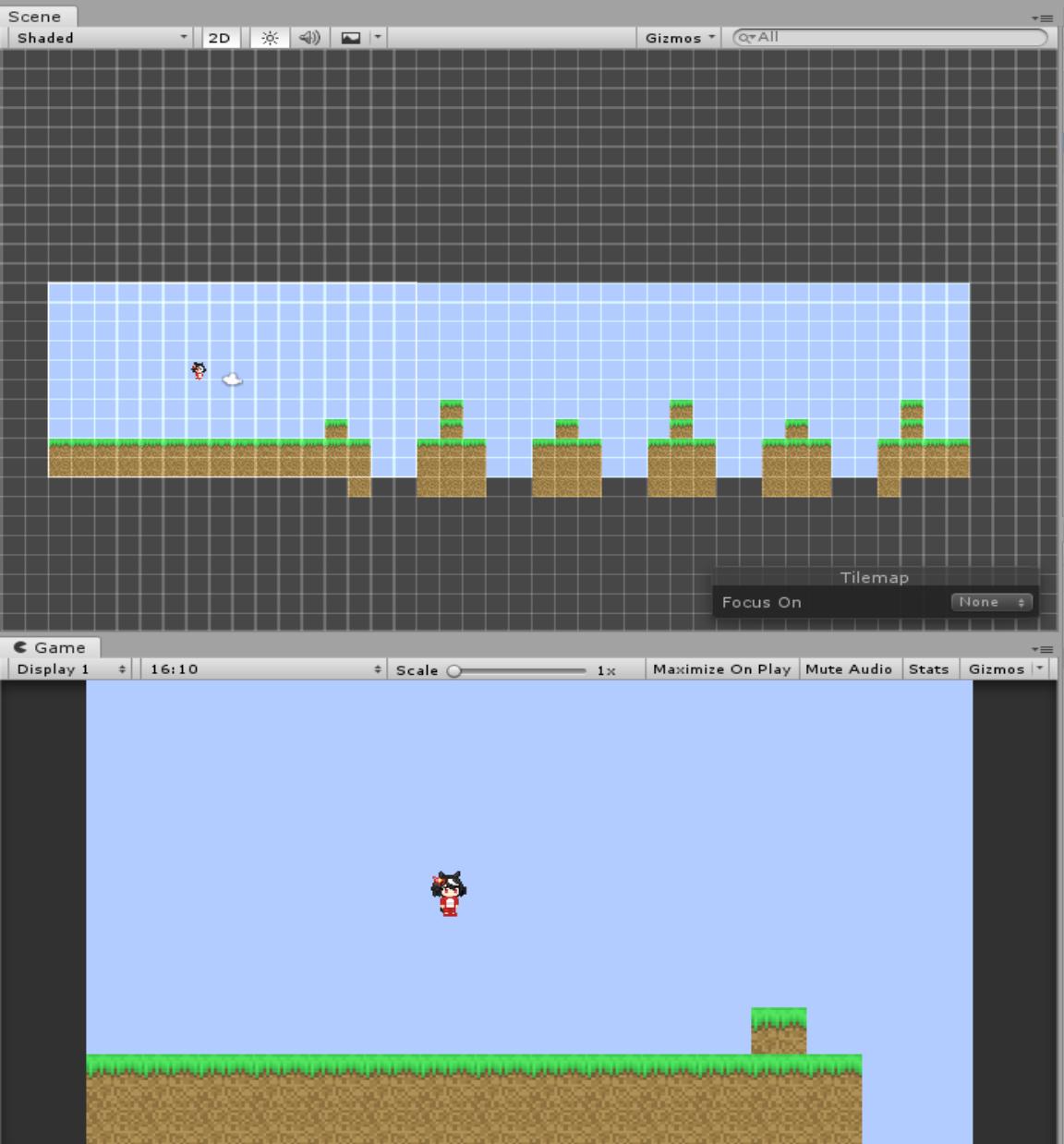
8:30
2024/05/21



Center Local

Gizmos

All



Collab Account Layers Layout

Hierarchy

Create

All

SampleScene*

- Main Camera
- Grid
- Terrain
- Background**
- InstaDeathTiles
- Player
- CameraLeftEdge
- CameraRightEdge

Inspector

Tile Palette

Background選択

Project

Create

Favorites

Assets

Scenes

Scripts

Textures

Player

Tiles

Packages

Assets > Scripts

C#

C#

CameraMo...

Player

Default Brush



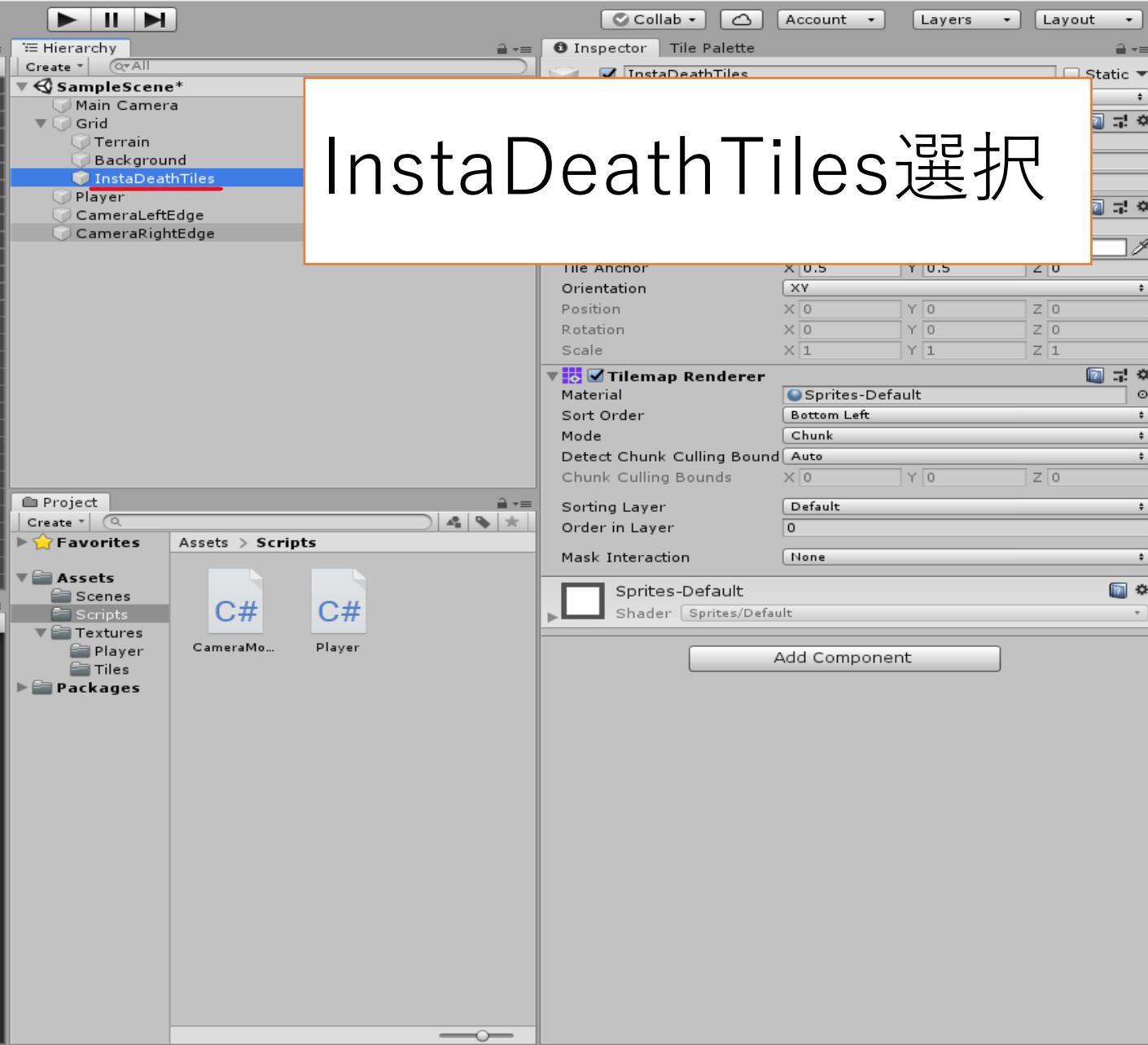
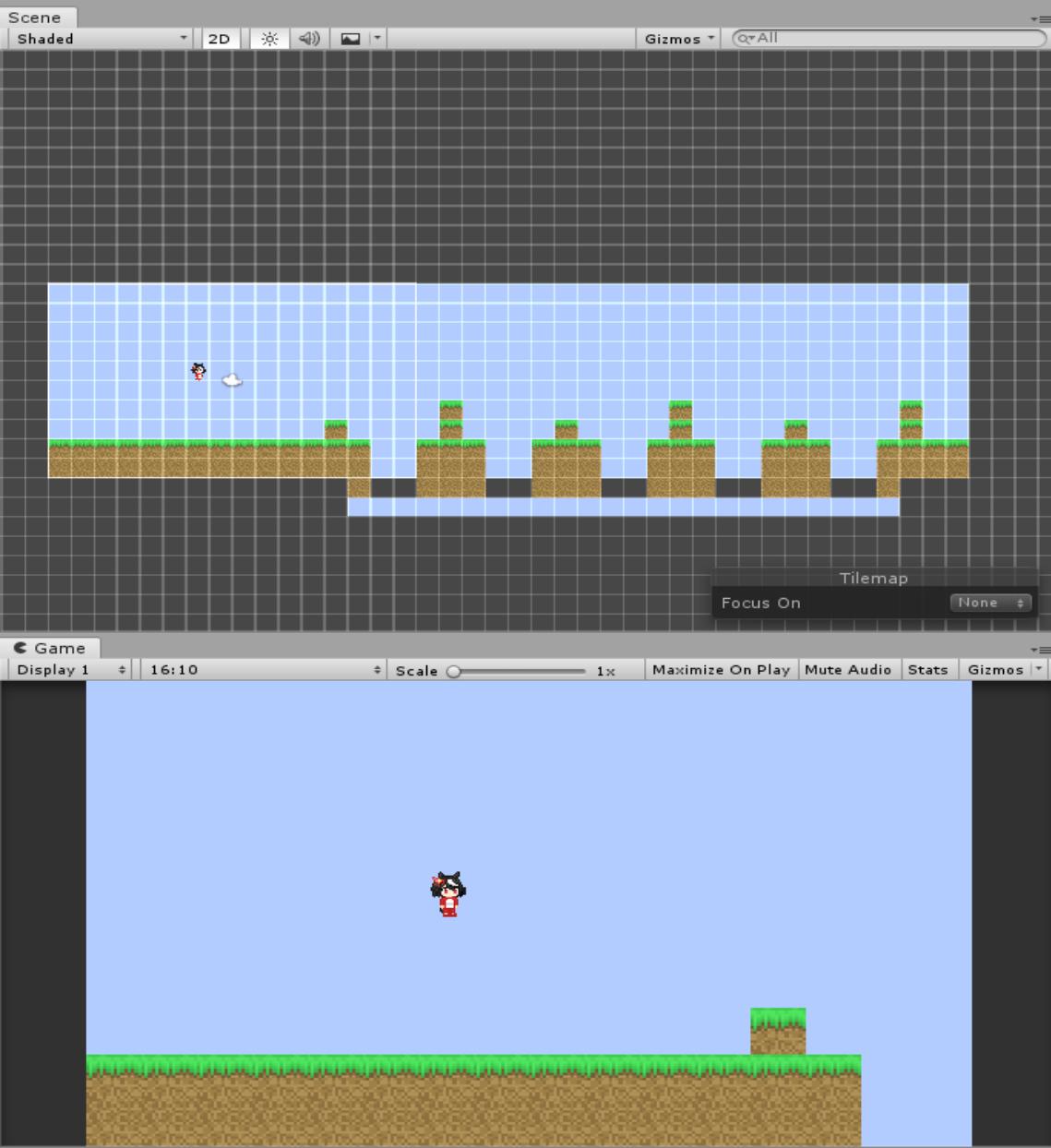
検索

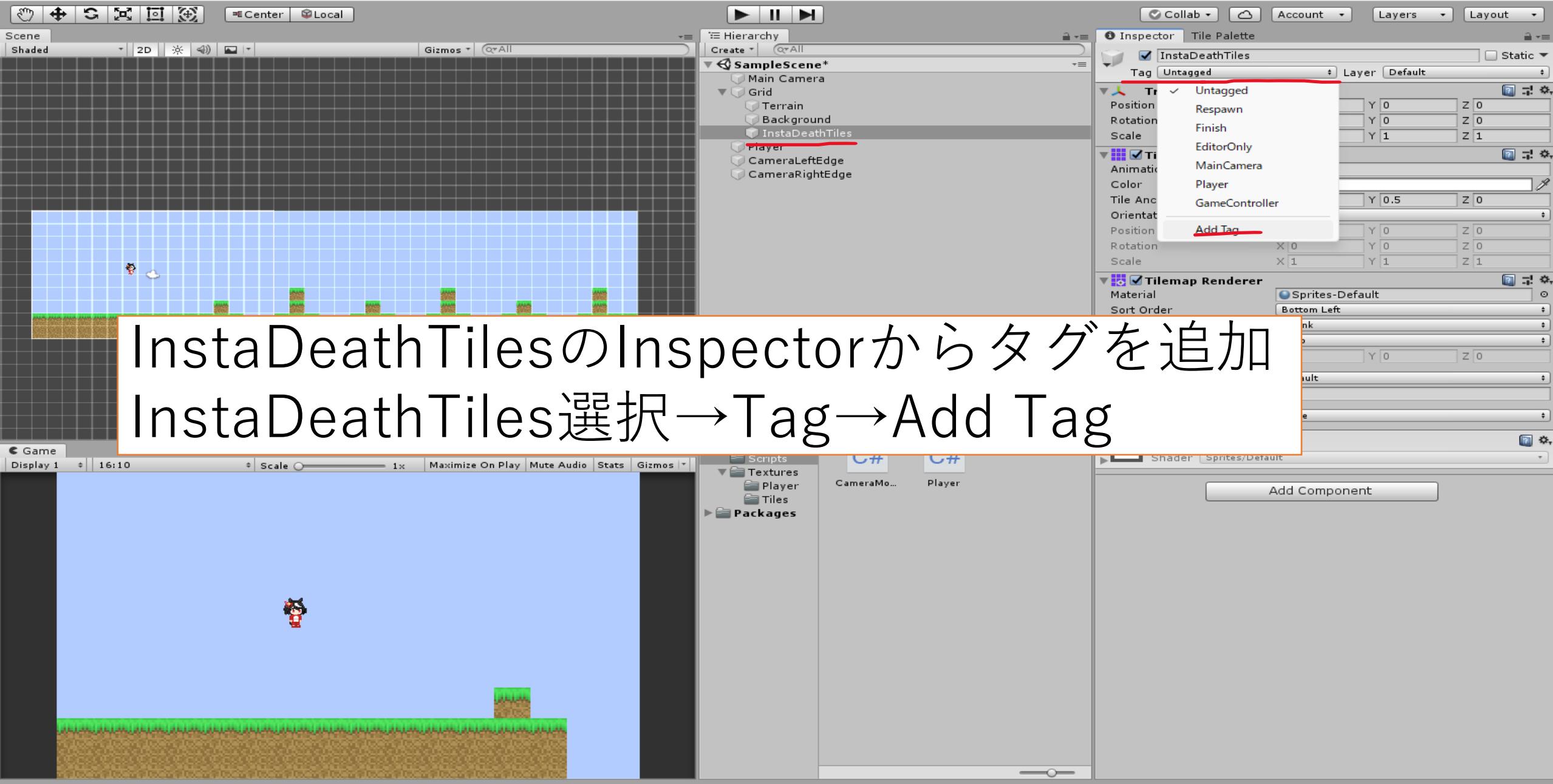
8:32
2024/05/21

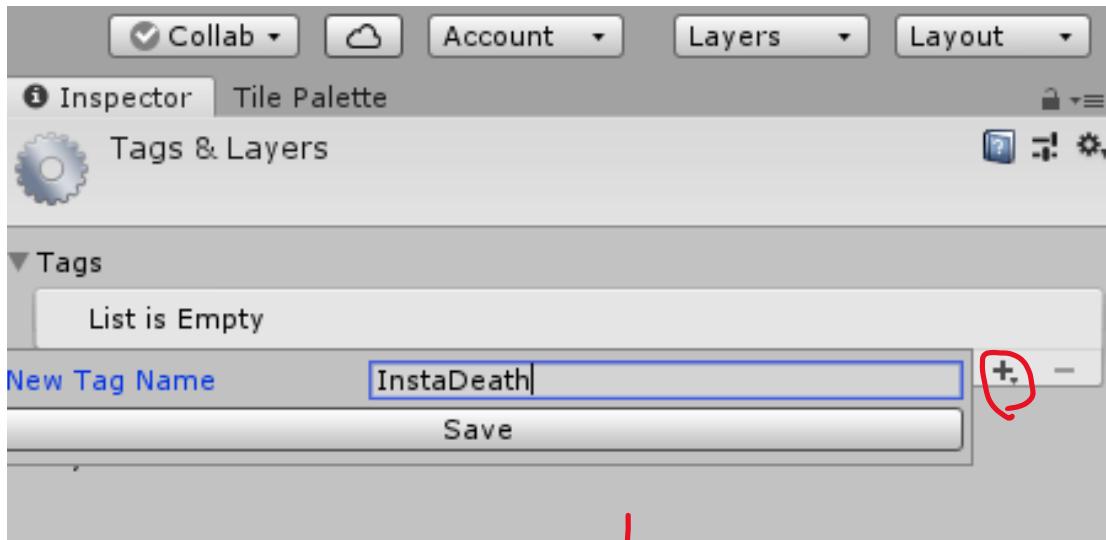


Center Local

Gizmos All







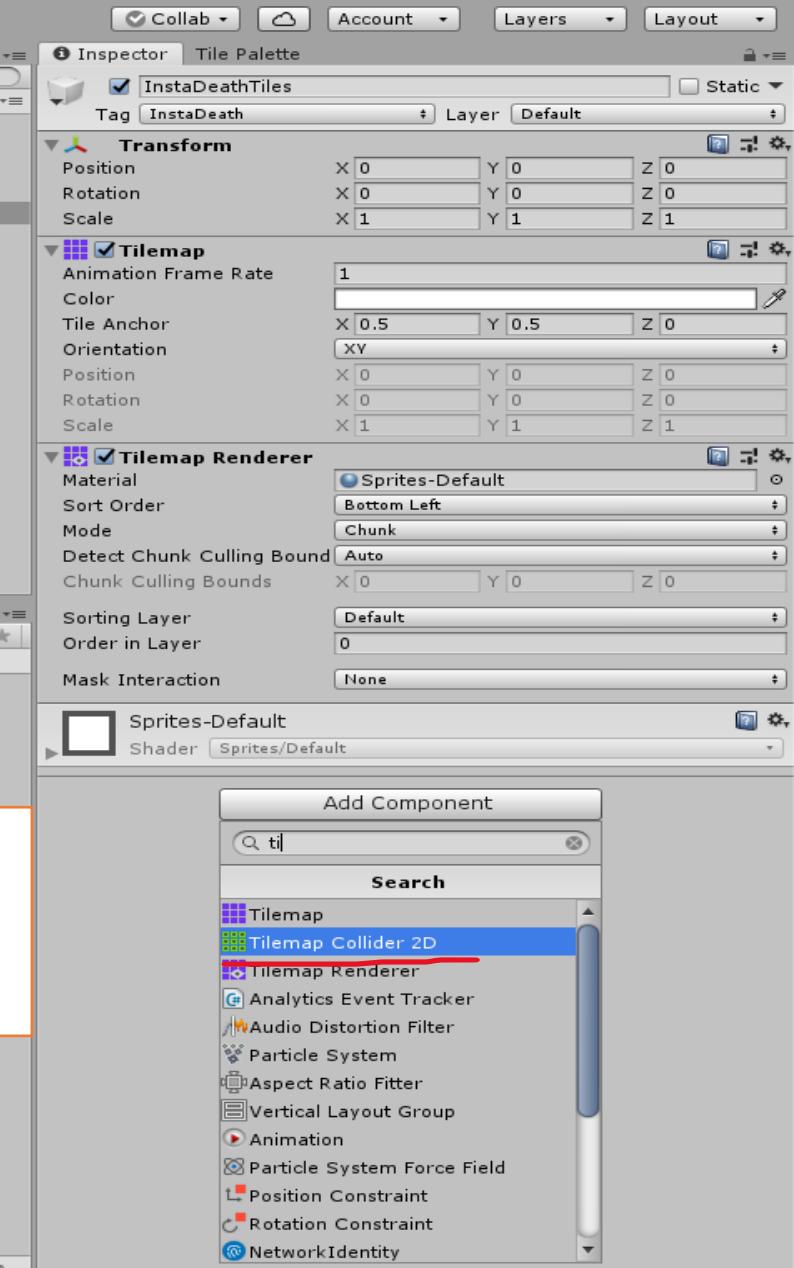
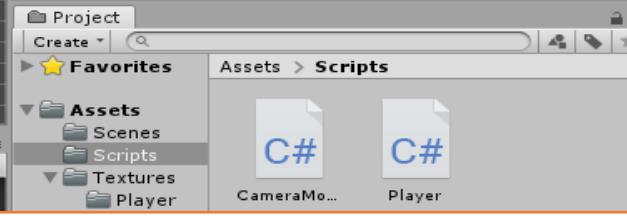
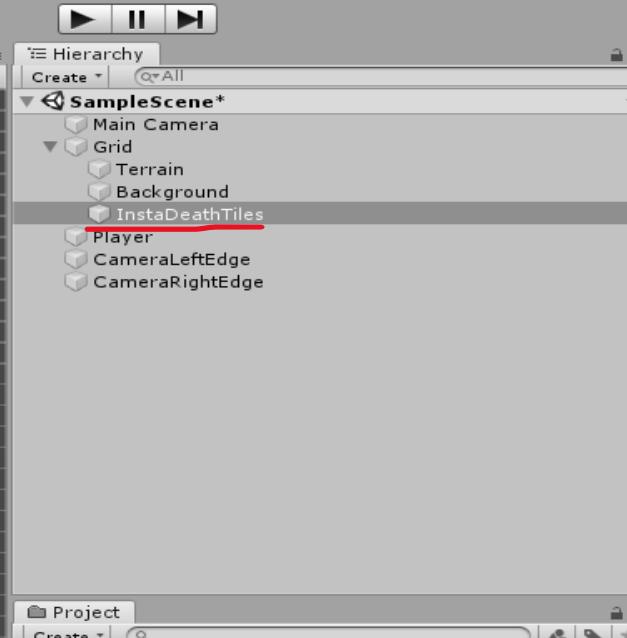
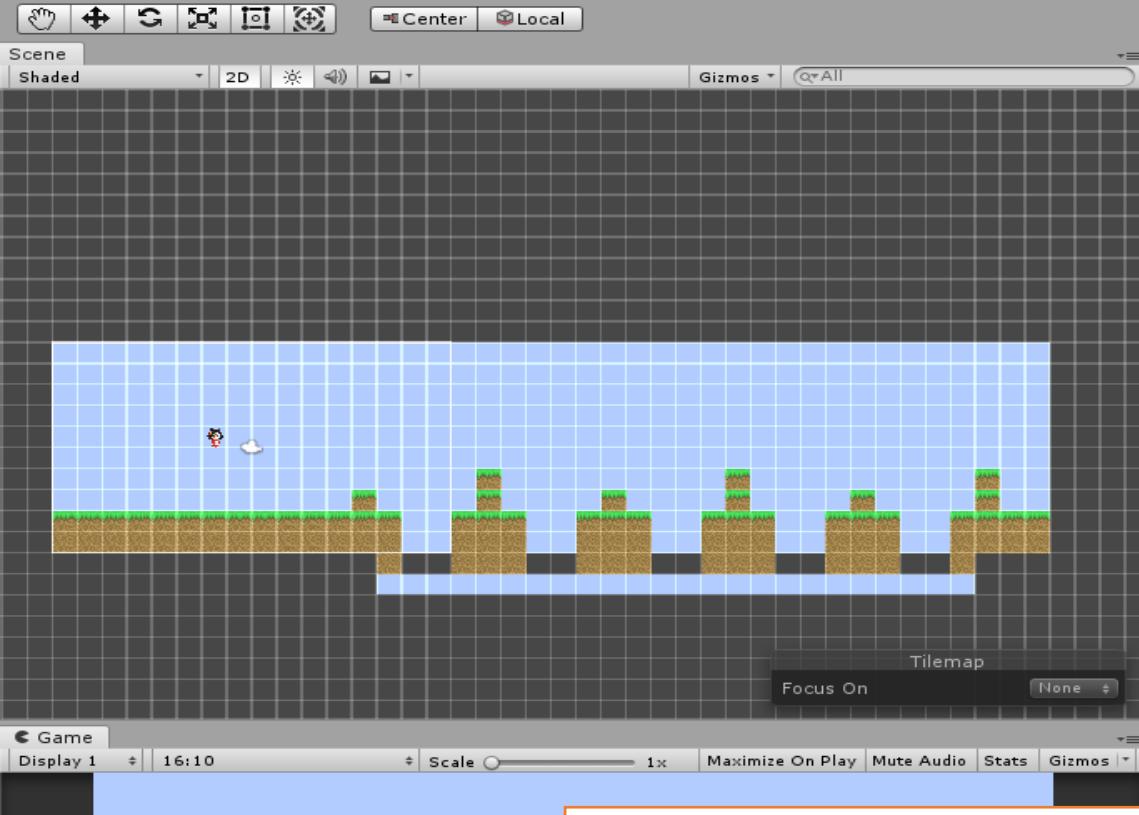
「+」→名前：InstaDeath
→Save

Player
CameraLeftEdge
CameraRightEdge

InstaDeathTiles

InstaDeath

プレイヤーがこのタグがついたオブジェクトに触れるとゲームオーバーになるようにする



InstaDeathTiles選択
→ Tilemap Collider 2D追加

```
//Player.cs
```

```
//前のコード省略
```

```
private bool isGrounded()
{
    BoxCollider2D c = GetComponent<BoxCollider2D>();
    return Physics2D.BoxCast(c.bounds.center, c.bounds.size, 0f, Vector2.down, .1f, GroundLayer);
}
```

```
private void OnCollisionEnter2D(Collision2D collision)//(m)
{

```

```
    if (collision.gameObject.CompareTag("InstaDeath"))
    {
        SceneManager.LoadScene("SampleScene");
    }
}
```

```
//タグがInstaDeathのオブジェクトと衝突した際に、シーンをロードし直すという処理を行っている
```

```
}
```

```
//Player.cs
using UnityEngine;
using UnityEngine.SceneManagement;

public class Player : MonoBehaviour
{
    public float MoveSpeed = 3f;
    public float JumpForce = 15f;
    public LayerMask GroundLayer;

    private Rigidbody2D rb;

    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    void Update()
    {
        //Player Movement
        rb.velocity = new Vector2(Input.GetAxisRaw("Horizontal") * MoveSpeed, rb.velocity.y);

        if (Input.GetButtonDown("Jump") && isGrounded())
        {
            rb.velocity = new Vector2(rb.velocity.x, JumpForce);
        }

        //Sprite Flip
        if (rb.velocity.x > 0)
        {
            GetComponent<SpriteRenderer>().flipX = false;
        }
        else if (rb.velocity.x < 0)
        {
            GetComponent<SpriteRenderer>().flipX = true;
        }
    }

    private bool isGrounded()
    {
        BoxCollider2D c = GetComponent<BoxCollider2D>();
        return Physics2D.BoxCast(c.bounds.center, c.bounds.size, 0f, Vector2.down, .1f, GroundLayer);
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.CompareTag("InstaDeath"))
        {
            SceneManager.LoadScene("SampleScene");
        }
    }
}
```