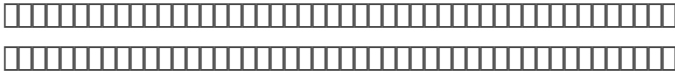
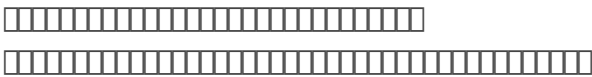


????????



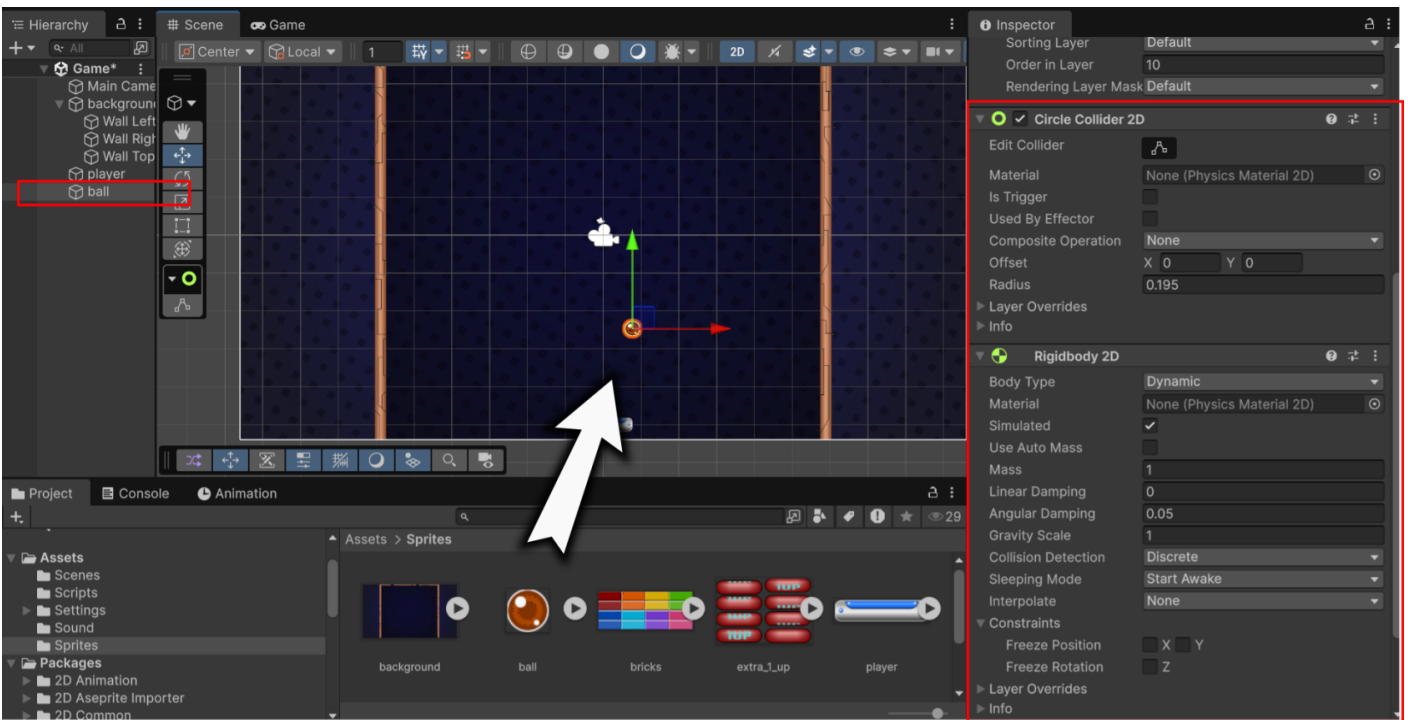
??????



CircleCollider2D



Order In Layer



Gravity Scale

???????????????? BallMove



BallMove



Unity

??????????

```

using UnityEngine;

// 球の移動速度を Inspector で設定できるようにする
public class BallMove : MonoBehaviour
{
    // Inspector で設定できるようにする
    [SerializeField]
    private float speed;

    // RigidBody2D を取得する
    private Rigidbody2D _rigidbody;
}

```



## ??? - Start()



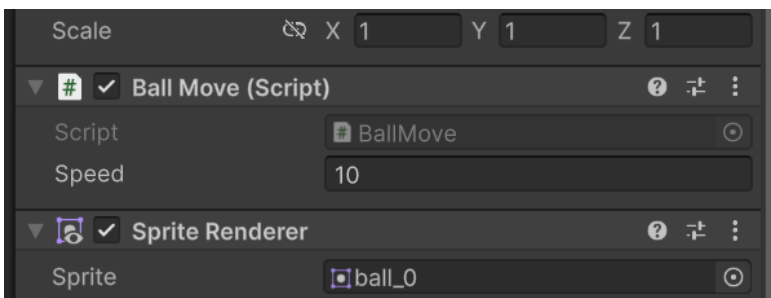
```

private void Start()
{
    // RigidBody2D を取得する
    _rigidbody = GetComponent<Rigidbody2D>();

    // 球の移動速度を設定する
    _rigidbody.linearVelocity = new Vector2(0, 1) * speed;
}

```

## ??



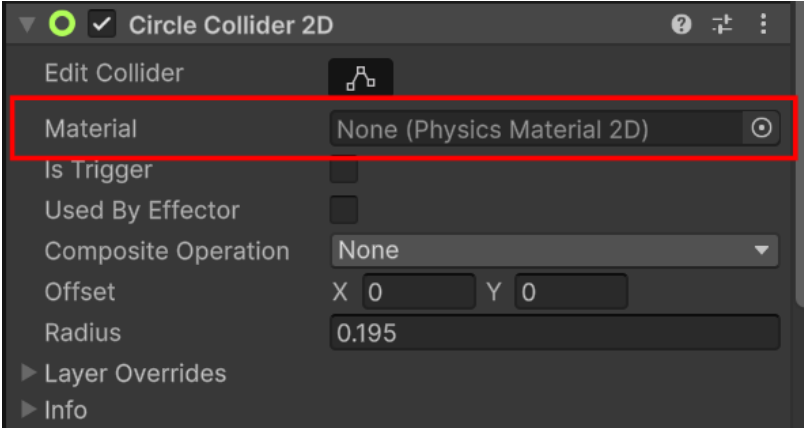
????????????????????

Unity [ ] (bounciness)

[ ]

[ ]

Physics Material 2D [ ] Physics: [ ] Material [ ]



[ ]

100%

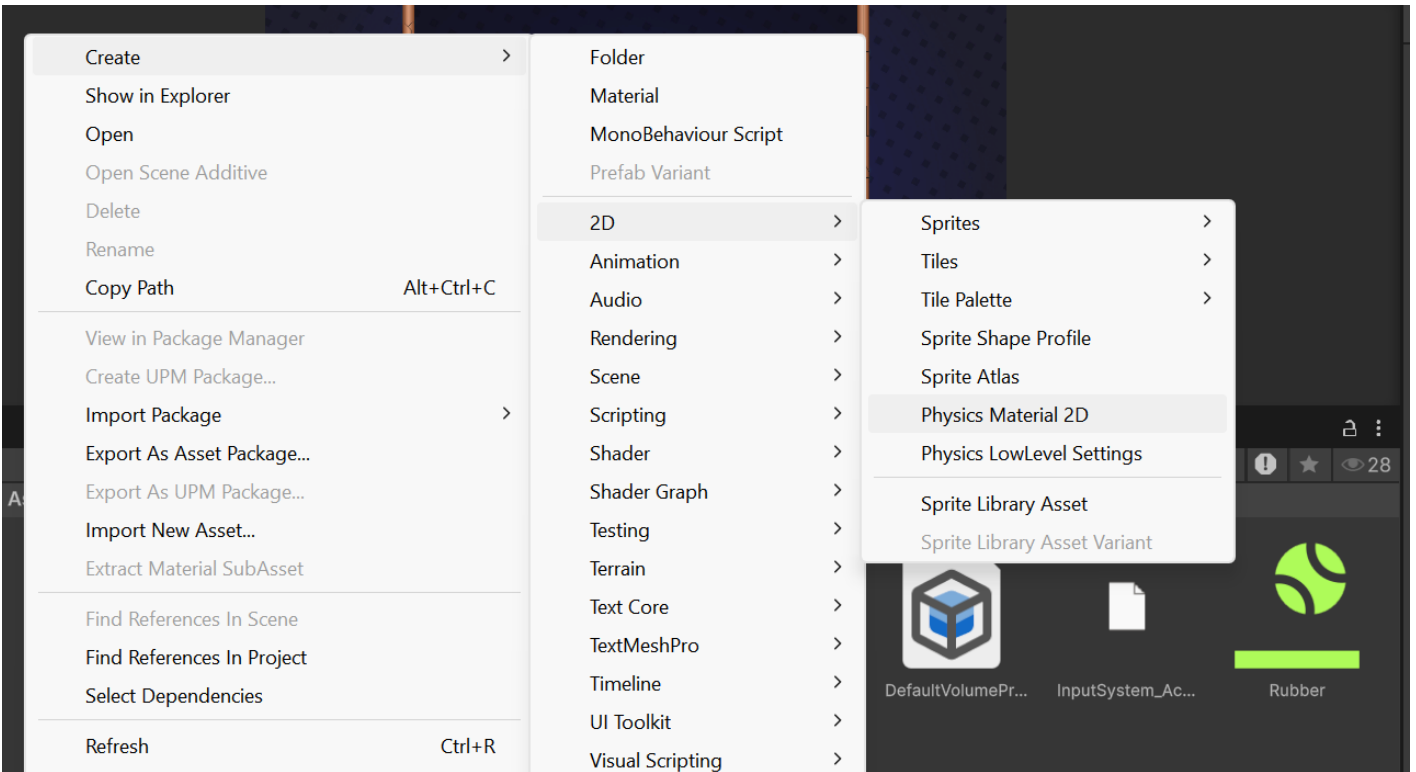
[ ]

Assets

[ ]

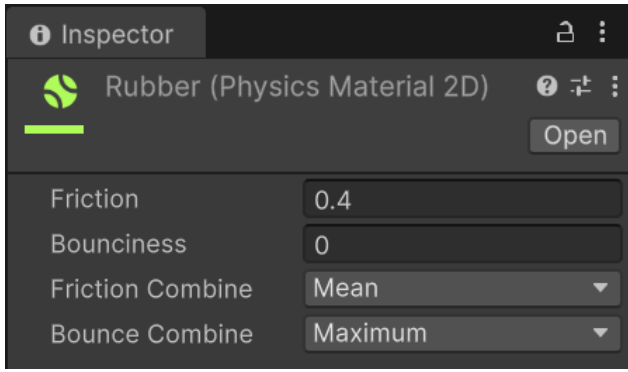
Create > 2D > Physics Material 2D [ ]

Rubber [ ]



Rubber [ ]

Inspector [ ]



Friction:  - 2

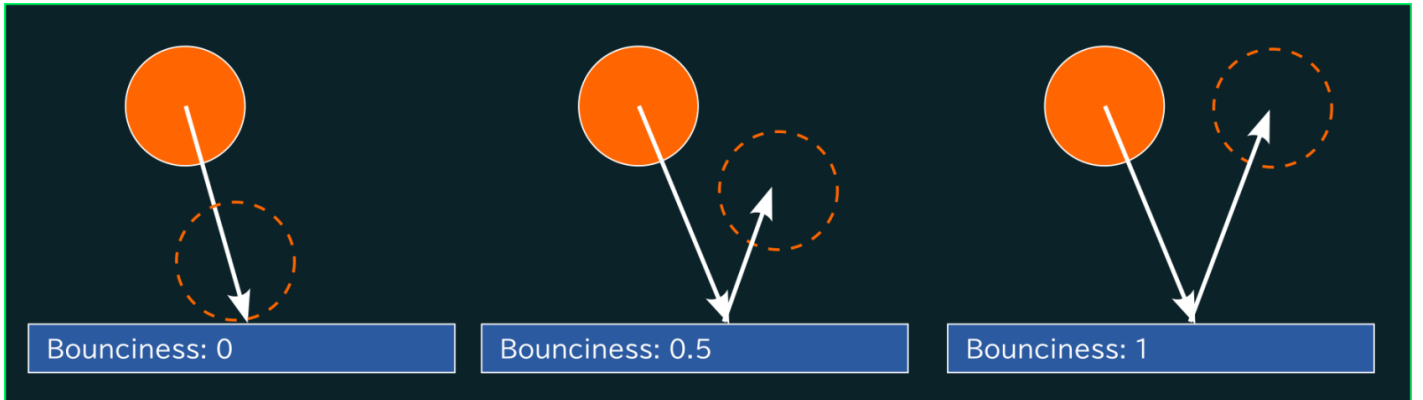
Bounciness  -

Friction Combine / Bounce Combine

Minimum

Maximum

Mean  2



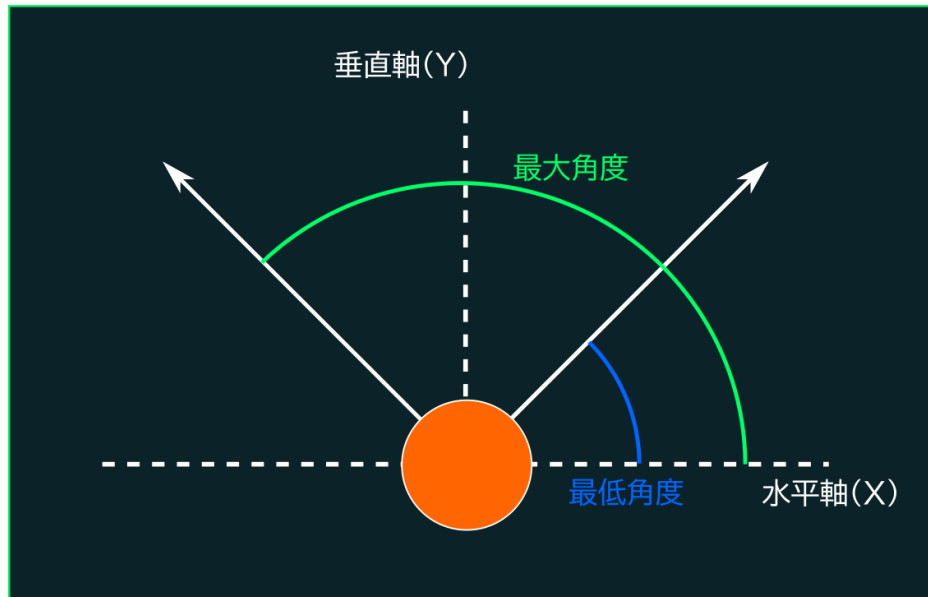
- Friction
- Bounciness
- Friction Combine  Minimum
- Bounce Combine  Maximum

Rubber

CircleCollider2D

??????????

※



[Progress bar]  
 [Progress bar] Start [Progress bar]

C#

```

private void Start()
{
    // 初始化变量
    _rigidbody = GetComponent<Rigidbody2D>();

    // 初始化速度
    // _rigidbody.linearVelocity = new Vector2(0, 1) * speed;

    float minAngle = 45; // 最小角度 - 45度
    float maxAngle = 135; // 最大角度 - 135度

    // 生成随机角度
    float angle = Random.Range(minAngle, maxAngle);

    // 将角度转换为弧度
    angle = angle * Mathf.Deg2Rad;

    // 计算方向向量
    var direction = new Vector2();
    direction.x = Mathf.Cos(angle);
    direction.y = Mathf.Sin(angle);

    // 应用速度
    _rigidbody = GetComponent<Rigidbody2D>();
  
```

