

Học sử dụng Accelerometer trong Unity

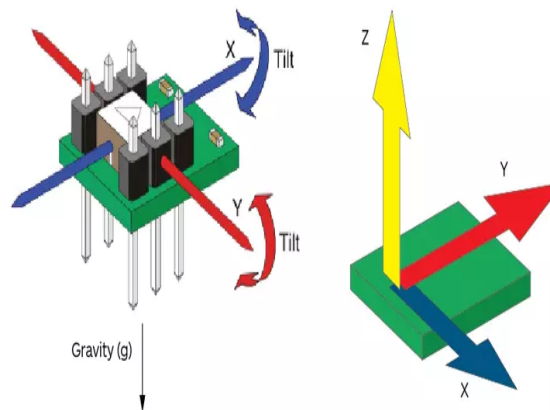
Đã ai từng chơi game Temple Run thì cũng biết cách chơi của nó là chỉ cần nghiêng điện thoại để di chuyển nhân vật trong game



Ta thấy nhân vật di chuyển có ma sát, tăng tốc nhanh chậm... chuyển động y như ngoài đời thật.

Vậy làm sao chúng ta có thể làm được như vậy. Câu trả lời là Unity đã cung cấp cho chúng ta 1 công cụ mô phỏng chuyển động này là Accelerometer.

Accelerometer là gì? Nói ngắn gọn: Accelerometer là 1 con chip nhỏ được gắn ở điện thoại nhằm phát hiện di chuyển của điện thoại khi nó di chuyển hoặc thay đổi trạng thái.



Diagrams illustrating the axes of 2-axis (left) and 3-axis accelerometers. This particular 2-axis sensor is also capable of tilt measurement.
Image credit: Parallax / Kerry Wong

Chúng ta muốn lấy data về việc di chuyển điện thoại thông qua Accelerometer trong Unity thì rất đơn giản. Unity cung cấp Input.acceleration để lấy data từ accelerometer. Chúng ta code demo như sau:

```
using UnityEngine;
using System.Collections;
public class AccelerometerInput : MonoBehaviour {
    void Update () {
        transform.Translate(Input.acceleration.x, Input.acceleration.y, 0);
    }
}
```

Chúng ta chỉ sử dụng 2 biến là Input.acceleration.x & Input.acceleration.y để có thể lấy tọa độ di chuyển cho nhân vật. Khá dễ đúng không a e. Nâng cao hơn 1 tí mình sẽ cho nhân vật di chuyển và xoay theo accelerometer

```

public class Controller : MonoBehaviour
{
    public float smooth = 0.4f;
    public Text inputValue;
    public float newRotation;
    public float sensitivity = 6;
    private Vector3 currentAcceleration, initialAcceleration;
    void Start()
    {
        initialAcceleration = Input.acceleration;
        currentAcceleration = Vector3.zero;
    }

    void Update () {
        //pre-processing

        currentAcceleration = Vector3.Lerp(currentAcceleration, Input.acceleration -
initialAcceleration, Time.deltaTime/smooth);

        newRotation = Mathf.Clamp(currentAcceleration.x * sensitivity, -1, 1);
        transform.Rotate(0, 0, -newRotation);
    }
}

```

Ta có thể tinh chỉnh code để ra được 1 kết quả như mong muốn.