?????

スマホの加速度センサーを使い、強い揺れを検知したら「歩」として検知し、移動処理を追加しましょう
。

?????

スマホは純粋の万歩計がないが、加速度センサーで動きを検知ができる。「 StepDetector
」というスクリプトを作成し、歩調を検知しましょう。なお、回転と同様に、まずキーボードを使い、 PC
で確認できるようにしましょう。

```
// 歩調を検知する
public class StepDetector : MonoBehaviour
{
    // 検知したときのイベント宣言
    public event System.Action Step;

    // 1歩の時間
    [SerializeField] private float stepTime = 0.2f;

    // 次の歩までの時間
    private float waitTime;

    private void Start()
    {
        waitTime = stepTime;
    }

    private void Update()
    {
        // 次の歩までの時間になっていない
        if (waitTime > 0)
        {
            // 待機時間を減らす
            waitTime -= Time.deltaTime;
            return;
        }
```

```
        // 入力ある？（0以上）→なければ終わり
        if (Input.GetAxis("Vertical") <= 0)
            return;

        // 1歩進み、時間をリセット
        Step?.Invoke();
        waitTime = stepTime;
    }
}
```

そして、イベントに登録し、前を進む「MoveController」は:

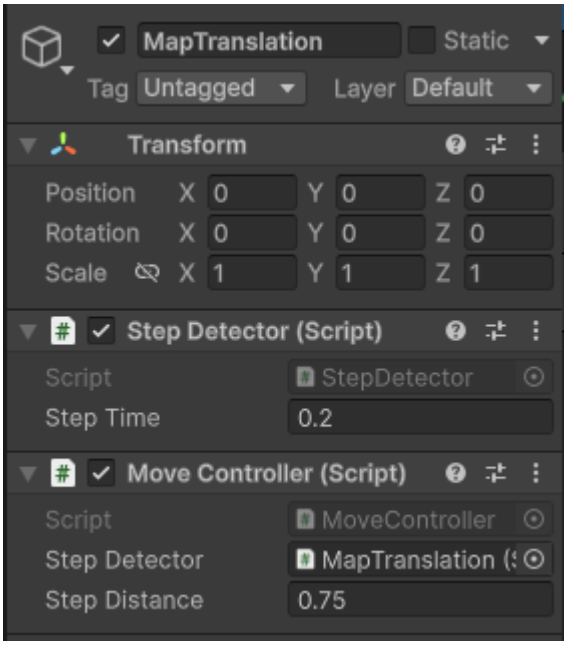
```
// レーダーの移動を制御する
public class MoveController : MonoBehaviour
{
    // 歩調を検知するため
    [SerializeField]
    private StepDetector stepDetector;

    // 1歩進んでら、移動する量
    [SerializeField] private float stepDistance = 0.75f;

    private void Start()
    {
        // 「歩」イベントに登録
        stepDetector.Step += OnStep;
    }

    private void OnDestroy()
    {
        // イベント登録解除
        stepDetector.Step -= OnStep;
    }

    // 1歩を検知した
    private void OnStep()
    {
        // 地図を下の方に移動（前に進むように見える）
        transform.position += Vector3.down * stepDistance;
```



```
    }
}
```

この2つのスクリプトを「 MapTranslation」にアタッチし、キーボードで確認しましょう:

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

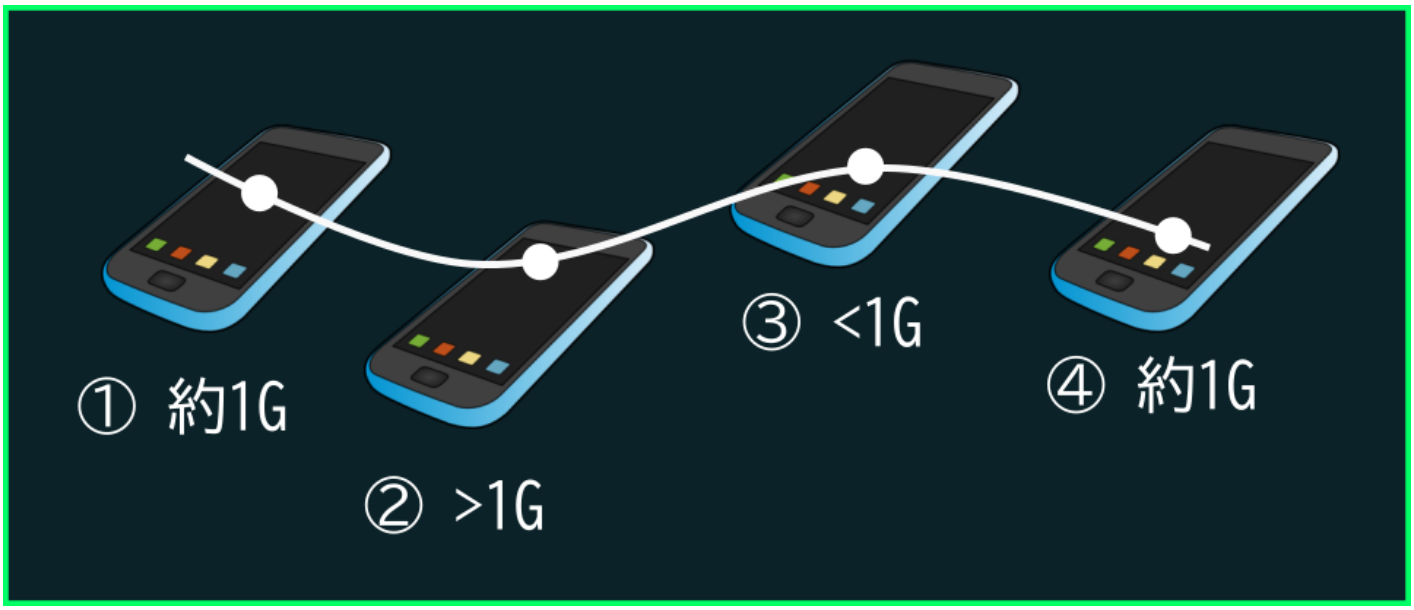
加速度センサー「 G」単位で加速度を測定する装置である。 1G

は普通の地球の重力なので、スマホは動かないときは、真下に向かう 1G のベクトルを測定する:



ただし、歩くときは、人間の動きはまっすぐでななく、上がったり、下がったりする：

とすると、加速度センサーの値が変わる：



① 動いていないときは、地球の重力のみで、加速度センサーの強さは約１ Gである
② そして歩き始め、体が下がり、スマホが下の方加速するので、加速度センサーの強さは約１ Gを超える
③
次の歩を踏む前、足を上げないといけないので、体が上の方に加速するので、加速度センサーの強さは約１ G よりも小さくなる
④ 最後に止まったら、また１ Gに戻る

この現象を使い、単純な歩調を検知を実装できる。加速度センサーの強さが「 1G
」を超えたら、一歩を進んだことにする。 StepDetector を編集し、スマホの対応を追加しましょう：

// 歩調を検知する
public class StepDetector : MonoBehaviour
{
// 検知したときのイベント宣言
public event System.Action Step;

// 1歩の時間
[SerializeField] private float stepTime = 0.2f;

// 「歩」とする加速度の強さ
[SerializeField] private float stepForce = 1.05f;

// 次の歩までの時間
private float waitTime;

private void Start()
{
// 加速度センサーあったら、有効化する

```
    if (Accelerometer.current != null)
        InputSystem.EnableDevice(Accelerometer.current);

    waitTime = stepTime;
}

private void OnDestroy()
{
    // 加速度センサーあったら、無効化する
    if (Accelerometer.current != null)
        InputSystem.DisableDevice(Accelerometer.current);
}

private void Update()
{
    // 次の歩までの時間になっていない
    if (waitTime > 0)
    {
        // 待機時間を減らす
        waitTime -= Time.deltaTime;
        return;
    }

    // PCの場合
    if (Accelerometer.current == null)
    {
        // 入力ある？(0以上)→なければ終わり
        if (Input.GetAxis("Vertical") <= 0)
            return;
    }

    // スマホの場合
    else
    {
        // 加速度ベクトルの長さ（強さ）を求める
        var accel = Accelerometer.current.acceleration.magnitude;

        // 十分に強くない？ → なにもしない
        if (accel < stepForce)
            return;
```

```
        }

        // 1□□□□□□□□□□□□
        Step?.Invoke();
        waitTime = stepTime;
    }
}
```

□□□□□□□□□□□□□□□□□□□□□

Revision #2
Created 2026-07-29 02:50:04 UTC by Menendez Francisco
Updated 2026-07-29 05:04:41 UTC by Menendez Francisco