

アクションゲームの制作

藤井 太雅

1. まえがき

今回の課題研究で、3D のアクションゲームを作った。

2. 原 理

Unity というゲームエンジンを利用して作った。

3. 研究内容

(1)ゲーム制作の準備

まずゲームのオブジェクトで簡単なステージを作った。後々やりたいことが増えて最終的なステージはこのようになった。

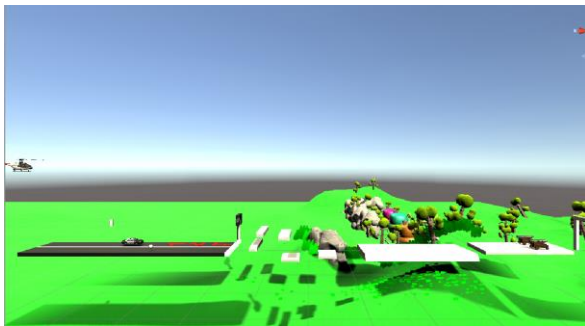


図1 ステージ

(2)Player の動作

今回 ps4 のコントローラーで操作できるように設定した。

△ボタンでジャンプ

左スティックでカメラの操作

右スティックで Player の操作

左スティック+L1 ボタンで 1 速

左スティック+L2 ボタンで 2 速

左スティック+R1 ボタンで 3 速

左スティック+L2 ボタンで 4 速

```
var x = Input.GetAxis("Horizontal");
var y = Input.GetAxis("Vertical");

// このvは移動方向を設定したベクトル
var v = new Vector3(x, 0, y) * speed * Time.deltaTime;

if (Input.GetButton("Rs") && Input.GetButtonDown("second"))
{
    speed = 10.0f;
}
if (Input.GetButton("Rs") && Input.GetButtonDown("L2"))
{
    speed =20.0f;
}
```

図2 Player のプログラム

(3)追尾してくる敵の動作

敵と Player の位置をベクトルの計算で求めて、最短の距離で追尾してくるようにした。

ゲームオーバーになると初期位置に戻るようにした。



図3 追尾してくる敵

```
void Update()
{
    var heading = player.transform.position -transform.position;
    var distance =heading.magnitude;
    var direction =heading/distance;

    transform.position+=direction*Time.deltaTime*4;
}

private void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.tag=="Player")
    {
        transform.position= new Vector3(13.38f,6.145087f,-0.582529f);
    }
}
```

図4 敵が追尾してくるプログラム

(4)大砲の動作

この大砲は **Player** が一定の範囲内に来ると、追尾機能付きの砲丸をインターバル 0.5 秒間隔で放つようにした。

このプログラムは、大砲と **Player** の距離を計算して、ある一定の数値以下になると、弾丸を発射するようにした。

`posa` という変数に現在のターゲットである、**Player** のポジションを取得している。

`posb` は大砲のポジションを取得している。`Distance` 関数で、**Player** と大砲の距離の値求めている。その値を `dis` という変数に代入し、**Player** と大砲の距離が、X 軸、Y 軸共に 50 以下になると、弾丸を発射する制御をしている。

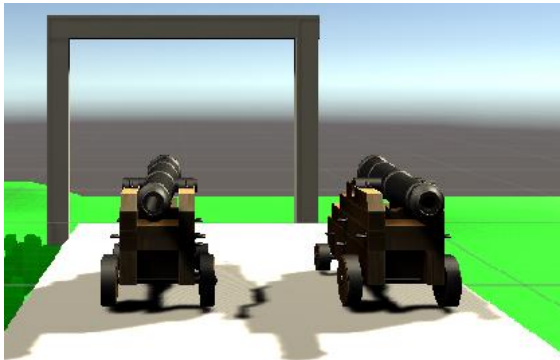


図 5 大砲

```
public GameObject tama;
private void Start()
{
    // 3秒後に実行したあと、以降1秒毎に実行
    Observable.Timer(TimeSpan.FromSeconds(3), TimeSpan.FromSeconds(0.5f))
        .Subscribe(_ => DelayMethod())
        .AddTo(this);

    // 常に3秒間隔で実行
    Observable.Interval(TimeSpan.FromSeconds(1))
        .Subscribe(_ => DelayMethod())
        .AddTo(this);
}

public GameObject player;
private void DelayMethod()
{
    Vector3 posa = player.transform.position;
    Vector3 posb = transform.position;
    float dis = Vector3.Distance(posa, posb);
    if (dis < 50)
    {
        GameObject cptama = (GameObject)Instantiate(tama, transform.position, Quaternion.identity);
        Rigidbody b = cptama.GetComponent<Rigidbody>();
        b.AddForce(transform.forward * 1000f);

        Destroy(cptama, 3.0f);
    }
}
```

図 6 大砲のプログラム

(4)ゲームクリア画面

ゲームクリアすると別のシーンに飛び、アニメーションで、横から”ゲームクリア”という文字が流れて出てくるようにした。

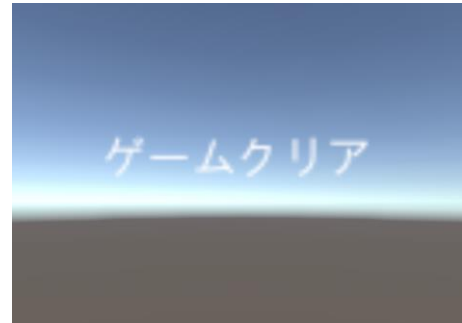


図 7 ゲームクリア

4. まとめ

Unity の知識ほぼ 0 の状態でスタートしたため最初は作業がスムーズにはできなかった。しかし、参考本やネットで調べることによって分からなかったところが理解でき、何とかゲームを完成させることができた。

5. あとがき

最初にこういうゲームにしようと思ってはいたものの、授業の後半になるにつれ、やりたいことがどんどん増えていき、ゲームを完成させるのに時間が掛かった。ゲームを作るのには膨大な時間が必要であることがわかった。

6. 参考文献

Unity の教科書

[【Unity】プレイヤーにカメラを追従しよう！\(3D\) | Camera 入門④ - XR-Hub](#)

[【Unity】RPG 制作 その 6-敵を複数出現させる。 - kuront's blog \(hatenablog.com\)](#)