Unity による 3D シューティングゲームの制作

竹本航大　横山由和

１．研究概要

今回私たちは、Blender と Unity の知識を深めるとともにゲーム開発への経験を積むことを目的とし、3D シューティングゲームを制作した。

２．研究テーマ

シューティングゲームを作る上で今回私たちは、“忍者”をテーマにゲーム作りをおこなった。理由としては、制作時期に２人ともがアニメ“NARUTO”にハマったからである。また昨年も実習でゲーム制作をおこなったが制作に失敗したため、その経験から今回は多くの情報が入手でき、学習しやすいシューティングゲームを制作した。

３．研究の具体的内容

(1)ゲーム制作の流れ

ゲーム制作は図１のように行った。

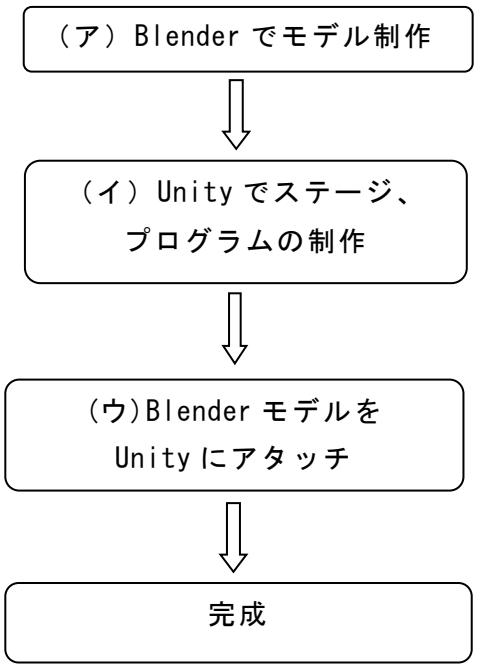
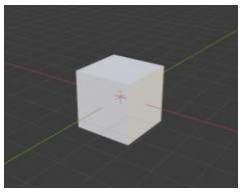


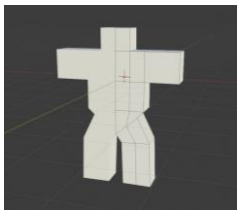
図１　制作の流れ

（ア）　Blender でモデル制作

Blender での制作過程を次に示す。

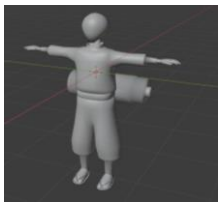


Blender 画面にオブジェクトの立方体を表示する



オブジェクトの立方体を押し出したり、回転、拡張することによって大まかな形を作る。

オブジェクトが人型の形になるように少しずつ成形していく。



オブジェクトにまるみをつけるフラットジェートをつけてより完成に近づける。

最後にオブジェクトに色をつけて完成。

図２　Blender での制作過程

（イ）Unity を使ったゲーム制作の流れ

Unity での制作過程を図３に示す。

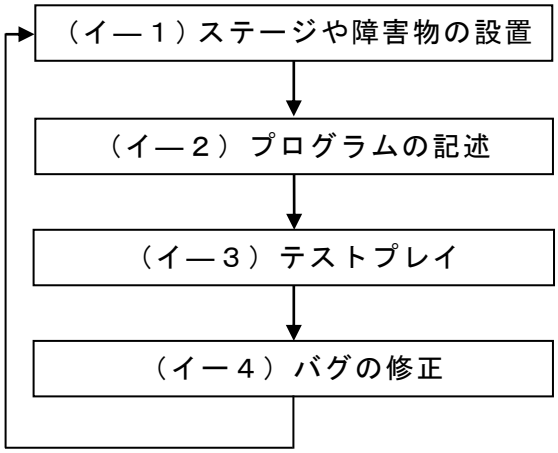


図３　Unity での制作過程

（イ―１）ステージや障害物の配置

ステージの地面を設置し、そこから敵や障害物を配置する。色の変更やコンポーネントの追加もここで行う。

（イ―２）プログラムを記述

プレイヤ操作や HP、敵の挙動、シーンの移動などをプログラムする。プログラムは Visual Studio を使って C#で記述する。

（イ―３）テストプレイ

記述したプログラムをアタッチして正常に動作するかどうかを確認し､バグが出ず想定していた状態になれば(イ―１)に戻る。

（イ―４）バグの修正

テストプレイ中に意図しない挙動があった場合､プログラムの修正を行う。修正したら再びテストプレイを行い､正常に動作すれば（１）に戻る。

作業画面とスクリプトを図４と図５に示す。

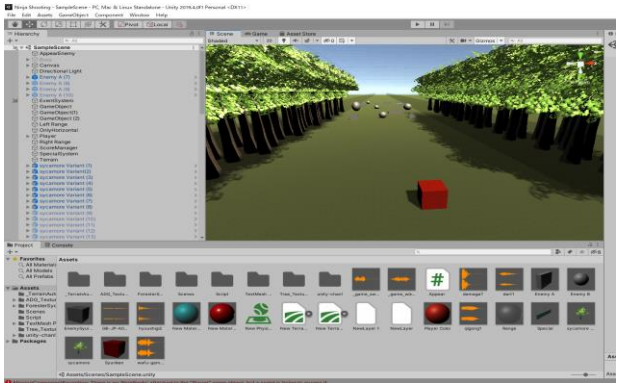


図４ Unity の作業画面

```
void CalculateMove()
{
        horizontalInput = Input.GetAxis("Horizontal");
    Vector3 direction = new Vector3(horizontalInput, 0, verticalInput);

    Vector3 velocity = direction * speed;
    velocity.y -= gravity;
    velocity = transform.transform.TransformDirection(velocity);
    controller.Move(velocity * Time.deltaTime);
    if (controller.isGrounded)
    {
        horizontalInput = Input.GetAxis("Horizontal");
    }
}
```

図５ プレイヤ操作のスクリプト

（ウ）完成

Blender モデルを Unity にアタッチして図６のようになり完成。

図６　完成図

４．研究のまとめ

今回課題研究を通して、私たちは Unity と Blender でできること、またどのようにして扱うのかを学ぶことができたが、扱うことに苦戦し、思うように作業が進行せず作品自体の質は当初予定していたものより大きく落ちてしまった。また初めての共同作業ということもあって意見の食い違いが多く苦労した。しかし今回の課題研究で得られたことは今後の作品作りや、就職したときに活きてくると思う。

５.参考文献

「Qiita」
https://qiita.com/
「blender2.8 入門：blender の操作方法」
https://czpanel.com/lecture/blender/basic/operation/