

# Unity を使ったスマートフォンホラーゲーム制作

木村 晟嗣

## 1. 研究概要

Unity を使って Android で動作する 3D ホラーゲームを制作し、Unity の使い方や 3D モデルの知識を深めて、個人でどこまで開発できるのか知る。

## 2. 研究の具体的内容

### (1) ゲームの設計

まず初めに、ゲームのジャンルを決めることにした。昔、青鬼などのフリーゲームを遊んでいて、自分もこんなゲームを作りたいと思っていたので、ホラーゲームを制作することにした。しかし、スマートフォンのホラーゲームは既に多くの種類が出ているので、他のホラーゲームと内容が被っていても面白くないと思い、ゲームクリアの目的を「脱出ではなく敵を倒すこと」に決定した。

### (2) モデリング

ジャンルが決定した後は、プレイヤーと敵キャラの形となる 3D モデルのデザインを行った。最初は Blender という 3DCG ソフトを使ってモデリングをしようと思ったが機能がとても多く、これの勉強だけで一年間が潰れてしまいそうだったので、Magica Voxel というフリーのモデリングソフトを使用した (図 1)。

Magica Voxel は、マインクラフトのようにブロックを積み重ねて 3DCG を作成することができるので、誰でも簡単に作ることができる。また 3D モデルの模様となるテクスチャの出力もできるため Unity にデータを取り込む際もスムーズに行えた。

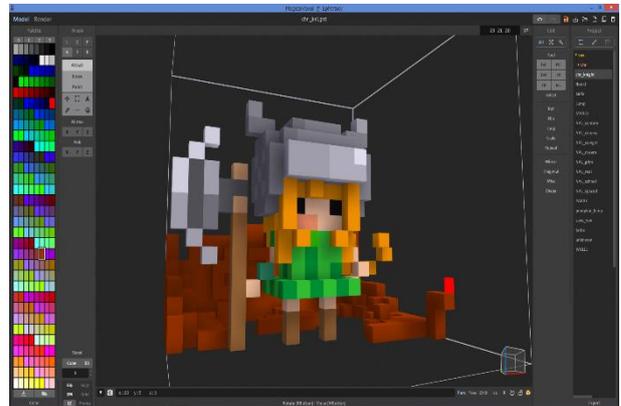


図 1 Magica Voxel

最初は敵キャラの 3D モデルも Magica Voxel で作成していたが、歩行のアニメーションを付けようとした際に、Magica Voxel の 3D モデルでは相性が悪かったため、アセットと呼ばれる他の人が作ったモデルを使用した。

Unity には Asset Store というショップページがあり、キャラクタやマップを無料、又は有料でダウンロードすることができ、プログラムしかできない人間でもゲームが作れるようになっている。

### (3) タイトルシーン

今回はスマートフォンのアプリを作るため、なるべく動作が軽く実行できるようシンプルな設計を目指した。ホラーゲームなので、タイトルから雰囲気を出そうと思いフリー素材である怨霊フォントというホラー系フォントを使用した (図 4)。

またゲームの世界観を表すためにアニメーションを付けたプロローグを作成した (図 5)。一行ずつ文字がフェードインしていくという風にしたかったので、Unity の機能である

Animation を使い、文字の透明度を下げていくアニメーションを作成した。

プロローグに対して、一行ずつ個別のアニメーションを作るのは大変なので、Prefab という Unity の機能を使用した。Prefab とは一度作成したオブジェクト（物体）から設計図を作り出す機能であり、設計図から、同じ機能を持っているが独立したオブジェクトを作ることができる。インスタンスに変更を加えてもオリジナルは保たれて、Prefab のデータを変更すると、Prefab から作成した全てのオブジェクトに変更が加わるので、後から機能を変えたいと思った時にも便利である（図2）。

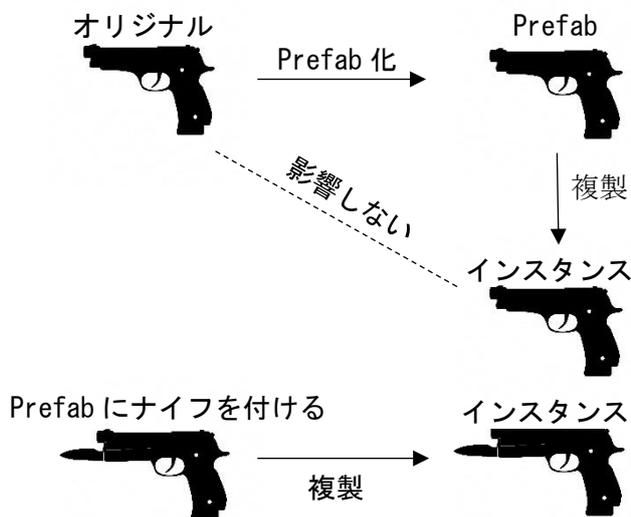


図2 Prefab 化のイメージ

アニメーションの管理にはコルーチンを使用した。コルーチンとはサブルーチンの亜種のようなものであり、処理を中断した後に、続きから再開できるという特徴がある。これはC#に備わっている仕組みだが、Unity では処理を中断してから再開するまでの時間を指定することができる（図3）。

```
14 public IEnumerator Loading()
15 {
16
17     for (int i = 0; i < 5; i++)
18     {
19         Setsume[i].SetActive(true);
20         yield return new WaitForSeconds(2);
21     }
22     Bloods.SetActive(true);
23     skip = true;
24
25 }
```

図3 アニメーション管理のプログラム

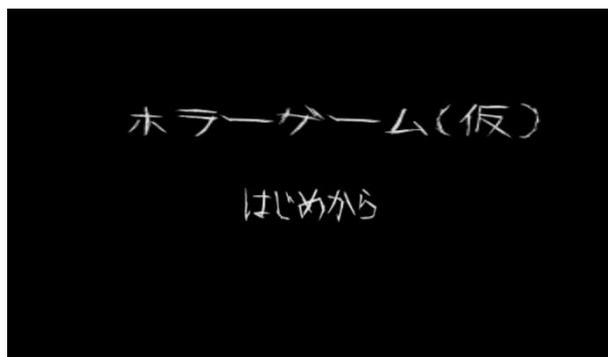


図4 タイトル画面

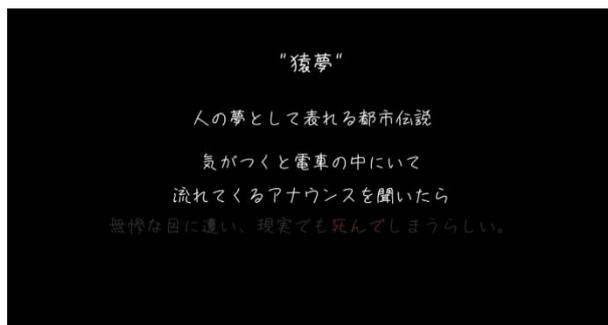


図5 プロローグ

#### (4) 探索について

今回のゲームでは、クリアの目的は敵を倒すことにしているが、ゲーム中でその説明はしていないのでマップの探索がメインとなっている。だが、黙々とマップ内の探索を続けていても、ゲームのストーリーが分からず今何をすればいいか分からなくなってしまうので、スマートフォンの画面下部にキャラクターの発言を表示させることでゲームの進行が滑らかになるようにした（図6）。



図6 ゲームスタート時の画面

実機でのプレイヤーの移動方法にはバーチャルパッドを用いた。バーチャルパッドとは、タッチパネルに入力された座標の遷移によって、処理を切り替えるプログラムであり、本作で

は、画面左を上下にスライドすることでキャラクターの前後移動をし、画面右をスライドすることで、視点を上下左右に回転するようにした。

探索メインのゲームとなっているので、マップのオブジェクトに触れた際に個別のメッセージを表示するようにした(図8)。

Unityにはタグという機能があり、好きな名前のタグをオブジェクトに付けることができる。Unityでは当たり判定を持つオブジェクトに、同じく当たり判定のあるオブジェクトが接触するとタグ情報などが取得されるようになっている。そのため、プレイヤーのカメラの前に当たり判定を付けることで、オブジェクトに対応した個別のメッセージを表示することができた(図7~9)。

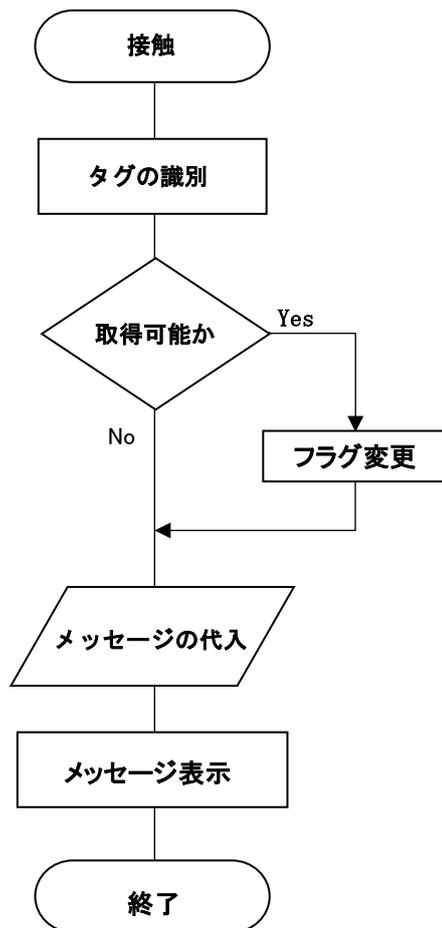


図7 オブジェクト判定のフローチャート

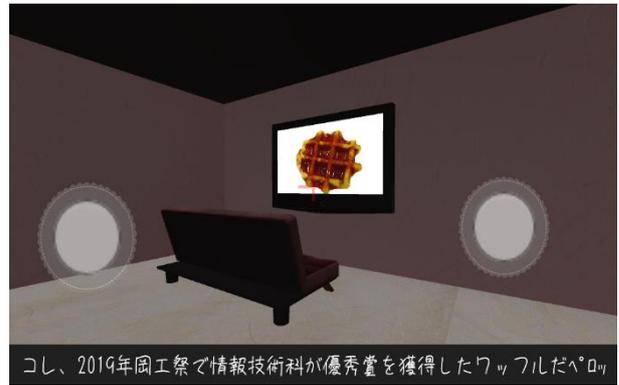


図8 テレビに接触した際のメッセージ

```

else if (other.gameObject.tag == "Monitor") {
  StartCoroutine(Messageoff("コレ、2019年岡工祭で情報技術科が優秀賞を獲得したワッフルだべロッ"));
} else if (other.gameObject.tag == "LED") {
  StartCoroutine(Messageoff("灯りがある...よかった....."));
  sound.PlayOneShot(Effects[0]);
  LedJmp.SetActive(true);
  Ledobj.SetActive(false);
}

```

図9 オブジェクト識別のプログラム(一部)

### (5) 動作テスト

作成したプログラムのテストには、Unity Remote というスマートフォンのアプリを用いた。このアプリは、PC とスマートフォンを接続することで、Unity で開発しているプロジェクトをアプリにしないままスマートフォン上で動かすことができるため、実行可能な形式へと変換する時間を短縮でき、スムーズに動作テストを行うことができた。

### (6) 敵キャラについて

ホラーゲームといえば敵に追い掛けられるのが怖い、という先入観が自分の中にあっただので、それを実現するために、Unity に備わっている NavMesh という機能を使用した。NavMesh とは、当たり判定のあるオブジェクトにぶつからないように、ターゲットまで辿り着く AI である。処理は少し重くなってしまうが、少ない操作とプログラムでターゲットであるプレイヤーまで、敵キャラを追跡させる機能を実装できた。

また、本作は敵キャラを倒すことが目的のため、プレイヤーの攻撃手段としてナイフを投げることにした。最初は銃での攻撃にしようとも考えたが、銃弾の作成の手間や、射撃時に発生

するエフェクトが処理を重くすることを考慮し、銃と比べて処理の軽いナイフを投擲するという方法に決定した（図10）。



図10 ナイフの取得&攻撃

### 3. 研究のまとめ

今回の課題研究で分かったのは、ゲーム制作は個人ではなく、必ずチームを組んで臨んだほうが良いということだ。ゲームを制作しているときに辛かったのは、「本当にこのままで面白いのか」と考えたときに、改善点などを相談できるチームがないので、一人で悩み抜くしかなかったことである。しかしながら、制作に必要な知識は全て自分に取り込むことができるので、個人での制作も悪い面ばかりではないとい。ゲームクリエイターを目指して専門学校に行く人は、課題研究としてではなく、趣味として、一度個人でのゲーム制作を経験してみるのも良いかもしれないと思った。

### 4. 参考文献

衝突判定のあれこれ

[https://qiita.com/moscoara\\_nico/items/8ee1de552601a8a8f1f](https://qiita.com/moscoara_nico/items/8ee1de552601a8a8f1f)

マルチタッチについて

<http://blacker1017.hatenablog.com/entry/2017/07/22/234642>

テラシュールブログ

<http://tsubakit1.hateblo.jp/>