

Unityによるシューティングゲーム の制作

西野 翔大 水田 紘渡

1. まえがき

私達が普段やっているゲームを作りたいと思い、“unity”を使用して制作した。

2. 原 理

今回私達が使用した unity とは無料で使う事が出来、かつプログラム初心者でも簡単にゲームが作れるゲームエンジンである。

また、私達は他のゲームエンジン”unreal engine”を使用しようとしたが、学校のパソコンでは使用することが出来ず、学校でも使用できる unity に決めた。



図1 unity と unreal engine のロゴ

3. 研究内容

(1) 去年を振り返る

私達は去年も同じく unity を使用して 3D ゲームを作った。

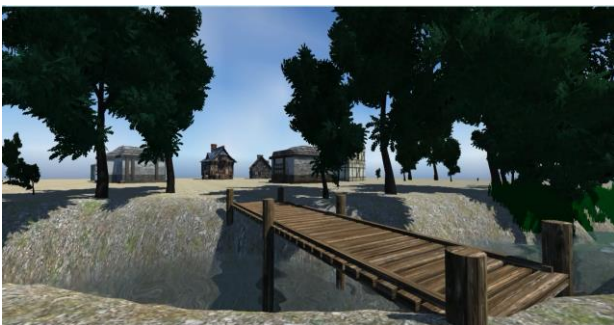


図2 去年作った、町の探索ゲーム

去年の私達は、unity を使うのが初めてだった為、勉強しながらこの探索ゲームを作った。

(2) どんなゲームを作るか考える

去年で大体の事を学んだ私達は去年よりクオリティの高い物を作ろうと話し合いをし、そこで出てきた案が FPS だった。

① FPS(英:First Person shooter、略称 FPS)

主にシューティングゲームの一種で、プレイヤー(主人公)の視点でゲーム中の世界・空間を任意で移動でき、武器などを用いて戦うゲームである。基本的に画面に表示されるのはプレイヤーキャラクターの一部(腕等)と武器・道具のみである。



図3 実際に販売されている FPS ゲーム (BATTLEFIELD V)

(3) キャラクターの実装

まず、自身が動かすキャラクターが居ないと何も始まらないので主人公を作ることにした。

私達が作ろうとしたのは、主人公の目線なのでモデルを作る必要が無かった。

また、unity の Standard Assets に FPS Controller(以下 FPSC)という物があった為それを使用し、武器となる銃を持たせた。



図4 FPS Controller と銃を統合化したもの

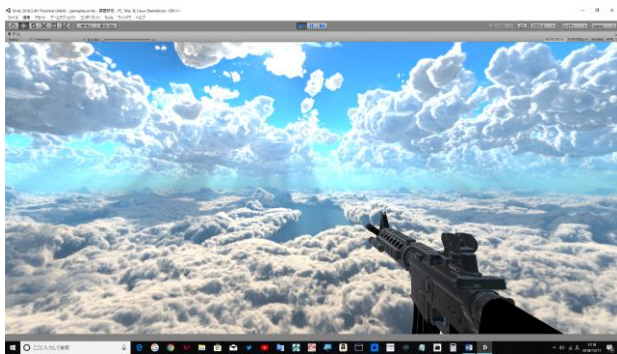


図5 図4を実際にプレイしてみた物

FPSCには遊ぶ為に必要な移動、ジャンプなどが最初からプログラムされていたのでFPSCを使い、主人公を完成させた。

(4) システムの作成

① ゲームに必要な物を作る前に

とりあえず動くキャラができたので、次はゲームに必要なステージとシステムを作る事にしたが、私達はここで一つ問題に気が付いた。

それは、当初の予定では図3のような対人のFPSを作るはずだったが、今の私達の技術では作る事が難しいと判断したので、急遽予定を変更し、対人FPSから、ホラーFPSを作る事にした。

② 作るシステム一覧

作る物が本格的に決まってきた為、必要なシステムを作ることにした。

主人公の移動はもうできているので、大まかに必要なのは「敵の作成」「銃関連」「ステージ作成」「ライト」となった。

③ 銃関連

主人公周りを片付けようとまずは「銃関連」に手を付けた。

銃を撃つためには、「弾」、「弾が出てくる場所」、「撃っている時の音」、「残弾数」が必要なので、スクリプトを作成した。

弾、音はunityが無料で出しているストアからダウンロードした。また弾が出る場所はunityのクリエイトからボックスを選択し、

銃口部分に取り付けた。

```

6 public class Amd : MonoBehaviour
7 {
8     public Amd()
9     {
10    }
11 }
12 public void Shooting()
13 {
14 }
15 }
16 // bullet prefab
17 public GameObject bullet;
18
19 // 弾丸発射点
20 public Transform muzzle;
21 //public GameObject effect;
22
23
24 // 弾丸の速度
25 public float speed = 1000;
26 //弾の所持数
27 public int shotCount;
28
29 public Text shellLabel;
30
31 private AudioSource sound01;//銃撃音
32 private AudioSource sound02;//リロード音
33 private AudioSource sound03;//弾が切れたとき
34

```

図6 銃を撃つ為に必要な物の宣言

④ 弾の発射とリロード

銃、弾を作り音も出せるようになったので、一番肝心の発射を作ることにした。

PCでFPSを遊んでいると大体のゲームが左クリックで覗き込み、右クリックで発射が大半なので私達もそれに倣うことにした。

```

45 // Update is called once per frame
46 void Update()
47 {
48     if (Input.GetMouseButton(0))
49     {
50     }
51 }
52 //残弾数が1未満の時は何もせずリターン
53 if (shotCount < 1)
54 {
55     if (Input.GetMouseButton(0))
56     sound02.PlayOneShot(sound02.clip);
57     return;
58 }
59 else if (shotCount > 1)
60 {
61     if (Input.GetMouseButton(0))
62     sound01.PlayOneShot(sound01.clip);
63 }
64 if (shotCount < 1)
65     return;
66 shotCount -= 1;
67 shellLabel.text = shotCount + "/" + shotCount;
68 }
69 }
70 }
71 // 弾丸の複製

```

図7 弾の発射コード

```

69
70
71
72 // 弾丸の複製
73 GameObject bullets = Instantiate(bullet) as GameObject;
74
75 Vector3 force;
76
77 force = this.gameObject.transform.forward * speed;
78
79 // Rigidbodyに力を加えて発射
80 bullets.GetComponent<Rigidbody>().AddForce(force);
81
82 // 弾丸の位置を調整
83 bullets.transform.position = muzzle.position;
84 /*Instantiate(effect, muzzle.position, Quaternion.identity);
85 effect.transform.position = muzzle.position;
86 */
87
88 //Rリロード
89 if (Input.GetKeyDown(KeyCode.R))
90 {
91     if (shotCount == 30)
92     {
93         return;
94     }
95     else if (shotCount < 30)
96     {
97         sound03.PlayOneShot(sound03.clip);
98         shotCount = 30;
99         shellLabel.text = shotCount + "/30";
100
101

```

図 8 リロードコード

当初階層は一階だけの予定だったが、それだとすぐにゲームが終わってしまう為二階層にし、一階部分を迷路にして中々ゲームが終わらないようにした。

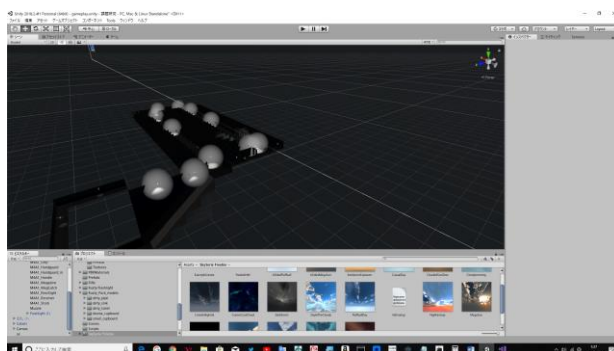


図 10 作成したステージ(二階部分)

⑤ 敵の作成

敵が出てこないFPSを制作してもよいが、それだと計画していた「ホラーFPSを作る」ではなくなってしまうので、敵を作ることにした。と言っても今回は敵の攻撃モーションは付けず「主人公に触れるとダメージを受ける」というふうにした。

敵のモデルには、unity が無料で出している「ADM」というロボットを使用した。

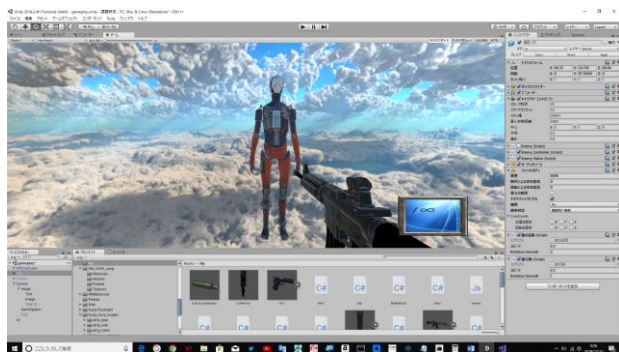


図 9 実際に配置した ADM

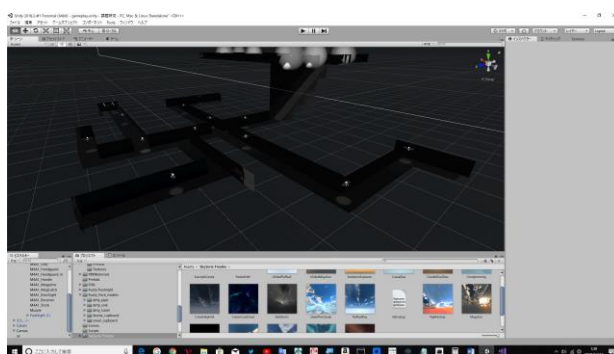


図 11 作成したステージ(一階部分)

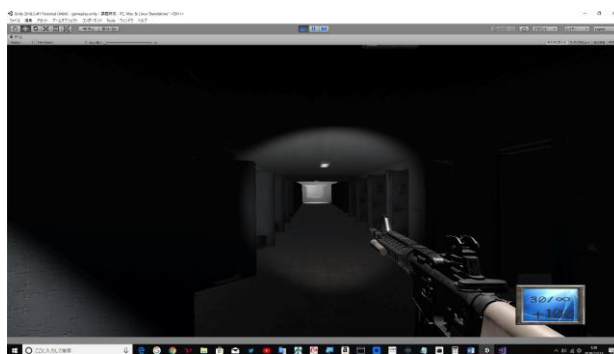


図 12 実際のプレイ画面

⑥ ステージ作成

今回作るゲームはホラーゲームという事なので空を暗くし、研究所のような施設から脱出するというテーマでステージを作成した。

最初は明かりを使用せず暗闇の中を進むようにしたかったが、そうすると暗すぎて方向が分からなかった為蛍光灯を追加した。また脱出口までの蛍光灯は数少ないため、明かりを絶たせない為に銃身にライトを取り付けた。

⑦ 懐中電灯

実際にゲームをやってみて思った事は光が少ないので見えづらいという事だった。したがって、蛍光灯とはまた別に手動でスイッチの切り替えができる懐中電灯を作った。ゲーム中に「C」を押すと、光の切り替えができるようになった。

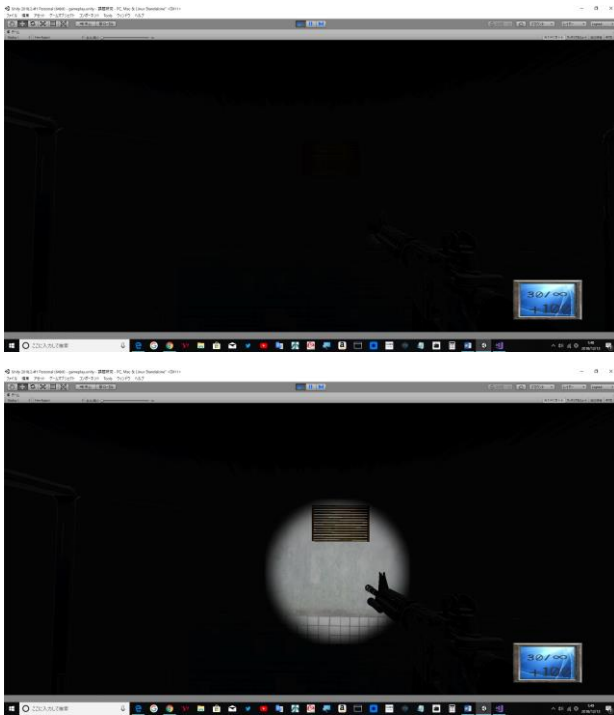


図 13 ライトの OFF と ON の比較画像

4. まとめ

(1)成果

約二年間 unity を使用してゲーム作成をしてきたが、unity 側にバグがあったり、言語が英語なのでエラーが発生しても何がダメなのかわからなかったりして途中何度もやめたくなったが、ゲームを作るという事が楽しかったので、少しずつ起きている問題を解決し完成させることができた。今回作成したものは、所々断念しないといけない箇所が出てきたが、最初に考えていた「FPS を作る」という目標は達成することができたので満足している。

(2)反省

今回の課題研究では一つの大きな目標を立て、それを終わらせる形で開発を進めていたので、エラーが起きたら今まで作ってきた物を全て消し、一から作る事をしてきた。今まで必要が無いと思っていた英語が初めて必要になったので英語を勉強すべきだと激しく後悔した。

(3)改善

12月現在で追加すべき項目や解決しないといけないバグなど多々存在している。今は発表する為の資料を作成しているが、発表日までには資料を完成させ、バグを直すなどの改善をしようと思

う。

5. あとがき

私は去年に引き続き、unity でのゲーム制作に取り組んだ。去年は unity を使うのが初めてで予想完成図とはかけ離れた物ができたが、先ほどにも書いたように、断念せざるを得ない箇所が出てきたが、どんな形であれ「FPS を作る」という目標は達成することができた。

この二年間開発を続けてきたが一番強く残っているのは、「ゲーム制作は楽しい」という事だ。自身が創造した物がそのままプレイできるという喜びはないと思った。

これからも時間が空いていけば少しずつではあるが、開発を続けていこうと思う。(西野)

私は去年に引き続き、unity でのゲーム制作に取り組んだ。課題研究がスタートして一つ一つのプログラムにとっても時間がかかり、私たちが制作したいものができるか不安だったが制作することができてよかったと思う。

今後制作機会があればもっと時間をかけていいものを作りたい。(水田)

6. 参考文献

・ Code Genius

<https://codegenius.org/>

・ Unity を使った 3D ゲームの作り方(かめくめ)

<https://gametukurikata.com/>

・ increment Log

<https://increment-log.com/unity-sound-se-play/>