

# 2Dシューティングゲームの制作

安岐 龍也 高原 康生  
小野田 晴仁

## 1. 研究概要

ゲームエンジンとして代表的である unity を使用し、シューティングゲームの制作を行う。ゲーム制作を通してプログラム言語である C#や C++の特性や言語構造などを学びながら身近にあるゲームがどのように作られているかを知る。

## 2. 研究の具体的内容

### (1)unity について

「unity」は、unity Technologies が 2004 年に開発したゲームエンジンである(図 1)。ゲームエンジンとは、簡単に言えば 3D の計算や影の表示、サウンド、メニュー遷移など、ゲーム作りによく使う機能を 1 つにまとめて使いやすくしたものである。ウェブプラグイン、デスクトッププラットフォーム、ゲーム機、携帯機器向けのコンピューターゲームを開発するために用いられるため、全世界で 100 万人以上の開発者が利用している。最近では、PlayStation4 や Xbox、VR/AR などの開発にも対応している。



図 1 unity のロゴ

### (2)2D シューティングゲームについて

unity の 2D モードを使用し、プレイヤーを操作しながら出現する敵を倒していく 2D シューティングゲームを制作した(図 2・3)。



図 2 2D モードで制作したシューティングゲーム

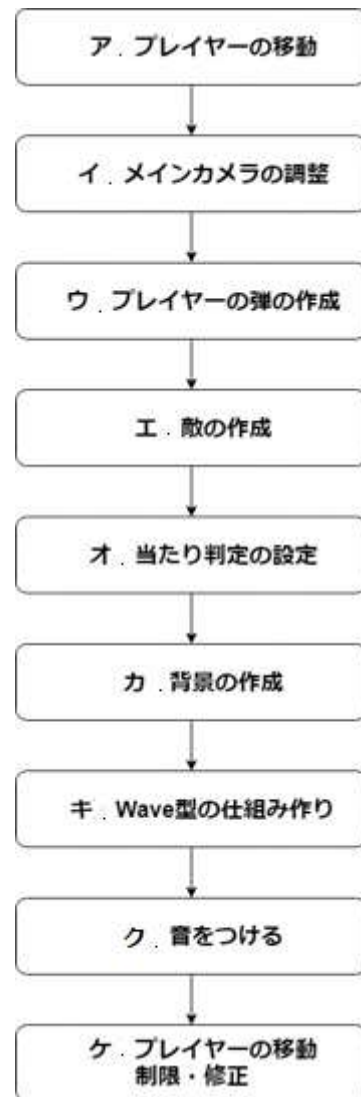
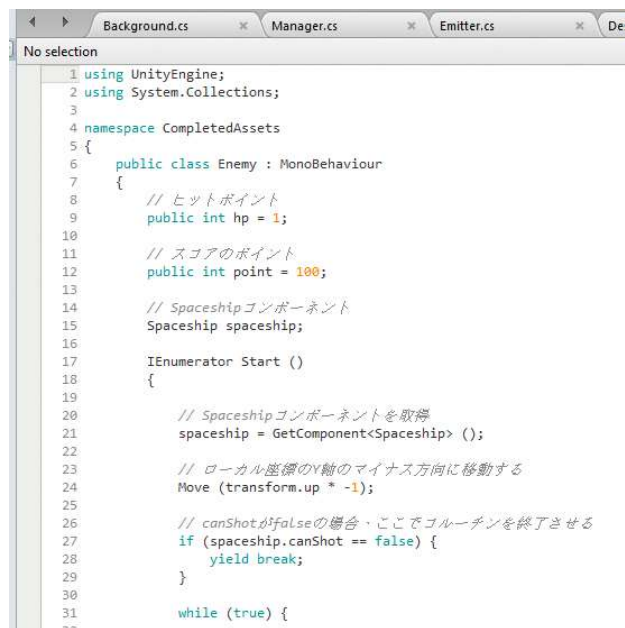


図 3 制作過程

プログラムの作成には MonoDevelop というソフトを使用した。MonoDevelop は対応する環境であれば OS を問わず動作させることが出来る。言語は C# という C 言語に改良や制限を加えた言語を使用した (図 4)。Visual Studio を使用してもプログラムを作成することは出来るが起動が遅いため今回は MonoDevelop を使用した。



```
1 using UnityEngine;
2 using System.Collections;
3
4 namespace CompletedAssets
5 {
6     public class Enemy : MonoBehaviour
7     {
8         // ヒットポイント
9         public int hp = 1;
10
11         // スコアのポイント
12         public int point = 100;
13
14         // Spaceship コンポーネント
15         Spaceship spaceship;
16
17         IEnumerator Start ()
18         {
19
20             // Spaceship コンポーネントを取得
21             spaceship = GetComponent<Spaceship> ();
22
23             // ローカル座標のY軸のマイナス方向に移動する
24             Move (transform.up * -1);
25
26             // canShotがfalseの場合、ここでルーチンを終了させる
27             if (spaceship.canShot == false) {
28                 yield break;
29             }
30
31             while (true) {
```

図 4 プログラミング画面

### (3) 制作過程

#### ア. プレイヤーの移動

プレイヤーの移動は矢印キーで行い、キー入力によってプレイヤーの移動する向きと速度を検出している。

#### イ. メインカメラの調整

メインカメラとはゲーム画面の全体を映すものである。メインカメラを調整することでカメラがゲームの範囲外を映すことを防ぐことが出来る。

#### ウ. プレイヤーの弾の作成

弾の画像に弾を動かすためのプログラムを挿入する。作った画像の子要素として空のゲームオブジェクトを作成するとそのオブジェクトの位置から弾が出るようになる (図 5)。



図 5 プレイヤーの弾

#### エ. 敵の作成

敵は複数の異なる位置や角度で弾を撃つことが出来るようにするため、軸を 20 度ずつ傾けることで様々な方向へ弾を発射出来るようにした。

#### オ. 当たり判定の設定

当たり判定は敵とプレイヤーと弾に設定した。プレイヤーの当たり判定は中心に設定することで範囲を小さくし、難易度を簡単にした。

#### カ. 背景の作成

今回は奥行きを作るために 2 枚の画像を使用した。それぞれの画像が自動でスクロールされるようにするために 2 枚の画像の移動速度を変えた。

#### キ. Wave 型の仕組み作り

Wave とは敵をまとめて前進させるための仕組みであり、座標を使用して敵の配置を決める (図 6)。

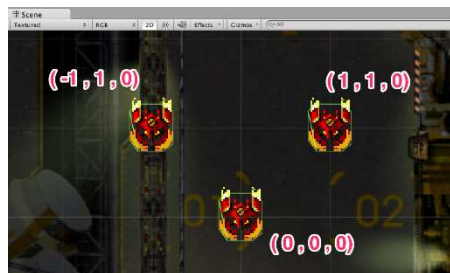


図 6 Wave 型の仕組み

#### ク. 音をつける

このシューティングゲームには BGM とショット音、敵を倒したときの爆発音をつけた。

#### ケ. プレイヤーの移動制限・修正

プレイヤーが画面外に移動できないようにするために、座標をもとにプレイヤーが移動できる範囲を制限した。

#### (4) RPG ゲームについて

2Dシューティングゲームとは別にRPGゲームも制作した。今回制作したRPGにはバトル機能やストーリー機能が搭載されており、やり込み要素が含まれている(図8)。



図8 RPGゲームの画面

#### (6) 制作過程

RPGを製作にあたって、処理を以下のように分けた。

##### ア. ステージの作成

##### イ. キャラクターの作成

##### ウ. カメラワークの調整

##### エ. シーンの設定

##### ア. ステージの作成

今回のステージ制作では teratain という3Dオブジェクトを使った。このオブジェクトは地形の製作を容易に行うことが出来る上、ごく自然な風景を演出することが出来るため使用した(図9)。

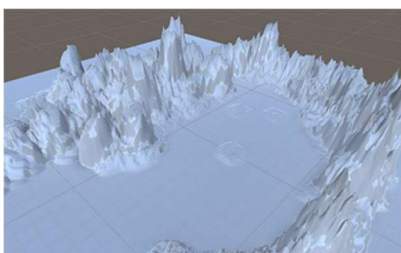


図9 ステージ作成

##### イ. キャラクターの作成

キャラクターの作成では主人公であるペンギンと敵であるトラを使った。主人公が移動する際、移動する方向を前にすることで自然な移動を再現することが出来た(図10)。



図10 キャラクター

##### ウ. カメラワークの調整

今回はプレイヤー用カメラとマップ用カメラの2つを用意した。これらはキーボードの「1」を押すことで切り替えることが出来る(図11)。

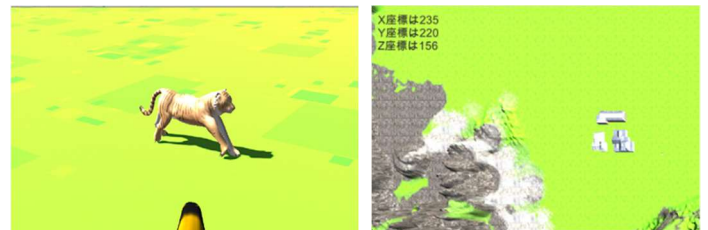


図11 プレイヤーカメラとマップカメラ

##### エ. シーンの設定

今回はタイトル画面とプレイ画面、はじめの町の画面である3つのシーンを作った。シーン制作の際にはプログラムを変更しなければならなかったのが難しかった。

##### オ. 音の作成

音はBGMと効果音の二つを作った。今回は audiosorce というオブジェクトを使った。このオブジェクトはスクリプトからも操作することが出来るので、使い勝手が良いので使用した(図12)。今回はバトルシーン、プレイシーン、始まりの町シーン、タイトルシーンで音を分けた。

音楽をそれぞれのシーンに合わせるのはとても難しかった。



図 12  
audiosource の設定

### 3. 研究のまとめ

私たちはゲーム作りの知識が全くないままテーマを決めてしまったため、最初のテーマとなっていた人生ゲームの制作を途中で断念してしまっただけでなく、そのためゲーム作りの知識をつけるために書籍やサイトなどを参考に基礎を身につけていった。その知識を用いて 2D シューティングゲームと RPG ゲームの制作に取り掛かった。サイトなどを参考にしたためバージョンが異なるなどつまづくことも多々あったが、スクリプト通りに動作した時はとても感動した。この課題研究で得た経験と知識をこれからも活かしていきたいと思った。

#### 参考文献

unity 2D シューティング

<https://unity3d.com/jp/learn/tutorials/projects/2d-shooting-game-jp>

2D シューティングゲーム作ってみた

<http://kto.hateblo.jp/entry/2018/02/06/233248>

シーン間のスクリプト内の変数の共有

<https://qiita.com/YuwUnknown/items/afd9c9cd01de7c559ba60>

Unity 使いそうな関数まとめメモ

[https://qiita.com/hiroyuki\\_hon/items/4c3bd290c795b2481ebd](https://qiita.com/hiroyuki_hon/items/4c3bd290c795b2481ebd)

Unity キー入力まとめ

<https://tech.pjin.jp/blog/2015/09/30/unity>

COREVALE コアベイル ただいまゲーム制作中

<http://corevale.com/unity/6944>

Unity で rigidbody の位置, 回転の固定をスクリプトから変更する

<https://loumo.jp/wp/archive/20131210003026/>

【Unity 入門】全方位シューティングを作る

<http://babas.hatenablog.com/entry/2018/04/01/190000>

Unity で敵キャラが見回りをするように動き回るプログラム - Unity を使った 3D ゲームの作り方 (かめくめ)

<https://gametukurikata.com/program/move-around-destination>

いますぐ使える手軽に使える無料音楽素材

<http://amachamusic.chagasi.com/>

【Unity 入門】Destroy を完全攻略! 初心者必見の便利テクも一覧まとめ

<https://www.sejuku.net/blog/53555>

【Unity】特定のゲームオブジェクトやコンポーネント、スクリプトにアクセスする

<http://albatrus.com/main/unity/6276>