

# Gamemaker を使用した 2D アクションゲームの制作

西江 勇真 浅野 遥哉  
糸原 悠登

## 1. 研究概要

研究課題は、Gamemaker を使用して、「I wanna be the guy」(以下、アイワナ)というゲームの二次創作作品を作ることにした。

アイワナとはプレイヤーキャラが 5mm ほどの大きさで、オブジェクトが予測不可能なタイミングで襲いかかり、プレイヤーは一度でもオブジェクトに当たると、ゲームオーバーになってしまう。一般的に「鬼畜 2D アクションゲーム」というジャンルで長年ユーザから愛され、同時に、ユーザを苦しめてきたゲームである。そしてゲームをチームで制作することの楽しさと苦悩を、課題研究を通して学びたいと思い、制作した。

## 2. 研究の具体的内容

### (1) ゲームの設計

アイワナを作るにあたって、主人公の旅をする目的を決めた。何を指すのか、何になりたいのか、目的を達成するには何が必要なのか。我々は情報技術科に所属しているので、情報科らしさを出していきたい、そう考えた結果、「目的はシステムエンジニアになること、そのためにはステージのボスを倒してパソコンのパーツを手に入れ、主人公はシステムエンジニアになる。」という設定にした。

ステージの数は少なすぎず多すぎない数にしようと思い、3 ステージにした。Gamemaker のプログラミングは基本的にオブジェクト思考型なので、自分で見たままにオブジェクトを配置してステージを作ることが可能であるが、いきなり配置するのではなく、紙にベースとなるステージ案を描いて、プログラム担当と相談しながら作った。紙に一度描くこと

により実際に配置したときに、このオブジェクト配置で、クリア可能か、ここでのトラップ配置は適格な位置にあり回避が可能かなどがわかりやすく確認できるので、プログラムもためらうことなく制作することができた。

図 1 はステージ 1 の 4 面(以下 0-4)のステージ案である。

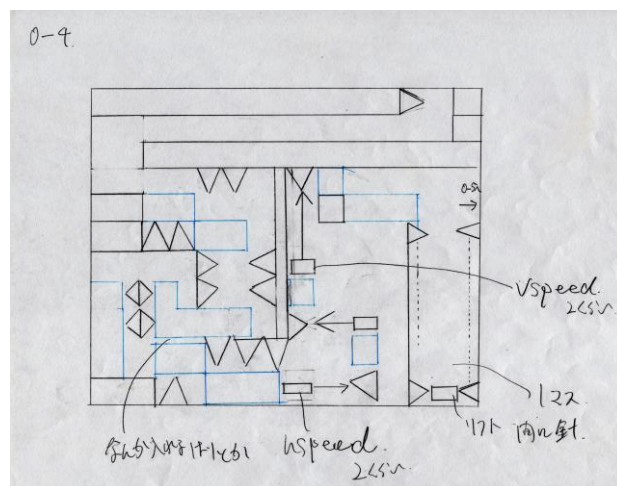


図 1 0-4 ステージ案

リフトに書かれている vspeed は vertical speed (縦方向)、hspeed は horizontal speed (横方向) に動くオブジェクトである。

例えば、vspeed=2 とすると、下方方向に 2 pixel/step (1 step=1/50 秒) 動く。hspeed=2 とすると、右方向に 2 pixel/step 動く。この時の指定した値に-をつけると vspeed の場合は上方方向に動き、hspeed の場合は左方向に動く。

尖ったオブジェクトはプレイヤーが触れると GAME OVER になってしまう針オブジェクトである。

薄く描かれているのは水オブジェクトで、プレイヤーが触れると、無限ジャンプができ、落下速度が遅くなる。2 段ジャンプが回復す

るという状態を得られる。

図1のステージ案をGamedMakerで配置したのが図2である。

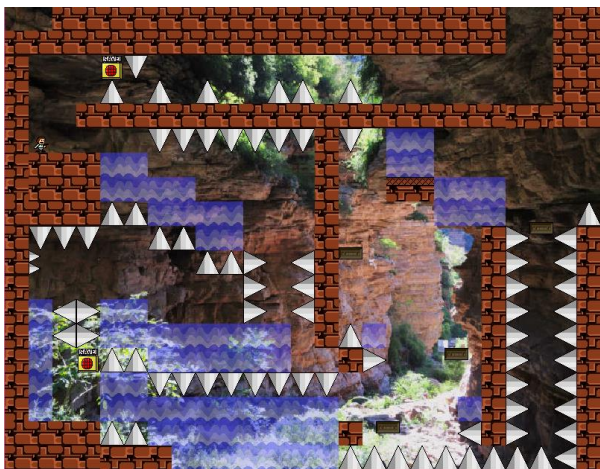


図2 0-4 プレイ画面

実際に配置をしてみると、針オブジェクトの数や、水オブジェクトの数など、仕様書と異なるところもあるがほぼ完璧に配置をすることができた。

図3に示すのはステージ1のボス面(仮称0-Boss)の行動案である。

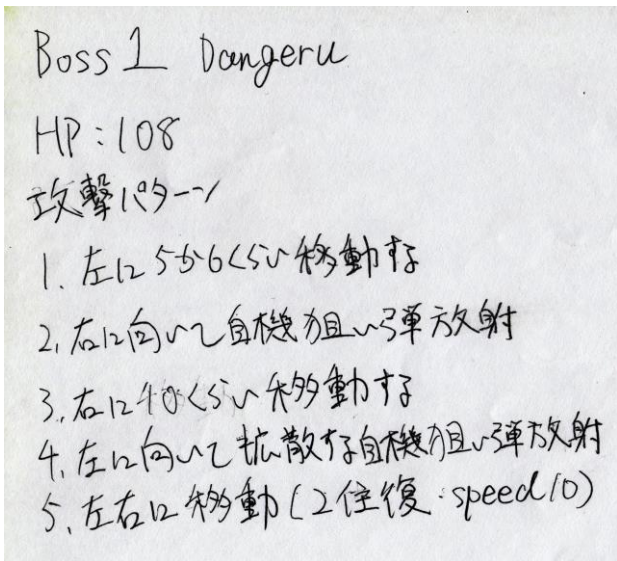


図3 ステージ1ボス行動パターン案

ステージ1のボスの体力(HP)は108。余談であるが除夜の鐘の数にしている。

自機狙い弾とは、プレイヤーに向かってくるオブジェクトで、このオブジェクトに触れてしまうとGAME OVERになってしまう。この自機狙い弾はプログラム項で記述している

TimeLine コマンドを使うことにより、設定できる。

図3の1~5に書かれている行動パターンはTimeLine コマンドを使用すると自動的にループするようになっており、ボスキャラクタ本体の体力が0になるまで実行を繰り返す。

図4に示すのはステージ2の7面-1(以下T-7-1)のステージ案である。

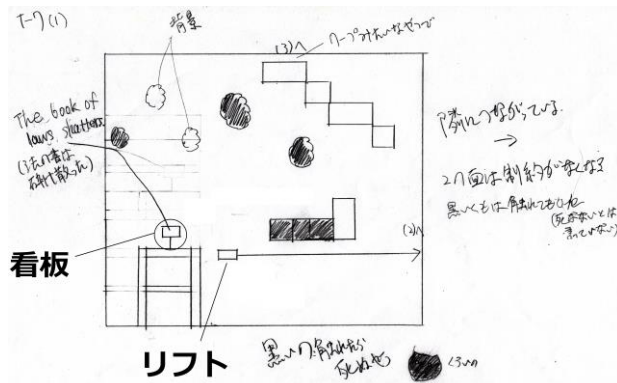


図4 T-7-1ステージ案

図4の面はT-7という1つのステージがあり、かなり広がっているため、3つに分割し、それぞれをT-7-1、T-7-2、T-7-3とした。

図4の左真ん中あたりにある看板オブジェクトには、指定した文字を表示する。

黒く塗り潰されたオブジェクトはプレイヤーが触れるとGAME OVERになってしまうオブジェクトである。

ステージ2には、リフトオブジェクトに触れてはいけない。黒い雲に触れてはいけないなど、制約というシステムがある。

このT-7-1からはその制約システムは存在せず、リフトオブジェクトに触れてもGAME OVERになることはない。T-7は横長になっており、リフトに乗って右へいくと次のエリアに移動することができる。

図4をGamedMakerで配置したのが図5になる。

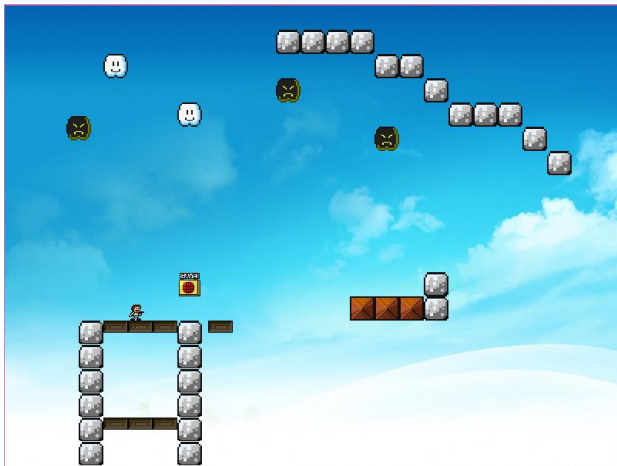


図5 T-7-1 プレイ画面

図5のステージ配置はオブジェクト間隔を広くとっているため、オブジェクト配置を忠実に再現できた。

### (2)グラフィック

ゲーム内で動かすキャラクタ、キャラクタの歩く地形、罨としてのオブジェクト、ボス等の敵キャラクタの描画を行った。図6に示すのは「スプライト」の編集画面である。

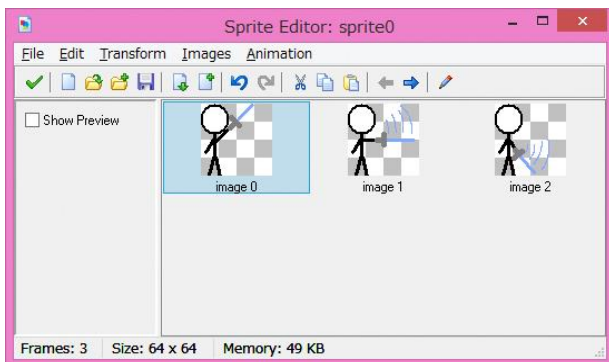


図6 スプライト編集

チェック柄のタイルが例えるならばマンガの一コマであり、これを「イメージ」と言い、この集合が「スプライト」である。ゲーム画面上では「イメージ」が一コマずつ送られていくのだが、このスプライト編集では、「イメージ」の追加や、「イメージ」の大きさの変更や変形を「スプライト」単位でまとめて編集できる。

「イメージ」の描画には Gamemaker に内蔵されたペイントツールを使用した。図7に示すのは「イメージ」の編集画面である。

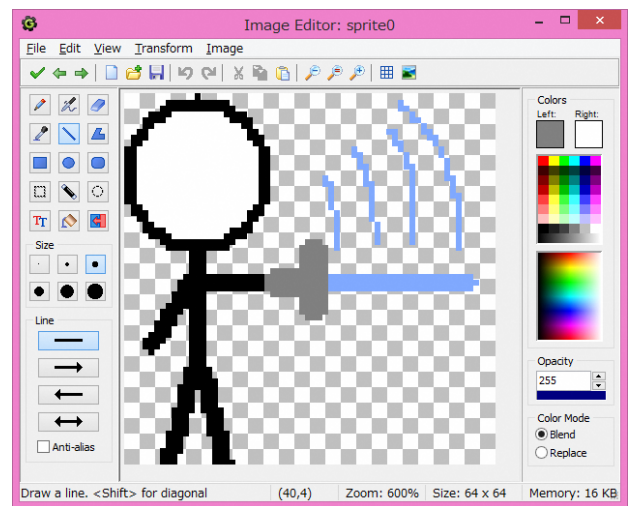


図7 イメージ編集

Windows 8 標準のペイントツールと比較すると、変形、効果といった機能が充実しているため、ゲームグラフィックの描画に適したものとなっている。

1 から描画していただくだけでなく、既存の素材のアレンジや、素材の再利用等、作業を効率的に行うことを心掛けた。例えば、アイワナの主人公の衣装は青地の服に赤いマントを羽織った小人のような容姿をしているが、これを岡工の作業服に変えて、独自感を出しつつも1からキャラクタを作る手間を省くことができた。

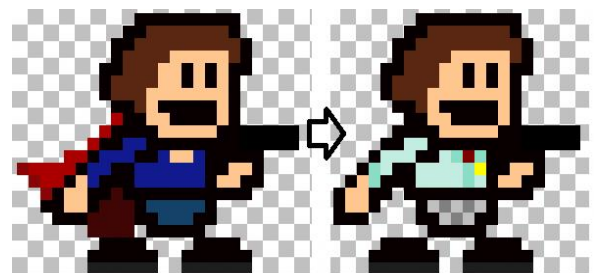


図8 アレンジ前→アレンジ後。

### (3)プログラム

作成したキャラクタや地形を利用し、原案をもとに実際に動く画面を作成した。一画面が完成する度に、本当にクリアできるのか、難易度は高すぎないかを調整しながら、ボスやトラップなどの仕掛けを動かすプログラミングを行った。

Gamemaker のプログラミング言語は独自の

GML という言語で、C#を参考にして作られた言語である。そのため、初めは右も左もわからない状態だったが、少し慣れれば、かなりわかりやすいコードを書くこともできるようになった。50分の1秒単位でオブジェクトを動かすプログラムを組むことができるため、調整はとても面倒なものだが思い通りにオブジェクトを動かすこともできた。

プログラムは、プログラムを実行するタイミングを指定するイベントというものを決め、コードを記入していく。イベントには、オブジェクトが生成されたときに実行する” Create” や、50分の1秒ごとに実行する” Step”、指定したオブジェクトが触れた時に実行する” Collision”、実行する時間を指定する” Alarm” などがある。そのため複雑なプログラムもイベントを利用して簡単にコードを組むことができる。

また、オブジェクトに直接コードを書き込むだけでなく、アクションコマンドを使えば、コードを記入せずにオブジェクトの行動を指定することができる。ただし、アクションコマンドでは、できることの制限があるため、複雑な動きの場合はコードを記入したほうが良い。

ほかにも、TimeLine と呼ばれる、” Alarm” イベントのサブルーチンを作ることができ、” Alarm” イベントよりも多くコードを記入することができる。ボスの行動などもこのTimeLineを利用してコーディングした。

以下に、プログラムの例を示す。

- ・ゲーム内に登場するマスコットキャラクターのプログラム

イベント:Step

コード

```
1 if (vspeed==0) {
2     if (sparkinglemon1.triggered) {
3         vspeed=40;
4         sound_play(sndkusso);}}
```

オブジェクトの動き

プレイヤーが目には見えない「trigger」という、いわゆるセンサーの役割をするオブジェクトに触れると、上からマスコットキャラクターを下に動かす。

解説

- 1:vspeedが0だと {} 内を実行する。
- 2:sparkinglemon1という名前の trigger にプレイヤーが接触したら {} 内を実行する。
- 3:オブジェクトの vspeed の値を 40 にする。
- 4:sndkusso という効果音を 1 回再生する。

- ・ステージ内に出てくる一定の場所までプレイヤーを追跡する敵キャラ

イベント:Create

コード

```
1 timeline_index=ring0;
2 timeline_position=0;
3 timeline_running=false;
4 timeline_loop=true;
```

イベント:Step

コード

```
1 if(deathtrigger.triggered) {
2     timeline_running=true;
3     sprite_index=chara_cloud;
4     image_speed=0.8;
5 }
6 if(deathdeathtrigger.triggered) {
7     instance_destroy();
8 }
```

オブジェクトの動き

オブジェクトが画面に生成されたら、オブジェクトに Timeline を設定する。その後、「trigger」にプレイヤーが接触すると、Timeline を実行する。

※Timeline にプレイヤーを追跡するプログラムをしてある。

解説

Create

- 1:rinG0 とい名前の Timeline をオブジェクトに設定。
- 2:Timeline を最初から実行するように設定。
- 3:Timeline を実行していない状態にする。
- 4:Timeline を繰り返し実行するように設定。

#### Step

- 1:deathtrigger という名前の trigger にプレイヤーが接触すると {} 内を実行する。
- 2:設定した Timeline を実行させる。
- 3:オブジェクトのSpriteを chara\_cloud に変更する。
- 4:Spriteのコマ送りの速度を 0.8 枚/Step に設定する。
- 6:deathdeathtrigger という名前の trigger にプレイヤーが接触すると {} 内を実行する。
- 7:オブジェクトを消滅させる。

### 3. 研究のまとめ

#### (1) 完成作品

今回作成したゲームを簡単に画像とともに紹介する。

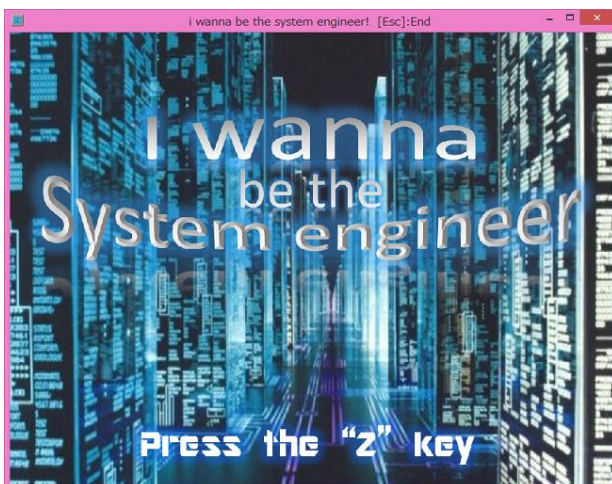


図9 タイトル画面

完成したゲームのタイトル画面である。こ

の画面でZキーを押すと次の画面へ進み、ゲームを開始する。



図10 プレイ画面

ステージ2の最終面で、ボス戦手前のステージである。上に進むとボス戦へ突入する。ボス戦は敵の攻撃を回避しながら、Zキーで打てる球を当て、撃破するというものである。

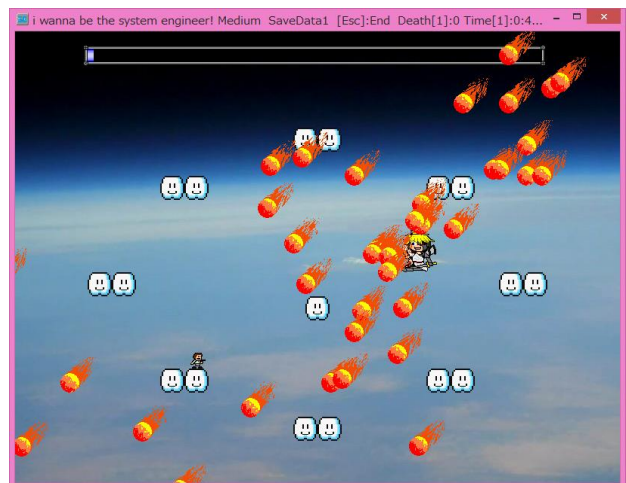


図11 ボス戦

今回作成したゲームのボスの攻撃の中で、最も派手な攻撃であり、回避できないように大量にオブジェクトを生み出している。



図 12 ゲームオーバー画面

数あるゲームの中でも、特徴の濃いゲームオーバー画面で、オブジェクトに当たると血とともにプレイヤーがはじけ飛ぶ、というものである。

## (2) 感想

(西江)

1年間の課題研究を通して、あまり触れてこなかったオブジェクト指向型プログラム言語に、かなり触れることができた。慣れてないうちは仕様書どおりにステージを作ることができなかつたり、うまくコーディングができず、様々なエラーと戦ったりした。たくさんさんのサイトを参考にし、最初はコピーするだけの作業だったが、徐々に自分でコードを考えられるようになり、後半からの課題研究は本当に楽しいものだった。最後までプログラムを組むことができ、楽しみながら自分を成長させることができた。

進学先では今とはまた違う環境でゲームを作ることになるため、今回で学習したことを参考にし、楽しくゲーム作りをしようと思う。

(浅野)

中学のころからこの Gamemaker を使って、アイワナというゲームを作ろうと思っていた。しかし、一人ではまだ使い方がいまいちわからずに企画倒れに終わっていたので、課題研究の研究課題として、ゲームを作る、アイワナを作るという一つの目標を達成することが

できてとても良い機会だった。

また、企画、グラフィック、プログラムと3つの班で分担して作業を行うことで一つの作業に集中でき、一人で作るよりも、様々なアイデアが生まれて楽しく作品制作をすることができた。将来ゲームを作る職に就くので、この課題研究の成果を一つの経験として、活かしていけたらよいと思った。

(糸原)

昔から自分の手でゲームを作りたいと思っていたので今回の課題研究は非常にいい機会になった。また、グラフィックの面、その中でもドット絵には特に興味があったのでより良い経験になった。

今回の課題研究を通して少しとは言えど実際に仕事としてゲームを作っている人の気分になれた。また、他の二人との意思疎通や計画など、普段以上に付き合う時間が長くなり、コミュニケーションの技術を改めて養うことができた。自分自身、将来そういったゲームを作る仕事に就きたいと思っているので、この感覚を忘れずに確実に将来へと繋げていきたい。

## 参考文献

Gamemaker まとめ Wiki

<http://www.wikihouse.com/GameMaker/index.php?Game%20Maker%20A4%DE%A4%C8%A4%E1Wiki>

Rightnya の趣味の広場

<http://rightnya.web.fc2.com/index.html>

アイワナ製作@基本・応用 wiki

<http://seesaawiki.jp/w/forusian/>