

令和6年度卒業研究報告書

産技短を舞台とした2DRPGの制作

情報技術科 齊藤 姿月

鈴木 悠斗

武田 怜士

中村 光希

指導教員 菅原 康雄

目次

令和6年度卒業研究報告書	0
第1章 はじめに	4
第2章 研究概要	5
2.1 開発環境	5
2.2 GMS2	5
2.3 アセットブラウザ	6
2.4 イベント	9
2.5 Delphi	13
2.6 Pascal	13
2.7 C++	14
第3章 ゲーム構成	15
3.1 ゲーム企画	15
3.2 操作方法	15
3.3 キャラクター	16
3.3.1 主人公キャラクターの経緯	16
3.3.2 情報技術科の先生を動物化したキャラクター	16
3.3.3 キャラクターの作成方法	17
3.3.4 キャラクターの動作方法	19
3.4 マップの作成	21
3.4.1 マップの重要性	21
3.4.2 マップの構造	21

3.4.3 タイルマップの作成.....	21
3.4.4 ロケーション・ハンティング.....	22
3.4.5 オブジェクトの作成.....	24
3.4.6 マップの壁の作成方法.....	25
3.4.7 マップの移動遷移.....	25
3.5 マップのスクリプト.....	26
3.5.1 イベントトリガー.....	27
3.5.2 パスファインディング.....	27
3.5.3 ダイナミックなマップ変更.....	27
第4章 ストーリー.....	28
4.1 ストーリー概要.....	28
4.2 敵キャラクターの設定.....	28
4.3 ボスキャラクターと最終目標.....	29
第5章 戦闘画面.....	30
5.1 たぬき（石館教授）.....	30
5.2 パンダ（菅野先生）.....	32
5.3 キツネ（飯坂准教授）.....	33
5.4 ネコ（強先生）.....	34
5.5 ライオン（安倍先生）.....	36
5.6 フクロウ（菅原先生）.....	37
5.7 ウサギ（ソソラ先生）.....	39
5.8 生徒会長（ラスボスのクマ）.....	40
第6章 プレイヤーへの配慮.....	41
6.1 マップ閲覧機能.....	41

6.2 マップ閲覧機能.....	41
6.3 フラグ管理.....	42
6.4 ルートの指示.....	43
第7章 環境構築.....	44
第8章 ゲーム環境.....	46
第9章 おわりに.....	47
第10章 参考文献.....	48

第1章　はじめに

本校では、これまでの卒業研究において数多くのゲームが開発されてきた。しかし、オセロなどのボードゲームの制作はあったものの、産技短を舞台とした作品は少なくゲームを通じた学校の認知度向上にはあまり寄与していないのが現状である。これを踏まえ私たちは、どの年齢層でも気軽に産技短を探索しながら、ゲームが出来る 2DRPG を制作しようと考えた。

そのために、GameMaker Studio2(以下 GMS2 と略す)という 2D の制作に特化した開発ツールを使用する。また、「Undertale」を参考に、戦闘画面やストーリーを展開している。この研究を通して、社会人でも生かせるようチーム開発の経験を蓄積し、今後の開発スキル向上にも寄与することを目的とする。

第2章 研究概要

2.1 開発環境

本研究の開発環境を以下に示す。

表1 開発環境

OS	Windows10,11
開発環境	GMS2
言語	GML(オリジナル言語)

2.2 GMS2

GMS2 とは YoYo Games が開発したマルチプラットフォーム対応の 2D ゲームエンジンであり、イラストやアニメーションを作成するためのエディタが多数存在している。世界的に有名なゲームエンジンである、「Unity」や「Unreal Engine」のに次ぐ知名度がある。GMS2 では、Delphi、C++ のプログラミング言語を主にベースとして作成されている。GMS2 は初心者でも扱いやすいドラッグ&ドロップ機能と、上級者向けのスクリプト言語 GML を備えている。また、プログラミングが慣れてきた人は、JavaScript に似たオリジナル言語である GameMaker Language(以下 GML と略)でコードを書くこともできる。

GMS2 は、2021 年 7 月に完全無料化され、現在はすべての開発機能が使用可能となっている。主な開発例として、「Undertale」、「Hyper Light Drifter」、「VA-11 Hall-A」などがある。

2.3 アセットブラウザ

GMS2 にはアセットブラウザがありそれぞれ機能がある。詳細は以下の通りである。



図1 GMS2 の機能一覧

- ・スプライト

スプライトはゲーム内で表示される画像・アニメーションである。ゲームの礎となるイラストや画像、アニメーションを作成する。アニメーションでは、FPS 数値を設定することで、イラストや画像の切り替わる速さを自由に変更できる。

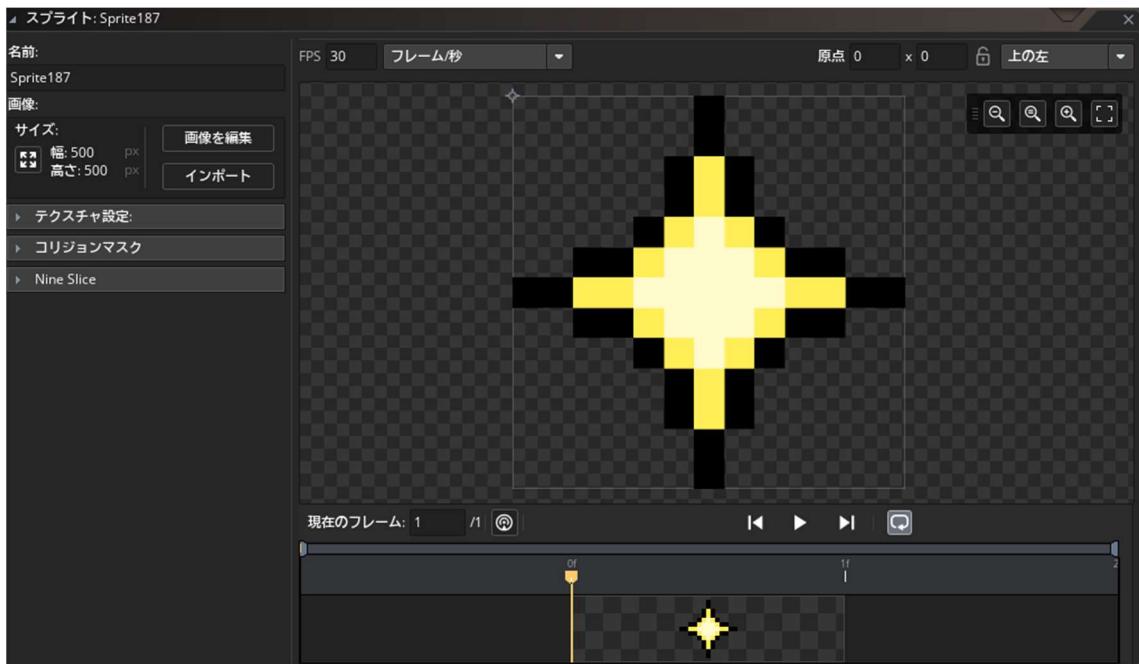


図2 スプライト

- ・オブジェクト

オブジェクトは作成したスプライトに魂を入れる機能である。主にゲーム内で動くキャラ・アイテム・敵などをプログラムで制御することができる。スプライトを設定し、イベントを設定して動きを作成する。

- ・ルーム

ルームはゲームのステージ・マップ・タイトル画面を作成する場所である。作成したオブジェクトをこのルームに配置する。

- ・パス

パスはオブジェクトを特定のルートに沿って動かすためのリソースである。例えば、敵キャラクターがマップ上をランダムに動く動作があるが、これを可能にしているパスという機能である。線を引きながら自由な動きを設定できるため、あたかもランダムで動いているように見える。

- ・スクリプト

スクリプトはGMLでのコードをまとめたものであり、プレイヤーの移動やAIの動きを制御することが可能、JavaScriptに類似したルールで記述可能である。

- ・サウンド

ゲーム内のBGMや効果音を管理するリソースである。MP3、WAV,OGGなどの音楽ファイルが使用できる。

- ・タイルセット

タイルセットは主にマップの基礎を作成するための画像である。タイルセットの画像は自身の作成したスプライトを使用することが出来る。また、オブジェクトと比較してメモリを圧縮しないため、ゲーム動作が軽くなる利点もある。

- ・タイムライン

特定の時間ごとにイベントを発生させる。例えば、「0フレーム、20フレームに光るアクションを追加する」となった際には、この機能を使用する。

- ・シェーダー

シェーダーは画面のエフェクト（主に影・光・色の変化）を制御するためのコードである。爆発のエフェクトもシェーダーを使用している。

2.4 イベント

GMS2 では「イベント」機能が実装されている。例を挙げると、「キーが押されたら動く」「当たったら消える」などの処理をプログラムで記述することが可能である。以下が各イベントの詳細について例を挙げながら説明する。



図3 イベント

・作成

作成では、オブジェクトが生成された瞬間に 1 回だけ実行されるイベントであり、主に変数の初期化やステータス設定などに使用する。

```
// 変数の初期化  
hp = 100; // 体力を 100 に設定
```

- ・破棄

破棄はオブジェクトが削除される直前に実行されるイベントであり、スコア加算やエフェクト生成、リソース開放などに使用する。

```
// 破棄時に爆発エフェクトを生成し、スコアを加算  
instance_create_layer(x, y, "Effects", obj_explosion);  
  
score += 50;
```

- ・クリーンアップ

クリーンアップはオブジェクトが削除された後に実行されるイベントである。しかし、破棄イベントとは違い削除後の処理に適用される。また、データリストや配列の開放などに使用する。

```
// メモリ解放（データリストの削除）  
ds_list_destroy(my_list);
```

- ・ステップ

ステップは毎フレーム実行されるイベントであり、オブジェクトの移動や重力処理、当たり判定などをを行うことが出来る。

```
// 右方向に移動  
x += speed;  
  
// 画面外に出たら削除  
if (x > room_width) {  
    instance_destroy();  
}
```

- ・アラーム

アラームはカウントダウン後に実行されるイベントである。例えば、"alarm = 60;"と記述すると、1秒後(60 フレーム後)に Alarm[0] イベントが発動される仕組みである。

```

// Create イベント

alarm[0] = 60; // 1 秒後に Alarm[0] イベント発動

// Alarm[0] イベント

show_message("1 秒経ったよ！"); // スペースキーを押したら 3 秒後にアラーム発動

if (keyboard_check_pressed(vk_space)) {

alarm[1] = 180; // 3 秒後 (60fps × 3 秒) に Alarm[1] 発動

}

//Alarm[1] イベント

show_message("スペースキーを押して 3 秒後だよ！");

```

・描画

描画はオブジェクトの見た目を変更するイベントである。しかし、デフォルトのスプライトを描画しない場合、”draw_self();”を使用する必要がある。

```

// 体力バーを表示

draw_text(x, y - 20, string(hp));

draw_healthbar(x - 20, y - 10, x + 20, y, hp, c_red, c_black, 0, true, true);

```

・マウス

マウスは名前の通りマウスの操作に応じて発動するイベントであり、クリックやドラッグ、ホバー時の処理に使用する。

```

// 左クリックでオブジェクトを削除

if (mouse_check_button_pressed(mb_left)) {

instance_destroy();

}

```

- ・キーボード

キーボードはキーが押されている間、毎フレーム実行されるイベントであり、ゲームの基礎であるプレイヤーの移動などに使用する。

```
if (keyboard_check(vk_left)) { x -= 5; }

if (keyboard_check(vk_right)) { x += 5; }
```

- ・キープレス

キープレスはキーボードと違い、キーを押した瞬間のみ1回実行されるイベントであり、ジャンプや攻撃のトリガーに使用する。

```
if (keyboard_check_pressed(vk_space)) { vspeed = -10; // ジャンプ }
```

- ・キーアップ

キーアップはキーを離した瞬間に1回実行されるイベントである。例えば「ボタンを離したら止まる」などに使用する。

```
if (keyboard_check_released(vk_left)) { hspeed = 0; }

if (keyboard_check_released(vk_right)) { hspeed = 0; }
```

- ・コリジョン

コリジョンは他のオブジェクトと衝突したときに実行されるイベントであり、敵との接触やアイテムの取得に使用する。

```
// 敵にぶつかったらダメージ

if (other.object_index == obj_enemy) {

    hp -= 10;

    instance_destroy(other);

}
```

- ・非同期

非同期とは時間がかかる処理の結果を受け取るイベントである。HTTP リクエストやデータのロード、オンライン通信などを行う際に使用する。

```
// 非同期 HTTP リクエストの処理

if (async_load[?"status"] == 200) {

    show_message("データを受信しました!");
}
```

2.5 Delphi

Delphi は統合環境(Integrated Development Environment)の一つで、プログラム開発に必要な機能を一つのソフトウェアパッケージにまとめた開発環境のこと、プログラミング言語の Pascal をベースとしている。視覚的なコンポーネントと開発ツールの豊富なセットを提供し、開発者が GUI を持つ Windows アプリケーションを多くのコードを書かずに作成できる。

2.6 Pascal

Pascal(パスカル)は、1970 年にスイスのニクラウス・ヴィルトによって、学生のプログラミング教育を主目的に開発されたプログラミング言語である。Pascal は逐次、条件分岐、繰り返しによる構造化プログラミングに適しており、すべての変数はデータ型を宣言する必要があり、データ型の種類も多い。

2.7 C++

C++は、汎用プログラミングの一つで、C言語を機能拡張した言語である。例を挙げると、オブジェクト指向や関数・演算子のオーバーロード、例外処理などが実装されている。また、C言語で書かれたプログラムはC++でもコンパイル（人間が書いたソースコードを機械命令に変換する）することができる。しかし、C++で書かれたプログラムをC言語でコンパイルすることはできない。

第3章 ゲーム構成

3.1 ゲーム企画

産技短を舞台としようとした経緯は、武田が「Undertale」というゲームに非常に興味を持っていた。

そこで、卒業研究で「Undertale」を参考にしつつ、産技短を舞台とした2DRPGを制作しようと考えた。

3.2 操作方法

本ゲームにおける主な操作キーを表2に示す。

表2 操作方法

キャラクターの移動	矢印キー
ダッシュ	Shift キー
決定	Enter キー
調べる	Space キー
メニュー画面	Escape キー
スクリーン切り替え	Function11 キー

3.3 キャラクター

3.3.1 主人公キャラクターの経緯

主人公を誰にするか考えたところ、A405 のホワイトボードに産技短のマスコットキャラクターが掲示されていた。そのキャラクターが過去に情報技術科の先輩が制作したものだと知り、キャラクターに「SANTAN」という名前を付け、主人公採用にした。

3.3.2 情報技術科の先生を動物化したキャラクター

敵キャラを誰にするかチームメンバーで考え、話し合った結果、情報技術科の先生にするのが良いのではという案が出た。そこで、Google Form を用いて、先生方にアンケートを取った。(図4) 先生方の完成したドット絵として一部例を以下に挙げる。(図5、6)

自分を動物に例えるならなんですか?

6件の回答

ねこ
逆に教えてほしいです。
キツネ。かな?
たぬき
ライオン
フクロウ(Web診断結果でした)

本研究で先生方をドット絵で作成した後、使用して問題ないでしょうか?

6件の回答

● はい
● いいえ

100%

図4 アンケートフォーム



図5 石館教授



図6 飯坂准教授

3.3.3 キャラクターの作成方法

GMS2では、「スプライト」を用いてオブジェクト化する必要がある。主人公のスプライトを作成する手順は以下の図の通りである。

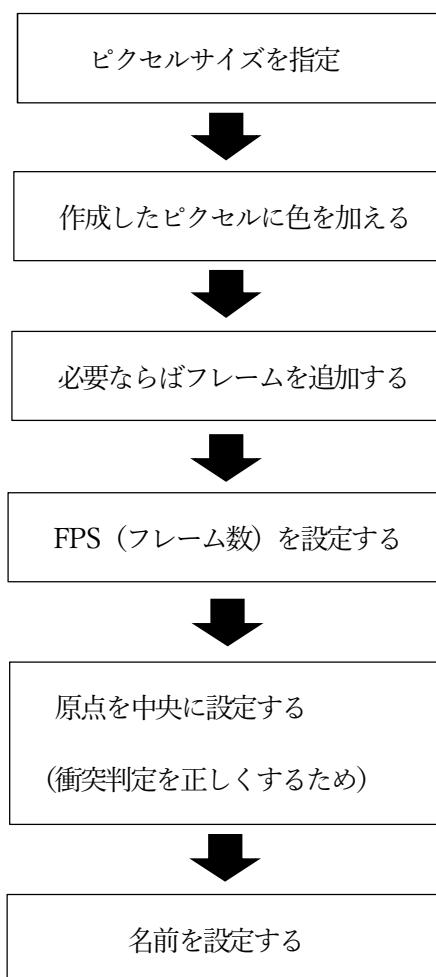


図7 スプライトの作成方法

キャラクターのアニメーションを作成するには、前後左右それぞれのドット絵が必要になる。また、

待機中のモーションと歩いているアニメーションの2種類を作成する必要がある。

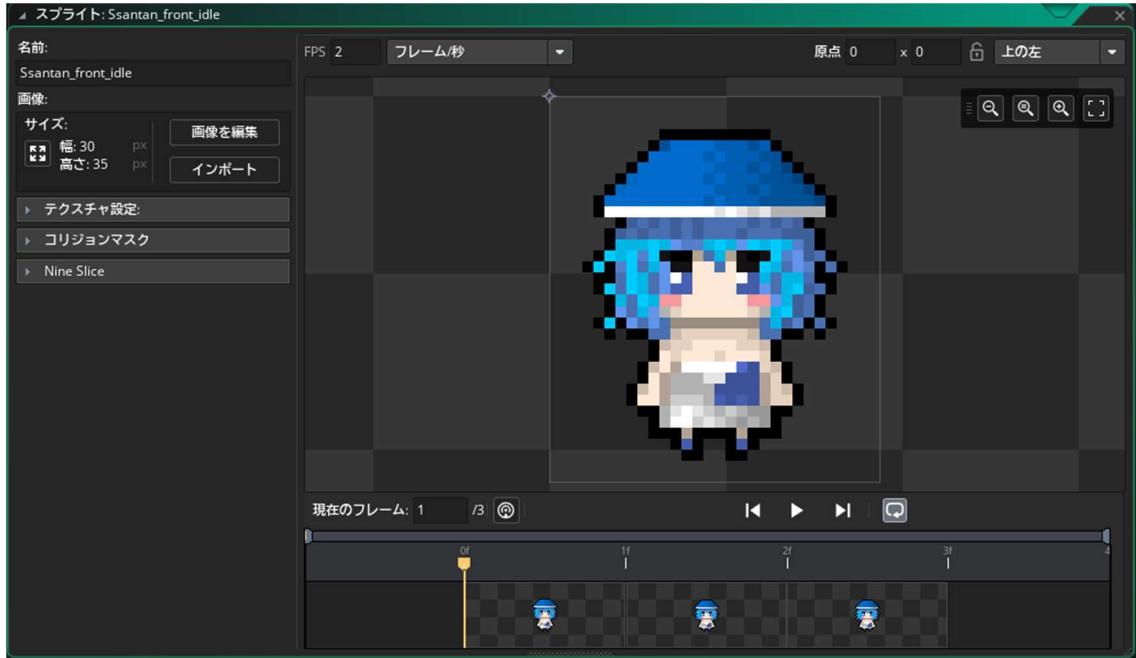


図8 待機モーション

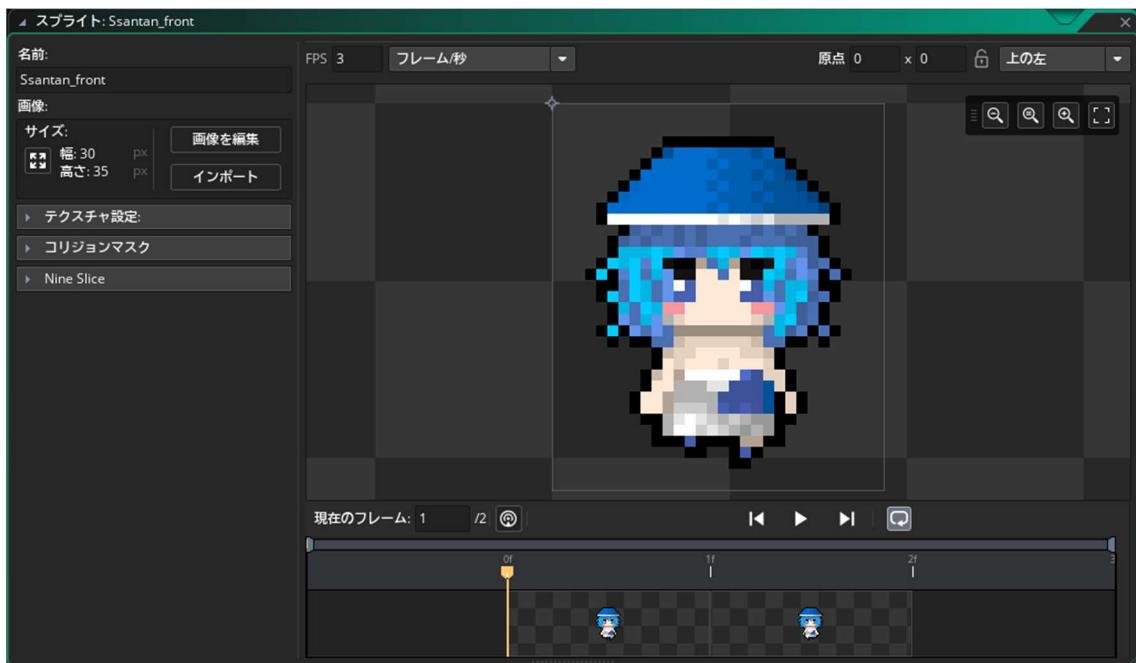


図9 歩いているアニメーション

3.3.4 キャラクターの動作方法

スプライトを作成しただけではキャラクターの操作はできない。まず、オブジェクト化というものが必要になる。オブジェクト化とは、スプライトに「役割」や「動作」を与えるためのプロセスであり、以下の手順で行う。

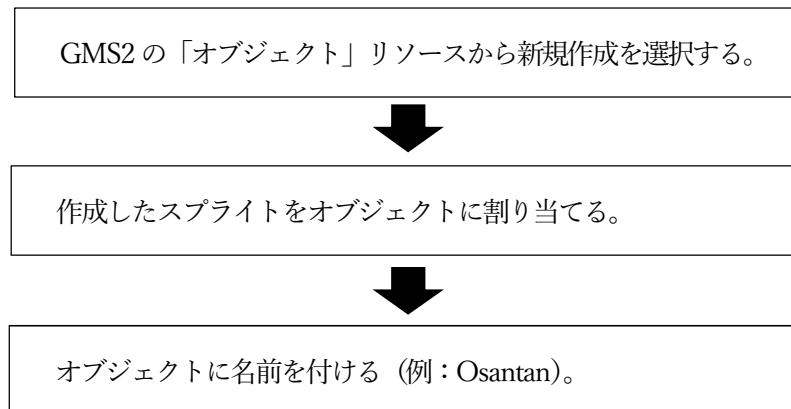


図10 オブジェクトの作成手順

しかし、オブジェクトは単にスプライトを持っているだけでは動かない。ここで「イベント」を追加し、キャラクターの動作を制御する。例えば、以下のようなイベントを追加することで、キャラクターは操作できるようになる。

- ・作成イベント: 初期設定 (変数の定義、スプライト設定など)
- ・ステップイベント: 每フレーム実行される処理 (移動、アニメーション制御など)
- ・コリジョンイベント: 他のオブジェクトと衝突した際の処理

イベントごとに GML を記述し、キャラクターの挙動をプログラムする。以下に主人公の歩くためのプログラムの簡単な例を示す。

```
// Create イベント  
  
walk_speed = 4;  
  
sprite_index = spr_Osantan_front;  
  
  
// Step イベント  
  
if (keyboard_check(vk_right)) { //vk_right は右矢印キー  
    x += walk_speed;  
  
    sprite_index = spr_Osantan_right;  
  
}  
  
if (keyboard_check(vk_left)) { //vk_left は左矢印キー  
    x -= walk_speed;  
  
    sprite_index = spr_Osantan_left;  
  
}
```

この過程でオブジェクトが作成できる。

3.4 マップの作成

3.4.1 マップの重要性

マップの作成は、ゲーム開発において非常に重要な要素の一つである。GMS2 では簡単に魅力的なゲームワールドを構築することができる。特に、タイルセットやオブジェクトをドラッグ＆ドロップで簡単に作成が可能である。

3.4.2 マップの構造

GMS2 でのマップは主にタイルマップとオブジェクトで構成される。タイルマップは、2D 空間に繰り返し使える小さな画像を並べることで、広大な世界を効率よく表現するために使用される。一方、オブジェクトはキャラクターやインタラクティブな要素（例えば、アイテム、NPC、トリガーなど）を配置するために使用される。これらを適切に組み合わせることで、ゲームの進行やストーリー展開を支えるマップが形成される。

3.4.3 タイルマップの作成

タイルマップは、タイルセットと呼ばれる画像群を用いて作成される。タイルセットは、一つ一つのタイル（小さな画像）の集まりであり、これを GMS2 の「タイルマップ」エディタで配置することで、広大なマップを簡単に作成することができる。タイルセットを作成する際には、各タイルのサイズを統一し、隣接するタイルが自然に繋がるようにデザインすることが求められる。例えば、地面、壁、水面、草などのタイルを作成し、それらを組み合せて多様な地形を表現する。

下記のようにタイルマップのレイヤーを作成し、実際にルームに画像を貼り付けて作成する。

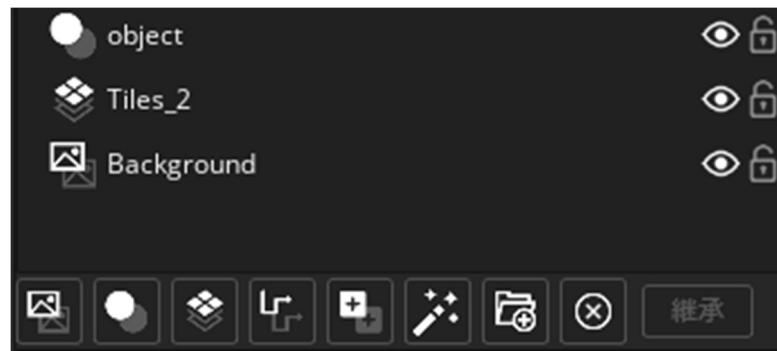


図11 タイルマップ

3.4.4 ロケーション・ハンティング

本研究では、産技短のドット絵のクオリティを高めるために、実際に写真を撮影した。この写真を活用しながら、様々なイベントの作成、細かいオブジェクトの作成をした。以下が実際の写真と GMS2 のドット絵の写真である。



図12 実際の玄関の写真



図13 GMS2の画面

3.4.5 オブジェクトの作成

タイルセットで床は完成できたが、飾りや椅子などのオブジェクトは完成していない。そこで、飾りなどのオブジェクトの作成方法について説明する。ここでは、食堂のドット絵を例に説明する。(図14)



図14 食堂のドット絵

オブジェクトの作成方法はキャラクターと変わらないが、「レイヤー」の重ねる順番が関係してくる。レイヤーの順番を間違えると、椅子が机の上に乗るなどの不可解なことが起こってしまうため、食堂のスプライトでは図15になっている。



図15 レイヤーの順番

3.4.6 マップの壁の作成方法

タイルセットとオブジェクトを配置したが、衝突判定がないため、このままでは主人公が壁を貫通してしまう。これを改善するために壁のオブジェクトが必要になる。以下が可視化できるようになった壁の衝突判定のオブジェクトである。(図 16)

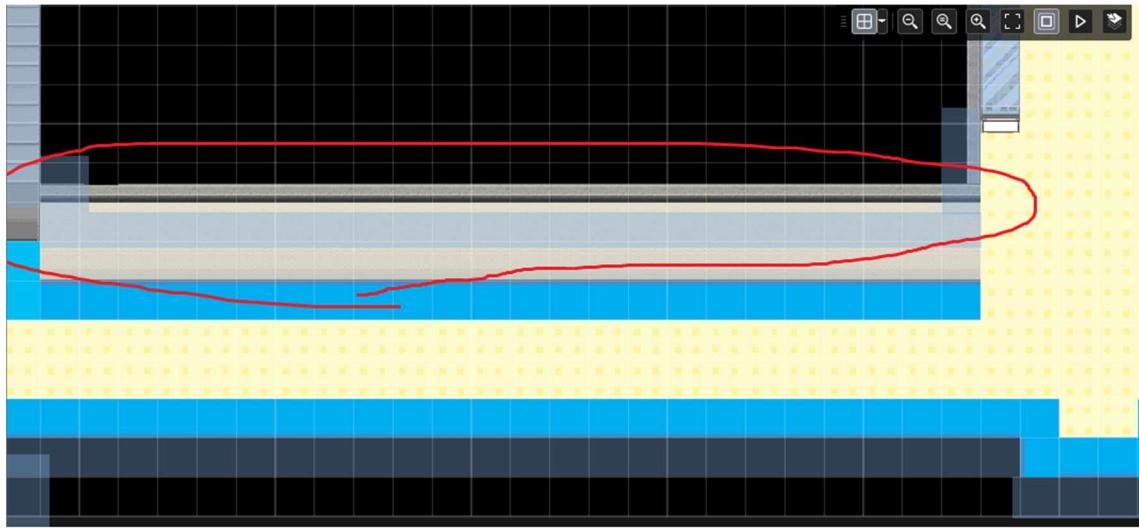


図 16 壁の衝突判定

この透明な青いオブジェクトが衝突判定である。壁に配置する際に、下に置きすぎると、通路が連れなくなってしまうので、位置を細かく調整する必要がある。

3.4.7 マップの移動遷移

最後に、マップ遷移（ルーム遷移）を作成する。

方法は簡単であり、移動したい場所にオブジェクトを配置し、移動するキャラクター（今回は主人公）にオブジェクトのコリジョン判定を追加することで移動ができる。（図 17）

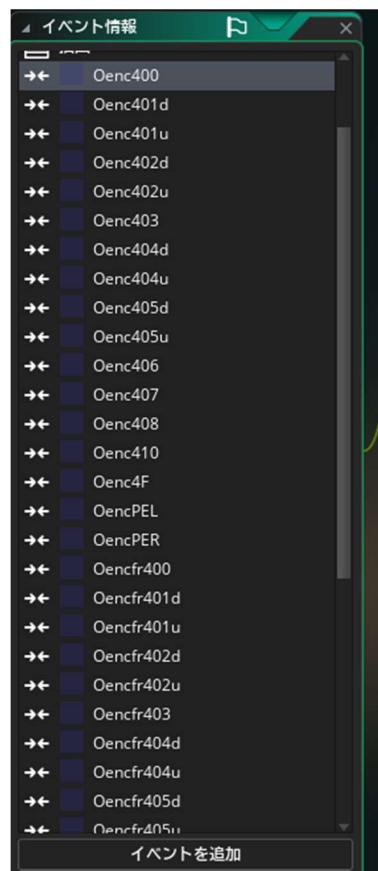


図 17 コリジョンイベント

3.5 マップのスクリプト

マップを作成する上で重要なのは、単に静的なタイルとオブジェクトを配置するだけではなく、動的な要素やイベントを実装することである。これにより、プレイヤーがゲーム世界を探索し、さまざまな反応を楽しめるようになる。ここからは楽しませる要素を紹介する。

3.5.1 イベントトリガー

GMS2 では、特定の位置にオブジェクトが到達したときにイベントを発生させる「トリガー」システムを利用することができます。例えば、プレイヤーが特定のエリアに入ると、マップの一部が変化する、敵が現れる、隠されたアイテムが出現する、というようなイベントを設定することができる。このようなインタラクションは、ゲームにおける進行やストーリー性を強化するために非常に効果的である。

3.5.2 パスファインディング

NPC や敵キャラクターがプレイヤーを追いかける、または特定の目的地に移動する必要がある場合、パスファインディングアルゴリズムが有効である。GMS2 では、「path」オブジェクトを使用して、NPC が動きたい場所に向かって最適なルートを選択することができる。これにより、キャラクターの動きが自然でリアルになる。

3.5.3 ダイナミックなマップ変更

プレイヤーの行動やゲームの進行に応じて、マップをダイナミックに変更することもできる。例えば、あるイベントをクリアすることで新たなエリアが解放される、もしくはアイテムを取得することで一部の障害物が取り除かれるなどのシステムが考えられる。

第4章 ストーリー

4.1 ストーリー概要

舞台は岩手県立産業技術短期大学校（通称：産技短）。一見普通の学校に見えるこの場所には、「卒業できない」という奇妙な噂が存在している。主人公である SANTAN は、そんな謎めいた学校に通う一人の学生。

ある日、卒業がかかった最終試験を控えていた彼のもとに、「このままでは卒業できない」という謎のメッセージが届く。次々と起こる不可解な出来事、行方不明になる生徒たち、そして怪しげな先生たち…。SANTAN は仲間と共に、学校に隠された「卒業できない」真相を解き明かすべく、産技短のさまざまな場所を探索し、敵対する先生たちと戦いながら、学校の闇に立ち向かう冒険に挑む。このゲームのテーマは、「真実の追求と成長」。謎を解くことで少しずつ明らかになる学校の秘密、そして SANTAN 自身の成長も描かれていく。果たして彼は、産技短の謎を解明し、無事卒業することができるのか…？

4.2 敵キャラクターの設定

産技短に潜む謎は、単なる噂話ではなかった。学校で起きる不可解な現象の背後には、動物の姿をした謎の存在たちがいた。彼らは、まるで学校を守るように立ちはだかり、主人公 SANTAN の行く手を阻む。この動物たちは、かつて産技短の教師たちだった。しかし、「卒業できない」という呪いの正体である《産技短の生徒会長》によって、学校への執着や教育への狂信的な思いを増幅され次第に人間性を失い、動物の姿へと変えられてしまったのだ。

彼らはそれぞれ、「正しい教育」へのこだわりが行き過ぎた結果の象徴だ。SANTAN は、この動物たちと対峙することで、教育の本質や成長の意味を見つけていく。

そして、この戦いを通じて明らかになる真実。「卒業できない学校」の呪いを解くためには、動物と化した教師たちの心を解放し、彼らが失った「教育者としての本当の役割」を取り戻さなければならない。

4.3 ボスキャラクターと最終目標

SANTAN が動物の姿になった教師たちと戦い、彼らの心を解放していく中で、この学校の呪いの真相が少しずつ明らかになっていく。その先に待ち受けるのは、「生徒会長」の存在。彼はこの呪いの中心にいる人物であり、全ての教師たちを動物に変え、卒業できない学校を作り出した張本人だ。でも、彼がそんな異常なことをする理由には、彼自身の悲しい過去が隠されていた。

かつて理想の教育を追い求めた彼は、「全ての生徒が完璧になるべきだ」という信念に取りつかれ、卒業できない学校=完璧になるまで外に出さないシステムを作り出してしまった。「君たちはまだまた未熟だ。外の世界に出るには早すぎる。」そう言って、生徒たちを学校に閉じ込め続ける生徒会長。でも SANTAN は、その間違った理想を打ち破るために、最後の戦いに挑む。最終決戦のテーマは「成長と解放」。

戦いを通じて、SANTAN は生徒会長の考えに立ち向かい、「完璧じゃなくてもいい。挑戦し続けることが大切だ」という、自分なりの答えを見つけていく。そして、学園長を倒したとき、学校の呪いは解け、生徒たちは自由を取り戻す。でも、それは「卒業」のスタートラインに過ぎない。SANTAN は、仲間たちと共に新しい未来へと踏み出していく。

第5章 戦闘画面

本研究の敵キャラでは、情報技術科の先生を動物化したキャラクターを制作した。ここでは、それぞれ敵キャラの戦闘画面の操作方法について説明する。

5.1 たぬき（石館教授）

体力 1000

戦闘フローチャートは以下のとおり。

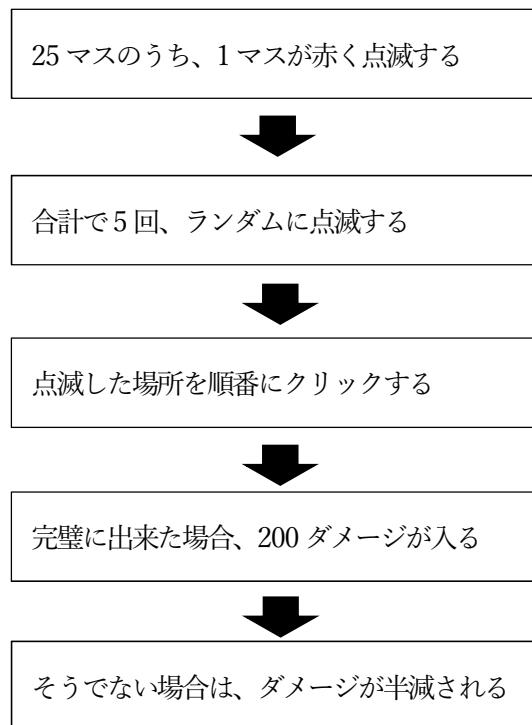


図18

この戦闘では、「Among us」というゲームを参考にした（図19）。どの世代でも楽しめるように分かりやすさを、第一に開発した。また、短期記憶のゲームであり、や兒童の方が楽しめるように工夫した。

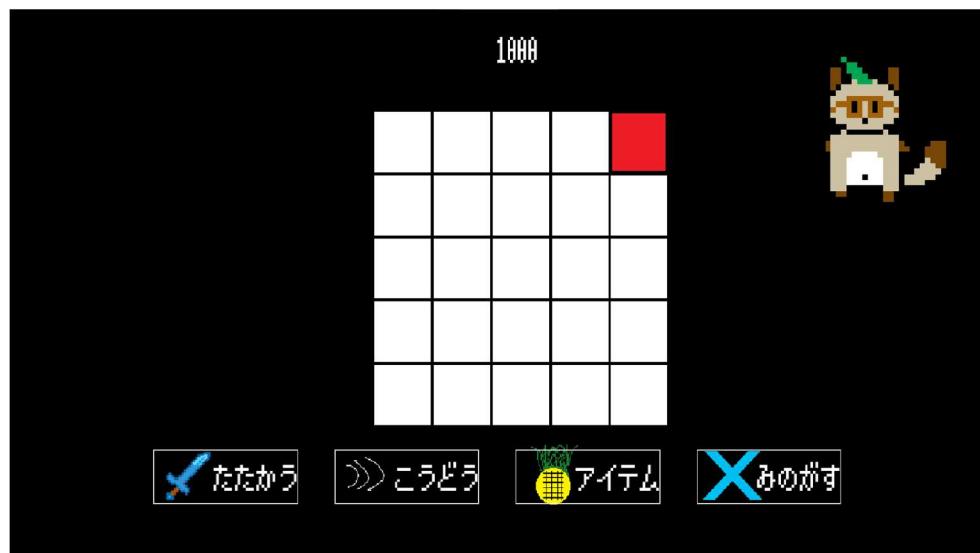


図19 戦闘画面

5.2 パンダ（菅野先生）

体力 250

戦闘チャートは以下のとおり。

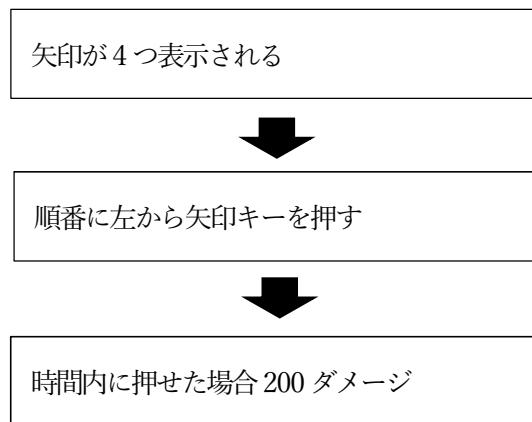


図 20

この戦闘では、ファイナルファンタジー10を参考にしている。(図21) 誰もが簡単にできるようになっている。実際に産技短展では、子供の方が楽しんでいる様子が見受けられた。

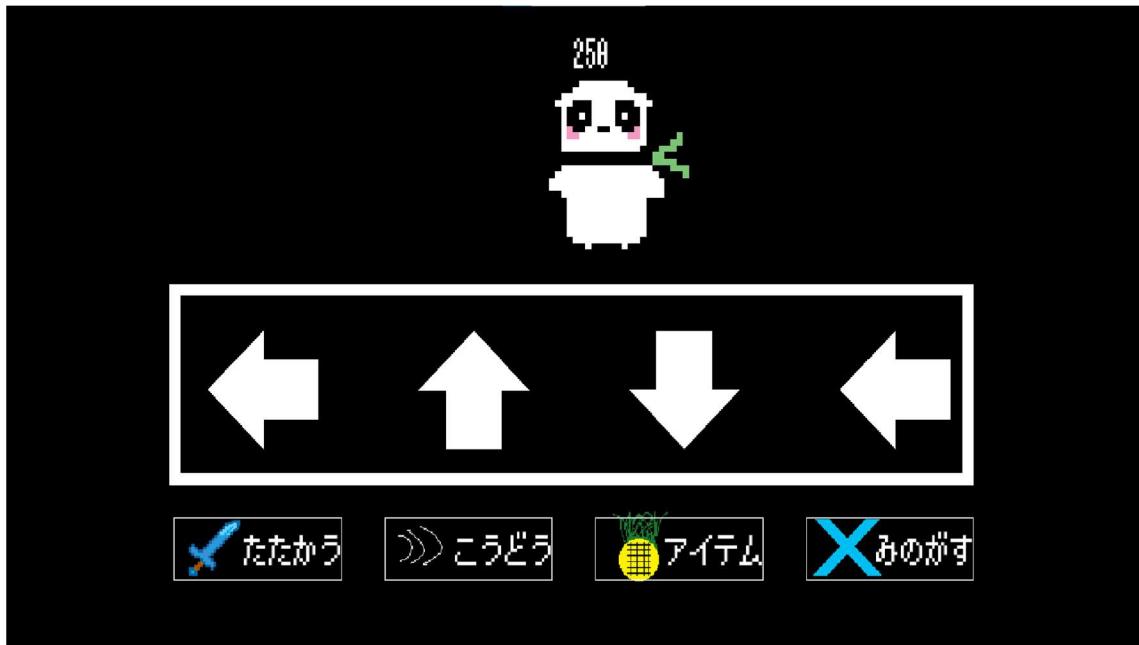


図 21 戦闘画面

5.3 キツネ（飯坂准教授）

体力 400

戦闘フローチャートは以下のとおり。

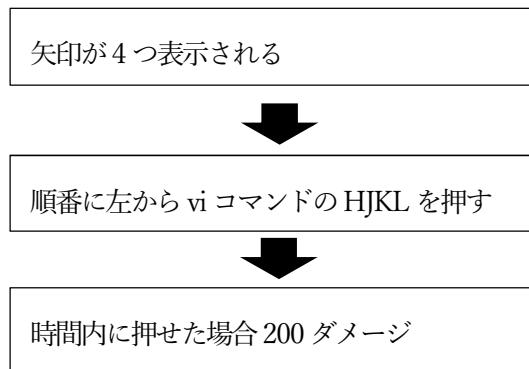


図 22

この戦闘は、パンダと同じ戦闘画面である。しかし、vi エディタの「HJKL」を用いることで、難易度を上昇させている。

以下が実際の戦闘画面である。

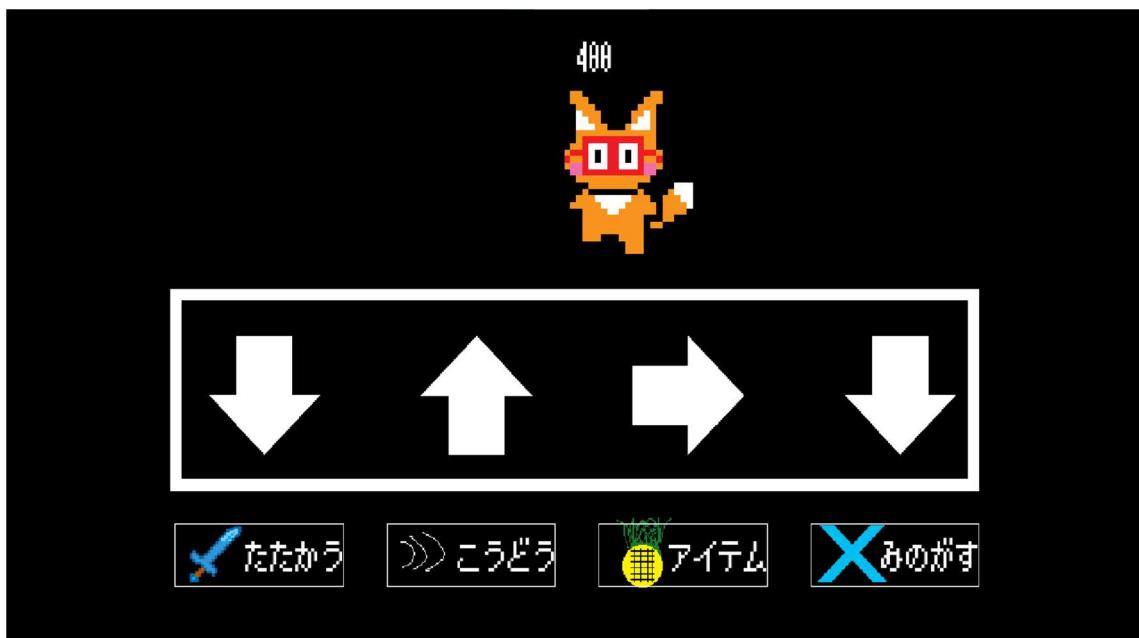
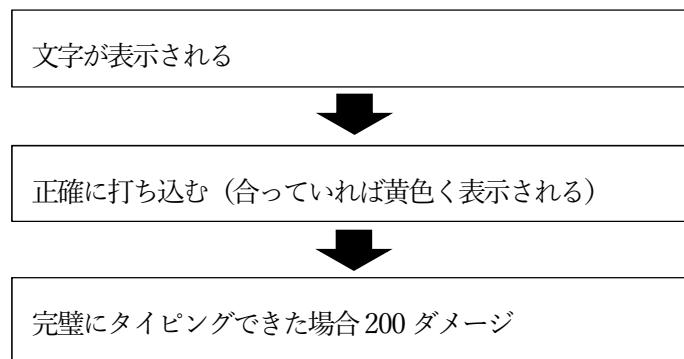


図 23 戦闘画面

5.4 ネコ（高橋強先生）

体力 1000

戦闘フローチャートは以下のとおり。



以下が実際の戦闘画面である。



図 25 戦闘画面

5.5 ライオン (安倍先生)

・体力 400

ライオンとの戦闘は、シンプルに Space キーを連打する戦闘となっている。およよそのダメージは、

10~20 回→75 ダメージ

20~30 回→100 ダメージ

30~40 回→150 ダメージ

となっている。しかし、こちらも裏技として Space キーを一度も入力しなければ 999 ダメージが入るようになっている。

以下が実際の戦闘画面である。



図 26 戦闘画面

5.6 フクロウ (菅原先生)

- ・体力 500

戦闘フローチャートは以下のとおり。



図 27

この戦闘では、Wii パーティを参考にしている。10秒からカウントダウンが始まり、0秒丁度を目指して Space キーを押すことが目的である。

- ・-1秒～+1秒の誤差で 100 ダメージ
- ・-0.5秒～+0.5秒の誤差で 150 ダメージ
- ・0秒丁度で 200 ダメージ

以下は実際の戦闘画面である。



図 28 戦闘画面

5.7 ウサギ（ソソラ先生）

・体力 1500

ウサギとの戦闘では、イライラ棒を参考にしている。またUndertaleのように、ハートが浮いておりハートが当たらないように棒を避けるのが目的である。ハートは荒ぶるようになっており、難易度がかなり上昇している。しかし、一度も当たらなければ700ダメージ入るようになっている。

また、物理判定の白い柱が右から左にスクロールして流れてくるが、このオブジェクトだけでなく、キャラクターも箱内上部でスクロールするようになっている。

以下は実際の戦闘画面である。

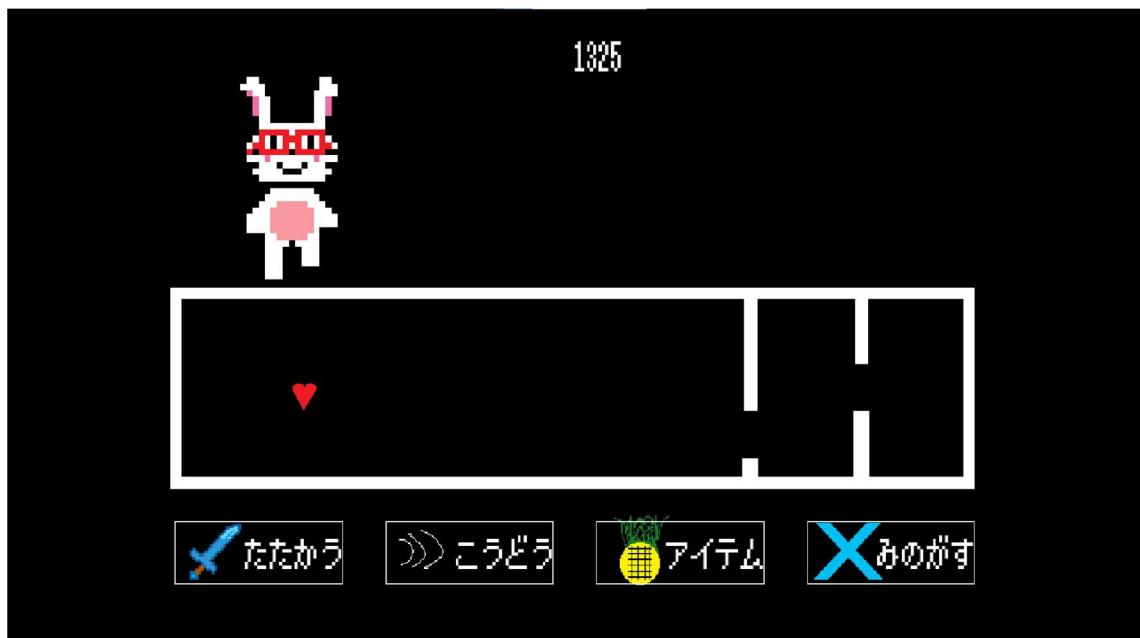


図 29 戦闘画面

5.8 生徒会長（ラスボスのクマ）

- ・体力 2500

最後の戦闘では、ともに探検したクマと戦闘することになる。今までの戦闘をすべて組み合わせてランダムで戦闘画面を表示する。また、攻撃が成功した場合、次ターンではセリフが変化するようになっている。

クマは目が赤くなり、本気で戦闘しているイメージを持たせた。また、ともに探検したキャラに裏切られた、主人公 SANTAN の気持ちを BGM に表している。

以下は実際の戦闘画面である。



図 30 ラスボス戦闘画面

第6章 プレイヤーへの配慮

6.1 マップ閲覧機能

本館棟4階に掲示されている校内案内図(図31)を作成し、各部屋を簡潔に表示し把握しやすいようにした。また、マップを閲覧している間は背景を灰色にし、現在地が理解しやすいうように赤色で点滅している。

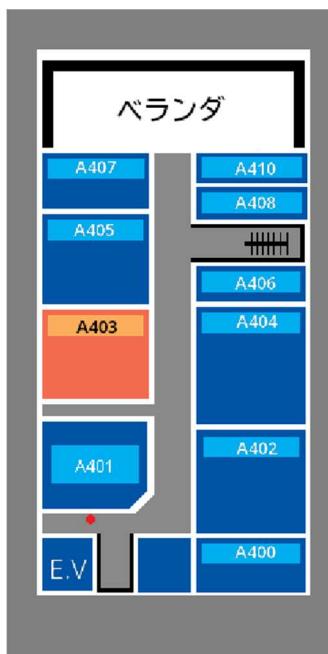


図31 内案内図

6.2 マップ閲覧機能

4階のA405では、ホワイトボードで入り口が隠されている部分がある。(図32) キャラクターとオブジェクトにプログラムを記述することで、Spaceキーで選択した際に左に動き、その先の部屋に入ることが出来るようになっている。

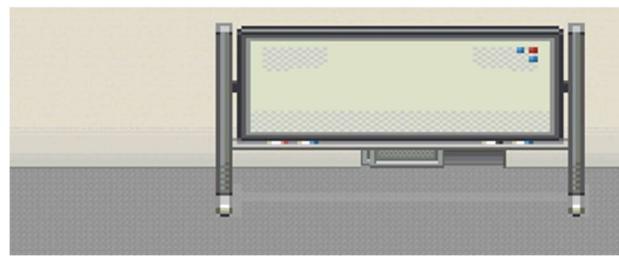


図32 ホワイトボード

6.3 フラグ管理

ゲームが始まった序盤では、クマの会話イベントを終了させないとその先のストーリーを進行する
ことが出来ないようになっている。これはGMS2で「flag」を用いてこの機能を実装
している。

これは、Systemのオブジェクトを配置しておりその先に進行できないようになっている。



図33 フラグ管理

6.4 ルートの指示

本研究では、ストーリーに沿って産技短を探索していくが、進行が止まらないようにクマが教えられるようになっている。下記の画像は正面玄関での会話が終わった際に話しかけられる画面である。これだけでなく、4階の各教室に入室した後もどの教室か、理解しやすいようになっている。



図34 進行

第7章 環境構築

7.1 GMS2 のインストール

公式サイト (<https://gamemaker.io/en>) を開き、右上の DOWNLOAD をクリックする。

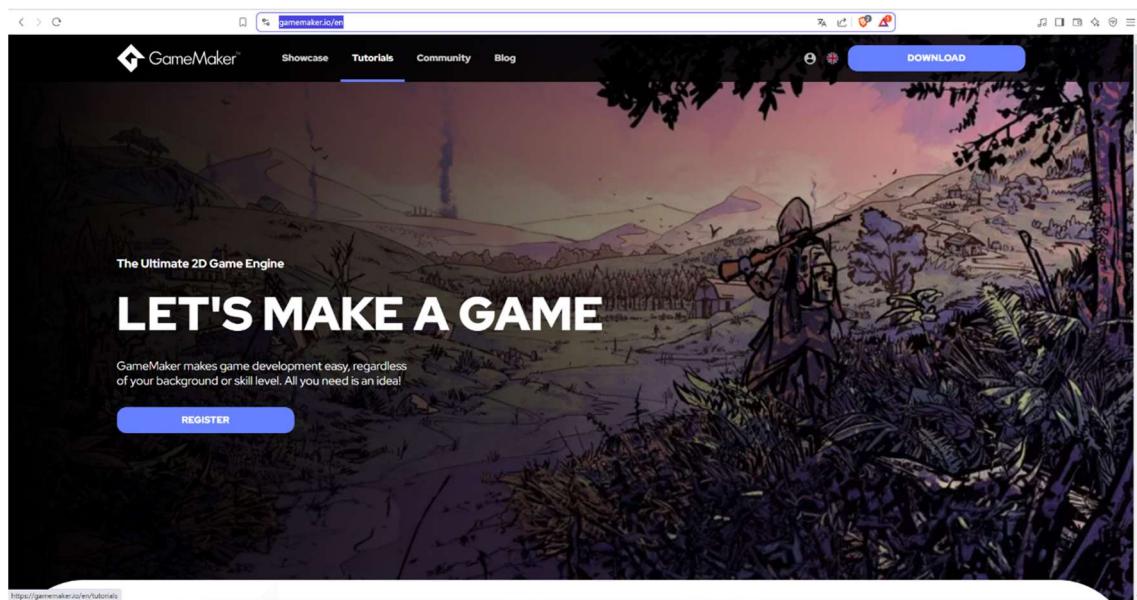


図35 公式サイト



図36 インストーラ

インストーラを起動し、画面の指示に従ってインストールする。

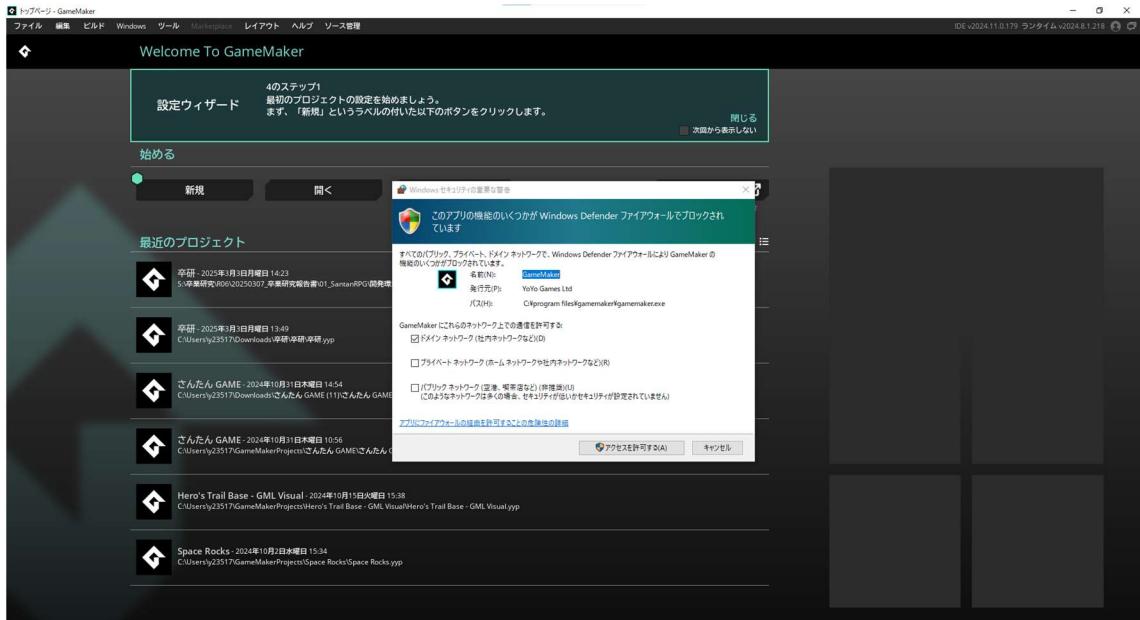


図37 GMS2 のホーム画面

インストールが完了すると、上記の画面が表示される。(図37)

表示されたらプロジェクトを作成する。しかし、最初はチュートリアルから始めるのがよいと考えている。

また、公式サイトからインストールもできるが、「Steam」からもインストールも可能である。

第8章 ゲーム環境

ゲームをプレイするには、GMS2 で F5 キーを押して実行するか、作成したプロジェクトの exe ファイルを実行する必要がある。

以下の画像は GMS2 であるが、F5 キーだけでなく赤い丸で囲んでいる場所を押しても実行できる。

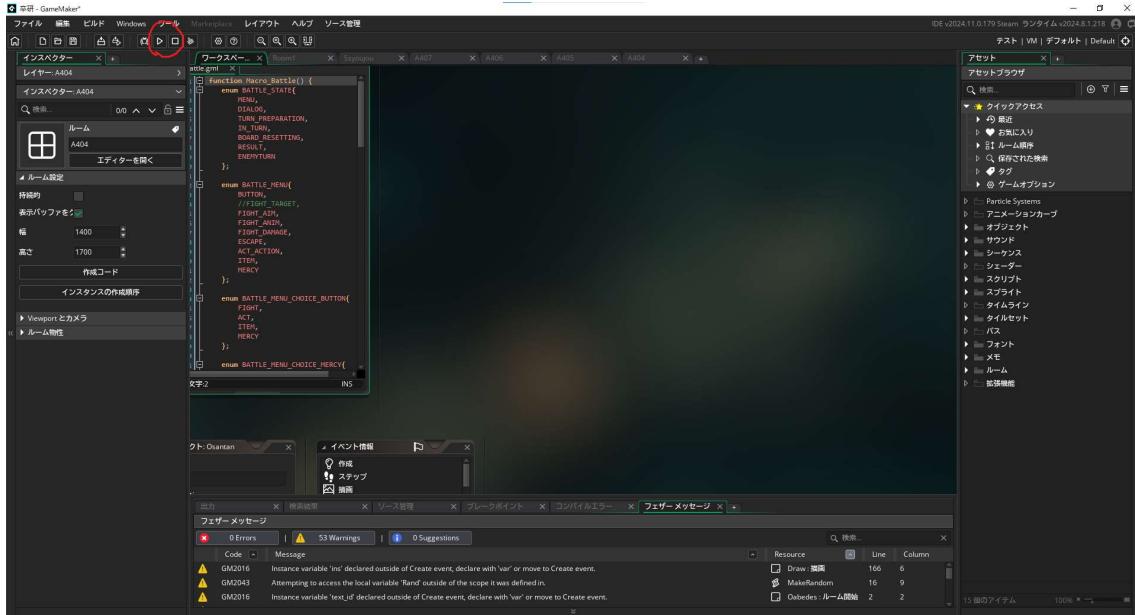


図 38 実行画面

exe ファイルを実行するためには、

S:\卒業研究\YR06\20250307_卒業研究報告書\01_SantanRPG\開発環境・完成品・動画等\卒研\

卒研.exe

を実行すると本作品がプレイできる。

第9章 おわりに

本研究を通じて、2D ゲームの制作手法を学ぶだけでなく、チームで協力しながら開発を進めることの重要性を実感することができた。ドット絵の作成が終わった後も、プログラムの調整や仕様のブラッシュアップを重ね、チーム全員で作品の完成度を高めることに尽力した。

結果として産技短展では、子供から大人まで様々な世代の方が楽しんでいた。また、ゲームを制作している方々の発想力や労力が素晴らしいことを理解できた。

この開発したゲームをこれからも産技短展やオープンキャンパスでは是非遊んでほしい。

第 10 章 参考文献

[1] Gaming Reverends – DevCorner. Shader Tutorial for GameMaker 01 - Introduction to GLSL

Part1.2017/10/9

https://www.youtube.com/watch?v=ch4BYqkL1w8&list=PL0kTSdIvQNCNE-BDKOIYu628AalMmXy_P

[2] @2dgames_jp. 【GMS2】 シェーダーの基本的な使い方.2020/12/06

https://qiita.com/2dgames_jp/items/1975aa91e169fa13b2d6

[3] @2dgames_jp. 【GMS2】 ゲームデータの管理方法.2020/12/23

https://qiita.com/2dgames_jp/items/cfa93d7dd61a50750b0c

[4] Peyton Burnham. How to Make an RPG in GameMaker Studio 2! (Part 1: The Basics with Player Movement and Collision).2021/03/26

<https://www.youtube.com/watch?v=KnfQo32ME5g&list=PL14Yje2sgzySnBUIQLhq2VJXRLi66gFf>

[5] @2dgames_jp 【GMS2】 会話イベントの作り方.2020/12/26

https://qiita.com/2dgames_jp/items/945d410b75722b368f94

[6] GameMaker 入門

https://manual.gamemaker.io/monthly/ja/Introduction/Introduction_To_GameMaker_Studio_2.htm

[7] GameMaker スクリプトの関数と変数

https://manual.gamemaker.io/monthly/ja/GameMaker_Language/GML_Overview/Script_Functions.htm