パーティゲームとコントローラの製作

石井 斗翔 石川 和久 井上 龍 木口 和也

1. 研究概要

私たちの班では、幼いときに家族や友達と 楽しんだ経験があるパーティゲームを制作し た。また、Arduinoを使用したコントローラ も製作した。

2. 研究内容

すごろくゲームを基盤にミニゲームやギミ ックを実装したゲームを、プログラム・グラ フィック・コントローラ・BGMに役割を分担 して作業を進めた。

<開発環境>

- Unity 2022.3.8f1
- Visual Studio Code ver1.95.3
- Studio One PRIME
- blender ver4.2
- Arduino

<制作過程>

(1)プログラム (担当:木口)

○開発初期

最初に小さなマップ(図1)のゲーム制作 の土台として用意し、ここに継ぎ足していく ような形で開発を進めた。



図1 開発初期のマップ

マスと次のマスへの移動のプログラム(図 2)とルーレット(図3)をこの時に作成 し、この時点でゲームとして遊べるような最 低限の要素を用意した。

```
position=Player.transform.position;
  position.x += STPlayer.Movex;
  position.y += STPlayer.Movey;
  position.z += STPlayer.Movez;
  Player.transform.position =
  position;
  position.y += 3;
  ca.transform.position = position -
  (Player.transform.forward * 5);
else
  Moving = false;
  textobje.GetComponent<Text>().text
= STPlayer.step.ToString();
  textobje2.GetComponent<Text>().text
= STPlayer.step.ToString();
  STPlayer.mass++;
  if (STPlayer.step == 0) {
    mass_test(Player, STPlayer, STPlaye
r.mass);
```

図2 マス移動のコードの抜粋

図2のプログラムの簡単な説明と流れ

- マスの移動を開始したらそこまでの距離 と毎フレームあたりの移動距離を計算 する。
- 毎フレームあたり計算した移動距離分移 動させカメラ(視点)を移動しているプ

レイヤーの位置に合わせる。

3. 移動が完了したらバグ対策でマスの位置 にテレポートさせる。



図3 移動数を決めるルーレット

自分の番がきたプレイヤーが自分のコント ローラの決定ボタンを押すとルーレットが回 転しルーレットが止まった時の矢印の所にあ る数字分マスを移動する

○ミニゲーム

パーティゲームらしさと出すために神経衰弱(図4)やカーレースなどのミニゲームを 作成した。



図4 神経衰弱

ミニゲームの抽選は Random. Range を用い て乱数でミニゲームを抽選するプログラム (図5)を作成した。

switch(Random.Range(1,2)) { case 1: Game1();//レース break; case 2: Game2();//神経衰弱 break; case 3: Game4();// break; default : Debug. Log("ERORR:02"); break;

図5 ミニゲームを抽選するコード

○マスのイベント

特定のマスを踏むとイベントが起こるプロ グラムを作成した。マスを踏むと書いてある 数字分前に進むというマス(図6)や逆に後 ろに戻るマスを追加した(図7)。



図6 前に進むマス 図7 後ろに戻るマス

緑のマス(図8)を踏むと選択肢等が出現 し岩を回避したり(図9)画面に表示された キーを入力して橋から落ちないようにするな どの普通のマスとは異なるマスを追加した。



図8 緑のマス



図 9 岩

選択肢には白色の画像と青色の画像を用意 し、キー入力によって青色の画像の位置を入 れ替えることによってあたかも選択肢かのよ うに見せている。

```
if (((Player.name == "lplayer" &&
key_script.key_A)
||(Player.name == "2player" &&
key_script.key_Player2_A)
||(Player.name == "3player" &&
key_script.key_Player3_A)
||(Player.name == "4player" &&
key_script.key_Player4_A))
&& pos == 2)
{
    pos = 1;
    boublue1.transfrom.position
= bousen1.transfrom.position;
}
```

図 10 選択肢のコード

図 10 のコードの説明

Player に移動しているプレイヤーのオブ ジェクトが入っているため、その名前とキー 入力が同じだったらという if 文が true の場 合、変数 pos の値を変えて青い画像の位置を 入れ替えいる。また、選択し終わったらバグ 対策のため0に(初期化)している。

○プレイ人数の変更

当初は4人でプレイすることを想定してい たが、2人や3人でも遊べるようにミニゲー ムやルーレットのコード(図11)を変更し2 人や3人でも遊べるようにした。

```
image.SetActive(false);
top_player++;
if(top_player == 4 ||
(mainsrc.Player_sum == 3 && top_player
== 3) || (mainsrc.Player_sum == 2 &&
top_player == 2))
{
    top_player = 0;
}
count = 0;
```

select_number_tate = 0;

(2) グラフィック(担当:石井)○マップ設計

主に Unity に備わっている terrain を使っ て高低差を作り地形を作成(図 12)。



図 12 マップ全体像

木や岩などのマテリアルを設置すること で地形が単調にならないようにしている(図 13)。



図 13 マテリアルの例

橋や鉄骨などを設置(図 14・15) すること で、ゲームを作る際イベントを作りやすいよ うにしている。当初の予定では橋は Asset Store から素材を持ってくる予定だったが、 見つからなかったため自作となっている。





図14 橋

図 15 鉄骨

○没マップ

当初、火山をモチーフとする裏マップ(図 16)を実装する予定だったが、不具合により 断念する結果となった。



図 16 裏マップ全体像(未完成)

橋や鉄骨を渡る処理はマスを移動するとき の処理と違い独立している(図 17)。マス移 動は関数を使っているが、橋などの移動は座 標を使い合わせている。

```
if (bridge_timer_timer >= 0)
{
       bridge_timer++;
       bridge_timer_timer--;
       if (bridge_timer <= 30)
       {
               Vector3 position =
       Player. transform. position;
               position. x = 0.2f;
               position.z += 0.15f;
Player.transform.position = position;
            else if (bridge_timer <=
90 && bridge_timer >= 31)
                Vector3 position =
Player. transform. position;
                position. x = 0.305f;
                position.z += 0.132f;
                position.y -= 0.01f;
Player.transform.position = position;
```

図 17 橋を渡る処理のコードの抜粋

(3) コントローラ(担当:石川)○使用部品

- Arduino pro micro
- Arduino micro
- ・スイッチ
- ・抵抗
- ・ユニバーサル基板
- ・コントローラのカバー

○コントローラの外装

コントローラの外装は Blender を使ってモ デルを作成し 3D プリンタで印刷した。初期サ イズだと少し大きく操作しにくかったため横 のサイズを2 c m程短くし、その他の細かい 部分も改良した(図 18・19)。





図 18 改良前

図 19 改良後

○プログラム

ボタンの入力をキー入力に変換する(コ図 20)。このコードでは「t」に変換している。 キー入力の後にディレイをかけることで1回 ボタンを押すと複数回入力されるのを防いで いる。

#include" Keyboard.h"
<pre>void setup() {</pre>
Keyboad.begin();
delay(1000);
pinMode(2, INPUT);
pinMode(3, INPUT);
<pre>pinMode(4, INPUT);</pre>
pinMode(5, INPUT);
pinMode(6, INPUT);
<pre>pinMode(7, INPUT);</pre>
}
void loop(){
if(digitalRead(2) == HIGH) {
Keyboard.press('t');
delay(300);
Keyboard.releaseAll();
delay(10);
}

図 20 Arduino のコードの一部

○配線・組み立て

ブレッドボードで配線を考え(図 21)、動 作を確認してから基板の作成をした。ユニバ ーサル基板に Arduino や抵抗をはんだ付けし た後、3D プリンタで作成した外装に接着し た(図 22)。



図 21 コントローラの配線



図 22 完成したコントローラ

(4) BGM (担当:井上)

ゲーム内で使用する BGM には楽しく明るい 雰囲気になるように心掛けた。ゲーム性に合 うメロディやテンポを考えて制作した。特に ミニゲームの神経衰弱の BGM はこだわって作 成した。



図 23 BGM 制作の様子

Studio one で BGM 制作をした。ここでメ ロディやビートの打ち込み、各楽器の音量調 整などをする (図 23)。オリジナル曲とメロ ディを一部サンプリングしている曲を制作した。今回は時間の関係で remix ができていないため完璧な作品ではないが、ゲームに合った曲ができているのではないかと感じた。

3. 研究のまとめ

今回の課題研究を通して、プログラミング や電子工作などの3年間学習してきた様々な 技術を活用することによってできた、いわば 3年間の集大成となる作品を制作することが できたのではないかと感じた。作業を進める 中で様々な問題が発生し、進行に遅れが出て しまったりしたことなどから満足にできてい ない部分もあった。そこから年間を通して計 画的に活動をしていくことの難しさを感じ た。また、各役割間でコミュニケーションを 取り、一つの作品を完成させていくことの難 しさを感じた。しかし、この課題研究で協力 して一つのものを作り上げていく能力を培う ことができたので卒業後も学んだことを生か して活動していきたいと考えている。

4. 参考・素材

○Unity プログラミングで参考にしたサイト 初めての Unity 入門!プログラミングの基礎 や独学法を徹底解説 | 侍エンジニアブログ https://www.sejuku.net/blog/56542

【簡単】Unity でキー入力する方法!キーコ ード一覧も掲載 | ラルトンブログ https://raruton.com/unity-how-to-useinput-key/

今日からはじめる Unity #初心者 - Qiita <u>https://qiita.com/nmxi/items/7950fb12ef9</u> <u>25efa276d</u>

【Unity】Unityで爆速でルーレットを実装 する #C# - Qiita https://qiita.com/kaku0710/items/3d2a581

77aec403709c9

【Unity 入門】パーティクルを覚える | STYLY <u>https://styly.cc/ja/tips/unity-</u> <u>introduction-particle/</u>

○グラフィック作成で参考にしたサイト
 Unity の Terrain の基本操作 | 山や川などの
 地形を作成してみる! | Liibercraft | リー
 ベルクラフト

https://liibercraft.com/archives/6165

OUnity AssetStore https://assetstore.unity.com/ja-JP

○BGM 作成で参考にしたサイト コード進行まとめ(100 種類以上)0-T0【音 楽理論ウェブアプリ】 <u>https://o-to.khufrudamonotes.com/o-to-</u> chord-progression