

# 音楽ゲームの制作

植田 光稀 津田 晃輝  
野上 綾斗 福永 歩  
松岡 晃嗣

## 1. 研究概要

私たちは、日頃からゲームで遊んでおりその中でも音楽ゲームはたくさんの方が遊んでいたことから、どのような役割分担でできているか興味を持ち、ゲーム制作の理解と大変さを学習するために音楽ゲームの製作を行った。

### 年間計画

4月～6月 担当作業のアイデア出し及び勉強

7月～11月 各自担当の作業

12月～1月 結合作業及び資料作成

## 2. 研究の具体的内容

音楽ゲームの基本となる楽曲、キャラクターの設定、ガチャの設定などについて決定し、音楽ゲームの設定に合うように、ストーリー制作、ガチャ制作、譜面制作、キャラクター制作に分担して作業を進めた。

〈それぞれの作業内容〉

### 1) ストーリー制作(福永)

ストーリーを制作する前に班のメンバーと話し合ってからキャラクターの性格や担当楽器などを決めてからストーリーを制作し始めた。ストーリーは三部制で構成されており、一部ずつunityで実装した。一部は、新入部員(4月)が入部する設定なので背景を部室の画像にし、キャラクターの話し方に敬語を混ぜ合わせた。二部は、一部から少し時間が立ち小規模な文化祭で演奏会を開催する設定にしたため背景を練習場所である音楽室にし、敬語をやめて友達と話すような話し方に変化させた。三部は、夏休み中にある夏の大会に向けて合宿に行く設定なので

背景を合宿で泊まる別荘やキャラクターが見に行った滝など場面に応じて背景の画像を変更していった。表1は、各キャラクターの設定、担当楽器を示しており、キャラクターの会話など

名前	学年	担当
桜(主人公)	2年生	ボーカル
葵	3年生	ベース
連	2年生	ギター
琴美	1年生	キーボード
凧	1年生	ドラム

全員軽音部で季節は、4月から8月まで

キャラクターの設定に合うようにストーリーを制作した。

表1 各キャラクターの設定

unityで制作したストーリーを動かすと背景、キャラクター、キャラクターが話すテキストが表示されテキストに関しては0.5秒ごとに一文字ずつ表示させる。また、マウスで左ボタンが押された場合は、テキスト表示中であればテキストが全文表示され、全文表示された状態でもう一度左クリックをすると次のシーンへと遷移する。



写真1 左クリックする前

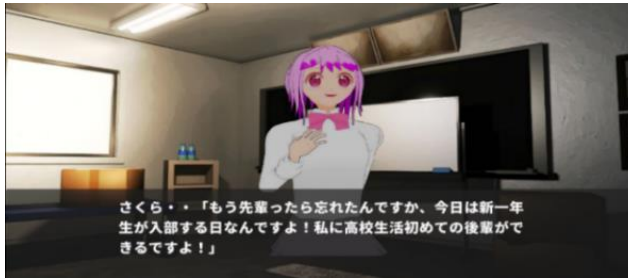


写真2 左クリックした後

```
//最初の行のテキストを表示、または命令を実行
string statement = GameManager.Instance.userScriptManager.GetCurrentSentence();
if (GameManager.Instance.userScriptManager.IsStatement(statement))
{
    GameManager.Instance.userScriptManager.ExecuteStatement(statement);
    GoToTheNextLine();
}
DisplayText();

// Update is called once per frame
void Update()
{
    //文字を一文字ずつ出す
    _time += Time.deltaTime;
    if (_time >= _feedTime)
    {
        _time -= _feedTime;
        if (!CanGoToTheNextLine())
        {
            _displayedSentenceLength++;
            _mainTextObject.maxVisibleCharacters = _displayedSentenceLength;
        }
    }

    //クリックされたとき、次の行へ移動
    if (Input.GetMouseButtonUp(0))
    {
        if (CanGoToTheNextLine())
        {
            GoToTheNextLine();
            DisplayText();
        }
        else
        {
            _displayedSentenceLength = _sentenceLength;
        }
    }
}

```

写真3 テキストを表示するプログラム

## 2) タイトル画面及びメニュー画面作成とガチャのロジック作成に関して (津田)

画面作成をするにあたり、音楽ゲームの雰囲気合いそうなものを、フリー画像を元に編集して Pixlr X という画像編集用サイトを使用し、タイトル名とキャラクターを配置。その後実際に Unity の Project に組み込んだ。音楽ゲームの雰囲気に合った効果音を流すためにフリーBGM から BGM を選定した。

タイトル画面には明るい背景を使用し、タイトル名にはその文字に合うようにフォントと文字色を使用し、メインキャラクターであるさくらとれんを配置した。このようにすることで実際にある音楽ゲームの雰囲気に近づけた。

```
public void OnClickToMenuSceneButton()
{
    FadeManager.Instance.LoadScene("MenuScene", 2.0f);
}

```

写真4 画面遷移プログラム



写真5 タイトル画面



写真6 メニュー画面

ガチャのロジック作成に関しては、プログラムでキャラの個数、一回で引ける個数、またガチャの値段をあらかじめプログラムで設定、そして Unity 側でキャラが排出される確率を設定できるようなプログラムを作成した。このプログラムを Unity 側で呼び出す、というプログラムを Web サイトのプログラムのソースコードを参考にしながら作成しようとしたができなかった。

うまくいかなかった原因は植田くんが制作しているガチャの仕様に自分の制作したロジック

のプログラムが設計段階からそもそも合わなかったことが原因だと考えている。植田君が想像していたプログラムは「直接カードを指定して、そこでランダムに確率を生成してガチャをする」というものだったので設計段階から植田君との調整不足というところがあり、結果的にガチャのロジックのプログラムを作成することができなかった。

### 3)Blender でのキャラクター作成(野上)

Blender とは CG を自由に制作できるアプリケーションソフトである。オープンソースのフリーソフトであるため、ダウンロードすれば誰でも簡単に利用できる。価格が無料なのに、有料ソフトにも劣らない機能を持っていることが、このソフトの特徴である。

この音楽ゲームの主人公が所属している軽音部のメンバーの 3D キャラクターを 5 名、そしてメンバー達の友人を 2 名、そして楽器を 5 個制作した。自分の作業がすべて終わった後時間が余ったので、今まで作成したキャラクターのリメイクを 2 名行った。

#### キャラクター制作の流れ

##### (1) モデリング

キャラクターの顔、髪、体、服、小物のモデリングを行った。顔から作り始め、体→髪→服→小物の順に制作を 1 体ずつ行った。オブジェクトのメッシュを細分化し表面を滑らかに見せてくれる機能のサブディビジョンサーフェスやミラーなどのモディファイアなどを使いキャラクターモデリングを行った。

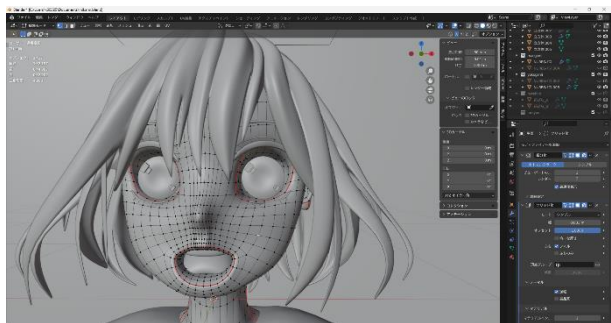


写真 7 モデリング

##### (2) UV 展開

UV とは 3DCG モデルにテクスチャを貼り付けるとき、位置や方向、大きさを指定するための座標情報のことで、キャラクターに色を塗るため UV 展開を行った。まずは 3D モデルを展開するための切れ目「シーム」を入れなければならない。色が変わる場所、凹凸が激しい場所、見づらい場所にシームを入れた。その後 UV 展開をし、UV の位置を調整した。

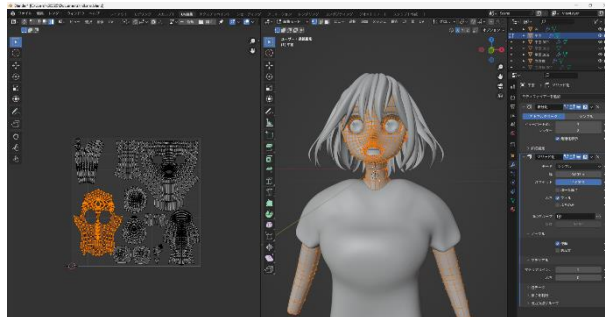


写真 8 UV 展開

##### (3) テクスチャペイント

UV 展開を行ったオブジェクトにテクスチャペイントでキャラクターの色塗りを行った。セルルック調のキャラクターを制作しようと思ったので、基本的に色はベタ塗りで影はライティングで制御を行った。



写真 9 テクスチャペイント

##### (4) リギング

ボーンといわれる人間の骨の部分の制作を行った。このリギングとはオブジェクトを動かす仕組みを作成する工程のことです。この工程を行うことで簡単にキャラクターのポーズをつけることができ、アニメーションを行うことが可能になる。作成したキャラクターに合わせてボ

ーンを配置し、キャラクターとボーンを紐付けることでボーンに合わせてキャラクターを動かすことができた。

## 楽器制作の流れ

### ① モデリング

ギター、ドラム、ボーカル、ベース、キーボードを実際の楽器を参考にモデリングを行った。

### ② マテリアル設定

それぞれの楽器に色と質感を付けた。



写真 10 ドラム

## (5) カード制作

キャラクターとそれぞれの担当楽器を組み合わせたカードを合計 15 枚制作した。それぞれのポーズやカードの背景を考えるのに時間がかかった。下の写真は制作した三枚のカードである。



写真 11 完成したカード

## (6) リメイク

少し時間に余裕ができたので、なにかできることはないかと考えた結果、今まで作ったキャラクターのうち 2 体をリメイクした。今までの作っていたキャラクターとは作り方を変えた。今まで

は、3D モデルをアニメ調に見せるセルルックという技法を使用していたが、このリメイクではセルルック技法と折り目、穴、互いに近接する表面に発生するソフトシャドウを疑似する機能であるアンビエントオクルージョンを組み合わせ、アニメ調の色塗りで 3D モデル感を出した。



写真 12 リメイクキャラ 1



写真 13 リメイクキャラ 2

## 4) ガチャ・カードリスト作成(植田)

### ・画面制作

プログラム制作に取り掛かる前に UI のデザインを行った。デザインには Fotor という画像編集サイトを利用し、ガチャのメニュー画面や演出画面、リザルト画面、提供割合画面を制作した。その際、ユーザにとってわかりやすいデザインになるように考えながら制作した。

### ・画面遷移

Unity エンジンの SceneManagement というクラスと Unity 内の Button を組み合わせて画面遷移を実装した。また、FadeManager というパッケージを利用し、画面遷移の際、フェードイン・フェ

ードアウトといったアニメーションをつけた。

#### ・ガチャ演出制作

実際のゲームのガチャの演出を参考にしながら、簡単な演出機能を実装した。具体的には Unity 内の Particle を利用して背景にエフェクトをつけたり、C#でキーの入力を検出し、球体の座標を変えて移動させたり、AudioSource コンポーネントを用いて効果音を鳴らすといった機能を制作した。

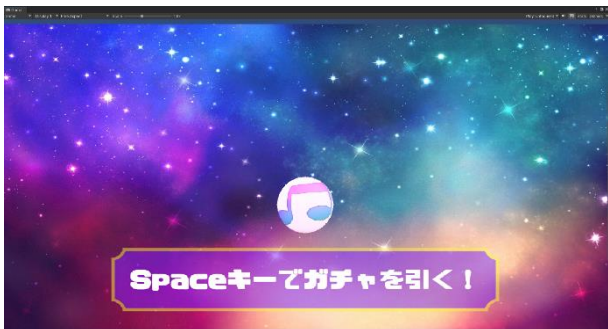


写真 14 ガチャ演出画面

#### ・ガチャプログラム制作

ガチャのランダム要素の作成には、Random というクラスの Range メソッドを利用した。Range メソッドで取得した 1~100 の乱数をグローバル変数に格納し、その値に応じて分岐処理を実行するという仕組みでガチャを設計した。また、分岐先ではカードの情報を Unity 側から受け取り、ゲーム画面で表示する場所やカードの大きさを設定するプログラムを記述した。

```
GameObject work_image1 = new GameObject("Chara1");
work_image1.transform.parent = GameObject.Find("Canvas").transform;
work_image1.AddComponent<RectTransform>().anchoredPosition = new Vector3(0, 0, 0);
work_image1.GetComponent<RectTransform>().localScale = new Vector3(3, 3, 3);
work_image1.AddComponent<Image>().sprite = Resources.Load<Sprite>("カード風N");
work_image1.GetComponent<Image>().preserveAspect = true;
```

写真 15 ガチャプログラム

#### ・リザルト画面プログラム制作

リザルト画面ではカードの表示やカード名、レアリティ、プロフィールといったカードの情報を表示するプログラムを制作した。カードの表示はガチャプログラムの一部を利用し、座標や大きさをレイアウトに合わせて調整した。カード情報の表示は乱数を格納したグローバル変

数の値を参照し、その値に応じてテキストファイルを読み込むという方式で実装した。

```
string loadText1 = file1.text;
text1.text = loadText1;
```

写真 16 テキストファイルの表示



写真 17 完成したリザルト画面

#### ・ガチャ BGM の設定

BGM を DOVA-SYNDROME というフリーBGM の配布サイトからダウンロードし、BGM を鳴らすプログラム(写真 17)を制作した。このプログラムでは、インスタンスを Unity 側から取得し、そのインスタンスを DontDestroyOnLoad というシーンを切り替えたときにオブジェクトを放棄せずに引き継ぐ関数の対象にすることでシーン遷移に対応している。ここでいうインスタンスは AudioSource コンポーネントを追加した gameObject のこと。

```
static public GachaSoundScript instance;
void Awake()
{
    if (instance == null)
    {
        instance = this;
        DontDestroyOnLoad(gameObject);
    }
    else
    {
        Destroy(gameObject);
    }
}
public bool DontDestroyEnabled = true;
// Use this for initialization
void Start()
{
    if (DontDestroyEnabled)
    {
        DontDestroyOnLoad(this);
    }
}
```

写真 18 BGM を鳴らすプログラム

### ・カードリストのUI、プログラム制作

ガチャで入手したカード、入手していないカードの種類や枚数を表示する UI、プログラムを制作した。入手しているかどうかの判別は、カードの上に半透明の黒の image コンポーネントを被せ、プログラムでそれぞれのカードのフラグを参照し、その状態によって image のSetActive(※)を false にして破壊することで実装した。またカード枚数に関しては、ガチャを引いた瞬間にカード別の枚数計算用の変数の値を加算し、その値をUnity側で表示した。

※ SetActive・・・オブジェクトの表示・非表示を bool 値 (true か false) によって切り替えるメソッド。

```
if (RandomResult.cardNumFlag1 == true)
    card1.SetActive(false);
```

写真 19 カードの入手状態の検出



写真 20 完成したカードリスト画面

### ・ゲーム内通貨の実装

ガチャを引くために必要な通貨の要素を追加した。通貨の所持数計算用のグローバル変数に初期値として5を格納し、その値をメニュー画面などで表示することで常に通貨の所持数がユーザーにわかるようにした。また、それに応じてストーリーを読み終わった際や音楽ゲームをクリアした際に通貨が加算、ガチャを引いた際に通貨が減算されるよう変数値を調整するプログラムや、通貨の所持数に応じてボタンやテキスト、image をSetActiveメソッドで表示・非表示にするプログラムを追加した。

```
// UI Text指定用
public TextMeshProUGUI MoneyText;
// 表示する変数
public static int money = 5;

// Update is called once per frame
void Update()
{
    MoneyText.text = money.ToString();
}
```

写真 21 通貨の所持数表示

```
int money = MoneyCount.money;
if (money >= 5)
{
    StopText.SetActive(false);
}
else
{
    GachaButton.SetActive(false);
    DiamondUI.SetActive(false);
}
```

写真 22 表示・非表示の切り替え



写真 23 通貨が足りているとき



写真 24 通貨が不足しているとき

### ・10連ガチャの実装

岡工祭の展示が終わってから課題研究の時間で少し時間に余裕ができたので、なにかできることはないかと考えた結果、10連ガチャを引くことができたならおもしろいと思い、この機能の実装に至った。アルゴリズム自体は基本的に単発ガチャの処理を10回繰り返すということで実装できたが、演出の面で違いができたのでそれ

に合わせる形でプログラムを組んだ。具体的には、時間差で一枚ずつカードを表示したり、効果音を鳴らしたりするプログラムを追加した。

```
sec += Time.deltaTime;
if (secflag == false)
{
    if (sec >= 3f)
    {
        sound.PlayOneShot(SEa);
        capsule1.SetActive(false);
        Debug.Log(rnd);
        if (rnd >= 1 && rnd <= 12)
        {
            GameObject work_image1 = createG
```

写真 25 時間差処理のプログラム (抜粋)

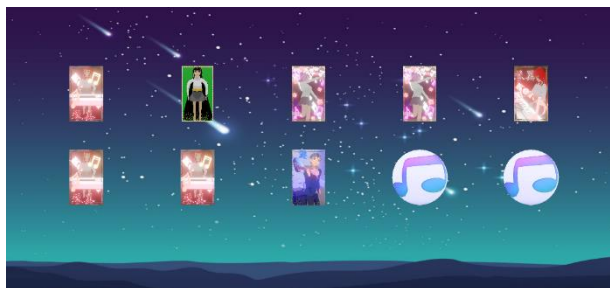


写真 26 完成した 10 連ガチャの結果画面

## 5) ステージ、譜面制作(松岡)

### ・ステージ作成

最初に画面の設計を行った。まずノーツと呼ばれるリズムに合わせて上から流れてくるオブジェクトが流れるレーン road とキーボードを押したときノーツの衝突判定機能をもつオブジェクト collisionbox、collisionbox と衝突しなかったオブジェクトを破壊しコンボ数をリセットするライン resetbox を配置し、次にボタンやスコア、コンボ数、リザルト画面などの UI を配置した。(写真 26)その後 VisualStudio を使用し、C#でこれらのオブジェクトにアタッチするスクリプトを作成した。アタッチとは、オブジェクトに対して別のオブジェクトやコンポーネントを紐づけする操作を指す。

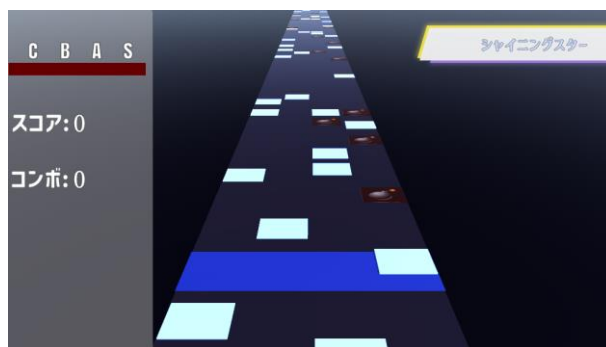


写真 27 プレイ画面

### ・譜面作成

譜面作成にあたって著作権フリーの楽曲を利用した。当初は J-POP やボカロなどの楽曲を利用する予定だったが、著作権の関係で利用が困難と判断し断念した。譜面は外部のアセットを利用して制作する予定だったがアセットが動かなかったため、ノーツのプレハブを作成し、C#スクリプトでプレハブ化したノーツのインスタンスをゲーム開始時に座標を指定して生成させる方法に切り替えた。プレハブとは再利用可能にしたオブジェクトのことであり、今回のように同じオブジェクトを大量に複製したりすることができる。また、ほかの音楽ゲームを参考に、通常のノーツの中に障害物として collisionbox に衝突するとスコアがマイナスされる黒いノーツ bakudan を設置することで緊張感を与えることにした。(写真 28)

```
void Start()
{
    GameObject objbakudan = (GameObject)Resources.Load("musicbox.bakudan");
    //左から1番目のレーン
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 109.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 183.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 174.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 183.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 203.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 263.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 274.0f), Quaternion.identity);
    Instantiate(objbakudan, new Vector3(-3.02f, 0.35f, 283.0f), Quaternion.identity);
}
```

写真 28 bakudan のプログラム (抜粋)

### ・結合作業

岡工祭の前にストーリーとガチャをパッケージ化しインポートすることで一つのゲームにした。そのあと、もともと画面サイズによって変動していたアスペクト比を 16:9 に固定したほか、メニュー画面から各項目に移動するために画面遷移のプログラムを流用したりして 1 つの

ゲームとして動くようにした。

#### 6) オープンスクール、岡工際の反応について

オープンスクールのときガチャや音楽ゲームを実際にプレイしてもらってとても楽しんでもらえるような表情をしていた。また作品の概要に説明しているときは「この音楽ゲームはすごい！」と中学生の人たちを感じてもらうことができた。オープンスクールなどを通して人を楽しませることができたので課題研究で音楽ゲームを制作して良かった。岡工祭では、オープンスクールと同様に実際にプレイしてもらった。生徒、一般客共に好評で常に人が集まっていたみんな楽しそうにプレイしていたので、当初の目的であった岡工祭で楽しんでもらうという目的を達成することができた。

#### 7) タイトル作成(班のメンバー全員)

タイトル作成は4月に班のメンバー全員で案を出し合いながら題名は「SAKURA WITH AO HARU」にした。理由は主人公の名前がさくらなのと、主人公を中心にストーリーが進行していくのでこのタイトル名があっていると思い設定した。

「SAKURA」のところは後ろの背景にある桜のイメージがあったのでピンク色、「WITH」のところは背景の色に合うようにするために緑色、「AO HARU」は青春をイメージが浮きやすいように青色に設定した。



写真 29 タイトル名

### 3. 研究のまとめ

キャラクターや譜面などが完成した後もキャラクターのリメイクをしたり、初めに想定した機能以外にも制作途中で欲しいと思った機能な

どを班メンバーで検討して実装していった。具体的には、ガチャの10連機能や譜面を追加した。また、制作をしていたDe11室がパソコンの入れ替え作業のため使用不可となったので、データの移行作業などで実際より実装の時期が遅れたこともあり、自分たちのスケジュール管理能力の不足を痛感した。しかし、当初の研究目的であったたくさんの人に遊んでもらうことと、ゲーム制作の理解と大変さを学習することができたため、よい課題研究ができたと思う。

#### 4. 感想

##### ・福永

最初に目標としていた機能は全て実装できたが、実装するまでの過程で時間がかかり過ぎてしまった。それに加えてバグ修正にかなり費やしてしまい実装目標日直前の完成となってしまった。しかし、プログラミングなど楽しめたのでよい課題研究になったと思う。

##### ・津田

受験対策に時間がかかりガチャのロジック作成については、最初は私がする予定で実際にソースコードをC#で書いていたがガチャ機能と結合したときにあまりうまく行かず結局植田君がガチャのロジックを作り仕事を増やすはめになってしまったのでしっかりと植田君が作っている仕様通りに合わせるべきだと思った。またチームと協力してゲームなどを開発する場合は「このソースコードだったらうまくいくかな？」とチームのメンバーと確認しあいながらしていくべきだと感じた。しかし、仲間と協力して一つの作品を制作できたこととUnityの知識やC#などのプログラミングの知識が少しでも身についたのでよかったと思う。また大学に行ってもチームで作品を制作すると思うので今回の課題研究で培った経験や反省を生かしてもものづくりをすることが出来ればよいと思う。



・松岡

スケジュール管理を怠ったため、完成までに時間をかけすぎてしまい、完成予定日当日まで作業をすることになってしまった。しかし、試行錯誤の末にプログラムが思い通りに動いた時の達成感があり最後までモチベーションを維持することができた。また、作業の過程でゲーム制作の大変さやスケジュール管理の重要性を、身をもって知ることができた。

・野上

今回の課題研究で3Dモデリングの難しさが分かった。キャラクターを制作するにあたって全てオリジナルキャラクターだったので、キャラクターデザインを1から考えるのに苦戦した。YouTubeや本、インターネットで3Dモデリングの仕方やテクスチャの作り方などを勉強した。自分の思い描いていたキャラクターにあまり近づけることができなかった。課題研究を通して3DCGの良さやゲーム制作の難しさ、計画を立てて行動する難しさを知ることができた。

・植田

最初ガチャ制作の役割を頼まれたときは不安があったが、班のメンバーの協力があったり、インターネットの有用な情報を自分で見つけたりでき、思ったよりスムーズに作業ができたので良かった。また、ゲーム制作をしているうちに、自分のイメージしているものを実際に作ってみたいと思うようになり、UnityやC#の知識を習得しながらゲーム制作の楽しさを実感することができ、とても満足感の高い課題研究にできた。

カードリストのUIを制作している時に、本当はカードをクリックすると情報が表示される機能を実装しようと考えていたが、カードをボタン化してクリック判定をつける処理を完成させることが、時間の都合上で実現できなかったのは反省点だと思った。

参考文献

【絶対できる！】Unityでの2Dノベルゲームの作り方を詳しく解説

[https://original-game.com/how-to-make-a-novel-game-in-unity2d/#m\\_heading-4](https://original-game.com/how-to-make-a-novel-game-in-unity2d/#m_heading-4)

みんちりえ(フリー背景サイト)

<https://min-chi.material.jp/>

重み付きランダム選択アルゴリズムの速度計測

【C#】

[https://qiita.com/makihiro\\_dev/items/95d11ba2cecd16e38d](https://qiita.com/makihiro_dev/items/95d11ba2cecd16e38d)

【Unity入門】超簡単！別のシーンへ切り替える方法

<https://www.sejuku.net/blog/49352>

【Blender】顔モデリング講座 ～平面から始めるキャラクターモデリング～

[https://youtu.be/uUgQw6VpFP8?si=8MeCn2gA\\_VS24HH7](https://youtu.be/uUgQw6VpFP8?si=8MeCn2gA_VS24HH7)

【Blender】髪の毛モデリング講座 ～カーブを使ってアニメ調の髪の毛をモデリング～

<https://youtu.be/Ng8Fb0HvsLQ?si=axhvvtYRJB4Jg7dG>

【Blender】目を可愛く仕上げるモデリングのコツ 5選 ～目はこうやって描く！～

<https://youtu.be/RAA4V0kqw-I?si=IBUBfA4KJKMmRh7R>

【Blender】セルルック集中講座 ～セルルックを極める！～

<https://youtu.be/c9ZudJuBI0g?si=oMu3EjMAIZoCLcxT>

【Blender 2.83 Tutorial】キャラクター表情の作り方・顔のリギング【シェイプキー×ドライバー】

<https://youtu.be/o8FbhlkYIZA?si=gTTBAeJ111Q0I3qB>

【Blender 2.9 Tutorial】キャラクターモデリング解説 #1 -Character Modeling Tutorial #1

[https://youtu.be/BBtgjTiLlXQ?si=Pd\\_Nuba1C0tUH\\_01](https://youtu.be/BBtgjTiLlXQ?si=Pd_Nuba1C0tUH_01)

ピクトアーツ(無料イラスト配布サイト)

<https://pictarts.com/10/index.html>

Unity 入門の森

<https://3dunity.org/game-create-lesson/music-game/stage-musicblock/>

魔王魂(フリー音源サイト)

<https://maou.audio/>

【初心者 Unity】ランダムな要素の作成には乱数を使おう

[https://tech.pjin.jp/blog/2021/03/31/unity\\_howto\\_random/](https://tech.pjin.jp/blog/2021/03/31/unity_howto_random/)

【Unity】シーンを切り替えても BGM を鳴らし続ける

<https://soft-rime.com/post-14064/>

Unity オブジェクトの表示・非表示切り替え【非表示オブジェクトの読み込みも】

<https://your-3d.com/unity-set-active/>

効果音ラボ

<https://soundeffect-lab.info/>

DOVA-SYNDROME

<https://dova-s.jp/>

【Unity】ガチャ(抽選)を実装してみた

<https://unitygame.slavesystems.com/unity/?p=112>