令和６年度
電子科
課題研究発表会

令和７年1月21日（火）

東京都立杉並工科高等学校

目次

午前 10:50～	1	「Unreal Engineでの3Dゲームの制作」	
	2	「サイクルコンピュータの制作」	
	3	「マイクロビットラジコンカーの制作」	
	4	「デジタル多機能カレンダーの制作」	
午前 11:50～	5	「Unityを用いたオンラインFPSの制作」	
	6	「pythonによるマインスイーパーの制作」	
	7	「学校案内の制作」	
午後 13:10～	8	「クレーンゲーム機の制作」	
	9	「Nゲージで作った車両と街灯の制御制作」	
	10	「Music Gameの制作」	
午後 14:00～	11	「ペットボトルフィラメント製造機の制作」	
	12	「エアホッケーの制作」	
	13	「電子ピアノの制作」	
	14	「運転シュミレータの制作」	

Unreal Engine での 3D ゲーム制作

3 年 C 組

[目的]

Unreal Engine の使い方を学び、3D ゲームを制作する。

[内容]

Unreal Engine の基本的な使い方について学び、自分の作りたいゲームに必要な技術を習得する。Unreal Engine に使われている Blueprint と C++の使い方を勉強し、ゲームの基本的なプログラム作成する。

C++とは最大限の制御とパフォーマンスを提供する強力な低レベル言語である。ゲームエンジンのコアな部分や高性能が求められる場所で使われる。プログラムを書く必要がある。

Blueprint とは基本的に、スクリプトをゲームに視覚的に追加していくシステムである。ノード、イベント、関数、そして変数をワイヤーで接続して複雑なゲームプレイ要素を作成することができる。Scratch と同じイメージでプログラムを書く必要がない。

Blueprint は C++と違い、実装できない機能があります。主な機能は Blueprint のみでほぼ実装できますが、ゲームエンジンのコアな部分にアクセスするような機能は C++でないとできない。

[使用器具]

ノートパソコン(Windows 11、64bit、24H2)
Unreal Engine 5.5.1
Visual Studio Code 1.94.2

[制作]

シューティングゲームの制作において、キャラクターの移動、攻撃、敵ボットを実装行う。キャラクターとマップは Unreal Engine のマーケットからダウンロードしたものを使用している。

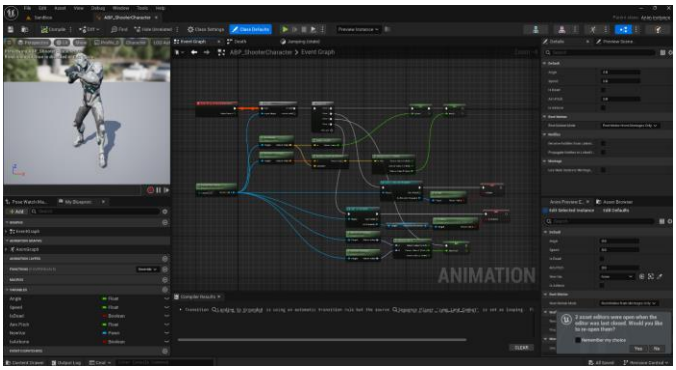


図 1　アニメーションの設定(Blueprint)

キャラクーの動き:

キーボードとマウス、またはコントローラーで操作を入力し、プレイヤーキャラクターを動かせるようする。

アニメーション：

キャラクターの動きに合わせて、スケルタルメッシュを基にアニメーションを作成する。2D ブレンドスペースや Blueprint を活用している。

シューティング：

マウスの左クリックまたはコントローラーの右トリガーで銃が発射できるようにする。弾丸の衝撃エフェクトや銃声を追加し現実感を持たせる。

敵ボット：

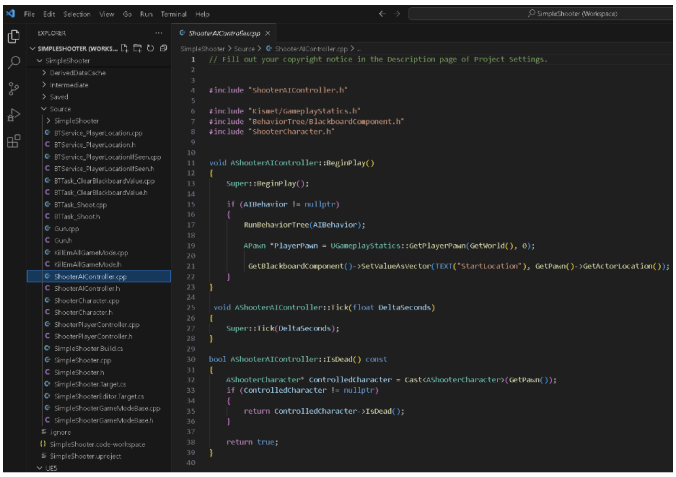
プレイヤーはボットの前に出でると、プレイヤーに接近して攻撃する敵ボット。

HP、ダメージ：

攻撃が当たった場合、ダメージをして HP が減少する。プレイヤーまたはマップ上で全ての敵ボットの HP が 0 になるとゲームを終了になる。

クロスヘア，HP バー:

クロスヘアは画面中央に表示されるマークで、敵に合わせて攻撃することでダメージすることがでる。HP バーはプレイヤーの HP を表すものである。



```
// Fill out your copyright notice in the Description page of Project Settings.

#include "ShooterAIController.h"

#include "Kismet/GameplayStatics.h"
#include "BehaviorTree/BlackboardComponent.h"
#include "ShooterCharacter.h"

void AShooterAIController::BeginPlay()
{
    Super::BeginPlay();

    if (AIBehavior != nullptr)
    {
        RunBehaviorTree(AIBehavior);

        APawn *PlayerPawn = UGameplayStatics::GetPlayerPawn(GetWorld(), 0);

        GetBlackboardComponent()->SetValueAsVector(TEXT("StartLocation"), GetPawn()->GetActorLocation());
    }
}

void AShooterAIController::Tick(float DeltaSeconds)
{
    Super::Tick(DeltaSeconds);
}

bool AShooterAIController::IsDead() const
{
    AShooterCharacter* ControlledCharacter = Cast<AShooterCharacter>(GetPawn());
    if (ControlledCharacter != nullptr)
    {
        return ControlledCharacter->IsDead();
    }

    return true;
}
```

図 2　ゲームのプログラム(C++)

[課題]

このゲームはまだ初期段階にあり、追加できる機能や改善できる点が多く存在する。今後の課題として ADS、足音、ゲーム UI、ゲームモード、マップ、武器、キャラクターなどの制作とオンラインマルチプレイヤー機能がある。

ゲームはリリースされた後もアップデートを行い、新しいコンテンツを追加し、バグがないようにしながらゲームのバランスとパフォーマンスなども考えことも開発する上で大切なことである。

[感想]

初めて使うソフトで様々なことを 1 から学び、自分の最初のゲームを作ることができました。バグやエラー直面し、混乱することもあったが、それを解決する過程と、その後の達成感は大きな喜びがありました。ゲーム制作を通じて、ゲーム開発の大変さと楽しさを実感でき、自分が作りたいものを実現できた時の満足感と、バグ解決の苦労を感じることができました。

発表番号２　　令和６年度

サイクルコンピュータの製作
３年Ｃ組

【研究背景】

普段から自転車によく乗っていて、サイクルコンピュータも普段から使っている。そこで、その仕組みを知り、自分で作ってみたいと思った。

サイクルコンピュータとは、自転車に取り付けることで、速度、走行距離などを表示できる装置である。ホイールに磁石を取り付けることで、走行距離、タイヤの回転数、速度をディスプレイに表示できるサイクルコンピュータについて研究した。

【使用機器と開発環境】

使用機器	**規格**
マイコンボード	Arduino Nano
液晶ディスプレイ	画面サイズ：0.96 インチ 解像度：128x64 ピクセル ドライバーSSD1306 通信方式：IIC
ホールセンサ	SK8552G-G03-K

開発環境	
PC	Windows 11 Pro
エディタ	Arduino IDE 1.8.19
ディスプレイ ライブラリ	Adafruit SSD1306 (ver. 2.5.10)
タイマ割込み ライブラリ	MsTimer2 (ver. 1.1.0)

【研究内容】

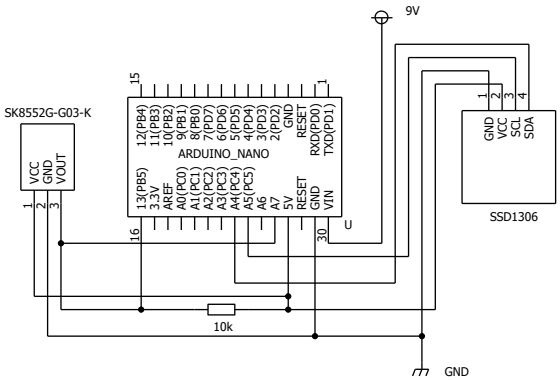
・SSD1306 を使用したディスプレイの制御

ディスプレイドライバに合わせてライブラリを用意し、文字を表示させた。

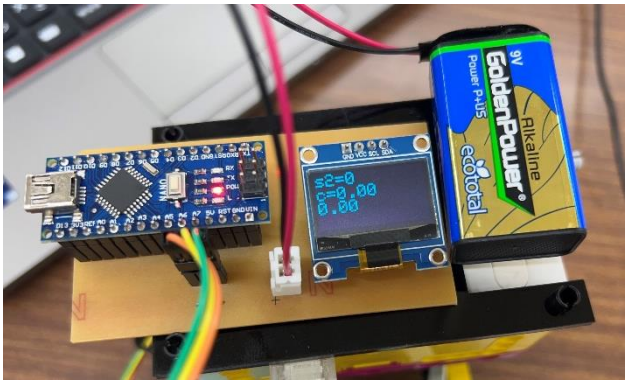
・ホールセンサの制御

使用したホールセンサ（SK8552G）は、磁石の極性をデジタル出力することができるので、タクトスイッチと同様に制御できる。(S 極：0、N 極：1)

【回路図】



【製作物】



【測定原理】

・走行距離

タイヤの回転数が分かれば、次の式で走行距離が計算できる。

走行距離＝ 円周率×タイヤの直径×回転数

（タイヤの直径：5.6cm）

回転数は、ホールセンサを使用することで、タイヤの回転角度から求められる。

また、走行距離が分かれば、次の式で速度が求められる。

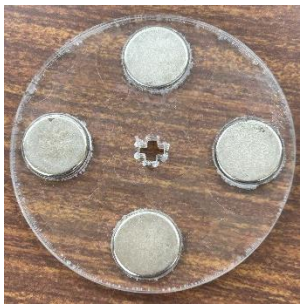
$$速度 = \frac{距離}{時間}$$

時間は、マイコンで 1 秒ごとにタイマ割り込みを発生させることで測定した。

【測定精度の向上に向けた工夫】

磁石が小さいと、センサで読み取れないことがあった。

タイヤに取り付ける磁石の個数を増やすと、測定精度を向上できる。ただし、磁石の取り付け間隔が狭すぎると、センサの読み取りに影響がある。



【感想】

①プログラミングが大変だった。

②市販のサイクルコンピュータは磁石一つで高精度の測定ができているのがすごいと思った。

③どんな機能をつけるのか自由にえらぶことができるのが自作の面白いところだと思った。

発表番号 3　　令和 6 年度

マイクロビットラジコンカーの製作

3 年 C 組

【目的】

マイクロビットやプログラミングを使ってラジコンカーを制御する。

【関連事項】

①マイクロビットとは

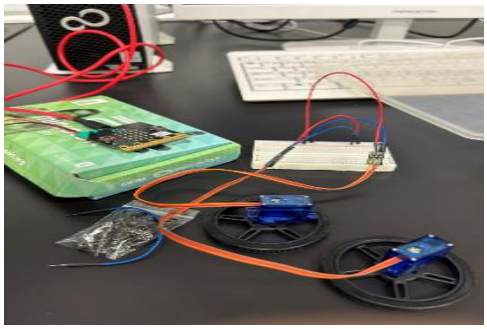
BBC が開発した小型のプログラミングデバイス

② 使用機器・開発環境

名称	型番
サーボモーター	FS90R
サーボホイール	FS90R
マイクロビット	SEDU-079525
ユニバーサルプレート	26.7 x 15.9 x 2.1 cm; 0.28 g
モータドライバボード	1 x 1 x 1xcm;24g

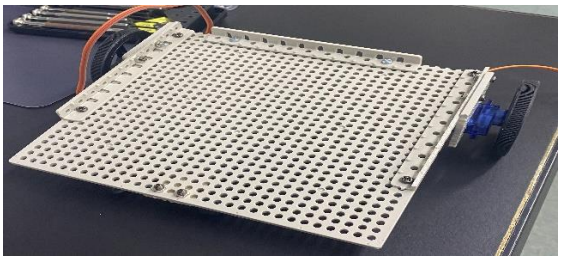
【内容】

①プログラムをマイクロビットに書き込む動作を使ってホイールを動かしたり、ブレッドボードとマイクロビットインジケーターを用いてマイクロビットを接続した。



②ユニバーサルプレートを使っての車体の製作

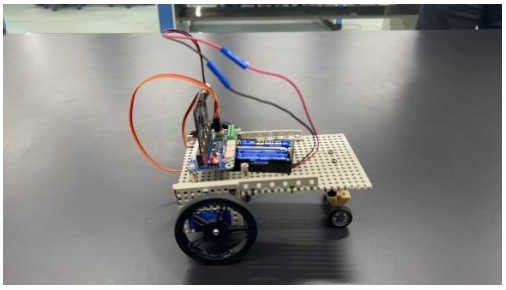
ユニバーサルプレートを切って、タイヤを取付。



③受信側のプログラム

④送信側のプログラム

⑤完成品



⑥感想・反省

：今回マイクロビットを用いてのラジコンカーの製作をしました。タイヤの取りつけたときタイヤの位置が低くなってしまい調整するのに苦戦しました。

：今回ラジコンカーのプログラムを中心に作りました。サーボモータでタイヤを制御するプログラミングに苦戦しました。プログラムに時間をかけすぎてしまい、完成が遅れてしまった。

Raspberry Pi を使用した多機能デジタルクロックの製作

3年 C 組

【目　的】
　Raspberry Pi を使用してニュースやカレンダーなどが表示できる多機能デジタルクロックを制作する。

【概　要】
　本研究では、コンピュータボードとして Raspberry Pi を使い、MagicMirror2 というアプリを使ってデジタルクロックを製作した。
　MagicMirror²は単純に時計を表示するだけでなく、YouTube の動画を背景として再生したり、ニュースやカレンダーなどの情報もモジュールを追加することで表示できるようになる。デジタルクロックのモジュールは CSS ファイルを操作することで追加できる。いくつかのモジュールを追加してアレンジを加えることで、オリジナルの多機能デジタルクロックを製作した。

【使用したもの】

名称	型番
Raspberry Pi 4 Model B	UD-RP4B
ディスプレイ	

【開発環境】

名称	規格
Raspberry Pi OS	Debian GNU/Linux 12
HTML ファイル生成ソフト	MagicMirror2
プログラミング言語	JavaScript, CSS

【研究の流れ】
① Raspberry Pi のセットアップ
② MagicMirror²をインストールする。
③ 天気や祝日の情報を時計に表示させる。
④ YouTube の動画を時計の背景に表示させる。

【Raspberry Pi のセットアップ】
　本研究では、Raspberry Pi というコンピュータボードを使用してデジタルクロックを製作する。このため、まず初めに Raspberry Pi のセットアップを行った。
　Raspberry Pi は SD カードから OS を起動する。作業用の PC を別に用意し、Raspberry Pi 財団のソフトウェアダウンロードページからダウンロードできる「Raspberry Pi Imager」を使うことで SD カードに OS をインストールできる。※1
　SD カードに「Raspberry Pi OS」をインストールしたら、Raspberry Pi 本体に SD カードをセットし、初期設定をすれば使用可能となる。
※1 https://www.raspberrypi.com/software/

【MagcMirror2 のインストール】
　MagcMirror2 は、もともと鏡の裏に液晶ディスプレイを仕込んでおいて、鏡を見ながら今日の予定を確認したり、天気や交通情報を表示したりする、近未来的な仕組みを作るためのソフトウェアです。構築環境やモジュールが揃っており、ディスプレイと組み合わせることで便利な多機能デジタルクロックになります。

【モジュールの導入方法】(MMM-Embed YouTube の場合)
MMM-Embed YouTube の GitHub ページにアクセスする

※2 https://github.com/nitpum/MMM-EmbedYoutube?tab=readme-ov-file

コマンドプロンプトに以下のスクリプトを順番に書き込んでモジュールをインストールする

```
cd ~/MagicMirror/modules
git clone https://github.com/nitpum/MMM-EmbedYoutube.git
cd MMM-EmbedYoutube
npm install
```

① /home/pi/MagicMirror/config/にある config.js ファイルに GitHub に記載されているモジュールを挿入する

Using this module

To use this module, add it to the modules array in the config/config.js file:

```
modules: [
  {
    module: 'MMM-EmbedYoutube', // Path to youtube module from modules folder Exmaple: MagicMirror/mo
    position: 'bottom_bar', // This can be any of the regions.
    config: {
      // See 'Configuration options' in README.md for more information.
      video_id: 'w3jLJU7DT5E',
      loop: true,
    },
  },
]
```

② config.js ファイルの video_id::に YouTube のリンク末尾にある ID を貼り付ける

https://www.youtube.com/watch?v=R34ot1LETK0
video_id

【結果】
・時計の表示
・ニュースの表示
・YouTube の表示
・YouTube 動画の全画面表示
・天気の表示
・カレンダーの表示
・複数ページの表示

【感想】
　思ったより早く完成形に近づいたものの細かい調整が難しくて苦労した。Twitter で似たようなものを見て実際に作ってみようと思ってから、実際に形になってくることがとても楽しかった。こんなに小さいパソコンでもできることはかなり多くて驚いた。

Unity を用いた FPS ゲームの制作

３年Ｃ組

1.【目的】

Unity で、美麗かつ軽量な FPS ゲームを制作する。C#やマテリアル、アニメーションの制御などの開発プロセスを包括的に学び、Unity を通して 3D ゲーム制作の基礎を身につける。

2.【使用機器・開発環境】

使用機器	バージョン・定格	メーカー
パソコン	CPU: Intel(R)Core(TM) i5-10500 3.10 GHz GPU: Intel® UHD Graphics 630 350 MHz RAM:8GB	富士通
OS	Windows 10 Enterprise LTSC	Microsoft
開発ソフト	Unity 2022.3.28f1	Unity Technologies

3.【内容】

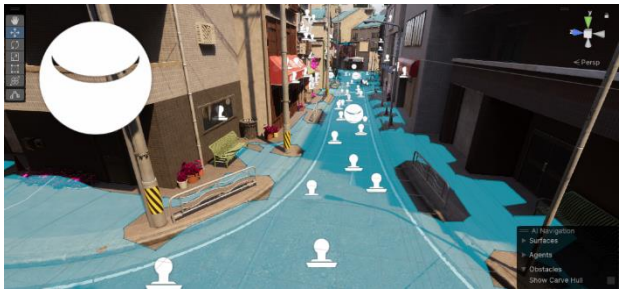


図 1 敵 NPC

3.1 敵の実装

ランダムで移動し、発見次第プレイヤーを攻撃する CPU を作成した。列挙型(enum)を使用してステートマシンを製作、Unity の標準機能である NavMesh を用いて、NPC の行動範囲を管理・制限した。

図 2 ゲーム映像

3.2 美麗なグラフィック

物理現象を忠実に再現する HDRP を使用することで、Unity で高解像度、高品質なゲームを実現した。

3.3 プレイヤーの制御

C#script を使用し、アニメーションやサウンドを包括的に制御。入力キーに基づき、アニメーションやサウンドを制御する関数を呼び出す。アニメーションは AnimatorController で管理し、スクリプトの関数内で実行される。サウンドは、インスペクター内で、任意のファイルをアタッチすることで再生される。さらに、CharacterController コンポーネントをアタッチすることで、当たり判定の制御が自動的に行われ、移動関連の制御を容易に行うことを可能とした。
実装済みの動作は歩行、走行、射撃、リロード、待機の 5 つであり、弾数・タイムは Canvas の TextMeshProUGUI コンポーネントで表示し、体力は Slider コンポーネントで管理した。

3.4 スポーン管理

スポーン時の確率を残り時間（変数）を基準に Update 関数内で Mathf.Lerp 関数を用いて設定することで、確率を段階的に変化させることができ、ゲームバランスを効果的に改善することが可能である。

3.5 一人称の滑らかな視点移動

Vector2(Unity の変数型)に mouse の X 軸と Y 軸の入力を格納し、メインカメラに反映させる。さらに、Mathf.Clamp 関数を使用することで、視点の移動範囲を-90～90 に制限。 最後に、メインカメラをプレイヤーの子オブジェクトにすることで、リアルな一人称を実現した。

3.6 銃の実装

assetstore で HDRP 対応の銃のモデルを入手し、Player の手の子オブジェクトにした。C#script であらかじめ Prefab 化した硝煙、45ACP、マズルフラッシュを呼び出し、サウンドを再生することで"銃"を実装した。本研究で使用した銃は M1A1 である。

4.【課題】

- 稀に空撃ちになるバグが存在し、アニメーションとエフェクトの紐付けが甘い。
- パフォーマンスの最適化が不十分であり、処理負荷の軽減が必要。
- Post Process Effects の理解が不十分で、HDRP を最大限に活かすことができなかった。

5.【感想】

ほぼ全ての動作を、C#script で制御したため、プログラミングについての理解が深まり、良い経験となった。

課題研究　マインスイーパの制作

３年Ｃ組

【目的】

以前の Windows に初期からインストールされていたようなマインスイーパを python によって製作する。

【使用機器】

名称	メーカ	型番	仕様
パソコン	arkhive	T3 PLUS	Windows 11 Home Ver. 23H2
マウス	ELECOM		
キーボード	ELECOM		

【内容】

図１は、セルを開く open_cell() メソッドである。

```python
    def open_cell(self, event):
        if not self.play_game:
            return
        label = event.widget
        for y in range(self.height):
            for x in range(self.width):
                if self.labels[y][x] == label:
                    j, i = y, x
        cell = self.cells[j][i]
        if label.cget("relief") != tk.RAISED:
            return
        text, bg, fg = self.get_text_info(cell)
        if cell == MINE:
            self.open_mine = True
        label.config(text=text, bg="lightgray", fg=fg, relief=tk.SUNKEN)  # 背景色を灰色に変更
        self.open_num += 1
        if cell == 0:
            self.open_neighbors(i, j)
        if self.open_mine:
            self.app.after_idle(self.game_over)
        elif self.open_num == self.clear_num:
            self.app.after_idle(self.game_clear)
```

図１　open_cell() メソッド

これは、左クリックでセルを開く処理である。セルを開いたときにそのセルが地雷であれば game_over() を呼び出してゲームオーバーにする。

図２は、game_over() メソッドである。

```python
    def game_over(self):
        self.open_all()
        self.play_game = False

        # ゲームオーバー用の新しいウィンドウを作成
        game_over_window = tk.Toplevel(self.app)
        game_over_window.title("ゲームオーバー")
        game_over_window.geometry("200x100")  # ウィンドウサイズを指定

        label = tk.Label(game_over_window, text="GAME OVER", font=("Arial", 16))
        label.pack(pady=20)

        # クローズボタン
        close_button = tk.Button(game_over_window, text="閉じる", command=game_over_window.destroy)
        close_button.pack(pady=5)
```

図２　game_over() メソッド

ゲームオーバーになった時に呼ばれるメソッドである。すべてのセルを開いて（open_all()）、ゲームが終了したことを示す。

図３は、open_all() メソッドである。

```python
    def open_all(self):
        for j in range(self.height):
            for i in range(self.width):
                label = self.labels[j][i]
                text, bg, fg = self.get_text_info(self.cells[j][i])
                label.config(text=text, bg=bg, fg=fg, relief=tk.SUNKEN)
```

図３　open_all() メソッド

ゲームオーバーまたはクリア時に、ボード上のすべてのセルを開くためのメソッドである。すべてのセルに対して、地雷があるか、周囲の地雷数を表示し、背景色などを変更して、最終状態を示す。

【実行結果】

図４・５は、実行結果である。

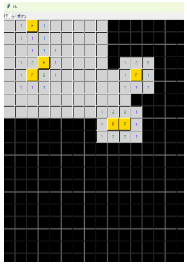


図４　実行結果①

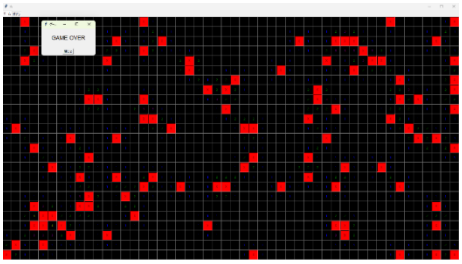


図５　実行結果②

ゲーム内の「Start」のボタンを押すと、全て載せるが黒くなる。地雷のないセルをクリックすると、セルが灰色に変わる。もしも、地雷があるセルをクリックするとすべての地雷の位置が赤色でわかり、ゲームオーバーとなる。また、「GAME OVER」の文字が表示される。

【感想】

もともと、製作しようとしていたものがうまくいかなかったために急遽内容を変更しマインスイーパを今回製作しましたが、完成予定の時期から遅れることなく、自分の目標通り完成させることができたため安心しました。課題研究はとても良い経験になったと思います。

発表番号７　　令和６年度

学校案内の制作

３年Ｃ組

1,【題目】

学校案内パノラマビューの作成

2.【目的】

学校案内をオンラインでできるようにする。

3.【研究内容】

(1)学校への許可申請

学校内にも映してはいけない場所などがあるため、学校長と副校長に時間をもらい、学校内撮影の許可をいただいた。

(2)使用機器

バージョン・定格	メーカー
Intel(R) Core(TM) i5-12450H	GALLERIA
CPU：2.00GHz	
メモリ　16 GB	
windows 11 home	Microsoft
Unity2022.3.33F1	Unity
RICOH THETA SC2 約 1400 万画素	RICOH

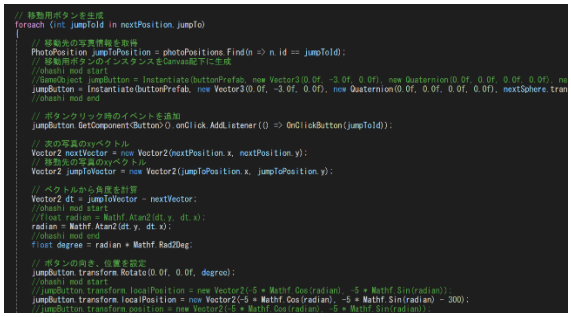
(3)撮影

カメラの向きや高さなどの配置、撮影時の周囲の明るさや撮影する間隔などを調節し、画像をきれいにつなぎ合わせられるように撮影をした

写真１．撮影の様子

(4)プログラム

画面効果や画像を変更できる球体を用意しプログラミングで紐づけられるようにした。またその球体と紐づけたメインプログラムに加え、画像と画像を切り替える際のジャンプボタンやその詳細設定を行った。



```
// 移動用ボタンを生成
foreach (int jumpToId in nextPosition.jumpTo)
{
    // 移動先の写真情報を取得
    PhotoPosition jumpToPosition = photoPositions.Find(n => n.id == jumpToId);
    // 移動用ボタンのインスタンスをCanvas配下に生成
    //ohashi mod start
    //GameObject jumpButton = Instantiate(buttonPrefab, new Vector3(0.0f, -3.0f, 0.0f), new Quaternion(0.0f, 0.0f, 0.0f, 0.0f), ne
    jumpButton = Instantiate(buttonPrefab, new Vector3(0.0f, -3.0f, 0.0f), new Quaternion(0.0f, 0.0f, 0.0f, 0.0f), nextSphere.tran
    //ohashi mod end

    // ボタンクリック時のイベントを追加
    jumpButton.GetComponent<Button>().onClick.AddListener(() => OnClickButton(jumpToId));

    // 次の写真のxyベクトル
    Vector2 nextVector = new Vector2(nextPosition.x, nextPosition.y);
    // 移動先の写真のxyベクトル
    Vector2 jumpToVector = new Vector2(jumpToPosition.x, jumpToPosition.y);

    // ベクトルから角度を計算
    Vector2 dt = jumpToVector - nextVector;
    //ohashi mod start
    //float radian = Mathf.Atan2(dt.y, dt.x);
    radian = Mathf.Atan2(dt.y, dt.x);
    //ohashi mod end
    float degree = radian * Mathf.Rad2Deg;

    // ボタンの向き、位置を設定
    jumpButton.transform.Rotate(0.0f, 0.0f, degree);
    //ohashi mod start
    //jumpButton.transform.localPosition = new Vector2(-5 * Mathf.Cos(radian), -5 * Mathf.Sin(radian));
    jumpButton.transform.localPosition = new Vector2(-5 * Mathf.Cos(radian), -5 * Mathf.Sin(radian) - 300);
    //jumpButton.transform.position = new Vector2(-5 * Mathf.Cos(radian), -5 * Mathf.Sin(radian));
```

写真２．ジャンプボタンの詳細設定

5.【まとめ・反省】

本当は学校全体をパノラマビューで作りたかったが、画像を増やすとジャンプ先がおかしくなってしまうなどのバグが発生してしまい。とぎれとぎれのものしか作れなかった。

作品は完成しなかったが私たちは都立の工業大会やドリームフェスタなどで行った発表を通して学べることが多かったのでとても有意義だった。

写真３．発表の様子

発表番号８　　令和６年度

クレーンゲームの製作

３年Ｃ組

【目　的】

今まで学んだプログラミングの知識を利用してモーターの制御を行えるようにする。

【使用機器】

	型番
レーザー加工機	FusionEdg24
3D プリンター	Adventurer 5M Pro
高速切断機	EM16
ボール盤	YDB-360

【使用部品】

名称	型番
ギアモーター	380 モーター

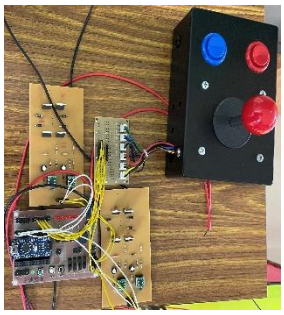
【開発環境】

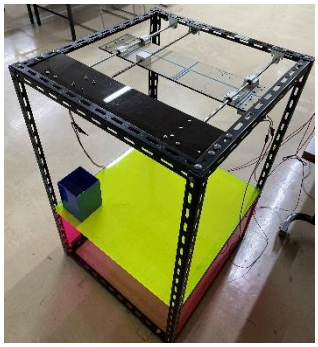
Ardoino IDE　Ver2.3.4
Flash Print 5
CorelDRAW
Epilog Dashboard
Autodesk Fusion 360

【製　作】

フレーム製作にはアングル材を使用した。また、アングル材の切断には、機械科の高速切断機を用いた。フレーム内部は二段の形に分けている。上部ではアームを操作して景品を取る場所になっており、下部は制御装置や電源等を入れるスペースを確保している。

アーム動作部分であるスライドは３Ｄプリンタの 600mm ガイドレールをＸ、Ｙ軸に２本使用した。動作にはＤＣモータを使用し、回転方向の制御にはｎチャネルＦＥＴのスイッチングＨブリッジで行った。





クレーンゲームフレームと動作部分

【プログラム】

制御には ArduinoIDE プログラム開発環境を使用した。アームをＸＹ軸の方向移動させるために switch～case 文によって処理を多分岐で８方向と停止の９種類 case 用意した。

```
switch(pattern){
  case 0: //STOP
    digitalWrite(8,LOW);
    digitalWrite(9,LOW);
    digitalWrite(10,LOW);
    digitalWrite(11,LOW);
  break;
```

制御プログラムの一部

【反　省】

アーム動作の動作は前後左右のみで、上下動作まで完成させられなかった。また、アーム単体まではできたが、本体に取り付けるところまでできなかった。役割分担は行えていたが、手順通りに作業を進めることができず、考えながらきちんと計画を立てる大切さがわかった。ぜひこの続きは後輩に完成させてもらいたい。

発表番号 9　　令和 6 年度

N ゲージを利用した車両の制御

3 年 C 組

【目的】

1,N ゲージの車両・レールの制御。

【内容】

N ゲージを使って小さい鉄道模型を作り、車両・レールの制御を目標に課題研究を進めた。

【これまでに行ったこと】

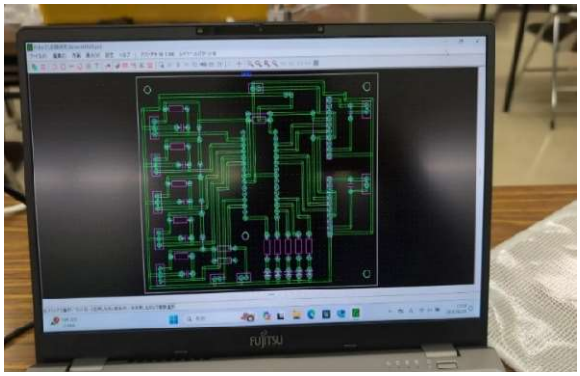
まず最初に土台となる部分を作るところから始めた。素材はアクリルにし、形を箱状にすることで後から作る基盤などの回路を入れることができるため、形は決まりました。

次に、N ゲージのレールや制御に必要なセンサーなどを購入しました。購入したものは以下の通りです。

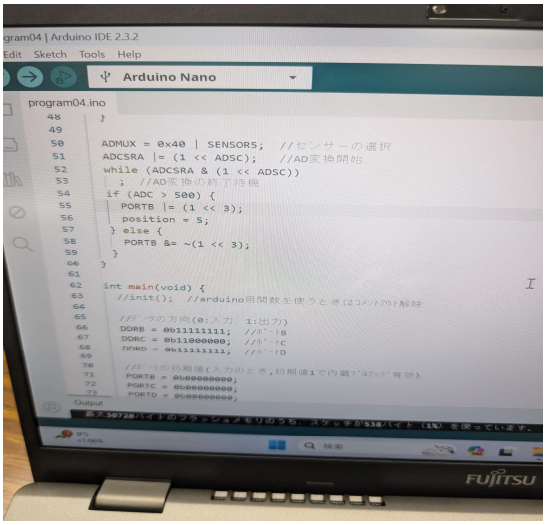
【購入・用意したもの】

コンパクト曲線線路 R117mm	反射型フォトセンサ RPR-220
コンパクト電動ポイント R150mm	フォト IC ダイオード 560mm
直線線路 124mm	LED チップ
直線線路 62mm	ホール IC SK8552GSIP-3
フィーダー線路 62mm	サーボモータ
実習用マイコンボード	人工芝　210mm × 297mm

次にホール IC という磁気センサーを各ポイントのレールの下に取り付ける。ホール IC は S、N 極関係なく磁石で反応するセンサーで、車両の下に磁石を取り付けることにより車両がどこを走っているのかを把握することができるため、磁気センサーを取り入れた。またサーボモータを使い、車両の正逆転の制御も取り入れた。最後に各制御を一つの基盤にするために PCBE を使い基盤の製図を行い、レールに取り付けた。基盤は以下の通りである。



最後に、LPKF protomat というプリント基板加工機を使って、基盤を作り各センサをつなげて、一通り行ってきたプログラムを組み合わせて一つの大きいプログラムにする。プログラムの例は以下の通りです。



また、人工芝を装飾として使うことで、見た目も良くなりました。

【反省】

本来は車両の制御の他に、外灯の制御をする予定だったが、最後までやり遂げることができなかった。だが、この 1 年間の課題研究が無かったら N ゲージに触れる事も無かったし、自分が思う理想の形にはならなかったけど、最終的にはとてもためになったと思います。

Music Game の作成

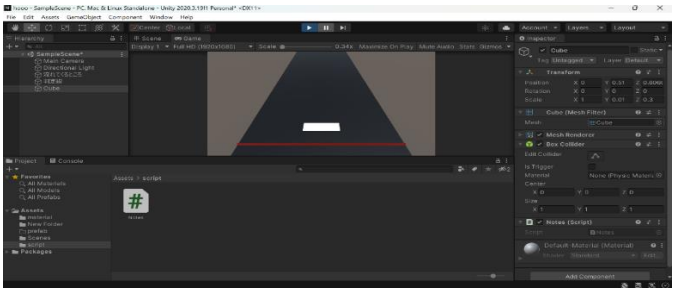
3C

【目的】
・Unity を使ってゲーム制作をする。
・Music Game の仕組みを理解する。
・譜面の作成方法について学ぶ

【開発環境】

ゲームシステム作成用 PC1	AMD2.1Ghz Windows10
ノーツ作成用 PC	Intel2.4Ghz Windows11
Unity 用プラグイン	Note Editor
Unity バージョン	2022.3.22f1

【Unity】
　自分のパソコンに Unity をインストールする。その後バージョンで 2021.3.21 を使用。

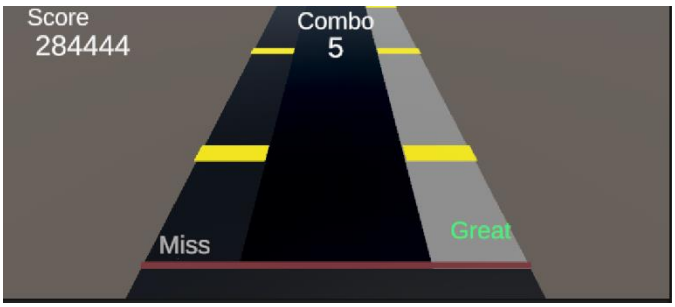


【レーンとノーツと譜面の作成】
ノーツとは音に合わせてプレイ画面の上方から流れてくる白色のブロックで、タイミングよくキーを押すための目印である。また、ノーツは 4 つに区切られたレーンを流れてくる。ノーツを流すための譜面データは Note Editor というアプリを使って作成する。Note Editor によって、ノーツの位置、量、長さを決めることができる。wav 形式の音楽ファイルを読み込み、それをもとに譜面データを作成していく。

　上の図は譜面データ作成の様子である。画面下部の波形は音楽データから取り込まれた音声を表している。
波形が大きく揺れている個所でリズムをとるようにノーツを置いていく。

【譜面の読み込み】
Unity のゲームデータに譜面を読み込むには、JSON ファイルが必要となる。JSON ファイルは Note Editor から出力できる。

【判定とスコア計算】
　赤色の判定線の上でノーツがタイミングよく押されたときにポイントが加算されるようにした。
　判定は Perfect、Grate、Bad、Miss の 4 つとした。それぞれ押すべきタイミングとの時間のずれをプログラムで計算し、それぞれの判定を行っている。
　プレイ画面の上部にはスコアとコンボ数を表示している。コンボには perfect または Great と連続して判定された回数を表示する。Bad や Miss が出た時にはコンボの数はリセットされる。スコアの計算はそれぞれの判定の数、コンボ数などで決まる。



【まとめ】
・Music Game に必要な機能を実装することができた。
・バグなどが多かったのでプログラムの仕様を変えた。
・ノーツを流す際に音ズレが多かったので曲が流れるタイミングをずらしたりして対応した。

【感想・反省】
・最初のバグ修正にとても時間がかかってしまい思ったよりも作るのに時間がかかってしまい大変だった。
文化祭までには完成できたので良かった。
・最初は音ズレ、難易度調整、ノーツのパターンがひどかったので見直しをして軽減することができたこと。文化祭では皆様に満足してもらえたのでよかった。

発表番号 11　　令和 6 年度

ペットボトルフィラメント製造機の製作

3 年 C 組

1、目的

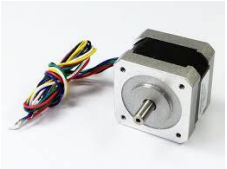
・3D プリンタで使用できる実用的なペットボトルフィラメントを作る

・ペットボトルをリサイクルできるものを作る

2、内容

① 使用機器・部品

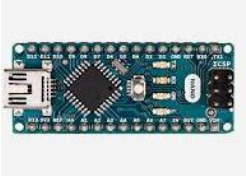
名称	型番	メーカー
12V 8.5A 電源	LRS-100-12	MEAN WELL USA inc
ｽﾃｯﾋﾟﾝｸﾞﾓｰﾀ	ST-42BYG0506H	MERCURY MOTOR
ｽﾋﾟｰﾄﾞｺﾝﾄﾛｰﾗ	Arduino Nano	Gravitech 社
FET	2SJ681	TOSHIBA
温度ｺﾝﾄﾛｰﾗﾎﾞｰﾄﾞ	YYW-2	LanGuShi
ﾋｰﾄﾌﾞﾛｯｸｾｯﾄ		DAGIJIRD
3D プリンタ	Adventurer 5M Pro	FLASHFORGE
ボール盤		
ﾌﾟﾘﾝﾄ基盤加工機	ProtoMat S62	LPKF
レーザー加工機	FusionEdge24	EpilogLaser



←ステッピングモータ

温度ｺﾝﾄﾛｰﾗﾎﾞｰﾄﾞ→

ｽﾋﾟｰﾄﾞｺﾝﾄﾛｰﾗ→



② 開発環境

・Autodesk Fusion 360（3D cad）

・FlashPrint 5（スライサソフト）

・jw_cad（2 次元 cad ソフト）

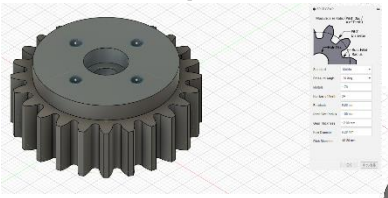
・CorelDRAW（ベクトル画像編集ソフト）

・Arduino IDE 2.3.2（プログラム開発環境）

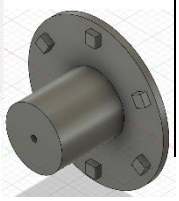
③ 作業工程

(1) Fusion 360 と 3D プリンタを使用し、使用機器以外の必要な部品を製作する。

↓Fusion 360 の SpurGear でギア(小)を作成

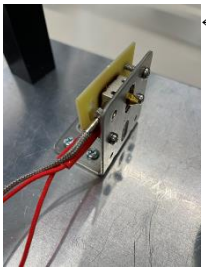


車輪とギア(大)を固定する部品→



(2) ボール盤を使用し、土台に部品を固定するための穴を空ける。

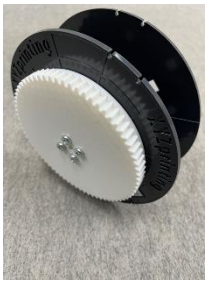
(3) 部品の組み立て



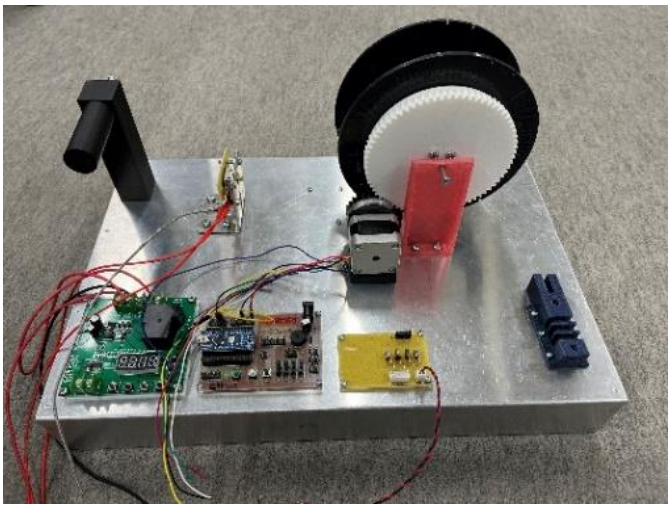
←ヒートブロックの固定

ステッピングモータにギア(小)を固定→

車輪とギア(大)→を固定



(4) 必要な配線やプログラミングをする



完成写真

3、課題点

・温度コントローラーボードの制御

・ステッピングモータの制御

・土台上での配線

・ペットボトルの加工方法

4、反省・感想

・自分のペースを考慮して居残りをするかしないかの判断をしたほうが良いと思いました。

・温度コントローラーボードの制御、ステッピングモータの制御が難しかったです。

・配線の仕方は工夫したほうがいいと思いました。

発表番号１２　　令和６年度

課題研究　エアホッケーの制作

３年Ｃ組

【目的】

・課題研究を通してものづくりを体験し、それがどんな事なのかを学ぶ。

【使用機器】

①　UP BOX+

造形方式	FDM（熱溶解積層法）
造形マテリアル	ABS 樹脂、ABS+樹脂、PLA 樹脂
プリントエリア（mm）	255(幅) x 205(奥行き) x 205(高さ)
積層ピッチ（mm）	0.05 / 0.10 / 0.15 / 0.20 / 0.25 / 0.30 / 0.35 / 0.40

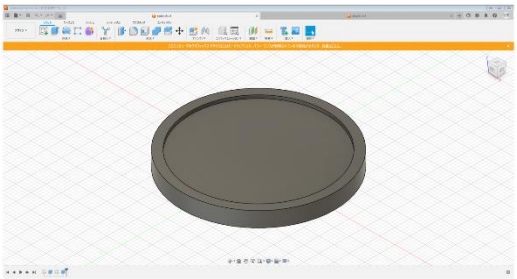
②Adventurer 5M Pro

エクストルーダー	ダイレクト式
造形速度	造形速度：10 - 300mm/s 最高移動速度：600mm/s
最大造形サイズ	220×220×220mm
対応フィラメント	PLA / PETG / ABS / ASA
ノズル径	0.4mm（デフォルト）/ 0.6mm / 0.8mm / 0.25mm

③アプリケーション

・Autodesk Fusion 360

Autodesk Fusion は、オートデスク社が開発しているクラウドベースの 3D CAD、CAM、CAE ソフトウェアである。3D モデル、プリント基板、2D・3D 図面の設計が可能。



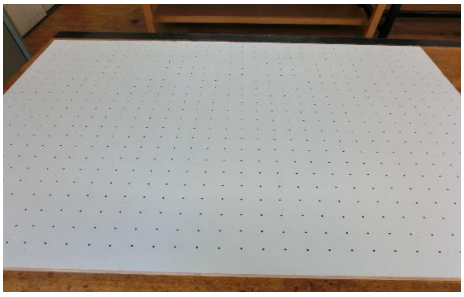
【内容】

・デザイン

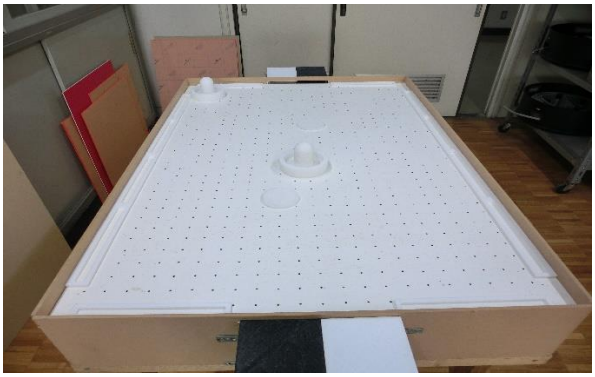
①　大きさを決める

②　切断

③　穴あけ



④　組み立て



・3D プリンターで制作したもの

①　パック

②　マレット（打つやつ）

③　パックのゴール

【感想】

楽しさと達成感を感じました。自分でゲームを作ることで、物理的な仕組みやデザインを考える楽しさを実感しました。エアホッケーの台を作る過程では、風を均等に吹かせるための工夫や、パックがスムーズに滑るようにするための調整が重要だと感じました。最初はうまくいかないこともありましたが、改善を重ねることで完成度が上がり、遊べる状態になったときの満足感は格別でした。

発表番号１３　　令和６年度

電子ピアノの製作

3C

【研究背景】

これまでに学習したプログラミングや回路などを応用し、電子ピアノを作ろうと考えた。また、ピアノの鍵盤には、実習で使用しているタクトスイッチではなく、PC のタッチパネルのように触るだけで入力できるものを使ってみたいと考えた。

【使用機器】

品名	規格
マイコンボード	Arduino Nano
PC	Windows 11 Pro
音源 IC	YMZ294-D
水晶発振子	4.9152MHz
スピーカー	8Ω ダイナミック型
シフトレジスタ	74HC595
静電容量センサ	TTP223

・音源 IC とは

ヤマハ製の音源 IC で、ゲームや電子楽器で使われる。マイコンのプログラムによって３つの音を同時に発音できる。

インターネット上の情報を参考にして、制御用の関数を用意して使用した。(https://blog-e.uosoft.net/entry/20160703/1467476171)

```
周波数設定関数
void SetFrequency(int ch, word freq)
```

```
レジスタセット関数
void set_register(byte addr, byte value)
```

・シフトレジスタとは

シフトレジスタとは、少ない数のピンで多くのデータを扱うための部品。音源 IC にたくさんのピンが必要になるので使用した。

・静電容量センサとは

静電容量センサは、指とセンサの間に発生する微弱な静電容量の変化から入力を検知できる。タクトスイッチと同様に使うことができる。

【プログラム】

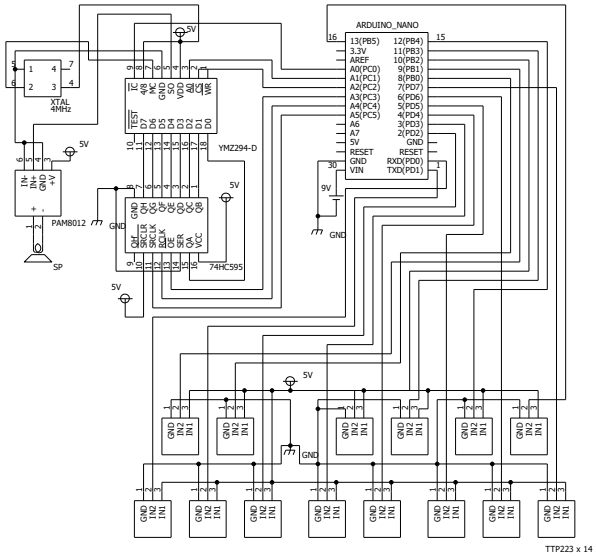
#define 指令を使って、プログラムがわかりやすくなるように工夫した。

```
#define do4 261.626     //ド
#define dok4 277.183   //#ド
#define re4 293.665     //レ
#define rek4 311.127   //#レ
```

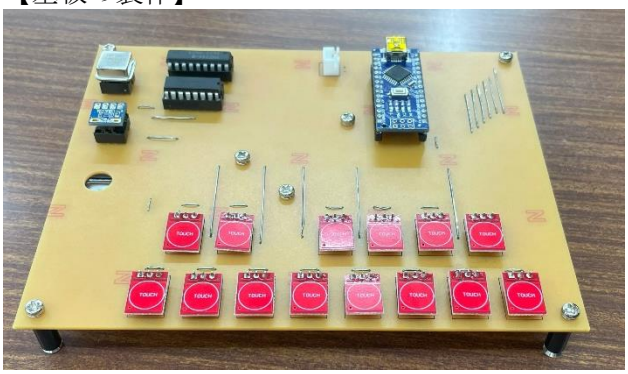
13 個の鍵盤のほかに、音階切り替えスイッチを追加した。音階を切り替えると変数 oct によって周波数が 2 倍になり、1 オクターブ上の音になる。

```
if (SW & (1 << 0)) {
        setFrequency(1, do4 * oct);
        set_register(0x09, 7);
} else if (SW & (1 << 1)) {
        setFrequency(1, re4 * oct);
        set_register(0x09, 7);
}
```

【回路設計】



【基板の製作】



【感想】

・YMZ294 の情報があまり見つからなかったので苦労した。

・学校に使用したい部品がそろっていたので、プログラム作成まで早く進めることができた。

・回路の設計が大変で、端子をつなぐ線が他の線と交差しないようにするのが難しく、時間がかかってしまった。

・和音を鳴らすプログラムが完成できなかった。

課題研究　運転シミュレーター制作

3 年 C 組

【目的】

運転シミュレーターのアプリを自分たちで作りたいと感じたことがきっかけである。

【使用機器・環境】

このような機器や環境で開発を行った。

- パソコン
  GALLERIA RM5C-R36T
  Windows11 Home 64bit　バージョン 23H2
  CPU ： 13thGen　lntel(R)　Core(TM)　i5-13400F 2.50GHz
  RAM：16.0GB
- キーボード
  ELECOM
- コントローラ
  PS3 コントローラ　CECHZC2J
- Unreal Engine 5　バージョン 5.4.2

【研究内容】

本制作では、Unreal Engine 5 を使った運転シミュレーターの作成をした。

Unreal Engine 5 での既存の車のスペックでは本来の車の動きをしていなかった。そのため、実物に近づけるために以下の修正を行った。

- 車のギア変速
- 最高時速の変更
- 曲がり方やブレーキの利き方の修正

また、コース作成を以下の通り行った。

コースの全体像は図 1 のような山を想定した。

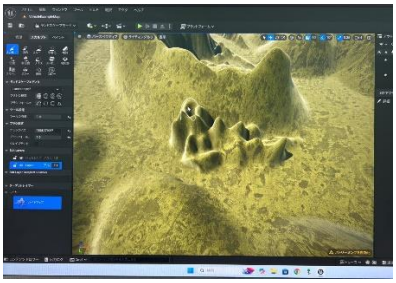


図 1 コースの全体像

コースの作成は図 2 のようにアスファルトを敷き詰め実物の道路になるように作成した。また、その上で図 3 のように木や石などを置いて実物の山になるように作成した。

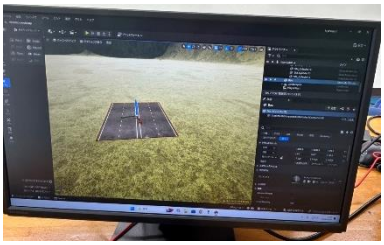


図 2 コースの作成

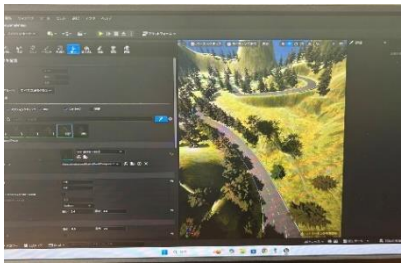


図 3 木や石の配置

【操作方法】

操作は表 1 のように行う。

動作	キー
前進	W
左	A
右	D
バック	S
ギアをあげる	LSHIFT
ギアを下げる	LCTRL
サイドブレーキ	SPACE

表 1 車の操作方法

【感想】

1 年間を通して役割分担がうまくできず、一人に作業が集中してしまったため、もう少し協力して行うべきだった。

当初予定していた PS3 コントローラで動かすことができなかった。

作品全体は現実のように近づけられたと思ったが、もう少しバグなどを減らして動かせるようにしたかった。