

```
1 using UnityEngine;
2
3 /// <summary>
4 /// ステージがスタートしてからクリアするまでのタイム 計測 クラ
5 /// </summary>
6 public class Timer : MonoBehaviour
7 {
8     private float counter;
9     private bool isCounting;
10
11     // 初期化処理
12     public void Start()
13     {
14         counter = 0.0f;
15         isCounting = false;
16     }
17
18     // 更新処理
19     public void Update()
20     {
21         // 計測
22         if (isCounting == true)
23         {
24             counter += Time.deltaTime;
25         }
26     }
27
28     // カウンターが有効かどうかを設定する関数
29     public void SetCounterEnable(bool flg)
30     {
31         isCounting = flg;
32     }
33
34     // カウンターの値を取得する関数
35     public float GetCounter()
36     {
37         return counter;
38     }
39 }
```

# PORTFOLIO



HAL東京  
昼間部 ゲーム4年制学科

向山陸登

# PROFILE



志望職種： クライアントエンジニア

氏 名： 向山 陸登

生 年 月 日： 2002年6月25日

趣 味： 散歩・料理・アニメ鑑賞

G m a i l： mukoyama2002@gmail.com

作品データ： <https://bit.ly/4bG3913> 

## － 資格 －

2018/12 - マルチメディア検定 ベーシック 合格

2021/12 - CGエンジニア検定 ベーシック 合格

2022/07 - 情報処理活用能力検定 2級 合格

2023/08 - 基本情報技術者試験 合格

## － 言語・ツール 習熟度 －

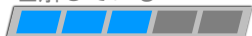
触れたことがある



なんとなく理解している



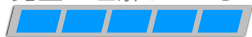
理解している



ほとんど理解している



完璧に理解している



C



DirectX



Blender



C#



Unity



Maya



C++



UnrealEngine



GitHub



HLSL



VisualStudio



PlasticSCM







# 回せ！ぺん忍道畳

制作期間 3ヶ月

制作環境 Unity 2021.315f1/C# VisualStudio 2017 Plastic SCM

動作環境 Windows10 Professional 64bit

制作メンバー プログラマー X 2 レベルデザイナー X 2  
グラフィッカー X 4 サウンドエフェクト X 2

受賞履歴  ゲーム大賞 アマチュア部門 一次審査通過  
 ゲームクリエイター甲子園 デジタルハーツ賞 受賞

担当箇所 タイトル、チュートリアル、一部ギミック、一部システム



コンセプトアート



## - 概要 -

“畳を回転して道を作り、ゴールを目指す”パズルアクション。  
パズルが苦手な人も楽しめるよう、丁寧なレベルデザインや  
可愛いキャラクターにこだわりました。



## アピールポイント「インスペクターにもコメントを表示」

ScriptにインスペクターでPrefabやSceneなどを登録する際、  
**何処に何を登録すればいいのか**分かりづらく毎回Scriptのコメントを確認してました。  
Scriptを確認するの手間を省くため、**Tooltip**と**Header**を利用しました。

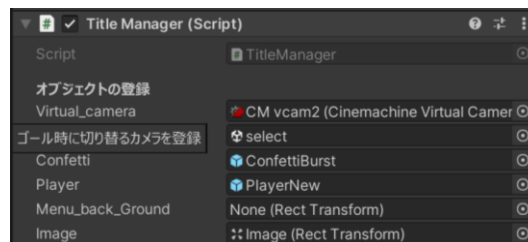
**Header**は常にテキストが表示されるのに対し、  
**Tooltip**は特定の変数をマウスオーバーした時だけテキストが表示されます。

**Tooltip**&**Header**の存在をチームメンバーに共有した結果…  
→チームメンバーに好評、何処に何を登録するか分かりやすい

### ソースコード▼

```
[Header ("オブジェクトの登録")]  
[Tooltip ("ゴール時に切り替えるカメラを登録")]  
[SerializeField] private CinemachineVirtualCamera virtual_camera;
```

### インスペクター▼



## アピールポイント「ポストプロセスでグラフィック強化」



ポストプロセスなし



ポストプロセスあり

ゲーム内の要素が揃い始め完成に近づき始めたころ、  
試遊してもらったプレイヤーから**面白いけど物足りない**  
という意見をもらいました。

チームメンバーと話し合っていくうちに、**画面が単調**なのが  
原因かもしれないという結果に至りました。

グラフィック向上の方法を調べるうちに、**ポストプロセス**という  
処理があるということを知りました。

### 役に立ったポストプロセス2選

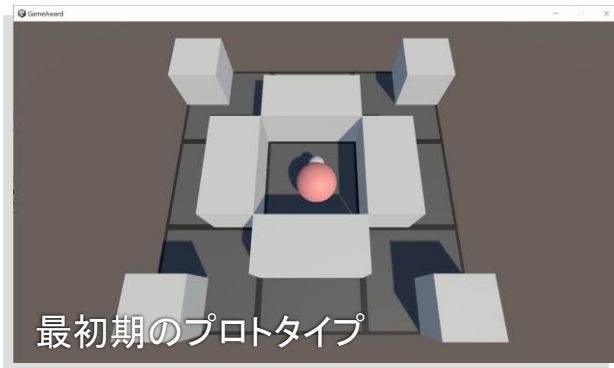
#### Color Grading

色味を調整し、畳の色を強調する

#### Depth Of Field

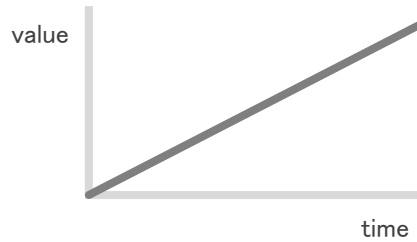
背景をぼかし、ステージに注目してもらう

## アピールポイント「細かい変化で触り心地を良くする」



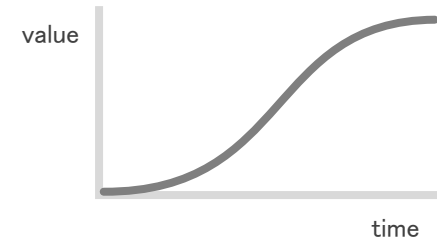
(プロトタイプでの処理)

**Transform.Rotate**



(マスター版での処理)

**Ease.InOutQuart**



量の回転をTransform.RotateからDOTweenのEase.InOutQuartに変更、気持ちよい操作感を実現しています。



制作期間 1ヶ月

制作環境 C++  
DirectX  
VisualStudio

動作環境 Windows10



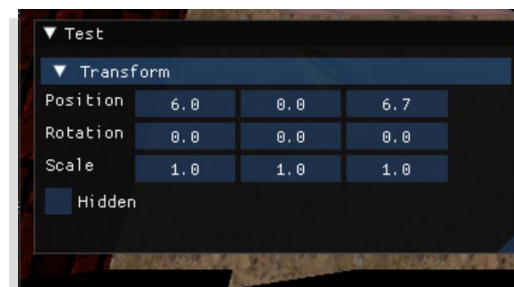
## - 概要 -

市販のゲームを参考に、自動生成される迷路を実装

## アピールポイント「デバッグ機能を実装して気付いた処理順の大切さ」

プレイヤーの座標をリアルタイムで表示する為imguiを導入、  
プレイヤーの座標とimguiの値を同期するのに苦労しました。

- 不具合1.Imguiの座標を変えてもプレイヤーが動かない
- 不具合2.プレイヤーが動いてもimguiの座標が変更しない
- 不具合3.Imguiの座標をそもそも変えることが出来ない
- 不具合4.プレイヤーがそもそも動けない



```
vec4a[0] = player->GetPosition().x;  
vec4a[1] = player->GetPosition().y;  
vec4a[2] = player->GetPosition().z;
```

```
if (ImGui::CollapsingHeader("Transform", ImGuiTreeNodeFlags_DefaultOpen))  
{  
    ImGui::Text("Position");  
    ImGui::SameLine();  
    ImGui::DragFloat3("##Position", vec4a, 1.0, 0.0, 0.0, "%.1f");  
  
    //プレイヤーの値を更新  
    player->SetPosition(vec4a);  
}
```

## 実装手順

float x,y,zからなる構造体vecを用意  
Playerクラスのポジションをゲッターで呼び出し、vecに代入  
ImGuiの関数DragFloat3の第二引数にvecを指定  
プレイヤーのセッターにvecを渡す



制作期間 1週間

制作環境 C++  
DirectX  
VisualStudio

動作環境 Windows10



ゲーム制作の息抜きで挑戦した地形生成。  
あらかじめ処理を実装することで今後の制作で必要になった場合、  
すぐに使用できるよう心掛けました。

まず初めに通常の乱数を用いて制作したものの、  
ブロックの場所はバラバラで、地形とは呼べませんでした。  
乱数について調べていく中でパーリンノイズという技術を知りました。 }

パーリンノイズのイメージ▼



## - 概要 -

自然な地形を生成するツール、凹凸具合の調整可能。  
パーリンノイズという技術を使用。

ソースコード▼

```
double PerlinNoise::noise(double x, double y) const {  
    int X = static_cast<int>(std::floor(x)) & 255;  
    int Y = static_cast<int>(std::floor(y)) & 255;  
  
    x -= std::floor(x);  
    y -= std::floor(y);  
  
    double u = fade(x);  
    double v = fade(y);  
  
    int a = p[X] + Y;  
    int b = p[X + 1] + Y;  
  
    return lerp(v, lerp(u, grad(p[a], x, y), grad(p[b], x - 1, y)),  
        lerp(u, grad(p[a + 1], x, y - 1), grad(p[b + 1], x - 1, y - 1)));  
}
```





制作期間 1ヶ月  
制作環境 Unreal Engine 5  
制作メンバー プログラマー X 2  
プランナー X 1  
グラフィッカー X 2  
動作環境 Windows10

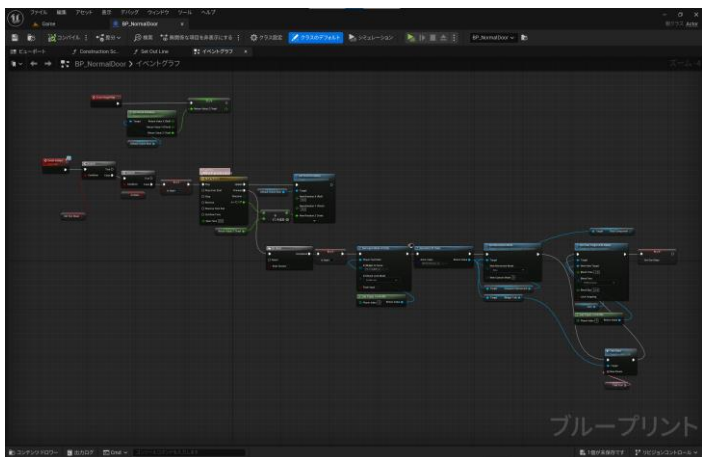


## - 概要 -

一人称視点  
謎解きメガネ  
脱出ゲーム

## Point1.ブループリントを活用

ギミックやウィジェットを担当



## Set View Target with Blend

カメラの切り替えを行う

カメラA



滑らかに遷移

カメラB





お忙しいところ最後までご覧いただきありがとうございます。

私の強みは常に向上心を持ち続けていることです。

羽田空港で手荷物を運ぶアルバイトをはじめた当初、お客様の問い合わせが自分の担当範囲外で答えられず、心残りだったことがありました。

お客様の立場になって考えると担当範囲などとは関係なく、どんな質問にも回答してもらえることが満足度につながると思った為、これを機に出来る限り多く質問に対応できるよう、搭乗手続きの方法や搭乗口の場所など空港のあらゆることを時にはスタッフに教わり調べました。

クライアントエンジニアになった際は現状に満足せずどんな時でも向上心を持ち、チームメンバーと円滑にコミュニケーションを取るためプログラム以外の業務も積極的に学び、開発が迅速に進むよう貢献したいと思います。

HAL東京

昼間部 ゲーム4年制学科

向山陸登