

```
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・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

## － 経歴 －

2015/09－初めてのプログラミングに触れる

2018/04－N高等学校 入学

2018/04－バンタンゲームアカデミー 入学

2021/03－N高等学校 卒業

2021/03－バンタンゲームアカデミー 卒業

2021/04－HAL東京 昼間部 4年生課程 ゲーム4年制学科 入学

2024/02－HAL東京 昼間部 4年生課程 ゲーム4年制学科 3年 在学中

作品データはQRコードからも観覧できます



# - PROFILE -

## - 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 ■■■■■

C# ■■■■■

C++ ■■■■■

HLSL ■■■■■

DirectX ■■■■■

Unity ■■■■■

UnrealEngine ■■■■■

VisualStudio ■■■■■

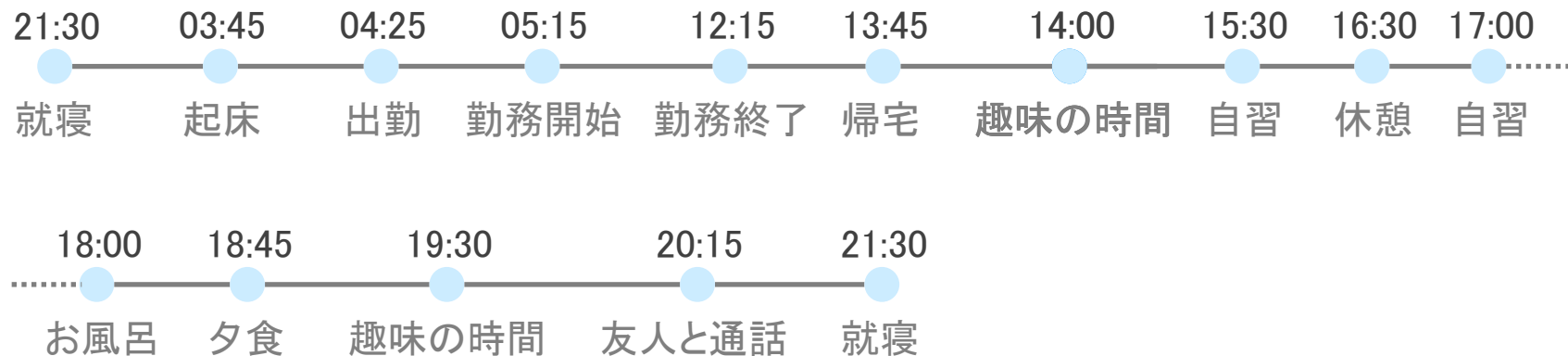
Blender ■■■■■

Maya ■■■■■

GitHub ■■■■■

PlasticSCM ■■■■■

## - 休日のスケジュール -




# 回せ！ぺん忍道畳

制作期間 3ヶ月

制作環境 Unity 2021.3.15f1/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



## - 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。



# 回せ！ペン忍道畳

## Point1.プログラミングでの工夫

### TitleManager.sc(一部抜粋)

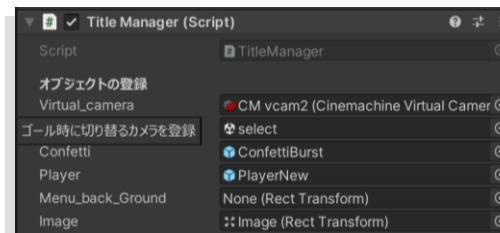
```
/// <summary>
/// タイトルシーンを管理するスクリプト
/// </summary>
public class TitleManager : MonoBehaviour
{
    //管理するスクリプト
    private ClearManager clear_manager;
    private Fade fade_manager;
    private GameAward input_manager;
    private SaveManager save_manager;
    private Sound sound_manager;

    [Header("オブジェクトの登録")]
    [Tooltip("ゴール時に切り替るカメラを登録")]
    [SerializeField] private CinemachineVirtualCamera virtual_camera;
    [SerializeField] private SceneObject scene;
    [SerializeField] private GameObject confetti;
    [SerializeField] private GameObject player;

    :
    :
```

### TooltipやHeaderの活用

[Header("オブジェクトの登録")]  
[Tooltip("ゴール時に切り替るカメラを登録")]



### フェード中か否か気にせず利用可能

```
public void FadeOut(Color color, float speed)
{
    if(state == FadeState.None)
    {
        state = FadeState.FadeOut;
        fadeColor = color;
        fadeSpeed = speed;
    }
}
```

### Fade.sc

```
void Update()
{
    switch(state)
    {
        case FadeState.None:
            break;
        case FadeState.FadeIn: //徐々にゲーム画面が表示される
            fadeColor.a -= Time.deltaTime / fadeSpeed;
            if (fadeColor.a <= 0.0f)
            {
                state = FadeState.None;
                fadeColor.a = 0.0f;
            }
            break;
        case FadeState.FadeOut: //徐々にゲーム画面が表示されなくなる
            fadeColor.a += Time.deltaTime / fadeSpeed;
            if (fadeColor.a >= 1.0f)
            {
                state = FadeState.None;
                fadeColor.a = 1.0f;
            }
            break;
    }

    image.color = fadeColor;

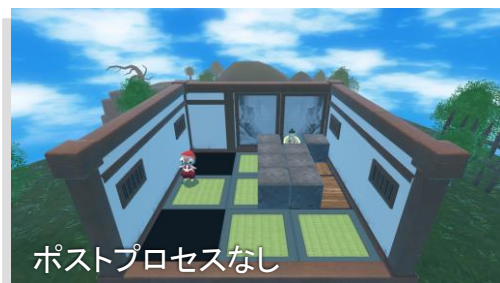
    //もしフェード中でなければスタートをフェードアウトに変更
    public void FadeOut(Color color, float speed)
    {
        if(state == FadeState.None)
        {
            state = FadeState.FadeOut;
            fadeColor = color;
            fadeSpeed = speed;
        }
    }

    //もしフェード中でなければスタートをフェードインに変更
    public void FadeIn(Color color, float speed)
    {
        if (state == FadeState.None)
        {
            state = FadeState.FadeIn;
            fadeColor = color;
            fadeSpeed = speed;
        }
    }
}
```

# 回せ！ペン忍道畳

## Point1.グラフィックの強化

グラフィックの表現を向上させる為、ポストプロセスを追加



### Color Grading

色味を調整し、畳の色を強調

### Depth Of Field

背景をぼかし、ステージに注目

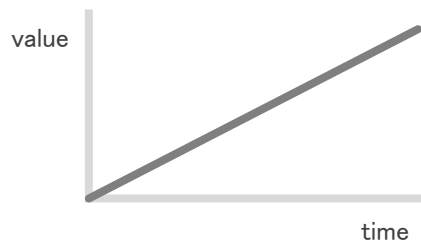
## Point2.操作感の改善

畳の回転をTransform.RotateからDOTweenのEase.InOutQuartに変更、気持ちよい操作感を実現



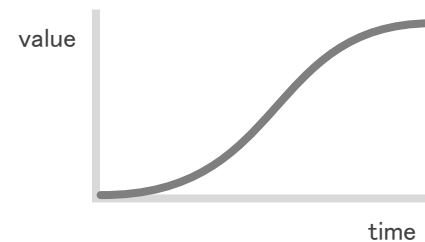
(プロトタイプでの処理)

Transform.Rotate



(マスター版での処理)

Ease.InOutQuart





# ダンジョン25

制作期間 1ヶ月  
制作環境 DirectX  
C++  
動作環境 Windows10



## - 概要 -

遊ぶたびに姿を変える不思議な迷宮。  
敵に捕まらないよう気を付けながら迷宮の最深部、  
“25層”までたどりつくのが目的。

# 自然な地形生成

制作期間 1週間  
制作環境 DirectX  
ImGui  
C++  
動作環境 Windows10



## - 概要 -

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

# 奇妙なメガネハウスからの脱出

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

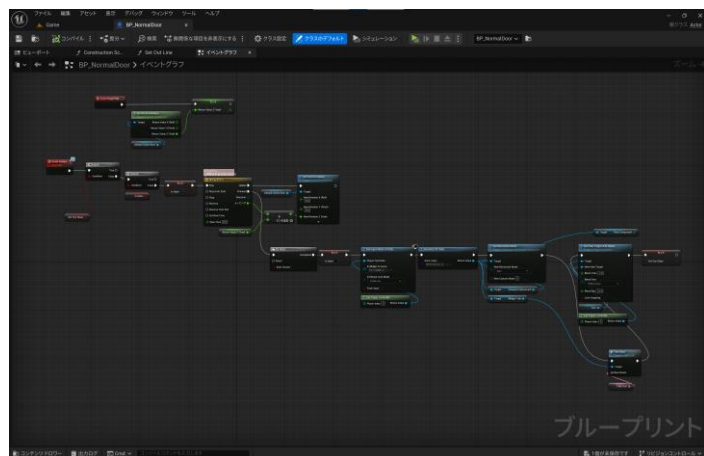


## - 概要 -

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