

ポートフォリオ

総合学園ヒューマンアカデミー横浜校
ゲームカレッジプログラマー専攻

鈴木海凧

プロフィール

氏名 : 鈴木海凧(19)
生年月日 : 2004年9月3日
性別 : 男
希望職種 : プログラマー

所持スキル

言語

C/C++, C#, ActionScript,
HTML/CSS, PHP



開発環境

VisualStudio, Unity,
Xampp



Unity®

開発支援ツール

GitHub, Adobe系列

GitHub

自己PR

私は、諦めずに続けることを意識しています。
プログラムは分からない事の方が多いですが、なんとか理解していこうと諦めずに頑張っております。
制作においては、与えられた役割をとにかくこなすことを意識して行いました。
現在はDirectXを学習しているため、そちらも頑張りたいと思っています。
今後は、様々なことに対応できるように、もっと知識を蓄えていきたいと思っております。

目次

Unity制作作品

2Dアクション

- ・ Frustration Stick P.4

3D脱出ホラー

- ・ chamber of sin P.5

Animate制作作品

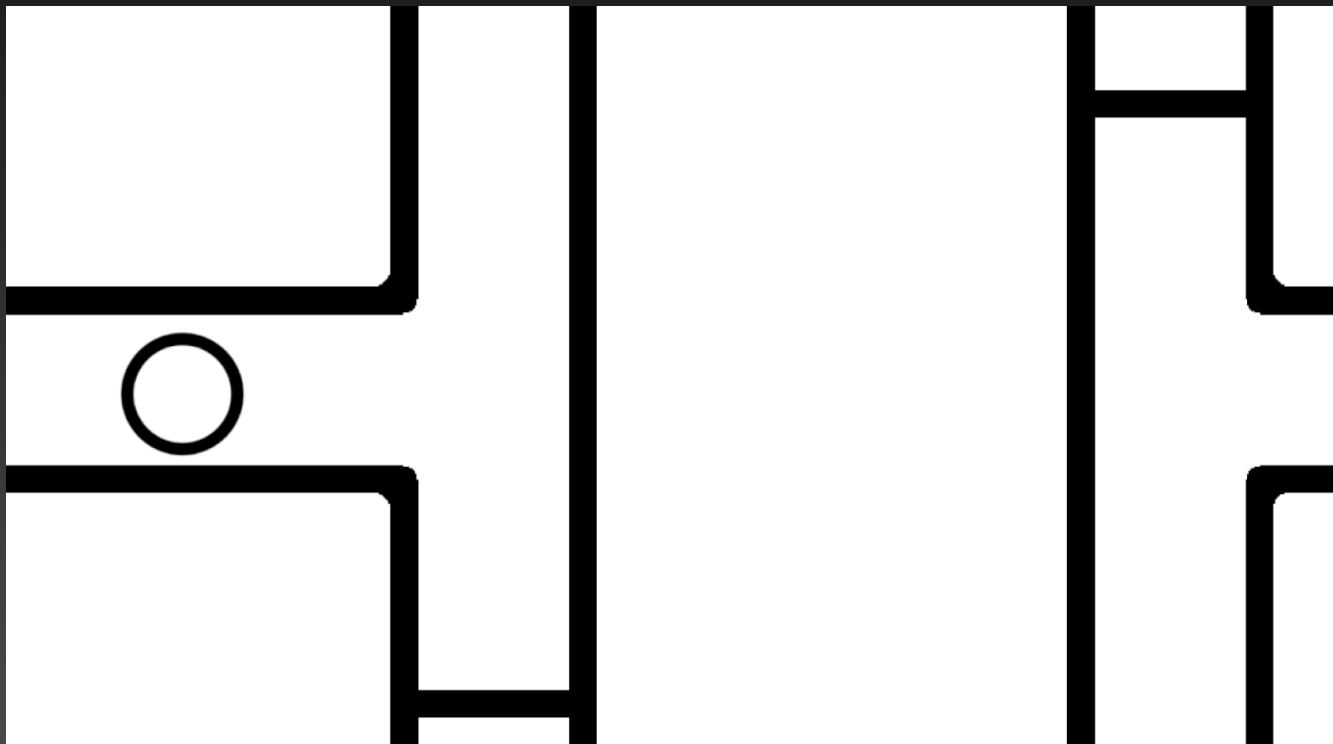
アクション

- ・ イライラ棒 P.9

シューティング

- ・ 的当て P.10

Frustration Stick



ジャンル：2Dアクション

制作人数：1人

制作期間：1週間

使用開発環境：Microsoft Visual Studio 2022,
Unity, GitHub

説明

キーボード操作で、自機(画像の丸)を操作しながら、壁に当たらないようにゴールを目指すゲームです。

Unityチーム制作作品

chamber of sin



作品名 : **chamber of sin(チェンバーオブシン)**
ジャンル : **3D脱出ホラー**
使用言語 : **C#**
制作人数 : **CGデザイナー : 2名**
 プログラマー : 3名
制作期間 : **3ヶ月**
担当箇所 : **一部ギミック、オブジェクト取得等**
使用開発環境 : **Microsoft Visual Studio 2022,**
 Unity, GitHub

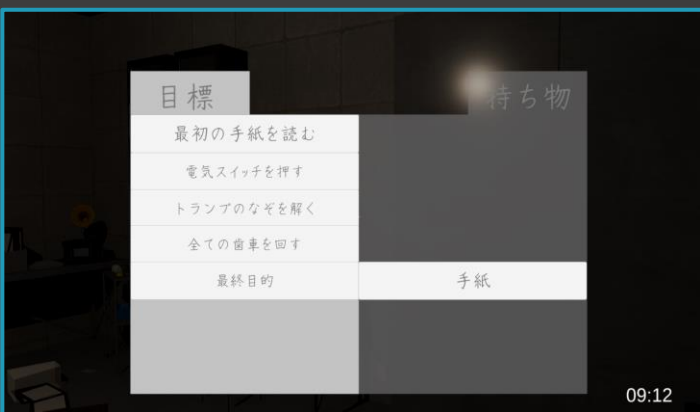
ゲーム説明



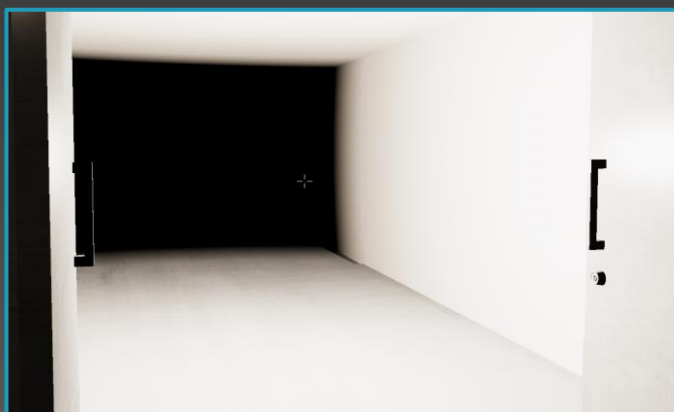
主人公が目覚めるとそこは目の前には一枚の紙がおいてあり10分以内に脱出せよと書いてあった。



主人公は部屋を調べながら、脱出の手がかりを探していくことになる。



分からなくなった際には目標を確認すると、少しだけヒントが書いてあるので、たまに見てみると楽になるかもしれない。



全ての謎を解いて、鍵を手に入れたら、扉を開き脱出ができるようになる。

鍵の付いた扉の開閉

鍵のオブジェクトのActive状態を確認し、Falseの場合は開く。Trueの場合は動作しない様にしました。

アイテムの取得等

```
Unity メッセージ10 個の参照
private void Update()
{
    // オブジェクトのActive状態を確認
    KeyActiveSituation = Key.activeSelf;
}

1 個の参照
public void OpenDoor()
{
    // 鍵のActive状態がFalseの場合処理
    if (!KeyActiveSituation)
    {
        // 扉が閉じている時に処理
        if (!opened)
        {
            animator.SetBool("Open", true);
            Light.SetActive(true);
            Open();
        }
        // 扉が開いている時に処理
        else
        {
            animator.SetBool("Open", false);
            Open();
        }
    }
}

2 個の参照
private void Open()
{
    opened = !opened;
}
```

```
#region RayCast設定
1 個の参照
public void RayCast()
{
    // Rayをカメラの位置から飛ばす
    var rayStartPosition = fpsCamera.transform.position;
    // Rayをカメラの向いている方向に飛ばす
    var rayDirection = fpsCamera.transform.forward.normalized;

    // RayがItemレイヤーを持つObjectにあたった場合処理
    if (Physics.Raycast(rayStartPosition, rayDirection, out raycastHit, distance))
    {
        if (raycastHit.collider.gameObject == What[0])...
        else if (raycastHit.collider.gameObject == What[1] && textLabel.text == string.Empty)...
        else if (raycastHit.collider.gameObject == What[2])...
        else if (raycastHit.collider.gameObject == What[3])...
        else if (raycastHit.collider.gameObject == What[4])...
    }
}
#endregion
```

アイテムの取得や、オブジェクトに向かってクリックした際に何かを表示したい場合等の処理に、RayCastを用いました。また、オブジェクトを配列に入れて、if文によってRayCastにヒットしたオブジェクトを区別するようになりました。

金庫のギミック

謎解きの一つとして金庫のギミックを実装しました。入力した数値に合わせて、対応した画像を表示させます。5桁目を押そうとすると桁がResetされます。

CLRボタンを押した際には、pwResetを呼び出して、表示した画像と、入力した内容をすべてリセットするようにしました。

ENTボタンを押した際には、入力したパスワードがあっていたら、金庫を開くAnimationを起動、間違っていたらパスワードをリセットするようにしました。

```
// ボタンを押した際に処理
0 個の参照
public void ClickPW(int num)
{
    if (num <= 9)
    {
        if (pwColumn < 4)
        {
            InputColumn[pwColumn].sprite = sprites[num];
            pwArray[pwColumn] = num;
            pwColumn++;
        }
        else if (pwColumn == 4)
        {
            pwReset();
            pwColumn = 0;
        }
    }
}
```

```
// CLR(リセット)ボタンを押した際に状態をリセット
else if (num == 999)
{
    pwReset();
    pwColumn = 0;
}
// ENT(決定)ボタンを押した際にパスワードの正否判定
else if (num == 111)
{
    for (int i = 0; i < pwArray.Length; i++)
    {
        // int型のpwArrayをStringに変換してconnectPWに入れる。
        connectPW += pwArray[i].ToString();
    }
    // connectPWをint型に変換してcheckPWに入れる。
    int.TryParse(connectPW, out checkPW);

    if (PASSWORD == checkPW)
    {
        kinko.KinkoAnim();
        clear = true;
        CloseUI();
    }
    else
    {
        pwReset();
        pwColumn = 0;
    }
}

connectPW = null;
checkPW = 1111111;
```


ライトの点滅



テスト段階の際にホラーゲームということなので、雰囲気を出すためにLightの点滅等もあったほうがいいかと思い作成したLightを点滅させるプログラムもありましたが、部屋が狭く、使用するLightも二つで、点滅させると部屋が暗くなってしまったり、目に悪かったりなどの理由で使用しませんでした。

PerlinNoise
を用いて、
Lightの点滅
を行いました。
ランダム性が
得られるため、
良さそうかな
と思い使用し
ました。

```
@Unity メッセージ10 個の参照
void Start()
{
    blinkLight = this.gameObject.GetComponent<Light>();
}

// Update is called once per frame
@Unity メッセージ10 個の参照
void Update()
{
    Blinking();
}

1 個の参照
public void Blinking()
{
    if (blinkLight.intensity > maxIntensity / darkness)
    {
        // ライトの点滅を計算
        blinkLight.intensity = Mathf.PerlinNoise(Time.time * blinkspeed, 0) * maxIntensity;
    }
    else
    {
        // ライトの明るさが一定より下回った場合激しく点滅させる
        blinkLight.intensity = Random.Range(0, maxIntensity / 2);
    }
}
```

イライラ棒



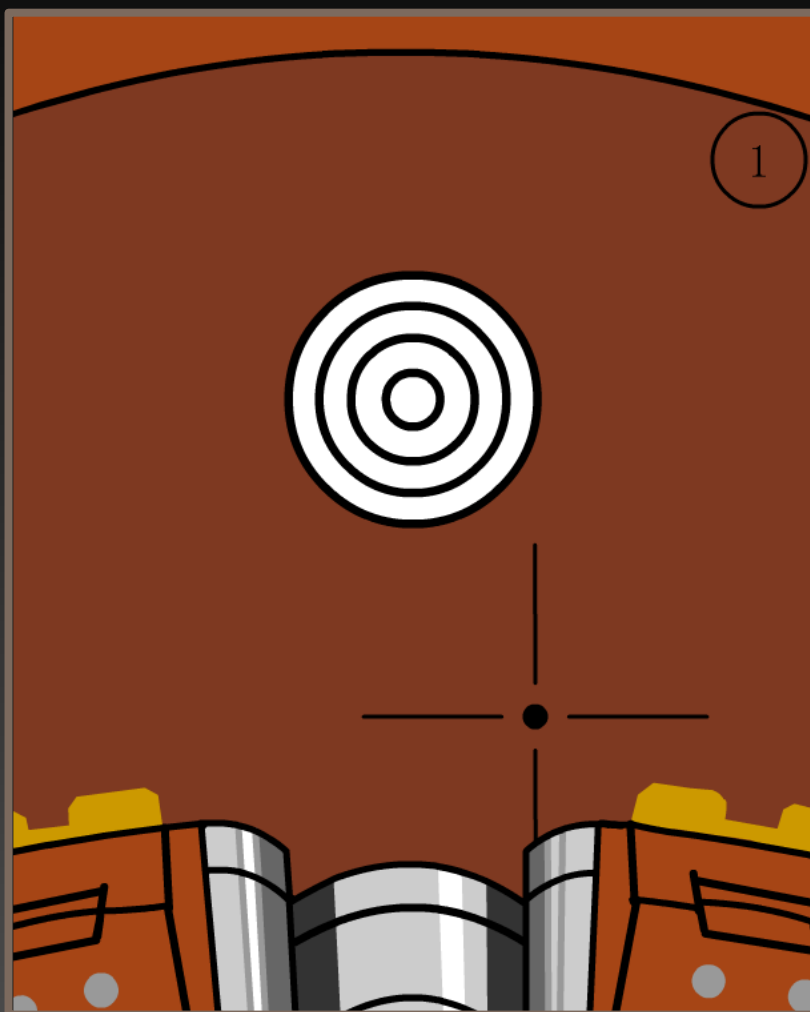
使用言語 : **ActionScript**
使用ツール : **Adobe Animate**
制作人数 : 1人
制作期間 : 約2日

説明

黒いボールをマウスでドラッグし、壁に当たらないように気を付けながら、ボールをGoalまでもっていくとクリアです。

Animate制作作品②

的当て



使用言語 : **ActionScript**
使用ツール : **Adobe Animate**
制作人数 : 1人
制作期間 : 約2日

説明

マウスで十字のクロスヘアを操作し、ランダムな座標に出現する的を5回打ったらクリアになります。