

# 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1年プログラマー専攻

内田 開

## 自己紹介

氏名 : 内田 開  
年齢 : 19才  
希望職種 : プログラマー

## 対応言語

C/C++/C#/  
Action Script/  
HTML/JavaScript  
/PHP

## 開発環境

Unity /Git(GitHub,TortoiseGit)/  
Microsoft Visual Studio2022/  
Adobe Animate/Adobe PhotoShop/  
Adobe Illustrator/



## 自己PR

私はチーム制作で自身の担当箇所を進めながら、他の担当箇所をサポートし、資料探して必要な機能を実装しました。

# 目次

- Unity作品① 4p  
2.5Dシューティングゲーム
- Unity作品② 10p  
2Dアクションゲーム
- Animate作品 13p

# Unity 作品



作品名 : Sorcier

使用言語 : C#

動作環境 : コントローラー/キーボード、マウス

制作人数 : プランナー1人・CGデザイナー2人・  
プログラマー3人

担当 : プログラマー

開発期間 : 3ヶ月

使用ツール: Microsoft Visual Studio 2022

Unity ver.2022.3.10f1

GitHub

# ゲーム概要

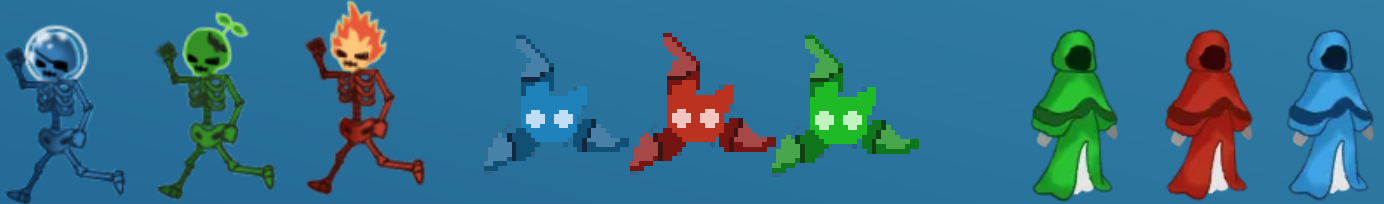


主人公が属性を切り替えながら、敵を倒して戦うシューティングゲーム

主人公と敵の属性は3種類、上手く敵の弱点属性で攻撃を当てる事で効率良く敵が倒せる。

画像は木属性の弾を撃っている。

# 担当箇所①



自分の主な担当  
箇所は敵キャラ  
全てです。

敵キャラの属性  
ごとの処理

主人公が切り替  
えた属性により  
敵キャラの行動  
も変化する。→

```
void Update()
{
    // 属性ごとの分岐処理
    switch (currentState)
    {
        case TypeState.TypeFire:
            UpdateForTypeFireState();
            break;
        case TypeState.TypeWater:
            UpdateForTypeWaterState();
            break;
        case TypeState.TypeLeaf:
            UpdateForTypeLeafState();
            break;
    }

    // 状態ごとの分岐処理
    switch (moveState)
    {
        case MoveState.Chase:
            UpdateForChase();
            break;
        case MoveState.Stay:
            UpdateForStay();
            break;
        case MoveState.GetAway:
            UpdateForGetAway();
            break;
    }
}
```

## 担当箇所②

Unity メッセージ 10 個の参照

```
private void Start()
{
    // エネミーのプールを作成
    enemyPool = new List<GameObject>();

    // 指定した数のエネミーをプールに追加
    for (int i = 0; i < poolSize; i++)
    {
        GameObject obj = Instantiate(enemy, transform.position, transform.rotation, transform);
        obj.SetActive(false);
        enemyPool.Add(obj);
    }

    // エネミーを生成する。
    1 個の参照
    private void Generate()
    {
        // ランダムなエネミーを呼び出す。
        int randomEnemy = Random.Range(0, enemyPool.Length);
        // ランダムな位置を決定
        int randomPosition = Random.Range(0, generatePosition.Length);
        GameObject obj = enemyPool[randomEnemy].GetObjectFromPool();
        // エネミーをアクティブにする。
        obj.SetActive(true);
        // オブジェクトの位置を設定
        obj.transform.position = target.transform.position + generatePosition[randomPosition];
        // 再度生成間隔をカウントする。
        lastGenerateTime = Time.time;
    }
}
```

最初に非表示にした敵キャラを生成し、Destroyせずに同じオブジェクトを使い回す事で負荷を減らせるようにしています。

## 担当箇所③

主人公の属性  
に対応した処  
理

この敵キャラ  
が木属性の場  
合呼び出され  
ます。  
ここで敵キャ  
ラのStateを変  
更します。

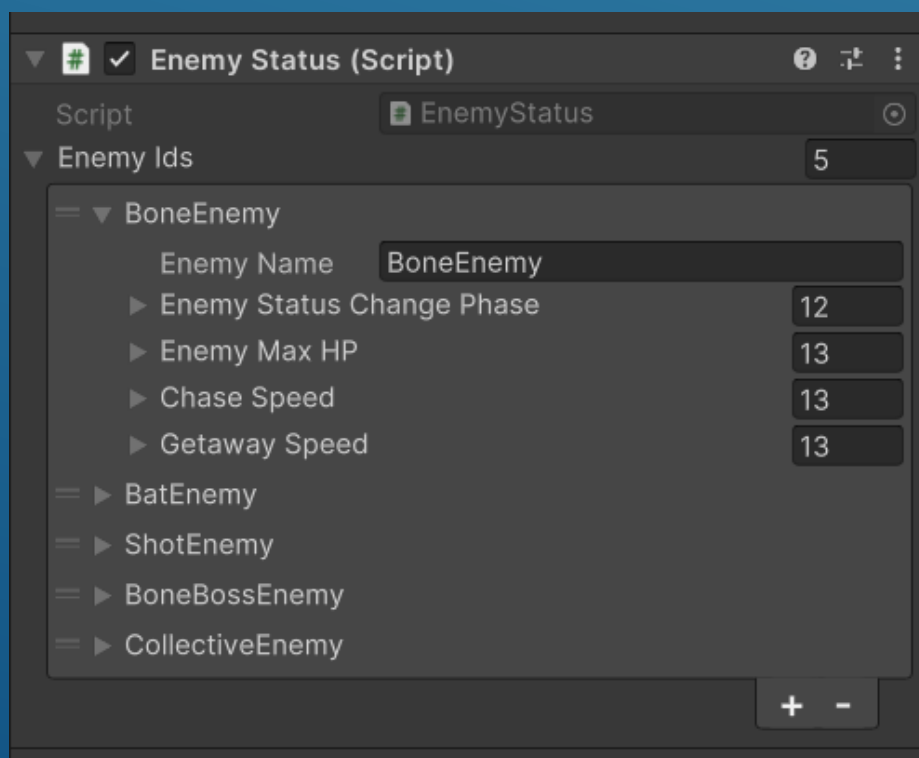
```
// 草属性の処理
1 個の参照
private void UpdateForTypeLeafState()
{
    if (target = GameObject.FindWithTag("Leaf"))
    {
        // 同属性に対する処理
        typeCompatibility = 0;
        moveState = MoveState.Stay;
    }
    else if (target = GameObject.FindWithTag("Water"))
    {
        // 有利属性に対する処理
        typeCompatibility = 1;
        if (!isShotReady)
        {
            moveState = MoveState.Chese;
        }
        else
        {
            // Stopアニメーションの再生
            animator.SetBool(isStopId, true);
        }
    }
    else if (target = GameObject.FindWithTag("Fire"))
    {
        // 苦手属性に対する処理
        typeCompatibility = 2;
        moveState = MoveState.GetAway;
    }
}
```



## 担当箇所④

敵キャラは時間  
が経つごと  
に強化されて  
いきます。

敵の種類ごと  
にIDを設定し、  
Unity の  
Inspector上で  
入力した値を  
参照してHPや  
移動速度をID  
をもとに変化  
させています。



# Unity 作品



作品名：カタナキャッチベンゲイ

使用言語：C#

動作環境：コントローラー/キーボード、マウス

制作人数：プランナー1人・CGデザイナー2人・  
プログラマー3人

担当：プログラマー

開発期間：1週間

使用ツール:Microsoft Visual Studio 2022

Unity ver.2022.3.7f1

GitHub

# ゲーム概要



主人公のベンケイが落ちてくる武器を拾い、スコアを稼ぐゲーム  
刃のある方向から落ちてくる武器を拾ってしまうとダメージを受けてしまうので上手く取っ手側から取ろう。

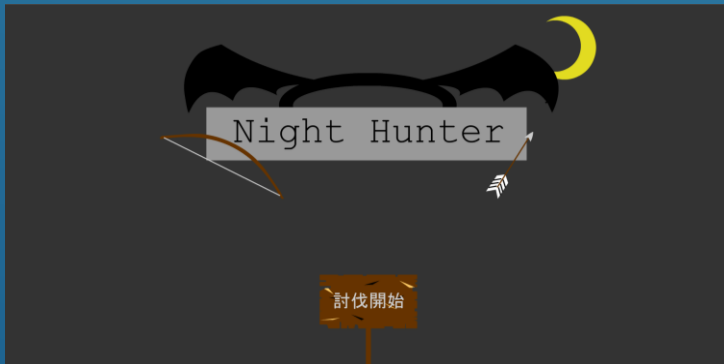
# 担当箇所①

Unity メッセージ 10 個の参照

```
private void FixedUpdate()
{
    // レイを飛ばし、当たった時追跡状態
    if (Physics2D.Raycast(transform.position, raySize, rayLength, playerLayer))
    {
        enemyMove.isChase = true;
    }
    // レイを飛ばし、当たった時攻撃状態
    if (Physics2D.Raycast(transform.position, attackRaySize, attackLength, playerLayer))
    {
        enemyMove.isAttack = true;
        if (lastAttackTime + attackTime < Time.time)
        {
            enemyMove.Invoke(nameof(enemyMove.OnAttack), attackTime);
            lastAttackTime = Time.time;
        }
    }
}
```

レイを飛ばしてプレイヤーに当たった時に状態を変更する。  
追跡状態で速度を変更。  
攻撃状態で攻撃エフェクトを生成する。

# Animate 作品



作品名 : Night Hunter  
使用言語 : ActionScript  
動作環境 : キーボード、マウス  
制作人数 : 自分のみ  
開発期間 : 3日  
使用ツール: Adobe Animate