

北海道情報専門学校ゲームクリエイタ科

水上 新太

希望職種 ゲームプログラマ

無双アクションゲーム 開発期間：2 週間(実働 10 日間)

開発人数：1 人

開発環境：VisualStudio2019

○個人製作の目標

- ・ 200 体の敵が動いていても FPS を 100 以上に保つ
- ・ 就活作品としてプログラムだけでも理解しやすいコードを書く

○アピールポイント

- ・ オブジェクトプール
- ・ バッチング処理
- ・ オクルージョンカリング
- ・ 就活作品としてプログラムだけでも理解しやすいコードを書く

・オブジェクトプール

- ・ Instantiate と Destroy は処理が重いので、頻繁に行うのは向いていない

Instantiate を最大数までは繰り返すがその後は Instantiate と Destroy を行わない為、重くならないようにした。

```
//敵の数が_enemyMaxCountより少ない場合敵を出す
if (_enemyCount < _enemyMaxCount)
{
    //敵のレイヤーにだけ当たるように指定
    int layerMask = 1 << 8;
    //リスポーン場所全てにRayを飛ばす
    for (int i = 0; i < _respawn.Length; i++)
    {
        //Rayに当たったオブジェクト
        RaycastHit hit;
        //リスポーン場所のpositionを取得
        Vector3 respawnPosition = _respawn[i].transform.position;
        //リスポーン場所からRayを飛ばすと低いので少し上げる
        respawnPosition.y += 3.0f;
        //リスポーン場所からRayを飛ばし敵にぶつかった
        if (Physics.Raycast(respawnPosition, _respawn[i].transform.up * -1, out hit, 10.0f, layerMask))
        {
            //ぶつかった場合敵が被るのでそのリスポーン場所はリスポーン出来ないなのでfalseにする
            _isEnemyRespawn[i] = false;
        }
        else
        {
            //ぶつからない場合はリスポーン出来るのでtrue
            _isEnemyRespawn[i] = true;
        }
    }
    //0~リスポーン場所の数の数値をランダムで生成
    int rnd = Random.Range(0, _respawn.Length);
    //ランダムで生成したリスポーン場所がtrueだった場合は敵を生成
    if (_isEnemyRespawn[rnd] == true)
    {
        InstEnemy(_respawn[rnd].transform.position, transform.rotation);
        _enemyCount++;
    }
}

//敵を生成するかtrueにするか
void InstEnemy(Vector3 pos, Quaternion rotation)
{
    //アクティブでないオブジェクトを子の中から探索
    foreach (Transform t in _poolTransform)
    {
        if (!t.gameObject.activeSelf)
        {
            //非アクティブなオブジェクトの位置と回転を設定
            t.SetPositionAndRotation(pos, rotation);
            //アクティブにする
            t.gameObject.SetActive(true);
            return;
        }
    }
    //非アクティブなオブジェクトがない場合新規生成
    //子オブジェクトとして生成する
    Instantiate(_enemyPrefab, pos, rotation, _poolTransform);
}
```

- ・ バッチング処理

- ・ 同じマテリアルの床や壁をバッチング処理し、無駄なドローコールを減らした

- ・ オクルージョンカリング

- ・ 敵にバッチング処理を使用できなかった為、視界外のオブジェクトを描画しないようにすることで、バッチング処理と合わせ描画面での軽量化をした

○プレイ動画

<https://youtu.be/ObHX8qGKm7E>