

Unity を用いた 3D 迷路ゲームの制作

石川 晴己

1. まえがき

Unity というゲームエンジンを使って 3D の迷路ゲームを制作した。

2. 原 理

このゲームは、プレイヤーが車を操作しゴール地点まで行くというものである。また、途中に落ちてあるアイテムを拾うと、図 1 の右下で表示されたキーで使うことも出来る。

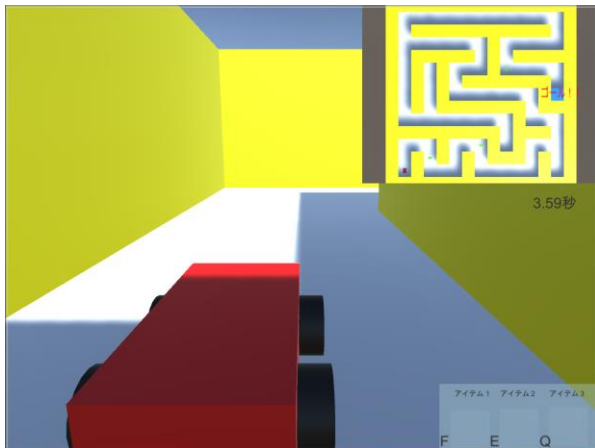


図 1 ゲーム画面

3. 研究内容

(1)迷路の作成

迷路を作成するアルゴリズムの中の一つである、穴掘り法というアルゴリズムを用いて迷路を自動生成するようにした。

```
do
{
    switch (array[k])// 4方向をランダムに取得して道を開く
    {
        case 1:
            if (fy + 2 < vertical && maze[fy + 2, fx] == 1)
            {
                goru++;
                maze[fy + 1, fx] = 0;
                maze[fy + 2, fx] = 0;
                Mining(fy + 2, fx);
            }
            k++;
            break;
        case 2:
            if (fx + 2 < horizontal && maze[fy, fx + 2] == 1)
            {
                goru++;
                maze[fy, fx + 2] = 0;
                maze[fy, fx + 1] = 0;
                Mining(fy, fx + 2);
            }
            k++;
            break;
    }
}
```

図 2 穴掘り法アルゴリズム

(2)プレイヤーの制御

まず、簡単な車の 3D モデルを作成し、次にタイヤを制御出来る WheelCollider というものを、タイヤのオブジェクトにアタッチした。公式リファレンスのスクリプトを参考にし、車を動かせるようにした。

また、このとき、既存の 3D モデルを使用しようとすると、上手く制御できないために、それは使用しなかった。

(3)メインゲームの制御

タイマーを作成し、スタートからゴールまでの、タイムとランクを表示させるようにした。その際左クリックでメニューに戻るるようにした。

また、サブカメラを取り入れることで右上に迷路の地図とプレイヤーの位置を表示させた。

(4)ランキングの実装

SQL を使い、データベースへのアクセスが出来るようにし、ランクやタイム、最短距離をゴールした際に保存するようにした。

メニューのランキングボタンを押すとこれまでの結果を一覧で表示し、並び替えが行えるようにした。

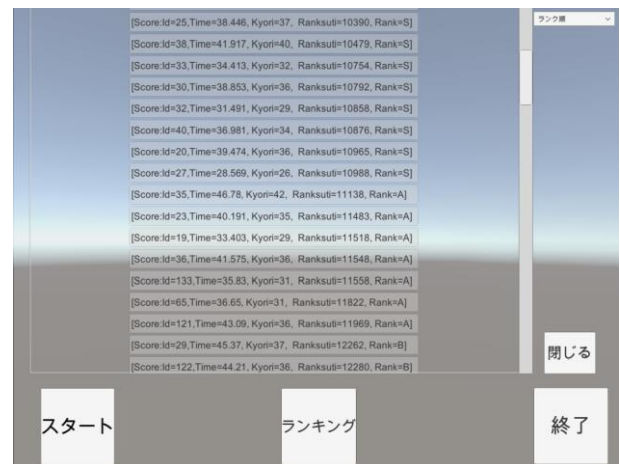


図 3 ランキング UI

(5)アイテムの作成

Item クラスを作成し Unity 上でアイテムが簡単に作成できるようにした。

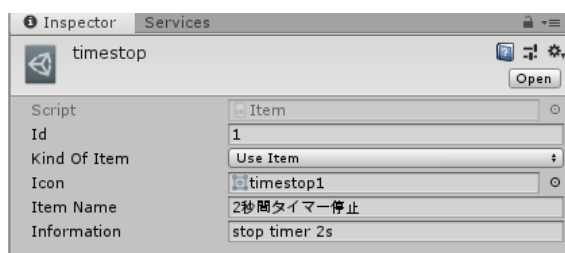


図4 アイテムデータの入力

```
[CreateAssetMenu(fileName = "Item", menuName = "CreateItem")]
public class Item : ScriptableObject
{
    public enum KindOfItem
    {
        UseItem,
        StackItem
    }
    //アイテムID
    [SerializeField]
    private int id;
    // アイテムの種類
    [SerializeField]
    private KindOfItem kindOfItem;
    // アイテムのアイコン
    [SerializeField]
    private Sprite icon;
    // アイテムの名前
    [SerializeField]
    private string itemName;
    // アイテムの情報
    [SerializeField]
    private string information;
}
```

図5 Item クラスのスク립ト

迷路上にランダムに配置されてあるアイテムIDが名前のオブジェクト(Cube)に触れた時に、アイテムを取得できるようにするために、車に当たり判定の処理を追加した。

```
void OnTriggerEnter(Collider t)
{
    if (t.name == "gouru(Clone)")//ゴールした時の処理
    {
        Meiro = GameObject.Find("TimeAttack");
        TimeAttack = Meiro.GetComponent<TimeAttack>();
        TimeAttack.result();
    }
    else
    {
        //アイテムを取得する処理
        Item = GameObject.Find("ItemManager");
        ItemManager = Item.GetComponent<ItemManager>();
        ItemManager.GetItems(t.name);

        Destroy(t.gameObject);//設置されたアイテムの削除
    }
}
```

図6 車の当たり判定のスク립ト

また、指定されたキー(図1の右下)を押すとそのアイテムの効果を発動出来るようにした。

4. まとめ

私は、Unityでのゲーム開発及びアプリケーション開発を学校外でも行っており、ある程度の知

識はあったので作業の進行はとてもスムーズに行えた。しかし、まだまだ未知のことが多くて、新しいことをしようとするとなんにも勉強しないと使いこなせなかった。

また、時間に余裕があればAIや対人も考えていたが、思ったよりもバグの対処に手間を取ってしまったために断念してしまったので、いつか制作してみたいと思った。

5. あとがき

今回の課題研究では1人でのゲーム開発ということで自由に制作出来ていたが、今後は複数人での制作も行わなければならない。そのためプログラムの可読性を上げたり、他のプログラムとの結合を出来るだけ少なく済むようにしようと思った。

また、自分にはデータベースや物理演算など、様々なことに対する知識が不足していたため、基礎知識ももう少し勉強していきたいと思った。

6. 参考文献

・Wheel Collider チュートリアル – Unity マニュアル

<https://docs.unity3d.com/ja/current/Manual/WheelColliderTutorial.html>

・UnityEngine.WheelCollider – Unity スクリプトリファレンス

<https://docs.unity3d.com/ja/2017.4/ScriptReference/WheelCollider.html>

・UnityでScriptableObjectを使ってアイテムデータベースを作成する—Unityを使った3Dゲームの作り方(かめくり)

<https://gametukurikata.com/program/scriptableobjectitemdatabase>

・穴掘り法で迷路を作成する | 極めて冴えない普通人の気まぐれブログ

<http://kiwasalog.com/caprice/algorithm-make-maze.html>