

Unity での 2D シューティングゲームの制作

井上 光輝

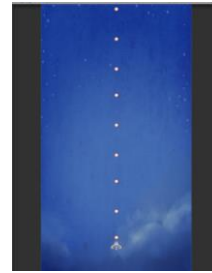


図 2 弾の発射

1. まえがき

2年生の時に Visual Studio で将棋のゲームを作ったので、今回は Unity というソフトを使ったシューティングゲームの制作に挑戦した。

2. 原理

Unity では 2D や 3D のゲームを制作することができ、私は C# というプログラミング言語を使いシューティングゲームの作成をした。

3. 研究内容

(1)ゲーム制作の準備

ゲーム画面のサイズや、自機や敵の画像を Unity 内に取り込み、背景の画像を貼った。

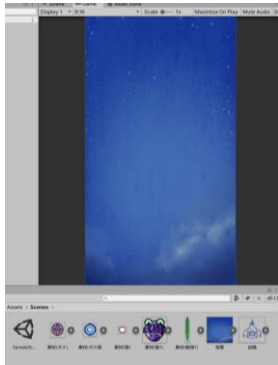


図 1 ゲーム画面

(2)自機の動作

①自機を動かす

キーボードの十字キーで自機を操作できるようにし、左の Shift キーを押すと自機の動くスピードを遅くするようにした。

②弾の発射

Z キーを押すと弾を発射するようにした。

(3)敵の出現と動作

一定の間隔で敵を出現させ、弾を自動的に発射するようにし、自機の発射した弾に当たると消えるようにした。

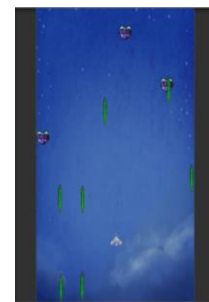


図 3 敵出現

```
public Enemy[] m_enemyPrefabs; // 敵のプレハブを管理する配列
public float m_interval; // 出現間隔 (秒)

private float m_timer; // 出現タイミングを管理するタイマー

// 毎フレーム呼び出される関数
private void Update()
{
    // 出現タイミングを管理するタイマーを更新する
    m_timer += Time.deltaTime;

    // まだ敵が出現するタイミングではない場合、
    // このフレームの処理はここで終える
    if (m_timer < m_interval) return;

    // 出現タイミングを管理するタイマーをリセットする
    m_timer = 0;

    // 出現する敵をランダムに決定する
    var enemyIndex = Random.Range(0, m_enemyPrefabs.Length);

    // 出現する敵のプレハブを配列から取得する
    var enemyPrefab = m_enemyPrefabs[enemyIndex];

    // 敵のゲームオブジェクトを生成する
    var enemy = Instantiate(enemyPrefab);

    // 敵を画面外のどの位置に出現させるかランダムに決定する
    var respawnType = (RESPAWN_TYPE)Random.Range(
        0, (int)RESPAWN_TYPE.SIZEOF);

    // 敵を初期化する
    enemy.Init(respawnType);
}
```

図 4 敵を出現させるプログラム

(4)スコアとゲームオーバーの表示

①スコアの表示

スコアを表示するようにし、敵を倒すとスコアの数値を加算するようにした。

②ゲームオーバー画面

敵の弾に当たると自機を画面から消し、「GameOver」の文字を表示するようにした。

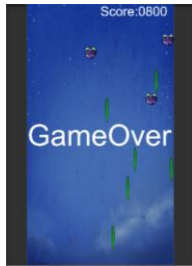


図5 スコア画面とゲームオーバー画面

```
public class Score : MonoBehaviour
{
    int score = 0;
    GameObject scoreEmpty;

    public void AddScore()
    {
        this.score += 100; // scoreを100加算
    }

    void Start()
    {
        this.scoreEmpty = GameObject.Find("Score");
    }

    void Update()
    {
        scoreEmpty.GetComponent<TextMesh>().text =
            "Score:" + score.ToString("D4"); // scoreを表示
    }
}
```

図6 スコアを表示するプログラム

```
public class GameOver : MonoBehaviour
{
    GameObject gameOverEmpty;

    public void AddGameOver()
    {
        gameOverEmpty.GetComponent<TextMesh>
            ().text = "GameOver"; // GameOverの文字を格納
    }

    void Start()
    {
        this.gameOverEmpty = GameObject.Find("GameOver");
    }
}
```

図7 ゲームオーバーを表示するプログラム

(5)ボス敵の追加

スコアが一定数加算されるとボス敵を出現するようにした。ボスの発射する弾は自機を狙うようにした。



図8 ボスの出現

4. まとめ

(1)成果

今回、初めて Unity を使ったゲーム制作をしたということもあり、最初はわからないことが多く戸惑うこともあったが、作業を進めるうちに慣れ、自分の考えていたものに近づけることができた。

(2)反省点

Unity に慣れるまで時間がかかってしまい、最低限ゲームとして成立させるのに精いっぱいだったため、敵の種類を増やす、難易度を変更できるようにするといった改造ができなかった。

5. あとがき

Unity での 2D シューティングゲームの制作は、思い通りに作業を進めることは難しかったが、自分が考えていた通りに動くようになるととても嬉しかった。今回の課題研究でもとても良い経験を得ることができたので、またゲームを制作する機会があったら活かしていきたい。

6. 参考文献

- ・【Unity 入門】全方位シューティングを作る

<https://baba>

[s.hatenablog.com/entry/2018/04/01/152100](https://baba.s.hatenablog.com/entry/2018/04/01/152100)

- ・【Unity 入門】60分で作るシューティングゲーム

<https://nn->

[hokuson.hatenablog.com/entry/2016/07/04/213231](https://nn-hokuson.hatenablog.com/entry/2016/07/04/213231)