

VisualBasic 2D シューティングゲームの制作

1. 目的・動機

私が遊んでいるゲームに登場するミニゲームを再現してみたいと思った。

2. 使用機器、ソフトウェア

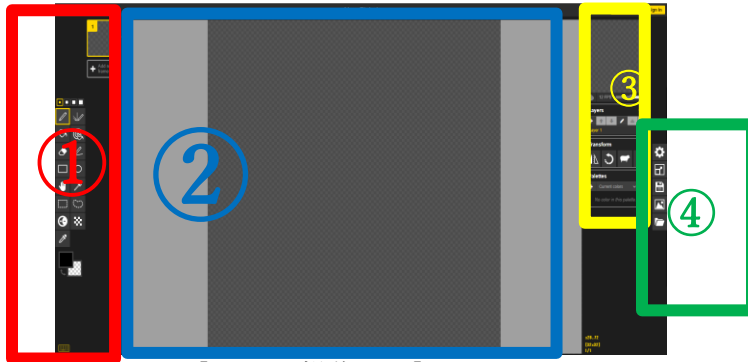
- ① piskel
- ② VisualStudio 2017

3. 作業内容

< 1 学期 >

Piskel の操作に慣れる。

自キャラ、敵キャラ、自分の撃つ弾、敵の撃つ弾を Piskel で描く。



[Piskel 操作画面]

- ① 様々な機能のペンがあり、描きたいものによって変えることができる。
- ② 絵を描く場所。ここでクリックやドラッグすることで絵を描くことができる。
- ③ レイヤーに関する操作ができる。
- ④ 描いたものの保存やマス目の数を変えることで画像の粗さを変えることができる。

< 2 学期 >

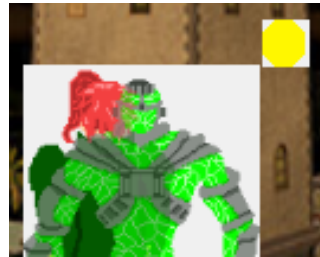
弾の移動と当たり判定などのプログラミングをする。

リザルト画面を作る。

自キャラが左右に動けるようにする。



[リザルト画面]



[当たり判定の画面]

< 3 学期 >

ボスを Piskel で描く。

ボス戦を作る。

発表準備



[ボスの絵]

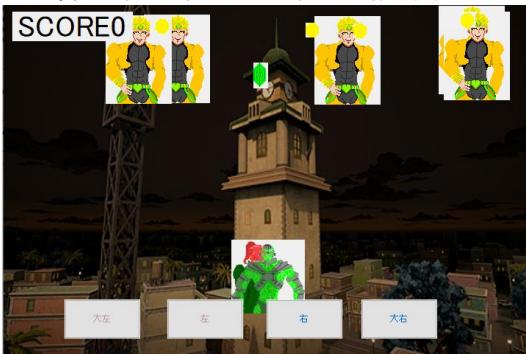
4.

作品内容

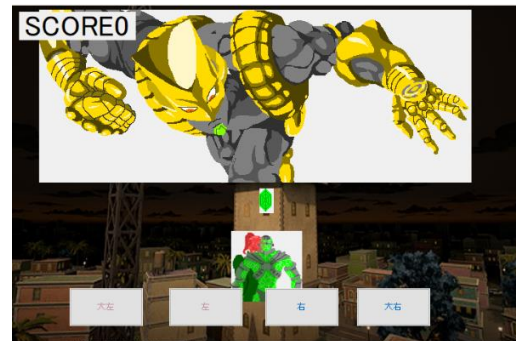
今回作ったのは2Dのシューティングゲームで、自機は動かず上から敵が出てくるようになっている。Unityでプログラムを完成させることができるか不安だったので、実習で使ったことのあるVBでプログラミングをした。その代わりにゲームのキャラはすべて自分で描くようにし、今回は僕の好きなジョジョの奇妙な冒険という漫画(アニメ)に登場するキャラを描いた。

敵キャラを10回倒すと演出が入りボスが出てくる。ボスは10回攻撃を当てると倒せる。

自分の撃つ弾は敵の弾より結構早くなっているが自機は敵の攻撃が1度当たれば負ける。



[プレイ中の画面]



[ボス戦の画面]

5. 感想

Piskelの操作に慣れるのに時間がかかり、キャラ制作がなかなか終わらず少し焦ったが間に合ったので良かった。

最初は画像を透過させた状態で戦う予定だったが、透過させた状態での当たり判定が予想以上に難しく、それまでに作ったプログラムのほとんどを変更し、透過しない状態で当たり判定をしっかりとさせた。透過させた状態で頑張ろうと思って予定を組んでいたため、時間的に苦しくなるかと思ったが、2学期にはボス戦以外を完成させることができた。

ボス戦を作るとき登場するキャラはすべて自分で描くと決めていたので、久しぶりにPiskelを触ったがほとんどの操作を覚えていたのでスムーズに進めることができた。

6. 参考文献

[1] ドット絵が描けるアプリ Piskel の全ツール説明書

(URL: <https://shamoei.com/illustration-design/piskel-tool/>)

[2] メテオゲーム「隕石をよける」を作ろう

(URL: <https://www.greenowl5.com/gprogram/vb1/visualbasic1310.html>)