

研究テーマ シューティングゲームの製作(縦スクロール型)

動機

情報技術科で学んできたことを活かし、作ってからも楽しめるものを作りたいと思います。ゲーム制作を選びました。
ゲーム制作でC言語を選択した動機は、情報技術科で最も使用した言語だったからです。

概要

ジャンル

シューティング

操作

十字キー：メニュー選択、自分の移動。
Shift キー：鈍足移動。ゆっくり動きます。
Z キー：メニュー決定、弾の発射。

仕様

処理落ち：かなり多めです、敵が出てこなければ再起動してください。
当たり判定：当り判定がかなり小さめに設定されています。
弾が多い：敵の弾がかなり多めに設定されています。
弾エフェクト：弾が多いため弾出力時の音が長時間続き
マシンガンのように聞こえます。
多彩な弹幕：弾が漂うのはバグではなくて仕様です（製作者：松森）。
敵を倒しても：弾は消えません。

使用画像

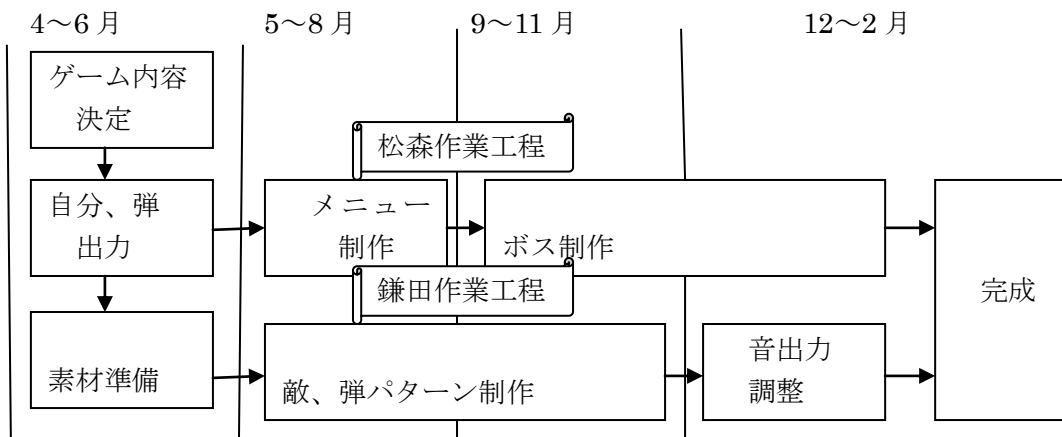


自分画像



敵画像

工程



ゲームの内容

起動するとメニューが開きます。

本来はヘルプなども実装する予定でしたが、未実装です。



左側はメニュー画面。右側はゲーム画面（縮小版）です。

丸い球 1 つ 1 つに当たり判定があり、操作キャラと接触すると消滅。同時に右下の HP の数字が消えていきます。0 になるとゲームオーバーです。敵の HP は表示されません。

まとめ

当たり判定に関して、今までとは違うやりかたで行った。

x、y 座標の正方形ではなく、円を使った当たり判定である。

これは、敵の弾の大きさと操作キャラの大きさをあらかじめ **range** という関数に入力しておき、それぞれの x、y 座標の距離を算出。一定距離以下であれば接触していると判断させる方法である。計算方法は

```
x=(int)(PlayerShot[i][j].x-enemy[s].x);
```

(相手の弾と操作キャラの x 座標の距離を求める)

```
y=(int)(PlayerShot[i][j].y-enemy[s].y);
```

(相手の弾と操作キャラの y 座標の距離を求める)

```
sqrt(x*x+y*y) < range+enemy[s].range
```

平方根 (x²乗+y²乗) =

弾と操作キャラの距離<操作キャラの大きさ+相手の弾の大きさ

操作キャラの当たり判定 (円) に、相手の弾の当たり判定 (円) がどれだけ近づいたかをはんだんすることで、設定した円上の当たり判定に接触しているかどうか判定できる。

上記の式の場合、距離のほうが短いので当たっている判定になる。

感想

構造体を 2 重に使う必要があったため、当たり判定と、弾の出現が一番大変だった。まず、敵の情報 (x 座標、y 座標など) を納める構造体を作り、その中に弾の構造体 (x 座標、y 座標、動作パターンなど) の構造体を入れる必要があり、これを行わないと、敵の情報と、その敵と弾を関連付けることが出来なかった (敵が沢山出現するため)。1 体作って動作確認をするまではさくさくと進んだが、2 体から先となるとエラーが勃発した。思えば、製作時間のほとんどはデバッグに使っていた。

敵の行動パターンを増やす際に、参考のために何作かのシューティングをプレイし参考にしましたが、さまざまな行動パターンがあり、その行動を実際に自分が作ることができきちんと動作した時とてもうれしかったです。

今回の経験を活かし、今後このような機会があれば 1 から自分で考え制作したいと思います。