

自機・敵機の弾の描画と移動

この資料ではゲームにちょっと凝った仕掛けを作っていきたいと思います。お暇な人はどうぞ。

◆ 敵機の弾の発射

今回のゼミで敵が弾を発射するようになりました。しかしこれは目の前に直線的に打つだけです。ということで今回は自機に向かって弾を発射するような仕掛けを作りたいと思います。

今回改変するのは敵の弾を作る関数 `void create_enemybullet` です。

自機に向かって敵機の弾が発射するのにはどうすればいいでしょうか。これには二つの変数の値を計算し代入する必要があります。その値とは `enemybullet[i][j].vx` と `enemybullet[i][j].vy` です。この値は敵が発射した弾の x 軸方向の速度と y 軸方向の速度を入れる変数です。今のプログラムではこの値は初期化変数で初期化したそのままの値、 x 軸方向へは 0 、 y 軸方向へは 15 が入っています。弾の移動で 1 ループ中に弾の x 座標に `vx` の値、 y 座標に `vy` の値を足しています。ためしに `vx` に 15 、`vy` に 15 を代入すると弾は右下 45 度の方向へ飛んでいくはずですが、なので自機に向かって弾を撃つには自機と敵機の位置から三角関数を使い、この `vx` と `vy` の値を計算し代入する必要があります。ややこしいですね。

さて、この計算に使うのが前回の資料でも紹介した `<math.h>`。これを `include` することによりいろんな計算に関する関数を使うことができます。今回使うのは `cos` 関数、`sin` 関数、`atan2` 関数。

`cos` 関数は与えられた数のコサインを `double` 型で返す関数です。

使用例: `cos(a);` → a のコサインが返ってくる

`sin` 関数は与えられた数のサインを `double` 型で返す関数です。

使用例: `sin(b);` → b のサインが返ってくる

`atan2` 関数は与えられた二つの数の逆正接（アークタンジェント）を `double` 型で返す関数です。

使用例: `atan2(c,d);` → d/c のアークタンジェントが返ってくる

まず void create_enemybullet 関数で自機と敵の角度を格納する変数を宣言しましょう。

```
double ang; //自機と敵の角度を格納する変数
```

次にこの ang を計算しましょう。存在していない弾を探し出し、存在させ弾を作り出す条件式に次の文を入れてみてください。穴埋めはプログラミングというよりかは数学です。

```
//ang に自機と敵との角度を入れる  
ang=atan2((double)((【穴埋め】)-(【穴埋め】)),(double)((【穴埋め】)-(【穴埋め】)));  
//計算した値をもとに vx,vy を算出  
enemybullet[i][j].vx = 【穴埋め】 *8;  
enemybullet[i][j].vy = 【穴埋め】 *8;
```

難しいですね。分からなかったら解答は班員に聞いてみてください。