第３回ｂ　敵の表示

ファイルを分割して少し見やすくなったと思います。次は、いよいよ敵を出したいと思います。まぁ、敵を出すといってもやり方は自機を出すのとほぼ同じなのでそこまで難しくはないです。また、今回は game.cpp しかいじりません。

敵のことをまとめる関数を作り敵を表示しよう！

まず関数を作る前にプロトタイプ宣言をしておきましょう。game.cppのプロトタイプ宣言をまとめているところにvoid g_teki(void);と書きましょう。また、敵を表示したりするときにx,y,flagなどを使いたいためheader.hにある構造体（mono）を利用できるように struct mono teki[TEKI_KAZU];をstruct mono jiki ; の上に書いてください。また、#define TEKI_KAZU 8,#define TEKI_SOKUDO 2を#define JIKI_SOKUDO 8の下に書き加えてください。

次にg_teki関数を作っていきたいと思います。

ソースは以下のようになります。

```
void g_teki( void){
        //敵に関する関数
        int i;

        for( i = 0; i < TEKI_KAZU; i++){
                if( teki[i].flag == 1){
                        //teki[i].flag = 1で敵がいる

                        if( time % 100 < 50){
                                //0 ～50フレーム目だと右
                                teki[i].sokudo_x = TEKI_SOKUDO;
                        }
                        else{
                                //50 ～100フレーム目だと左
                                teki[i].sokudo_x = -TEKI_SOKUDO;
                        }
                        teki[i].x += teki[i].sokudo_x;
                        //速度を座標に代入する

                        DrawGraph( (teki[i].x - teki[i].haba / 2), (teki[i].y -
teki[i].takasa / 2), teki[i].handle, true);
                        //敵を描画する
                }
        }
}
```

最初のfor文でteki[0]からteki[7]までを制御しています。

次に初期化をしていきたいと思います。

初期化するのはteki[0]からteki[7]までのflag,x,y,lrです。また、敵を２体出したいのでteki[1]とteki[2]も初期化します。

```
for( i = 0; i < TEKI_KAZU; i++){
                    teki[i].flag = 0;
                    teki[i].lr = 1;
                    teki[i].x = 0;
                    teki[i].y = 0;
          }
          teki[1].flag = 1;
          teki[1].x = 500;
          teki[1].y = 200;
          teki[1].haba = 32;
          teki[1].takasa = 32;
          teki[1].handle = LoadGraph( "data/teki.bmp");

          teki[2].flag = 1;
          teki[2].x = 600;
          teki[2].y = 100;
          teki[2].haba = 32;
          teki[2].takasa = 32;
          teki[2].handle = LoadGraph( "data/teki.bmp");
```

最後にゲームのループ内にg_teki();を書いて読み込んでコンパイルすれば今回は完成です。