第３回　画像を表示、動かして制御しよう

第１回のゼミでゲームの土台、画面を作ることはできたはずです。

ゲームでは画面内にキャラクターや物がでてきて表示され動いたり消えたりします。

今回はその画像を画面に表示させ動かして制御させます。

前回の復習

ウインドウでは左上を原点として右下方向に進むほど座標が大きくなります。

今回のウインドウのサイズは(640, 480)なので座標は

(0,0)　　(640,0)

(0,480)　　(640,480)

となります。

ウインドウ上に画像を表示させるには

LoadGraph(“画像のファイル名”)

でメモリに画像を読み込み

DrawGraph(座標 x,座標 y,読み込んだハンドル,TRUEorFALSE)

で表示することができます

例

menu = LoadGraph(“menu.bmp”);　　//LoadGraph で読み込み DrawGraph(0, 0, menu, FALSE);　　//DeawGraph で menu.bmp を(0,0)に表示

TRUEorFALSE のところは TRUE なら透過色を有効に

FALSE なら透過色を無効にすることができます。

準備

これから画像を動かしていく中で常に手利きや自機、背景、弾などの座標を記憶していかなければいけません。
なのでここで、関数をまたいでも使える大域変数(グローバル変数)でそれらの座標を記憶する変数を作っておきます。今回は宣言する変数の個数がすくないのであまり恩恵にはあずかれませんが構造体を利用します。
以下のような宣言を大域変数になるようなところでやってください。

```
struct zahyou {
        int x,y;//座標
        int img;//画像の保存
};

struct zahyou haikei;
```

```
struct charcter {
        int x,y;//座標
        int img;//画像の保存
};

struct character ziki;
```

今回必要な分は今回は書いておきますが、以降必要ならこれらの構造体のメンバを変更したり、あたらしい変数を宣言するなり各自考えて追加・変更していってください。

画像を動かす

DrawGraph の中の座標は数字だけでなく変数を入れることも可能なので
変数で数字を動かせば(If 文とか)画像を自動で動かすことができます。

例

```
if(haikei.y>480)
haikei.y = 0;
DrawGraph(haikei.x, haikei.y, haikei.img, FALSE)
haikei.y += 5
```

ここで一つ課題です。

例を参考に void SYOKIKA(void)に変数の初期化
void HYOUZI(void)で背景と敵を動かして表示するプログラムを書いてみよう。

キー操作

ただ画面上で画像が動いているだけではアニメや動画と同じです
ゲームの場合は自分で操作できなければいけないので
自分で画像を操作できるようにしましょう。
ここではキーボードでの操作を行います

キーボードで操作するにはキーボードが押されているか押されてないか
判定を取らなければいけません、それには

```
CheckHitKey(キーコード)
```

を使い判定します

キーコードの一例

```
KEY_INPUT_RETURN　：　エンターキー
KEY_INPUT_ESCAPE　：エスケープキー
KEY_INPUT_SPACR　：　スペースキー
KEY_INPUT_UP　：　上キー
KEY_INPUT_DOWN　：　下キー
KEY_INPUT_LEFT　：　左キー
KEY_INPUT_RIGHT　：　右キー
```

この関数は戻り値として
キーが押されている場合 1 を返し
キーが押されていない場合 0 を返します
例

```
if(CheckHitKry(KEY_INPUT_RETURN) == 1){        //エンターが押されたら
  owari = 1;                                          //終了処理
  break;
}
```

ここで一つ課題です。

例を参考に void SYOKIKA(void)に自機の変数を初期化
void HYOUZI(void)に自機の描画
void SOUSA(void)に自機を上下左右で操作できるプログラムを書いてみよう

これまで作成したファイルだと画像がウインドウの外からでてしまうことがわかります
これをウインドウ内で画像が出ないようにするには
if 文を使って画面から出たときにウインドウに戻すようにします。

言葉で言うことは簡単ですが式にするのはむずかしいはず・・・！

変な表現ですが
もし機体が全体の X 座標が機体の X 座標を引いた値より大きい値にきたら～　とか
考えると書きやすいかもしれません　（わかりにくかったらスルーで）

これらをふまえて以下の二つの課題をやってください。

void ZAHYOU_SYUUSEI(void)という自機が画面の外にでないようする関数を作ろう

背景を自動で動かし無限ループさせよう