ソフトゼミＢ　第５回

全体の流れ

今回まで四回にわたって配布したＳＴＧを解説しました。ＳＴＧ最終回の今回は全体の流れを一歩引いた眼で見渡してみます。

✤今回の作業

前回までで穴４まですべて埋めたと思います。今回はもう穴を埋める作業はないので、前回の最終ページに乗せたノルマ

1. 自機狙い弾の実装
2.ホーミング弾の実装
3.自機狙い弾、ホーミング弾と何も狙わない弾の stage1 関数内でのスイッチ化

を実装してください。

✤プログラムの流れ

今回のプログラムには次の流れがあります。

起動→main.cpp→（ループ突入）→update.cpp→draw.cpp→（ループ終わり）→終了

このとき、必要に応じてヘッダーファイルを読み込んで変数の定義等を読み込んでいます。順に見てみましょう。

起動

そもそも起動できなければゲームも何もありません。これができない時に理由がいくつか考えられます。

1. （プログラムミスによる）コンパイルエラー
2. 適切なファイルが入っていない

3. 適切にファイルを指定できていない

といった理由が考えられます。1.を除けば、起動できない時はプログラムが原因ではない事も多いです。

main.cpp

ここはほとんどいじりません。ここに書いてある記述はＤＸライブラリの初期化や、ウィンドウモード化、ゲームループの設定等、ゲームに必須な基本的な設定ばかりです。いじる必要もほとんどないでしょう。

ただし役割は重要です。ここで設定されたゲームループを一回行うごとに１フレーム進みます。別の言い方をすれば、ここにゲームの全体の流れが書いてあります。update と draw 関数の中に動作が入っているので、それらを参照して実際のゲームは動いています。

update.cpp

数値更新のための関数、update 関数が入っています。この update 関数では数値更新を行います。具体的には、敵座標更新のための関数(move_enemy)や弾の生成(create_bullet 系)、あたり判定(judge 系)等といった関数をここで実行しています。このとき、ゲームがどういう段階にいるのか、例えばタイトルにいるのか、ゲームステージを実行しているのかどうかも管理しています。update 関数がゲームの中身の６割以上を占めていると言っても過言ではないでしょう。

stage1 関数だけやや異質です。stage1 関数はこの中で敵の生成の流れと敵の弾の出し方を記述することを想定しています。なので、stage1 関数の中だけで敵の動きを記述できるように、create_enemy(あるいは独自の関数)で弾の出し方も決められる方が望ましいです。こうした工夫でゲーム制作の自由度、柔軟性を上げると独創的なゲームを簡単に作れます！

update 関数で実行される create,move,judge 系関数については前回までで解説したので今回は割愛します。

draw.cpp

各種画像の出力を担当します。ここはＤＸライブラリの面目躍如です。配布したサンプルコードでは DrawGraph 関数しか使っていませんが、画像のある部分を指定して表示できる

DrawRectGraph、画像を回転させる DrawRotaGraph、二つの機能が合わさった DrawRectRotaGraph といったところは非常に使い勝手の良い関数です。他にも単に長方形を書くものや、画像を拡大、縮小させるもの、円形に描画させるもの等たくさんの関数がＤＸライブラリにあるので、このあたりで困ったら一度ＤＸライブラリのホームページで関数を確認すると良いでしょう。

これは画像だけでなく全般に言えることですが、ゲーム制作の際はもちろん、著作権的に問題のないものを使ってくださいね。画像等は使う際はどこからダウンロードしたか残しておくと後々困りません。

ちなみに最終的に音もつけてもらいますから、今のうちにそのあたりも DX ライブラリでどのように行うか調べると良いでしょう。

✤ 最後に

今回をもってエレクトロニクス研究部ゼミＢＳＴＧ回は終了になります。次回からは山口によるアクションゲームの解説が行われることになります。ほとんどの人は今回までのＳＴＧと、次回からのアクションゲームのどちらかを発展させたゲームを作るかと思います。どのような形式にせよ、最終的にどのようなゲームを作るか、どんな風に面白いゲームなのかを意識してゲームを作ってくれれば良いゲームが作れると思います。

以上でゼミＢＳＴＧ回は終わりです。皆さんお疲れさまでした。