ソフトゼミB　第8回 重力

■　はじめに

前回までは、シューティングゲームについて扱ってきました。本当は、今回から 3 回にかけてアクションゲームを取り扱う予定だったのですが、進度調整と台風直撃の影響で、内容を 1 回に縮小せざるを得ませんでした。アクションゲームを作りたかったみなさん、申し訳ありません。というわけで、アクションゲームで最も重要なエッセンスの「重力」についてのみ今回は解説します。なお、今回のみ DX ライブラリの復習も兼ねて新規にプロジェクトを作ります。

■　準備（第 1 回の復習）

プロジェクトと main.cpp を作成し、DX ライブラリの設定も行なってください。ここまでの手順は別紙にまとめましたので、復習も兼ねてやってみてください。main.cpp の内容は別紙の「初期状態で書いておくべきコード」と同じものを打ってください。コメントは写さなくても OK です。

それと、プロジェクトのあるフォルダに「img」という名前のフォルダを作成し、中に 32*32 の画像「player.png」を作っていれておいてください。

■　いろいろ書き足す

➢　マクロ

今回は、別紙にある「WIDTH」「HEIGHT」以外に 4 つのマクロを定義します。

```
#define GRAVITY        1      // 重力加速度[px/frame^2]
#define V0            -16     //ジャンプ時のプレイヤーの初速度[px/frame](負の値)
#define MAX_SPEED      24     // 最大落下速度[px/frame]
#define PLAYER_SIZE    32     // プレイヤー(正方形)の 1 辺の長さ[px]
```

を、#define HEIGHT～の下の行あたりに入れておいてください。

➢　構造体

プレイヤーの構造体です。

```
struct SPlayer{
  int x, y;          //プレイヤー画像の左上の座標
  int vy;            // 速度 横方向の移動は実装しないので vx は無し
  int isJumping; //ジャンプ中かどうか 変数名の理由は書いている途中でわかるはず
```

```
  int graph;       // グラフィックハンドル
};
```

を、さっきの#define の下あたりに入れておいて下さい。

- 変数の宣言/初期化

今回は WinMain 関数のみで作ってゆきます。そのため、グローバル変数を使う必要がないので、今回使用する変数は全て WinMain 関数内でのみで使える「ローカル変数」です。

```
struct SPlayer player;
char keyState[256];
int pyBottom = HEIGHT - PLAYER_SIZE; //底にいる時の player.y の値
```

を、WinMain 関数の最上部(ChangeWindowMode の上)あたりに書いてください。pyBottom は、プレイヤーが地面(画面下端)に接しているときの player.y の値です。これを利用して、後で登場する「接地判定」を簡単にしています。

続けて、

```
player.x = 100;
player.y = pyBottom;
player.vy = 0;
player.isJumping = 0;
player.graph = LoadGraph( "img/player.png" );
```

を、DxLib_Initより下(SetDrawScreenより上)に書いてください。

- キー入力

まずはキー入力。

```
GetHitKeyStateAll( keyState );
```

を、ゲームループ最初の行(ClearDrawScreen の上)に入れてください。

- プレイヤーの移動

今回のキモです。4 行目の if 文条件式の「!」を抜かさないように。

```
if( player.isJumping ){
  player.vy += GRAVITY;  //速度を重力加速度分だけ増やす
}
if( !player.isJumping && keyState[ KEY_INPUT_Z ] ){ //接地時に Z キーで…
  player.vy = V0;          // 初速度で打ち上げ
  player.isJumping = 1;  //ジャンプ中であることを示すように
}
if( MAX_SPEED < player.vy ){  //最大速度は超えてはいけません！
  player.vy = MAX_SPEED;
```

```
}
player.y += player.vy;
if(pyBottom <= player.y ){   // 接地判定
   player.y = pyBottom;
   player.vy = 0;
   player.isJumping = 0;
}
```

を、さっきの GetHitKeyStateAll の次の行～ClearDrawScreen の間に入れてください。

- 描画

```
DrawGraph( player.x, player.y, player.graph, TRUE);
```

を、ClearDrawScreen の次の行に入れておいてください。

これで完了です。Z キーを押すと作った画像が跳ねますか？

■ 理屈

今回作った重力はすっごく擬似的なものです。実際には、重力加速度・初速度などから座標を求める公式を使う他に、微分方程式を解いて空気抵抗を求める必要があり、すっごく面倒なので、最大の速度を 24[px/frame]とすることで計算を簡単にしています。

まず、DX ライブラリでできた画面(に限らず、画像の座標系は全て)画面の左上の座標が(0,0)で、そこから x 軸正の方向が右, y 軸正の方向が下になっています。

そうしたときに、あるフレームでのプレイヤーの速度は、次のような式で求まります。ゲーム開始から f フレーム目のプレイヤーの速度を v_f [px/frame] 、重力加速度を g[px/frame²]とすると、f+1[frame]目の速度 v_{f+1}[px/frame]は、

$$v_{f+1} = v_f + g \times 1$$

で求めることができます。g×1 は 1[frame]での速度の変化量です。

同じ理由から、f+1 フレーム目の Y 座標 y_{f+1}[px/frame]も、

$$y_{f+1} = y_f + v_{f+1} \times 1$$

で求めることができます。

ただし、接地したり、最大速度に達したりした場合はこの限りではありません。

なお、1 でかける動作は次元を変える役割があるので、数学/物理的にはものすごく大事ですが、プログラミング的には意味をなさないので、ソース中では省いています。

■　参考: デバッグ出力

今のあるパラメータの値を見たかったとします。しかし、いちいち DrawFormatString していたのでは描画する座標が必要ですし、すっごく面倒です。

そこで printfDx 関数の出番です。使い方は stdio.h(または cstdio)の printf と全く同じです。違うところは、出力が画面上の左上に出てくるという点です。

なお、printfDx で描画した文字列は ClearDrawScreen で消えません。消すには専用の関数「clsDx」を呼び出してください。

例えば、今のプレイヤーの加速度を見てみたかったとします。描画処理のどこかの適当な場所で

```
printfDx( “vy: %3d”, player.vy );
```

みたいなのを書いてみましょう。このままだと出力が溢れてしまうので、忘れずに、ClearDrawScreen のすぐ下に

```
clsDx();
```

を入れてください。

printfDx/clsDx は、デバッグ作業において非常に心強い味方となるでしょう。

■　おわりに

いかがでしたか？これで長い長い必修のゼミは最終回です。ゼミ B はいろいろな要素を詰め込んでしまったので、本質的な理解はなかなか難しかったかもしれません。

さて、夏休みにはゼミ C の開講が予定されています。詳しい予定は決まり次第お伝えしますが、どんなことをやるかだけここに書いてみます。

・ゼミ B の復習 ・・・ ゼミ B で扱った重要な DX ライブラリ関数について振り返ります。
・C++と C の違い・・・C++でできることにはどのようなものがあるか、C との違いを軸に解説。(ゼミ B ではわざと C++でできる機能を使ってない部分があります。)
・文字列・・・ゼミ B で軽くしか触れることのできなかった文字列をより深く学ぶ。
・**ファイル出力**・・・ゲームデータのセーブなどに必要な知識などを学ぶ。
・**オブジェクト指向**・・・**【重要】**C++のクラスの利用方法、インスタンス化などを学ぶ。
・**アクションゲーム**・・・これらの応用として、簡単なアクションゲームを作成する。

なお、ゼミ C は任意参加です。　が…

☆ ゼミ B についていけるかどうか不安だった 1 年生へ:ゼミ B の復習・さらにわかりやすいプログラムの書き方をじっくり勉強する手段としてゼミ C をおすすめします。
☆ ゼミ B を余裕で終わらせて、魔改造してしまうような 1 年生へ: オブジェクト指向的な考え方を身につけることによって表現の幅を広げるためにゼミ C をおすすめします。

これでゼミ B は終了です。お疲れ様でした。