

# C++言語でのRPGの制作

石橋 勇太・藤原 健太郎・渡邊 遼平

## 1. 研究概要

高校生活で学んだ技術を使い、C言語の発展形であるC++言語を学ぶ為にゲームの制作をおこなった。

## 2. 研究の具体的内容

最初に担当を決めて各人が、絵、戦闘プログラム、フィールドプログラムに分かれて作業を行った。

開発環境には、Microsoft社の、C++言語の開発ソフト「Microsoft Visual Studio 2008」と「DXライブラリ」を使用することにし、ゲームをC++言語で開発できる環境を整えた。

### (1) C++言語について

C++言語とは1979年に開発され、日本においては「シータスタス」、「シープラブラ」と呼ばれている。

C言語からの派生なので、C言語とは似ているが様々な違いがある。主な違いはC++がオブジェクト指向であるということである。

### (2) 作品について

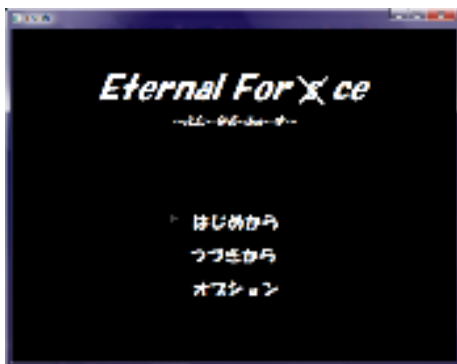


図1 タイトル画面

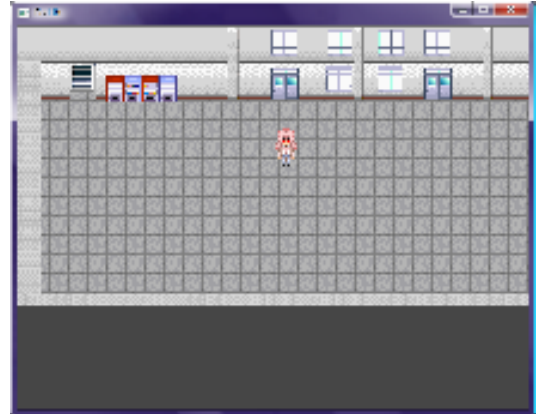


図2 フィールド画面

フィールドではAキーを押すことでメニューを開くことや矢印キーで移動ができる(図2)。画面端まで行くか階段を利用するとマップの切り替えを行うことができる。

また、移動中には敵とのエンカウト(遭遇)が起こる可能性がある。



図3 戦闘画面

フィールド上で敵とエンカウトし、行動を選択すると敵に戦闘ダメージを与える(図3)。



図4 ステータス画面

ステータス画面はフィールド上でメニューを開くことで見ることができる(図4)。ステータス画面では現在のレベルや各能力を確認することができる。

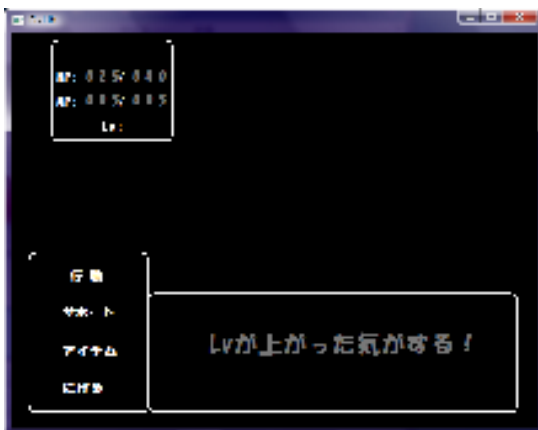


図5 レベルアップ時の戦闘終了画面

敵を倒していくと、倒した敵のレベルに応じて戦闘終了時に経験値が溜まり、それが次のレベルに上がるための必要経験値分を超えると主人公キャラのレベルが上がるようになっている(図5)。



図6 ゲームオーバー

主人公の体力が0になるとゲームオーバー画面が表示され、タイトル画面(図1)に戻るようにになっている(図6)。

```

if(ch.walking_flag==1)
if(GetRand(0)==0){
en = GetRand(1);
ClearDrawScreen();
DrawGraph( 0, 0, sentou, FALSE );//画像を描画
DrawGraph( 250, 150, teki[en], TRUE );//画像を描画
DrawGraph( 0, 0, sentou1, TRUE );//画像を描画
ScreenFlip();
Sleep(1000);
while(!ProcessMessage() && !ClearDrawScreen() && !GetHitKeyStateAll( Ke
ClearDrawScreen();
DrawGraph( 0, 0, sentou, FALSE );//画像を描画
DrawGraph( 250, 150, teki[en], TRUE );//画像を描画
if ( Key[ KEY_INPUT_UP ] == 1 ) { //上ボタンが押されたら
sentomenu--;
if (sentomenu <= -1)
sentomenu = 3;}
else if ( Key[ KEY_INPUT_DOWN ] == 1 ) { //下ボタンが押されたら
sentomenu++;
if (sentomenu >= 4)
sentomenu = 0;}

```

図7 敵とのエンカウト等のプログラム

エンカウト部分のプログラムでは、敵との遭遇は、歩いているときにランダムで数値を出し、その数値が一定の値になったら敵が出現するようになっている。敵と遭遇したあとにどの敵と出会うのかを決める処理を行い、出現する敵を決めたあと敵のステータスを読み込み、敵の画像を表示する(図7)。

```

if (sentomenu == 0)
if ( Key[KEY_INPUT_Z] == 1 ){
speed1 = speed1 - ste1[9];
speed2 = speed2 - ste2[en][9];
if (speed1 <= speed2){
damege1 = ste1[5]-ste2[en][6]+GetRand(5);
if (damege1 <= 0){
damege1 = 1;}
ste2[en][1] = ste2[en][1]-damege1;
PlaySoundMem( Sound, DX_PLAYTYPE_NORMAL );
ddd[0][0] = damege1/100;
dd1 = damege1*100;
ddd[1][0] = dd1/10;
dd1 = dd1*10;
ddd[2][0] = dd1;
for (s=0;s<=2;s++)
for (t=0;t<=0;t++)
dsg[s][t]=sujii[ddd[s][t]];
DrawGraph( 0, 0, dame1, TRUE );//画像を描画
DrawGraph( 388, 357, dsg[0][0], TRUE );//画像を描画
DrawGraph( 404, 357, dsg[1][0], TRUE );//画像を描画
DrawGraph( 420, 357, dsg[2][0], TRUE );//画像を描画
speed1 = speed1 + ste1[9] - 3;
}

```

図8 戦闘ダメージの計算のプログラム

戦闘関連のプログラムでは、戦闘中はステータスの速さが高い方から行動でき、行動をするたびに内部の数値が変動する。

長期戦になるとこの内部数値に差ができるようになっており、速さが高い方が連続攻撃をすることができて有利になる設定になっている。

また、サポートコマンドを利用することで、サポートキャラがサポート攻撃(図9)をおこなってくれて通常よりも多くのダメージを与えることができる。ただしサポート攻撃を利用するとMP(マルチパワー)ポイントを1消費するうえに、上記の内部数値をリセットすることになるので戦闘終盤でのサポート攻撃の使用は不利になるように設定してある(図8)。



図9 サポート攻撃

```

if (brek==3)
while(!ProcessMessage() && !ClearDrawScreen() && !GetHitKeySt)
ClearDrawScreen();
for(s=0;s<=9;s++)
stem[s][0] = stel[s]/100;
st = stel[s]*100;
stem[s][1] = st/10;
st = st*10;
stem[s][2] = st;
}
for(s=0;s<=9;s++)
for(t=0;t<=2;t++)
stemg[s][t]=sujii[stem[s][t]];

DrawGraph( 0, 0, stemm, FALSE ); //画像を描画
DrawGraph( 30, 60, kyara[0][0], TRUE );
DrawGraph( 539, 57, stemg[0][0], TRUE ); //画像を描画
DrawGraph( 555, 57, stemg[0][1], TRUE ); //画像を描画
DrawGraph( 571, 57, stemg[0][2], TRUE ); //画像を描画
DrawGraph( 508, 87, stemg[1][0], TRUE ); //画像を描画
DrawGraph( 524, 87, stemg[1][1], TRUE ); //画像を描画
DrawGraph( 540, 87, stemg[1][2], TRUE ); //画像を描画
DrawGraph( 571, 87, stemg[2][0], TRUE ); //画像を描画
DrawGraph( 587, 87, stemg[2][1], TRUE ); //画像を描画
DrawGraph( 603, 87, stemg[2][2], TRUE ); //画像を描画
DrawGraph( 508, 117, stemg[3][0], TRUE ); //画像を描画

```

図10 ステータス表示のプログラム

ステータス画面を表示するためのプログラムでは能力値を指定したフォントで出力することができなかつたので能力値の各桁を計算し数字の画像を1桁ずつ表示する形式をとつ

ている(図10)。

```

DrawGraph( 175, 73, stemg[4][2], TRUE ); //画像を描画
if (ste2[en][1] <= 0) //敵戦闘不能
ClearDrawScreen();
stel[10]=stel[10]+(ste2[en][0]*2); //経験値
if (stel[10] >= stel[11]){
stel[10]=stel[10]-stel[11];
stel[0]++;
stel[2]=stel[2]+GetRand(4);
stel[1]=stel[2];
stel[4]=stel[4]+GetRand(1);
stel[3]=stel[4];
stel[5]=stel[5]+GetRand(7);
stel[6]=stel[6]+GetRand(2);
stel[7]=stel[7]+GetRand(1);
stel[8]=stel[8]+GetRand(70);
stel[9]=stel[9]+GetRand(3);
stel[11]=stel[11]+GetRand(15)*stel[0];
flag2=1;
}
DrawGraph( 0, 0, sentou, FALSE ); //画像を描画
DrawGraph( 86, 44, stemg[1][0], TRUE ); //画像を描画
DrawGraph( 102, 44, stemg[1][1], TRUE ); //画像を描画
DrawGraph( 118, 44, stemg[1][2], TRUE ); //画像を描画
DrawGraph( 143, 44, stemg[2][0], TRUE ); //画像を描画

```

図11 経験値関係のプログラム

レベルアップのプログラムでは全部の能力が一定の値にランダムに出した数値を足した分だけ能力値が上がる様になっている。そして、ほかのRPGにはない仕様として次のレベルに上がるための必要経験値も、ランダムである程度上下するようになっていいる。つまり、運がよければレベルがすぐに次のレベルに上がるといったこともあるが、運が悪ければ次のレベルまで上がるのに多くの時間がかかるようになっていいる(図11)。

```

}
}
}
ch.img=image[(ch.x%32+ch.y%32)/8 + c];
ch.img2=basho[bx][by]; //
DrawGraph( 0, 0, ch.img2, TRUE );
DrawGraph( ch.x, ch.y, ch.img, TRUE );
ScreenFlip();
}
DrawGraph( 0, 0, gameo, FALSE ); //画
ScreenFlip();
Sleep(5000);
stel[1] = stel[2];
}
}
DxLib_End();
return 0;
}

```

図12 ゲームオーバー等のプログラム

ゲームオーバー関係のプログラム部分では、ゲームオーバーになった後にタイトル画面へ戻り、その際に体力などを回復するようにして

```

basho[0][0] = LoadGraph( "F-1.bmp" );
basho[1][0] = LoadGraph( "F-2.bmp" );
basho[0][1] = LoadGraph( "2F-1.bmp" );
basho[1][1] = LoadGraph( "F-2.bmp" );
basho[0][2] = LoadGraph( "3F-1.bmp" );
basho[1][2] = LoadGraph( "F-2.bmp" );
menuimage = LoadGraph( "menu.png" );
sankaku = LoadGraph( "sankaku.png" );
menumoji = LoadGraph( "menumoji.png" );
itemm1 = LoadGraph( "itemmenu.png" );
itemm2 = LoadGraph( "itemmenu2.png" );
stemm = LoadGraph( "stemenu.png" );
suji[0] = LoadGraph( "0.png" );
suji[1] = LoadGraph( "1.png" );
suji[2] = LoadGraph( "2.png" );
suji[3] = LoadGraph( "3.png" );
suji[4] = LoadGraph( "4.png" );
suji[5] = LoadGraph( "5.png" );
suji[6] = LoadGraph( "6.png" );
suji[7] = LoadGraph( "7.png" );
suji[8] = LoadGraph( "8.png" );
suji[9] = LoadGraph( "9.png" );
sentou = LoadGraph( "sentou.png" );
sentou1 = LoadGraph( "sentou1.png" );
teki[0] = LoadGraph( "teki.png" );
teki[1] = LoadGraph( "teki2.png" );

```

図 1 3 画像の読み込み

画像の読み込み部分では一度、変数に画像を読み込ませてから表示している（図 1 3）。

このようにすることで何度も直接画像を読み込ませるよりも処理を早くすることができる。

### 3. 研究のまとめ

実習で学んだ C 言語を生かせると思い C++ 言語を利用して RPG を制作しようとしたけれど、思った以上に新しく学ぶことが多く、とても大変だったがすこしずつ RPG として作品が完成していくのを見ると今回このテーマを選んでよかったと思った。

しかし、全体的に作業が進むのが遅く、アイテムやストーリーの実装、イベントの配置などが行えず RPG に重大な部分が未実装のままなので RPG としてはまだまだ完成は出来ていないので、今後も制作を続けていき卒業までには作品を満足のいく出来にしたい。

### 4. 感想

今回の実習では、ゲーム制作以上にほかの人と一緒に何かを作っていくことの難しさと大切さを学ぶことができ、これからの生活に役立てていきたいです。

石橋 勇太

3 年生になってから制作にとりかかり、ストーリーやどんなキャラクターを出すのか等の内部設定で結構な時間がかかってしまい、プログラムの方に本格的にとりかかれたのが 6 月頃からになってしまったのもう少し早くから出来ればもっと機能を実装することができただろうと思いました。

藤原 健太郎

C++ を使用してゲームを作るにあたって自分の C 言語への理解がかなり低いことが分かりこれからのために C 言語のことを理解しもっと積極的にプログラムを作っていく場数を踏んで行こうと思いました。

渡邊 遼平

### 5. 参考文献

作者 Dixq

C 言語何でも質問サイト

<http://dixq.net/>

作者 山田 巧

DX ライブラリ置き場

<http://homepage2.nifty.com/natupaji/DxLib/>