

C++によるシューティングゲーム制作

江田伊織 黒田裕希

1. 研究概要

今回の研究では、シューティングゲームの制作を行った。

プログラミングは黒田が、イラスト作成は江田が担当した。

2. 研究の具体的内容

(1) 内容決定までの流れ

昨年、江田がプログラミングコンテストに作品を出した。しかし、内容に納得できず課題研究でリメイクを行う事にした。

リメイクするにおいて、一人で作ると、前回と同じ状況となり、それでは新しい物は作れないと思った。そこで、プログラミングが得意な黒田を加えることで、より理想に近い作品ができると思った。

使用した開発環境は Microsoft 社の「Visual Studio」である。

今回のゲームを制作するうえで選択したプログラミング言語は C++である。なぜなら、DX ライブラリという C++言語専用のゲームライブラリがあるからだ。

DX ライブラリとはアプリケーションの開発に必要なプログラムを使いやすくまとめた形で利用できるようにしたものである。

(2) 制作過程

(a) ゲームの仕様の決定

今回は、ききやま氏開発の「ゆめにつき」の「ファミリーゲーム NASU」のパロディゲームを制作した。

ゲームタイトルは「DONATSU」で、操作キャラクターはプログラミングコンテストに出したゲームのキャラクターである「シュウ」である。

(b) 素材の作成, 収集

キャラクターや背景などのイラスト, 音楽の作成, 収集を行った。

イラスト作成において, 細かな点で原作に寄せることを意識した。特に意識した点は以下の5つである。

(ア) 操作キャラクター

「シュウ」の外見はプログラミングコンテストに出した時点では図1のように線がはっきりしていて, 表情があった。

しかし, 原作の操作キャラクターは輪郭の線なく, 空虚な雰囲気だ。そのため「シュウ」も輪郭の線を描かず, 棒人間に服を着せるイメージで描いた。結果, 「シュウ」の外見は, 図2のようになった。上から, 直立, ジャンプ, 飛行(横移動)である。

同時に, 遠目で見ても違和感がないように, 前のパーツの色を明るくし, 後ろのパーツの色を暗くした。



図1 元のキャラクター



図2 操作キャラクター

(イ) ドーナツ

ドーナツのデザインは、図3のようになった。

「NASU」では、落ちてくるナスを拾うことで加点される。同じように、「DONAT

SU」では、落ちてくるドーナツを拾うことで加点される。

点数は10点、50点、100点となっていて、点が高いほどデザインが鮮やかになっている。



図3 ドーナツの種類

(ウ) 背景

雲の配置とトーンの暗さを、「NASU」(図4)とほぼ同じようにした(図5)。



図4 「NASU」のプレイ画面

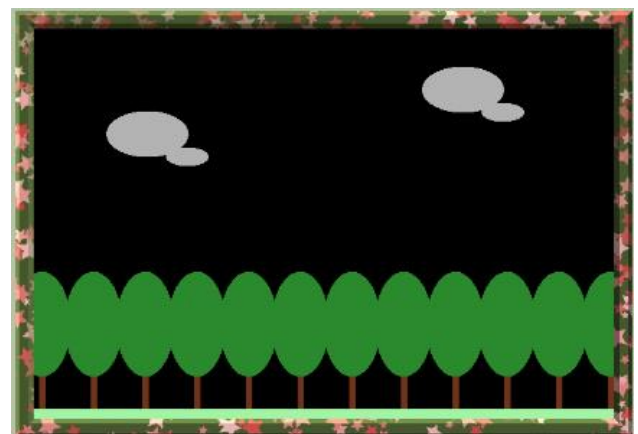


図5 制作ゲームの背景

(エ) 画面サイズ

ウィンドウ画面の比は16:9だが、プレ

イ画面の比は「NASU」と同じの4：3にしている（図6）。



図6 制作ゲームのプレイ画面

(オ) タイトルロゴ

パロディ先（図7）である「NASU」の文字のスペルをヘボン式にすることで、タイトルにいった。「NASU」の文の色は原作と同じ色の紫を使っている（図8）。

NASU

図7 原作のタイトルロゴ

DONATSU

図8 制作ゲームのタイトルロゴ

(c) プログラミング

「Visual Studio」を使用し、プログラミングを行った。

今までの授業で学んだプログラムは、単純なもので、すべてを1ページに収めていた。しかし、今回のような複雑なものだととても見難いため、ファイルを分け整理する方法を学んだ。主な方法は以下の3つである。

(ア) 管理部の作成

管理部を作成することにより、メインループの仕事は管理部にお伺いを立てるだけになり、とても見易くなる。

管理部は、シーンや関数の動きを管理する。

(イ) シーン分け

シーンを分けることにより、その時に必要な関数のみを呼び出すことができる。

図9では各シーンとなっているが、今回はスタート画面、ゲーム画面、ゲームオーバー画面の3つに分けた。

(ウ) 役割分け

シーンの中でも、役割ごとに関数を分けた。

図9のように、今回は、初期化、終了処理、更新、描画の4つに分けた。

初期化と終了処理はシーンが変わるごとに一回でいいが、更新、描画は処理し続けなければならない。これを分けることによって、無駄な処理を減らし、変更するときもわかり易くなった。

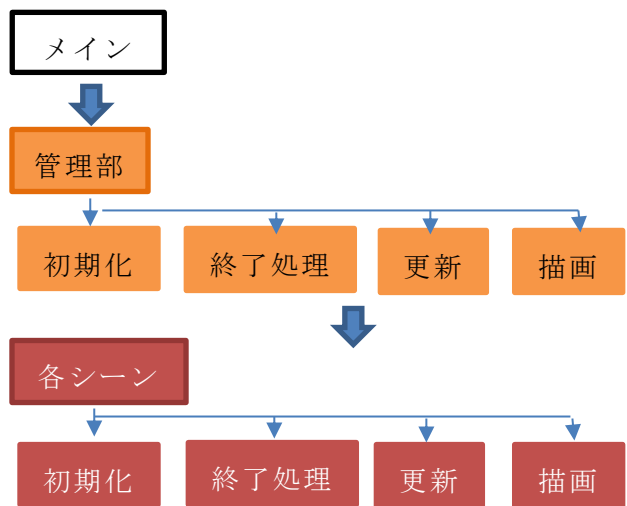


図9 シーンや役割について

この3つを行うことにより、見易さは格段に向上した。

メインループはたったの26行に収まった（図10）。

```
1 #include "DxLib.h"
2 #include "SceneMgr.h"
3 #include "Keyboard.h"
4 #include "Global.h"
5 #include "dSound.h"
6
7 int Ikura, allKey;
8 #if WINAPI #ifndef WINAPI
9 int WINAPI #ifndef WINAPI
10 ChangeWindowMode(TRUE), DxLib_Init(), SetDrawScreen(DX_SCREEN_BACK); //ウィンドウモード変更と初期化と裏画面設定
11
12 SceneMgr_Initialize();
13
14 while (ScreenFlip() == 0 && ProcessMessage() == 0 && ClearDrawScreen() == 0) { //画面更新 & メッセージ処理 & 画面消去
15     Keyboard_Update(); //キーボードの変更
16     SceneMgr_Update(); //更新
17     SceneMgr_Draw(); //描画
18     dSound_Update(); //音源
19 }
20
21 SceneMgr_Finalize();
22
23
24 DxLib_End(); // DLLライブラリ終了処理
25
26 }
```

図10 プログラミングの画面

最終的なファイルの数は14個になった(図11)。

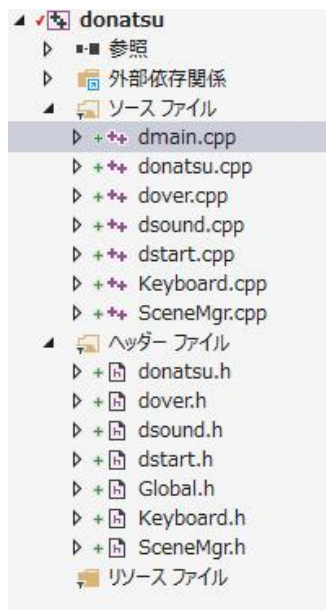


図11 ファイル一覧

(d) テストプレイ

自分たちで行うだけでなく、岡工祭で様々な人達にゲームプレイしてもらい、不具合の調査を行った。

発見した不具合は、

- ・一番左端に出現したドーナツが取れない
→当たり判定の式の中に消し忘れがあった
- ・ゲームオーバーになった後、ゲームを開始するとキャラクターが表示されないことがある。

→画像の初期化を見直すことと、その場し
のぎで追加していったプログラムを整えること
でいつの間にか直っていた

3. 研究のまとめ

ゲーム制作をすることで、計算式の書き方や考え方を知ることができたり、DXライブラリにおけるコマンドの利用方法を学べたり、プログラミングの基本的な知識を得ることができた。

本当は、ストーリーのあるもう少し長いゲームを作ろうと考えていたが、時間が足りなかった。

しかし、それで諦めるのは勿体ないので、

今回のゲームはミニゲームということにして、卒業後も作成を続けようと考えている。

参考文献

- ・「全曲無料・フリー音楽素材/魔王魂」

<https://maoudamashii.jokersounds.com/>

- ・「新・C言語 ～ゲームプログラミングの館～ [DXライブラリ]」

<https://dixq.net/g/>

- ・「シューティングゲーム作成入門 ～bituse～」

<https://bituse.info/game/shot/>

- ・「ポンクソフト」

<https://ponk.jp/>

- ・「プログラマ専用 SNS ミクプラ - トップページ - C言語何でも質問サイト」

<https://dixq.net/forum/>

- ・「DXライブラリ置き場」

<https://dxlib.xsrv.jp/index.html>

- ・「DXライブラリ 関数リファレンスページ」

<https://dxlib.xsrv.jp/dxfunc.html>

- ・「ゆめにつき」

<http://yumenikki-game.tkool.andapp.jp>