

what!!

wow!!

RATIONALE

VOL. 33

Copyright S. R. D. G

Booom!!

目次

代表挨拶	01
高専プロコン30周年記念に際して	02
-プログラミングコンテスト-	
競技部門	
鈴木 暁真	電気システム工学科5年
高橋 謙大	電気システム工学科4年
菊地 輝	電気システム工学科4年 「第30回プログラミングコンテスト」 04
-活動報告-	
鈴木 晴斗	総合工学科 II類 1年 「OpenSiv3Dを使ったゲーム制作」 10
富山 結都	総合工学科 II類 1年 「音楽制作」 12
鈴木 雄裕	機械・エネルギーコース2年 「迷路ゲーム制作」 14
岩佐 佳慧	建築デザインコース3年 「イラスト制作」 16
渡邊 天翔	建築デザインコース3年 「相撲っぽいゲーム」 18
根地戸 龍生	機械・エネルギーコース3年 「ポスター制作」 20
菊地 輝	電機システム工学科4年 「定石作成ツール」 22
鈴木 暁真	電機システム工学科5年 「高専プロコン競技部門のすゝめ」 24
-プロコン旅行記-	
日野 綾瀬	総合工学科 II類 1年 28
編集後記	34

代表挨拶

会長挨拶

機械・エネルギーコース 3年 根地戸 龍生

今年はプロコンが近かったせいか少し慌ただしくなりましたが、無事「Rationale」を完成することができました。部員達がそれぞれテーマを持って「Rationale」の作成に取り掛かりました。この「Rationale」を読んで、少しでも興味を持ってくだされば私達としても幸いです。毎年春休みに部室の大掃除をするのですが、今年も少しリフォームをしました。といってもカーペットを変えただけですが、以前のカーペットの状態が所々ガムテープで補強していた状態でしたので、とても清潔感が増しました。今年の新入生をこの状態で迎えられてよかったです。去年私は副会長を1年間務め、今年は会長を務めることになりました。私が最初部員達に話したことといえば、「いろいろと失敗すると思うので...その...皆さんサポートしてくれると助かります!!!」でした。「本当に大丈夫だろうか...」と各々思いつつも部員達は優秀なのでいつもいろいろと助けてもらっています。こんな私ですが今年1年間部長として部をまとめ上げれるよう頑張っていきたいと思います! 最後になりますが、この「Rationale」の作成に関わった皆様ならびに顧問の北島先生、佐藤先生そして今読んでいる皆様にこの場を借りてお礼申し上げます。

副会長挨拶

ME コース 2年 鈴木 雄裕

Rationale を手に取っていただきありがとうございます。本誌は部員によるイラストや音楽、ゲーム作り等様々な活動をまとめていて読み応えのあるものとなっております。部員の今年の成果を見て、楽しんでいただければ幸いです。去年に引き続き Rationale を Web 上に公開しています。Web では部員が作った音楽が 公開されており画像も色付きで鮮明に公開されています。去年の Rationale も公開しているのでそちらも楽しんでいただければと思います。ゲーム作りの記事では軽くプログラミングについて触れますがプログラミングが分からぬ方でもなんとなく雰囲気はつかめるように書かれています。フローチャートを見て難しそうと思うかもしれませんがあなたが一度本文を読んでみて下さい。どのようにゲームが作られているかわかると思います。最後に Rationale 作成に協力してくださった皆様、そして今この部誌を読んでいる皆様にお礼申し上げます。

人間が使うプログラム

ソフトウェア研究部会 4年 高橋 謙大

高専プロコンの競技部門に始めて参加したのは第 27 回鳥羽大会でした。ゲーム制作が目的で部活に入ったプログラミング歴 0 年、1 年生の私にとってはとても新鮮な経験でした。問題の難易度も高く、競技本番では会場の光の当たり方でピースが認識できずにプログラムが思うように動かないなど、競技部門は厳しい戦いであることを痛感した大会でした。

次に出場したのは 2 年後の第 29 回阿南大会でした。競技課題を見て人間が競技に及ぼす影響が大きいと直感したことを記憶しています。そこで、探索のアルゴリズムの選択とともに人間が間違えにくいようにするという方針を念頭に置いて開発を始めました。が、やはり苦難の連続でした。定期試験期間などで開発が中断することも間々あるなか、強く印象に残っているのは夏休み期間に行った模擬試合です。これは機械学習のための対戦データの蓄積が目的でしたが、一日に何戦も戦うのは苦行でした。結局のところ機械学習は採用しなくなりましたが、競技テーマに対する理解が深まり、その後の探索手法の実装に大いに活かされたと考えています。

さらに、夏休みが明けて開発も佳境に入り、本番環境に近い状態で練習を行った際にも、探索時間に余裕はあるが人間の指示や操作が全く間に合っていないという問題に気付きました。そのため、GUIに対する操作の入力をマウスからキーボードに急遽切り替え、探索に関するパラメータ調整を隨時行いつつ練習に努めました。

いよいよ大会本番。ここに来ても致命的なバグが見つかり、本番環境に適応できずに操作時間が間に合わないなどトラブルは多々ありました。一日目の夜にはそれらのバグを修正し、万が一操作時間に間に合わなくなつたとしてもリカバリができる機能を追加しました。その甲斐もあり、二日目は最良なパフォーマンスを発揮でき優勝することができました。

探索などの計算処理は、基本的に人間は計算機に敵いません。計算機同士では、より効率的に処理を行う方が当然有利です。しかし、それらの間に人間の介入が必要であれば状況は少し変わり、全処理におけるボトルネックが人間の操作であることも少なくありません。また、人間はミスをします。いくら計算機が正しく計算できるとはいっても、大本の人間の操作が間違つていれば意味がありません。現実で動いているプログラムの大半は人間の操作があつて初めて動きます。競技部門は人間の介入は少ないですが、人間が間違つにくくする工夫や、操作性を上げる工夫などは多くの場合で必要だと感じます。人間はミスをするということを念頭に置いて開発を進めることが非常に重要であると改めて気付かされた高専プロコンでした。



第30回全国高等専門学校 プログラミングコンテスト競技部門

電気システム工学科 5年 鈴木 暁真

電気システム工学科 4年 高橋 謙大

電気システム工学科 4年 菊地 輝

1. 高専プロコンとは

今年度も、全国高等専門学校プログラミングコンテスト(以下、高専プロコン)に参加した。高専プロコンとは、全国の高専生がソフトウェアに関わる作品を持ち合い、作品の優劣を競う大会である。与えられたルールに対して速度や正確性を競う競技部門。与えられた課題に沿った作品を作り、実用性や独創性、プレゼンテーション能力を競う課題部門。また課題はないが同様に自由な発想で作られた作品の総合的な能力を競う自由部門の3部門がある。

私達は競技部門に参加し、全国でベスト8という成績となった。ここでは、競技ルール及び開発したプログラムについて解説する。

2. 競技ルール

2.1. 競技概要

今年のテーマは「踊って舞って回って」である。昨年同様対人戦の「陣取りゲーム」であるが、サーバーを介した通信方式に変更されており、フィールドの大きさ、エージェントの人数、同時に進行する試合数の全てがスケールアップしている。この競技は、最終的なフィールドにおける合計ポイントの高さを競う。

2.2. 競技に用いられる用語

2.2.1. フィールド

フィールドとは複数のマスからなる陣取りゲームにおける盤面である。形は矩形で最小 10×10 、最大 20×20 の大きさである。マスには-16～+16点までの点数がつけられている。

2.2.2. エージェント

エージェントとはフィールド上を動く陣取りゲームにおけるコマである。毎ターンそれぞれのエージェントに対して各チームが指示を出す。指示の詳細は3.3.を参照頂きたい。

2.3. 勝敗決定方法について

試合の勝敗は後述する点数計算方法によって計算された合計ポイントが大きいほうが勝ちとなり、等しい場合はタイルポイントが大きいほうの勝ちとなる。

また、リーグの勝敗については以下の順位で決定される。

- ・ 勝率が高いチームを上位とする
- ・ 当該チーム間の試合の勝率が高いチームを上位とする
- ・ 当該チーム間の試合の得失点率の高いチームを上位とする
- ・ 当該リーグ戦全試合の得失点率の高いチームを上位とする
- ・ 主催者が指定する方法で上位を決定する

2.3.1. 点数計算方法

点数はタイルポイントと領域ポイントの2つに分けられる。タイルポイントは自チームのタイルが置いてある場所の点数の総和である。領域ポイントとは自チームのタイルに囲われたマスの絶対値の総和である。図1を例に挙げるとタイルポイントは $2 \times 4 = 8$ の8ポイント、領域ポイントは $| -1 | = 1$ の1ポイントとなり、合計9ポイントである。

1	4	3	4	1
4	0	2	0	4
3	2	-1	2	3
4	0	2	0	4
1	4	3	4	1

図 1 フィールドの例

3. 試合の方式

3.1. 対戦フィールドについて

試合は 1 試合につき 2 対戦行われる。最初に運営側からあらかじめ提示された公開フィールドにて対戦を行う。次に事前情報のない非公開フィールドにて対戦を行う。

3.2. 1 ターンの流れ

1 ターンは作戦ステップと遷移ステップの 2 つに分けられる。作戦ステップはサーバーからフィールド情報を取得し、行動を決定してサーバーに送信する時間である。遷移ステップはサーバーが各チームから受け取った行動情報をもとにフィールドに適用させるための時間である。

1 ターンの時間は 5 秒～15 秒で、遷移ステップは 1 秒で固定であった。

3.3. エージェントの指示について

エージェントには 8 方向($\leftarrow \cdot \nwarrow \cdot \uparrow \cdot \nearrow \rightarrow \cdot \downarrow \cdot \swarrow$)のいずれかに指示を出すことが出来る。指示は「移動」、「タイル除去」、「停留」の 3 種類がある。

「移動」はタイルがないマスもしくは自チームのタイルがあるマスに指示が出来る。「移動」によってタイルがないマスに移動すると、移動先のマスに自チームのタイルを置く。

「タイル除去」はタイルがあるマスに指示が出来る。「タイル除去」をしたターンはマスの移動は行えない。

「停留」はマスの移動を行わずその場にとどまる指示である。「移動」や「タイル除去」が行えないマスに対してそれらの指示を出した場合は「無効」となり「停留」となる。

3.4. 同時に行われる試合数

同時に 1 対 1 の試合がリーグ内で最大 3 試合行われ 4 チームのリーグ(A,B,C,D)では A-B、A-C、A-D、B-C、B-D、C-D の試合が同時に進行される。

3.5. 通信方式について

通信は全てサーバーを介して行う。フィールド情報や試合情報は http の GET 方式、行動情報の送信は http の POST 方式を用いて行われる。送受信するテキストの形式は全て json が用いられた。詳細は高専プロコン公式サイトを参照頂きたい。

4. 開発したプログラムの概要

4.1. プログラムの構成

プログラムは 1 つの実行ファイルで動作し、全て C++ で書かれている。プログラムを実行するとまず試合情報を取得して試合を選択する画面が表示される。行われる試合の選択を終えるとその後はフィールド情報の取得や行動情報の送信が全て自動で動作する。

試合中はマルチスレッドで動作し、GUI、通信部(HTTPCommunication クラス、以下、通信クラス)、各試合(Game クラス)で 1 つずつ独立したスレッドを持つ。次項ではマルチスレッドの仕組みについて解説する。各クラスで独立したスレッドを持たせた理由として

- ・ アルゴリズムの実行時に他クラスの動作に影響を与えないようとする
 - ・ 非同期に実行することにより 1 ターンのタイミング処理を簡便にする
- といったことが挙げられる。

4.2. マルチスレッドの基本動作

今回は試合ごとに個別にスレッドを持たせることを考慮して各クラスのメンバに std::thread の変数を持たせることとした。std::thread とは C++ の STL にて実装されているマルチスレッド実行用のクラスである。Game クラスのソースを例に挙げる(リスト 1)。

試合開始前の非同期実行を開始するタイミングでメンバ関数の ThreadRun()を呼び出すと同じ実体内の Loop()が別スレッドで実行される。Loop()はプログラムが終了するまで無限ループし、各クラスの動作をする関数である。

std::thread の実体はメンバ変数に move され、それぞれの実体で持つことになる。std::thread の join(同期)は実体が破棄されるときにデストラクタで行う。

リスト 1 Game クラスのスレッド実行部

```
1. void Procon30::Game::ThreadRun() {
2.     std::thread th(&Game::Loop, this);
3.     this->thisThread = std::move(th);
4.     return;
5. }
6.
7. void Procon30::Game::Loop() {
8.     while(true) {
9.         //処理
10.        if (programEnd->load() == true)
11.            break;
12.    }
13.    return;
14. }
15.
16. Procon30::Game::~Game() {
17.     if (thisThread.joinable())
18.         thisThread.join();
19. }
20. }
```

4.3. スレッドの終了方法

プログラム終了時のフラグは std::atomic を用いる。std::atomic はスレッド間で同期的にやり取りするデータがある際に一般的に用いられるクラスである。Main 関数で実体を生成し、そのポインタを各クラスに shared_ptr を用いて格納する。今回は GUI をメインスレッドとして扱っているため、GUI が終了したとき(ウィンドウが閉じられたときなど)にプログラムの終了フラグを立て、他クラスの Loop()を終了させている。この処理により確実に全ての thread を join させている。

5. 各クラスの解説

5.1. GUI

GUI は試合や通信の状態を取得して表示を行うが、別スレッドで操作しているデータに不注意にアクセスしてはいけない。そのため GUI に表示したい情報を保持、更新する Observer クラスを用意した。Game クラス、通信クラスに Observer のポインタを shared_ptr を用いて持たせる。ポインタから notify()関数に自身を渡して呼び出すことにより表示される情報の更新を行っている。

GUI は C++ のライブラリである OpenSiv3D を用いた。ラジオボタンを用いて試合を切り替えることにより 1 つのウィンドウで全ての試合の情報を表示できるようにした。また、フィールドの点数とエージェントの表示を分離することにより、シンプルで見やすいものとした(図 2)。



図 2 作成した GUI

5.2. HTTPCommunication(通信クラス)

通信部は C 言語のライブラリである libcurl を用いた。http の通信を開始するときは全て std::async を用いて非同期に実行している。これは Game クラスから送信すべき行動情報が送られてくるからである。しかし、複数の通信を非同期に実行する方式はとらなかった。これは、libcurl が C 言語のライブラリである故に、C 言語と C++ のマルチスレッド記法の違いにより生まれる混乱を防ぐためだ。

プログラム間の送受信に用いるデータは全て 1 度ファイルに保存している。そのため、Game クラスとの試合に関わる情報のやり取りはファイルパスをやり取りすることで行っている。これにより、整形すると数千行にもおよぶ巨大な json ファイルのデータのやり取りによるパフォーマンスの低下を防ぐ。

5.3. Game(試合クラス)

Game クラスにおける 1 ターンのサイクルは通信クラスから情報を受け取り、次の行動をアルゴリズムにより決定。行動情報を json ファイルに書き出し、ファイルパスを通信クラスに送るといった流れである。ターンのサイクルは無限ループを回して時間で管理という方式はとらず、次のターンのデータが受信されたことを通知されるまで Game クラスのスレッドをロックした。これにより冗長な書き方を無くし、シンプルに 1 ターンのサイクルを実装出来ている。

json ファイルの読み取りは OpenSiv3D の JsonReader クラスを使用した。

6. アルゴリズム

アルゴリズムは交互ビームサーチを採用した。通常のビームサーチでは自チームの情報しか考慮しないため、相手の動き方によっては損をする可能性がある。交互ビームサーチは先に敵チームにおいて探索を行い、行動する手を予測することによって最適な手を判断しようというものである。

しかし、各エージェントが取りうる全ての方向の組み合わせをとると最大 8 体の 8 方向であるため、1 ターン読むのに $8^8 = 1677$ 万通りを試さないといけない。今回の制限時間では全ての方向に対して探索を行った場合、深く探索出来ないため、良い手を見つけるのは困難である。そこであらかじめ各エージェントが取りうると予想する動きを決め打ちすることにより、計算量の削減を図った。例えばエージェントが 8 体の場合、3 方向まで絞ることにより $3^8 = 512$ 通りとなり現実的な時間で探索をおこなうことが可能である。

また、今回使用したノート PC の CPU が 6 コア 12 スレッドであることを考え、ビームサーチの並列化を行った。CPU のスレッド数に達するまでは線形に計算時間の減少が期待できる。

7. おわりに

今回の開発では本格的なマルチスレッドプログラミングに挑戦した。複雑で制約の多いプログラミングであったが、とてもいい経験であった。また、通信部も初日は不具合が生じたが、2 日目は好調であった。アルゴリズムの不具合によりベスト 8 という結果になり、近年と比べると少し劣ってしまったが技術的な実りのあった高専プロコンであったと思う。

参考文献

高専プロコン公式サイト

<http://www.procon.gr.jp/>

今年度の競技部門開発リポジトリ

<https://github.com/YASAI03/KosenProcon30N>
atori

OpenSiv3D

<https://github.com/Siv3D/OpenSiv3D>

libcurl

<https://curl.haxx.se/libcurl/>



OpenSiv3D を使ったゲーム制作

総合工学科 II類 1年 鈴木 晴斗

1. はじめに

私は OpenSiv3D を使い、アクションゲームを制作しました。OpenSiv3D とは C++ を使い、ゲームやアプリを作ることができるオープンソースのライブラリです。これを用いることで、ゲーム内のキャラクターの操作、当たり判定などの実装が簡単になります。

2. ゲーム制作

2.1. テクスチャの制作

最初にゲーム画面に表示するテクスチャをペイント 3D で制作しました。制作途中に、このゲームにレトロゲームのような雰囲気を出したいと思い、テクスチャはできるだけシンプルに制作しました。図1はキャラクターの画像です。

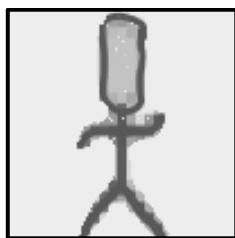


図1 キャラクター

2.2. クラスの宣言

次にクラスの宣言をしました。クラスの宣言とは、クラスの性質や状態、機能をまとめることです。リスト1が実際のコードです。例えば、キャラクタークラスでは、キャラクターのスタート位置、画像のポリゴン、ジャンプしている時間などを宣言しています。また、public メンバを使うことで、プログラム中のどこからでもメンバにアクセスすることができます。

リスト1 クラスの宣言

```
class Character {  
public:  
    Point mybeforePosition;  
    Point myPosition;  
    Polygon polygon;  
    Texture textureCharacter;  
    int jumpTime;  
    bool isGround;  
};  
class Needle {  
public:  
    Polygon polygon;  
    Texture textureNeedle;  
    bool Needle;  
};  
class GLine {  
public:  
    Polygon polygon;  
    Texture textureGoalLine;  
    bool GLine;  
};
```

2.3. キャラクターの移動

次にキャラクターの操作です。ゲーム画面の右を x 軸の正の方向、下を y 軸の正の方向として考えます。リスト2は左右キーで移動するためのコードです。左キーを押したときは x-、つまり負の方向に移動します。反対に右キーを押したときは x+、正の方向に移動します。リスト3はスペースキーでジャンプするためのコードです。スペースキーを押し、キャラクターがジャンプし地

面につくまでの時間を20とし、その値が1以上(上昇中)の時はy-、負の方向に移動します。反対にそれ以外の時(下降中)の時はy+、正の方向に移動します。

リスト2 左右キーでの移動

```
if (KeyLeft.pressed()) {  
    humman.myPosition.x -= delta;  
}  
  
if (KeyRight.pressed()) {  
    humman.myPosition.x += delta;  
}
```

リスト3 スペースキーでのジャンプ

```
if (KeySpace.pressed() && humman.isGround){  
    humman.jumpTime = 20;  
}  
  
if (humman.jumpTime >= 1) {  
    humman.jumpTime--;  
    humman.myPosition.y -= 11.9;  
}  
else {  
    humman.myPosition.y += 12;  
}
```

2.4. 当たり判定

次に当たり判定です。まず、ゲームで使用するテクスチャのポリゴンを取得します。次にg.intersects(h)を使いキャラクターと各テクスチャが交差しているかを調べます。これをしてことで、当たり判定ができるようになります。また、とげに当たった時はリザルト画面、ゴールラインを越えたときはクリア画面を表示するようにしました。それぞれの画面ではゲームのリスタート・終了ボタンを表示するようにし、g.mouseOver()でマウスカーソルがボタンと重なっているか、g.leftClicked()でクリックされたかをそれぞれ調べ、各ボタンの操作ができるようにしました。

2.5. テクスチャの配置

最後に2.1で作成したテクスチャを二次元配列を使い、ステージのどこに配置するかを決めます。できるだけ難易度を高くしようと思い、とげをたくさん配置しました。最終的には図2のようになりました。

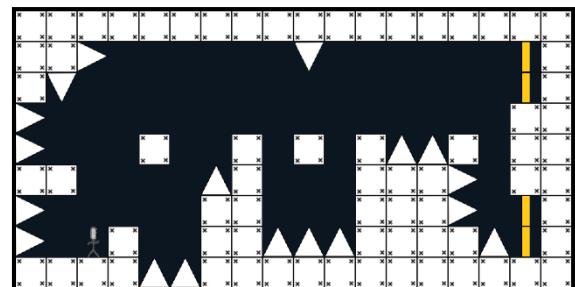


図2 ゲーム画面

3. おわりに

私は、今回初めてゲームを作りました。プログラミングについてまだまだ知らないことがたくさんあり、先輩に教えていただきながら完成させることができました。作業は自分が当初思っていた以上に難しく、特に当たり判定のコードを書く作業は苦戦しました。これからもプログラミングの勉強に励み、スキルの向上を目指していきたいです。

参考文献

OpenSiv3D リファレンス
<https://siv3d.github.io/ja-jp/>

cpprefjp - C++日本語リファレンス
<https://cpprefjp.github.io/>

音楽制作

～初心者の時間～

1. はじめに

私は音楽編集ソフトの Studio One を使って音楽制作をしました。このソフトは有料版と無償版があり公式サイトからダウンロードすることができます。無償版でも十分な楽器やエフェクトがついており、作れない音はほとんどありません。インターネット上に使い方の解説したウェブサイトが沢山あるため、すぐにこのソフトが使えるようになります。楽しく音楽を作ることができました。図1は音楽制作中の画面です。

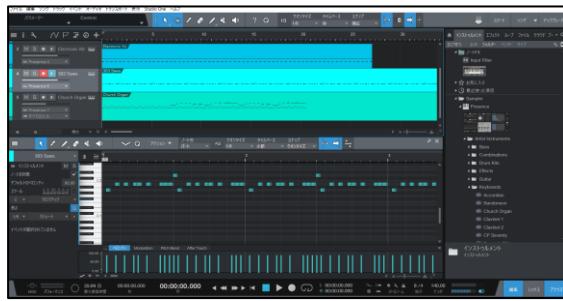


図1 音楽制作中

2. DTM と DAW の話

インターネット上の解説を読む上で知つておいた方が良い言葉として、「DTM」と「DAW」があります。「DTM」とはパソコンで音楽を作る事を指します。そして「DAW」とは前項で紹介した Studio One のような音楽を作るためのソフトを指します。「DAW」はほかにもたくさんあり、人それぞれ合うソフトが違います。

3. 周辺機器

メロディー等を作る際に、midi キーボードがあると、楽に作ることができます。midi キーボードはパソコンに繋げ、鍵盤を叩くと対応した信号

総合工学科1年 II類 富山 結都
が打ち込まれるという物です。ただ打ち込まれるだけでなく、音の強弱もつけることができ、物によってはビブラートもつけることもできます。これによってマウスカーソルを、ピアノロールにつづつ合わせクリックするという心躍らない作業ではなくなります。

またオーディオインターフェイスという機械があります。この機械は PC に出力される音声信号を処理するものです。ほとんどのパソコンに入っていますが、音質や性能はそこまで高くないう場合もあるので、なるべく高音質で聞きたい人向けです。

4. コードの話

コードとは二音以上での音の重なりの事です。基本形は三つの音でできたトライアド(三和音)といい、そこから派生しセブンスコードやサスフオーコードがあります。また、コードは大きくメジャーとマイナーに分けることができます。メジャーは明るい印象で安心感を持たせます(例:C,F)。マイナーは暗いイメージがわき不安定な気持ちにさせます(例:Cm,Fm)。

5. コード進行の話

コード進行とはコードを組み合わせていく音の流れです。コード進行が変われば曲の雰囲気がガラッと変わり、陽気な気分や期待が高まる曲調になります。よく使われるコード進行にはカノン進行や小室哲哉進行などがあり、J-POPなどでよく聞きます。カノン進行とは、パッヘルベルの「カノン」というクラシックの曲に使われたコード進行です。このコード進行は、「負けないで」や「勇気100%」等、有名な曲に使われてい

ます。小室哲哉進行とはそのまま小室哲哉が愛用したコード進行で、「Get Wild」や「愛唄」に使われています。最近ではボーカロイド曲にもよく使われています。

6. 音楽の制作手順

音楽を作る上では、まず全体像を考えるところから始めます。サイケデリックな曲なのか牧歌的な曲なのか、拍子は何分の何拍子かを決めます。次にメロディーを考えます。流れができたら録音、またはその場で「DAW」に打ち込みます。そのあとはコードとリズムに脳を働かせます。コード進行は初心者がすぐに作れる代物ではないので参考にしましょう。先ほど紹介したカノン進行や小室哲哉進行はサビにあてれば、寂しかった自分のメロディーが輝きだすでしょう。またコード進行がたくさん載っているサイトページに U-FRET があります。このサイトには様々な曲のコード進行が載っています。自分のつくりたい曲の雰囲気に合ったコード進行を参考にしましょう。図2に普段私が音楽を作るときに行っている工程を示します。

7. 終わりに

この文章を書くにあたり、音楽の専門用語のみならず「DTM」の用語も調べ始め、コードの読み方も確認しました。これを書く前は音楽を作るにはどうしたらいいか、もやもやしていました。しかし、今ではきちんとした制作手順を理解できています。私は初めて Studio One を触りましたが、操作自体は一日もたてばすぐにできるようになりました。音楽を難しいと思っている人にこそお勧めします。今はまだ理想の音楽が作れませんが、自分が最高だと思う音楽を作れるようになりたいです。

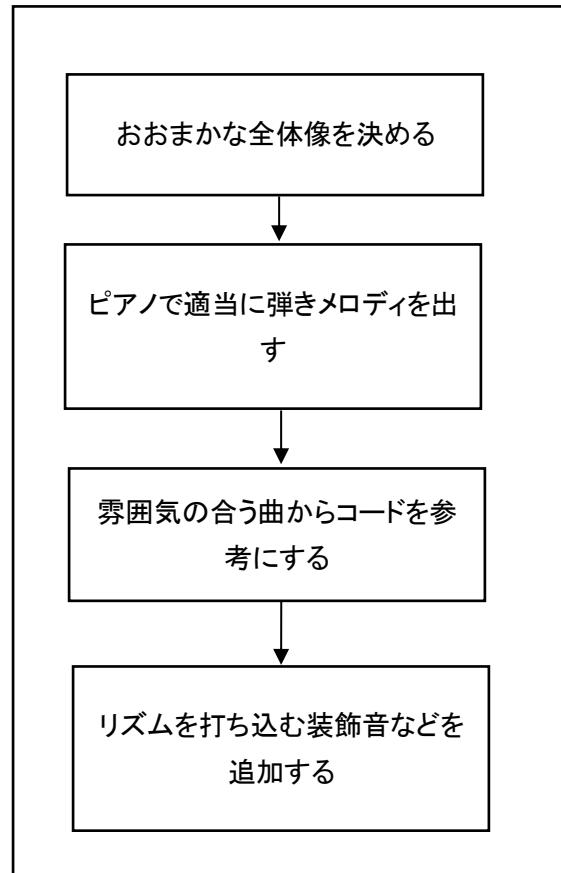


図2 作曲工程

参考文献

(株)ネクスト・デザイン

<https://www.nextdesign-jp.com>

たむぎすた～

tamusguitar.com

うちやま作曲教室

sakkyoku.info

Make Inspiring Beats

<https://www.mib-dtm.com/>

U-FRET

<https://www.ufret.jp/>

迷路ゲーム制作

機械・エネルギーコース 2年 鈴木 雄裕

1. はじめに

私は CreateJS という JavaScript ライブラリを使い、迷路ゲームを作りました。ゲーム部分は JavaScript で作り、スタート画面などは HTML で作りました。今回は JavaScript の学習も兼ねてこのゲームを作りました。

2. CreateJS について

CreateJS とは HTML で滑らかに動くものを作る JavaScript ライブラリのことです。これを使うと綺麗な描画表現ができます。JavaScript は Web ページ上で実行されるプログラミング言語です。

3. ゲームの仕様

このゲームはプレイヤーを移動させてゴールを目指し、タイムを競います。図1はゲーム画面です。十字ボタンで自機を操作し、自機がゴールに触れたらゲームクリアです。もしミスをしてもりセットボタンでゲーム開始時の状態に戻すことができます。

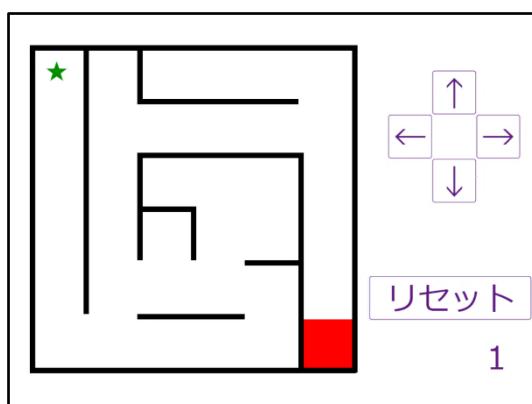


図1 ゲーム画面

4. ゲーム仕様

始めにゲーム全体の仕様を説明します。まず迷路やボタン、自機を作成します。次に移動ボタンが押されたら時間をカウントし始めます。

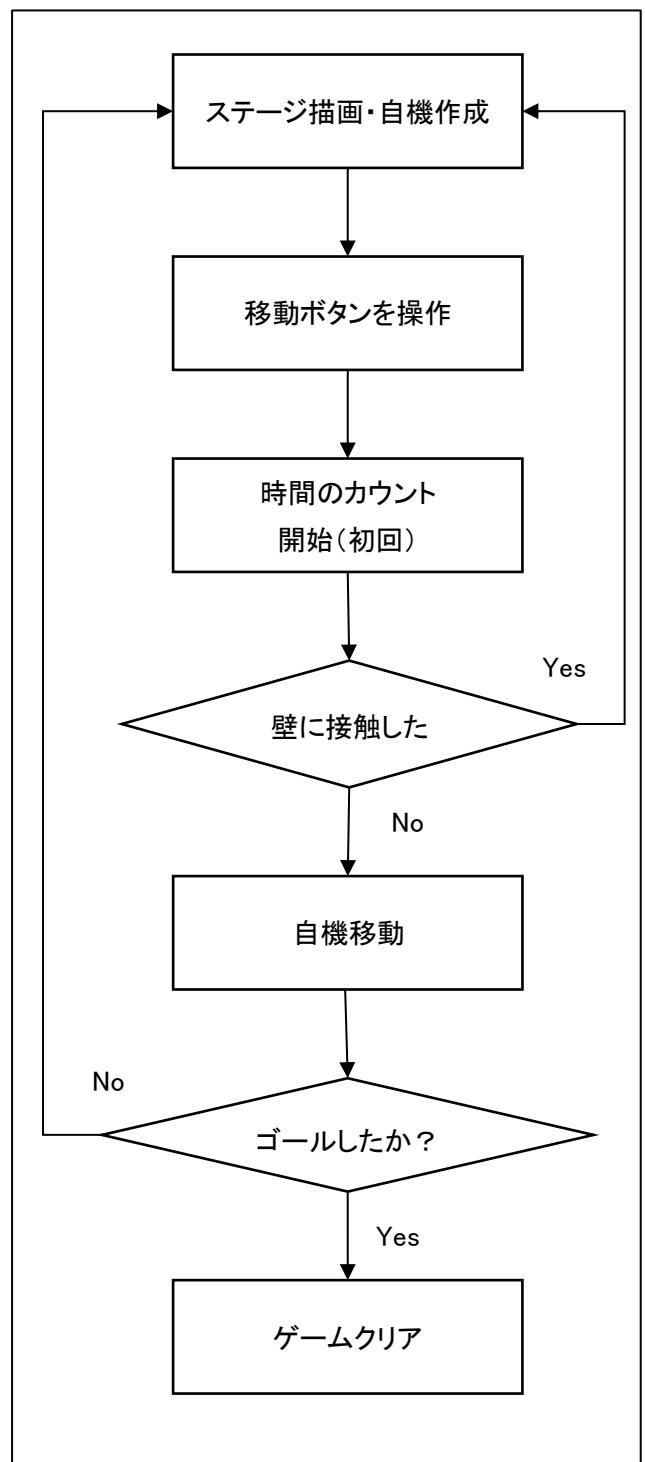


図2 フローチャート

壁に当たっているか判定し、当たっていないければ移動ボタンの向きに移動し続けます。ゴールに当たらクリア画面に遷移します。ゴールするまでステージの描画からゴール判定まで繰り返します。

5. 接触判定のプログラム

フローチャートの壁との接触判定について説明します。リスト1は壁との接触判定のプログラムです。

リスト1 接触判定

```
1. var point = miniStar.localToLocal  
2. (0, 0, wall);  
3. var isHit = wall.hitTest(point.x, point.y);  
4. if (isHit == true) {  
5.     miniStar.x -= vx;  
6.     miniStar.y -= vy;  
7.     vx = 0;  
8.     vy = 0;  
9.     miniStar.graphics.clear()  
10.    .beginFill("yellow")  
11.    .drawPolyStar(0, 0, 20, 5, 0.6, -90);  
12. }  
13. else {  
14.     miniStar.graphics.clear()  
15.     .beginFill("Green")  
16.     .drawPolyStar(0, 0, 20, 5, 0.6, -90);  
17. }  
18. miniStar.x += vx;  
19. miniStar.y += vy;
```

1行目で自機の座標を求めます。3行目で自機が壁に当たっているか判定します。4行目から8行目で壁に当たった場合の反応を決めます。壁に当たった場合、当たった瞬間とは逆の向きに動き、壁から少し離れたところで止まります。同時に壁に当たったか分かりやすいように自機の色を緑から黄色に変えます。壁から離れたら色を緑に戻します。一連の確認が終わったら自機が動き始めます。ゴール判定は壁の当たり判定と同様にゴールに当たっているか判定します。もし当たっていればクリアタイムをブラウザに保

存し、クリア画面で自分のクリアタイムが表示されます。

6. リセットボタン

まずボタンを描画します。もしこのボタンが押されたらリスト2のプログラムが動作します。

リスト2 リセットボタン

```
1. button.addEventListener  
2. ("click", handleClick);  
3. function handleClick(event) {  
4.     miniStar.x = 100;  
5.     miniStar.y = 100;  
6.     vx = 0;  
7.     vy = 0;  
8.     var now = new Date();  
9.     var end = Date.now();  
10.    start = end;  
11.    isStarted = false;  
12. }
```

初めに自機を初期位置に戻します。次に時間のカウントを始める前の状態と同じ条件にします。そしてゲームの進行を停止します。進行を停止しないとリセットしたあとすぐに時間のカウントが始まるからです。

7. おわりに

今回初めてのゲーム製作で JavaScript やゲームの設計など多くのことを先輩から教わりました。何とか形にすることはできましたが勉強不足が身に沁みました。これからも勉強に励み、よりよいゲームを作れるよう頑張ります。

参考文献

CreateJS

<https://ics.media/tutorial-createjs/basic/>

WEB 色見本 原色大辞典-HTML カラーコード

<https://www.colordic.org/>

イラスト制作

建築デザインコース 3年 岩佐 佳慧

1. はじめに

私はスマートフォンの無料イラストアプリ ibisPaintX を使いデジタルイラストを制作しました。今回は iPad と ApplePencil を使用し、制作にあたりました。制作にあたり、自分がお勧めする、イラストの描き方を紹介したいと思います。

2. 構想

まず、自分がどんなものを描きたいかざっくりと文字に書き起こしていきます。描きたいものが具体的に決まるので作業効率が上がります。

3. レイヤー

レイヤーとは、透明な紙のようなもので、作品のパーツを複数のレイヤーに分けて描画することで、レイヤーごとに編集したり、レイヤーの重ね順を変更する、表示するレイヤーを選択するといった操作が可能になり、複雑な作業を効率よく行うことができます。しかし、レイヤーは使う枚数の分だけ余計にメモリを消費するので、枚数が極端に多くなるとメモリ不足を招くがあるので注意しましょう。

4. メイキング

4.1. 制作準備

今回の用紙サイズは 1100×1100px の正方形、解像度は 350dpi で作成しました。描くものによってサイズは変えるのが良いでしょう。解像度は上げるほど綺麗なイラストを描くことが可能ですが、制作環境によってはソフトの動作が遅くなることがあるので 300～350dpi が丁度良いでしょう。

4.2. ラフ

まず、先程の構想を元に簡単にラフを描いてから制作にあたりました。制作はラフ、線画、色塗りという順で行いました。ラフとは想像したイラストのイメージを、時間をかけず簡単に描画したものです(図 1)。



図 1 ラフ画

4.3. 線画

ラフが終わったら、次は線画です(図 2)。線画を描くときは手順を二つに分けて行いました。最初は正確な線画を描き、その後立体感を出す為に線画に加筆し線を太くするという手法です。加筆する部分としては主に影が落ちる部分に接した線です。例としては、前髪から肌に落ちる影を意識して、前髪の流れの交点に線を描き加えていきます。また、細かい作業がある時は、髪、体、顔もレイヤーや色を分けて行っても良いでしょう。



図 2 線画

4.4. 塗り

線画まで終わったら、塗りに入ります。塗るパートごとにレイヤーを分けたら、塗りつぶしツールで各パートを塗っていきましょう。塗りつぶしツールの効果が他のパートにまで及んでしまった場合、一旦線画の作業に戻り修正しました。また、髪の毛の先端や指先などは、塗り残しやはみ出しが生まれてくる箇所なので注意しながら描きました。



図 3 描画ツール:乗算

4.5. 描画ツール:乗算

影や光を描画したい場合は描画ツールを積極的に使っていきましょう。描画ツールとは下のレイヤーに対しての合成が設定できる機能の事です。ここでは影をつけるため乗算を使いました。乗算は下にあるレイヤーの色と、設定中のレイヤーの色を掛け合わせて合成します。合成後は、元の色より暗い色になるため、影を描きたい場合にお勧めです(図 3)。



図 4 描画ツール:加算(発光)

4.6. 描画ツール:加算(発光)

加算・発光は下にあるレイヤーの色と、設定中のレイヤーの色を足します。色を加算すると明るい色に変化するので光を描きたい場合に使用すると綺麗に描くことができます(図 4)。

4.7. 仕上げ

塗り残しや違和感を修正したら完成です(図 5)。



図 5 完成図

5. 終わりに

今回描いた絵の Rationale 版はグレースケール、ポスター やオンライン版ではカラーで掲示されていますので是非ご覧ください。

また、私は絵を描き始めて 1 年くらいですが、昨年よりも絵を描くための手法を少しづつ実践に持ち込めるようになりました。これからも今まで以上に上手い絵を描けるように頑張りたいと思います。

参考文献

イラスト・マンガ描き方ナビ

<https://www.clipstudio.net/oekaki>

相撲っぽいゲーム

～相撲ではない～

1. はじめに

私は OpenSiv3D を使用して以前から制作をしてみたいと思っていた、1対1で対戦できる格闘ゲームを制作しました。このゲームの処理の内、特に重要なプレイヤーの攻撃の動きなどを中心に説明していきたいと思います。

2. OpenSiv3D について

OpenSiv3D とは、C++で簡単に GUI アプリケーションを作れるライブラリです。OpenSiv3D では、コンソール画面上ではなく、ウィンドウを表示してグラフィックスやオーディオを自由自在に操ることができます。また、ゲーム制作に深く関わるオブジェクトの当たり判定なども簡単にプログラムする事が出来ます。

3. ゲームの仕様

今回制作したゲームは、1対1で押し合い場外に飛ばし合うゲームです(図1)。2人で遊ぶゲームで、それぞれがキャラクター(図2)を動かします。アクションの種類は体当たり、ガード、カウンター、ガード崩し、下攻撃の5種類です。場外に出た場合は、ライフが1つ減り画面中央の上部から復活します。

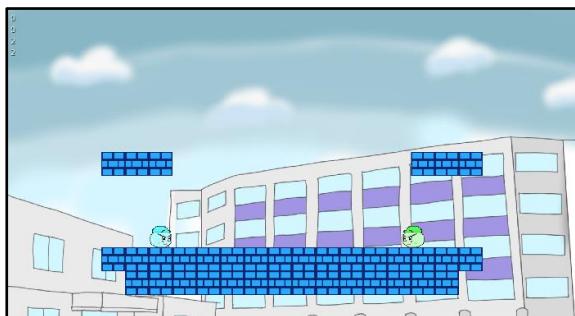


図1 ゲームの全体像

建築デザインコース 3年 渡邊 天翔



図2 操作するキャラクター

4. プレイヤー同士の当たり判定

プレイヤー同士の接触判定について解説します。接触判定のプログラムを下のリスト1に記載します。

リスト1 プレイヤー同士の接触判定

```
1. if (player.directionX == 1) {  
2.     if (intersects(player.m_positionR)) {  
3.         directionX = -1;  
4.         m_damageFrame = 10;  
5.         pushRate += 2;  
6.         m_isJump = false;  
7.         m_actionFrame = 0;  
8.     }  
9. }  
10. else if (player.directionX == -1) {  
11.     if (intersects(player.m_positionL)) {  
12.         directionX = 1;  
13.         m_damageFrame = 10;  
14.         pushRate += 2;  
15.         m_isJump = false;  
16.         m_actionFrame = 0;  
17.     }  
18. }
```

1行目では相手の攻撃時の体の向きを判定します。相手の向きを判定することで自分がダメージを受けた場合に、どちらの向きにノックバックするかがわかります。2行目で自分の体と相手の体の前部が接触しているか判定します。3行目で自分の体の向きを相手の向きと反対にします。4行目でダメージフレームを10フレームに変更します。ダメージフレームとは何フレーム分キャラクターがアクションを行えないかを表しており、これが0より大きい場合キャラクターはアクションを行うことができません。5行目で1フレーム毎にどのくらいノックバックするかを設定(加算)します。6行目でジャンプフレーム(あと何フレーム分ジャンプできるか判断するための数)を0にしてジャンプをキャンセルします。7行目でアクションフレーム(自分の行動を何フレーム行うかの数)を0にします。

5. ゲームの流れ

ゲームの大まかな流れをフローチャートに示します(図3)。はじめに Wii リモコンの接続をします。ゲームをスタートし、ステージを読み込むと、ゲームがスタートします。ゲームがスタートしたら、プレイヤーの移動からオブジェクトとプレイヤーの描画までの処理を繰り返します。プレイヤーが画面外に落下すると落ちたプレイヤーのライフ数を判定し、0であるならばリザルト画面に移動します。その後 Enter キーを押すことで、スタート画面に戻り、これらの過程を繰り返します。

6. おわりに

今回初めて設計からゲーム制作をしたのですが、思ってたより去年と比べて出来るようになったことが多く、完成度も高かったと思います。

クラス分けなどまだ慣れないところがありますが、日々精進してより完成度の高いゲームを作れるようになれたら良いと思っています。また、来年も違うゲームを作りたいと思っています。

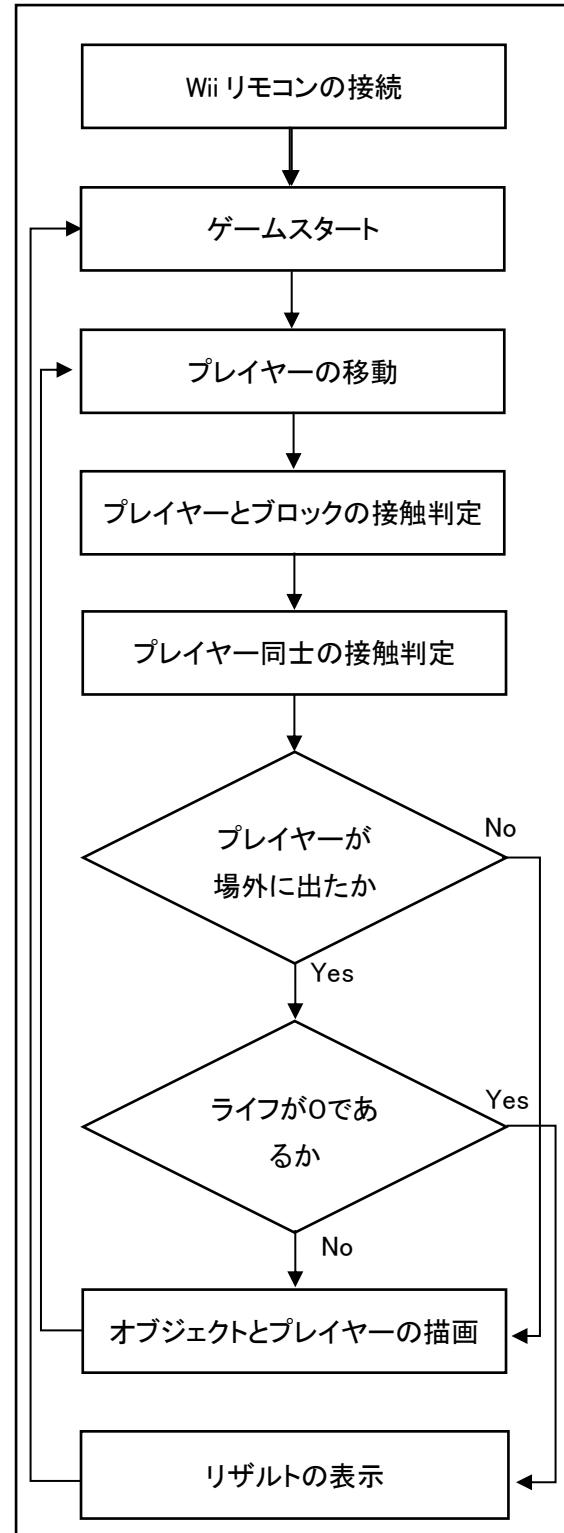


図3 ゲームのフローチャート

参考文献

OpenSiv3D

<https://github.com/Siv3D/OpenSiv3D>

ポスター制作

機械・エネルギーコース 3年 根地戸 龍生

1. はじめに

私がイラスト制作にあたって使用したソフトは、CLIP STUDIOです。このソフトはイラスト以外にも漫画やアニメーションを作ることができます。今回のイラストは風景画にしようか迷いましたが、校内に貼る予定のポスターでもあったのでキャラクターの絵を描きました。

2. メイキング

2.1. 下書き

まず線を大体の構図で描いていきました。色は後で変えられるので何色でも良いですが、線画の線が見えづらくならないような色にしました。体や顔を描く際、バランスが変になってしまう場合があるので下書きを描く前に、より大まかなラフ画を描くのも良いと思います。ポイントは、肘や膝などの関節部を意識して描くことです(図1)。

2.2. 線画

線画は“ペン”的サブツールの“G ペン”を使いました。下書きとは違い一本の線で描きました。線を描くとき線がブレないようにするために、手ブレ補正という機能を使いました。初めて描くときに違和感が生じるかもしれません、これを使うとききれいな線が描けます。この後色分けをする際、線と線の間に隙間があるとそこから色がはみ出てしまうので、なるべく間を閉じて描きました。

また今回は、ベクターレイヤーに線画を描きました。このレイヤーを使う利点は、線と線がはみ出した時その部分のみを消したり、描いた線の形や太さを変えることができるところです(図1)。



図1 左:下書き 右:線画

2.3. 色分け

色分けは“塗りつぶし”を使いました。“塗りつぶし”的サブツールは複数あり、私は“他レイヤーを参照”を使いました。この場合、他レイヤーとは線画のレイヤーのことです。もしも色が漏れ出た場合は、隙間閉じを使うと多少の隙間でも塗りつぶせます(図2)。



図2 色分け

2.4. 影、ハイライト

影とハイライトは“水彩”と“色混ぜ”、“エアブラシ”を使いました。光の当たり方を意識して描きました。影とハイライトの境界を“色混ぜ”でぼかして描きました。

2.5. 背景

背景には“グラデーション”的“描画色から背景色”を使いました。キャンバス上でドラッグすると、グラデーションが描画されます。また三色以上でグラデーションを描画したいときは、“グラデーション”的下部をクリックすると色を追加することができます(図3)。

2.6. 乗算

髪の先端には乗算をかけました。乗算とは下にあるレイヤー設定中のレイヤーの色を掛け合わせて合成するものです。乗算をかけた部分は、元の色より暗い色になります(図4)。

2.7. 文字

今回はポスターということで、文字を入れました。文字は“テキスト”を使いました。“テキスト”に文字を入力した後、文字の大きさを変えたり、フォントを変えたりすることができます。またこのレイヤーの上に新しいレイヤーを乗せクリッピングすることで、文字にグラデーションをかけることができます(図5)。

3. おわりに

私は CLIP STUDIO を約二年間使ってきました。少し慣れてきましたが、機能が多く扱うのが難しいです。今回はキャラクターの絵を描きましたが、次は建物や風景などを描いてみたいと思います。

参考文献

- CLIP STUDIO 創作応援サイト
https://www.clip-studio.com/clip_site/
- イラスト・マンガ描き方ナビ
<https://www.clipstudio.net/oekaki>
- CLIP STUDIO TIPS
<https://tips.clip-studio.com/ja-jp/>



図3 背景



図4 上:乗算前 下:乗算後



図5 完成

定石作成ツール

～人力補助ツール～

電気システム工学科 4年 菊地 輝

1. はじめに

このプログラムは高専プロコンで使用するために作成しました。今年の競技部門をざっくり説明すると、決められたフィールドで陣取りを行う競技です。

陣取りゲームを行う際に BeamSearch で探索を行うのですが、どうしても近い場所にいるエージェントは同じ方向に向かってしまいます。また、あまり効率の良い陣の取り方が出来なかつたりします。

そこで定石として最初の数ターンの行動を埋め込み、探索結果に依らずに必ず指示通りに動かすようにします。これにより、エージェントが密集しなくなり効率よく陣を取ることが出来ます。

2. 作成環境

定石作成ツールを開発するにあたり使用した環境は以下の通りです。

- Visual Studio 2019
- OpenSiv3D(0.4.1)

3. 仕様について

定石の仕様は以下のリストになっています。

リスト1 定石の仕様

1. 自チームエージェントの誰かの座標
2. 相手チームエージェントの誰かの座標
3. エージェント数
4. 埋め込むターン数
5. 自チームの座標 x,y 行動の方向 dx,dy
6. 相手チームの座標 x,y 行動の方向 dx,dy

1行目と2行目は5行目と6行目のどちらの行動をとるか判定するためあります。フィールドには自チームと相手チームの初期座標が記載されています。対戦の際にどちらに配置されるかわからぬため、2パターンの定石を埋め込む必要があります。

3行目はチームのエージェント数です。これは5行目および6行目の行動情報を読み取る際に使用します。

4行目には埋め込むターン数を書き込みます。リスト1では1ターンを想定した埋め込みですが、2ターン3ターンと増えた場合には自チーム、相手チームの行動情報が下に続きます。

行動情報ですが、各エージェントを特定するために現在いる座標(x,y)をまず書き込みます。そこから周囲8マスのどの方向に移動するかの情報(dx,dy)を書き込みます。具体的には、右に行く際は(1,0)で上は(0,-1)になります。これをエージェントの人数分行います。

これら全てを書き込んだファイルが定石ファイルとなります。しかし、これを全て人間の手で書くのはとても非効率なので、プログラムに任せることにしました。

4. 実装

OpenSiv3D で実装していきます。定石ファイルを作成するにはTextWriterを使用します。GUIは、なるべく人間が見てもわかりやすくなるように盤面描画も行います(図1)。

盤面には初期座標に配置されたエージェントが描画され、上から順番に行動を選択していきます。行動はGUI右側にある8方向の矢印ボタンで選択していきます。

また、各行動が分かりやすいうように移動の方向を矢印として描画することで定石作成におけるミスを減らしています。本当は初期座標からの全遷移の様子を矢印で示したかったのですが実装が面倒だったので見送りました。

全エージェントへの行動指示が終わるとTextWriter が txt ファイルに行動情報を書きだします。GUI も行動情報と同じく遷移するので 2 ターン目以降の定石も連続して書き込むことが出来ます。

盤面外に移動する処理の抑制などは記述する時間がなかったので実装していませんが、動かす必要のないエージェントはデフォルトで(-1,-1)座標を指定すると定石から外すことが出来ます。この場合の移動の方向は(0,0)で、これは適用されずに BeamSearch によって得られた結果に依って遷移します。

終了する際は、全エージェントへの行動指示が終わり、GUI のエージェント座標が更新されたタイミングで Esc を押すとプログラムから抜けることが出来ます。

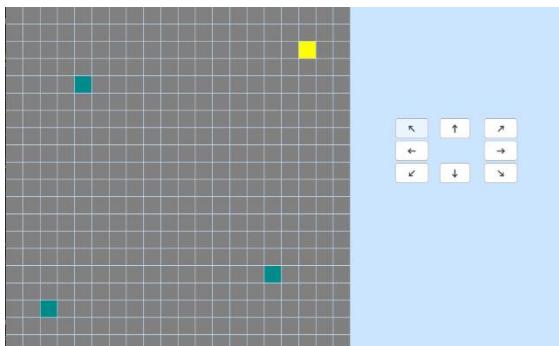


図1 GUI

5. 定石について

定石についてですが、全ての公開フィールドに対して作成したわけではありません。特に、エージェントの開始座標が密集しているフィールドなどではエージェント同士の衝突により定石通りに動いてくれない場合があります。なので、定石を書いていないフィールドもあります。

そのようなフィールドのときの解決方法として

は人力補助があります。人力である程度の指向性を持たせることで探索が上手いくので、埋め込みと人力補助を使い分けると効率のいい動きが出来たかなと思います。ウチは PC1 台しかないので無理でしたが。

6. おわりに

今回のこの定石作成ツールでは行動情報を吐き出すことしか出来ないため、仕様通りのファイルを吐き出すことができません。また、エージェント座標はコードに埋め込むことで対処しているため、危険なコードとなっています。それ以外にも、エージェントの対称移動を実装していかなかったりなど、もう少し良いものが作れたのではないかと心残りがあります。

ですが、人の手でこれを全てポチポチ入力していたら絶対にミスがあつただろうし、これを作ることで絶対にミスのない行動情報を得られるのなら安いのかなと思いました。

今年の高専プロコンは現地参加 2 回目だったのですが、昨年は初現地で全国優勝だったため、今年は初めての敗北でした。今思うと、GUI が見にくかったり、人力補助の工夫が出来ていなかったり至らないところがあったと感じています。来年は気を引き締めて頑張っていきたいと思います。

参考文献

OpenSiv3D

<https://github.com/Siv3D/OpenSiv3D>

高専プロコン競技部門のすゝめ

1. はじめに

高専プロコンの競技部門では、パズルや対戦型ゲームが主に課題として出題されます。そういう課題をプログラミングで解決しようとすると、さまざまな手法があります。

パズルであれば全探索、深さ・幅優先探索、動的計画法、機械学習、山登り探索及び焼きなまし法、遺伝的アルゴリズム、幅制限探索。

対戦型ゲームであれば、Min-Max 法、Alpha-Beta 法、モンテカルロ木探索、Q 学習、Grundy などがあります。上に挙げたのは特に有名なものです。もし、聞いたことがない名前があれば一度調べてみることをお勧めします。またここでお断りしておきますが、アルゴリズムの実装や動作原理については触れません。各自 wikipedia や過去のレーショナルで確認をお願いします。

この文章内では、競技部門に挑む上で必要とされる様々なアルゴリズムについて第 26 回から第 30 回までのテーマを例として説明します。

2. 問題解決のためのアルゴリズム選定

2.1. パズルゲームについて

現在状態から、列挙される状態が 10^{10} 以上の場合、全探索は不可能です。また、特殊な場合を除き深さ・幅優先探索も同様です。

解の厳密性を問わないならパターンに合わせてビームサーチ、焼きなまし法や Chokudai サーチを使うことでそれなりの解を導けると思われます。Chokudai サーチは、精度の高い評価が一定ターン後ではないとできない場合に特に向いています。

解の厳密性が重要なら、A*探索や動的計画法が適用できないか考えます。A*探索はヒューリスティックの精度によって探索時間が大きく変わります。

電気システム工学科 5年 鈴木 暢真

最後に機械学習について、説明を行います。機械学習は与えられたデータにフィットする関数を求めたり、データを分類したりするタスクに向いていますが、パズルなどの複雑なルールに対して適用することは難しいです。また学習に際して膨大なデータが必要になるので、使用する際は前もって十分に検討すべきです。

2.2. 対戦型ゲームについて

対戦型ゲームは、4 つの大きな分類があります。完全情報、不完全情報と逐次手番、同時手番です。逐次に動く場合、Min-Max 法、Alpha-Beta 法が使えます。同時に動く場合は、厳密に最適解を求めるならナッシュ均衡などを計算しなければいけません。しかし、計算量的に増大することが多く、難しいテーマなので逐次に動くと見なす、思い切って相手の動きを考えない、などの工夫が必要です。

3. 歴代の高専プロコンについての思索

3.1 第 26 回大会「石畳職人Z」

敷地と一定の順番でブロックが与えられるので、なるべく敷地を埋めよという課題です。

この年の優勝チームは、並列化されたビームサーチを使用していました。特に工夫されていた点は、ソルバー(解答プログラム)が解像度を変更でき、正確に答えを求められるモードと速さ重視で答えを求められるモードがあった点です。また強化学習によって求めた評価値を複数用いて、評価値 1 基準の上位 X 個、評価値 2 基準の上位 Y 個のように状態を保存することで、解の多様性を確保している点と状態が重複しないように hash を用いて重複除去を行っている点も素晴らしいかったです。

3.2 第 27 回大会「ホントの魅力がミエますか？」

形状の制限がないパズルピースを枠に入れる課題でした。また、ピースと枠の頂点データ等もチームごとに撮影するので形状を誤差なく読み込むことが非常に難しかったです。

順位が枠に入れたピース数で決まるため、コンピューターを使用せずに一番大きいピースを除きピースを枠に無理やり詰め込む戦略がほとんどのチームで取られました。

3.3 第 28 回大会「おいでませ、ホントの魅力へ」

この年の課題は、昨年のルールに加えて頂点データが QR コードで渡され、また頂点が 1cm × 1cm のグリッド上に存在するという制限が加えられました。順位付けは、QR コードを用いた数とパズルを完成させた時間により決まります。

優勝チームは、その他全てのチームが QR コードのデータを使用する中、QR コードで渡されるピースのデータを用いず、撮影を行い画像処理により極めて精度が高いピースの頂点データを求めて解答を行いました。アルゴリズムは、頂点の角度と辺の長さによる状態の列挙を行うビームサーチを用いました。

3.4 第 29 回大会 「巡りマス」

タイルに -16 から +16 までの点数がついたフィールドが QR コードで与えられ唯一フィールドを把握できる司令塔が、2 人のエージェントに対して指示を行う陣取りゲームが課題でした。

この年の優勝チームは我が校なのですが、基本方針はビームサーチでした。1 ターンの時間が 15 秒程度なので人間による指示伝達と移動方向の入力時間を考え、ビームサーチの探索時間が 2~3 秒になるように高速化を行いました。高速化は、Visual Studio の分析ツールを用いました。また評価値のパラメーター調整を手作業で行いました。さらに競技フィールドを校内に作り、本番の練習を行いました。本番環境に即した練習を行うことは、非常に重要です。

3.5 第 30 回大会 「踊って舞って回って」

この年の課題は、昨年のルールと違いサークルを介して対戦を行いました。また、エージェント数も最大 8 と増え、より計算量を減らすことを意識しなければなりませんでした。

優勝チームは、エージェントごとに独立で深さ 10、幅 30 の領域ポイントを考慮しないビームサーチを用いました[4]。

4. まとめ

ここまで、過去 5 年間の優勝チームのアルゴリズムを見てきました。毎年の優勝チームは、基本こそビームサーチですが工夫を凝らしていることが分かります。また注意しなければならないのは、優勝チームは様々なアルゴリズムを試作検討した上でビームサーチを選択したという点です。

5. おわりに

この文章を残すのは、私が入部した頃、競技部門についての知識の継承が部活内で一切行われておらず、苦労した記憶があるからです。参考文献に第 28 回から第 30 回までの我が校のソースコードのリンクを残します。末筆にはなりますが、仙台高専ソフトウェア研究部会の更なる躍進をお祈りしております。

参考文献

- [1] 仙台高専・名取、高専プロコン・競技部門
28 回大会のソースコード
<https://bitbucket.org/nasatame/procon2017ofnatori>
- [2] 仙台高専・名取、高専プロコン・競技部門
29 回大会のソースコード
https://bitbucket.org/nasatame/procon29_natori_open
- [3] 仙台高専・名取、高専プロコン・競技部門
30 回大会のソースコード
<https://github.com/YASAI03/KosenProcon30Natori>
- [4] 東京高専、高専プロコン・競技部門
30 回大会のソースコード
<https://github.com/tnct-spc/procon2019-comp>



プロコン旅行記

プロコン旅行記

総合工学科Ⅱ類 1年 日野 綾瀬

1. はじめに

今年の第30回全国高等専門学校プログラミングコンテストは、宮崎県都城市で10月13日、14日の2日間開催されました。プロコンを全く知らない人でも、楽しく読んでもらえるように頑張って書いたので、是非読んでください。

2. プロコン会場に着くまで

2.1. 仙台から博多へ

今回のプロコンは、不幸にも観測史上最大級の台風19号の上陸と被ってしまいました。それによる影響は大きく、飛行機の欠航や、JRが計画運休を発表するなど、台風19号はプロコンへ参加した高専生にとって非常に辛いものとなりました。もちろん私たちも例外ではなく、何とかして会場に着くためにも、本来12日に出発する予定を、急遽1日前倒して、11日に出発することにしました。

11時に仙台駅で集合し、みんなで昼食の駅弁を買って、新幹線で博多まで移動しました。約6時間と長い移動時間の中、先輩方はずっと開発をしていました。19時半頃に博多駅に着いて、そのままホテルにチェックインしました。そのホテルは先生が当日に予約したところだったので、正直あまり期待はしていませんでしたが、思っていたよりも部屋が豪華で驚きました。荷物を置いた後は、「博多一幸舎」というラーメン屋で先生にご馳走になりました。本場の博多ラーメン(図1)はこってり感が一味違っていて、「本物」の味がしました。とても美味しかったので、皆さんも博多に行く機会があったら、ぜひ博多ラーメンを食べてみてはいかがでしょうか。お店を出た後は、ホテルに戻って明日の用意をしてシャワーを浴び、すぐに寝ました。



図1 博多ラーメン

2.2. いざ会場へGO

6時半に起きました。私はよく二度寝をしてしまうので、念のためにアラームを5分おきにセットしておいたのですが、一度の目覚ましで完全に起きることができたので安心しました。朝食を食べてチェックアウトをして、新幹線で鹿児島中央駅へ移動しました。先輩方は昨日に引き続き開発をしていて、私はこの旅行記を書いていました。1時間半ほどで、鹿児島中央駅に着きました。鹿児島県に到着して、最初に出た私の感想は「暑い」でした。10月中旬にも関わらず、汗をかくほど日差しが強く、本当に宮城と同じ日本なのかと疑いました。昼食は駅ビルの中の「いちにいさん」というお店に行き、バラかつ定食を食べました。お肉がとても柔らかく、付いてきた豚汁も今まで食べたことないと思ったほど美味しいと印象に残りました。食べ終わったら、次の都城駅行の電車まで1時間ほど時間が空いていたため、駅ビルの最上階にあった観覧車に乗りました。風が強かったためか大きく揺れて怖かったのですが、窓からとても綺麗な桜島(図2)を見ることが出来たので、満足でした。



図2 観覧車から撮影した桜島

その後駅周辺の観光をして、15時16分発の電車に乗りました。都城行の電車は2両しかなく、乗客の多くがプロコン会場へと向かう高専生で終点の都城まで降りないため、私たちは1時間半もの間立ち続けることになりました。都城駅に17時頃に着いてから、受付をするためにすぐに会場(図3)へと向かいました。



図3 プロコン会場(都城市総合文化ホール)

会場に着いて受付を済ませたら、宿泊先であるアパホテルに移動して、チェックインを済めました。部屋に入った時に、電気の点け方が分からず戸惑いましたが、先輩に教えていただいたおかげで、無事点けられました。

夕食はパスタのチェーン店に行き、カルボナーラを食べました。トロトロしていておいしかったです。ホテルに帰った後、先輩方は明日の試合

に向けて深夜まで開発をしていました。私は貰ったパンフレットに一通り目を通した後、明日の準備をして寝ました。

3. ついにプロコン開催

3.1. 競技部門1日目

前日と同じく6時半に起きました。朝食はバイキング形式でした。宮崎名物のチキン南蛮がおいしかったです。8時にロビーで集合して、会場へと向かいました。会場では既に多くの高専生が開場を待っていて、張り詰めた空気を感じました。入場後は、競技会場である中ホールへ入り、参加者連絡会議を聴講しました。ステージ上には、長机と椅子がセットされていて(図4)、今日ここで試合をするのかと思うと、なんだか身が引き締まりました。



図4 競技会場の様子

参加者連絡会議が終わった後は、大ホールに移動し、開会式に参加しました。開会式には宮崎県知事も出席していて驚きました。開会式が終わって中ホールに戻った後、すぐに予行練習が始まりました。今回の競技部門は「踊って舞って回って」というタイトルで、内容は簡単に言えば陣取りゲームです。詳しい内容については競技部門の解説記事に記載されているので、そちらをご覧ください。予行演習の対戦相手は、石川、広島、そして東京でした。東京は、前回の第29回プロコンの決勝で争った相手で、まさか予行演習から戦うとは思ってもいませんでした。1試合目は、2勝1引き分けで1位と順調だったものの、2試合目はバグが生じてしまい、すべてのエージェントが動かず、0勝3引き分けで広島

商船と同率2位でした。最終結果としては東京が1位、仙台名取が2位でした。ファーストステージでは上位2チームまでがセカンドステージに進出できるため、順位だけ見れば決して悪くはない結果だったのですが、致命的なバグが予行演習で見つかってしまったため、先輩方は昼休みのお弁当を食べる時間を削って、デバッグ作業(バグの原因を発見、修正)をすることになりました。私と先生はその間にお弁当を食べました。唐揚げ、エビフライ、ハンバーグと、かなりボリュームのあるお弁当(図5)で、お腹いっぱいになりました。



図5 1日目に配布されたお弁当

昼休みが終わり、YouTubeでのライブ放送のトラブルで少し遅れましたが、いよいよファーストステージが始まりました。私たちの対戦相手は、津山、松江、そしてまたもや東京でした。予行演習で戦った相手とファーストステージでも戦うというのは、マッチング的にどうなのかと疑問に思いましたが、何か運命のようなものを感じました。1試合目は、東京が3勝、仙台名取が2勝1敗でした。2試合目も同じ結果で、最終結果は、東京が1位、仙台名取が2位で、無事セカンドステージ進出を決めることができました。この試合では通信部に不具合があり、プログラムを起動して最初のターンしか動かなかったため、プログラムを起動→落とすを15秒おきに30ターン×2セット繰り返すという荒業を先輩がしていました。大会1日目の予行演習とファーストス

テージを見ていると、バグが生じてエージェントが動いていないチームが結構多くあり、少し残念だと感じると同時に、プログラミングにバグは付き物なのだと思い知らされました。

3.2. 自由部門と学生交流会

1日目の競技が終了してから、学生交流会が始まるまで1時間半ほど時間があったので、自由部門の展示を見に行きました。個人的に気に入った作品は、熊本高専八代キャンパスさんのSilent Recorderという作品でした(図6)。

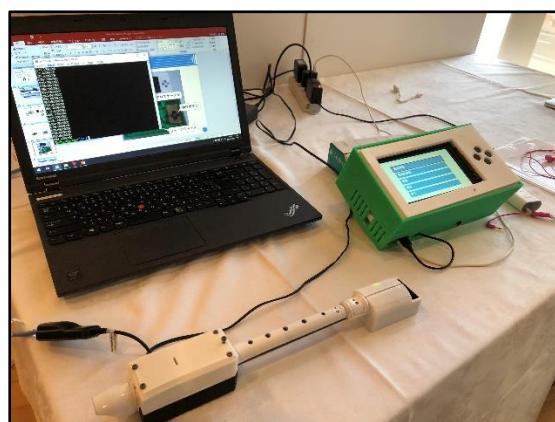


図6 Silent Recorderの写真

この作品は、リコーダー型演奏デバイスと専用デバイスを使用して、小学生のリコーダー演奏の練習を補助するための機器です。イヤホンを繋ぐことで、家でも騒音を気にせずに練習することができます。さらに正確性診断機能や演奏の記録機能、メトロノーム機能などがついていて、一人での練習をサポートしてくれると説明してもらいました。私は小学生のときリコーダーの演奏が苦手で、たまに家で練習してはうるさいと母に怒られていたため、実用化できたら便利そうだと思いました。また、私が詳しい仕組みや手入れの仕方を質問すると、担当の学生がひとつひとつ丁寧に答えてくださったので、なんだか嬉しくなりました。

いろいろな作品を見て回っていたらあっという間に時間が過ぎてしまい、学生交流会がはじまりました。内容は、宮崎に関するクイズ大会でし

た。途中まで全問正解していましたが、なぜか急に接続が切れてしまって、その後の問題が回答できなくなってしまったので、悔しかったです。

18時半に大会1日目の日程が全て終了しました。先輩方はコンビニで夕食を買って、ホテルでデバッグ作業をすると言うので、私もコンビニで適当におにぎりでも買って食べてようと考えていたのですが、先生からせっかくだし外に食べに行こうと誘っていただいたので、2人で「麺勝」といううどん屋に行きました。私は旅行中肉しか食べてなかつたので、魚が食べたくなり、小うどん付きの鯖の味噌煮定食を頼みました。中学校の給食で食べたあの味のままだったので懐かしい気持ちになりました。食べ終わってホテルに戻ってからは、明日の朝二度寝しないためにすぐに寝ました。

3.3. 競技部門2日目

6時15分に起きました。朝ごはんを食べて口べに集合し、会場へと向かいました。競技部門では朝から敗者復活戦とセカンドステージが行われました。セカンドステージの対戦相手は、北九州、大島商船、阿南でした。前日のバグが果たして直ったのかが心配でしたが、無事に動き、1戦目、2戦目共に3勝して、ファイナルステージへ進出を決めました。昼休みにお弁当を食べてしっかりと元気をつけて、私たちはファイナルステージへと臨みました。

ファイナルステージでは、これまでのような3チーム同時対戦ではなく、1対1で戦いました。準々決勝の対戦相手は、熊本高専熊本キャンパスでした。1試合目は、序盤からこちら側のエージェントが弱い動きをしてしまい、点差が大きく引き離されて負けてしまいました。2試合目も同じような展開になり、途中動かないエージェントも出てきて、大差で負けてしまいました。セカンドステージの試合で快勝したので、もっと上を目指せるのでは、と思っていた分悔しさは少しありましたが、多くの参加チームがいる中で、ベスト8まで行けたことが素直に嬉しかったです。

その後準決勝が行われ、決勝は東京と八戸、3位決定戦は久留米と熊本高専熊本キャンパスという組み合わせになりました。特に3位決定戦が熱い試合で、1ターンごとに順位が入れ替わっていく試合は、見てて手に汗握る戦いでした。最終結果としては、優勝が東京、準優勝が八戸、3位が久留米という結果になりました。最後の最後で最高の試合が見れたので、幸せでした。

3.4. そして閉会

競技部門が終わった後は、特別講演を聞きました。株式会社コロプラの代表取締役である馬場さんと、株式会社アラタナの代表取締役である濱渕さんが、テクノロジーの過去・現在・未来というテーマで話し合いました。これからの技術についての話はとても勉強になったのですが、どちらかというと、私はお二人の話がとても面白いことが印象に残りました。会場では笑いが常に起きていて、自分もユーモアセンスを磨きたいと思いました。

特別講演が終わると、そのまま閉会式が始まりました。そこでは各部門の表彰がありました。今年はなんと東京が競技部門で優勝、課題部門、自由部門で最優秀賞を獲得するという前例のない結果になりました。企業賞の授与では、最新のSurface Proを人数分といった豪華なものが贈られていました。閉会式が終わった後は、打ち上げを兼ねてみんなで「仏跳麺」というラーメン屋に行って、みそ仏跳麺を食べました（図7）。思ったよりもあっさりとしていて、食べやすかったです。



図7 仏跳麺で注文したみそ仏跳麺

ホテルに帰ってからは、先輩の部屋にみんなで集まって「コマンドー」と「シックス・デイ」をアパートにあった大画面テレビで鑑賞しました。どちらもアーノルド・シュワルツェネッガー主演の映画で、筋肉が凄かったです。

4. 神戸観光、そして宮城へ帰る

6時に起きました。ホテルをチェックアウトして、特急列車とバスに乗って、鹿児島空港へ移動しました。家族へのお土産選びに夢中で、手荷物検査の時間を想えていなかったせいで、検査のときに外したベルト片手に搭乗口へダッシュする羽目になりました。これからは時間に余裕を持って行動したいと思います。飛行機は、40分ぐらいで目的地の神戸に着きました。次の仙台行きの飛行機まで5時間ほどあったため、まず昼食をとるために、三宮の南京中華街に行きました(図8)。



図8 長安門

そこには、ラーメンや小籠包、北京ダックといった様々な中華料理のお店や屋台がありました。私たちは通りの端の方にあったラーメン屋に入りました。そこで私は、担々麺の小籠包セット(図9)を頼みました。小籠包は、一かじりした瞬間に肉汁がピューと噴き出してくるぐらい肉汁が詰まっていました。私は小籠包を中華街で食べることが小さいころからの夢だったので、今回の旅行でそれが叶って良かったです。次来た時は北京ダックを食べてみたいです。



図9 肉汁たっぷりの小籠包

お店を出てから、どこを観光するか話し合った結果、近くの動物園に行くことに決まりました。駅まで戻るときに通ったアーケードは、平日の昼間なのにも関わらず人が大勢いて賑わっていて、まだまだ仙台は田舎なのだと、ショックを受けました。駅のコインロッカーに買ったお土産を預けた私たちは、電車で10分ほどかけて神戸市立王子動物園に着きました(図10)。



図10 神戸市立王子動物園

事前に調べておいた情報に、この動物園にはパンダがいると思ったので、初めてのパンダにワクワクしながら入園しました。中にいた動物たちは、時間帯がお昼だったのが原因なのか、やる気がなく、のんびりしていました。お目当てのパンダも、最初は奥で寝ているだけでしたが、人が集まると歩き回り始めたので、さすが人気者なだけあるなど感心しました(図11)。



図11 ジャイアントパンダのタンタン(24歳)

その後も、水槽をぐるぐると回り続けていたアシカや、カメラを向けると近づいてきたとんでもない大きさのクマ(図12)など、個性的な動物が沢山いて、とても癒されました。



図12 クマ

一通り見て満足した私たちは、神戸空港に戻りました。鹿児島空港での失敗を反省して、搭乗案内が始まる20分前には手荷物検査を済ませて、中で待っていました。そして仙台行きの飛行機に乗り、無事全員が宮城県に帰ってくることができました。

解散して家に帰った後は、家族に写真を見せながら今回の旅行について話しました。5日間分を一気に話したので、つい長くなってしましました。その後は、ゆっくり湯船に浸かって、久々の家のベッドでぐっすりと眠りました。

5. おわりに

このプロコン旅行記を読んでいただきありがとうございました。自分自身Wordで約6000文字もの文章を書くということは初めての経験だったため、怪しい部分があったかもしれません、少しでも楽しかった旅行の様子が伝えられたなら幸いです。

今回のプロコンの感想ですが、一言で言うと「もっとプログラミングしたくなった！」です。私はプログラミング歴半年の初心者で、開発には参加しませんでした。そんな私がプロコンに参加して、それぞれのチームが自分たちで作り上げたプログラミングを信じて真剣勝負をしている姿を見て、感動しました。今は意味が理解できなくても、これからもっとプログラミングを勉強して、今度は自分が作ったプログラミングでプロコンに出場したいです。

今これを読んでいて、仙台高専名取キャンパスに入りたいと考えている人は、ぜひ私たちソフトウェア研究部会に入部して、一緒にプロコンに出場しましょう！

改めて、最後まで読んでいただきありがとうございました。

6. おまけ

スペースが余ったので、面白かったカンガルーの写真を載せておきます(図13)。

次は八木山動物公園に行きたいです。



図13 おっさんみたいなカンガルー

編集後記

猛暑も終わりを迎え、随分と涼しくなってきました。

さて、今年はプロコンと高専祭が被らなかったので全員で「Rationale」の配布に携われます。時間さえ合えばゲームや音楽の制作者に直接会えるので「Rationale」を見て疑問に思った部分や裏話等を聞いてみるのもいいかと思います。

また、今年も「Rationale」は web サイト上に公開されます。部員の作った音楽は web サイト上でしか聞くことができないので是非ご覧ください。

最後になりますが今回の「Rationale」制作にあたり、寄稿してくれた部員の皆様、校正をしてくださった先輩方、北島先生と佐藤先生、そしてこの「Rationale」を手に取って頂いた皆様、全ての方に感謝を申し上げます。ありがとうございました。

編集長 菅原 伶太

Rationale Volume.33

発行日 10月28日
発行団体 仙台高専(名取) ソフトウェア研究部会
印刷所 仙台高専(名取) 事務棟複写室

STAFF

代表	根地戸 龍生
編集	菅原 伶太
校正	高橋 謙大 鈴木 暁真 高橋 遥伽 菊池 輝 早坂 達海 原 龍平 渡邊 天翔 岩佐 佳慧
	根地戸 龍生
表紙	岩佐 佳慧
執筆	ソフトウェア研究部員
製本	ソフトウェア研究部員
プロコン旅行記	日野 綾瀬

**copyright S. R. D. G
Software Research and
Development Group**