RATIONALE

vol. 35

目次

顧問挨拶 01

代表挨拶 02

−プログラミングコンテスト−

競技部門

日野　綾瀬　機械・エネルギーコース３年　「第３２回プログラミングコンテスト」 04

−活動報告−

大高　康介　総合工学科　Ⅱ類　１年　「ブロック崩し」 07

鈴木　佑　総合工学科　Ⅱ類　１年　「じゃんけんゲーム」 09

眞島　駿宏　総合工学科　Ⅱ類　１年　「映画予告の作り方」 11

池田　悠介　総合工学科　Ⅲ類　１年「深さを入れると地球の構造がわかる
プログラミング」 13

稲垣　颯起　ロボティクスコース２年　「音声ファイルの文字起こしと返答」 15

佐藤　至　ロボティクスコース２年　「2D アクションゲームの作成」 17

渡辺　龍臣　ロボティクスコース２年　「3D モデル作成」 19

宮原　光敏　ロボティクスコース２年　「C++の練習」 21

鈴木　晴斗　ロボティクスコース３年　「Unreal Engine4 で楽してゲーム制作」 23

冨山　結都　ロボティクスコース３年　「音圧を上げる」 25

鈴木　雄裕　機械・エネルギーコース４年　「Fusion360 を用いた有限要素法」 27

編集後記 29

顧問挨拶

オンライン実施の良さと難しさ

ソフトウェア研究部会　顧問　北島　宏之

先日、第 32 回全国高等専門学校プログラミングコンテストが、秋田工業高等専門学校の主管でオンライン開催された。今回は、競技部門もオンライン開催となり、ソフ研からも出場の機会を得、現地に赴くことはできなかったものの、作戦の競技部門中止と比べとても嬉しく思えた。

昨年度から今年度にかけて、学校現場でもオンライン化が随分進んだ。同時双方向の遠隔授業をはじめ、学会との会議、学内の会議であっても密回避のため会議室に集合することなく開催されていることも多い。このことは、新型コロナ感染症予防の観点において、全てではないにせよ教室・会議室での密状態や登下校など移動時の人混みを回避できる点では望ましいことに疑いはない。他方、実験や実習など対面での実施が必要な授業では、対面であるが故に受講する際には幾分かの緊張感が必ずあり、集中力も求められ(強制され?)、遠隔よりも対面が合うといった学生の声も実際聞かれるところである。

遠隔と対面のどちらが良いかではなく、いずれにも良さと難しさがあることが見えてきた、ということかと思う。対面授業・会議という一昨年までは当たり前であった形態がコロナにより当たり前でなくなった時、遠隔を織り交ぜる必要から授業や会議自体の工夫が求められるようになった。ここで、遠隔側に着目すると、お互いカメラの向こう側で服装や姿勢など気にせず授業や会議をするなどは普通となり、また移動も不要であることから、授業・会議開始ギリギリにアクセスし、終了後はすぐに切断することで時間を効率よく利用できているように思える。加えて、スライドや資料開示も明確になり、チャットなどは複数手段により双方向の意見交換も対面時と比較して多い傾向があるようである。一方で、集中力を維持することは、対面時よりも実は容易ではないことも皆さん感じているように思う。特に、話し手の説明が単調であるなど内容や進行に時折の刺激が入れられていない場合には、視聴側はとてもツライこと経験済みではないだろうか。

さらに、遠隔時のコミュニケーションには、対面時の意識がいっそう強く求められることがあることにも注意が必要である。先日のプロコンで、ある部門の審査委員の発言が一部で問題に挙げられていた。私も確認したが、審査委員としてのコメントの内容は想定されるような指摘や意見であったが、その際の言葉遣いや、カメラ越しに見えた姿勢や服装、そこから伝わってくる雰囲気は、確かに必ずしも良いとは言えないように感じた。と同時に、もしも対面であったならば、少なくとも遠隔・オンライン時の悪い雰囲気よりかは幾分緩和されたように思えるし、司会者などが即座にフォローすることもできたように思われ、遠隔・オンラインの難しさを学んだ気がした。

“ポストコロナ”や”with コロナ”というキーワードがあるように、今後は遠隔(オンライン)と対面(オフライン)の両方を使い、状況により使い分けることが普通になると想像される。私もうまく使い分けていかねばと強く思った今回のプロコンであった。

代表挨拶

会長挨拶

機械・エネルギーコース　3年　日野　綾瀬

「Rationale」を読んでいただきありがとうございます。昨年は新型コロナウイルスの影響で、毎年参加していたプロコン競技部門の中止、ゲームの展示や「Rationale」の配布を行っていた高専祭の中止と、思うようにソフトウェア研究部会としての活動ができなかった一年でした。今年はプロコン競技部門のオンライン開催や、高専祭の実施など、少しずつですが例年のような活動ができるようになってきており、とても嬉しく思います。

今年のプロコンは、メンバー全員が開発未経験ということもあり、計画通りに開発が進みませんでした。それでも、メンバー全員が協力してコードを書き、本番で動いたプログラムを見たとき、初めてプロコンの面白さに触れることができました。プロコンについての記事は、私が締め切りに追われながらも精一杯書いたので、ぜひお楽しみください。

最後になりますが、この「Rationale」を書いてくれた部員の方々、顧問の北島先生と佐藤先生、今読んでいる皆様にこの場を借りて感謝申し上げます。

副会長挨拶

ロボティクスコース　2年　佐藤　至

先週まで猛暑が続いていると思っていたら、秋を通過して一気に寒くなりましたね。暖かい料理がおいしくなり、布団やこたつから離れられない人も多いのではないでしょうか。

さて、今年も高専祭の季節がやってきました。今回は体育祭と一緒に行うようで生徒会が忙しそうにしていたのを覚えています。そんな今回でも変わらずにソフトウェア研究部会は「Rationale」を作成してきました。今回の「Rationale」も部員が丹精を込めて作成した、一人一人の個性が輝くとても面白いものとなっております。

最後に、この「Rationale」を手に取ってくれた方に最大の感謝を示すとともに、副会長挨拶を終わらせたいと思います。

プログラミングコンテスト

第 32 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

機械・エネルギーコース　3年　日野　綾瀬
ロボティクスコース　3年　鈴木　晴斗
機械・エネルギーコース　4年　岡田　曜

1.　はじめに

私たちソフトウェア研究部会は、毎年全国高等専門学校プログラミングコンテスト(以下、プロコン)に参加しています。今年のプロコンは、新型コロナウイルスの影響で、去年に引き続きオンラインでの開催となりました。プロコンには課題部門、自由部門、競技部門の 3 部門があり、今年は競技部門に参加しました。

2.　競技概要

2.1.　競技内容

今年の競技部門のテーマは「技術廻戦」です。

競技内容は、1 枚の原画像から同サイズに切り分けられた断片画像をバラバラに並べて作られた問題画像を、元の画像に戻すパズルゲームです。原画像は問題画像から推測し、並び替えは隣り合う断片画像同士の交換だけで行わなければなりません。

この競技は 2014 年に岩手県一関市で開催された第 25 回大会でも実施されましたが、今年は上下、左右を飛び越えての交換が可能になったり、断片画像の回転が追加されたりと、復元がより困難になっています。その代わり、回答時間の要素がなくなり、制限時間をフルに活用できるようになったため、より最適な解を求めることが重要になっています。

2.1.　用語説明

原画像は、自然画やイラスト画などの 24 ビットカラー画像です。画像サイズは幅、高さともに最小で 32 ピクセル、最大で 2048 ピクセルです。問題によって異なります。

断片画像は、原画像を分割して作成した各々の画像です。分割数は縦、横ともに最小で 2 分割、最大で 16 分割です。断片画像のサイズは幅、高さともに最小で 16 ピクセル，最大で 256 ピクセルです。問題によって異なります。断片画像の幅と高さは同じで、すべての断片画像のサイズは同じです。断片画像の座標は 00 から FF までの 16 進数で表します。

問題画像は、断片画像を無作為に並べた画像です。断片画像は 90 度、180 度、270 度の度刻みで回転します。ただし，問題画像の座標 00 の断片画像は全体の向きの基準となるため、回転しません。

選択コストは、選択回数に選択コスト変換レートを掛けた値です。選択コスト変換レートは 1 以上 500 以下の整数で問題によって異なります。

交換コストは、交換回数に交換コスト変換レートを掛けた値です。交換コスト変換レートは 1 以上 100 以下の整数で問題によって異なります。

総コストは、選択コストと交換コストの合計の値です。

2.2.　問題フォーマット

問題画像はバイナリ PPM フォーマットです。問題ファイルのヘッダ部には、分割数、選択可能回数、選択コスト変換レートおよび交換コスト変換レート、ピクセル数、RGB の最大値の情報が記載されています。

2.3. 順位決定方法

勝敗判定は以下の優先順位で決定します。
①座標不一致断片画像数（原画像との座標が異なる断片画像が少ないチームが上位）
②向き不一致断片画像数（原画像との向きが異なる断片画像が少ないチームが上位）
③総コスト（総コストの少ないチームが上位）
④選択コスト（選択回数の少ないチームが上位）
⑤サイコロ（目の合計が多いチームが上位）

2.4. 制限時間

問題ごとに 8 分～20 分の回答の制限時間が決められています。

3. プログラムの説明

3.1. 画像の復元

問題画像の座標 00 の断片画像は回転しないことから、これを基準画像とします。他の断片画像の四辺について 1 ピクセルごとに RGB 値の差分を取ります。その合計値が最小である辺を持っていれば、隣接していると判断し、結合します。このとき、断片画像を回転して結合する場合は、その回転情報を記録しておきます。また、合計値が評価値を下回らなかった場合は、端に位置すると判断します。その後、結合した画像と隣接する画像を探索、結合します。これらの手順を繰り返し、原画像を復元します。

問題画像の例と、この画像復元プログラムを用いて復元した画像を図 1 と図 2 に示します。

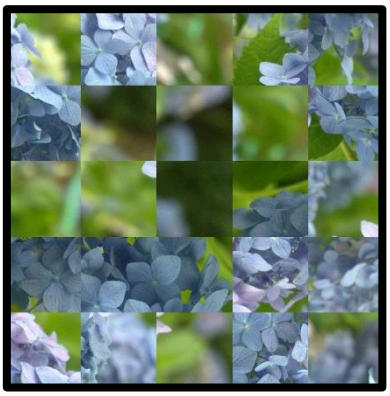

図 1　問題画像の例

図 2　図 1 の復元画像

3.2. 断片画像の並び替え

3.1.の画像復元プログラムによって復元した画像を元に、手動で並び替えるプログラムを採用しました。

4. 試合結果

1 回戦では 10 チーム中 5 位、敗者復活戦では回答の際にエラーが出てしまい、未提出という結果でした。

5. おわりに

今年のプロコンに参加したメンバーは、全員が開発未経験だったため、手探り状態で開発を進めていくこととなりました。そのような状況でも、本番でプログラムを動かして、他の高専と戦えたことが本当に嬉しかったです。しかしながら、画像復元の失敗や未完成の並び替えプログラム、敗者復活戦での小さなミスによるエラーなど、多くの反省点も見つかりました。これらを踏まえて、来年のプロコンでは 1 回戦を突破できるようアルゴリズムを中心にプログラミングの勉強に励んでいきたいと思います。

今年のプロコンの様子は YouTube Live で配信されていました。アーカイブが残っているので、興味のある方はぜひ視聴してみてください。

参考文献

第 32 回高専プロコン公式サイト
https://www.procon.gr.jp/

活動報告

ブロック崩し

Visual Studio 2019 による制作

総合工学科　Ⅱ類　1年　大高 康介

1.　はじめに

この文章では「Visual Studio 2019」という統合開発環境（IDE）を利用したプログラミングについて説明します。今回私は、このIDEを利用してブロック崩しを制作しました。使用した言語は C++言語で書かれたフレームワークの OpenSiv3D を活用しました。プログラムは、インターネット上にある「クリエイターのための C++ライブラリ-Siv3D」の「サンプル」のうち「ゲーム」に載っていたブロック崩しのプログラムを参考にして作りました。このゲームを制作した理由は、サイト上に載っていたプログラムが一度しかプレイすることができない仕様になっており、何度でもプレイできるようにしたいと考えたからです。

2.　OpenSiv3D について

OpenSiv3D は標準的な C++言語と書き方が似ており、ほとんどのプログラムを便利な型や関数を使って記述することができます。また、非常に短いコードでプログラムを書くことも可能です。その他には、ソフトの容量が小さいため、迅速にインストールができるという特徴があります。

3.　今回制作したプログラムの仕様

今回制作したプログラムは、もしボールがパドルよりも下の座標にあればゲームをリセットし、何回でもブロック崩しで遊ぶことができるようにさせています。ゲームのリセットを行う処理をリスト 1 に記載します。リセットするプログラムは、まずゲームを動作させるプログラムの中に gameover という変数を設定します。そして、もし gameover に true が代入されていれば、reset という名前の関数に移動させて、以降ループさせます。reset には、ゲームを動作させるプログラムと同じことが書かれています。

リスト 1　ゲームのリセットを行う処理

```
while (System::Update())
{
    bool gameover = false;

    if (ball.y >590)
    {
        gameover = true;
    }

    if (gameover)
    {
         reset();
    }
}
```

4.　プログラムの大まかな流れ

4.1.　ブロックやボールなどの表示

ブロックはグラデーションカラーにしました。また、ボールとパドルは、背景の黒と対をなす白にしました。フォントはこの中で使われてない色の水色にしました。

4.2.　ボールの挙動

スピードを-1 にすると進む方向が逆になるので、ボールがパドルや壁、ブロックに当たったら、ボールのスピードを-1 にして跳ね返します。ブロックに関しては、当たったブロックを消した後にスピードを-1 にして跳ね返します。

4.3.　ゲームのリセット

ボールのy座標が 590 を超えたら、ブロックの配列とボールの位置を初期化します。

このプログラムの大まかな流れを、フローチャートにして図 1 に示します。

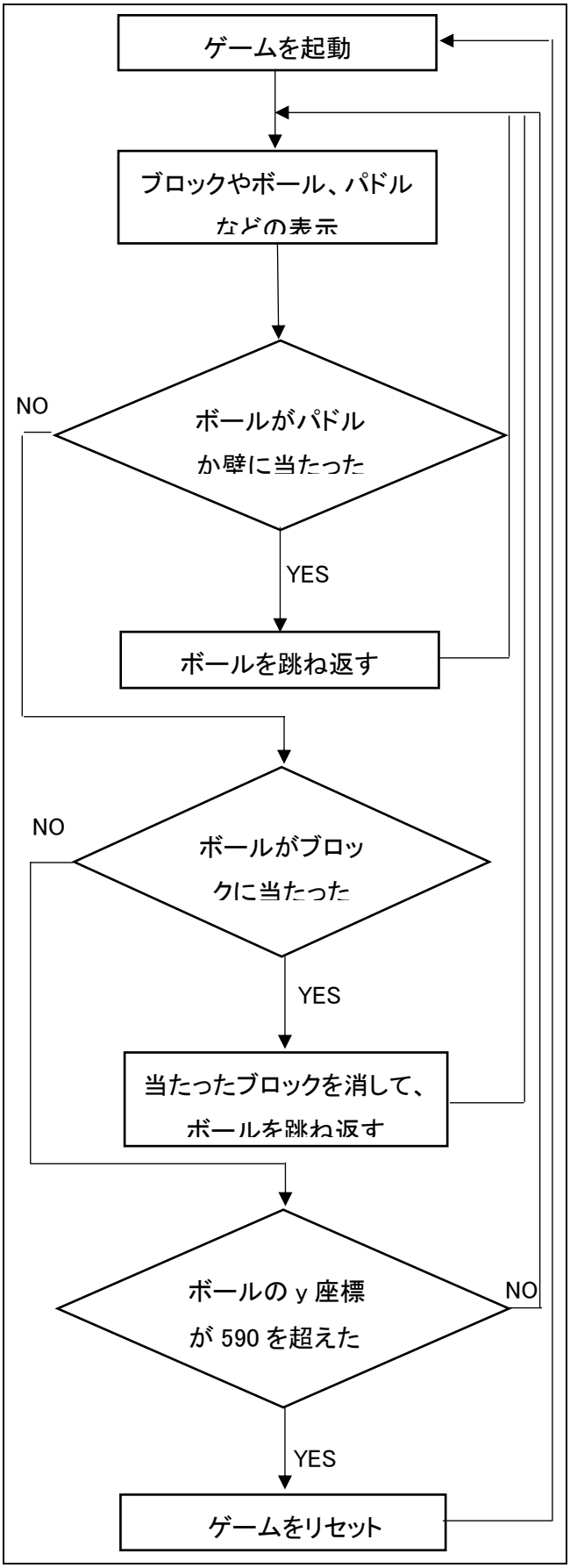

図 1　ゲームのフローチャート

5.　おわりに

今回私は、初めて自分でプログラミングをしてゲームを作りました。プログラミング「言語」というだけあって、学び始めたときは異国の言語のように何を言っているのか全く分かりませんでした。しかし、今ではほんの少しだけ分かるようになってきて、自分の成長を感じました。ただ、今でも何をしているのかはあまり分からないので、もっと勉強していきたいと思いました。

私が今回制作したプログラムは、単純なものなのですが、それだけでも私にとってはとても難しく苦労する課題でした。次にまたゲームを作るときは、もっといろいろな知識を増やして、さらに出来のよいゲームを制作したいです。

最後に、今回制作したゲームの画像を以下に示します。

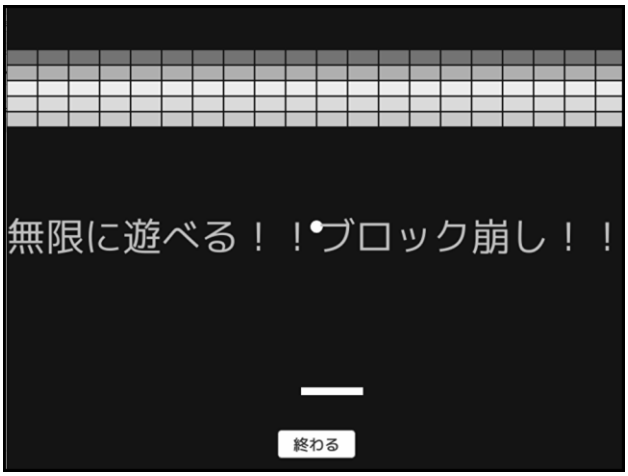

図 2　ゲーム画面

参考文献

「クリエイターのための C++ライブラリ-Siv3D」
https://siv3d.github.io/ja-jp/

じゃんけんゲーム

総合工学科　Ⅱ類　1 年　鈴木　佑

1. はじめに

私は今回 OpenSiv3D を使いじゃんけんゲームを制作しました。ゲームを制作するのは初めてで、手間取った部分もありましたが無事完成することができました。これから私が制作したゲームについての解説をします。

2. ゲームの動作とプログラム

2.1. ゲームの流れ

このじゃんけんゲームはまずプレイヤーがグー、チョキ、パー、の 3 つの中から選び、その後にランダムで相手側が選択し、勝敗を決めます。

2.2. じゃんけんの勝敗の決め方

図 1 のようにプレイヤーがグー、チョキ、パーの中から 1 つ選びます。その後、ランダムで相手(プログラム)側も同じように 3 つの中から選び、勝敗の結果が図 2 のように表示されます。ゲームの勝敗判定と結果を表示するプログラムをリスト 1 に示します。

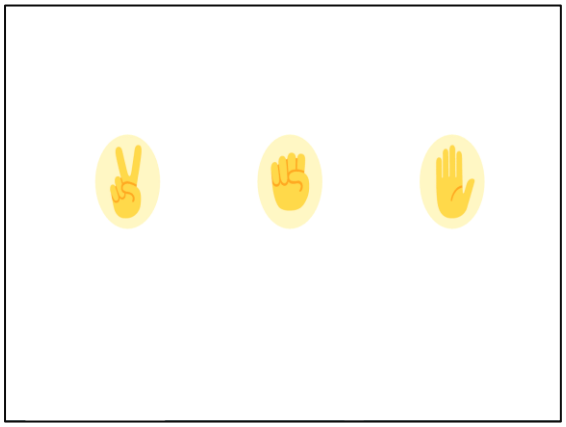

図 1　ゲーム画面

図 2　結果の表示

リスト 1　勝敗と結果を表示するプログラム

```
if (chokiCircle.mouseOver()){
Cursor::RequestStyle(CursorStyle::Hand);}
if (guCircle.mouseOver()){
Cursor::RequestStyle(CursorStyle::Hand);
}
if (parCircle.mouseOver()){
Cursor::RequestStyle(CursorStyle::Hand);
}
if (chokiCircle.leftClicked()){
text = Sample({ U"✊ 負け",U"✌ あいこ",U"✋ 勝ち" });
}
if (guCircle.leftClicked()){
text = Sample({ U"✊ あいこ",U"✌ 勝ち",U"✋ 負け" });
}
if (parCircle.leftClicked()){
text = Sample({ U"✊ 勝ち",U"✌ 負け",U"✋ あいこ" });
}
```

2.3. リストの解説

このプログラムは if 文の条件式でまずグー、チョキ、パーのどれかをクリックしているか判断をします。クリックをしたのなら、ランダムでグー、チョキ、パーを表示します。その後、勝利しているなら“勝ち”、敗北しているなら“負け”、あいこなら“あいこ”と表示する仕組みになっています。

2.4. プログラムの大まかな流れ

このゲームの大まかな流れを図 3 に示します。

3. 終わりに

私は今回初めて自分でゲームを制作し、こうした活動報告を書かせていただきましたが、改めて自分の実力や知識の無さを感じさせられました。本来はシーンや、関数なども追加したかったのですが、締め切りもあり、今回は追加しないことを決めました。来年はいろいろな技法を学び、さらに複雑なゲームを制作したいと考えています。

最後まで読んでいただき、ありがとうございました。

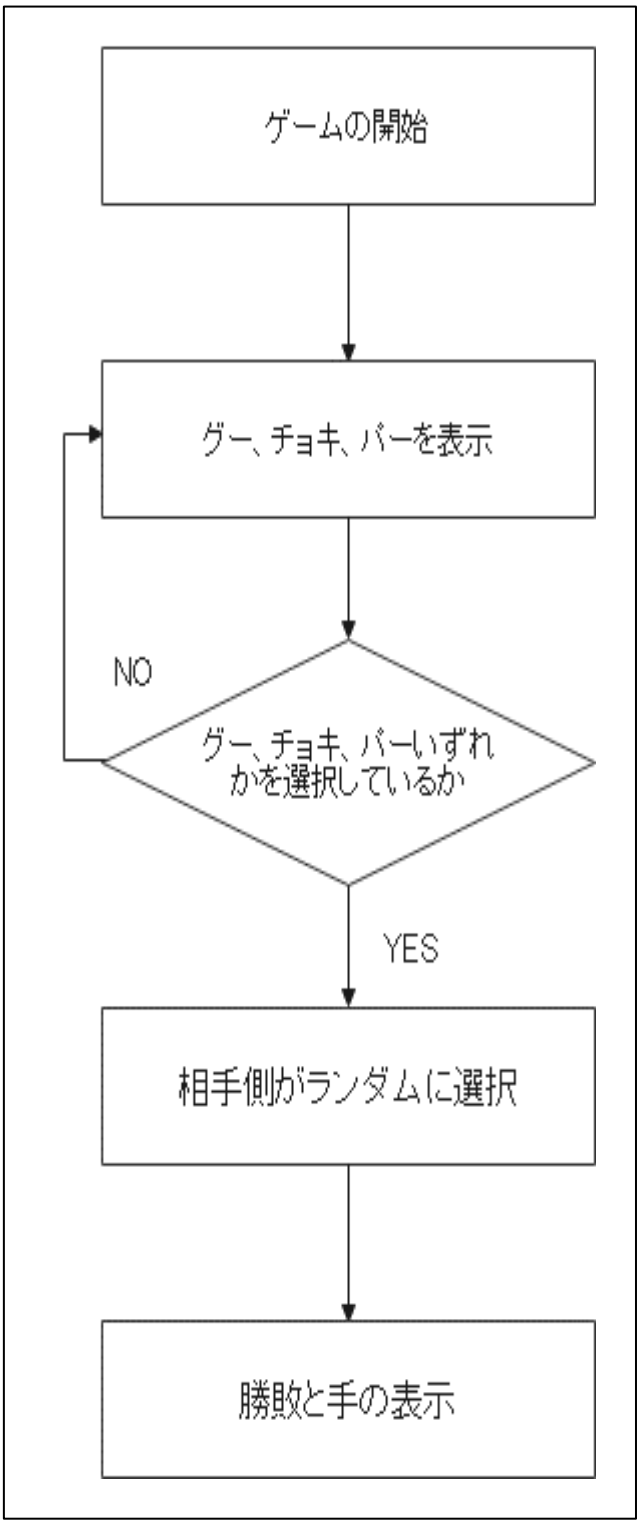

図 3 ゲームのフローチャート

参考文献

OpenSiv3D
https://siv3d.github.io/ja-jp/

Siv3D リファレンス v0.6.2
https://zenn.dev/reputeless/books/siv3d-documentation

映画予告の作り方

～ありがちなくっさい青春映画編～

総合工学科 Ⅱ類　1年　眞嶋　駿宏

1.　はじめに

私はそれぞれ違う高校に行った中学の同級生４人と、ありがちな B 級青春映画の予告編を作りました。撮影は私の iPhone12Pro Max を、編集は Final Cut Pro X を使用しました。今回は、この動画の発案から完成までを説明しようと思います。

2.　Final Cut Pro X について

Final Cut Pro X は、動画にタイトルやテロップ、サウンドなどをプラスできる Mac ユーザー向けの動画編集ソフトです。macOS のみに対応していて、Mac に標準で搭載されている iMovie の高機能版とも言われています。

3.　テーマ、設定、構成決め

予告全体の内容を決めます。ここで大事なのは、「ありがちな」映画予告を作るということです。そのため、さまざまな「あるある」を詰め込む形になります。

3.1.　テーマ

まずは映画予告テーマを決めます。今回は CG など難しいことはできないので、特に特別な編集が必要ない部活がテーマの青春映画にしました。

3.2.　設定

内容の詳しい設定を決めます。登場人物は村上、佐藤、眞嶋、ゴーイングメリー・豪、平地の４人で、「眞嶋が起こした事故の示談金を払うために『ディクシオン』という競技の大会に優勝する。」というのがざっくりとしたストーリーです。

3.3.　構成

最後に動画の構成を決めます。図１は全体の構成を表したものになります。

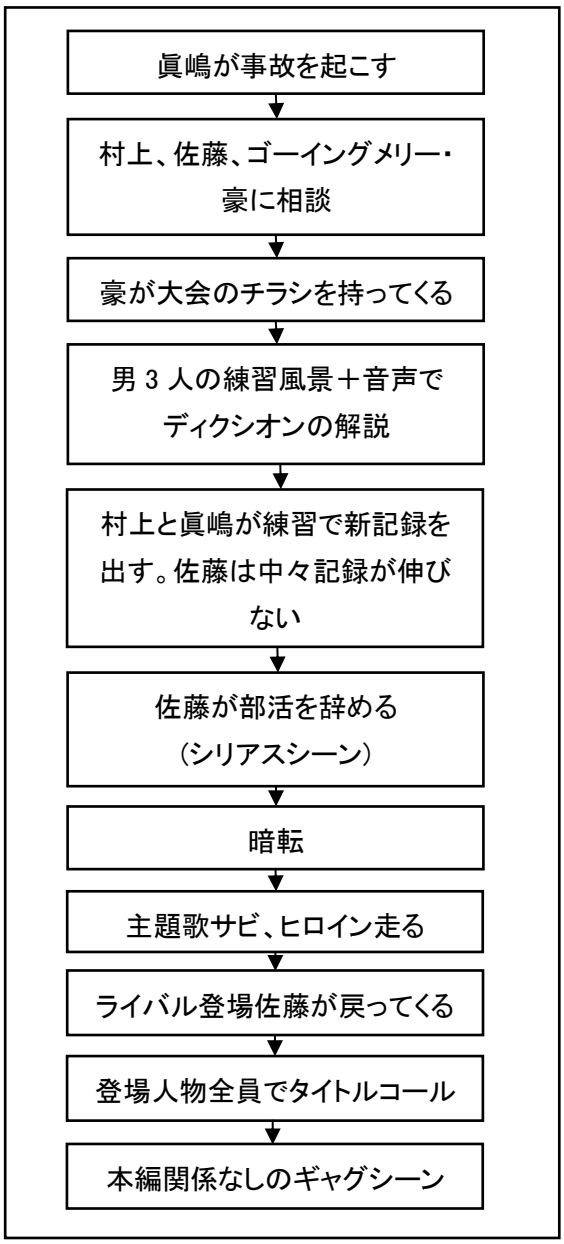

図 1 動画の構成

4. 撮影

いよいよ撮影です。良い動画を作るためにはぶっつけ本番ではなく、事前に何をどう撮るかを決めた計画的な撮影が欠かせません。そのため撮影の際には事前に「いつ」「どこで」「誰と」「どのシーンを」「どれくらいの長さ」撮るかを決めておくことを強くお勧めします。それらに追加して、絵コンテなどがあるととてもスムーズに撮影が進むのでお勧めです。

4.1. 道具準備

撮影で使う道具などを決めます。今回のテーマは部活の青春ものなので登場人物はみんな制服にしたいと思ったのですが、高校によって制服を SNS にアップしてはいけない校則があったり、そもそも高専生は制服を持っていなかったりと色々問題があったので全員「ワイシャツと黒地のズボン」で統一しました。あとはやる競技で辞書を使うので1人一冊以上国語辞典か英和辞典、最後の切り札として電子辞書も持っていきました。あとは本編に関係ない扇子や防弾マスク、ゴム製ハンマーなどを持って行きました。小道具はあればあるほど可能な表現が増えるのでできる限りたくさん持って行きましょう。

5. 編集

文字通り撮った動画データを Final Cut Pro X で編集します。といっても特に特別な編集は必要ないので、画面比を変えたり、指定された秒数に収まるようにカットしたり気になるところを直せば完成です。

6. 終わりに

8月ごろ友人に「1回でいいから丸1日使ってガチの動画作らないか」という提案を受け、今回の企画に至りました。撮影場所もお金を払ってレンタルするのも初めての体験でした。実はまだ細かい修正点があったり本来のナレーション役の人が収録に来られなかったりと色々あって中途半端な出来になってしまって、そこだけが心残りです。ここで紹介した作り方に基づいて私たちが作った動画は、下の QR コードを読み取ると試聴するとができます（図2）。顔が写ってないことを条件にメンバーの写真を貼る許可が出たので、載せておきます（図3）。やはり、仲のいい友達と協力して何かを作り上げるというのは何にも代え難い楽しさがあります。皆様もぜひ、周りの人たちを巻き込んでこのような動画を作ってみてはいかがでしょうか。きっと何年間も心に残る、いい思い出になると思います。

図 2 動画が見られる QR コード

※SNS へのアップロードはご遠慮ください

図 3 平地以外のメンバーの写真

深さを入れると地球の構造がわかるプログラミング

～普通に調べたほうが早いんだからね～

総合工学科　Ⅲ類　1年　池田　悠介

1.　はじめに

私は地球の深さを入力するとその深さの組成構造がわかるプログラムを作りました。これを使う場面というのは想定していないのですが、地層に興味があったので作りました。調べてみると地球は大きく5つの構造で分かれていることがわかりました。上から順に、地殻、上部マントル、下部マントル、外核、内核です。遷移層やD“層というのもあったのですが細かい部分だったため省きました。おもしろいと思ったのは、地球の半径は約6370キロといわれていますが、人類が到達している最深地点は12キロであり、全体でみると浅いということです。

2.　プログラムの説明

リスト1がプログラムの全体です。長いですが書いている内容はシンプルなので細かく説明していきます。

1行目は iostream、2行目は string というファイルを読み込むものです。ファイルにはそれぞれ宣言が含まれています。

3行目では28行目から0という数値を受け取り、プログラムを正常に終了させています。

6行目ではaという変数に、入力した数値を代入させています。

8～27行目では a に代入された数値が特定の数値の範囲に当てはまったときにそれぞれ地球の組成構造の名前が出るようになっています。

28行目では、0という数値を3行目に返してプログラムが正常に終了したことを知らせるためのものです。

図1にはプログラムの動きの流れが載ってあります。初めに文を出力し、次にaに代入させた数値が特定の範囲に当てはまったときにそれぞれ地球の組成構造の名前を出力させています。

リスト1　完成したプログラム

```
1.  #include <iostream>
2.  #include <string>
3.  int main(void) {
4.  std::cout << "深さを入れて下さい" <<std::endl;
5.  std::cout << "単位はキロです" << std::endl;
6.        int a;
7.        std::cin >> a;
8.        if (0 <= a < 5) {
9.        std::cout << "地殻" << std::endl;
10.       }
11.       else if (5 <= a < 60) {
12.       std::cout << "上部マントル" << std::endl;
13.       }
14.       else if (60 <= a < 2900) {
15.       std::cout << "下部マントル" << std::endl;
16.       }
17.       else if (2900 <= a < 5100) {
18.       std::cout << "外核" << std::endl;
19.       }
20.       else if (5100 <= a <= 6370) {
21.       std::cout << "内核" << std::endl;
22.       }
23.       else if (a < 0) {
24.       std::cout << "地上" << std::endl;
25.       }
26.       else if (a > 6370) {
27.       std::cout << "裏側" << std::endl;}
28.       return 0;
29.    }
```

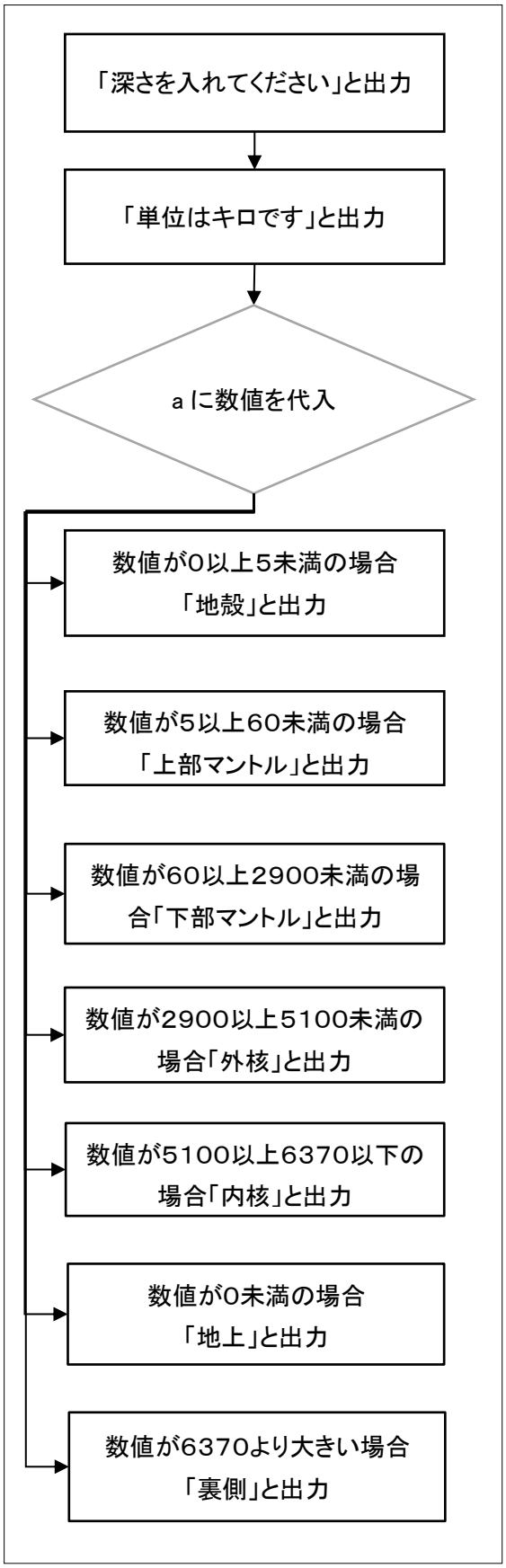

図1　主な流れ

3.　おわりに

今回作ってみたときに知ったのは、地球は球体になっているため地球の中心である6370キロを通過しても内核、外核、下部マントル、上部マントル、地殻という順で存在しているということです。今回は裏側の組成構造も出せるようにするとコードが長くなると思ったのでしませんでしたが、短くする方法があればそういうこともできると思いました。もっとプログラミングの勉強をして、実用的なものが作れるようになろうと思います。

ここまで読んでいただき、ありがとうございました。

参考文献

地球の構造−地球調査総合センター
https://www.gsj.jp/geology/geology-japan/geology-earth/index.html

音声ファイルの文字起こしと返答

～Python による if 文 と SpeechRecognition を組み合わせる～

ロボティクスコース　2年　稲垣　颯起

1.　はじめに

この文書は GoogleColaboratory というサイトを用い、そのサイト上でプログラムを書き、実行させた手順とその結果についてのレポートです。このプログラムは、wav 形式の音声ファイルを SpeechRecognition という無料の API を用いて文章化し、その文章が特定の内容であればそれに対する返答をするというものです。

2.　実験手順

GoogleColaboratory というサイトを用いてそのサイト上でプログラムを書き、実行します。その後、音声データを入力し SpeechRecognition を用いて文字起こしをします。最後に文字起こしの内容を if 文に移し、狙った答えをプログラムに出力させます。

3.　音声ファイルとプログラム文

今回はプログラムの都合で録音した mp3 ファイルを wav ファイルに変換しています。変換は「Convertio」というサイトのファイルコンバーターを利用して音声ファイルの形式を変更しています。また、音声ファイルの文字起こしの部分や返答に関する部分などで GoogleColaboratory のセルでプログラムを分けています。

4.　プログラムの本文

プログラムについて説明します。プログラム 1 では GoogleColaboratory に SpeechRecognition をインストールしています。プログラム 2 では、SpeechRecognition を使い音声ファイルの文字起こしを行っています。

リスト 1　プログラム 1

```
pip install SpeechRecognition
→Collecting SpeechRecognition
                Downloading
https://files.pythonhosted.org/packages/26
/e1/7f5678cd94ec1234269d23756dbdaa4c8
cfaed973412f88ae8adf7893a50/SpeechRec
ognition-3.8.1-py2.py3-none-any.whl
(32.8MB)

     | 32.8MB 125kB/s
Installing collected packages:
SpeechRecognition
Successfully installed SpeechRecognition-
3.8.1
```

リスト 2　プログラム 2

```
import speech_recogniti
on as sr

AUDIO_FILE = '音声ファイル.wav'
 #ここを変更。アップロードした音声ファイル（.wav 形式）名に変更してください。

r = sr.Recognizer()
with sr.AudioFile(AUDIO_FILE) as source:
 audio = r.record(source)

print('音声データの文字起こし結果：¥n¥n',
r.recognize_google(audio, language='ja'))
```

このプログラムの「AUDIO_FILE = ' '」のコロンの間に任意の音声ファイルを入力すると文字起こしをしてくれます。また、このときに入力するファイルは Google ドライブなどからクラウド上の GoogleColaboratory のフォルダに保存しておく必要があります。プログラム 3 では 2 の結果を A と置き換え、A に「こんにちは」が含まれていたら"こんにちは"と出力し、そうでないなら"用意されてないよ"と出力します。

プログラム 4 も 3 と同様に A に対する返答を行います。4 では A に「こんばんは」「生麦」のどちらかもしくは両方が含まれているとき"生麦おじさん"と出力し、そうでないなら"用意されてないよ"と出力します。

5. 結果

音声は人が聞き取れるくらいの音声は問題なく文字起こし出来ました。次が音声に対する返答の結果です。

表 1 実行結果

1.やあ→プログラム 3：用意されてないよ
　　プログラム 4：用意されてないよ
2.こんにちは　生麦
　　→プログラム 3：こんにちは
　　プログラム 4：生麦
3.こんばんは
　　→プログラム 3：用意されてないよ

6. おわりに

今回のプログラムの文字起こしの部分はほとんどインターネットで見つけた部分を参考にしました。プログラムの 3 と 4 は if 文という決まりに基づいて簡単に作っています。日本語の部分を書き換えたり、増やしたりするとより高度な返答プログラムを簡単に作ることができます。GoogleColaboratory は無料のサイトなので皆さんも試してみてはいかがでしょうか。

リスト 3 プログラム 3

```
A = r.recognize_google(audio, language='ja')
 #A = "生麦おじさん"
if "こんにちは" in A:
    print("こんにちは")
else:
   if A is not "こんにちは":
    print("用意されてないよ")
```

リスト 4 プログラム 4

```
if "こんばんは" in A or "生麦" in A:
 print ("生麦")
else:
 print ("用意されてないよ")
```

参考文献

子供プログラマー～おとなのためのプログラム・コンピュータ学習支援サイト～
https://child-programmer.com

電脳産物
https://dianxnao.com

Colaboratory へようこそ
https://colab.research.google.com

Python プログラミングパーフェクトマスター
Python/Anaconda/PyQt 対応第3版
金城 俊哉 著

2D アクションゲームの作成

ロボティクスコース　2年　佐藤　至

1.　はじめに

今回私は、Unity を利用して 2D アクションゲームを制作した。2D アクションゲームとは水平、垂直移動を主としたアクションゲームの一つである。ここではその大まかな仕組みについて解説していく。

2.　プレイヤー側のプログラム

プレイヤーのプログラムには、大きく 2 つの要素がある。当たり判定についてのプログラムと、行動するプログラムである。まず当たり判定のプログラムから解説していく。

2.1.　当たり判定

当たり判定の作成には Unity のタグという機能を利用している。タグ機能とは、操作キャラや敵キャラなどのオブジェクトをグループ分けする機能である。私は操作キャラに Player、敵キャラに Enemy というタグをつけている。この機能を利用するとプログラムを整理することができる。表 1 を見てほしい。これはプレイヤーのプログラム内の敵キャラとの接触判定についての項目である。プログラムの内容は Enemy タグを持つオブジェクトに触れるとプレイヤーがダメージを受けるというものである。このプログラムを導入することで、新しい敵キャラを作る際に Enemy タグをつけておくだけで敵キャラのプログラムを簡略化することができる。

表1　接触判定

```
OnCollisionEnter2D(Collision2D collision) {
    if (collision.collider.tag == enemyTag)
        Debug.Log("敵と接触した！");
}
```

2.2.　移動するプログラム

行動に関するプログラムには縦方向と横方向の 2 種類がある、まず横移動のプログラムから解説していく。まず横移動のプログラムについて解説していく。表 2 を見てほしい。プログラムの中に horizontalKey という文字がよく出てきているが、これは Unity の機能の一つで、ボタンを指定する必要がなくなるため、あとからの調整が楽になるのだ。ちなみにこの時移動を正負で考える。今回はその性質を利用して、if 文を利用して horizontalKey が正に入力されると右へ、負にでは左の方向に進む。最後の 1 行は移動のスピードの変異についてのプログラムである。dashTime という変数を参照することによって、徐々にスピードが速くなるといった、細かい移動の設定が可能になるプログラムである。またこのプログラムでは、speed という変数に好きな数値を入れることで、最高速度を調整することができる。

表 2　横移動のプログラム

```
 private float GetXSpeed()
  {
floathorizontalKey
                = Input.GetAxis("Horizontal");

  float xSpeed = 0.0f;

  if (horizontalKey > 0)
 {
transform.localScale = new Vector3(1, 1, 1);
  dashTime += Time.deltaTime;
  xSpeed = speed;
```

```
 }
else if (horizontalKey < 0)
 {
transform.localScale = new Vector3(-1, 1, 1);
    dashTime += Time.deltaTime;
    xSpeed = -speed;
  }
  else
   {
    xSpeed = 0.0f;
    dashTime = 0.0f;
  }
if (horizontalKey > 0 && beforeKey < 0)
{
    dashTime = 0.0f;
}
else if (horizontalKey < 0 && beforeKey > 0)
{
    dashTime = 0.0f;
}
beforeKey = horizontalKey;
xSpeed *= dashCurve.Evaluate(dashTime);
beforeKey = horizontalKey;
return xSpeed;
}
```

次に縦方向についてのプログラムである。表 3 を見てほしい。地面に Ground というタグを導入することで地面に触れているときのみにジャンプができるようにしている。またプログラム内の verticalKey というのが、縦方向の変数である。

表 3　ジャンプのプログラム

```
private float GetYSpeed()
{
float verticalKey = Input.GetAxis("Vertical");
float ySpeed = -gravity;
if (isGround)
{
if (verticalKey > 0)
{
   ySpeed = jumpSpeed;
   jumpPos = transform.position.y;
 }
 }
```

4.　最後に

ここまで Unity での 2D アクションゲームの仕組みの解説を行ってきたが、複雑で難しい印象を受けたと思う。だが安心してほしい。Unity はゲームエンジンの中でも扱いやすく、直感的な操作がしやすいうえ、インターネットに情報が多いので、初心者向けのゲームエンジンと言える。少しでもプログラミングを始めてみたい方は Unity を利用してみるのはいかがだろうか。

参考文献

Unity

https://unity.com/ja

ゲームの作り方

https://dkrevel.com/

ゲームの作り方講座「Unity でゲームを作ろう！」へようこそ！

https://unity.moon-bear.com/

3D モデル作成

自分の V の皮をつくろう！

ロボティクスコース　2年　渡辺　龍臣

1.　はじめに

今回は無料のツールで 3D モデルの作成、またそのアバターを外部のゲームで読み込む過程を紹介します。

理由としては最近の流行である Vtuber や、ゲームのキャラクターのモデルなど様々な場所で使われる 3D モデルについて自分なりの方法ですが大まかにまとめておこうと思ったためです。

勉強し始めて日が浅いですがどうぞお付き合いください。

2.　記事の紹介

2.1.　制作に用いたソフト

3D モデルの作成−「VRoid Studio」

3D モデルの編集−「Blender」

2.2.　VRoid Studio について

VRoid Studio では VRM 形式で 3D モデルの制作ができます。素体はあらかじめ作られていて髪や服装、体型の設定をすることによって自分の好きなキャラクターを作ることができます。

3D モデルの要であるボーンの設定もソフト側が処理してくれるため、VRoid Studio で完成させた時点で互換性のある外部のソフトで使用できる状態になっています。

3.　3D モデリングの開始

3.1.　3D モデルの作成

まずは VRoid Studio をダウンロードしましょう。VRoid Studio は Windows 64bit 版、macOS、そして Steam からもダウンロードが可能です。

インストールが終わったら「あなたのモデル」から新規作成を選択します。

また、この時サンプルモデルから髪型のプリセットや体形のサンプル等を使用することができます。権利上の問題もないので公のコンテンツでもサンプルをそのまま使用可能です。

3.2.　髪型のモデリング

髪型のモデリングは始めにパスを通すことから始まります。網目状の髪を描画するためのパスを配置、調整しその上をマウスのドラックでなぞることで髪のモデリングが可能になります。

髪型のモデリングにはいくつものレイヤーを使うことができ前髪用のパス、後ろ髪用のパスなどを分けて使うことができます(図 1)。

また髪の太さ、毛先にかけての太くなり方やハイライトの位置、テクスチャまで高度な設定が可能です。

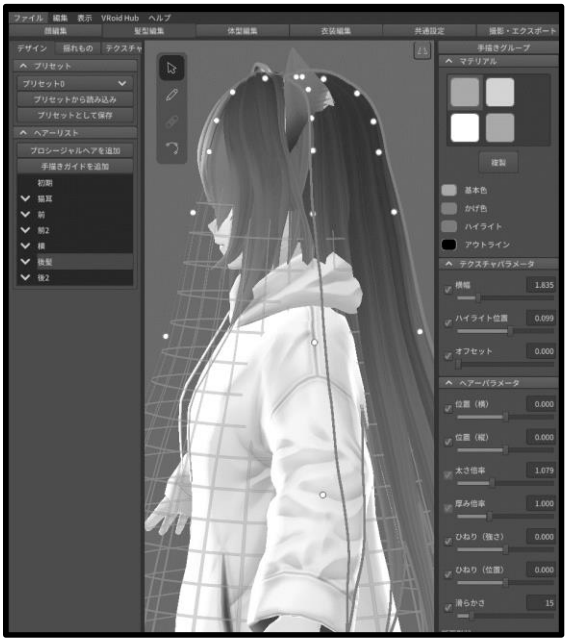

図 1　髪型の編集画面

3.3. その他の設定

3D モデルの個性ともいえる髪のモデリングが終われば残りの工程はあと少しです。

ここからは自分がどれくらい手をかけるかによって作業量が変わります。主にテクスチャの設定、体型設定、ボーンの設定です。

まずはテクスチャの設定ですが VRoid Studio では基本のテクスチャがしっかりと作られていて、こだわりがないならそのままのテクスチャで問題ありません。しかしながら VRoid Studio はオープンソース、Google で検索するだけで様々なツールが出てきます。自分は目のテクスチャを作るツールなどを借りて使用しています(＊1)。

髪やアイライン、まつげや皮膚などのテクスチャも変更することができます。

次は体型の設定です。名前のままですが体型の変更や細かな設定が可能です。

ボーンの設定ですがこれは非常に難しい設定で知識がないと難しいです。

ボーンは 3D モデルを動かすときの骨であり支えになります。この設定次第では髪が体にめり込む、重力の影響を受けて激しいモーションについて来ることができない等様々な問題があります。

自動生成の機能が備わっていますがうまくいかないことがほとんどです。満足のいく形に行かないときはボーンを入れないという選択肢もありです。

ここまでの操作を経てモデリングは完成です。服は自分の好みで着せてください。

4. ゲームでの読み込み方

4.1. 読み込むゲームについて

今回は Craftopia というサンドボックス系のゲームを使っていきます。Steam で購入できるゲームで VRM ファイルが読み込める都合上先ほど作ったモデルを含め好きな 3D モデルを使うことができます。

4.2. 読み込む方法

読み込むときの下準備として、5.2.で紹介する VRoid Hub にキャラをアップロードする必要があります。

Craftopia をダウンロード後、新規キャラデータを作成します。キャラクター選択時に VRoid Hub から VRM が選択できるようになっているのでアップロードしたモデルを読み込みましょう。

5. 外部での使用方法

5.1. Blender

Blender を使うことで、作ったモデルを編集したり、3D 作品に使ったり、モーションを入れることによって MMD の制作などができます。

しかし Blender で VRM ファイルを読み込むためには、有志が制作したアドオンをダウンロードする必要があるので注意してください。

5.2. VRoid Hub

VRoid Hub ではみんなが作成した 3D モデルを VRM 形式で読み込むことができ、自分のモデルをダウンロードしてもらうことが可能です。モチベーションの上昇にもつながる点やほかの人のモデルを 360 度から見ることができるので参考にすることができます。

6. おわりに

僕は 3D モデルの勉強をするにあたって VRoid Studio はとても良いツールだと思っています。この文章をきっかけに少しでも人が増えて界隈が賑やかになればうれしいです。

参考文献

(＊1) https://booth.pm/ja/items/1308549

VRoid Studio、VRoid Hub

https://vroid.com/

Blender ダウンロードページ

https://www.blender.org/download/

VRChat

https://hello.vrchat.com/

C++の練習

AtCoder の練習問題をもとに

ロボティクスコース　2年　宮原　光敏

1.　はじめに

私はプログラミングを学んでいますが、私の知識はお世辞にも豊富とは言えません。そのため、プログラミングの知識を増やすために、競技用プログラミングサイトである AtCoder で実際にプログラムを作ってみました。プログラムの一つ一つがどういう働きをするのか、どうすればこういう動作をするのか、練習問題を通して覚えていくことにしました。

2.　AtCoder について

AtCoder とは上で述べたように競技用プログラミングのサイトです。様々なコンテストを通して、自分がどれくらいの実力を持っているかの指標となるレートが出されます。
今回は初心者向けの”AtCoder Programming Guide for beginners”(通称 APG4b)にある練習問題と、そこまでの知識を持って解くことのできるコンテストの問題を解いてみることにしました。

3.　練習問題

実際に解いてみた練習問題は 10 番目の問題 EX10「棒グラフの出力」です。実際の問題文などは図 1 を参照してください。練習問題ではサンプルプログラムがあらかじめ存在しており、このプログラムを基に要求された動作を実行するプログラムを作っていくことになります。この問題では点数 A、B それぞれに入力された数字の数だけ「] 」を表示させるように動作させれば正解となります。プログラムの一部はリスト 1 となります。

問題文

A さんと B さんのテストの点数 A, B が与えられます。

2 人の点数を表す横向きの棒グラフを出力してください。

棒グラフは 1 点を一つの]で表し、次の形式で出力します。

A さんの点数個の「] 」

B さんの点数個の「] 」

例えば、A さんの点数が 5 点、B さんが 9 点だった場合、次のように出力します。

A:]]]]]

B:]]]]]]]]]

B さんの棒グラフを出力した後にも改行が必要なことに注意してください。

図 1　EX10 の問題文

リスト 1　プログラムの一部

```
1.  int i = 0;
2.    cout << "A:";
3.    while (i < A) {
4.                cout << "]";
5.                i++;
6.    }
7.                cout << endl;
```

このプログラムは A の部分のみ表示するプログラムです。while という条件が満たされている限りループし続ける文を用いて「] 」を

入力された数だけ表示させます。

ここではAの部分のみ切り取りましたが、Bの部分も同様にして解くことができます。

4. コンテストの問題

今回解いたコンテストの問題は”Hina Arare”という問題です。問題文は図2を参照してください。

問題文

日本では、3月3日にひなあられという、色のついたお菓子をお供えする習慣があります。

1つの袋があり、ひなあられがN個入っています。

この袋には、桃色、白色、緑色の3種類か、桃色、白色、緑色、黄色の4種類のひなあられが入っていることが分かっています。

桃色をP、白色をW、緑色をG、黄色をYと表したとき、袋からひなあられを1粒ずつ取り出していったところ、i番目に取り出したひなあられの色はSiでした。

この袋に3種類のひなあられが入っていた場合はThree、4種類のひなあられが入っていた場合はFourと出力してください。

図2 “Hina Arare”の問題文

この問題はひなあられがN個入っている一つの袋に何種類のひなあられが入っているか求める問題です。袋の中には桃、白、緑の三種類か桃、白、緑、黄の四種類のひなあられのいずれかが入っていることが問題文から知ることができます。この問題のプログラムはリスト2です。

リスト2 プログラムの一部

```
1.  bool flag = false;
2.  cin >> N;
3.  for (int i = 0; i < N; i++) {
4.      cin >> S;
5.      if (S == "Y") {
6.          flag = true;
7.      }
8.  }
9.  if (flag == true) {
10.     cout << "Four" << endl;
11. }
12. else {
13.     cout << "Three" << endl;
14. }
```

この問題の肝はひなあられが必ず黄色を除いた三種類が必ず存在することです。そのため、ひなあられの種類は黄色があるかどうかで決まります。そのため、初めに真か偽かを示すフラグを作ります。

フラグは最初に偽とおいて、黄色を示す「Y」が入力されたときに真となるようにします。そのためフラグが偽の状態だと結果は三種類となり、逆に真の状態だと結果は四種類となります。それぞれに対応した出力をすることでこの問題は正解となります。

5. まとめ

AtCoderの練習問題、コンテストの問題を通して、自分の実力の無さを痛感しました。そして、実際にプログラムを動かして、検証することも大切だと分かりました。これからも競技用プログラミングを続け、実際に使えるようになるまでプログラミングの知識と柔軟性を高めていきたいです。

参考にしたサイト

AtCoder Programming Guide for beginners
https://atcoder.jp/contests/apg4b

Unreal Engine4 で楽してゲーム制作

ロボティクスコース　3年　鈴木　晴斗

1.　はじめに

今回は Epic Games 社が開発した Unreal Engine(以下 UE と略記します)を使用し、とにかく楽をしてゲームを制作するというのがテーマです。ゲーム制作と聞けば、難しいコードをたくさん書く作業を思い浮かべる方もいると思います。しかし、UE に内包されている Blueprint システム(以下 BP と略記します)を使用すれば、面倒なコードは一切書かずに、ノードとノードをピンで接続するだけで、誰でも簡単にゲームを制作することができます。

2.　制作したゲーム

まず初めに今回制作したゲームについて説明します。このゲームは障害物を乗り越えながら、ステージ上にあるすべてのコインを集め、ゴールを目指すゲームです。ステージ内には踏んではいけない足場がいくつか設置されており、そこを踏んでしまうとゲームオーバーとなります。

3.　制作過程

3.1.　基本的な機能の実装

ここからは制作過程の説明に移ります。初めにゲームを構成する上で最低限必要な機能を実装します。実装するのは以下の 3 つです。

- コイン取得
- ゲームオーバー処理
- チェックポイント

コイン取得は初めにプレイヤーキャラクターの BP にコインの枚数を保持する変数と、取得したときに加算処理をする関数(図 1)を定義します。次に新たに BP を作成し、コイン本体を作成していきます。On Component Begin Overlap ノード(図 2)でコリジョンとプレイヤーが触れたかどうかを検知します。コリジョンとは触れたときに何かしらの処理に繋げるイベントを発生させるために追加するものです。次にプレイヤーキャラクターの BP をキャストし関数の参照、加算処理を実行することでコインの取得が可能となります。最後にコインをステージ上から削除することで、コインの取得の一連の流れが出来上がります。

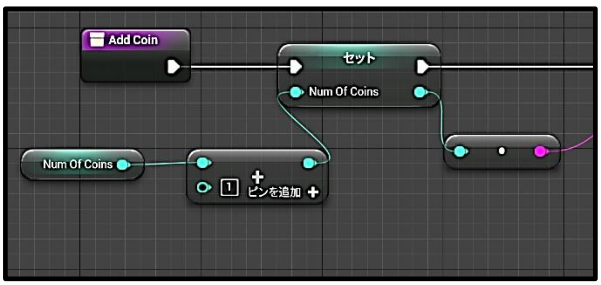

図 1　加算処理を行う関数

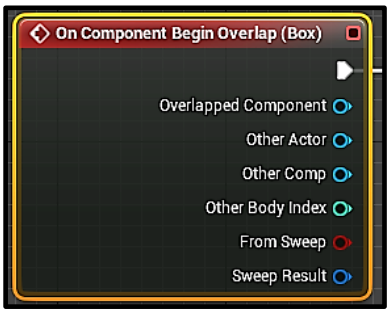

図 2 On Component Begin Overlap ノード

ゲームオーバー処理はまずプレイヤーキャラクターの BP にゲームオーバー時の処理を行う関数を定義します。今回はゲームオーバー時にプレイヤーキャラクターの全身をフニャフニャにすることで、ゲームオーバー感を出します。次に踏むとゲームオーバーとなる足場本体を作成していきます。On Component Begin Overlap ノードでコリジョンにプレイヤーが触れたかどうかを検知します。プレイヤーキャラクターの BP をキャストし関数の参照、処理の実行をします。最

後に画面上にゲームオーバーのメッセージを表示します。このメッセージもウィジェット BP という BP の一種を使用して作成します。

チェックポイントは最初にプレイヤーの座標を保存する BP を作成しておきます。次にチェックポイント本体を作成していきます。On Component Begin Overlap ノードでコリジョンとプレイヤーの接触を検知した後、Sequence ノード(図 3)を使用し「Then 0 を実行したら Then 1…」というように順に処理を実行していきます。初めに Then 0 でプレイヤーの位置を取得し変数に代入します。次に Then 1 で指定したセーブデータが存在するか確認します。存在する場合は指定したセーブスロットに変数を保存します。存在しない場合は新たにデータ保存用の BP を作成し、同様に保存します。次にレベル BP にセーブデータのロード処理を実装します。レベル BP とは各レベル(ステージ)に存在し、レベルに配置されたオブジェクトを操作するなど、レベル全体に渡る処理を記述できる BP です。指定したスロットをロードした後、データ保存用 BP をキャストし保存された座標を取得、それをキャラクターの開始位置として設定します。これでチェックポイントが出来上がりました。しかしこのままだと、新規にプレイをしたとき、開始位置がチェックポイントのままになってしまいます。そこでプレイヤーの初期位置を定数であらかじめ定義しておきます。それをゲームが終了されたときやタイトル画面に戻るときにデータ保存用 BP に上書きすることで、チェックポイントから開始されるのを防ぎます。

図 3 Sequence ノード

3.2. ステージデザイン

基本的な機能の実装ができたら次はステージを作っていきます。ちょうどいい難易度になるようにオブジェクトの位置を調整しながら配置していきます。配置が完了したら、それらにテクスチャを割り当てていきます。テクスチャは UE に付属したものを使用します。ステージは小学生の時によく遊んだ「スーパーマリオ 64」のステージをイメージして作成しました。次の図 4 が実際に作成したものです。

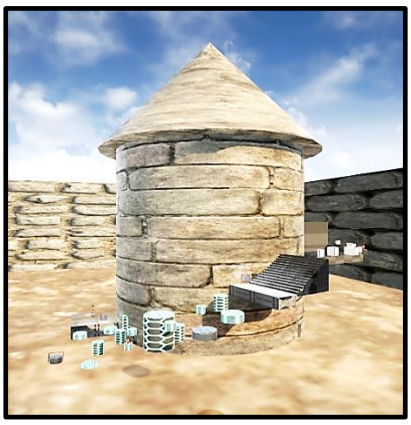

図 4 今回作成したステージ

4. おわりに

今回は 2 度目の UE を使用したゲーム制作でした。以前の作品よりもセーブなどの機能やステージ内のギミックは増えましたが、BP を使うことでとにかく「楽」をしながらそれらを実装することができました。その反面、キャラクターのコリジョンの設定やステージの難易度調整など、機能やギミックだけではない、実際にプレイしたときの快適さや違和感のなさ、操作感を考えてゲームを作るのは大変だと感じました。次作ではそれらを考慮しつつ、さらに面白いゲームにしていきたいです。

参考文献

Unreal Engine 4 で極めるゲーム開発
湊 和久 著

Unreal Engine 4 ドキュメント
https://docs.unrealengine.com/4.27/ja/

.

音圧を上げる

ロボティクスコース　3年　冨山　結都

1.　はじめに

私は自分の作った音の音圧が低いと思ったのでDAWソフトのCubase Elements使って音の音圧を上げました。ここで使ったコンプレッサーはCubase Elements付属のCompressorを、マキシマイザーはこのCubase Elements付属のMaximizerを、エンハンサーはCubase Elements付属のStereoEnhancerを使いました。Cubase Elementsは無料のソフトでは無いですが、DAWソフトは他のDAWソフトでも似通った機能が多いので、ここでの知識を他のDAWソフトでも生かせると思います。

2.　どうして音が小さいか

著者が作った音楽とプロのアーティストの音楽の波形を見て比べます。図1を参照してください。上が著者の作った音楽の波形で下がプロのアーティストの音楽の波形です。上の波形は余白が多く音も小さいです。下の波形は0dBに届きそうなほどあり、上の波形の音よりも音が大きいです。これを見るに波形の縦の大きさと音の大きさは比例の関係にあります。そのため単純に音を上げてみます。

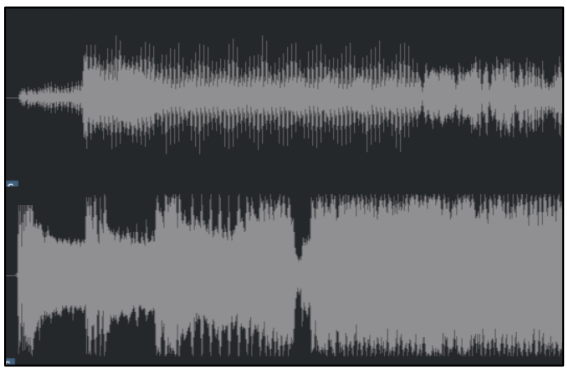

図 1　波形

上げすぎると0dBを超えて音が割れてしまう（クリップ）のでそれだけは絶対にしないようにしましょう。こうして上の波形の音を上げたものが図2です。しかしこれでもまだ音はプロの音と比べると小さく聞こえます。元のトラックで周波数が被ってないかもチェックする必要があります。特に低域のキックやベースが出すぎていると、後述のコンプレッサーやマキシマイザーで音圧を上げにくくなるので、イコライザーで削るといいでしょう。

図 2　波形

3.　コンプレッサー

図2で音がまだ小さいのには、この波形の中で音の大きさの高低差が大きいのが原因でもあります。この高低差を無くすにはコンプレッサーが有効です。コンプレッサーは音を圧縮して小さい音を上げることができるものです。そしてコンプレッサーをかなり使ってさらに音を上げたのが図3です。図2の波形の音よりも大きな音が出るようになりました。途中コンプレッサーをかけすぎてノイズが発生したので、発生しないギリギリを攻めて調整しました。またこの操作の際キックドラムの音が大きい場合、コンプレッサーをかなりかける必要があります。あきらかにキックドラムの音が大きいときは、あらかじめキックドラムの音を小さくした方がいいです。

図 3　波形

4.　マキシマイザー

マキシマイザーも音圧を上げるものです。コンプレッサーと似たように音を圧縮して小さい音を上げるものです。そしてマキシマイザーはコンプレッサーよりも音圧を上げるのに特化したプラグインでもあります。今回は Maximizer の OPTIMIZE を 100%に設定しました。そうしてできたのが図 4 です。大分波形で埋まってきました。

図 4　波形

5.　エンハンサー

エンハンサーは音の広がり感をコントロールするプラグインです。今回は音を広がっているように聞こえるよう使います。WIDTH のつまみを操作して音の広がり感を操作できます。広げすぎると豪華にはなりますが、もともとのキレが無くなるのでちょうどいいところを模索しましょう。

6.　おわりに

私は常々自分の作った音楽はプロが作った作品と比べて音が小さいなと思っていました。前から音圧を上げようとは思っていましたが、プロがやることは自分もできないだろうと思っていたので、調べていませんでした。今回調べてプロの作品にはまだまだ追いつけないけど、昔の自分と比べると劇的に作品が良くなりもっと早くから調べておけばよかったです。

参考文献

マキシマイザーとは？DTM の必須プラグインマキシマイザーについて！オススメ・フリーのマキシマイザーも紹介！
https://tokep.hatenablog.com/entry/What-is-a-Maximizer%3F

コンプレッサーの使い方＆テクニック【初心者のためのミックス講座 6】
https://www.dtmfb.com/comp-howto/

ミックスで「コンプのかけ過ぎ」を避ける 7 つのコツ
https://wavesjapan.jp/articles/waves-7tips-to-avoid-over-compressing-your-mix

DTM で絶対にやってはいけないたった 1 つのタブー
https://yuuto-kannami.com/dtm-taboo

音圧の上げ方（音を大きくする方法）
https://www.dtmfb.com/onatsu/

ミックスの音圧を上げる 7 つの方法
https://dtmhacker.com/mixing-loud/

ステレオエンハンサー(イメージャー)とは？効果を max やマイナスかけで検証
https://mmcd-web.sounds-stella.jp/2110/what-is-stereo-enhancer-imager/

Fusion360 を用いた
有限要素法

機械・エネルギーコース　4年　鈴木　雄裕

1.　はじめに

物質の特性を調べる方法は様々な方法があります。今回は比較的手軽にできる有限要素法を試してみました。有限要素法は物体を細かく分割して計算し、計算結果を足し合わせることで全体の変位を解析する方法です。

今回は比較的容易に使用できる FreeCad を用いました。FreeCad は無料で利用できる 3DCAD ソフトウェアです。無償のソフトウェアでありながら、3D モデリング、メッシュデザイン、製図、有限要素法、レイトレーシング、ロボティクス機能など、便利な機能が多く実装していることが特徴です。

2.　有限要素法

2.1.　片持ち梁

今回は片持ち梁の解析を行いました。片持ち梁は片端が固定されているだけの梁です。片持ち梁は工場で荷物を吊り上げ、移動に用いるジブクレーンや建物の軒先に用いられます。片端しか固定されていないので他の梁と比べるとたわみやすいことが特徴です。

2.2.　有限要素法

FreeCAD を公式サイトからダウンロードし、起動し、新規作成を選択します。ワークベンチの”Part Design”を選択し”Body を作成”から梁の平面を描きます。これを押し出すことで梁を作ります。先ほど作成した平面を選択し”選択されたスケッチを押し出し”を選択し、押し出します。これで平面に厚みを持たせることで立体になりました。今回は図 1 のように 10mm の正方形を 100mm 押し出した梁を作成しました。

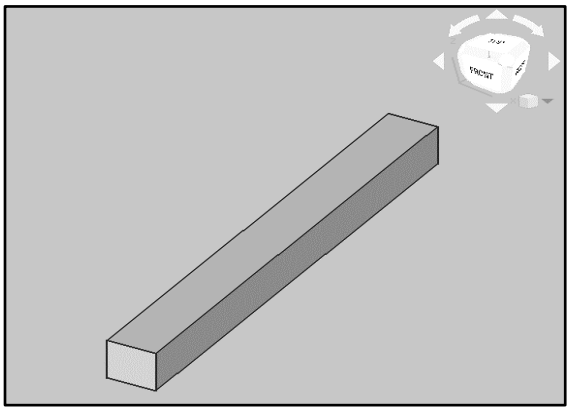

図 1 片持ち梁

梁を作成したら梁を解析が出来るような形に変形します。ワークベンチから”FEM”を選択します。”Create a FEM mesh” を選択しメッシュを作成します。メッシュは物体を細かく分割した要素のことです。曲面等の解析することが難しい形状は解析しやすいような形状に分ける必要があります。基本的にメッシュを小さく切るほど解析したい形状に近くなり、計算精度が上がります。FreeCad では主にメッシュの最大サイズや細かさを設定します。最大サイズは最大でどれくらいの大きさまで許容するかを設定する項目です。細かさはどれぐらい丁寧にメッシュを切るかを決める項目です。どちらも細かく設定したほうが精度向上に繋がりますがパソコンに負荷がかかるので注意が必要です。

次にどのような条件で力がかかっているかを設定します。メニューバーから、”FEM 固定拘束を作成”を選択し片端を固定します。次に”FEM 荷重拘束を作成”を選択し任意の面・線に荷重を掛けます。今回は図 2 のように設定します。左端の矢印がかかっているほうは荷重がかかっており、もう一端が固定されています。

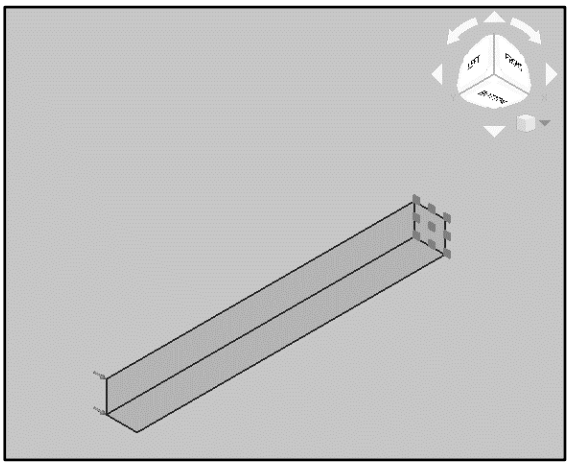

図 2 片持ち梁の条件設定

“Create a FEM material”を選択し任意の性質を付加します。具体的に数値を代入して計算することも出来ますが FreeCad が設定している数値を使います。今回は Steel-C40E の値を使用しました。これは炭素鋼の一種です。”選択したソルバーの計算を実行” からデータを読み取ります。力を掛けた方向の”Displacement” を見るとどれくらいたわむか分かります。力のかかり具合の強弱は図 3 のように色付けされ表示されます。赤に近いほど強い力がかかっており青に近いほど弱くなります。図 3 では荷重がかかっている右端が赤く表示され、固定されている左端に向かうにつれ青く表示されています。また変化した量が小さく変化がわかりにくい場合は”表示”のスライダーから変化した量を大きく誇張して表示させることが出来ます。

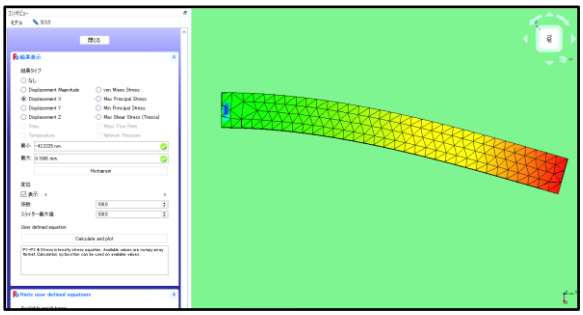

図 3 解析結果

3. おわりに

今回の部誌では片持ち梁の解析を行いました。経験のないなか手探りで作業しましたが後ほど微分方程式を用いてどれくらいたわむのか計算してみたところほとんど同じ値になりました。何となく難しいものと思っていましたが、実際に触ってみると丁寧に解説されたサイトが数多くあり、操作自体もあまり複雑なものがなかったので躓かずに出来ました。これを読んだ方も試してみて頂けると嬉しいです。

参考文献

[1] FreeCAD の公式ページ
https://www.freecadweb.org/downloads.php?lang=ja

[2] FreeCAD 初心者講座
https://www.youtube.com/watch?v=sp2xwiUJoTs&list=PLM6uUGzQqAmpA8p9V58ASKT05wLaPsnIM

[3]無償 3D CAD「FreeCAD」で有限要素法解析に挑戦！
https://monoist.atmarkit.co.jp/mn/articles/1601/14/news022.html

[4] 片持ち梁のたわみを求める方法
http://kentiku-kouzou.jp/struc-harinotawamikatamoti.html

[5] 断面二次モーメントとは？1 分でわかる意味、計算式、h 形鋼、公式、たわみとの関係
http://kentiku-kouzou.jp/danmen2zi.html

編集後記

レーショナル 2021web 版、いかがでしたでしょうか。

今年はコロナウイルスの影響により、例年行われていた高専祭の一般参加は中止となり、また部誌の配布も行わなかったため、このレーショナルをご覧になられた方は少ないのではないでしょうか。もしご覧になられていない方は、ぜひ部員一人ひとりの個性あふれるレーショナルをお楽しみください。

最後に、このレーショナルに関わってくれたすべての方々に感謝いたします。

ありがとうございました。

編集長　鈴木　晴斗

Rationale Volume35

発行日	10 月 25 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	日野　綾瀬	
編集	鈴木　晴斗	
校正	鈴木　雄裕	岡田　曜
	日野　綾瀬	鈴木　晴斗
	冨山　結都	稲垣　颯起
表紙	鈴木　晴斗	
執筆	ソフトウェア研究部員	
製本	ソフトウェア研究部員	