

プログラマ

プログラマ



立命館コンピュータクラブ

RCC会誌  
2022

だ  
け  
し  
や  
な  
い



# まえがき

冊子を手にとっていただきありがとうございます、編集者のdaikiとhaneです。RCCとして一年振りの出版です。立命館コンピュータクラブ(以下、RCC)では、情報系に関する内容を自由に決めて活動しているサークルです。

この冊子はRCCの主な活動や近況、また活動の中でもメインであるプロジェクト活動の成果報告、そして会員の個人活動を紹介するコラムを掲載しています。プロジェクト活動のコラムでは今年度行われている内容を掲載しています。今年度は会員コラムの量が例年ほどはありませんが、今年もさまざまなジャンルのお話が載っていますので、どのような方でも楽しく読むことができると思います。

さて、今年度はCOVID-19の感染拡大が比較的落ち着いていた為対面での活動を再開出来ました。また、部室も使えるようになり、定例会議前は椅子が埋まるくらい会員が集まりとても賑やかになっています。会員の皆様にはバラエティに富み、大変興味深い個人コラムを提出してくれて編集者として大変喜ばしく思います。どうぞ、ごゆっくりお楽しみください。

また会員の皆さんにおきましては編集者である私の都合で大変な迷惑をかけたことをここに謝罪し、この会誌に関わった方々に感謝申し上げます。

2022年

編集者 羽田秀平  
宮寺大樹

# 目次

1章 イントロダクション	..... 3
1.1 立命館コンピュータクラブとは	..... 3
1.2 活動内容	..... 3
1.3 普段の活動	..... 3
1.4 活動拠点(サークルルーム)	..... 4
1.5 入会について	..... 4
1.6 アクセス	..... 4
1.7 2022年度 年間行事予定	..... 5
2章 プロジェクト活動	..... 6
Unity班	..... 7
LT プレゼン班	..... 9
3章 会員コラム	..... 11
1回生	..... 12
2回生	..... 29
3回生	..... 41

# イントロダクション

## 1.1 立命館コンピュータクラブとは

立命館コンピュータクラブ(Ritsumeikan Computer Club、以下RCC)は、立命館大学びわこ・くさつキャンパスを拠点とした学術系サークルです。

## 1.2 活動内容

RCCは主に2点の活動を軸に行っています。

- 研究活動  
情報に関する分野について、自由にテーマを決めて調査・議論を行い、情報技術と社会についての見識を深めていきます。研究成果は報告書という形でまとめ、Webサイト上での公開や、本会誌にまとめて配布をしています。
- 制作活動  
ソフトウェア、ハードウェア、映像、音楽などの制作を行います。情報技術を使うものであれば自由に制作できます。会内では、Welcomeゼミ、ハッカソン、夏季制作など制作活動を体験できるイベントを設けています。

## 1.3 普段の活動

普段は、週一回、定例会議があり、活動予定などの連絡が行われます。この時にライトニングトーク(LT)という、情報系の内容をテーマとして自由にプレゼンテーションを行う時間もあります。全体向けに勉強会が開催されることもあります。また、会員はいずれかのプロジェクトに参加し、対面で活動しました。

## 1.4 活動拠点(サークルルーム)

サークルルームには、様々な備品が保管されています。本棚には研究活動で使用する資料や、制作活動で参考になる技術書が数多く保管されています。

## 1.5 入会について

新規入会は、通例では年度初めの5月から受付を開始しています。学年を問わず、本学の学生ならどの回生からでも入会することができます。

会費は年間6000円です。会費は会の重要な資金源となっていて、日々の活動や情報のシステム維持、会員専用PCの購入費などに利用されています。

## 1.6 アクセス

サークルルームは、BKCバイオリンク一階のサークルルーム8です。食堂や書店などがあるリンクスクエアの裏側の建物の1階に位置しています。サークル宛に質問や連絡等がございましたら、以下の連絡先までお願いします。

----- RCC Information -----

Address:〒525-0058

滋賀県草津市路地東1丁目1-1

バイオリンク1Fサークルルーム8

Email:[rcc.liaison@gmail.com](mailto:rcc.liaison@gmail.com)

Twitter:[@rits\\_cc](https://twitter.com/rits_cc)

Website:[www.rcc.ritsumeai.ac.jp](http://www.rcc.ritsumeai.ac.jp)



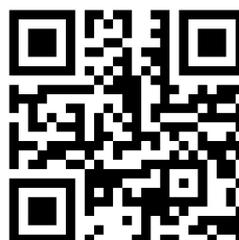
バイオリンクの入口

## 1.7 2022年度 年間行事予定

4月	Welcomeゼミ	新入生と上回生が共同で勉強・製作
5月	前期活動開始	前期プロジェクト活動のスタート
9月	KC3	関西情報系学生団体交流会
	RCC総会	RCC会内の前期総会
10月	後期活動開始	後期プロジェクト活動のスタート
12月	学園祭	制作物の展示・発表
2月	追い込み合宿	後期か通洞の制作物・報告書の仕上げ
	プロジェクト成果発表会	後期活動の成果(制作物・研究)を発表
	RCC総会	RCC会内の後期総会
3月	追い出し会	卒業する4回生と過ごせる最後のイベント

KC3とはKC3運営委員会が毎年開催している交流会で、関西圏にある大学のコンピュータ関連団体を対象として、お互いの活動を紹介しあったり、勉強会を開催して新たな技術を学んだりしながら、他の団体の方々と交流するイベントです。今年度は3年ぶりとなる会場での開催となり大盛況でした。

ウェブサイト→



# 2章 プロジェクト活動

プロジェクト活動とは、1年を通して行われるRCCのメイン活動です。情報科学の研究をし、その成果の発表を活動の基本として会員間で相互の親睦を図るとともに学術分科の想像と発展に寄与することを目的とします。

今年度は4つのプロジェクトが立案され、ここではその活動の成果を報告します。

## 2.1 Unity班

ゲームエンジンであるUnityの使い方を学び、そしてその幅広い応用分野を生かした制作物を作ることを目標として活動しました。

## 2.2 LT・プレゼン班

LTを見つめ直すことにより、専門性とわかりやすさを両立したLTを制作、探求できるようになることを目標に活動しました。

# Unity班

山本 京介                      田尻 聖奈                      古川 聡悟  
情報理工学部 3回生   情報理工学部 3回生   情報理工学部 2回生

駒谷 亮叡                      佐久間 智也                      KIM Woohyeon  
理工学部 3回生                      理工学部 3回生                      情報理工学部 1回生

助野 朔蘭                      早川 竜生                      CHOI Youchan  
情報理工学部 1回生                      理工学部 1回生                      情報理工学部 1回生

## 活動概要

ゲームエンジンUnityについて学習する。初学者を主な対象とし、Unityとは何をするもので何ができるのかといったことから、基本的な操作方法、スクリプトの書き方などを学習する。

その後各自制作に移っていく。最終的に各自1つ作品を完成させることを目標とする。ただし、ここでは作品をゲームに限定しない。というのも、Unityは”ゲーム”エンジンではあるが、ゲーム制作以外にも様々な分野に応用できるため、ここでは自由な発想で何かUnity製の作品を作ってもらうべくあえて限定しなかった。

## 春学期の活動

春学期は「Unity2021入門」という書籍をベースにUnityのチュートリアルを行った。各章の注釈や先の書籍には載っていない小ネタなどの資料はScrapboxにまとめ、共有した。以下、各週の活動内容について述べる。

## 第0週 環境構築

事前にUnity IDの作成とUnity Hub、Unity2021.3.4f1、Visual Studioのインストールなどを行ってもらった。

## 第1週 Unityに触れてみる

第1週目はUnityの操作方法について解説した。その他、Unityそのものについての解説や活用事例の紹介を行ったり、LTSストリームの説明やドキュメント類の向き合い方についての説明を行った。

Unityは使用者が多く、参考サイトが多いことがUEなどのゲームエンジンとの差別化点だが、反面古い情報や不確かな情報も多いため、注意が必要であり、可能ならば公式ドキュメントを参照するべきである。

## 第2週 2Dの基本

第2週ではUnity2021入門を参考に2Dゲームの制作を行った。とにかく手をうごかしてUnityで作業する感覚を身に付けてもらうべく、素材やスクリプトは書籍付属のものを使い、手早く行った。ただし、スクリプトの中身は書籍よりも詳細に解説を行い、今後のスクリプティングに活かせるようにした。他には軸の概念やプレファブの作成方法などを解説した。

## 第3週 3Dの基本

第2週同様書籍を参考にしつつ3Dゲームの制作を行った。なお、この章では配布素材はなくAsset Storeから素材を取ってくる方式だった。この章ではキャラの動かし方やステージの制作方法の他、UIの使い方、サウンド効果の設定方法などを学習した。

## 第4週

第4週では制作の取り組み方についての説明を行った。今まで扱ってきた書籍は入門書であるためこれだけでは満足なものを作ることはできない。そこでPhotonやAR Foundation、MRTKなどのツールや今後役立つような書籍を紹介した。

## 今後の展望

秋学期は各自制作を進めてもらいつつ、活動日は進捗確認を行っていく。

# LT プレゼン班

宮寺 大樹

梶原 悠人

理工学部 2回生

情報理工学部 1回生

坪田 幸輝

XU Zimeng

情報理工学部 3回生

生命科学学部 1回生

## 1. LT・プレゼンテーションとは？

プレゼンテーションとは情報の伝達手段の一つであり、特に視覚情報を重視しています。図や表、アニメーションなどが多く使われています。次にLTとはLightning Talksの略であり簡単に言えば短いプレゼンテーションのことです。

## 2. LT・プレゼン班とは？

プレゼンテーションをする機会が高校と比べて多くなっていく大学や企業において効果的なスライドの作り方やプレゼンのやり方を勉強し、各自スライドを作りそれを班員で評価・ブラッシュアップしていこうという活動をしていました。

## 3. 知識

### 3.1 聴衆の想定

プレゼンを行ううえで1、2を争うほど大事なことは聴衆を想定することです。

聴衆を想定するとは聞く集団がどのような属性や特性を持っているかを把握し、それに合わせてプレゼンテーションの内容やデザインを吟味します。

### 3.2 5W2H

人に会話や文章で情報を伝えるときにとても大事な5W2Hですが、プレゼンにおいても押さえておくべき重要なものとなります。ちなみに5W2Hとは(When いつ, Where どこで, Who 誰が, What 何を, Why なぜ, How どのように, How much どれぐらい)のことです。

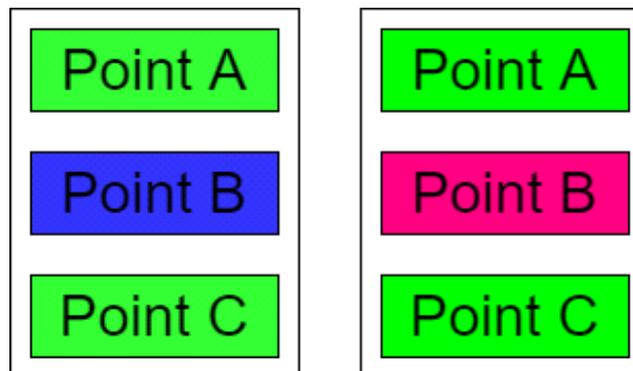
### 3.3 配色

視覚で情報を伝達する手段において色はとても重要な役割を持っています。その色を適切に配置するのが配色と言われるもので、アニメなどの映像作品では配色を専門に扱う色彩設計という仕事が存在します。私は専門家ではないのであまり偉いことは言えないのですが、ここで有名なものを2つ紹介したいと思います。

一つは色相環と呼ばれるもので下の図のようなもので環状に様々な色が並んでいます。



この環の中心を挟んで反対側に位置している色が補色関係となっており、これを利用したものが下の例です。



Point Bを強調させるように2つの図を作ってみました。左の図は何も考慮せずに作り、右の図は色相環を考慮して作っています。多分ほとんどの人は右の図のほうが(理論上)見やすいと思います。

また、色相環は2色の場合のパターンですが、これを3色以上で用いたものをカラーパレットと言います。アドビカラーやカラーマインド、カラーホイールなど様々なサイトがあり、これを利用すると簡単に色の選択が出来るようになります。

次に、色には特性がありそれを利用すると他の色を使う時よりも効果的に情報伝達することが出来ます。例えば禁止を表すスライドでは赤色を、許可を表すスライドでは緑色を使うなどがあります。

このように色に関する知識を用いればスライドのレベルが上がるので、もし興味があれば勉強をしてみてくださいはいかがですか？ちなみに先輩のおすすめの本は「なるほどデザイン 筒井 美希(著)」だそうです。

## 4. 最後に

ここまでいくつかの知識を書いてきましたが最終的にはプレゼンの経験の量が重要となってきます。特に本番での緊張は場数を踏みプレゼンという環境に慣れるのが一番早いと思います。

## 3章 会員コラム

この章では、学年にかかわらず、会員一人ひとりの情報系の技術に関する知見や興味のある分野に対する熱意や想い、その魅力などを綴ってもらいました。情報系の技術に初めて触れたことについて語ってくれた会員もいれば、既に精通していて専門的な内容を語ってくれた会員もいます。

コラムの内容は、アプリケーション、Webサービス、プログラミング言語、セキュリティ、AI、ゲーム、デザインと多岐にわたり、本会の活動の幅広さと、各会員がどのような分野に興味を持っているのかを感じていただけるかと思います。

初めて情報系に触れる方でもわかりやすいように書かれておりますので、様々なコラムに目を通して情報系の魅力を感じていただけたらと思います。ぜひお気に入りのページを探してみてくださいね。

# C言語で計算機を作成する

学部/回生	情報理工学部 知能情報・1回生
名前	koji



## 1,初めに

私は今現在プログラミングに関してあまり明るくなく、C言語を学んでいる最中であった。故に今回はC言語を用いたプログラムにて初歩である計算のプログラムの作成を通しC言語への理解を深めることを目標とした。

## 2,研究内容

ソーシャルゲーム「アイドルマスターシャイニーカラーズ」、通称「シャニマス」におけるWING育成でのオーディションのアピール値の計算を行うプログラムを作る。使用言語は「C言語」。

シャニマスのオーディションにおけるアピール値の計算方法は既出であり、具体的には以下の通りである。

”プロデュース(Pアイドルのライブスキル)のとき

$= 2.0 * Pアイドルの該当ステ + Sアイドル4人の該当ステ合計 * 経過週補正$

経過週補正

$= 0.2 * (1.0 + 0.1 * 経過週)$

経過週 = 0 ~ 33

プロデュース(Sアイドルのライブスキル)のとき

$= 0.5 * Pアイドルの該当ステ + (Sアイドル4人の該当ステ合計 + 3.0 * アピールするSアイドルの該当ステ) * 経過週補正$

経過週補正:Pアイドルのときと同様”

(シャニマス非公式Wikiより引用)

今回はこれを参考にした。

また、上に記されている計算以外に、アピールの成功時、失敗時の倍率計算もプログラムに組み込む。

## 3,結果

計算値と実測値との間に平均約+0.58%程度の誤差が出てしまった。誤差の原因として、小数点以下の値の扱いが相応しくなかったと考えられる。実際この計算の過程で小数点以下の値を切り落とす作業があり、その処置が上手く行ってなかったのではないかと推測される。

アピールの判定:1  
アピールの値は10689となる

計算値



実測値

## 感想

授業の課題以外にて初めてプログラムを組んだが、存外面白かった。ループ処理や配列の扱いに若干手間取ったが、既存の計算式を流用するのみであったため、想像よりかは容易であった。最も苦労した点は変数の名前の決定であり、一目で何の値であるのかが分かるように努めていたが

終盤では自身が理解できれば良しといった名付けになってしまったのは反省点である。Vo, Da, Viの値の入力を分けたが、意味を為さなかった点も反省点である。

今回扱った計算は特別な要因が絡まない場合のものであり、特殊な処置を行う必要がある計算も実際は存在する。しかし今回はその点は考慮せずにプログラムを組んだ。改善案として、今後その処理も上手く行えるようにしていきたい。以下プログラムを示す。

```
1 #include <stdio.h>
2 int main (void)
3 {
4     int p, s, i;
5     int Pap [3];
6     int Sap [4][3];
7     int ap;
8     int app1, app2, app3, app, appe;
9     int ProS;
10    int week;
11    double weekp;
12    int buffer;
13    double buf;
14    int touch;
15    double mag;
16    double appmag;
17
18    do{
19        printf("[0]Vo\n");
20        printf("[1]Da\n");
21        printf("[2]Vi\n");
22        printf("アビールする属性は？:");
23        scanf("%d", &ap); /*アビールする属性の選択*/
24    }while(ap > 2 || ap < 0);
25
26    if(ap == 0){
27        printf("アイドルのVoの値:");
28        scanf("%d", &Pap[ap]); /*アイドルのVoの値の入力*/
29        for(i = 0; i <= 3; i++){
30            printf("アイドル%ldのVoの値: ", i+1); /*アイドルの各Voの値の入力*/
31            scanf("%d", &Sap[i][ap]);
32        }
33    }
34    else if(ap == 1){
35        printf("アイドルのDaの値:"); /*アイドルのDaの値の入力*/
36        scanf("%d", &Pap[ap]);
37        for(i = 0; i <= 3; i++){
38            printf("アイドル%ldのDaの値: ", i+1);
39            scanf("%d", &Sap[i][ap]);
40        }
41    }
```

```
42    printf("アイドルのViの値:"); /*アイドルのViの値の入力*/
43    scanf("%d", &Pap[ap]);
44    for(i = 0; i <= 3; i++){
45        printf("アイドル%ldのViの値: ", i+1); /*アイドルのViの値の入力*/
46        scanf("%d", &Sap[i][ap]);
47    }
48    }
49    }
50    }
51    /*アイドルの担当ステータスの合計*/
52    for ( i = 0; i <= 4; i ++){
53        buffer += Sap[i][ap];
54    }
55
56    /*経過週の入力及びそれに付随する計算*/
57    do{
58        printf("経過週は？(0~33週):");
59        scanf("%d", &week);
60        }while(week < 0 || week > 33);
61
62    weekp = 0.2 * (1.0 + 0.1 * week);
63
64    /*アビール礼選択*/
65    do {
66        printf("[1]PV\n");
67        printf("[2]SV\n");
68        printf("アビールするのはPVかSVか?");
69        scanf("%d", &ProS);
70        }while(ProS < 1 || ProS > 2);
71    if(ProS == 1)
72        app1 = 2*Pap[ap] + buffer * weekp;
73    else if(ProS == 2){
74        printf("アビールするサポートの番号:");
75        scanf("%d", &i);
76        app1 = 0.5*Pap[ap] + (buffer + 3 * Sap[i-1][ap]) * weekp;
77    }
78
79    printf("アビール倍率は?");
80    scanf("%lf", &appmag);
```

```
81
82    /*バフの入力および計算*/
83    printf("該当属性のバフの合計(小数入力):");
84    scanf("%lf", &buf);
85    app2 = app1 * (1 + buf);
86
87    printf("[1]Perfect\n");
88    printf("[2]good\n");
89    printf("[3]normal\n");
90    printf("[4]bad\n");
91    printf("アビールの判定:");
92    scanf("%d", &touch);
93    }while(touch < 1 || touch > 4);
94
95    if (touch = 1)
96        mag = 1.5;
97    else if (touch = 2)
98        mag = 1.1;
99    else if (touch = 3)
100        mag = 1.0;
101    else if (touch = 4)
102        mag = 0.5;
103
104    app3 = app2 * mag;
105
106    appe = app3 * appmag;
107
108    printf("アビールの値は%dとなる", appe);
109
110    return 0;
111 }
```

# Flutter入門

学部/回生

情報理工学部・知能情報コース 1回生

名前

ako



はじめに

ハッカソンに参加した時に使用したFlutterについて書いていきます。FlutterとはUIを開発するためのフレームワークのことです。アプリを作成する際によく使用されます。例えば、地図アプリ、マッチングアプリ、画像編集アプリを開発することが出来ます。環境構築から説明します。

## 1.環境構築

flutter\_macos\_2.0.2-stable.zipをダウンロードします。  
任意のフォルダで解凍します。

## 2.コマンドを実行

```
$cd ~/development  
$unzip ~/Downloads/flutter_macos_2.0.2-stable.zip  
$export PATH="$PATH:~/development/flutter/bin"  
$source ~/.bash_profile  
$flutter precache
```

開発に必要なデータがインストールされているか確認する。

```
$flutter doctorを実行
```

最後の行にdoctor found issues in 1 categoryが表示されることを確認

## 3.iOSシミュレータでアプリを実行

```
$ open -a Simulator
```

アプリを起動します。

```
$flutter create my_app
```

```
$cd my_app
```

```
$flutter run
```

実行すると↓のような画面が表示される。この画面になるとアプリを起動出来ている。



コマンドプロンプトでqキーを押すと  
アプリが終了する。

# 壁紙をカッコよくしよう

学部/回生 情報理工学部実世界情報コース1回生

名前 尾崎真央(MAOZ)



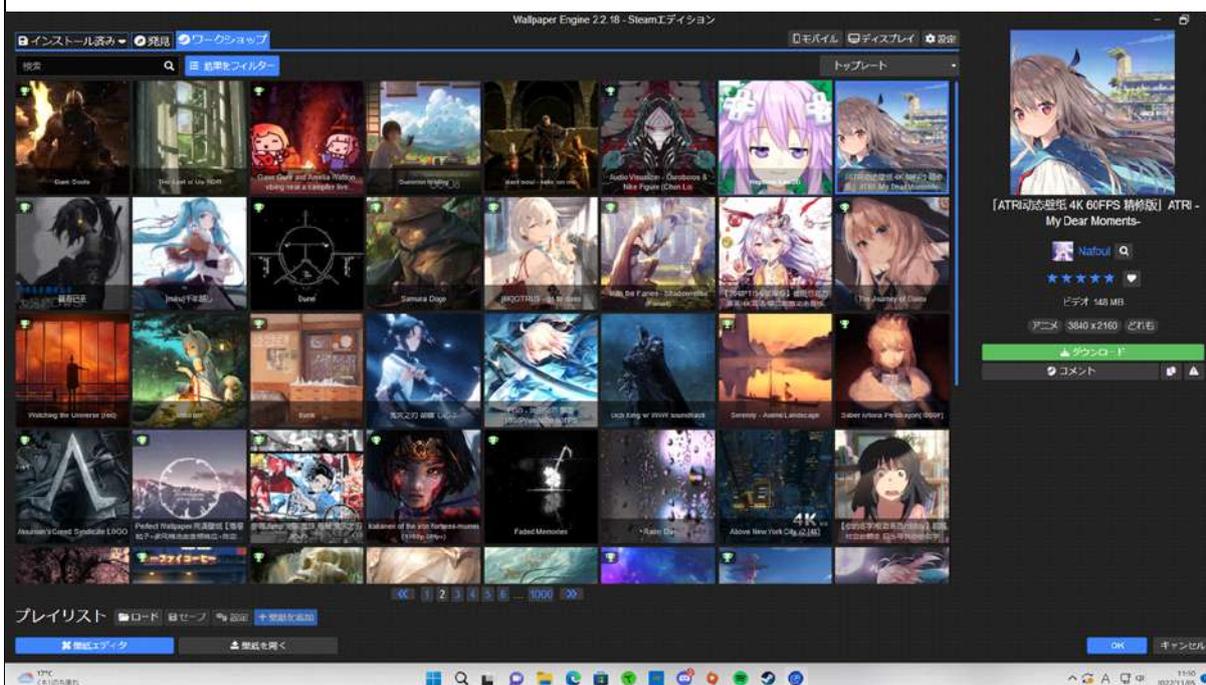
PCをおしゃれにするのに必要な壁紙。今回そんな壁紙にアニメーションやエフェクトを付けられるソフトがあるということを知ったので、そのレビューを書こうと思う。

## 1.使用ソフト

steamで配信されている”wallpaper engine”を使用する

## 2.壁紙の設定方法

”wallpaper engine”を起動したら、上のほうにあるワークショップを選択する。



そうすると様々な壁紙を選択できるようになり、気に入った壁紙を見つけ、選択すると右側中央にダウンロードという表示が出るので選択する。

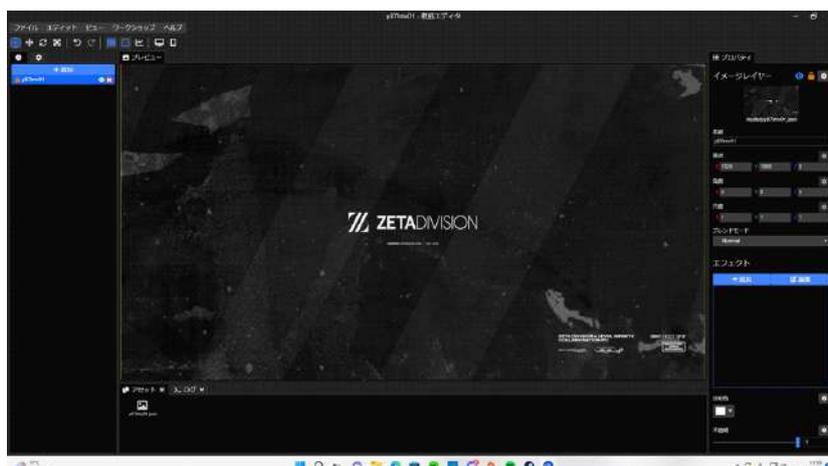
こうすることでソフトにDLされ、その壁紙を設定できるようになる。壁紙の変更方法は、上にあるインストール済みから変更したい壁紙を選択し、先ほどのダウンロードと同じ位置にある登録を押せば壁紙を変更できる。

### 3.壁紙のデコレーション方法

このソフトでは、持っている壁紙をデコレーションすることもできる。  
下のほうにある壁紙エディタを開くと、壁紙を作成というのが出てくるのでそちらを選択する。



そして既存の壁紙を選択すると次の画面になる



左側の画像ファイル名が書いてある所の上のほうに書いてある追加を押すとエフェクトを追加できるようになる。  
雷や雲などさまざまに設定可能だ。  
ぜひこの機能を使って、壁紙をデコレーションしてみたい



# RPGゲームを作ろう



学部/回生	情報理工学部 SAコース 1回生
名前	梶原悠人

簡単なゲームを作りたいが、プログラミングに自信がない。そんな場合、ゲーム制作ツール「RPGツクール」を用いることを勧める。今回は簡単なRPGゲームを作ることを目的とする。

## 1. RPGツクールとは

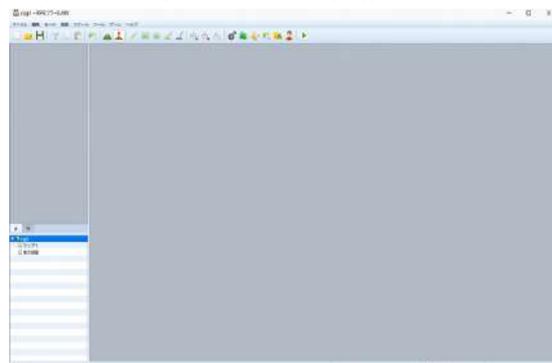
株式会社KADOKAWAが発売しているRPGの制作ツール。このツールの最大の特徴は、プログラミングをせずにオリジナルのRPGを制作することが出来る点である。あらかじめセットされている命令、画像、音楽等を組み合わせることで、簡単にゲームを作ることを可能としている。

## 2. 準備

RPGツクール公式サイトにて、「RPGツクールMV」を購入する。また、30日間お試して使える体験版が同サイトにてダウンロードできるのでそちらでもよい。(体験版と製品版で相違点はない)

## 3. 起動とプロジェクトの作成

ダウンロードして、「RPGツクールMV」を開くと下のような画面になるので、左上のファイルからプロジェクトの新規作成を行う。

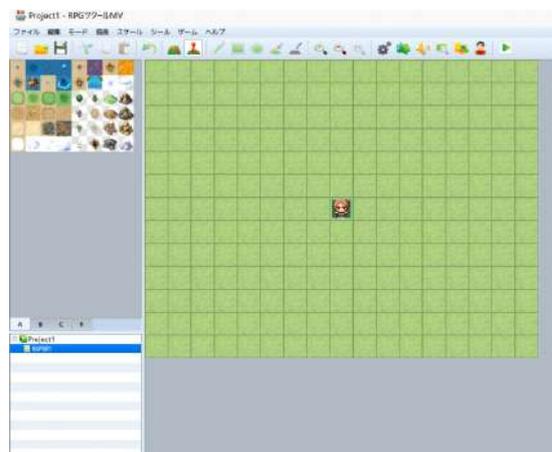


## 4. マップを作る

プロジェクトの作成を終えると右下のような画面になる。画面左上にマップタイルと呼ばれるものがあり、使いたいものを左クリック後に中央画面でドラッグすることでマップを描くことが出来る。

### 1. イベントの設定

2つ以上のマップを作り終わったらそれらをキャラクターが行き来できるように設定する必要がある。マップの出入り口にしたいところで右クリックをした後にイベントの簡単作成、移動をクリックするだけでマップの移動の設定をすることが出来る。



他のイベント(看板を読む、ゲームクリアのためにある場所に触れる)といったものもマウスでクリックしたり、キーボードで文字入力をするだけで簡単に設定することが出来る。

## テストプレイ&完成

最後にテストプレイをして、自分の思うように設定できており、問題もなければ完成である。で、完成したマップが右の画像で、ゲームの内容としては洞窟にある台座に触れればクリアといった簡単なものである。



## 7. 終わりに

「RPGツクール」を用いてゲーム制作を行ってみたが、感覚的に操作することができ、複雑な操作も必要としないのにかく簡単にゲームを作ることが出来た(複雑な設定を行っていないからかもしれないが)。プログラミング能力を養うことはこのツールでは出来ないが、ゲーム制作に必要な発想力やアイデアを得ることは出来ると思われる。プログラミング無しでゲームを作ることが出来るというのがこの制作ツールの魅力であるが、もちろん自分や他人が作ったプログラミングコードを利用することが出来る。それ故、内蔵されていない設定を自分で加えることができ、制作範囲の幅を広げることも可能なので、ゲームを作りたい初心者の方だけでなく、そうでない方も一度触れてみたらいかがだろうか。

### 参考文献

「RPGツクールとは || RPGツクールMV」

<https://rpgmakerofficial.com/product/mv/about/index.html>

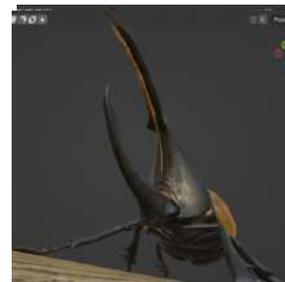
# 幸せならAPI叩こう

学部/回生

情報理工学部・SN 1回生

名前

ヘルクレスオキシデンタリス



幸せなら手をたたこう なぜ？

童謡としても親しまれている。作詞は早稲田大学人間科学部名誉教授の木村利人。元々は木村が学生時代フィリピンでボランティア活動をしていた際に原曲を耳にし、木村が帰国の途に着いた際に、旧約聖書の詩篇47篇を参考にして詞をつけたものが元になっている。

～API とは～

アプリ と プログラム の 仲立ち

Application Programming Interface

例えばその道の企業が公開したAPIを利用することで提供サービスをプログラムに組み込んで開発しやすくするような物を指します。

中でもWeb APIは分かりやすく、多種多様です。

開発のインスピレーションになるかもしれません。

<https://qiita.com/danishi/items/42d8adf6291515e62284>

Web APIはhttp/httpsを利用してネットワークから呼び出すことができるAPIです。

表題通り、curlコマンドでWeb APIを叩きたいところですが

著作権などの関係上叩きたいAPIを紹介できないため割愛します。

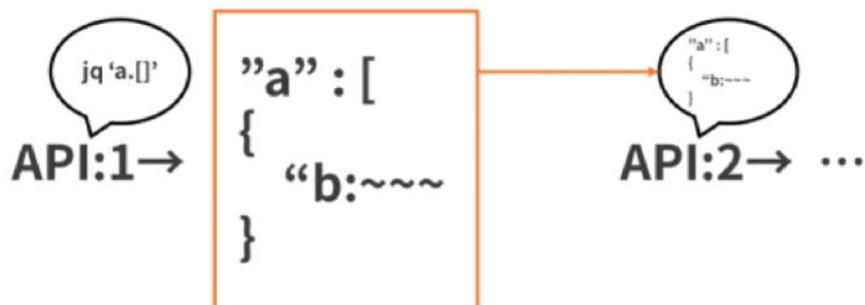
～API 運用～

APIによりプログラム内の機能に応じた開発、いわゆるマイクロサービスが実現できます。このようなサービスの形式が確立され、現在のWebアプリケーション開発においてAPIは欠かせない存在となっているようです。API設計やAPIの保護に関して分厚い本が書かれるのも納得です。

上記を踏まえて、APIの挙動に着目することでプログラムの解析が容易になる例を見ていきましょう。

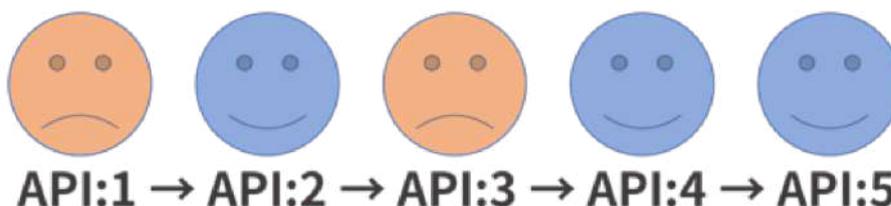
次ページより、APIコール間のデータ依存を用いた例です。

## APIコール間のデータ依存



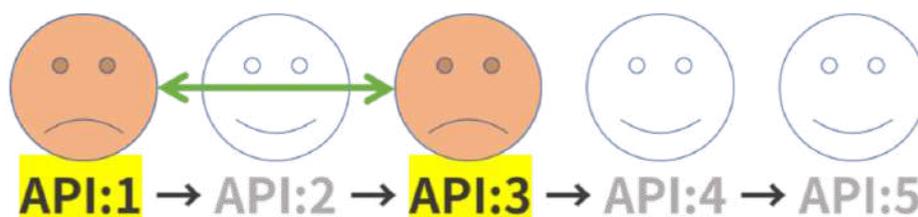
APIの挙動に注目すれば、そのプログラムが何をするのかが予想できるかもしれません。上の図はAPIがAPIを呼び出すような依存関係を持つ例を示していますが、このような挙動をするAPIを追っていけば、そのプログラムが果たす目的を容易に導き出せるということです。そしてこの考え方は不正なプログラム、例えばマルウェアの解析に応用できそうです。

## APIコール間のデータ依存を用いたマルウェア解析



↑ 機能に無関係なAPIを中継し偽装

マルウェアが利用する不正なAPIを検知するのは困難に見えます…



↑ 依存関係に注目することでAPIのシャッフル、挿入にも対応可能に

しかし不正なAPI同士の連携を見出すことができればマルウェアの検知も容易になるかもしれませんね。

プログラムを組むにあたって重要な役割を果たすAPIは当然ながらセキュリティとも深い関わりがあるとのことでした。以上、幸せでいたいなら安全にAPIを叩こうでした。

# 自分に合ったデバイスは何だろう？

学部/回生	情報理工学部・SN 1回生
名前	sakura

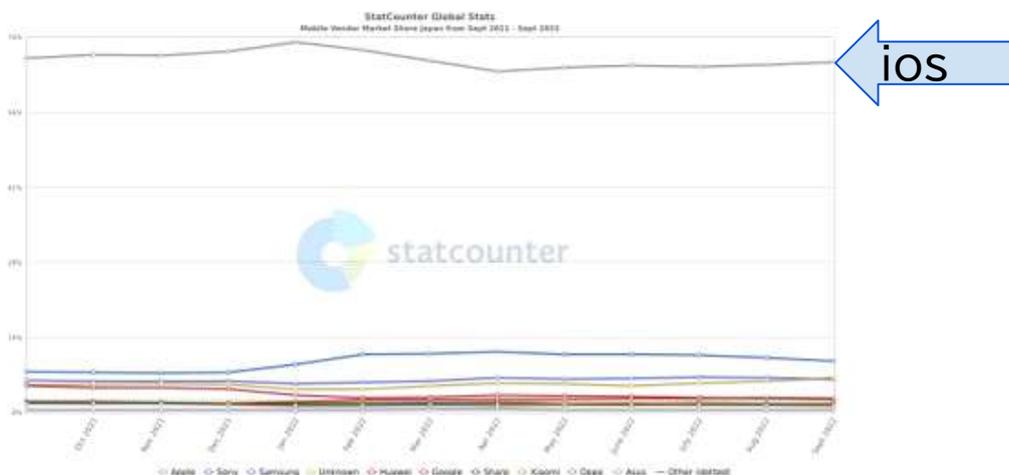


## 1:自己紹介

初めまして。情報理工学部1年生のsakuraです。私は、まだプログラミングなどの技術が乏しいので、今回は私が日常的に使っているApple製品について紹介していこうと思います。そして、特に私が「買って損はない」と思った商品を紹介したいと思います。

## 2:日本での普及率

日本ではAppleの製品を多くの人が使っていると思います。ある情報記事によると日本でのiOS普及率は半分を超えているそうです。



## 3:主なApple製品

皆さんはApple製品と聞いて、何を思い浮かべますか？

- iPhone
- iPad
- Apple Watch
- Appleペンシル
- AirPods
- Mac...ext



代表的なものと言われるとこんな感じでしょうか！

## 4:メリット

では、ここからは実際私が使っていて感じたメリットを簡単に上げていきたいと思います。

①1つ目はなんといってもAirdropの機能です。この機能は他の人との資料や写真の共有がスムーズにでき、特に大学の課題を友達とやる時は便利です。

②2つ目は、AppleIDの使用です。すべてのデバイスで同じIDを使用することでデータを共有することが可能です。

## 5:デメリット

①1つ目は、Apple製品以外との連動がうまくいかないことです。実際私は、AirPodsではなく他の製品を使用しているのですが少し設定ができない部分があり不便な部分があります。

②2つ目は、ひとつひとつの単価が高いです。全てをAppleの製品でそろえようと思うと相当な金額になります。

## 6:まとめ

私がいくつか挙げたメリットやデメリットは個人的に感じたことなので他にも多くの機能などがあると思います。また、私が今回メリットに挙げていることは少し技術が必要ですが他のサイトやアプリを使うと代替出来ることも多々あると思います。なので、自分が持ち合わせている技術と相談しているんなデバイスを試してみるのもいいかもしれません。ちなみに私ももう少し技術をつけて色々なデバイスに触れていきたいと思っています。(私がApple製品を使っている理由はみんなが使っていてうらやましかったからです…(笑))皆さんも自分に合ったデバイスをぜひ追求してみてください！

最後に:私はApple製品の中だとAppleペンシルとiPadの組み合わせを激押しします!(大学の授業には便利ですよ)



# プログラミング練習

学部/回生

情報理工学部・SAコース1回生

名前

CHOI YOUCHAN



プログラミングの練習をするためには自分のパソコンにプログラミングの環境を直接作った方がいい。しかし、ここではそれより便利な方法でプログラミングの練習ができる方法を紹介する。

## 練習のサイトの紹介

ここで私が紹介することはgoormというサイトである。このサイトでは図1のように自分がしたいと思っているプログラムを選び、そのプログラム言語の練習ができる。また、その言語のバージョンも自分で選べるようになっている。自分が練習したい言語を選ぶと図3のようにコンテナというものが作られる。そのコンテナを実行すると図2のような仮想のプログラミング環境が作られる。

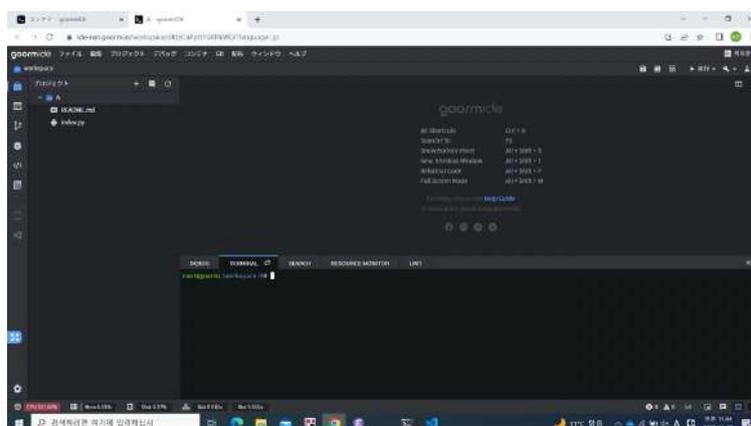


図 1

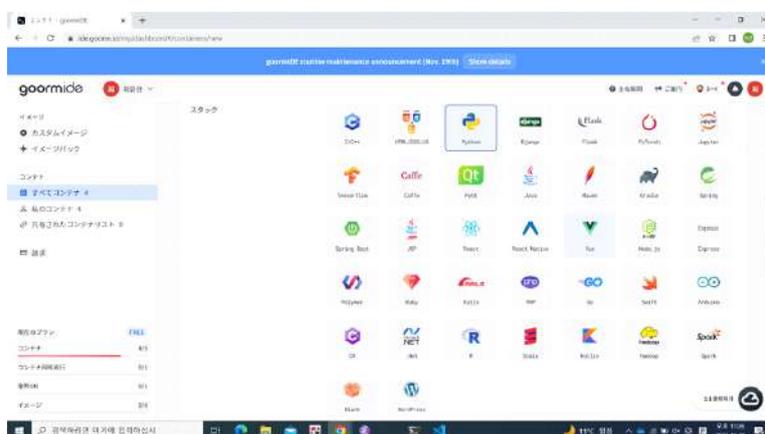


図 2

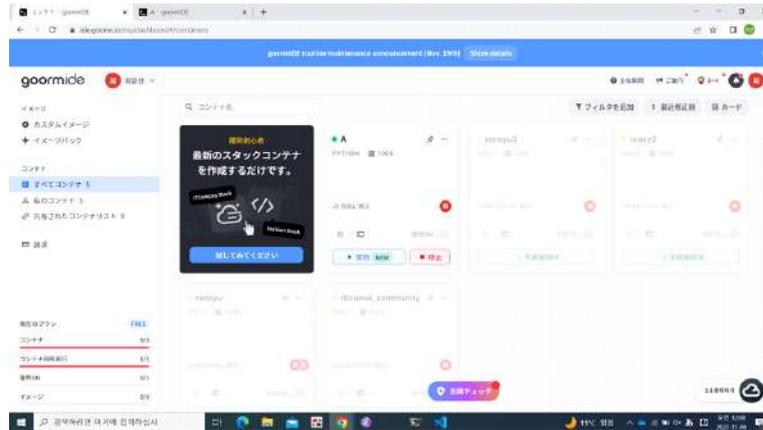


図 3

## 仮想プログラミングの必要理由

上述した通りプログラミングの練習というのはパソコンに直接環境を作った後すればいい。しかし、仮想プログラミング空間は環境を作れない人には必要になる。例えば、私のように軍隊に行く人はその中では自分のパソコンがなくてプログラミングの練習のための環境を作ることができない。つまり、個人用のパソコンではなく公共のパソコンを使ってプログラミングの練習をしなければならないときこのような仮想プログラミング空間が必要になる。

# セキュアコーディング

学部/回生

情報理工・SN 1回生

名前

Woohyeon



この会誌では脆弱性であるバッファオーバーフローと、その防ぎ方であるセキュアコーディングについて説明する。

## 概念の説明

オーバーフローとは情報が溢れることである。

バッファオーバーフローとはバッファという空間に入れられないほどの情報を入れたため、情報が溢れることであり、オーバーフローの一種である。

セキュアコーディングとはハッキングなどの攻撃の原因になりそうな脆弱性をなくすことで、より安全なプログラムを開発する技術である。

## バッファオーバーフローの問題

<図1>を見ると、gets関数を用いて文章が入力されている。ユーザーが開発者の思い通りに10文字以内の入力すればこのコードは何の問題も起こらない。

しかし、10文字を超えるとどうなるだろうか。

gets関数は文字数を気にしないまま入力した文章全てをarr\_に書き写そうとするため、10文字を超えたらarr\_という変数ではなく、違うところに書く。

<図2>は、スタックというものであり、スタックの中に、変数の内容物が保存される。この図の変数1が<図1>のarr\_である。そして、arr\_の容量を超えた文字は<図2>の変数1の下に書かれることになる。

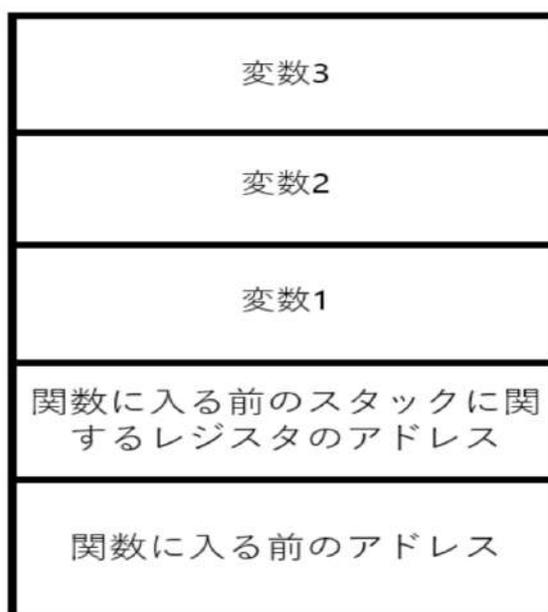
```
3 #include <stdio.h>
4
5 int main(void)
6 {
7     char arr_[10];
8
9     gets(arr_);
10    puts(arr_);
11    return 1;
12 }
```

<図1>

これがどんな問題を起こすかという、関数に入る前のアドレスというのは関数の活動が終わった時に帰るところであり、その情報を改ざすることで、パソコンが他の所に活かせることが可能になる。

## どう解決するか

ここまでの内容で問題になったのは、ユーザーの入力が開発者の意図とは違う入力であったということである。つまり、開発者が思った通りに10文字だけを入力させたら解決ができる。その方法は色々あるが、getsの欠陥を補完したgets\_s関数を使うことで解決できる。gets\_s関数は文字を入力されるとき、指定したサイズを超えると、エラーをはきながらプログラムを終了させる機能を持っている。そのため、悪用の余地がなくなる。



<図2>

```
3 #include <stdio.h>
4
5 int main(void)
6 {
7     char arr_[10];
8
9     gets_s(arr_, 10);
10    puts(arr_);
11    return 1;
12 }
```

<図3>

# ARM ベースのパソコン

学部/回生	生命科学部・生命情報 1回生
名前	Reiraku



ARMアーキテクチャは、縮小命令セット (RISC)を使い、ARMアーキテクチャを採用しているCPUは小型で低消費電力なCPUで、性能が普段よく使っているIntel x86より少し低いが、電力の消費が減らせるため、昔からスマホなどによく使われている。近年、性能の向上に従い、Apple M1などの高性能製品も増えた。



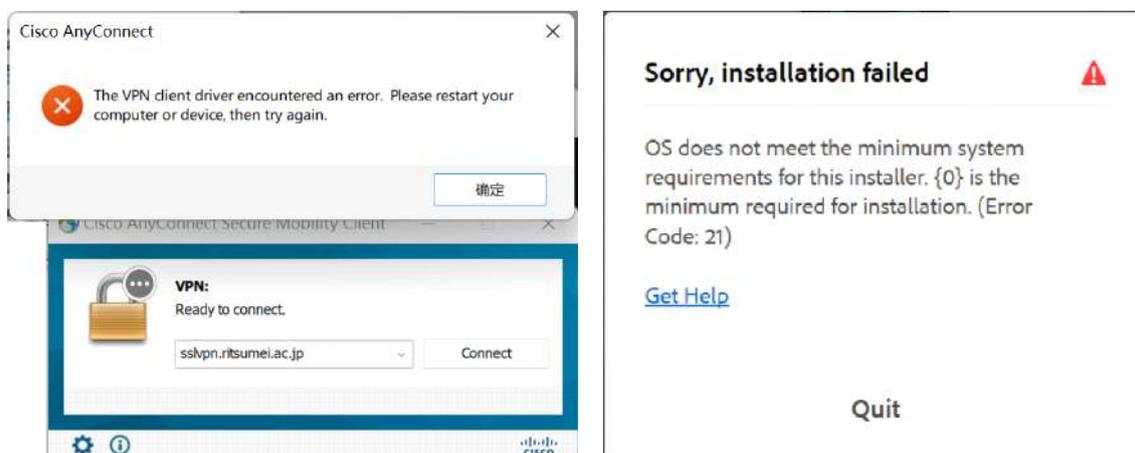
一方、近年ARM搭載したPCが増えて、多くの方は次のPCがARMに切り替えるかどうか迷っている学生に適しているかどうかを検証したいと思う。研究のために、自分も製品を購入して、テストを行い、実用性を検証した。



ARMの最も優れる点は、小型で省エネであり、バッテリー駆動時間がIntel CPU搭載のPCよりすごく長い。また、多くの製品がファンがなく、静かなものである。でも、性能がやや低く、アプリや拡張ボードの互換性が低いかも。



ARMとIntel x86は、命令セットは違うから、現行のx86ソフトを実行するとき、システム内のバイナリ変換(Binary translation)が必要である。しかし、バイナリ変換は完璧なものではない。ハードウェアドライバが対応できなかったり、ソフトウェアのバグが出たりすることが多く、性能もネイティブ実行より低下の場合が多い



だから、ARMの互換性がまだ弱いため、ARMベースのPCは大学生活に適していないかもしれない(特に理系の学生)。あまり専門的分野のソフトウェアを利用していない場合は、2台目パソコンとして購入することを検討できますが、特定のソフトウェアを使用するクラスを受講している場合は、ARMでのソフトウェアのサポートがまだ不足している可能性がある。

しかし、ARMの実行効率は高く、潜在力が大きい。ミライを創るのは、私たちである。不足があるならば、だんだん向上させることができる。みんなも一緒にしましょう！

# 自宅VPNを構築しよう

学部/回生

情報理工学部知能情報コース2回生

名前

naoya



こんにちは、naoyaです。

今回は自宅のネットワークを外部から接続する方法を紹介します。

皆さんは学校や外にいるときに、自宅のPCへアクセス・操作したいと思ったことはありませんか？私は大学でディープラーニングなど重いプログラムを動かしたいと思ってもノートPCだとスペック不足で出来ず、自宅のデスクトップPCを使いたいと思ったことが何度もありました。そこで自宅でVPNを構築して外部のネットワークから自宅のネットワークへ接続出来るようにしました。

今回は自宅のPCをVPNサーバにするために SoftEther VPN というソフトウェアを使いました。SoftEther VPN は筑波大学が開発、配布をしているソフトで無償で誰でも利用できます。

## サーバ構築手順

### 1. 公式ホームページからダウンロード&インストール

- a. 公式サイトへアクセス(<https://ja.softether.org>)
- b. コンポーネントは SoftEther VPN Server 、プラットフォームは対応するOSを選択してダウンロード

### 2. VPNServerを起動

- a. 管理マネージャからlocalhostへ接続
- b. パスワードが求められるが入力せずに進む
- c. パスワードの設定

### 3. VPN設定

- a. リモートアクセスVPNサーバを選択
- b. 仮想HUB名の設定
- c. ダイナミックDNSホスト名の設定
- d. 接続方式設定(今回はVPN Azure方式で接続した)

## 4. ユーザ設定

a. ユーザ名、本名を設定

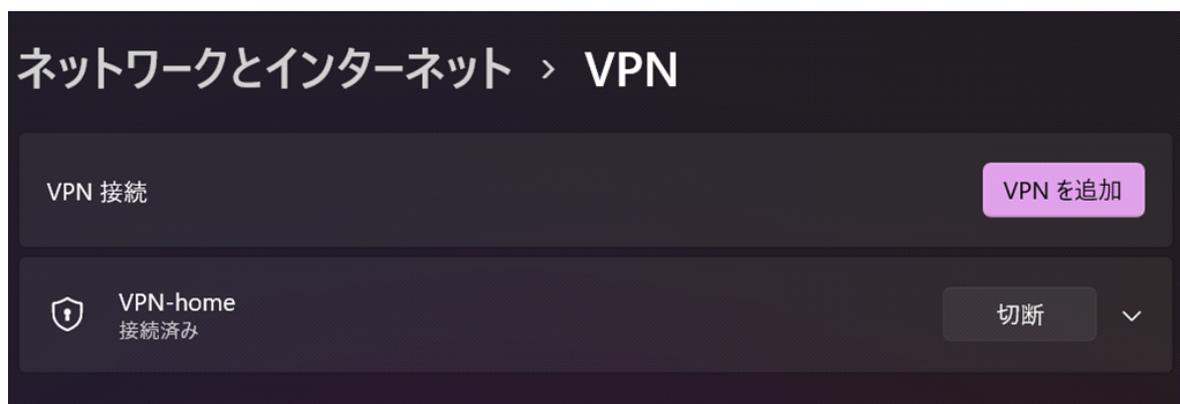
b. 認証方式をパスワード認証にしてパスワードを設定

### 実際に接続してみる

接続設定(Windows10, 11の場合)

- 接続名に適当な名前を入力
- サーバ名へ接続方式に合ったDNSホスト名を入力  
(今回はVPN Azure方式で接続したためホスト名+.azure.net)
- 設定したユーザメイトパスワードを入力

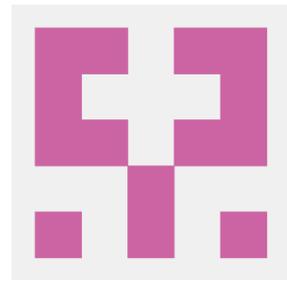
接続完了



最後に

VPNを建てたことによって外から自宅のネットワークへ接続できるようになりました。ただこれだけでは自宅のPCを常に起動する必要があり、電気代やPCに掛かる負荷が多くなってしまいます。そこで将来的には別でVPNサーバを建てて、メインのPCを起動して操作できるようにしたいです。

<h1>デザインパターン</h1>	
学部/回生	理工学部 機械工学科 2回生
名前	河合賢伸



今回はデザインパターンについて紹介しようと思います。OSSや会社の良いコードはどのような考えのもとコーディングされているのか気になって調べていたところ、どうやらデザインパターンというものがあるらしく

興味深い内容だったので、今回取り上げることにしました。ただ、デザインパターンは20年以上前から存在するもので、現在のアプリ開発では特に意識する必要はなかったり、フレームワークが標準でデザインパターンの一部を取り入れていたりします。なので、あくまで一つの面白いトピックとして共有させていただきます。

## デザインパターンとは

- ・コードのいい書き方集みたいな
- ・全23パターン
- ・ギャング・オブ・フォーという4人の研究者が出した書籍によって「デザインパターン」として世に認知されていった

### メリット

- ・再利用性
- ・共通言語
- ・リーディング力



ここは〇〇パターンでいきましょう。

いや、~~なので、ここは△△パターンがいいと思います。



デザインパターンを学ぶメリットは再利用性、共通言語、リーディング力の3つがあると思います。共通言語というのは↑のように言葉にしづらい設計に〇〇パターンというふうに簡潔に開発メンバーとコミュニケーションができるようになります。また、会社やOSSのコードを読む際もパターンを知っていると設計をある程度予想できるためリーディング力も培われます。自分はデザインパターンを勉強することにより今まで新たな気づきやなぜこういう書き方をしているのかも腑に落ちるようになったと感じています。

今回はデザインパターンの一つを例に倣って紹介しようと思います。  
月次でプロジェクトの進捗レポートを出力する機能を作るケースを考える  
このようなケースでは Template Methodパターンを採用する  
改善前

改善前のコードでは、新しくフォーマットを追加する際、フォーマットの数だけif文が増えてしまいます。また、変化する部分(出力する内容)と変化しない部分(出力の一連の流れ)が同じクラス内に存在しているため、メンテナンスがしづらいというデメリットがあります。

Template Methodパターンを使い次のように改善されると、変化する部分と変化しない部分が分離されメンテナンスに強くなりフォーマットが増えても最小限のコーディングで実装ができるようになります。

```
1 class Report
2   def initialize(format: :plane, text: '順調です')
3     @title = '月次報告'
4     @text = text
5     @format = format
6   end
7
8   def output_report
9     output_start
10    output_title
11    output_body
12    output_end
13  end
14
15  def output_start
16    puts '<html>' if @format == :html
17  end
18
19  def output_title
20    if @format == :plane
21      puts @title
22    else
23      puts '<head>'
24      puts "&<title>#{@title}</title>"
25      puts '</head>'
26    end
27  end
28
29  def output_body
30    if @format == :plane
31      puts @text
32    else
33      puts '<body>'
34      puts "&<p>#{@text}</p>"
35      puts '</body>'
36    end
37  end
38
39  def output_end
40    puts '</html>' if @format == :html
41  end
42 end
43
44
```

改善後

```
1 class Report
2   def initialize(text: '順調です')
3     @title = '月次報告'
4     @text = text
5   end
6
7   def output_report
8     output_start
9     output_title
10    output_body
11    output_end
12  end
13
14  def output_start; end
15
16  def output_end; end
17 end
18
19 class HTMLReport < Report
20   def output_start
21     puts '<html>'
22   end
23
24   def output_title
25     puts '<head>'
26     puts "&<title>#{@title}</title>"
27     puts '</head>'
28   end
29
30   def output_body
31     puts '<body>'
32     puts "&<p>#{@text}</p>"
33     puts '</body>'
34   end
35
36   def output_end
37     puts '</html>'
38   end
39 end
40
41 class PlainReport < Report
42   def output_title
43     puts @title
44   end
45
46   def output_body
47     puts @text
48   end
49 end
50
```



テキストマイニングはRやPythonなどを使っていますが今回はPythonを使った方法を紹介します

まず、前提としてMeCaBと辞書をPythonにインストールしてもらう必要があります。MeCabとはオープンソースで利用できる形態素解析(文章やフレーズを意味を持つ最小単位に分割する)ソフトウェアです。

例えばtwitterなどのSNS文章のテキストマイニングをする場合にはtwitterAPIの承認をうける必要があります

そのあとに検索キーワードの設定やデータフレームの作成を行い、そののちにテキストマイニングを行うコードを書いて分析を行っていきます。

最後にwordcloudを用いて解析したデータを可視化します。

実際のコードの一部はこのようになります

```
word_list=[]
while node:
    word_type = node.feature.split(',')[0]
    if word_type in ["名詞", '代名詞']:
        word_list.append(node.surface)
    node=node.next
word_chain=' '.join(word_list)

c=collections.Counter(word_list)
font_path='/System/Library/Fonts/ヒラギノ明朝 ProN.ttc'
words = ['https','t','co','自民','し','w','そう','ない','いる','する']
result = WordCloud(width=800, height=600, background_color='white',
                    font_path=font_path,regexp=r"[\w]+",
                    stopwords=words).generate(word_chain)
result.to_file("./wordcloud_sample1.png")
print(c.most_common(20))
fig = plt.subplots(figsize=(8, 10))
```

詳細はこちらのURLからサイトにとんでいただければ細部まで詳しく書かれたコードが公開されています。

<https://python-man.club/python-twitter-textmining/>

興味がある方はもっと詳しくテキストマイニングについて調べて自分でコードを書いて動かしてもらうことをお勧めします

# hello.cのその後

学部/回生

情報理工学部 SN 2回生

名前

hane



みなさんこんにちはhaneです

立命館大学 情報理工学部のシステムアーキテクトコースとセキュリティネットワークコースの2コース(?)では一回生の秋学期にC言語を授業で学びます。その一回目の授業で、プログラミングをはじめた人なら誰もがするであろうHelloWorldをC言語でやりました。

2回生になるとデータパスやOSなど低レイヤーに関する授業が多くなりました。その影響もあってかHelloWorldがどうやって動いているのか気になったので調べました。書いてから思ったのですが、最後の方が本編な気もしないので最後だけでも読んでくれると嬉しいです。

下がおなじみのプログラムです

```
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
```

## プリプロセス

はじめにプリプロセッサがおまじないとして習った”#”で始まるディレクティブと呼ばれるものに従ってhello.cを変換します。今回の場合は1行目の#include <stdio.h>を見て「システム・ヘッダ・ファイルのstdio.hを読み込んで！」と解釈し、hello.iのプログラムテキストに変換されます。

## コンパイル

つづいて、hello.iをより機械語に近づけるためにアセンブリ言語で書かれたテキストファイルhello.sに変換します。

## アセンブリ

次に、hello.sを機械語に翻訳します。どうやらここでできたファイルをバイナリファイルと呼ぶみたいです。

## リンク

そしてリンカがprintf関数をコンパイルしたprintf.oというオブジェクトファイルと合体させて実行可能オブジェクトファイルであるhelloができます。

最後にメモリにロードしてHelloWorldすることができるようになりました！

## おわりに

私が会誌を書いていたら隣にいた先輩から「ハロー“Hello, World” OSと標準ライブラリのシゴトとしくみ」と「数学ガール」という本を勧められました。前者はHelloWorldがどのように動いているのかを詳しく調べている本らしいのでこの会誌を読んでくださった方もぜひ読んでみてください。そして数学ガールという本はタイトルの通り、三人の女子高生と一人の男子高校生が数学の話をする本みたいなのですが全部で6巻あってちゃんと厚みがありました。とくに4巻がアルゴリズムの話について書いてあるみたいなのでぜひみなさんも読んでみてください。話によると、どうやらジャンルは恋愛小説だと聞いたので、てっきり数学を通じて男子高校生の恋愛模様を描いているのかと思いきや数学と恋愛するみたいです。RCCには数学と恋愛する人はいなかったと思うのですが、RCCの人たちはどこか抜けてて面白かったり、奇抜な人がいたり、最近では韓国や中国の人と話をしたり毎週のように食事に行ったりしていてとても面白いのでぜひ部室の見学に来たり、説明会に参加してくれるととても嬉しいです。

# 無線でつないでみよう!

学部/回生

情報理工学部・知能情報コース2回生

名前

古川聡悟(FkSg)



みなさん、こんにちは情理、知能2回の古川です。今日は簡単に、「コンピュータ」と「コンピュータ」で値やデータの受け渡しをする通信方法を1つ取り上げたいと思います。

その中でも今回は、同じLAN内にある2つのコンピュータの間でsocket通信をしようと思います。

socket通信には、ICP(プロセス間)通信の1つ(プロセス = 稼働中のプログラム)です。なので例えば、「1つのコンピュータ内で完結する2つのプロセスの通信」といったものもsocket通信と呼びます。

片方の「IPアドレスを定めて接続を待つ方」を「Server」と言いもう片方の「そのIPアドレスに繋がりにいく方」を「Client」と呼びます。

## [画像1] Client側のPythonコード

```
1 #TechAcademyマガジン, Pythonでソケット通信を行う方法【初心者向け】, https://techaacademy.jp/magazine/19147, 2022年3月3日.
2
3 #クライアント側
4 import socket
5
6 PORT = 50000 #TCP, UDPに関わらず1000よりも小さな値を指定しない方が良い。(メールなど他の通信で既に使用されているため)
7 BUFFER_SIZE = 1024 #やりとりできる最大のデータの大きさ
8
9 #AF_INETと設定することで192.168.11.**の4つの種類の値でIPアドレスを指定できる。
10 #AF_INET6では6つの値でIPアドレスを指定できる。
11
12 #SOCK_STREAMは、バイトストリーム(バイト単位=8bitでのデータの連続)でデータを通信する。
13 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
14     s.connect(('192.168.11.**', PORT)) #クライアント側からサーバー側へ繋がりに行くため、サーバーのIPを第一引数に記入する
15     data = input('Please input > ') #これは、接続のテストとして文字を入力し、サーバー側で繋がったかを確認する。
16
17     s.send(data.encode()) #入力した文字列pythonのbyte形式に変換しサーバー側から送信する。
18     print(s.recv(BUFFER_SIZE).decode()) #受信したデータをpythonのbyte形式に変換し、文字列として表示する。
```

Client側の流れ: 13行目:(1)ソケットの生成、14行目:(2)ServerのIPへの接続、17行目:(3)入力文字列のServerへの送信、

18行目:(4)Serverからの返信の受信 (コードは上の画像を参照)

Server側の流れ:

13行目:(1)ソケットの生成。

14行目:(2)IPとPortを設定(バインドという) 15行目:(3)接続待ち

18行目:(4)Clientからの接続の判定

[(5)と(6)を接続中に繰り返す]

22行目:(5)Clientからのデータの取得

26~28行目:(6)条件満了時でのClientからのデータに対する送信

31行目:(7)ソケットの閉鎖と終了 (コードは下の画像を参照すること)

## [画像2 Server側のPythonファイル

```
1 #TechAcademy マガジン, Pythonでソケット通信を行う方法【初心者向け】, https://techacademy.jp/magazine/19147, 2022年3月3日.
2
3 #サーバー側
4 import socket
5
6 PORT = 50000 #TCP, UDPに関わらず1000よりも小さな値を指定しない方が良い。(メールなどの通信で既に使用されているため)
7 BUFFER_SIZE = 1024 #やりとりできる最大のデータの大きさ
8
9 #AF_INETと設定することで192.168.11.**の4つの種類の値でIPアドレスを指定できる。
10 #AF_INET6では6つの値でIPアドレスを指定できる。
11
12 #SOCK_STREAMは、バイトストリーム(バイト単位=8bitでのデータの一続き)でデータを通信する。
13 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s: #ソケットを生成し、sとして扱う。
14     s.bind(('192.168.11.*', PORT)) #ソケットにIPアドレスとポート番号を設定する(このことをバインドという)
15     s.listen() #ソケットを接続待ちの状態にする
16
17     while True:
18         (connection, client) = s.accept() #クライアント側から通信する動きがあるまでプログラムを停止し、繋がったときにプログラムを再開する。
19
20         try:
21             print('Client connected', client)
22             data = connection.recv(BUFFER_SIZE) #データを受け取る。
23
24             print(str(data))
25
26             if str(data) == "b'Hello'": #クライアント側からの入力文字列が"Hello"になったときに
27                 data2 = "I'm fine thank you!" #サーバー側から"I'm fine thank you!"の文字列を送信する。
28                 connection.send(data2.encode())
29
30         finally:
31             connection.close() #ソケットを閉じて、通信接続を終了する。
```

UDP通信では、ServerからClientに送信に失敗してもデータを送信し続けることがリアルタイムに向いているという利点がある一方で、正しく送信できたかがServer側からは分からないといった欠点があります。

使用した環境 コンピュータ2台(OS:Windows10 Ubuntu18など, 言語:Python3.8, ライブラリ:PySocket で同じLANに接続済)

### 参考文献

[1] Quora, Toshiaki Takada, めちゃめちゃ分かりやすく言うと、ソケット通信とは何ですか?, <https://jp.quora.com/%E3%82%81%E3%81%A1%E3%82%83%E3%82%81%E3%81%A1%E3%82%83%E5%88%86%E3%81%8B%E3%82%8A%E3%82%84%E3%81%99%E3%81%8F%E8%A8%80%E3%81%86%E3%81%A8-%E3%82%BD%E3%82%B1%E3%83%83%E3%83%88%E9%80%9A%E4%BF%A1%E3%81%A8%E3%81%AF>, 最終閲覧日2022年11月4日。

[2] TechAcademy, マガジン, Pythonでソケット通信を行う方法【初心者向け】, <https://techacademy.jp/magazine/19147>, 最終閲覧日2022年11月5日。

[3] Qiita, Socket()とかBind()とかを理解する, <https://qiita.com/Michinosuke/items/0778a5344bdf81488114>, 最終閲覧日2022年11月5日。

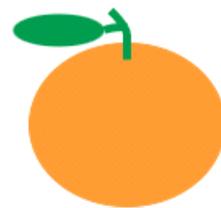
# 1Rでラックサーバー

学部/回生

理工学部 ロボティクス 2回生

名前

daiki



こんにちは自宅サーバー系Youtuberの動画を見ていたら自分も欲しくなってラックサーバーとL2スイッチ、業務用ルーターを買ってしまったんです。

このコラムではその中で自宅にあるのが一番場違いなラックサーバーをどのようにすれば1Rでも稼働させることができるのかについて紹介したいと思います。

はじめに使っているラックサーバーがHPE ProLiant DL120Gen9という1Uのものです。最大6個の2.5inchベイでRAID組み放題、リモートでサーバーの状況を管理できるツール付きというラックサーバー入門にはちょうどいい感じでESXiを載せています。



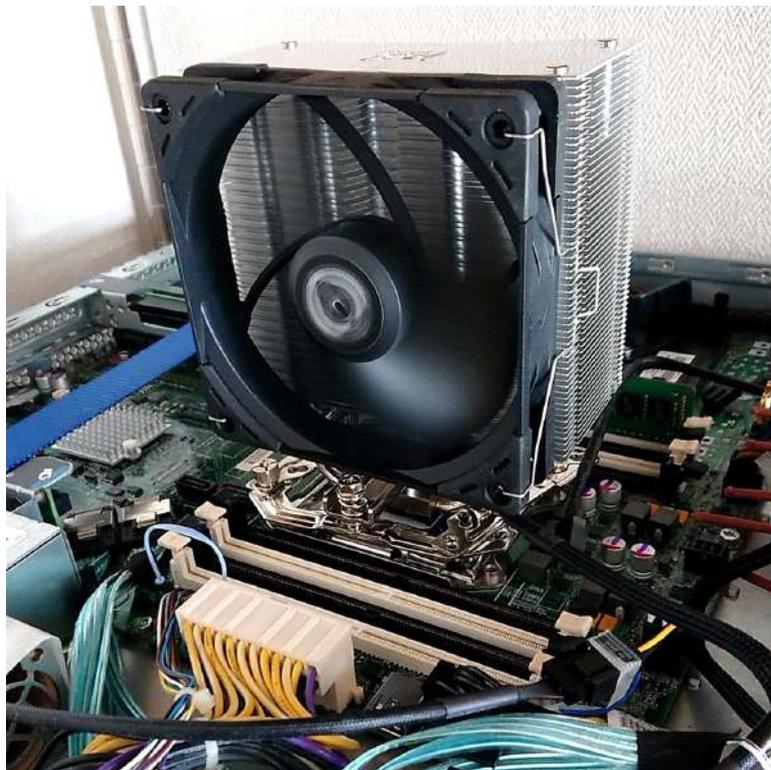
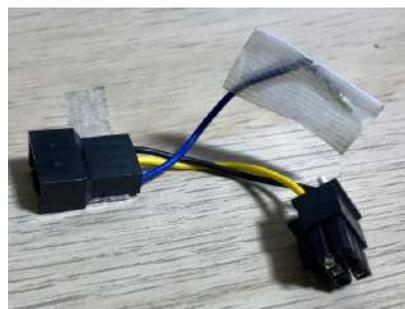
これを1Rで稼働させるうえで一番の問題となるのはこの冷却ファンの音です。Youtubeで「server startup」と検索して出てくる動画を視聴して見るとわかると思うのですが掃除機以上のタチの悪い騒音を出します。これを静かにさせるのが今回の一番の仕事です。私が最初に試したのは右の爆音ファン4つをNoctuaのNF-A4x20 PWMに置き換える方法でした。



この方法が一番シンプルで

簡単な方法でしたが、コンシューマー向けのファンコネクタ形状とHPEのコネクタ形状が異なるという大きな問題がありました。つなぐところを間違えるとショートし最悪燃えるのでなかなかやる気が起きませんでした。探してみるとインターネットに同じようなことをしている人のブログがあり、結線仕様を知ることができたので作ってみました(次ページ)

これを使って動かしてみたところ全然風量が足りておらず温度が上がりすぎてシャットダウンしてしまいました。なのでこの方法は諦めて別の方法を探しました。次に思いついたのはCPUに直接サイドフロー型のCPUクーラーをくっつける方法で早速買ってやってみました(下の写真)



若干無理やりくっつけた感はありますが、うまくいって負荷最大でも問題なく動き1Rでも常時動かせるようになりました！

ちなみに、、、

HPE ProLiant DL120 Gen9	¥16,000
Noctua 40mm Fan	¥8,000(無駄だった)
強化用CPU	¥5,000
サイドフローCPUクーラー	¥4,000

でした。

# 仮想技術の攻撃手法

学部/回生

情報理工学部 3回生

名前

lufe



近年コンテナ技術が進んでいるが、仮想マシンについても同様に使用されている。BlackHat2021ではVMエスケープの攻撃手法とKVMによる防御手法が発表された。近年ではクラウド化も進んでいるため、VMエスケープについては重大視されているものである。よってここではVMのセキュリティについてざっくりと概要を記載する。

計算機仮想化ではデバイスドライバーなどのハードウェアについてはしっかりとした分離はできないため、I/OデバイスやNICなどに対しては直接操作ができるとおもいいと考えている。そのため現在でハードウェアについての脆弱性やKVMなどの仮想化自体に脆弱性がある場合とVMM(仮想マシンモニター)に脆弱性がある場合にゲストOSからホストOSへの権限昇格が可能である。

ゲストOSからホストOSへの新規プロセスの作成からの権限昇格が存在する。

最近でもVMWAREの脆弱性が見つかったので今後ともチェックが必要である。

XenやVirtualbox、OpenStackはOSSであるため、コードを見れるのでそこからの脆弱性が多く発見されている。

もしかしたら、OSSではないHypervisorの製品にも同様の脆弱性があるって修正されていない可能性もある。

これからの脆弱性情報に期待と、不安を抱えつつ、今後のセキュリティ動向について注意深くみていきたい。

FROM <https://www.slideshare.net/suzaki/cloud-computingvm14jan2010>

<https://www.youtube.com/watch?v=ue2KW6oy7A8>

<https://engineers.ffri.jp/entry/2021/10/07/183613>

# i3をいい感じにする

学部/回生

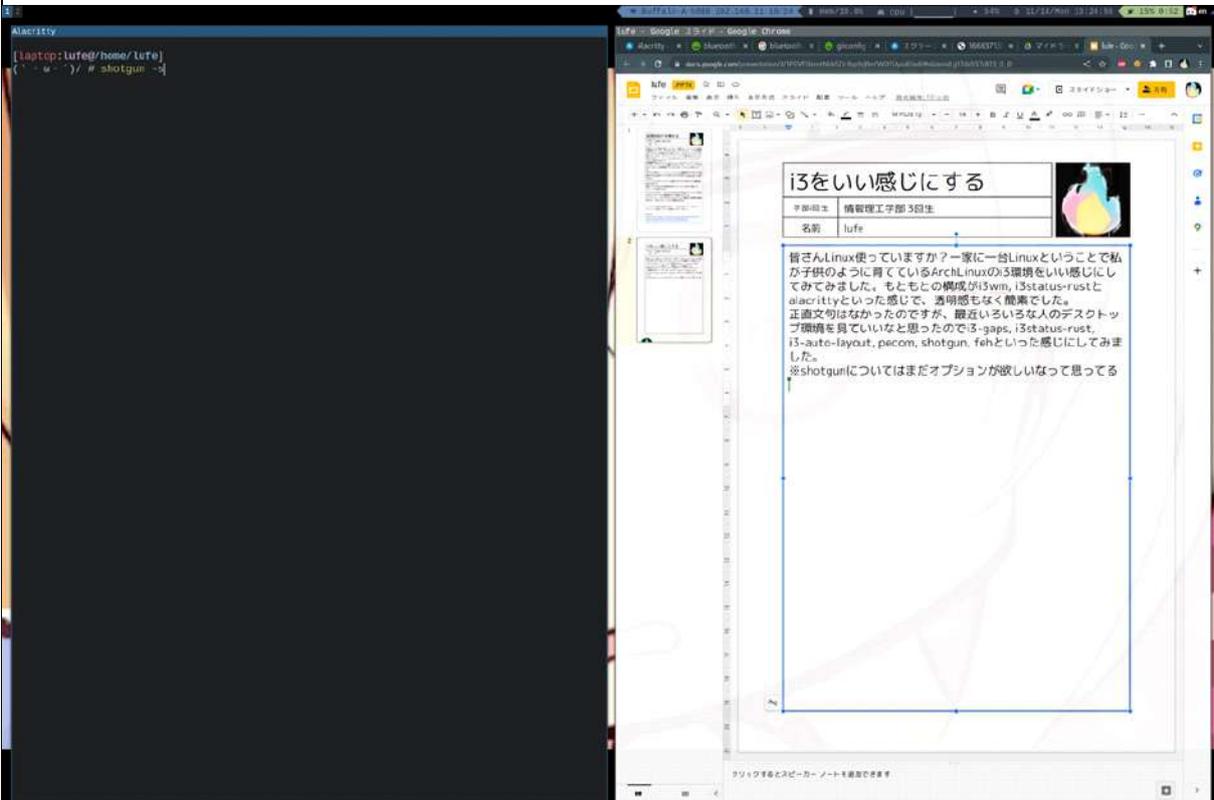
情報理工学部 3回生

名前

lufe



皆さんLinux使っていますか？一家に一台Linuxということで私が子供のように育てているArchLinuxのi3環境をいい感じにしてみました。もともとの構成がi3wm, i3status-rustとalacrittyといった感じで、透明感もなく簡素でした。正直文句はなかったのですが、最近いろいろな人のデスクトップ環境を見ていいなと思ったのでi3-gaps, i3status-rust, i3-auto-layout, pecom, shotgun, fehといった感じにしてみました。  
※shotgunについてはまだオプションが欲しいなって思ってるまだ改善点はあるが、盆栽のようにゆっくりきれいにしていこうと思う。



# Wordは使いたくない！！

学部/回生

理工学部・3回生

名前

こま



Word・PowerPointを使いたくないので、VSCode+Markdownで書類やスライドを作ってみました。

全て無料で使えます！

**Markdown Preview Enhanced** - 最強に快適で拡張性の高いマークダウンのプレビュー拡張機能

リアルタイムで編集内容がプレビューできるのでとても良いです。

## 数式や

```
test.md
1 ちまめじんのえんしゅうりつこうしき
2
3 $$ \frac{1}{\pi} = \frac{2\sqrt{2}}{99^2} \sum_{n=0}^{\infty} \frac{(4n)!}{n!^4} \frac{26390n + 1103}{396^4(4n)} $$
4
```

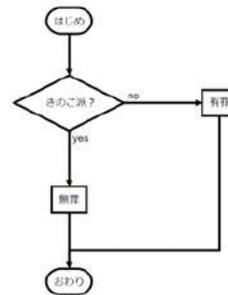
ちまめじんのえんしゅうりつこうしき

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{99^2} \sum_{n=0}^{\infty} \frac{(4n)!}{n!^4} \frac{26390n + 1103}{396^4}$$

## フローチャートや

```
test.md
1 フローチャート
2
3 ...flow
4 開始->start: はじめ
5 終了->end: おわり
6
7 条件->condition: きのこ袋?
8 処理1->operation: 煮詰
9 処理2->operation: 煮詰
10
11 開始->start
12 条件(yes)->処理1
13 条件(no)->処理2
14
15 処理1->終了
16 処理2->終了
17 ...
```

フローチャート



## コードブロックも

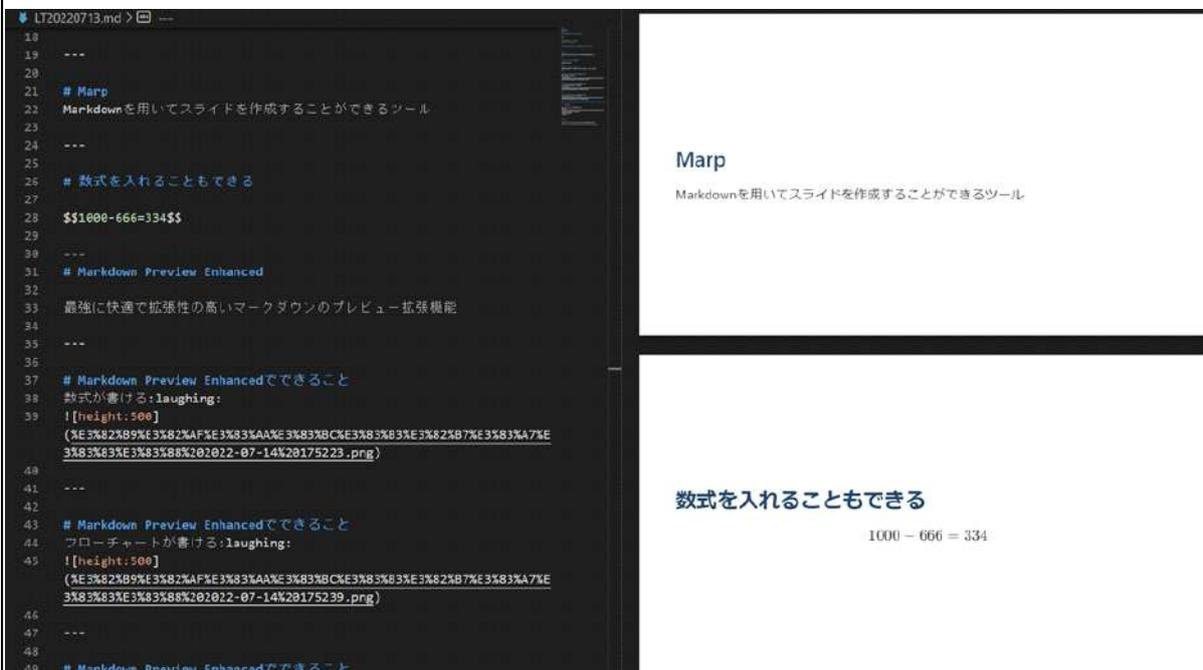
```
test.md
1 ...
2 #コードブロックも書けるよ!
3 for i in range(1, 101):
4     if i % 3 == 0 and i % 5 == 0:
5         print("FizzBuzz")
6     elif i % 3 == 0:
7         print("fizz")
8     elif i % 5 == 0:
9         print("buzz")
10    else:
11        print(i)
12 ...
13
```

```
#コードブロックも書けるよ!
for i in range(1, 101):
    if i % 3 == 0 and i % 5 == 0:
        print("FizzBuzz")
    elif i % 3 == 0:
        print("fizz")
    elif i % 5 == 0:
        print("buzz")
    else:
        print(i)
```

このように書くことができます。

## Marp - Markdownを用いてスライドを作成することができるツール

Markdown Preview Enhancedと同じようにリアルタイムで編集内容がプレビューできます。



とても便利なのでぜひ使ってみてください！

(ちなみにMarkdown Preview Enhancedを入れてMarpと同時に使おうとするとプレビュー機能が干渉してうまくいきませんでした…

Marpを使いたいときにはMarkdown Preview Enhancedを無効にしておきましょう)

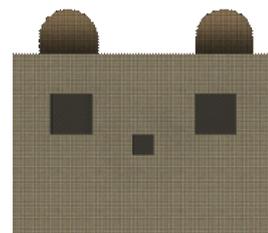
# Vimをもっと便利に

学部/回生

理工学部 物理科学科 三回生

名前

クマさん



初めましての方は初めまして。また会ったねという方はお久しぶりです。どうも、クマさんです。このコラムの執筆ももう三回目ということで、なんかもう三年たったのかと時の流れが早く感じるようになった今日この頃です。

ところで皆さんは普段の開発で定義検索や型定義、参照箇所の検索といった機能を使っていますでしょうか？

おそらく、使っている方が大半だとは思いますが、念のため「何それおいしいの」と思っている方向けに簡単に説明しますと、定義検索はその名通り、定義を検索する機能のことで、シンボル(変数名や関数名、クラス名のこと)の定義を調べる際に便利です。また、同じようなものに型定義の検索といった機能があり、こちらは変数の型の定義を調べる際に使われます。最後に参照箇所の検索はシンボルの使用箇所を検索する機能のことであり、仕様変更の際などに対象のクラスや関数がどこでどれだけ使用されているかなどを調べる際に非常に強力な味方となってくれます。

これらの機能はIntellijやVisual StudioなどのIDEからVSCodeといったテキストエディタには標準で備えられています。一方でVimやNeovimには当たり前ですが標準搭載ではありません。しかし、この便利機能たちをVimでも使いたいという方は多いはず。というわけで今回はこれらの機能をVimでも使えるようにしようというお話になります。

## プラグイン管理ツールの導入

今回の環境はLinuxとNeovimであるため、下記に書かれているコマンドは別の環境では違うコマンドになる可能性があることに注意してください。

詳しくはツールのドキュメントで確認してください。

さて、まずはVimにプラグインマネージャの中でも最も有名な物の一つであるvim-plug(<https://github.com/junegunn/vim-plug>)を導入するために下記のコマンドを実行しましょう。

```
sh -c 'curl -fLo "${XDG_DATA_HOME:-$HOME/.local/share}"/nvim/site/autoload/plug.vim --create-dirs \
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim'
```

何らかのエラーが出なければ成功です。

## プラグインを入れてみよう！

プラグイン管理ツールを導入したので実際にプラグインをVimに入れています。なお、今回使用するプラグインは下記に依存しているので先にインストールしておきましょう。

```
npm, fzf, ripgrep
```

依存関係を導入したらinit.vimにインストールしたいプラグインを記述します。

```
call plug#begin()
Plug 'neoclide/coc.nvim', {'branch': 'release'}
Plug 'junegunn/fzf', { 'do': { -> fzf#install() } }
call plug#end()
```

その後保存し、Vimの中で下記のコマンドを実行しましょう。

```
:PlugInstall
```

インストールしたプラグインは

- **coc-nvim**(<https://github.com/neoclide/coc.nvim>)
- **fzf**(<https://github.com/junegunn/fzf>)

の二つです。これらのプラグインの詳細について知りたい場合はそれぞれのドキュメントを確認してください。

さて、ここまで来たら残りの作業は一つだけです。  
先程インストールした**coc.nvim**の拡張機能をインストールします。

```
:CocInstall coc-json coc-tsserver coc-fzf-preview
```

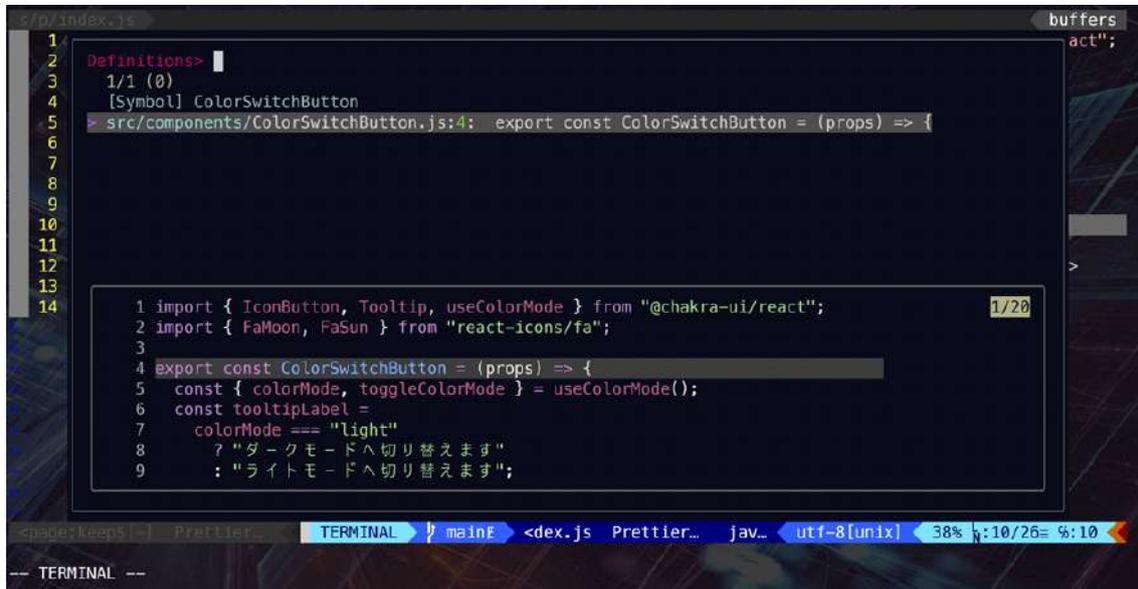
これで準備完了です。

次は実際に使ってみましょう。

# 使用方法

## 定義検索

```
:CocCommand fzf-preview.CocDefinition
```

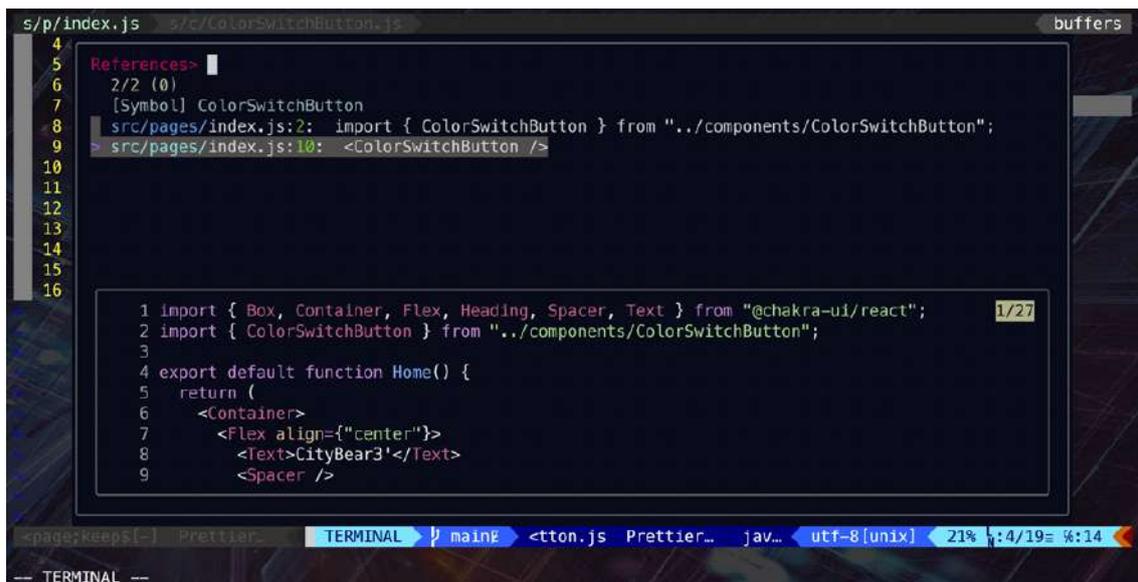


The screenshot shows the Coc Definition search interface. At the top, the command `:CocCommand fzf-preview.CocDefinition` is entered. The main window displays the search results for the definition of `ColorSwitchButton`. The results show the file path `src/components/ColorSwitchButton.js:4` and the definition code. The code is as follows:

```
1 import { IconButton, Tooltip, useColorMode } from "@chakra-ui/react";
2 import { FaMoon, FaSun } from "react-icons/fa";
3
4 export const ColorSwitchButton = (props) => {
5   const { colorMode, toggleColorMode } = useColorMode();
6   const tooltipLabel =
7     colorMode === "light"
8     ? "ダークモードへ切り替えます"
9     : "ライトモードへ切り替えます";
```

## 参照個所の検索

```
:CocCommand fzf-preview.CocReferences
```



The screenshot shows the Coc References search interface. At the top, the command `:CocCommand fzf-preview.CocReferences` is entered. The main window displays the search results for the references of `ColorSwitchButton`. The results show the file path `src/pages/index.js:2` and the reference code. The code is as follows:

```
1 import { Box, Container, Flex, Heading, Spacer, Text } from "@chakra-ui/react";
2 import { ColorSwitchButton } from "../components/ColorSwitchButton";
3
4 export default function Home() {
5   return (
6     <Container>
7       <Flex align={"center"}>
8         <Text>CityBear3</Text>
9         <Spacer />
```

## エラーリストの表示(おまけ)

```
:CocCommand fzf-preview.CocDiagnostics
```



The screenshot shows a code editor with a diagnostics window open. The diagnostics window displays three error messages:

- src/pages/index.js:10: Error JSX element 'ColorSwitchButton' has no corresponding closing tag.
- src/pages/index.js:16: Error '>' expected.
- src/pages/index.js:16: Error JSX element 'Box' has no corresponding closing tag.

Below the error messages, a code snippet is shown:

```
1 import { Box, Container, Flex, Heading, Spacer, Text } from "@chakra-ui/react";
2 import { ColorSwitchButton } from "../components/ColorSwitchButton";
3
4 export default function Home() {
5   return (
6     <Container>
7       <Flex align={"center"}>
8         <Text>CityBear3'</Text>
9         <Spacer />
10      </Flex>
11    </Container>
12  );
13 }
14
15
16
```

## まとめ

IntellijやVSCodeなどと比べるとセットアップに時間はかかりますが、性能としては結構いい勝負なのではと私は思っています。

なお、今回紹介しきれなかった機能は**fzf-preview**

(<https://github.com/yuki-yano/fzf-preview.vim>)

のドキュメントに記載されているので各自で確認してください。

また、**coc.nvim**はLSPの拡張機能を入れることができるので皆さんの好きな言語の補完も可能です。

ここまで長くなりましたが、お付き合いいただきありがとうございます。

これを機に皆さんの開発体験が向上すれば幸いです。

それではまた、来年会えたら会いましょう。

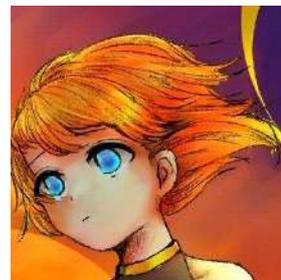
# Unityでのデータの扱い

学部/回生

情報理工学部・SNコース3回生

名前

TZ



Unityをご存じでしょうか？ゲームエンジンの一つとして、最近のモバイルゲームの大半はこれで作られております。今回はそのUnityでの話になります。

## 議題:データを扱いたい

RPGなどでは沢山のデータを扱うことと思います。最近RPGを作ったのですが、エネミーのステータス、使うスキル、プレイヤーが扱うスキル、アイテム、それらを表す文字列や数値や画像…。どう扱えばええんじやいという感じです。ということで、Unityでこれらのデータを扱う方法を見ていきます。

## 内容:データの扱い方

### ①ハードコーディング

サクサク行きます。やめましょう。

ハードコーディングというのは、スクリプトにデータとなる数値などをそのまま書くことですが、これはデータの数スクリプトが必要になります。スライムスクリプト、データ部分 `int attack = 5`…ってのが何個もある。えぐい。見にくい、変更がめんどくさい、無駄にスクリプトが増える

### ②SerializeField

これはさっきとあんまり変わりません。Unityでは変数にSerializeField属性をつけることでprivate変数でもInspectorから操作できるんですが、これ自体は便利です。下手にInspectorから操作したいからpublic変数〜とかするよりはよほどいいです。が、データを扱うのには向いてないかなと思います。構造的に前のハードコーディングと同じような、データの数だけスクリプトがあるって感じになると無駄ですし、じゃあ配列で管理すればいいというわけでもありません。文字列、数値、画像など多様で多くのデータを一挙に一つのデータとして扱いたい場合、SerializeFieldではきついものがあります。

### ③CSV,JSONなどをシリアライズ

いい感じの人はこれを使っている印象があります。

UnityはCSV,JSON,TXTなどいろんなファイルがTextAssetとして読み込めるので、読み込んで使うらしいです。(昔音楽ゲームの譜面データをtxtで作って読み込んだことがありましたが、単純な構成なのにだいぶ苦労したことがあって私は再チャレンジできてません…)

これのいいところは、CSVなどで扱うことで同じデータタイプのデータ(たとえばエネミーデータ一覧、アイテムデータ一覧)を一挙に見れることだと思います。また、マスターデータがUnityから切り分けられることで、アプリ配信をした際などにマスターデータの配信だけしたいのに

いちいちアプリの審査をする、なんてことをしなくてよくなります。サーバーにマスターデータを置いて、随時DLさせる方針にすればいいかららしいです。私はやったことはありません。(データをサーバーに置いといてとか高度なことできたことないので)

#### ④ScriptableObject

今回採用したものです。Unity公式のデータ管理機能？的なもので、データのベースとなるスクリプトを定義すると後は数値だけ違うけど同じ型というデータが何個も作れて参照できます。中身はYAML形式というもので書かれているらしいですがあまり意識することはないです。

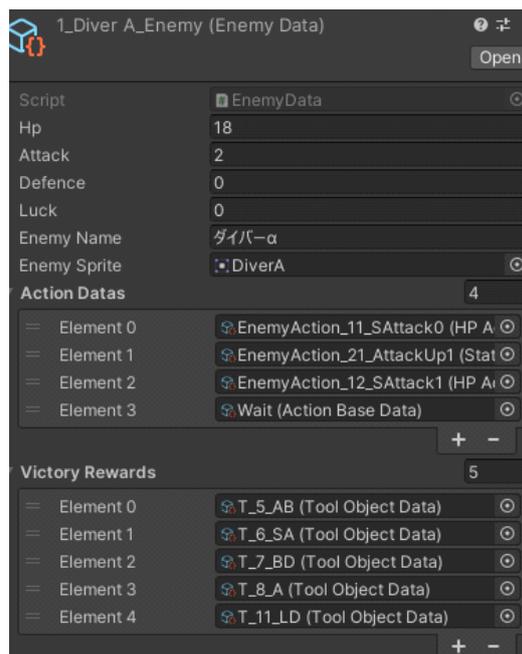
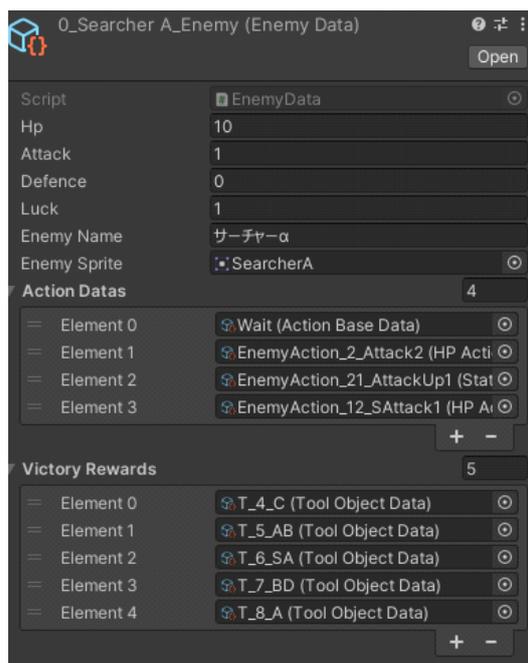
Enum形式がドロップダウンで選べる、画像なども入れられる

EnemyDataという名前で作ってあげれば中身がなんであろうとEnemyDataとして扱えるので差し替えも楽ちんというだけ。作るのがちょいめんどくさい(後輩曰く日が暮れる)のがちょっと難点かもしれません。

例:EnemyData ベースとなるスクリプト部分

```
namespace NeetAI.Game.Enemy {  
    [CreateAssetMenu(menuName = "ScriptableObject/EnemyData", fileName = "EnemyData")]  
    public class EnemyData : ScriptableObject {  
        [SerializeField] int hp;  
        [SerializeField] int attack;  
        [SerializeField] int defence;  
        [SerializeField] int luck;  
        [SerializeField] string enemyName;  
        [SerializeField] Sprite enemySprite;  
        [SerializeField] List<ActionBaseData> actionDatas;  
        [SerializeField] Player.ToolObjectData[] victoryRewards;  
    }  
}
```

例:生成したEnemyData



展望:将来的にCSVをScriptableに流し込みたい

# テーマロゴ作成の裏話

学部/回生 情報理工学部 知能情報コース 3回生

名前 つぼ



えー、今年の学園祭テーマロゴを作った人です。  
去年まで使っていたヒノアラシのアイコンは深い深い訳があって封印されました。かといって会誌で顔出しは抵抗があったので非常に悩みました。

尚、ロゴ案の募集は7月7日から27日まで行われていました。

応募しよう！と決意した経緯としては「Adobe Illustratorというソフトを使えるようになりたい！実践あるのみ！！」という理由で試作品という聞こえが悪いですが、自身のスキルアップを目的に取り組みました。

正直なところ僕としては去年のロゴに近いイラストちっくなデザインを作りたいかった。その為、当初は以下の第1案を軸に制作に取り組んでいた。

**しかし、没となった。**

没となった理由としては

- ・ 色付けが非常に難しかった
- ・ なんとなく素人感がした
- ・ デザインの意図という点で少し迷走した



第1案(結果的には没になった)

主にこの3点、特に色付けに苦戦した。

過去のロゴを見ていると、暗黙の了解で『キャンパスカラーの3色を入れる』というのがあり、形を作ってから着色する方法ではどう足掻いても上手く着色出来なかった。力不足だった。

そこで考えたのが以下の第2案で、配色とデザインの意図をしっかりと考えながらこのロゴを制作した。

今回テーマロゴのデザインに落とし込みたかった点は

- ・ '21年度や'19年度以前よりも進化した学園祭を創りたいという意味
- ・ 3色のキャンパスカラーを用いたい
- ・ 今年のテーマ「Next Step」らしさ



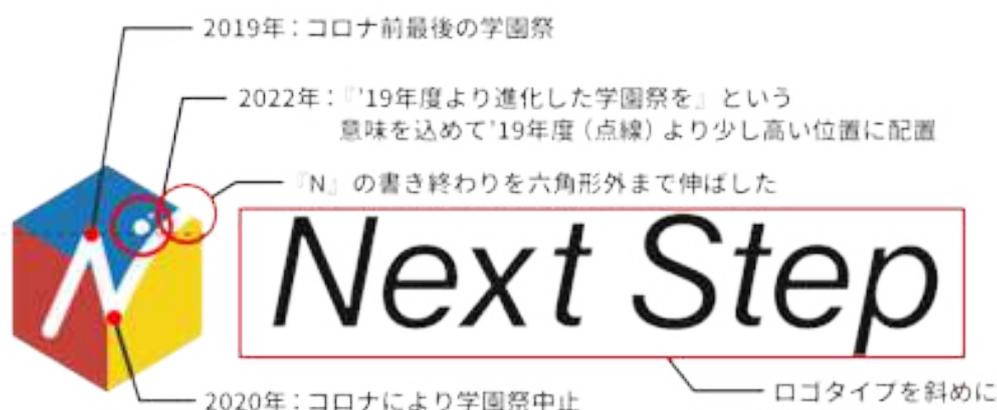
第2案(実際に採用されたもの)

次のページで詳しく解説します。

## > テーマロゴ徹底解説

テーマロゴについての思いは立命館大学新聞社による記事や学園祭パンフレットですで見ただ人もいると思うがここではさらに詳しく説明していこうと思う。

- シンボルマーク中心に位置する「N」は学園祭の盛り上がり度合いの推移を表しています('19年までは順調に盛り上がりを見せ、'20年で中止になったことで大暴落、'21年は開催できたが制限下での開催だった)
- 書き終わりを六角形外まで伸ばすことで『どこまでも続く』ことをイメージし、上部にある('22をイメージしているもの)矢印と共に『立命館大学の学園祭が未来永劫発展を続ける』ということ在意図しています。
- 「N」上部にある丸は'22年現在の位置を表しており、『'21年度はもちろん'19年度以前よりも進化した学園祭を』という意図で'19年度よりも少し高い位置に配置しました。
- キャンパスごとのイメージカラーである「赤」「青」「黄」を用いて、立命館らしさにこだわりました。
- ロゴタイプを少し斜めにすることで全体的に勢いを感じられるデザインに
- 「19年度以前のロゴに似てるよね」と思われた学園祭テーマロゴオタクのみなさん、そこは『コロナ前である19年度までの学園祭をもう一度』という意味も込めて寄せました(という後付けの理由です🙄)



選ばれるとすごい嬉しいし、いい経験になった！みんなも挑戦してみるといいよ。あと日頃見かけるデザインの意味を考えるのも意外と面白かったりする。

<h1>hackathon++</h1>	
学部/回生	情報理工学部・知能 3回生
名前	husky



大学生になって一気に参加する機会が増えたであろうハッカソン  
 まずハッカソンとは短期間でアプリやサービスを作り上げるイベントですが、アイデア発案からタイムスケジュールまで全て自分達で決めるところが大変です。

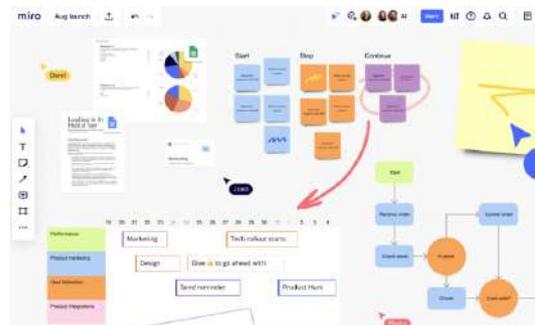
そこでチームまたは個人でハッカソンを行う時に便利なツールを二つほど紹介します。

## ・Miro

ハッカソンの初期段階ではいろいろなアイデアを出したり、細かい仕様を決めたりしますがどこかにメモしておかないと忘れていたら大変です  
 また、文書を作っても一人しか編集できなかつたり、内容の関係が分かりにくかつたりします

そこでMiroを使いましょう

Miroは複数人が自由に付箋を貼ったり、図形やコメントを残すことができるオンラインホワイトボードです。付箋形式なので小さなアイデアや気になったことを書きやすいところが使いやすいです



## ・Canva

ハッカソンのゴールはアプリやサービスを作って終わりではありません  
 発表資料を作って他の人の前でプレゼンすることで評価をもらい、めることができます。スライドを作る時にももちろんPowerPointを使ってもいいですが複数人で作る時にレイアウトが統一できなかつたりデザインがありがたきたりになってしまいます

そこでCanvaを使いましょう

Canvaはブラウザでスライドが作成できるサービスです  
 テンプレートの多くがうるさすぎず、ありがたきたりではないデザインになっており、共同でスライドを作る時にも分担がしやすいところが使いやすいです  
 また、わざわざスライドを書き出してダウンロードせずに、リンクを踏むだけでだれでもスライドを見れるところも便利です

# Unityでの点群表示

学部/回生

情報理工学部・SA3回生

名前

きょうすけ



Unityは何も手を加えない状態だと点群を表示することができないのでやり方をいくつか紹介

## Unityのサポートするファイル形式

Unityは.fbx, .dae(Collada), .3ds, .dxfのファイルを読み込むことができます

逆に言えば点群のファイル形式(.las, .ply等)はそのままでは扱えません  
ちなみにBlenderは.plyファイルはいけます

## 点群の入手方法

表示する点群の入手方法についてです

最近だと専用の機械がなくともiPhoneやiPadでEveryPoint や SiteScape といったアプリを使えば点群が撮影可能です

そういった環境がない人は点群大国静岡の点群データを使えばよいでしょう

「VIRTUAL SHIZUOKA」や「静岡 点群」などでネット検索すると出てきます

ちなみにVIRTUAL SHIZUOKAのデータは今年9月に発生した豪雨災害の被害状況確認のときにも役立てられました

## 点群を表示するためのツール

### Pcx(<https://github.com/keijiro/Pcx>)

このパッケージをインストールすればUnityでPLYファイルを扱えるようになります

PLYをそのままインポートするだけなので表示するだけなら一番手軽です  
このパッケージにはカスタムシェーダーが2つ含まれています

#### ・Point Cloud/Point

このシェーダーを使えば点群はプリミティブとしてレンダリングされます

こちらを使えば後者よりきれいに描画されますが、点の大きさが少なくともD3D11/12環境では変えられないため、VRワールドなどで間近でみるには不向きです

#### ・Point Cloud/Disk

Geometry Shaderです

点の大きさが自由に変えられるため、間近で見ても密度の高い点群が表示されますが、あまり見栄えはよくないです

## Las2Mesh (<https://github.com/ksasao/Las2Mesh>)

Unity用のツールではないですが、お世話になっているので一応紹介  
このツールを使えばLASファイルをPLYファイルに変換することができます

LASをPLYに変換すると先程のPcxでUnityに点群をインポートすることができます

また複数のLASファイルをまとめて変換し、1つのPLYファイルにまとめることができます

オプションでメッシュの細かさを指定できたり、出力形式を.stlや.objに変更したりできます

なお、このツールの様にPythonで点群処理したい方は「[詳解 3次元点群処理 Pythonによる基礎アルゴリズムの実装 \(KS理工学専門書\)](#)」といった書籍を読めばいいと思います

## BA PointCloud ([https://github.com/SFraissTU/BA\\_PointCloud](https://github.com/SFraissTU/BA_PointCloud))

Potreeというフォーマットを表示するツールです

LASファイルをPotreeの変換にはPotreeConverter

(<https://github.com/potree/PotreeConverter/releases>)

というツールが使えます

カメラに写っている箇所を優先的に描画するため、ある程度大きな点群を表示することができます

## さいごに

Unityで点群を表示する方法はこれら以外にもクワマイさんの「[ソーシャルVRに点群を持ち込みたい](https://kuwamai.hatenablog.com/entry/2020/12/17/013711)」(<https://kuwamai.hatenablog.com/entry/2020/12/17/013711>)など色々方法があります  
自分のやりたいこと(ただ表示したいだけなのか、VRで使いたいのか等)に合わせて色々試してみるのがよいのではないのでしょうか

<h1>怒りの日本語Wordle</h1>	
学部/回生	情報理工学部・SNコース 3回生
名前	Jin



プログラミングを始めようとしている人に一番よく聞かれるのが「何からやればいいのか分からない」というものです。技術的に重要で基礎に当たるものから真面目に積み上げて行くのもいい方法ですが、個人的には目に見える何かを作って、その成果物が楽しいものであることがモチベーションを保つのにすごく役に立つ気がします。

そこで、割と最近ネットで流行ってた暇つぶしゲームWordleというものがあつたので、誰でも簡単に作れるチュートリアルを持ってきました。あと日本語Wordleを見つけてやってみたら負けた時に結局答えがなんなのか教えてくれなかったのが激怒して作った成果物だったりします。

まずルールを説明すると、正解の五文字単語を当てるのが目的です。英語なら「ABCDE」、日本語なら「あいうえお」という感じですが、勿論本番では辞書に載っている単語だけを扱います。解答する機会は6回で、当たりの文字は緑、位置だけが違う文字は黄色、正解に含まれてない文字はそのまま灰色にします。正解が「ABCDE」の時に「ABDCZ」を入力すれば「**ABDCZ**」になるということです。

1回目は不正解 →

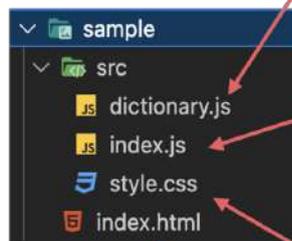
「XXXNT」でどこかにEが入る！ →

5回目で正解！ →

S	T	A	R	T
P	L	A	N	E
P	L	A	N	T
L	E	A	N	T
E	V	E	N	T

## 基本骨子の準備

作業をするフォルダーで  
ファイルを4つ作成します。



```
1 export const Dictionary = [  
2   'abcde',  
3   'apple'  
4 ];
```

```
1 alert("Hello World!");
```

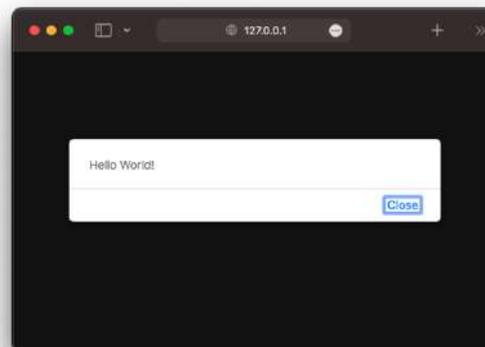
```
1 #game{  
2   display: grid;  
3   place-items: center;  
4 }
```

dictionary.js:登録する単語のリスト  
index.js: 関数など  
style.css:見た目, アニメーションなど  
index.html: エントリーポイント

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4   <meta charset="UTF-8">  
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7   <title>Wordle</title>  
8   <link rel="stylesheet" href="./src/style.css">  
9 </head>  
10 <body>  
11   <div id = game</div>  
12   <script src="./src/index.js" type="module"></script>  
13 </body>  
14 </html>
```

htmlのファイルは赤線の部分以外は  
自動生成されます. 上からタイトル,  
style.cssと連結, gameというIDを  
付与, index.jsと連結しているだけ.

これをLive Serverで開いてindex.js  
で書いたメッセージが見えればOK!



## ゲームを作っていく

dictionary.jsに入る単語リストは  
<https://github.com/charlesreid1/five-letter-words/blob/master/sgb-words.txt>  
から, style.cssでは文字が入るboxと  
boxが5x6で入るGridのクラスを作成して  
色を指定しておく.

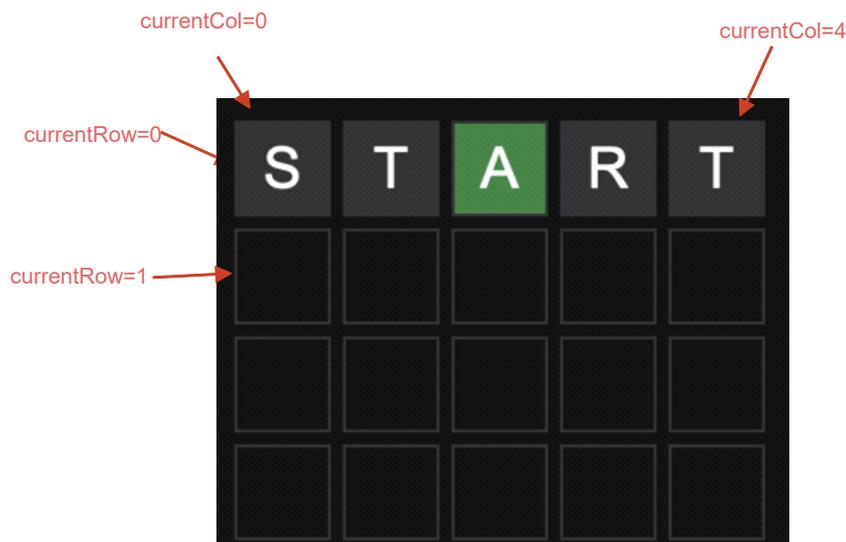
```
:root{  
  --default: #121213;  
  --empty: #3a3a3c;  
  --wrong: #b59f3b;  
  --right: #538d4e;  
}  
.box.empty{  
  background: var()  
}  
.box.wrong{  
  background: var(--wrong)  
}  
.box.right{  
  background: var(--right)  
}
```

index.jsではDictionary.jsからランダムで単語を  
取ってきて正解としてセーブする.

それから入力値がアルファベットか, 単語はDictionary  
に載っているかを確認する関数や,  
エンターキー, バックスペースが  
働くようにする.

```
1 function removeLetter(){  
2 function addLetter(letter){  
3 function isLetter(key){  
4 function revealWord(guess){  
5 function isValidWord(word){  
6 function getCurrentWord(){  
7 function drawGrid(container){  
8 function updateGrid() {
```

## Gridナビゲーション



入りに反応して何かが変わる時、プログラムがGrid内を移動する必要があります。まずはBox(currentRow, currentCol)で今フォーカスしているBoxを把握し、

文字入力: 文字を今のBoxに入れてcurrentCol++で1を  
足す事で1つ右のBoxにフォーカス。

バックスペース: 今のBoxの中身を消してcurrentCol--で1を引  
く事で1つ左のBoxにフォーカス。

エンターキー: 5文字を全部入力した状態ならcurrentColが5  
になっているはずなので、currentColが5でな  
いなら無視。5の場合は正解と比較するなどの  
動作をしてから  
currentRow++, currentCol = 0にする。

## まとめ

以上がメインの動作ロジックになります。

色を変えたり、アニメーションを入れたり、違う言語にしたり、単語帳を特殊なものに変えたり、いろんな方法でカスタマイズできて、実際に遊ぶのも楽しいプログラムでした！

私が作った普通Wordleと日本語版Wordle  
のコードを見て真似てみたい方は

[https://github.com/RosesAndViolets/Wordle\\_Eng](https://github.com/RosesAndViolets/Wordle_Eng)

[https://github.com/RosesAndViolets/Wordle\\_Jap](https://github.com/RosesAndViolets/Wordle_Jap)

こちらに！

実際に遊んでみたい方はこちらにどうぞ！

日本語  
Wordle

英語  
Wordle



# Arduinoに関して

学部/回生

理工学部 3回生

名前

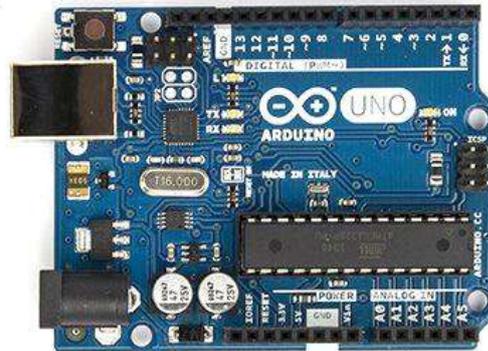
キムテユン



ロボットや機械などの作業によく使われる基板であるArduinoに関して紹介する。

## 1. Arduinoとは？

ArduinoはイタリアのMassimo BanziとDavid Cuartiellesが初めて開発した。名前の意味はイタリア語で強力な親友という意味である。この基板はいわゆる小型パソコンのようなもので、



簡単な回路を組み立て目的に合うようにプログラムすることで、

ほとんど機械や電子作業が可能である。また、Arduinoは構造事態がオープンソースであるため、知識があれば誰も自作で作ることが可能であり、部品を作る会社でもArduinoに合う部品を出していて、気軽に扱えるものになった。

## 2. Arduinoの種類

Arduinoにはいくつかの種類があって、自分が構想している作品に合わせて使える。

まず、ArduinoUnoが一番基本的で、一般的に使われるボードである。上のイメージがArduinoUnoで、主にArduinoを扱えるとすればこれを意味する。最も接続性がよいが、USBを用いてコンピュータと通信するため、2つのピンを使えないという短所がある。これを解決したのがArduinoLeonardoである。これはコンピュータと通信していても制限なくすべてのピンを使える。そして、これらを小型にしたものがArduinonanoとmicroである。

しかし、UnoとLeonardoの性能はほぼ同じなので時々作ろうとするものに対して性能が足りない場合もある。それに合わせて、性能が向上したボードがある。まず、メガ2560はUnoよりピンの数が多い。また、ゼロは処理速度が速いため、より高性能が必要な作業にあっている。



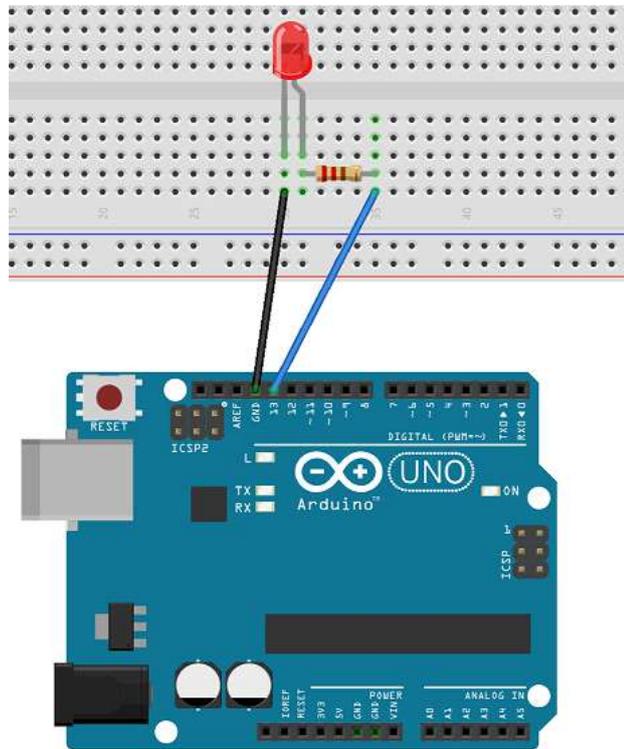
Arduino Leonardo



Arduino mega2560

### 3. Arduino有用性

先述した通りほとんどの機械で電子作業が可能である。今までは専門知識がないと、自作で機械的なものを作るとは不可能に近かった。しかし、Arduinoによって専門知識がなくても簡単な機械は誰でも作れるようになって、今は大学はもちろん高校や中学校、少ないが小学校でもArduinoの教育をしている。また、プログラミング言語がC言語をベースにしているため、関連知識があれば速く学べることも長所である。



つまり、基本的なプログラムさえわかれば誰でも自分が欲しいものを作ることが可能である。特に規格化された基板にはんだ付けをする必要がないことがハードルを低くしたことだと思う。プログラムと回路はネットを調べるとすぐ出るが、はんだ付けは道具も必要なので専門知識と場所が必要である。しかし、Arduinoはブラッドボードにピンを差し込むだけで回路を組み立てることができ、学校はもちろん極端には公園でも可能である。上の絵はArduinoで最も基本的な回路を描いたものである。本来、これをするためには基板にはんだ付けをして、作動させることであり、一度作れば分解することも困難であった。Arduinoを使うことで、簡単に作れるし分解することも可能である。

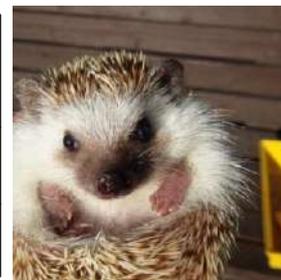
# AIが描く未来

学部/回生

理工学部3回生

名前

キムミヨンス



こんにちは。キムミヨンスです。最近話題になった画像生成AIの発展とこれ  
がもたらす未来に関して書いてみようと思います。

## 1. NovelAI

2022年10月3日、NovelAI Image generator(以下、NovelAIという)が登場し、業界に大きな波紋を投じた。このNovelAIは、単純にタグを入れるだけで、誰でも右のようなクオリティーの高い絵が得られる。

しかし、ネット上にある絵を利用し、学習させて絵を生成する原理を持っているため、著作権が問題になっている。

このため、一部の作家は自分が書いたイラストをネット上で削除したりAI学習に使用しないように表示する場合もあった。



NovelAIが描いた絵

## 2. AIの絵が米国の美術品評会で優勝

米コロラド州で開催されたファインアートコンテストでゲームプランナーのジェイソン・M・アレンさん(39)が画像生成AI「Midjourney」で生成した絵が優勝し、話題になった。

一部の芸術家はアレンがAIで得た絵で優勝したことは「不正行為」と非難した。しかし、アレンはこれに対して「筆を使わなかったことが問題になるのか」、「芸術界は結局AIも一つの道具として認めることになる」と述べた。

現代美術も筆と絵の具を使わない作品が多だけに、様々な問題をどのように解決していくのか期待される。



画像生成AI「Midjourney」が生成した絵  
(Theatre D`opera Spatial)

### 3. 我々は芸術の死を見ているのか

このように、AIの性能は非常に優れており、ある程度のクオリティはAIが代替することで近い将来に絵関連の仕事はAIに奪われるものと見られる。

人間の学習と人工知能の学習は違うのかという質問に「本質的に違わない」と答えたい。「模倣は創造の母」という言葉のように、実は創造とは、模倣を基盤とする場合が多い。

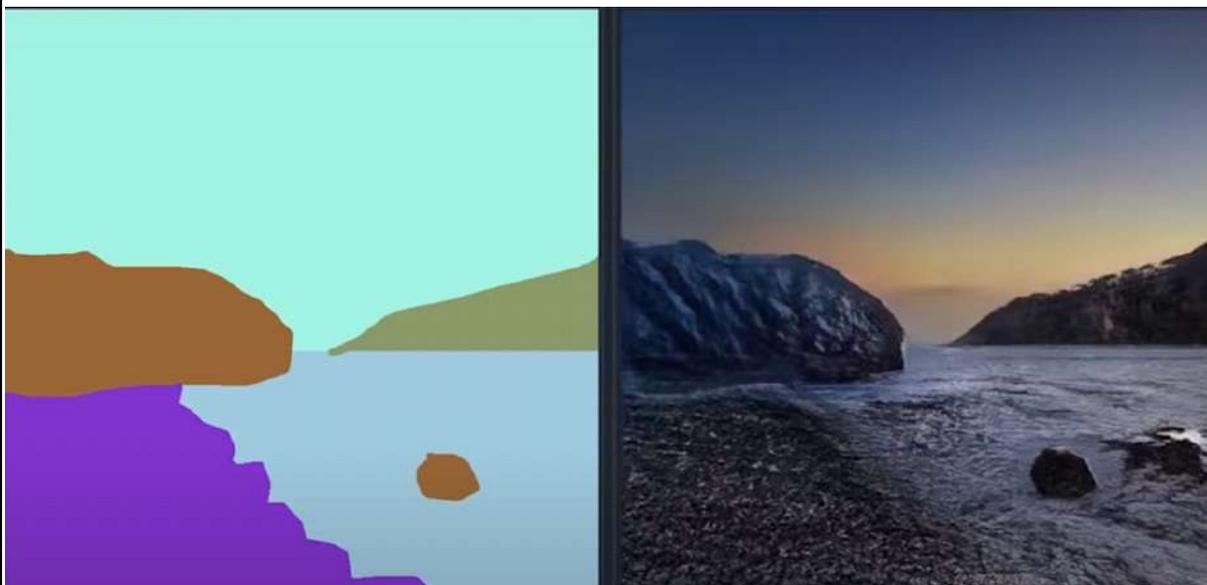
写真機の導入と静物画の衰退、自動車の導入と馬車の衰退のように技術の発展に伴う自然な衰えにすぎない。

変化を恐れてAIを規制して排斥しようとするならば、これは技術的な退歩をもたらすだろう。

AIの発展を脅威と感じる人がいれば、これは彼らの作業があまり創造的ではないということであり、これは絵だけでなく芸術、さらにはプログラミングまでもこの範疇に含まれる。

いずれにしても、AIの発展は避けられないことで、新しい時代にどう備えるかが大事な課題である。

2016年、イ・セドル9段を破ったGoogleの囲碁の人工知能 AlphaGoを筆頭に、2022年の現在は人間の領域とだけ考えていた芸術まで人工知能が侵している。急速に変化する時代の中で、それに見合う法律が後押しされなければならないだろう。



GauGANが生成した絵(左:入力、右:結果)  
(<https://youtu.be/MXWm6w4E5q0> から引用)

# RCC会誌 2022

## —RCCの活動や部員の成果を紹介

---

立命館コンピュータクラブは、立命館大学びわこ・くさつキャンパスを拠点とした学術系サークルです。

特定の分野について調査したり、議論し合うことにより、情報技術や社会についての見識を深めていきます。研究の成果は論文としてまとめて、イベントで頒布しています。

ソフトウェア、ハードウェア、映像、音楽、ゲームなどの制作を行います。制作物は情報技術を使ったものであれば、何でも構いません。

普段は週一回の定例会議があり、ここで活動予定などの連絡が行われます。活動の予定告知やライトニングトークもこの時間を用いて行われます。全体に向けての短時間の勉強会が行われることもあります。

コロナ禍によって、対面でのコミュニケーションがどれだけかけがいのないものであったか、強く肌で感じることとなりました。

今年度はCOVID-19の感染状況が落ち着いていることから対面での定例会議やプロジェクト活動を再開しました。

ぜひ見学にお越しください。

2022年11月15日 初版第1刷発行

2022年12月9日 第2版第1刷発行

制作・発行



学術部公認団体

立命館コンピュータクラブ

Ritsumeikan Computer Club

---



