



大熱血!

RCC会誌
入門

RCC【著】

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，
編集者の堀田と申します．絶賛炎上中です．

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，今年度は新型コロナウイルスの影響もあり部室が使えず，友人，先輩，後輩に会えないとても苦しい1年でありましたが，その中で会員たちが日々の活動を過ごし作り上げた会誌となっています．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ます．

2021年

会誌担当者 堀田隆成

目次

まえがき

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	イベントの開催	3
1.4	普段の活動	4
1.5	活動拠点	4
1.6	入会について	4
1.7	アクセス	4
1.8	年間行事予定	5
2章	前期プロジェクト活動報告	7
2.1	C言語班	8
2.2	AlphaZero班	11
2.3	Unity班	13
2.4	競馬AI班	15
2.5	自然言語処理班	17
2.6	DTM班	19
3章	会員コラム	21
3.1	1回生	22
3.2	2回生	34
3.3	3回生	61
3.4	4回生	73
	奥付

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 イベント

毎年9月頃に、関西情報系学生団体交流会(以下、KC3)を開催しています。これは、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KC3初のオンラインでの開催となりました。(Website:kc3.me)

1.4 普段の活動

普段は、週一回、ZOOMを用いたオンラインでの定例会議があり、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、オンラインでそれぞれの活動を行いました。

1.5 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。

1.6 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円ですが、今年度は設備が一部使えないことなどから4500円となりました。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。

1.7 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

Twitter: @rits_cc

Website: www.rcc.ritsumeai.ac.jp



バイオリンクの入り口

1.8 2020年度 年間行事予定

月	行事	内容
4月	Welcomeゼミ	新入生と上回生が共同で勉強・製作
6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
9月	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	クリスマス会	プレゼント交換など楽しいイベント
2月	追い込み合宿	後期活動の制作物・報告書の仕上げ
2月	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
2月	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

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5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 C言語班

C言語の学習を主軸に、コンピューターシステムの基盤となる部分についての理解を深めました

2.2 AlphaZero班

AlphaZeroで用いられている機械学習の基礎となる部分を輪講の方式で勉強しました

2.3 Unity班

物理演算ソフトである「Unity」に関する知識を深め、各々のメンバーが自作のプロダクトを作ることを目標に活動を行いました

2.4 競馬AI班

競馬をテーマとしたAIの作成を通じ機械学習への理解を深めました

2.5 自然言語処理班

日本語や英語といった自然言語をコンピュータで処理する手法を学びました

2.6 DTM班

作曲の知識を深め、実際に楽曲を作成することで各個人の技術力向上に努めました。

C言語班

山本 京介
情報理工学部 1回生

佐田 淳史
理工学部 1回生

佐久間 智也
理工学部 1回生

川崎 秀昌
情報理工学部 1回生

西村 雅貴
情報理工学部 1回生

Park Jooinh
情報理工学部 1回生

奥川 莞多
情報理工学部 2回生

中尾 龍矢
情報理工学部 2回生

西見 元希
情報理工学部 3回生

坪倉 奏多
情報理工学部 3回生

1. 活動概要

本班はC言語及びそれを用いたシステムプログラミングの基礎を学ぶことを目標としたプロジェクトである。C言語を学ぶことでC言語の流れを組む現在のほぼ全てのプログラミングの習得を容易なものとする事ができる。また、ハードウェアに近い言語であるという特徴を活かし、OSのシステムコールを叩いてプロセスやスレッドを走らせる体験をすることでOSの仕組みについての理解を深めることができる。

本班の目玉となる活動はC言語を用いたShell自作とEditor自作である。ただプログラミングを初めて行う新入生でもこの活動に参加できるよう、前期の間は上級生が新入生に対して指導を行い、その上で後期に各自で実践的なプログラムの作成を行うようにした。なお前期後期を通してshellの基本的な操作は適宜指導を行った。

2. 前期の活動

前期は上級生が交代で新入生にゼミ形式でC言語学習会を行った。

参考資料は担当する上級生がscrapboxで作成していった。資料は主要テーマの説明と練習問題で構成され、毎週冒頭に前回の練習問題の解説を行った。各週の主題は以下である。

- **第1週**

ターミナルの扱い方、Hello, World、変数の紹介

- **第2週**

変数と型、演算子

- **第3週**

制御構文(if文・switch文・while文・do-while文・for文)、三項演算子、条件式

- **第4週**

配列、文字列

- **第5週**

関数

- **第6週**

構造体、列挙型

- **第7週**

ポインタ

- **第8週**

ポインタの応用

- **第9週**

マクロ、ファイル処理

- **第10週**

catコマンド自作の説明

夏季休暇には新入生に対して前期に学習したC言語の復習と後期の予行演習を兼ねcatコマンドの自作を課し、後期第1週で各自発表を行うことになった。

3. 後期の活動

後期は自作catコマンドの発表の後、Shell自作を本格的に始動した。後期も前期に引き続きscrapboxを活用して活動を行っていく、

- **第1週**

catコマンド発表、後期活動の事前説明

発表作品の一部は以下

<https://github.com/kyochn/c-project-kyochn/tree/master/cat>

https://github.com/lufeee/customc_cat

- **第2週**

OSの基礎知識、自作shellでsleepコマンドの実行

- **第3週**

環境変数の受け渡し、バックグラウンド実行機能の追加、exitコマンドの実装

- **第4週**

特定のsignalを受け取った時の挙動の変更、内部コマンド(jobs/fg)の実装

4. 今後の活動

自作Shellをある程度の形まで持っていく、Editor自作を行っていく.

AlphaZero 班

深田 紘希
情報理工学部 2回生

阿部 竜也
情報理工学部 2回生

田邊 雄士
情報理工学部 2回生

原 佑馬
情報理工学部 3回生

高山 紗世梨
情報理工学部 3回生

海原 義樹
情報理工学部 3回生

青木 雅典
理工学部 4回生

玄元 奏
情報理工学部 4回生

1. 活動概要

2015年にAlphaGoという囲碁AIが囲碁チャンピオンに勝利したことで世界を驚かせた。さらに2017年には自己学習のみでAlphaGoに勝利するプログラムであるAlphaGoZeroが制作された。同年、AlphaGoZeroを汎用化させて他のボードゲームにも応用可能なように改良されたAlphaZeroが制作されることになった。AlphaZeroは様々なゲームに応用することができ、棋譜などの教師データを必要とせず、また簡単ではないが比較的シンプルなコード構成をしているため、基本的な強化学習を勉強した後に入門しやすいのではないかと考えた。

本班の主な活動目的は、AlphaZeroのアルゴリズムを用いてオセロAIを制作することである。

本班は毎週担当者が指定された本を参考にしながら担当範囲を説明し、その後議論するという輪講の形式で進めた。コロナ禍の影響で実際に集まることが難しいため、Zoomを用いて活動を進めることとした。その際動画を撮影し、見返すことができるようにした。

2. 前期の活動

前期は深層学習・強化学習・ゲーム木を用いた探索の基礎を勉強した。

- 第0週目
Python基礎講座（任意参加）
- 第1週目
顔合わせ、GoogleColaboratoryの説明、スケジュール説明、
輪講の担当者決め
- 第2週目
深層学習①:分類・回帰について
- 第3週目
深層学習②:CNN、(Resnet)について

- 第4週目
強化学習①:多腕バンディット問題、方策勾配法について
- 第5週目
強化学習②:Q学習、DQNについて
- 第6週目
探索①:ミニマックス法、アルファベータ法
- 第7週目
探索②:原始モンテカルロ探索、モンテカルロ木探索

実際は予定がずれ込み夏季休暇中に第6週、第7週を行っている。

3. 後期の活動(予定含む)

前期の活動で得た知識をもとに本班の目的であるオセロAIの制作に向けてまずは三目並べのAIを制作の目標とする。

- 第1週目
三目並べ①:デュアルネットワークの作成
- 第2週目
三目並べ②:モンテカルロ木探索の作成
- 第3週目
三目並べ③:セルフプレイ部分の作成
- 第4週目
三目並べ④:パラメータ評価、ベストプレイヤーの作成
- 第5週目
三目並べ⑤:実際の学習サイクルについて
- 第6週以降
いままでの知識をオセロに転用する。班員のモチベーション維持のためZoomを開けておき、任意参加で質問や進捗把握を行う。

4. 今後の活動

後期の活動が現状進められていないため、計画を少し見直して進める必要があると感じている。参考書に指定した本にミスが多数見つかかり、修正を進めているものの進捗に影響が出てしまっている。進捗によっては、三目並べAIの完成や理論の理解という部分に目標を変更することも考えている。

Unity 班

堀越 俊行
情報理工学部 3回生

新藤 尚輝
情報理工学部 2回生

宇佐 基史
理工学部 2回生

原 佑馬
情報理工学部 3回生

北村 優奈
情報理工学部 3回生

中山 凌一
情報理工学部 3回生

服部 瑠斗
情報理工学部 3回生

岡本 陽太
情報理工学部 3回生

桐井 優実
情報理工学部 2回生

中川 拓海
情報理工学部 3回生

伊藤 佑
理工学部 3回生

山本 京介
情報理工学部 1回生

活動概要

Unity班は、unityの基礎から扱い、アプリケーションを作成できるようになることをメインに活動を行っている。全体を通して、Unityの知識を深め、プロダクトを作ること为目标にしている。

内容

前期は週に2度、月曜日と金曜日に活動を行っていた。活動はDiscordを用いてリモートで行っている。後期は、現在、週に1度、水曜日に活動を行っている。前期と同様に、Discordを用いてリモートで行っている。以下に日程を示す。

前期日程

第一回 Unityの勉強として、作成したい3Dゲームの考案

第二回 Unityの基礎
ゲーム企画発表

第三回 Unityの基礎
プロジェクト発表

第四回 Unityの基礎
ゲーム制作開始

第五回 Unityの基礎
ゲーム制作におけるトラブルシューティング

後期日程

- 第六回 Unityの基礎
ゲーム制作におけるトラブルシューティング
- 第七回 Unityの基礎
ゲーム完成、発表
- 第八回 1:作成したいXRプロダクトの考案
- 第九回 1:XRの基礎AR編 XRアプリケーション企画発表
- 第十回 1:XRの基礎VR編 プロジェクト発表
2:作成したいゲームの考案
3:Unityの応用 プロジェクト発表
- 第十一回 1:XRの基礎 XRアプリケーション開発の開始
2:Unityの応用 ゲーム企画発表
- 第十二回 1:XRの基礎 XRアプリケーション開発におけるトラブル
シューティング
2:Unityの応用 プロダクト制作開始
- 第十三回 1:XRの基礎 XRアプリケーション開発におけるトラブシュー
ティング
2:Unityの応用プロダクト制作におけるトラブルシューティング
- 第十四回 1:XRの基礎 XRアプリケーション完成、発表
2:Unityの応用 プロダクト完成、発表

今後の予定

今後は、それぞれの会員が考案したプロダクトを各々作成していく。

競馬AI 班

堀田隆成
情報理工学部 2回生

玄元 奏
情報理工学部 4回生

木村悠生
情報理工学部 3回生

坪倉奏太
情報理工学部 3回生

阿部健太郎
情報理工学部 2回生

深田紘希
情報理工学部 2回生

田邊雄士
情報理工学部 2回生

星名藍乃介
情報理工学部 2回生

水野悟朗
情報理工学部 2回生

齋藤竜也
情報理工学部 2回生

中尾龍矢
情報理工学部 2回生

佐田淳史
情報理工学部 1回生

西村雅貴
情報理工学部 1回生

1. 活動概要

本班は競馬AIの作成を通じて機械学習に関する知識、必要なテクニックを習得することを目標としたプロジェクトである。競馬を機械学習のテーマとして取り扱った理由としては、様々な要因がレースの結果を左右すると考えられるといったものである。

活動方針としては班員ごとに担当を割り当て、その者を中心とした勉強会方式で行なった。

2. 前期の活動スケジュール

前期には競馬AIの仮作成を目標として活動を行なった。また一週、二週の内容はアルファゼロ班と被っていたため合同での勉強会を行なった。

- ・一週目： イントロダクション 環境構築, 機械学習, 競馬について
- ・二週目： 機械学習① 分類, 回帰について
- ・三週目： 機械学習② CNNなどについて
- ・四週目； Webスクレイピング①
- ・五週目： 教師データのモデリング

- ・ 六週目：Webスクレイピング②実践
- ・ 七週目：前処理

3. 後期の活動

後期には実際に前期の目標でもあった競馬AIの軽量版を作成し、そのAIの精度向上を目標として活動する。現在完成している軽量版の競馬AIは学習のためのデータが不足しているため、今後の活動では過去12年分のレース情報、競走馬データ、騎手データ、調教師データの取得をメインに行ない、競馬AI完成を目指す。

4. 今後の活動の展望

作成した競馬AIの精度を高めるために類似した内容の論文を読み機械学習への理解を深める。現状の競馬AIは1着の予想をすることしかできないため、3着までの予想をするなどし、効率的な馬券の予想を行えるように改良する。競馬AIの予想をツイートするBotを作成したが、作成し数日で動作しなくなったためその原因を探る。

自然言語処理 班

阿部 健太郎
情報理工学部 2回生

山本 京介
情報理工学部 1回生

奥川 莞多
情報理工学部 2回生

小柳 雅文
情報理工学部 2回生

深田 紘希
情報理工学部 2回生

星名 藍乃介
情報理工学部 2回生

八木田 裕伍
情報理工学部 2回生

原 佑馬
情報理工学部 3回生

玄元 奏
情報理工学部 4回生

山岡 聖弥
情報理工学部 4回生

1. 活動概要

自然言語処理班では、日本語や英語といった自然言語をコンピュータで処理する手法を学ぶことを目的としている。前半は自然言語処理の基礎技術を、後半はそれを応用したテキスト分類、ネガポジ分析等の応用技術をハンズオンな勉強会形式で実践する。最終的にはチャットボットを開発し、自然言語処理アプリケーションを開発できるスキルを身につける。

2. 前期の活動

前期では、自然言語処理における基礎技術、前処理、コーパスを扱った。

- 第1週目
イントロダクション: 自然言語処理とは?
- 第2週目
Python講座 / 環境構築
- 第3週目
前処理①: コーパスを紹介 / テキストのクリーニング / 単語分割
- 第4週目
前処理②: 正規化 / ストップワード除去 / 単語のベクトル化
- 第5週目
基礎技術①: 形態素解析 / 意味解析
- 第6週目
基礎技術②: 構文解析 / 文脈解析
- 第7週目
応用: 単語の分散表現で遊ぶ

3. 後期の活動

前半では基礎技術とPythonでのテキスト処理についてを学んだが、後半ではそれらの手法を応用したより実践的な自然言語処理を扱う。

- 第1週目
応用技術①:センチメント分析
- 第2週目
応用技術②:ルールベースでのテキスト分類
- 第3週目
応用技術③:教師あり学習でのテキスト分類
- 第4週目
応用技術④:トピックモデリング
- 第5週目
応用技術⑤:テキスト生成
- 第6週目
チャットボット開発①
- 第7週目
チャットボット開発②
- 第8週目
チャットボット開発③
- 第9週目
チャットボット開発④

4. 今後の活動

後期の活動予定が達成されていないため、内容をソリッドに削ぎつつ、チャットボット開発に必要な応用技術のみ扱う予定である。今後は各班員がTwitterアカウントのアーカイブを用いて、自身の発言を模したボットを作成することを目標とする。

DTM班

齋藤 竜也
情報理工学部 2回生

宇佐 基史
理工学部 2回生

高山 紗世梨
情報理工学部 3回生

青木 雅典
理工学部 4回生

駒谷 亮叡
理工学部 1回生

海原 義樹
情報理工学部 3回生

田尻 聖奈
情報理工学部 1回生

堀越 俊行
情報理工学部 3回生

平井 柊太
情報理工学部 3回生

西村 雅貴
情報理工学部 1回生

伊藤 佑
理工学部 3回生

1. 活動概要

DTM班は11月に開催される学園祭に向けて自作楽曲を発表することと自身が好きなジャンルの楽曲を自力で制作できることを目標に活動した。

そのために、作曲に関する知識を深め、実際に楽曲を作成することで各個人の技術力向上に努めた。

2. 前期の活動

2.1 DTM入門

DTM初心者向けの入門講座を行った。DAWの説明やDTMとはどういうものなのかの説明を行った。

2.2 耳コピ

耳コピとは既存曲のメロディーやコードなどを聞き、そのまま再現するものであるが曲の雰囲気やDAWの使用法、作曲の構成などを学習するために行った。課題曲を決め、各自で再現したもの班員で聞き良い点・違っている点を指摘しあった。

2.3 音楽理論講習会

音楽理論の勉強会を行った。勉強会では、2回生以上が主催しスケール・リズム・コードに分け、それぞれ担当者を分けて開催した。

スケールの回では、メジャースケールとマイナースケールの説明をし、規則性と曲を構成している音との関係とを関連づけて説明した。実際にスケールの音を鳴らし雰囲気を掴むことも行った。

リズムの回では、ドラムをメインに曲のリズムパターンを説明した。また、拍子の説明を行った。休符や音符と関連づけて説明した。これによりある程度の楽譜の知識向上も測った。

コードの回では、コードの話とコード進行の話を行った。コードの話ではコードの名付け方の規則を説明した。コード進行ではダイアトニックコードの説明や主要なコード進行の例を紹介した。また4度圏表の説明も行い、ある程度自身でコード進行を作成できるよう説明した。

3. 後期活動

前期活動では、曲の制作に関する勉強を行ってきたため、後期活動ではmixなどの曲をより良くするための処理を説明する。またシンセサイザーなど操作が複雑なものの説明も行っていく予定である。

また前期で説明しきれなかった音楽理論の難易度が高い部分についても説明を適宜行う。

4. 今後の目標

学園祭に向けてCDを作成することが目標であるが、新型コロナウイルスの影響により学園祭が開催されないことも予想される。その際は、SoundCloudなどのサービスに自作曲をアップロードする。

会員コラム

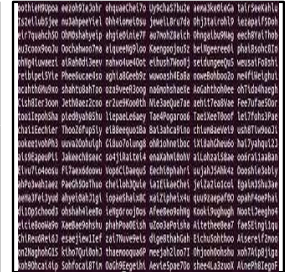
この章では、学年にかかわらず、会員一人ひとりの情報系の技術に関する知見や興味のある分野に対する熱意や想い、その魅力などを綴ってもらいました。情報系の技術に初めて触れたことについて語ってくれた会員もいれば、既に精通していて専門的な内容を語ってくれた会員もいます。

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、競技プログラミング、DTM、機械学習、アルゴリズム、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。

AES暗号とアルゴリズム

学部/回生	電子情報 1回生
名前	asupara



DES暗号は鍵長が56bitであったため計算機の高速度化(授業でもムーアの法則:チップに集積できるトランジスタの数は1年半で2倍になっている! みたいなお話がありましたね)に伴って時代遅れとなりました。AES competitionを経て2000年にルーベンカトリック大学のJoan DaemenとVincent Rijmenが公開したRijndaelという暗号がAESとして採択されたのは有名な話です。そしてAESはFIPS197として規定されました。

(<http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf>)

アメリカの政府ではこれを使うことが義務付けられ

CRYPTREC(電子政府推奨暗号の安全性を評価、監視し、暗号技術の適切な実装法、運用法を調査、検討するプロジェクト)でも推奨暗号となっています。

AESはブロック長が128bitのブロック暗号です。鍵長は128/192/256bitに対応しています。SPN構造*となっています。AESには細かい処理系がたくさん存在しますが基本的に8bit単位で処理されているため実装さえきちんとできていればソフトウェア、ハードウェアの両面において無駄なbitがないといえ非常に効率の良い優れたものの暗号といえるかとされています。

*注SPN構造(Substitution-Permutation Network)についてSub-keyと暗号化の中間データとの排他的論理和、substitution, permutationの処理を繰り返すことで暗号化する構造のこと。

Botをつくったよ



学部/回生	情報理工学部1回生
名前	husky

皆さん『Bot』聞いたことありますか？

「あ～あれやろ...ゲームで同じ動きしかしない敵とかやろ」

大正解です。BotはRobotから生まれた言葉で、視覚的に指定された事を淡々とこなすプログラムのことを指します。ゲームの他には、Lineの公式アカウントやTwitterのネタbotなどがそうです。

そんな身近にあふれているBot君達を自分で作ってみたいと思い、今回私はコロナ渦で利用が増えているDiscord上で稼働するチャットBotを作ってみました。

Pythonで実装しましたがコードは調べたら出てくるので割愛して、内容や手順を紹介します。まずDiscord Botについて軽く説明すると、アプリの性質上Lineのようなチャット応対型になるのですが、プログラム次第ではRythmやpokecordなどのように音楽を流したりチャット上でポケモン対戦が出来たりします。



せっかく自分で作るのならオリジナルのものが良い!
ということで、まず初めに入力した単語をWikipediaで調べてくれる機能を作りました。詳細としては入力された単語を取得してURLエンコードをし、wikiのURLに繋げて返しています。
(チョー簡単)

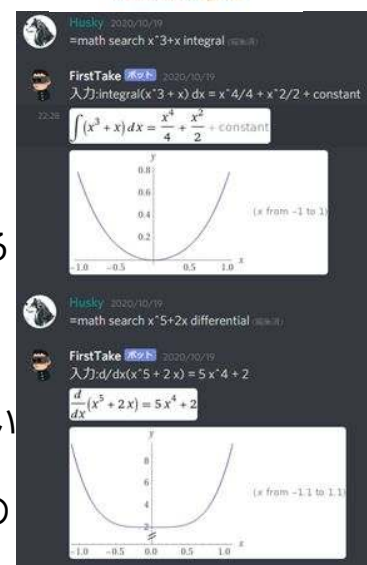


TRPG用のランダムサイコロも作りましたが、一番苦労したのは**数学の問題を解いてくれる機能**でした。Python初学者の私にとって自動で計算をしてくれるプログラムは荷が重いので、巷で有名なWolframAlphaのAPIを利用して実装しました。

結果、四則演算以外にも微分積分などもしてくれる
すげーBot君が出来てしまいました。

まとめ

Pythonの勉強も並行してやったため、合計12hぐらいかかってしまいました。(実力不足)
それでもこの機会が自分を成長させてくれたと思うのでヨシ。



テクニシャンユーチューバーになろう！

学部/回生 情報理工学部/ 1 回生

名前 朴 柱仁



こんにちは 仁君です。

私は日本に来て初めて見たんですが、情理の皆様は割とバーチャルキャラとかVチューバとかに興味があってプレゼンとかもそれでやってましたね。私はバーチャルとかはさっぱりなんですがユーチューブには興味がありまして！

そこで、Vチューバを目指す情理の皆様！どうせやるなら情理っぽく、映像のビューカウントに合わせて随時アップデートされるビデオタイトルを作ってテクニシャン認定されましょう！Tom Scottさんみたいに！

<https://www.youtube.com/watch?v=BxV14h0kFs0>

1. 環境構築

まず、serverless cloud functionを作るためにFirebaseをインストールします。

```
jintonic at DESKTOP-OTPCQM in /mnt/c/Users/qkrwn  
λ sudo apt install firebase |
```

Runtimeを使うのでNode.jsとnpmもインストールします（これは片方インストールしたらもう片方もくっついて来ます）

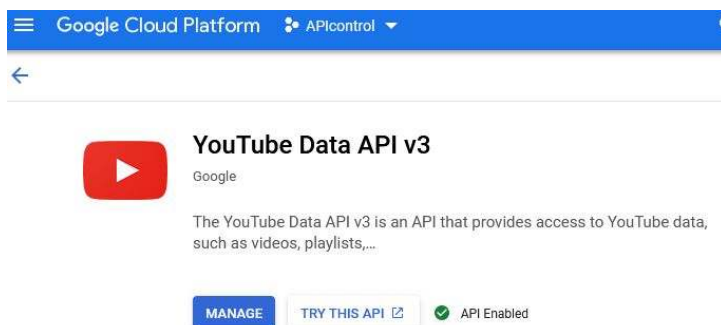
```
jintonic at DESKTOP-OTPCQM in /mnt/c/Users/qkrwn  
λ sudo apt install nodejs
```

Firebaseにログイン（Youtubeのグーグルアカウントでしたら楽かも）してからInit、新しいサーバーを作ります。

```
jintonic at DESKTOP-OTPCQM in /mnt/c/Users/qkrwn  
λ firebase login|  
jintonic at DESKTOP-OTPCQM in /mnt/c/Users/qkrwn  
λ firebase init|
```

2. APIを準備する

YoutubeのAPIを使うのでGoogle Cloud Platformに行ってDashboard->Enable APIs and Servicesを押してYoutube Data API v3を有効化します。



3. 鍵を用意する

Youtubeに書き込むためにはOAuth 2.0が必要なのでGoogle Cloud PlatformでCreate Credentials→OAuth client IDを使う。生成した鍵はJsonファイルとして保存してコードで読み込みます。

APIを有効化したのでgoogleさんのAPIを呼び込むことができるようになりました。

```
const functions = require('firebase-functions');
const admin = require('firebase-admin');
admin.initializeApp();

const { google } = require('googleapis');
```

4. 本体のコード

は私が好きIT系Youtube Channel、Fireshipが書いたデモバージョンを使います。

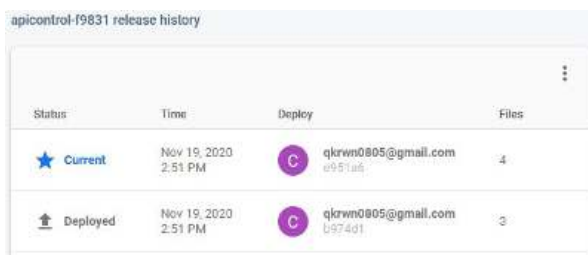
<https://github.com/fireship-io/this-video-has-n-views>



コードの中身は

- (1) ユーチューブにログイン（認証）
- (2) APIを使ってビデオの詳細を得る（ビューカウントの他にもいいねの数とかも可能）
- (3) OAuthの鍵を使って動画のタイトルを書き換える。
- (4) これらをX分ごとに繰り返す（デモでは3分ごとにしてるけどそれだと直ぐにQuotaをオーバーしちゃうので気を付けましょう）

という動作をしています。

5. 環境もコードも用意出来たらFirebaseを自分のHostingしてるサーバーにDeployします



Status	Time	Deploy	Files
★ Current	Nov 19, 2020 2:51 PM	 qkrwn0805@gmail.com e951a6	4
📦 Deployed	Nov 19, 2020 2:51 PM	 qkrwn0805@gmail.com b974d1	3

こうすればBackgroundで自動アップデートしてくれるようにすることができます！

ただし！YoutubeのAPIにはQuotaがあり無制限にAPIを無料で使えるわけではないので最初は動作確認だけして直ぐに解除しちゃいましょう！



私は勇気がないので直接ユーチューバにはならないと思いますが、陰ながら皆様を応援することにいたします。目指せ！テクニシャンVチューバ！

Excel VBAを使ってみる

学部/回生 理工学部 1回生

名前 こま

アイコン



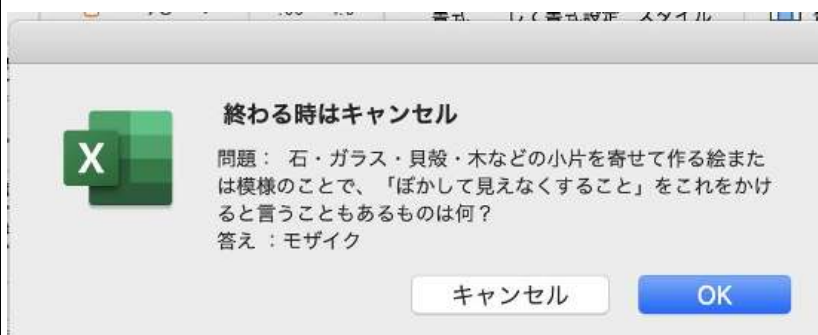
私はクイズサークルに入っています。
作ったクイズはExcelで保存することが多いです。
問題と答えをランダムに出力するマクロを作ろうと思います。

```
Sub 問題()  
  Dim fst, lst  
  Dim i  
  For i = 1 To 55  
    Dim ans As Integer  
    buf = Range(Cells(num_cell(i), 2), Cells(num_cell(i), 4)).Value  
    Dim msg As String  
    msg = "問題： " & buf(1, 1) & vbNewLine & "答え：" & buf(1, 3)  
    ans = MsgBox(msg, vbOKCancel, "終わる時はキャンセル")  
    If ans = vbOK Then  
      GoTo Continue  
    Else  
      Exit For  
    End If  
  Continue:  
  Next i  
End Sub
```

できました



リセットを押すと問題の順番がリセットされて、問題を押すと



こうなります

続ける時はOKを押して、終わる時はキャンセルを押します。

404 Not Found



学部/回生	理工学部 物理科学科1回生
名前	クマさん

タイトルが404 Not Foundということで、内容の方も空白のままにしたかったのですが、そうは問屋が卸してくれそうにないので、今回はwebでよく見る"404 Not Found"について解説してきたいと思います。

そもそも"404 Not Found"とは何ですか？

"404 Not Found"は簡単にいうと、そのページは存在しないぞということを表すHTTPステータスコードであり、webサーバーからwebブラウザなどのクライアントに送られるレスポンスのステータスを表していると思ってもらえればOKです。

Not Found

The requested URL /index.html was not found on this server.

Apache/1.3.37 Server at www.this-is-a-test.net Port 80

出典:<https://cocoa-inc.jp/344>

なお、最近のwebサイトは404専用ページを用意していることが多いためこのような質素な表示ではないことが多いでしょう。

ちなみに、HTTPステータスコードの3桁の数字の先頭の桁はどのようなレスポンスであるかを示しており、1はinformation (情報)を示しており、2はsuccess(成功)、3はredirection(リダイレクション)、4はclient error(クライアントエラー)、5はserver error(サーバーエラー)を示しています。

残りの2桁はより詳しい内容を示していると思ってください。

例えば、同じクライアントエラーレスポンスであっても、

404 Not found

400 Bad Request

というようにどのようなエラーであるのか詳しく表示できます。

(400 Bad Requestは構文が無効であるためサーバーがリクエストを理解できないことを表しています)

404 Not Foundの愉快?な仲間たち

ここからは404 Not Foundの仲間たちとwebで稀に見ることがあるかもしれないステータスコードの一部を紹介していきたいと思います。

・ 401 Unauthorized

unauthorized(不許可)となっていますが、正確にはunauthenticatedすなわち未認証を表しており、クライアントはリクエストしたレスポンスを得るためには認証を受ける必要があります。

・ 410 Gone

これはリクエストされたコンテンツが永久的に削除されており、転送先のアドレスがない場合に送られるコードですが、大抵の場合は404で代替されることが多いようです。

・ 502 Bad Gateway

このステータスコードはゲートウェイまたはプロキシサーバーが上流のサーバーから無効なレスポンスを受け取ったことを表しています。

・ 504 Gateway Timeout

このステータスコードはゲートウェイが時間内にレスポンスを受けられなかったことを示しています。

・ 200 OK

リクエストが成功したことを示しています。

他にもたくさんのステータスコードがあるので調べてみてください！

参考文献

- ・ <https://developer.mozilla.org/ja/docs/Web/HTTP/Status>
- ・ <https://digitalidentity.co.jp/blog/seo/seo-tech/howto-http-status-code.html>
- ・ https://ja.wikipedia.org/wiki/HTTP_404

C言語初学あるある(?)

学部/回生 情報理工学部 情報理工学科・SA1回生

名前 きょうすけ



C言語を書いていく上で個人的に躓いたところをまとめました。独断と偏見で選んだのでないないの可能性もあります。

1. `printf("%lf",double);`

C言語では標準入力を行う際は`scanf()`、標準出力を行う際は`printf()`という関数をよく使います。両者についている`f`という文字は`format`(書式)の頭文字で、その名の通り入出力する値の書式を指定しないとイケません。(例えばPython風に`printf(2+5);`といった書き方はできません。)

変数の書式指定を行う文字はフォーマット指定子と呼ばれています。各フォーマット指定子は対応する型が決まっております。対応する型と入出力するものの型が異なればコンパイルエラーがでます。

基本的に`scanf()`で使う指定子(入力指定子)と`printf()`で使う指定子(出力指定子)は各型で同じですが、異なる型もあります。それは`double`型です。`double`型の入力指定子は`%lf`で、出力指定子は`%f`です。

`double`型の出力指定子を`%lf`としてしまう人は昔からあまりにも多かったためC99(1999年に発行されたC言語の規格)以降は`%lf`でもコンパイルできるようになってしまいました。当然C99以降に対応してない環境ではエラーになるので、`double`型の出力指定子は変わらず`%f`を使うべきでしょう。

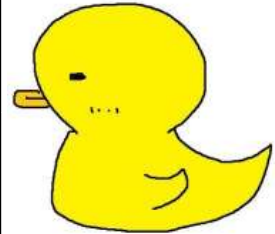
2. `while(true)`

これはC言語を初めてやるプログラマーに起こりそうなことですが、`while`の条件式を書く部分に`true`という`bool`型を書いてしまうといったミスがあります。

なぜこれが間違いかということ、C言語の`stdio.h`には`bool`型が定められていないからです。先の書き方をしようとすれば`stdbool.h`というライブラリを追加する等をしないといけません。つまり`#include<stdio.h>`を『おまじない』として習っただけの人や他の言語で扱った`bool`型変数を`stdio.h`だけ`include`した状態で扱ってしまい先の書き方をしてしまうという訳です。ちなみに`stdbool.h`が導入されたのはC99からです。

逆にCユーザーがC#で`while(1)`という書き方をしてしまうパターンもミスがあります。

GDBのすゝめ



学部/回生	情報理工学部 情報理工学科・SN 1回
名前	lufe

こんばんは、最近加入したlufeと申します。lufeの読み方がわからないと言われるので参考程度に。どうも「るふえ」です。今回の主題からは離れますが、最近いろいろな分野に触りすぎて何もわからない状態で悲しみを背負っています。さて、主題に戻りますが皆さんはGDBを知っていますでしょうか？多くの方は知っていると思います。そうです「デバッガ」です。デバッガはGDBの他にOllyDbgやx64dbgなど他にもありますが、やはりここはシェア率トップだと私が勝手に信じ込んでいるGDBを紹介したいと思います！

やはりデバッガといえば1つ1つのプログラムを実行させてどのような挙動をしているのかを見ることが出来る最高のツールです！CTFなどのpwnでよく使われるやつですね。そして今回話そうと思っているのがc言語のみんな大好きprintf関数ですね。皆さんはprintf関数の中で実際に何が行われているのかを気になったことはありませんか？あると思います。実際にprintf関数の中に文字列「だけ」を渡してあげるとほとんどのコンパイラの場合(私の環境はgcc)は最適化されてputs関数になっています。これも実際にGDBで見ることができます！まあ、とりあえず実際に見てみましょう。

以下のサンプルコードを用います。みんな大好きHello world!です

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

これをgccのgオプションでデバッガを可能にし、oオプションでHelloって名前の実行ファイルにしました！そこでgdb Helloを行うとデバッガが起動します。

そうするとターミナル上にいろいろと文字が出てきてgdb\$と出ます今回main関数の中でprintf関数によってHello world!を表示するプログラムを書いたのでmain関数の中を見に行きましょう！

disass mainとやればmain関数の動向をアセンブラ言語で下のように見ることができます。とりえあず見ていくと、<+15>のところで

puts関数が呼ばれていることが確認できました！

```
gdb-peda$ disass main
Dump of assembler code for function main:
0x000000000001149 <+0>:   endbr64
0x00000000000114d <+4>:   push   rbp
0x00000000000114e <+5>:   mov    rbp, rsp
0x000000000001151 <+8>:   lea   rdi, [rip+0xeac] # 0x2004
0x000000000001158 <+15>:  call  0x1050 <puts@plt>
0x00000000000115d <+20>:  mov    eax, 0x0
0x000000000001162 <+25>:  pop   rbp
0x000000000001163 <+26>:  ret
End of assembler dump.
```


このようにGDBでは実際の最適化されたプログラムを見ることができるとい
ことです。では今回はputs関数の中身を見ていきたいと思います。先ほど
printf関数と言ったがあれは嘘です。break Hello.c:5 を打つとHello.cの5行
目でブレイクポイントを打つというコマンドです。ブレイクポイントはrunコマン
ドなどで実行したとしても、ブレイクポイントまで実行して止まるという物で
す。これはプログラムの実行した後の動向を知ることができるという一番大事
な機能です。そして次にstepinというコマンドがあります。これは読んだ関数
の中を見に行くことができるコマンドです。逆にnextinというコマンドは関数の中
に入らずに次の命令を見に行くことができます。これらのコマンドを駆使し
てまずは1度puts関数の中身を見に行こうと思います！

```
code
0x5555555514d <main+4>:  push  rbp
0x5555555514e <main+5>:  mov   rbp, rsp
0x55555555151 <main+8>:  lea  rdi, [rip+0xeac]          # 0x555555556004
=> 0x55555555158 <main+15>: call  0x55555555050 <puts@plt>
0x5555555515d <main+20>:  mov   eax, 0x0
0x55555555162 <main+25>:  pop   rbp
0x55555555163 <main+26>:  ret
0x55555555164:      nop   WORD PTR cs:[rax+rax*1+0x0]
Guessed arguments:
arg[0]: 0x555555556004 ("Hello world!")
```

ここからstepinコマンドを2回入力すると見事にputs関数に入れました。

```
code
0x55555555044 <__cxa_finalize@plt+4>: bnd jmp QWORD PTR [rip+0x2fad]          # 0x555555557ff8
0x5555555504b <__cxa_finalize@plt+11>:  nop   DWORD PTR [rax+rax*1+0x0]
0x55555555050 <puts@plt>:      endbr64
=> 0x55555555054 <puts@plt+4>:  bnd jmp QWORD PTR [rip+0x2f75]          # 0x555555557fd0 <puts@got.plt>
0x5555555505b <puts@plt+11>:  nop   DWORD PTR [rax+rax*1+0x0]
0x55555555060 <_start>:      endbr64
0x55555555064 <_start+4>:  xor   ebp, ebp
0x55555555066 <_start+6>:  mov   r9, rdx
```

こんな感じで今度はputs関数の中で呼ばれている_GI_IO_puts関数に入る
など何回も行くと結果的にシステムコマンドのwriteコマンドを実行させてい
ることがわかります！実際にやってみると面白いですよ！是非やってみてくだ
さい！

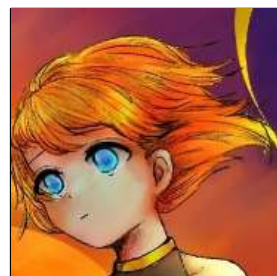
このように私たちが今までprintf()などを簡単に使っていると思いますが、実
際にこの中にどのようなプログラムが入っているのかを考えたことがなかった
と思います。GDBを駆使すればいろいろな関数の中身を見ることができるの
では是非GDBを使ってみてください！

ではよいGDBライフを！

一週間でゲームつくろう

学部/回生 情報理工学部 SNコース 一回生

名前 TZ



はい、タイトルの通り一週間でゲームを作ろうというか作った報告みたいな何かを書きます。

unityというゲームエンジンはご存知でしょうか。簡単にゲームが作れると話題のあれです。C#でスクリプトが書けます。いろんなプラットフォーム向けにビルドできてすごいとかあれこれ書くとUnityのコラムになってしまうので端折ります。

さて、そのUnityで作られたゲーム専用のフリーゲームサイト『UnityRoom』では、WebGL形式でビルドされたゲームをブラウザ上で遊べます。それだけでなく、UnityRoomは定期的に**Unity1Week**という1週間でゲームを作って遊ぼうという素晴らしい企画をやっています。毎回お題となる単語が発表され、一週間で作っては公開し相互プレイ、相互評価を行うお祭りです。

ということでそれに参加し、ゲームが生成されました。

<https://unityroom.com/games/proliferationescape>

以上、終わり。

としようと思ったのですが誌面が埋まっていないので続けます。

Q1.どうやって1週間でゲームを作るのか

A1.企業様が大勢で、大金をかけて作っているようなゲームを想像したら爆散します。タイトル→数分のミニゲームのみのシンプルさで作る。

Q2.どんなゲームが人気か

A2.お祭りで、多くのゲームが投稿されている都合上『話題性』、『分かりやすさ』が重要です。上のものは若干改善の余地があり、まず誰も注意書きなど読まないの注意書きに期待せず、ゲーム内チュートリアルにも期待せず、**触れば分かるUI**にする必要があります。また、話題性も大事で、バカゲーというか、見た目でなんかおもしろそーってのがあるとウケてます。

Q3.ゲームとしての体裁の整え方

A3.始まりと終わりが大事です。まずテキストでもタイトルがあると分かりやすいです。また、時間制限とかスコアアタックとかでいつ終わるか？を分かりやすく、終わったよーってするとゲームっぽくなります。

さらに、1週間で企画、実装ってやると時間がないためゲーム性が死滅しやすいです。初参加の作品とかがまさにそうで、作ってて鬱になった上面白さがゼロでした。その回避のためには**ランキング**、**Twitter共有**が便利です。

ランキング

UnityRoom用の簡単ランキングアセットが公開されています。手間もそこまで掛かりません。なんらかのスコアアタックをメインに据え、それにランキングを導入するとあら不思議、とってゲームっぽくなります。ヒトの向上心、上に立ちたい、勝ちたいという欲求も刺激でき、**再プレイ性が高まります**。

Twitter共有

これもアセットが公開されています。UnityRoom自体がTwitter認証なので、みんなアカウントを持っています。なぜ大事かというと、特にゲームジャム中はみんな**Twitterで盛り上がっているから**です。進捗報告、プレイ感想などなど眩きまくってますので、共有ボタンを作るとプレイ感想の導線にもなり、画像共有をつけるとさらに目を引きます。

スクショ最適化という概念があり、見ただけで凄そう、面白そう、ナニコレっていう点を作っておいて、そこを共有画像にすると素晴らしいらしいです。所詮口コミとか、プレイヤー同士による情報共有でないとゲームは広まりません。

ということで以上のことをある程度踏まえて、ゲームジャムが開催されていないのに勝手に1週間で作ったゲームも貼っておきます。

<https://unityroom.com/games/sweettrick>

いっしゅうかんでげーむつくろう。(これって本当に情報系のコラムなのか……?)

ぷよぷよ敵AIの歴史

学部/回生 情報理工学部・実世界情報コース 2回生

名前 Fukada



こんにちは！虚無に暮らしているFukadaです。

ぷよぷよは息の長いゲームです。もうすぐ初代発売から30年が経とうとしています。当時の時代背景などが如実に反映されているため、歴史を追ってみると面白いです。今回は敵AIを中心に初代と最新作でどのような違いが起きているか見てみます。さらに公式とは別の、有志でのAI開発についても見ていきます。

ー初代ぷよぷよー

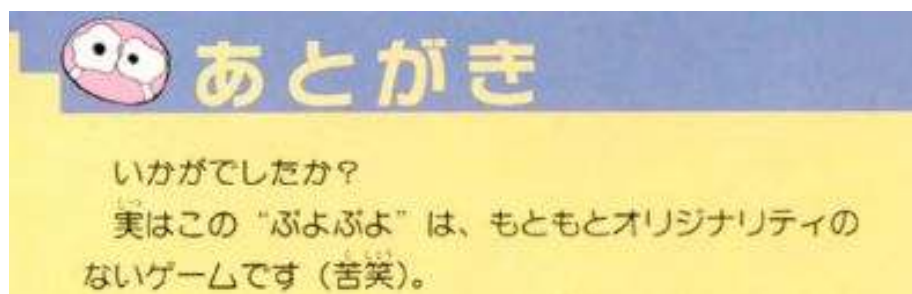
1992年非常にスペックが低かったのと、ノウハウがないため強いAIを作ることはできませんでした。よって強さを求めず特徴づけることに重点を置いています。このゲームは連鎖ボイスのさきがけでもあります。

- ・全く回転を使わないスケルトンT
- ・ゲーム開始直後に最も左と最も右の列に積み続けるハーピー積み
- ・適当にぷよを右3列に並べ運で連鎖を狙うかえる積み
- ・このゲーム中最高速度でぷよが落ちてくるサタン

彼らの仕様は後の作品にも受け継がれることになります。相殺という根幹システムがなく、AIは大きな連鎖を組む必要がなかったため、複雑なAIを作らなくても成り立っていました。



ぷよぷよの難易度はAIの賢さよりもぷよの色数と落下速度によって調整していました。初代ぷよぷよはテトリスの影響を非常に強く受けているため、テトリスと同様に落下速度での調整が主となっています。



—ぶよぶよ通—

相殺システムが採用されたことにより大連鎖を基本とした戦略性が生まれました。現在でも大会が開催されるほどに完成されたゲームといえます。このころの敵キャラクタは相殺に対応することはあまりないですが、初代よりも連鎖力が上がっています。単純なスペック上昇ですね。

—ぶよぶよeスポーツ—

現状の最新作です。新規プレイヤーもさることながら古参プレイヤーが増えたため、AIの強さにもバラエティが必要になってきます。初代のような置き方による個性の付け方はなされていません。ルールベースでぶよを置く場所を決めており、そのパラメータがキャラによって異なることで難易度調整をしているものと思われます。技術が初代と比べても格段に向上した代わりに、AIとしてのキャラクタの個性は減少した形になります。

—有志によるAI開発の動き—

有志でぶよぶよのAIを開発しようという試みが2016年前後に盛んになりました。目的は人類を倒す強いAIを作ること。ぶよぶよの状態数は 10^{116} なので研究の題材としても適しています。チェスが 10^{50} 、将棋が 10^{71} だと考えると非常に大きい数字であることがわかります。さらにぶよぶよは落ちものパズルであることから、囲碁や将棋以上に速さが求められます。結果として有志が作成した強いAIは昔から多くのゲームAIで用いられているルールベースの決定方式を採用していることがほとんどです。これでも最上位勢クラスの強さがあります。DeepLearningでは状態数が多すぎたり、最適解を出すには掛けられる時間が短すぎるのです。しかし1018年ごろからDeepLearningを用いたAIの開発もチャレンジされています。PCスペックの上昇とルールベースに限界を感じているからだそうです。現状あまり芳しい成果は出ていないようで、DNNを適用できなかったとされる論文も出ているほどです。これはぶよが4つ以上結合したら消えるという仕様上、畳み込み演算が機能しにくいことなどが挙げられます。成功している例もみられるため実装の難易度が高いのでしょうか。公式とは違うアプローチをとっているため今後期待ですね！

—最後に—

多くの人に遊ばれる商用ゲームのAIは、スペックが十分間に合っている現代でも、強さよりも“それらしさ”が大切になることがよくわかります。絶対勝てない敵がいても、プレイヤーは面白くないですからね。初代は特定のキャラクタらしいAI、現代は人間らしいAIが求められていることがわかりました。強さを求めるのは有志の開発に期待がかかります。ゲームAIは将棋や囲碁などが注目されがちですが、人類を超えたことで一旦落ち着いたと見るべきでしょう。これからは、まるで人間であるかのような振る舞いができるAIへの注目が今まで以上に高まりそうです！

ツイートをベクトル化してみた結果w

学部/回生 情理 SN 2回生

名前 堀田 隆成



最近原神にハマってて、特に甘雨可愛すぎはよ実装しろってうごめいてる堀田です！

こっからどうやってタイトルにつながるといいますか？

原神

↓

Gensin

↓

gensim

↓

Word2Vec

素晴らしい導入ですね！（ドッ！わー）

1. やりたいこと

今回はgensimというライブラリをつかって自分の過去のツイートに含まれる単語をベクトル化し、自分の中での単語の類似度を図って遊ぼうと思います。

うまく行けば

王一男+女=女王

みたいなことができるようになります。
というかこれなんか昔流行ったらしい。

2. 準備

過去のツイートを取得するには自分のアカウントの設定→アカウント→Twitterデータより取得することが出来ます。

しかし今回は時間がなかったためふりかえったーというサービスを使い、過去3200件のツイートを取得し、その内容をtxtファイルに保存しました。

3. 手順

最初に自分のツイートのファイルを読み込む

↓

ファイルの不要な部分(ツイート日時等)を削除する用にプログラムする

↓

Word2Vecでモデルを作る

4. 使い方

プログラムを実行し、単語を入力することで関連性のある単語を出力してくれます。

以下は結果の一部です。

```
単位
落とす 0.8228988647460938
取れる 0.7354998588562012
今期 0.6613162755966187
取る 0.602971076965332
来年 0.5690203309059143
やめる 0.5623185634613037
プレゼン 0.5594778656959534
謎 0.5584843158721924
対面 0.5504499673843384
決める 0.4688061475753784
```

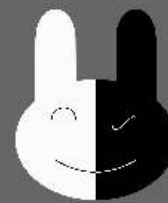
```
切る
髪 0.8199801445007324
フリーエ 0.5844773054122925
知り合い 0.5480997562408447
取れる 0.5419444441795349
わら 0.487948477268219
アホ 0.478755384683609
ミクフィア 0.4691268801689148
教科書 0.45780208706855774
講義 0.4557580351829529
来年 0.44911885261535645
```

5. まとめ

競馬AI班の活動の一環で買った本に書いてた内容で競馬AIとは関係なくて飛ばしてたのですがとても面白く、自然言語処理に興味をもちました。

(サンプルコードが動かさなくて阿部くんの一部手伝ってもらったのは内緒)

祝Logic Pro X アプデ



学部/回生 情報理工学部・実世界情報コース・2回生

名前 momonga

こんにちは。momongaです。今年はDTMについてつらつら話していこうかなと思います。

今年の5月ごろにLogic Pro Xの大型アップデートがありました。リリースから初と言われるほど規模が大きいアップデートでした。変更点がいっぱいあって説明しきれないので、いくつかだけ紹介して、それを使って曲作ったたわってというお話です。

1.samplerの変更

samplerが従来のやつから変わりました。以上

2.Quick-samplerの追加

1は正直どうでもいい。個人的には今回のアプデで一番いいんじゃないかなあと思ったり思わなかったり。

UIはこんな感じ



こいつ何もんか簡単に説明しますと好きなオーディオファイルをmidi感覚で入力できるツールです。オーディオファイルをこいつに打っ込むだけで音程を変えたりできるのでめっちゃ便利す。

こいつの良いところはこれだけに止まらないんすねえこれが。

SLICEってモードを選択するとこんな感じになります。



何をしてくれたか言いますとオーディオの良い感じのところに分けてくれてます。分ける箇所は自動で入りますがもちろん手動で変更が可能です。分けた各オーディオの子がmidiの鍵盤に割り当てられます。ボーカルチョップのような音がすぐ作れて入力も鍵盤押すだけなので感覚的にできて便利です。最初に説明したモードよりこのSLICEモードの方が真価を發揮できる感ありますね。

3.iPadとの連携が良い

Apple製品なんでもあって気もしなくもないですが、実はApp StoreにLogic Remoteっていうアプリがありまして、これの連携が良い感じになった気がする。（アプデ前使ったことないから知らん）Filter系が聴きながら指で調整できるので割と良い感じ。



後はmidiの代わりとして使えたり（これはあんましないが）、ドラムとかはmidiでやるよりやりやすいので割と使ったりします。

で。ここからはおまけですが、アップデートでテンションが上がり、酔っ払った会誌担当がTwitterで叫んでたのでそれを素材にノリで曲を作りました。

https://soundcloud.com/momonga_momo/hottadeyatter

今回説明したような感じで

Quick-samplerにその素材をぶちこみまして、SLICEモードで切って今回はパーカッション的な感じで使ってみました。あと最後にある申し訳程度の雑コーラス的なサムシングもこのQuick-sampler使ってやっています。これはただ単にぶち込んだ素材をコードみたいに入力してます。

こんな感じで新しいLogicええでっていうお話でした。

困ったらPyてよn

学部/回生 情報理工学部

名前 mizugoro



はじめましてmizugoroです。最近アイコン作ったりしてます。右上のアイコンも自分で作ったもので、GUMIちゃんをモチーフに作りました。可愛いでしょう〜。ほかにもミクちゃんモチーフや、ウナちゃんモチーフなど、ボカロ以外のアイコンも多数作っております。

そんな感じであんまりプログラミングが強くないmizugoroですが、今回は困ったらPyてよnということで、Pythonを使ってTweet Botを作ってきました。

必要なもの

1. Twitter API
 2. やる気
- これだけ。

手順

1. <https://developer.twitter.com/en/application/use-case> にアクセスして Twitter API を取得します。Product > Apply for a developer account > Making Bot の順に進んでいくと、このようになんで Twitter APIが必要なの?って聞かれます。

ここはdeepL使って適当に埋めてもらったら多分通ります。書けたら Next > Look good!で返事を待つだけ。

2. Pythonでプログラムを書きます。自分は Tweepy というライブラリを利用しました。ざっくり説明すると、PythonでAPIを取得してそこから Tweetするプログラムを書けば Pythonプログラムからツイートできちゃいます。とりあえず、文字列を入力してそれをそのままツイートするプログラムを書きました。

```
#!/usr/bin/env python

import sys

sys.path.append('/usr/local/lib/python3.6/site-packages')

import tweepy

CONSUMER_KEY = "[twitterAPI]"

CONSUMER_SECRET = "[twitterAPI]"

ACCESS_TOKEN = "[twitterAPI]"

ACCESS_TOKEN_SECRET = "[twitterAPI]"

auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
api = tweepy.API(auth, wait_on_rate_limit = True)

# ツイート内容の文字列を入力
message= input("tweet>>")

# ツイート
api.update_status(message)
```

あとはここを読んで、欲しい機能を追加しましょう。

<https://github.com/tweepy/tweepy>

追加で、Linuxのcrontabやwindowsのタイムスケジューラを用いて定期実行を可能にすれば、決まった時間にツイートする Tweet Botが完成します。

[まとめ]

Twitterに限らず Botは利用の汎用性が高く、作ってからも機能を追加したりできるので自分みたいな初心者にはおすすめです。素敵なプログラミング生活を、どうか。

お前らのPCはダザい

学部/回生 情報理工・実世界情報2回生

名前 @averak_jp



はじめまして、2回生の健太郎と申します。普段は音声処理やソフトウェア設計を扱っていますが、今回はおすすめのPC設定についていくつか紹介したいと思います。

※ Macに限る

◆ 神アプリ① - BetterTouchTool

Macにタッチバーが搭載されてから4年が経ちましたが、未だに不評であるように思えます。正直なところ、私もタッチバーは微妙であると感じています。そこで活躍するのがBetterTouchTool (以下, BTT) という神アプリです。有償ではありますが、タッチバーやショートカット, その他様々なカスタマイズをすることができるのです。

現在, 私は下記のようなタッチバーを使用しています。結局使うことはありませんが, デフォルトに比べ見た目は改善されているので良しとしましょう。(⌘+⇧+6でタッチバーのスクショが撮れます)



前述した通り, BTTはタッチバーのカスタマイズ以外にも様々な機能を提供しています。昨年escが物理に戻りましたが, それ以前はVimmerが「escを物理キーに戻せ！」などと騒いでいた時期もありましたね。物理であろうが液晶であろうが打ちにくい位置まで指を伸ばすのは非効率なので, 私は⌘+eにショートカットを割り当てています。このように柔軟なカスタマイズを可能とするBTTですが, 中でも強くお勧めするのが

- ⌘ + 矢印でWindow分割 (Win + 矢印でやるやつ)
- ctrlをダブルクリックするとcaps lock

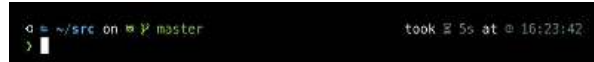
といった設定です。興味があれば他の機能もぜひ調べてみてください。



友人がMacを購入したら決まってお勧めするのがRunCatです。RunCatとは、CPU負荷に合わせて猫などのキャラクターのアニメーション再生速度が変化するツールです。

最初はジョークアプリだろうと思っていたのですが、だんだんと愛おしくなり「黄金のネコ」に980円課金したほどです。一部キャラクター以外は無償の範囲で使えるので、検討してみたいはいかがでしょうか。

◆ シェルをオシャレにする



皆さんはシェルについてどのようなイメージを抱いているでしょうか？真っ黒背景に白い文字，正直使いにくいと感じている人がほとんどでしょう。そこで，デフォルトシェルであるzshをオシャレかつ簡単にカスタマイズする方法を紹介したいと思います。

プロンプトのテーマは色々ありますが，私はpowerlevel10kを使用しています。現在のgit branchや各コマンドの実行時間及び終了ステータスなどが表示されるオシャレなテーマです。

```
$ git clone https://github.com/sorin-ionescu/prezto ~/.zprezto
$ cd ~/.zprezto
$ git pull && git submodule update --init --recursive
```

※ 追加でzshrcの編集及びnerd fontsのインストールが必要



◆ キー配列を変更する

詳しくは→

皆さんが普段使用しているQWERTY配列は，タイプライターの時代にあえて打ちにくく設計された背景があります。そのような配列を訳もなく使うのは非合理的なので，自分好みにキー配列をカスタマイズしましょう。キー配列の変更にはKarabinerを使用することをお勧めします。BTTでも可能ですが，公式は非推奨なのでこちらを使いましょう。

下記は私が使用しているAVERAK配列です。「きゃ・きゅ・きょ」以外の打鍵には同じ指を連続して使う必要が無く，親指でenter, back spaceを入力することができる便利な配列だと思いますので，参考にしてみたいはいかがでしょうか。



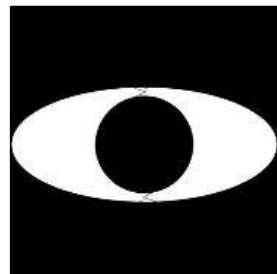
デバイスの観点で考える

学部/回生

おりこうな学部2回生 ドラえもんを作りたい

名前

USA(ウサです。英語はキライ)



この会誌を描いている約10日前、Appleの新製品が発表されました。ムーアの法則もビックリのSoCが発表されましたね。(10日前、つまりところ執筆現在=提出日という事には気づかないでください。)ただ、つい最近、それよりずっと有能な"媒体"を、見つけましてね。あの、え〜と、

“紙”

ってご存知ですか？第4次産業革命なんて言われるこのご時世だけど、なかなかやっぱり紙というものは偉大ななあ、と思うわけです。僕はロボット工学の人間なので本職の会員たちと比べてソフト面でのアドバンテージが少ないからハード面の話題に逃げているという事には気づかないで欲しいのですが、冷静に考えればデジタルデバイスが平気で紙のみが破損する環境の少なさに気づくのでは？スペック比較なんてデジタルデバイス同士の内輪もめに過ぎないということに気づくでしょう!!以下、紙のアドバンテージをつらつらと書きなぐり～

その1:丈夫

なんじゃそりゃ、といわないで欲しい。一人1台スマホという名のデバイスを持ち歩くこのご時世、落とした時に媒体に記憶された情報が破損しないという事はとっても重要。MIL(ミル)試験に代表される、自由落下試験、耐振動試験、は紙に敵うデジタルデバイスは無いのではないだろうか(打鍵試験は紙が不利かも)?こんなことを言っていると「軽いから当たり前じゃん」等と言われる事もあるのだが、軽いという事は位置エネルギーを蓄えにくいという事。“軽い=壊れる機会が減る”というのはロボット工学等においては重要な発想だったりする。

その2:記憶&出力の兼用

全ての紙は、記憶と出力の両方の機能を兼ね備えている。ノートパソコンだって各部品を寄せ集めているに過ぎなくて、一部不調なんてことはざらにある。紙媒体が偉大なのはすぐにメモを直感的に書き込めること。これは記憶と出力の二つの機能を兼ね備えているからこそできることといっても過言ではない。その上、「紙」という媒体でさえあれば機種やバージョン、メーカーに関わらずこの機能が保証されているという安心感も大きい。

その3:駆動時間

これは説明不能であろう。文庫本に対して残量バッテリーを気にする、出先でのコンセントの有無を心配するという事は一切ない!Apple Eventでは駆動時間の長さを一つの肝として自慢してたけど、紙が期待されている機能を発揮するのに時間制限等事実上は無いとっていいだろう。

その4:単純構造

壊れているか否か、壊れているならどこがどれほど壊れているのが基

本的にメーカーでなくとも瞬時に判断できる。これによって（値段的な要因もあるが）壊れた際に個人で修理か買い替えるかの判断がしやすい。

その5：世界規模の流通量

なんといってもこれが大きい。紙は世代、地域問わず使用法を学び、計算などの際に至っては運用自体が思考の一部とってはばからない。嘘かホントか、Twitterの創始者ですら上場の際に思い返したのが創設当時の手書きのアルゴリズムだったとか。老若男女誰でも使えるからいちいち使い方の説明も不要。歴史の長さのおかげで入力装置も鉛筆、ペン、筆と多種多様なうえ、入力の手法も多岐にわたり、それぞれが市場や芸術といった文化となっている。（この点はネット上で生まれた文化も同等か）。恐らく、世のどの電子機器よりユーザーフレンドリーであること間違いなし。扱いが知られているからこそ、コンピューターよりも多くの販売店での売買が可能。その上、「紙」という媒体でさえあれば、これらの仕様は基本的に変わらない。パソコンじゃあまあこうはいくまい。

これらの事から、紙というものは古くから存在する人類にとって最も偉大な発明品の最たるものと言えるだろう。その本質が持つ素晴らしさは、新参者のデジタル機器的な尺度で測っても簡単に揺らぐものではない。

ほぼ、唯一デジタルデバイスに軍配が挙がるのは記憶容量と情報の電子化のハードルの低さかな？情報の電子化はスキャナーや文字認識、画像認識の技術の進化によって少しずつ解決されているけど、紙の記憶容量というのはなかなか拡張するのは難しい。聞くとところによるとどうやら市販のマイクロSDカードで最大の容量の物は2TB。A4の紙に解像度:350dpiでA4サイズの画像を記録できるとすると、記録できるデータ量は3008666068bit。それに対して2TB=17,592,186,044,416bit。

A4サイズはSDカードのピッタリ378倍。それぞれの厚さ、単位面積の重さが同等で、紙が両面で記憶容量が2倍であること等を考慮すると。。。

市販の電子記憶媒体（マイクロSDカード）は紙の

2,210,986.771542456倍容量が大きいということになる。約220万倍である。1キロバイトと2ギガバイト位の差と言える。“桁違いの差”なんてままあるこのデジタル社会といえどもそれなりに大きい差だな。まあ、ここぐらいはデジタルに譲ってやってくださいよ。紙様。

まとめ

そもそもといえば一時期のスマートフォンにあったスタイラスを初め、文字情報の処理とかの諸々の機能、つまり“パーソナル”コンピュータが目指した能力は計算機の機能に加え紙の様な使いやすさ。紙を目指してるPCが紙に敵うわけ無いということは冷静に考えれば当たり前の事ではあるのだけど、一度原点に立ち返ってみると当たり前の事を多少は面白い視点で語れるんじゃないかな？って話でした。おしまい。

SwiftUIでView作成

学部/回生 情報理工学部 SNコース 2回生

名前 Okugawa Kanta



お疲れ様です。初めましておくがわです。
日々学習、大切なことですね。ではでは...

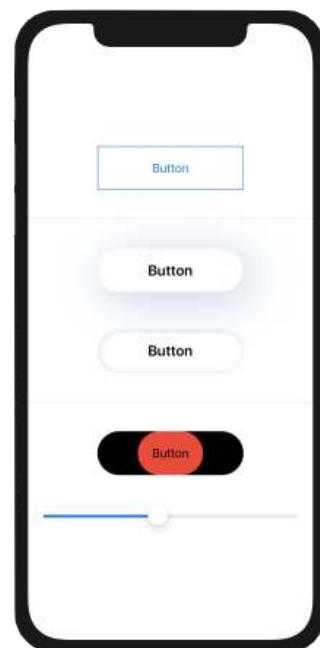
アプリ作りたいと思いSwiftを勉強し始めました。WWDC2019で発表されたSwiftUIを使っています。SwiftUIは新しいので仕様変更があってイライラしています。鬱陶しい。まあ仕方がない。

色々なカスタムボタンを作ってみました。

1つ目、ノーマルなやつです。Buttonとborderで囲っただけの一番シンプルなButtonです。

2つ目、流行りのNeumorphismのやつです。shadowで影をつけたりしました。あまり上手ではないですが。

3つ目、スライダーで赤い部分の大きさを変更できるButtonです。本当はPathを使うべき。



リストを作ってみました。

まあボタン以外にもよく見るiOSアプリに使用されているリストなどが簡単にできるのがSwiftUIでして、
たったこれだけ↓

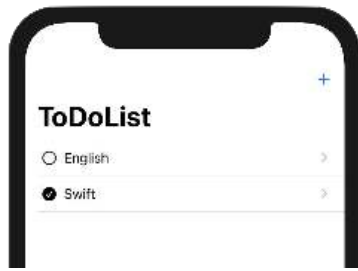
```
List{
    ForEach(0..
```



こういう良いところもあるやつなんです。
こういうところもある良いやつなんです。

ToDoListを作ってみました。

こんな感じにToDoListを作ることもできちゃったりします。各項目、
押したら右側に画面が遷移します。



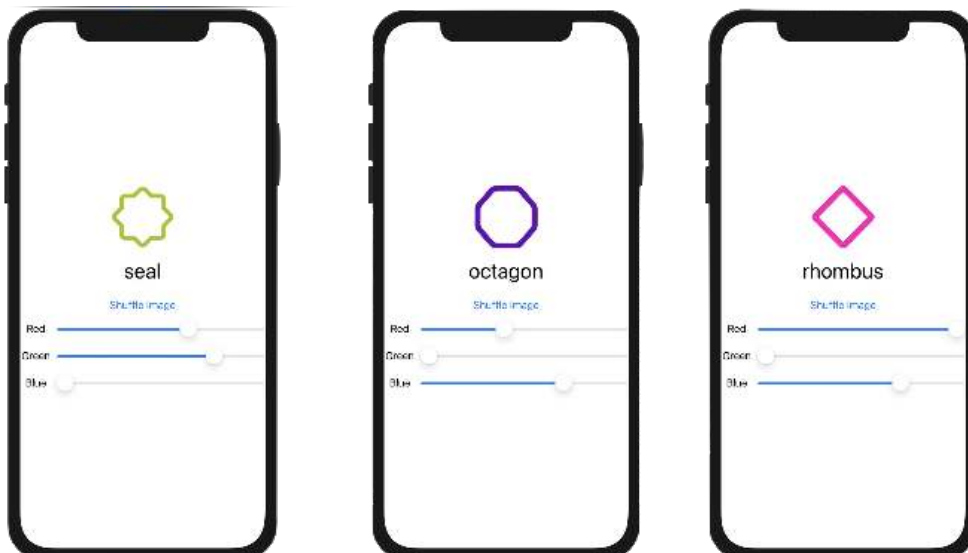
上記の左側のコードもたったこれだけ。（実際はこれだけではないがほぼこれ）

```
NavigationView{
  List (todoLists) { list in
    NavigationLink(destination: ToDoContentView(todoList: list)){
      ToDoListRow(todoList: list)
    }
  }
}
```

ガチャ的なのを作ってみました。

こんな感じで、ランダムに画像を変更するボタンも、RGBで色を変更するスライダーも、とにかく簡単につくれます。

画像もAppleが推奨しているSFSymbolsから呼び出しています。



Apple側で準備してくれているやつが良いので使いやすいです。やっぱりSwiftUIはいいやつ。Appleはいいやつ。

今回はView部分だけだったので、機会があればModelの部分やアプリの設計パターンの紹介も。
以上。

Gmailの既読を情理っぽくつけたい!!!



学部/回生	情報理工学部 SAコース 2回生
名前	桐井優実

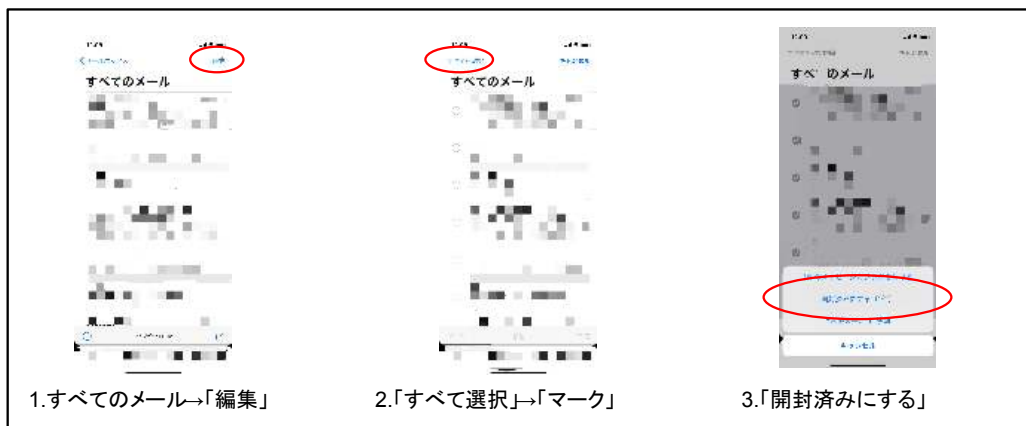
ことしの目標は、とりあえず同期の顔と名前を一致させることでした。むりそうです。

そんなことは置いといて、今回はGmailの未読メッセージをGAS(Google Apps Script)を使って一気に既読にしたはなしをします。

私は未読メールをためがちな人間なので、気がつくとアイコンがこんなことになってます→



一応、次の手順で「全部既読にする」ということはできたりします(iPhone)



が、今回はタイトルの通り「ちょっとは情報理工学部生っぽいことしてみたい!!」と思ったので次のような手順で未読メールを一気に既読にしてみました。

1. 「Gmail 既読 スクリプト」で検索

いろいろ出てきます。



2.よさげな記事を選ぶ

いくつかざっと見て、見比べながら選びました。

シンプルな既読機能なら5行程度で実装できるそうですが、そのままだと

- ・一気に既読にできる数が限られてしまう
(関数を使用すると一回の実行あたり500件までしか既読にできない)
- ・スクリプトの実行時間の制限をオーバーしてしまって途中で止まる
(一回のスクリプトの処理が5分以内という制約がある)

という問題があるらしいので、その点を解消してくれるスクリプトを載せてくれているサイトにしました。

参考サイト：Google Apps ScriptでGmailの大量の未読を一掃する - ふくらみ
<http://fukurami.hatenablog.com/entry/2017/10/01/030917>

3.実装してみる!!!

①script.google.comにアクセス

②「APPS SCRIPTを作成」で、出てきた画面に参考サイトのスクリプトをコピペして「実行」

こんなかんじ→



```
9
10
11
12
13 //取得するメールID
14 var mailIds = [];
15 var mailResp = PropertiesService.getProperties();
16 var mailTriggerId = mailResp.getProperty('mailTriggerId');
17 var triggerId = mailResp.getProperty('triggerId');
18 var trigger = ScriptApp.getTrigger(triggerId);
19 mailResp.removeProperty(triggerId);
20 mailResp.setProperty('mailTriggerId', mailTriggerId);
21 PropertiesService.setProperty('mailTriggerId', mailTriggerId);
22 break;
23
24
25 //メールを一掃する
26 function deleteUnreadEmails() {
27   //取得するメールID
28   var mailIds = [];
29   var mailResp = PropertiesService.getProperties();
30   var mailTriggerId = mailResp.getProperty('mailTriggerId');
31   var triggerId = mailResp.getProperty('triggerId');
32   var trigger = ScriptApp.getTrigger(triggerId);
33   mailResp.removeProperty(triggerId);
34   mailResp.setProperty('mailTriggerId', mailTriggerId);
35   PropertiesService.setProperty('mailTriggerId', mailTriggerId);
36   break;
37 }
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

4.ゆっくりまつ

処理速度はそんなに早くないので気長に待ちます

5.きれいになったー



わーい

あとがき

今回は先人の知恵をそのままコピペで利用させてもらいましたが、もっと発展させていけば「特定のアドレス以外は既読にする」とかいろいろ応用が利きそう。たのしいね。

おしまい。

Pyxel		アイコン
学部/回生	情報理工学部・SA・2回生	
名前	小柳	

情報理工学部生なのにプログラミングから逃げ続けている小柳です。Python向けレトロゲームエンジンである、Pyxel（ピクセル）を紹介していきます。

- ・ Pyxelとは

Pyxelは昔ながらのドット絵のゲームを作るためのPython用ライブラリです。オープンソースでGitHubで公開されています。

<https://github.com/kitao/pyxel/blob/master/README.ja.md>

使える色は16色，同時に再生できる音は4音までなど，シンプルな使用になっているため初心者でもとっつきやすいです。

また，作者が日本人のため日本語のマニュアルもありますし，なんなら作者本人に日本語で質問できちゃいます。

- ・ インストール方法

Windows：Python3（3.6.8以上）がインストールされた環境で，pipでインストールできます。コマンドプロンプトを開いて

```
pip install -U pyxel
```

Mac：Homebrewインストール後に

```
brew install python3 gcc sdl2 sdl2_image gifsicle
pip3 install -U pyxel
```

Linux：下手なこと書けないので，各自で調べてみてください。

- ・ 実際に作ったもの

夏休みに数時間で作ったChromeの恐竜ゲームのパチモンです。小学2年生ならギリ楽しんでくれそうな出来栄です。もうちょっと作りこみたいとは思っているんですがやる気がおきず数か月放置してます...



- ・ 最後に

詳しい使い方などはGitHubに丁寧に書かれていますので，そちらをご覧ください。ほんとに簡単にゲームを作れるので興味があれば試してみてください！

MMDモデル作成について

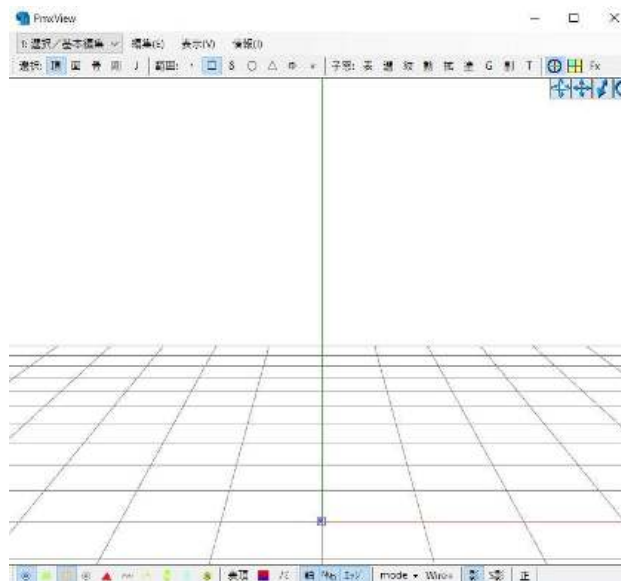


学部/回生 情報理工学部・実世界情報コース 2回生

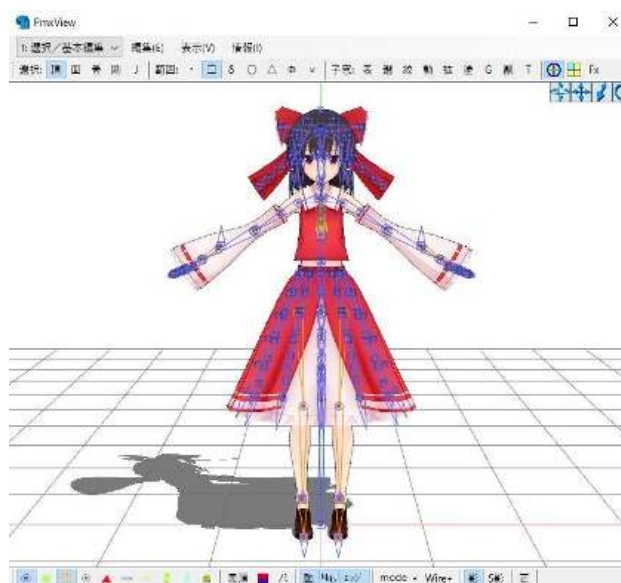
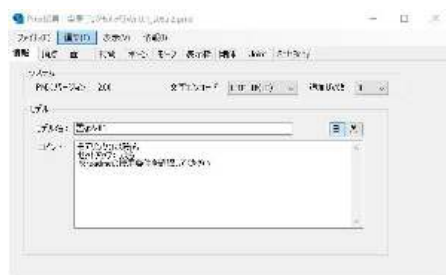
名前 新藤 尚輝

最近、私はモデリングをしている。作成したモデルはMMDに読み込めるようにしている。今回は、MMDモデルを作る際に便利なソフトウェアpmxEditorについて簡潔に話す。

起動した直後、以下のようなウィンドウが二つ出る。

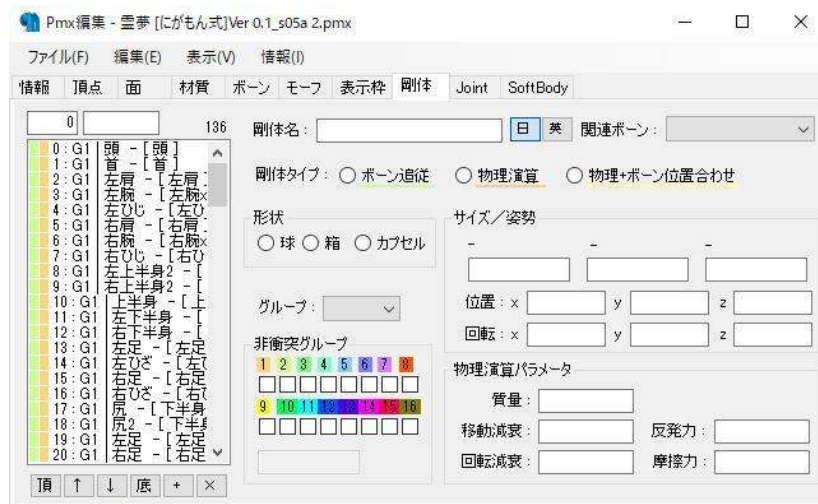
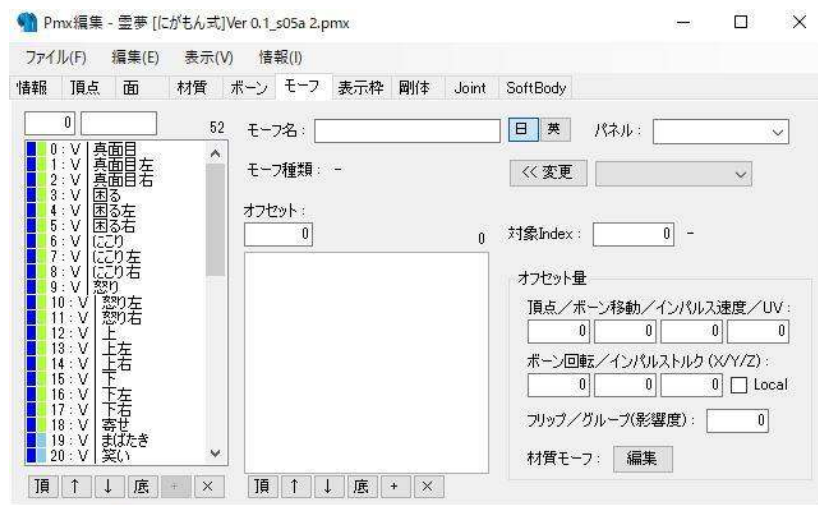


右側のウィンドウがモデルが表示される画面で、左側のウィンドウは具体的に操作する際に用いる。左側の画面はモデルを読み込んでいないため、何も表示されていない。ここにモデルを表示すると以下ようになる。今回はにがもん式霊夢を使わせてもらう。



このように、読み込むと詳細が表示される。そして、このソフトの良いと

こころはMMDモデルの調整に特化しているところだと考えている。MMDモデルではモーフというものを設定することで簡単な操作を行える。モーフを用いると簡単に口パクさせることもできる。さらに、剛体やjointを設定して簡単に物理演算を行える。これらの設定を簡単にできる事が一つの利点だろう。



加えて、プラグインもかなり多く他の形式のモデルも取り込める。このように、様々な用途で用いる事ができるので、モデラーの方でMMDモデルを作ってみたいという方は、一度使ってみてほしい。

歌詞可視化プログラムを作ってみた

学部/回生 情報理工学部・SNコース・2回生

名前 らん（星名 藍乃介）



はじめに

『カシカシカ』。

どこか語呂が良いから、という単純な理由で作ったプログラムです。がアーティストごとに、また曲ごとに特徴的な語彙が見られるので、完成して日が経った今でもお気に入りのプログラムです。（でもこれ実はLTでも紹介したネタなんです。が、他にネタも思いつかんしなあ、でも使いまわしもなんかなあ...ううん...まあ.....しかし歌詞可視化しかなあい!!!）...ということで作ってみた系の記事になります。

概要

アーティスト名と曲名を入力に、歌詞文字列を視覚表現した画像を出力する、歌詞可視化プログラム

出力例

米津玄師『アイネクライネ』



（わたしではなく
あたしなところすこ）

あいみょん

『貴方解剖純愛歌 ～死ね～』



（字面にするとやべえ）

使用技術

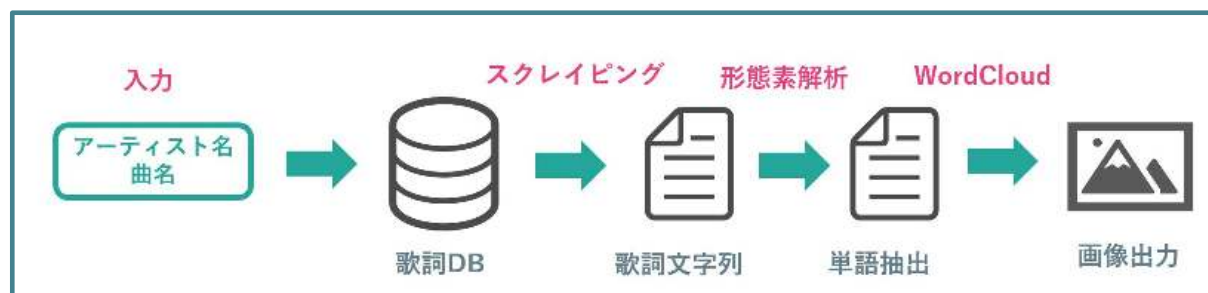
- Python3：いっぱい損しないための言語
- Requests：歌詞DBサイトと通信するのに使う
- BeautifulSoup4：html解析して、歌詞文字列を取り出すのに使う
- MeCab：形態素解析して歌詞単語を取得するのに使う
- WordCloud：文字の視覚表現を行うのに使う
- めげない心：不具合と闘うのに使う

環境構築

MeCabは辞書をインストールするので少し時間がかかりますが頑張っ
て頑張ります。ソースコードはGitHubにあげてますので適宜参照されて
ください。(https://github.com/Ran350/Lyrics-visualize)(フォローもお忘れなく)

フロー

フローは、アーティスト名と曲名が入力されると、スクレイピングし
てその歌詞を取得し、歌詞を形態素分析して品詞選択し、単語をその出
現頻度に応じた大きさで並べた画像を生成し、出力するという感じです。



スクレイピング

入力されたアーティスト名と曲名をもとに歌詞DBサイトをスクレイピ
ングしてページを取得してきます。

このとき、サイトの利用規約やサーバへの負荷に注意します。

形態素解析

形態素解析とは、文章を、形態素とよばれる意味を持つ最小単位の文字
列に分割（分かち書き）し、品詞などを判別する作業のことです。

歌詞可視化プログラムでは、最終的に歌詞単語の視覚表現を行うので、
なるだけ単体で意味をもたないような品詞や単語は除外します。この選択
の仕方が適切に可視化できるかの鍵になる重要な工程です。本プログラム
では、品詞のうち付属語である助詞と助動詞、頻出だが意味のうっすいサ
行変格活用「する」、そして「いる」「ある」などの一部の単語を除外し
て単語の選択を行います。

画像生成

頻出語を頻度に比例する大きさで雲のように並べた画像の生成は
、WordCloudモジュールを用いると簡単です。日本語フォントパスを指
定し、WordCloud()メソッドに歌詞文字列リストを渡すだけです。

おわりに

今回はスクレイピング、形態素解析、文字の視覚表現で歌詞可視化を行
いました。Pythonの豊富なモジュールの威力に驚きながら、中学国語で
習った品詞分類の登場に懐かしくなった開発でした。ではまた。

固有値、固有ベクトル

学部/回生 情報理工学部・SAコース 2回生

名前 たつや

こんにちわ！課題多きこの時期にハッカソンとか打ち上げとかを入れまくったせいで地獄を見ているたつやです。(ハッカソンでは反省点が数多見つきり、絶賛猛省の最中でございます)
今年前期の多変量解析の講義をきっかけにデータ解析の泥沼に足を突っ込み、白目を向きながら楽しく暮らしています。
複雑な内容をかみ砕いて話すのは苦手なので、コレが出来なければ話にならない！という行列の固有値解析のみについて解説します。(赤文字で書かれている部分は読み飛ばしていただければ幸いです)

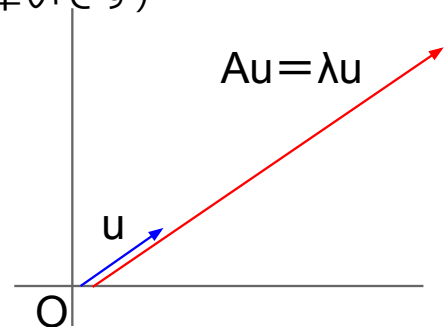
固有値、固有ベクトルとはなんぞ？

行列Aについて、 $Au = \lambda u$ となる

λ を固有値、 u を固有ベクトルと呼ぶ

※教科書では $\det(A - \lambda E) = 0$ と書かれる

※ \det は行列式



実際にこの固有値と固有ベクトルを求める際に上の式を満たす u, λ を探すには工夫が必要です(コンピュータに行列式の計算をさせるのは時間がかかりすぎる)

固有値、固有ベクトルの求め方

名だたる数学の天才達が編み出した方法を幾つか紹介します。

(※幾つかのアルゴリズムを通すことで固有値が求まる場合があります)

1. べき乗法(解が求まるまでがおそい)や、ヤコビ法(実対称行列の場合に有効)を利用して求める
2. Householder法→(二分法or分割統治法orQR法orQL法)の順に計算を行って求める(どこを切っても理屈が複雑！まるで金太郎飴)

具体的な手法の説明

べき乗法が一番手軽で簡単だと思うのでべき乗法の説明を行います。

1. $|x| = 1$ となる初期ベクトル x を選ぶ
2. $y = Ax, \lambda = (x, y)$
3. $x = y / |y|$
4. λ が十分に収束するまで2.へ分岐

こうして絶対値最大の固有値 λ が求まる。(その最大固有値以外の固有値を求めるには上記のヤコビ法や2.の方法を用います)

上記の方法の欠点は、固有値が求まるまでの収束に時間がかかってしまう所です。

そこで、 $B = A - pI$ (p は定数)と置くことで固有値の式 $Ax = \lambda x$ を変形すると以下のような式になります。

$$Bx = (\lambda - p)x$$

($\lambda - p$: B の固有値、 x : B の固有ベクトルとなり、算出後はこの関係式より A の固有値を取り出せば良い)

これをシフト法といい、ヤコビ法の収束を早くすることができます。数学とは不思議ですね。

求めた固有値の使いどころ

こうして求められた固有値達は以下の数学的技術などに用いられます。

- 主成分分析
- スパースモデリング
- 最小二乗法
- 固有値分解
- etc...

(これらの解説サイトなどはほとんど固有値の計算方法については何も言ってくれません、計算方法を理解した上で読んでくださいといったイメージです)

感想

ほとんどの場合、ライブラリで全部良い感じに解が求められるようですが、数式に興奮する惹かれる変態人たちはライブラリの使い方よりも理論を理解することに時間を使ってしまうので、ライブラリの使い方を交えた上での学習を行っていきましょう(自分に対する戒め)

アセンブリ言語入門

アイコン

学部/回生 情報理工学部 SAコース 2回生

名前 TNB

どうも最近グルメスパイザーが欲しくなってきた田邊です。
僕みたいに、締め切り当日に書き始める腐れ野郎は、他に、いますかっていねーか、はは

ところでみなさんアセンブリ言語は割と身近なものに使われています。なんだと思いますか？それはピカピカ光ってキュインキュイン音になるアレです！！（演出、役物はC言語ですが）という訳で今回はそんなアセンブリ言語のhello worldをやっていきたいと思います。

Macで使えるアセンブラはGASとNASMがありますが、今回はNASMを使います。まずはNASMをインストールしましょう！

```
brew install nasm
```

次にコードを書いていきます。

```
hello.asm
1  global  _main
2
3  section .text
4  _main:  mov     rax, 0x02000004
5  mov     rdi, 1
6  mov     rsi, message
7  mov     rdx, 100
8  syscall
9  mov     rax, 0x02000001
10 xor     rdi, rdi
11 syscall
12 section .data
13 message: db     "これが俺たちの絶唱だああああああ", 10
```

ざっくりと説明すると、raxに呼び出したいsyscallの番号、rdiに標準出力するための1、rsiに出力するデータの先頭アドレス、rdxに出力するメッセージのバイト数をそれぞれ入れてます。そして、システムコールを行いこの場合だと標準出力が行われます。そして最後に終了するためにrax,rdiに0を入れてシステムコールを行なっているんですが、xorはmov rdi,0でもいいんですが、なんでもxorの方が効率がいいとか。

それでは最後に実行してみましよう！

実行するには

1,オブジェクトファイルを作る

2,機械語に直す

という作業が必要になります。

1は以下で行います。

```
nasm -f macho64 hello.asm
```

基本的にオブジェクトファイルを作るコマンドはnasm hello.asmが良いのですがmacの場合64bitのアセンブリしか実行できないようなので、
-f macho64 と付け足し64bitに対応していること伝える必要があります。

2は以下で行います。

```
ld -macosx_version_min 10.14.5 -lSystem -o hello hello.o
```

2のldはLinuxやmacのコマンドでオブジェクトファイルを機械語にしてくれます。この時、バージョンは自分のものに書き換えてください。

最後に./helloで実行して見ると、



```
Apple > ~/Documents/assembleble  
[ ./hello  
これが俺たちの絶唱だあああああああ
```

問題なく出力されているようです。

今までアセンブリ言語には全く興味がありませんでしたが、やってみると以外と面白さを感じました。ゆくゆくはピカピカ光るピエロみたいなものを作ればいいなと思ってます。

アセンブリ言語は以外と面白いので、もしほんの少しでも興味が湧いた人がいれば是非触ってみてください！！！！

2020年において速く、速いWebサイトを作る方法 ～Webフロントエンドタイムカプセル～

学部/回生

情理SA2回生

名前

やぎちゃん（八木田裕伍）



※この記事において、特筆しない限りフロントエンドとはWebフロントエンドのことを指します

こんにちはやぎちゃん (@ygkn35034) です！！！！

フロントエンドは変化が速い分野と言われています。本が執筆され、出版されるまでに書かれている内容が古くなることもあるほどです。そこで、そんなフロントエンドと数年先まで残るであろう会誌の性質を逆手に取り、現在のフロントエンド事情について書き、数年先に「この頃はこんな技術を使ってたんだな～」とノスタルジーに浸る、いわばタイムマシンのような記事を書くことにしました。

「現在のフロントエンド事情」と書きましたが、僕自身がフロントエンドについて知っていることに限りがあること、スペースが限られていること、~~締め切り間近に書き始めたこと~~から僕が普段使用しているTypeScript、React、Tailwind、そしてNext.jsについて述べます。

TypeScript

TypeScript は、Microsoft が開発した JavaScript に変換して実行する Alt JS と呼ばれる言語の一つで、コードの実行前にその動作の型が正しいかを検査する静的型付けの機能を JavaScript に追加することを目的とした言語です。事前に静的に型を検査することによりケアレスミスを防止し、テキストエディタによるコード補完や型表示によりコーディング時に考えるべきことが減るので、より高速にコーディングができるようになります。この記事を書き終わった後に Template Literal Types が実装された v4.1 がリリースされました。

React

React は、Facebook が開発した「ユーザーインターフェース構築のための JavaScript ライブラリ」（公式Webサイト原文ママ）です。宣言的に、JSX という名前の XML を JavaScript のコード中に書いたような独自の記法で書かれた、コンポーネントを組み合わせて UI を構築するという特徴があります。また、2019年2月にリリースされた Hooks API によってこれまでクラスを用いたコンポーネントでしか使えなかった、状態などの副作用を関数コンポーネントでも使用できるようになり、TypeScript との親和性もより高くなりました。しかし、公式では関数コンポーネントをあまり推している記述がなく、クラスコンポーネントで書かれたライブラリなどのコードも多く存在するのが現状です。Facebook だけでなく Twitter、Discord、Slack など現在 React を採用している Web アプリケーションは枚挙にいとまがありません。この記事を書く少し前に新機能を実装せず、段階的なアップグレードを簡単にすることを目標とした v17.0 がリリースされました。

Tailwind

Tailwind は utility-first というコンセプトのもとに作られた CSS フレームワークです。utility-first とは、`display: flex` を表す `flex`、`padding-top: 1rem` を表す `pt-4` といったように、1つのスタイルを当てるクラスを1つのHTML要素に複数付けることによってスタイリングするというものです。このコンセプトは一見スタイリングのアンチパターンと言われるHTML要素に `width` 属性や `style` 属性を付けることと同じようなものに見えますが、予め生成しておいたクラスしか使えないという制約があるため、無秩序にスタイルを当てることを防止できます。また、大量のクラス名を作ることによってCSSファイルの容量が大きくなりそうですが、未使用のクラスをCSSファイルから削除する機能があるためむしろCSSの容量は少なくなります。この記事を書いているときにv2.0がまるで映画の予告編のようなドラマティックなPVとともにリリースされました。

Next.js

Next.js は React のフレームワークです。JavaScript のモジュールに分割されたコードをまとめる webpack、先述した JSX や TypeScript で書かれたコードを変換する Babel を用いてブラウザ上で実行可能な形式にする、逆にコードを分割し、必要なファイルのみを読み込む Code Splitting などアプリケーションの開発者が実装することが難しい細かい最適化処理を行う、ブラウザからリクエストされるたびにサーバー上で React を実行し HTML を生成するサーバーサイドレンダリング (SSR)、予め静的な HTML を生成する静的サイト生成 (Static Site Generation / SSG) などの機能があります。v9代で様々な機能が追加され、幅広い使い方が可能になり「React プロジェクトを作るならとりあえずこれを使っておけ」と言えるようなフレームワークとなっています。また、公式のドキュメントや例が豊富で、先述した TypeScript や Tailwind の他にも様々なライブラリの使用例が公式リポジトリに掲載されています。この記事を書く少し前に画像を最適化する `next/image` が追加された v10.0 がリリースされました。

まとめ

これらの言語、ライブラリ、フレームワークを使用することでより速く、より速いWebサイトやWebアプリケーションを作ることが可能になります。僕は他のサークルの公式Webサイトや、授業の課題で作成したWebアプリケーションをこれらを使用して作りました。

また、現在はまだアルファ版の Blitz.js と呼ばれる Next.js を組み込み、さらにデータベースとの統合機能や認証機能も備えたよりフルスタックなフレームワークがあります。現在僕は Hello World! を終えたところなのですが魅力的なフレームワークだと感じました。

最後に、これからフロントエンドの技術がどのように進化していくのか期待しつつ、ここで筆を置かせていただきたいと思います。

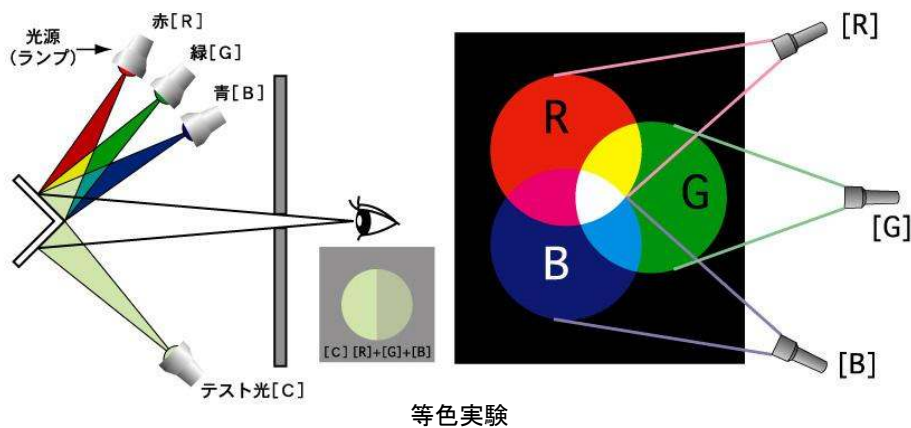
いろいろないろのはなし

学部/回生 情理学部 SAコース B3

名前



突然ですが、皆さんの身の回りはいろいろないろであふれているかと思えます。そんないろですが、実は三つのいろの合成で表現できるってご存じでしたか？まあこれはご存じかとは思いますが。ご存じでない方は中高の美術か理科あたりを再履修してきてください。分光とかの範囲かなと思います。まあここでも軽く説明するんですが、詳しいことはそっちによろです。実はいろは赤(Red)、緑(Green)、青(Blue)の組み合わせで表現できます。すごい。そしてこれらのいろの表し方で、国際照明委員会(CIE)で標準表色系として承認されたものが「XYZ表色系(Yxy表色系)」です。これは混色系の色体系であり、色の科学的測定の基礎となる表色系です。原色を設定してその混色量で色を表示する体系で、色の定量的表示に適しています。「XYZ表色系」は、[R][G][B](Red/Green/Blue)という「色光の三原色」の混色量を原理にしています。この原理は、人間はRGBの3種類の受容器によって色を知覚しているという「ヤング-ヘルムホルツの三原色説」の上に成り立っています。検索ワードがたくさん。CIEは、色の表示を標準化するために[R][G][B]という原色を、赤原色[R]=700nm、緑原色[G]=546.1nm、青原色[B]=435.8nmの単波長光と定め、これらの原色を用いた「等色実験」(三原色を調整しながら混色し、等色した=等しい色に見えた、その瞬間の三原色の各混色量を測る)を重ね、「等色関数」(人間の標準的な色覚を数値化した関数)を定義しました。



ああああああああああああああああああ

bash on Arch Linux on Windows

学部/回生 情報理工学部・SNコース 3回生

名前 3^5



お久しぶりです。今年は会誌担当ではありませんが弊会の副執行委員長になりました、3^5です。去年Arch Linuxの話を書いたので今年もArchでいきたいと考えていたところ、良さげな題材が見つかったので締め切り2日前にカラオケ店に籠って書いています。

人間なら誰だってWSLをUbuntuじゃなくてArch Linuxを使いたいと感じるときがありますよね…？別にUbuntuが嫌とかそういうわけではないのです。単純に身体がArchを覚えていてpacman(知らない方のために補足しとくとこれはaptのようなパッケージマネージャです)を求めてしまうのです。そこで今回はWSLでArch Linuxを用いるためのセットアップ方法を記述していきます。なお、WSLはバージョン2を使っていきます。

Archと言ってもWSLにインストールするのは簡単です。というのは、もう既にやっている人がいるからです(足を向けて寝られないですね)。手順は大きく分けて次のようになります。

- インストール用のexeファイルとtarファイルのセットをDLする
- exeファイルを実行し、インストールを行う
- 初期設定を行う

順番に説明していきます。

まずインストール用のexeファイルとtarのセットをダウンロードするところからです。wslidlというリポジトリがGitHub上に存在し、そこで初期状態のtarファイルを用いて任意のLinuxをWSLにインストールすることのできるexeが公開されています。これを用いてArch Linuxのtarファイルと合わせることでインストールが可能なのですが、当然のようにこれらのセットも公開されているため、我々はzipファイルを一つダウンロードするだけで済みます。



解凍すると↑のように2つファイルが出てきますが、同じディレクトリに入っていれば実行できます。“Users/<user_name>/wsl/arch”みたいな分かりやすいディレクトリを作ってそこに配置しておくのが良いでしょう。

Arch Linux以外にもwslidlのprebuild packageとして公開されているLinuxとしてAlpine LinuxやCent OS, FedoraやGentoo, Amazon Linux, Red hatがあります。Red hatのリリースファイルに含まれているrootfsは再配布可能なUniversal Base Imageらしいですが通常通りサブスクリプションマネージャを用いて登録し、RHELリポジトリにアクセスすることもできるようです。

次に、exeファイルを実行します。

実行するとwslidlが自動的にWSLのエントリにArchを追加してくれます。このとき、exeファイルの名前を変更しておくで追加されるディストリビューション名を変更できます。また、当然ですが同じディストリビューション名のエントリが存在しないかどうかは確認しておく必要があります。WSLのディストリビューションはpowershellで`wslconfig /l`を実行すると以下のように確認できます(表示がずれてますが)。

```
wslconfig /l
Linux 用 Windows サブシステム ディストリビューション:
Ubuntu-20.04 (既定)
docker-desktop-data
docker-desktop
```

最後に、WSLのArch上で初期設定を行います。

`/etc/pacman.conf`や`/etc/pacman.d/mirrorlist`に変更を施します。Archユーザなら多分分かるので好きに変更してください。続いてpacmanのキーリングのセットアップを行います。

`pacman-key --init`; `pacman-key --populate archlinux`のようなコマンドを実行し、pacmanのPGP鍵をセットアップします。

詳しくはArchWikiのpacman-keyのページを参照すると良いと思います。

あとは現在rootでログインされているので一般ユーザを追加したり好きなパッケージを入れてインストールは終了です。

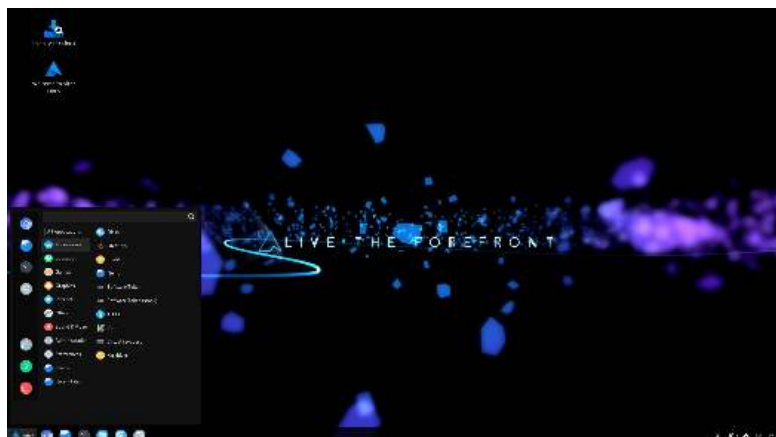
楽しいArchライフをお過ごしください！

個人的Arch関連ニュースのコーナー

最近Alter LinuxというArch派生のLinuxが気になっています。

ManjaroのようにGUIインストーラが存在し、インストール直後から当然のようにデスクトップ環境があり、さらには日本語入力機能まで整っているという便利具合です。それもそのはず、Alter Linuxは日本の学生コミュニティによって開発されているらしいのです。Webサイトやデフォルトの壁紙デザインも凝っていて非常にかっこいいので興味がある方は使ってみてはいかがでしょうか？

デスクトップ画面→
ロゴ↓



障子にMary作るよ

学部/回生 情報理工学部情報理工学科SA3回生

名前 チャムチャム



こんにちは、チャムチャムです。

突然ですが本会には壁に耳あり障子にMaryということわざから生まれたMaryという部室を見守る女の子がいます。

ただ、今はあまり部室は使われておらず、せっかくのMaryちゃんも暗闇を見守るだけ、という状況になっているので、せっかくなら今のうちに立体にしてみよう！という軽い気持ちでMaryちゃんを3Dにしてみました。

せっかくなので、どういう風に3Dで女の子を作るのかについて簡潔な流れと、ちょこっと補足を説明していこうかなと思います。全然知らない方でも雰囲気を感じていただけたら幸いです。

ちなみにBlenderは初心者なので何かありましたら教えてください。

使用ソフトはBlender2.8とクリスタになります。

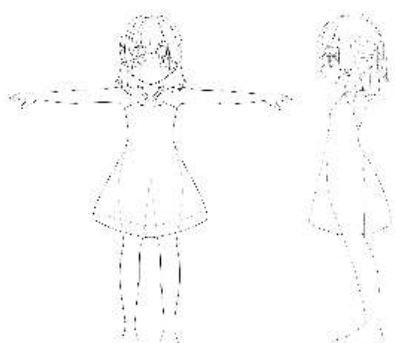
1. 下絵を用意します

クリスタのレイヤーカラーの部分から下絵の線は最終的に白色にすると、Blenderの背景は基本黒色なので見やすかったです。

また、左右対称定規という左右が対称になる定規も使用したのですが、レイヤーを分けた際に別個で作成したため、顔と体の左右対称になる部分がずれてしまうという痛恨のミスをしてしまいました。体を作っていく上でちょっとずらしてなんとかしましたが、左右対称定規はコピーして場所がずれないようにすると後々絶対楽です。

2. 形を作っていきます

頭→体→顔のパーツ→手足→髪の毛→服→小物という順番で作成していきました。結構その時やりたいパーツから作っていても、なんとか繋ぎ合わせる方法は結構あったので私は手足より好きな目とかから作っていくのがモチベが保てて楽しかったです。



3.UV展開

そもそもUV展開とは何ぞや…？という人のために一応説明しておく、色を塗りやすいように3Dのパーツのどの部分に2Dの画像の色を割り当てるか決める、みたいな処理のことを指します。算数とかの授業で立方体を紙で作ったりしたかもしれませんが、それをチョキチョキ切りなおして展開図にしていみたいな作業です。

立方体の展開図は11種類あるらしいですが、これがもっと複雑な形になると…とんでもないパターンチョキチョキ切る組み合わせがあります。

その中から一番塗りやすい場所でチョキチョキ切るのがめっちゃくちゃ大変な作業でした。

4.Blenderかクリスタで色を塗る

UV展開したUVを、その場所に応じて塗っていきます。

私は色を塗るのがとても苦手なのですが、とりあえず目を塗って他ベタ塗りでもひとまずなんかいい感じになりました！

ただ、ちょっと物足りないので色塗る勉強をしてもっと改善出来たらいいな、というのが今後の展望です。

おまけ: ハーフアップの作り方

頭の横の線をグッとハーフアップにしたい部分まで動かし、Ctrl+Rでループカットをして、いい感じの所でこう、ギュッとAlt+Sで膨らませるとなんか私的にいい感じのハーフアップになりました！

以下の画像みたいな感じですね。

真ん中にはリボンをつけているのでそこは少し手を抜いています。



以上です！もしよかったら皆さんも作ってみてください。

とても分かりやすくほとんどこれのおかげで作れましたという参考動画

https://www.youtube.com/playlist?list=PL0cOz9sfc328N_rV0sn5A1j_gD9VsyWoD

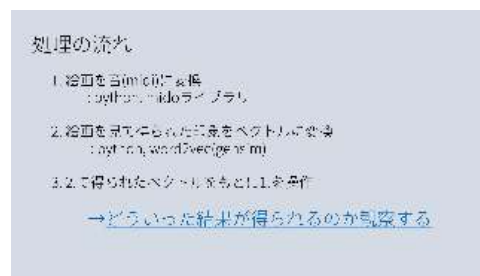
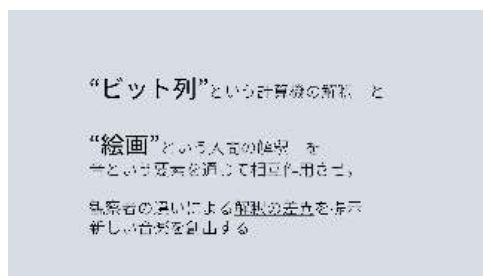
“情報”について

学部/回生	情報理工学部実世界情報コース3回生
名前	Tsubokura(@ukstbkrs)

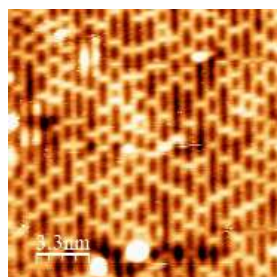


こんにちは。コロナで大学とか国からまとまったお金をもらえたので、どうせなら引っ越すか〜〜〜ってなって南草津とかいうディストピアから最近浜大津へ引越しを敢行したTsubokuraです。琵琶湖近い！駅近い！商店街ある！電車が道路を走ってる！すごい！たのしい！！京阪京津線は同じ車両で道路走って山越えて地下に潜るっていうかなり意味不明なアトラクションなので一回乗ってみて欲しいです。おすすめ。アイコンは深夜2時の琵琶湖ほとりで飲んだ角瓶です。うまかった。

さて、今回は“情報”についてのお話をだらだら書いていきたいと思えます。最近「絵とビット列の違いは何かを表現する作品」や、「これまで扱われてこなかったTiO²の原子レベルでの表面画像を画像解析する」など“情報”それ自体について考える機会が多かったからです



1.絵とビット列の違いについて考える



2.TiO²の表面画像。今までコンピュータで解析されてなかったらしい。

で、“情報”とはそもそも何なのか〜といいますが、「他との差異が認められるもの」だと思います。抽象的ですね。「他との差異がないもの」はそれ自体として“情報”を持つとは言えないので、その逆と考えれば分かりやすいと思います。

そして、「差異」を観察するためには意識的にせよ無意識的にせよなんらかの基準—「構造」を基にして行っています。この「構造を基にして差異を認識し、“情報”を扱う」というのが、学問にせよ運動にせよ人間の全ての行動です。

コンピュータサイエンス(長いので以下CSとします)の話に移ります。実は、CSは「構造を基にして差異を認識し、“情報”を扱う」ことをかなり直接的に、明示的に扱っているすごい学問なんです。当然ながら、コンピュータと人間の「差異を認める構造」は違うのですが、それをなんとかして理解し双方に利益をもたらそうという試みがCSです。その過程で“情報”もコンピュータに扱えるように数学的に理論付けされます。そして、コンピュータは人間より(単純な、という接頭語がつきますが)処理能力が遥かに優れているので人間が持つ「構造」では捉えきれなかった“情報”を捉えることができます。例えば、僕が今やってるTiO²の表面画像はこれまで人間の手でしか処理されてなかったのものでその傾向があまりわかってなかったのですが、画像解析をすることで新しい事実が発見できました。これがCSという学問が持つ価値です。なんとなくお金がたくさん貰えそうな気がしてきますよね。

このように普段勉強しているCSを改めて位置付けることにより、例えば今自分のやってることがどんな意味を持っているかとか、何か新しく見えてくるものがあるのではないかなあとと思います。この拙文がそのきっかけとなれば幸いです。

参考文献

増補 情報の歴史, 編集工学研究所, NTT出版

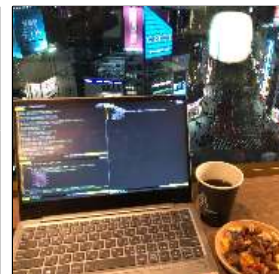
世界情報文化史講義 情報の歴史を読む, 松岡正剛, NTT出版

情報理論, 甘利俊一, ちくま学芸文庫

ZIPファイルを読む(狂気)

学部/回生 情報理工学部 SAコース 3回生

名前 uMa



uMaです。(現在会誌締め切り日当日朝6:30)

最近は参考書を読むばかりで、ネタになるような開発が出来ていなかったのも直前までネタに困っていたんですが、OSやらコンパイラやらgitやらを自作している人がちらほらと居たのを思い出したので、ZIPファイル展開できるプログラム作るか…と思いました。

そうと決まると、ZIP形式の仕様を理解する必要がある訳ですが、PKWareというところがテキストファイルとして仕様を公開してくれているので、そちらを読むことにしました。

結局、プログラムの実装まで漕ぎ着ける事は出来ませんでした。ZIPに関して色々勉強になる事が多かったのも、それをまとめていきたいと思えます。

ZIPファイルは中身が無くてもいい？

仕様書には、ZIPファイルは1つ以上のファイルを含む必要がないと書かれています。

つまり、中身が空であるZIPファイルが作成できるという事です。

必ず圧縮される訳ではない

ZIPに限った話ではありませんが、充分サイズが小さいファイルを圧縮しようとする、メタデータによって逆にサイズが膨れ上がってしまう場合があります。

また、ZIPでは圧縮されているかどうかのフラグが用意されており、実際小さいテキストファイルを圧縮して中身を見てみると、そのままの内容で保持されている事が確認できました。

```
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F 0123456789ABCDEF
0 50 4B 03 04 0A 00 00 00-00 00 34 27 73 51 81 AD PK.....4'sQ..
10 A8 5A 1B 00 00 00 1B 00-00 00 08 00 00 00 74 65 .Z.....te
20 73 74 2E 74 78 74 74 68-69 73 20 69 73 20 61 20 st.txtthis is a
30 74 65 73 74 20 66 69 6C-65 20 66 6F 72 20 7A 69 test file for zi
40 70 50 4B 01 02 3F 00 0A-00 00 00 00 00 34 27 73 pPK..?...4's
50 51 81 AD A8 5A 1B 00 00-00 1B 00 00 00 08 00 24 Q...Z.....$
60 00 00 00 00 00 00 00 20-00 00 00 00 00 00 74 .....t
70 65 73 74 2E 74 78 74 0A-00 20 00 00 00 00 01 est.txt.. .....
80 00 18 00 BE AA E9 10 E5-BD D6 01 BE AA E9 10 E5 .....
90 BD D6 01 AE 15 C5 05 E5-BD D6 01 50 4B 05 06 00 .....PK...
A0 00 00 00 01 00 01 00 5A-00 00 00 41 00 00 00 00 .....Z...A....
B0 00 ■ .■
```

図で、test.txtと書かれている部分が2箇所と、this is a test file for zipという文字列が確認できると思います。

また、以下にtest.txtとtest2.txtを圧縮したZIPファイルをバイナリエディタで開いた際の画像を示します。

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	0123456789ABCDEF
0	50	4B	03	04	0A	00	00	00-00	00	34	27	73	51	81	AD	PK.....4'sQ..	
10	A8	5A	1B	00	00	00	1B	00-00	00	08	00	00	00	74	65	.Z.....te	
20	73	74	2E	74	78	74	74	68-69	73	20	69	73	20	61	20	st.txtthis is a	
30	74	65	73	74	20	66	69	6C-65	20	66	6F	72	20	7A	69	test file for zi	
40	70	50	4B	03	04	0A	00	00-00	00	00	11	3B	73	51	2D	pPK.....;sQ-	
50	16	30	A2	22	00	00	00	22-00	00	00	09	00	00	00	74	.0."...".....t	
60	65	73	74	32	2E	74	78	74-74	68	69	73	20	69	73	20	est2.txtthis is	
70	61	20	73	65	63	6F	6E	64-20	74	65	73	74	20	66	69	a second test fi	
80	6C	65	20	66	6F	72	20	7A-69	70	50	4B	01	02	3F	00	le for zipPK..?.	
90	0A	00	00	00	00	00	34	27-73	51	81	AD	A8	5A	1B	004'sQ...Z..	
A0	00	00	1B	00	00	00	08	00-24	00	00	00	00	00	00	00\$......	
B0	20	00	00	00	00	00	00	00-74	65	73	74	2E	74	78	74test.txt	
C0	0A	00	20	00	00	00	00	00-01	00	18	00	BE	AA	E9	10	
D0	E5	BD	D6	01	25	1A	4B	AF-EE	BD	D6	01	25	1A	4B	AF%.K.....%.K.	
E0	EE	BD	D6	01	50	4B	01	02-3F	00	0A	00	00	00	00	00PK..?.....	
F0	11	3B	73	51	2D	16	30	A2-22	00	00	00	22	00	00	00	.;sQ-.0."..."...	
100	09	00	24	00	00	00	00	00-00	00	20	00	00	00	41	00	..\$......A.	
110	00	00	74	65	73	74	32	2E-74	78	74	0A	00	20	00	00	..test2.txt... ..	
120	00	00	00	01	00	18	00	7B-5A	68	97	F9	BD	D6	01	7B{Zh.....{	
130	5A	68	97	F9	BD	D6	01	A2-10	85	80	F9	BD	D6	01	50	Zh.....P	
140	4B	05	06	00	00	00	00	02-00	02	00	B5	00	00	00	8A	K.....	
150	00	00	00	00	00	00	00									

よく見るとtest2.txtという名前とその内容(this is a second test file for zip)が確認できます。

本来であれば、フォルダを含めた場合や、更に大きいファイルで圧縮される場合もお見せしたいのですが、余白も少なくなってきたので、気になる方はご自分の目でご確認ください。

構成について

最後に、ZIPファイルの構成について軽くご紹介します。

ZIPファイルでは、PKから始まる4byteのシグネチャによって、構造の判別を行っており、上の図でもいくつか”PK”という文字が確認できると思います。

ファイルの情報は、サイズや圧縮方式、名前等からなるヘッダと、本体部分に分けられており、圧縮方法は16種類あります。

全て実装するのは難しそうですが、Deflateだけでも実装できれば大丈夫そうですので、時間があれば実装してみたいですね。

内容は以上となりますが、最後に1つ。

ZIPの産みの親のフィル・カツツさんはアルコール中毒で亡くなられています。

皆さんもお酒の飲みすぎには注意しましょう！

良い子と悪い子とどちらでもない子のための

ダークウェブ超入門

学部/回生

情理3回 SNコース

名前

Yukimura(@yuki12017825|木村 悠生)



ダークウェブという中二病的なワードに引っかかって本記事を読み始めてしまった方、こんにちは、Yukimuraです。ダークウェブってなんかよくわかんないけど響きがかっこいいデスネ、ちょっと使ってみたいデスネ。ちなみに筆者は「ハッカー」とか「サイバー」とか「闇サイト」みたいな、そういう響きのカッコよさにつられてセキュリティ系のコースに進み、とうとう三回生になってしまいました。

閑話休題、本記事は、「ダークウェブを使ったハッキングか、“奴ら”もなかなかやるな、フハハハ」みたいな拗らせたぼくちゃん方から、内部告発をしたい方まで、ダークウェブってなんだか聞いたことはあるけど使ったことはないな、という貴方(貴女)に捧げる「ダークウェブ超入門」です。早速、ダークウェブへのアクセス環境構築から、やってみよう！

※なにができるようになるか、っていうと、「誰がアクセスしてるかわからなくできる」です。サイトへのアクセス履歴であったり、匿名掲示板とかへの書き込みとかが、本当に匿名(特定不能)になります。

決して悪用はしないように

1. ダウンロード

<https://www.torproject.org/download/> から自分の環境に合わせてTor Browserをダウンロードする

2. インストール

展開とかしてインストールする

3. 使う

普通にブラウザとして使う



4. 以上

It' a true wolrd. welcome to underground...

以上4ステップで、通信元を秘匿しての任意のWebサイトへのアクセスが可能になりました。さらに「onionドメイン」のサイトにもアクセスできるようになります。いわゆる普通のブラウザからだとアクセスできない、「闇サイト」ってやつですね。~~うーんとっても中二病心を擦る響き~~

...とまあこれだけではつまらない(というか技術的な要素がない。Torブラウザのチュートリアルでも読んどけ案件)ので、秘匿ネットワーク上にサーバを公開する方法も書きます。中二病っぽく言うと「貴様もこれで闇サイトの管理人だ」です。やったね。

そんなわけでやっていきましょう。今回もたったの4段階です。

1. パソコンと環境を用意
OSは一身上の都合でUbuntuを使って説明します。Webサーバシステムは本稿ではnginx。違う環境の人は適宜読み替えてください。
2. 環境設定とコンテンツ作成
以下を「端末」で一行ずつ入力して実行

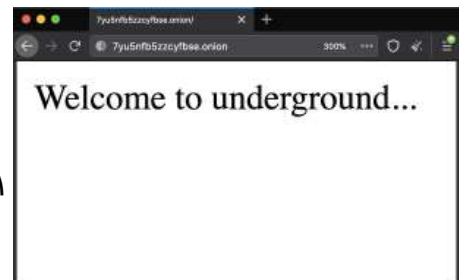
```
sudo apt install tor
sudo systemctl start tor
sudo apt install nginx
sudo more /var/lib/tor/hidden_service/hostname
```

ここまでできたら「XXXXXXXX.onion」というのが画面に出くるはずですので、これをメモ。続いて

```
sudo mkdir /var/www/onion/
echo "server{
server_name XXXXXXXX.onion;
root /var/www/onon;
}" | sudo tee /etc/nginx/conf.d/onion.conf
cd /var/www/onion/
echo "welcome to underground..." | sudo tee index.html
sudo chown -R www-data:www-data /var/www/onion/
sudo systemctl restart nginx
```

XXXXXXXX.onionは、さっきメモした文字列を入力してください。

3. アクセス
前ページでインストールしたTorブラウザでXXXXXXXXXXXXX.onionにアクセス！！(普通のブラウザからではアクセスできないことを、普段使っているブラウザで確認するとなお良い)



4. 以上！
やべーハッカーみたいな顔をして「実は俺さ...闇サイトの管理人なんだよね...」と友達に自慢しましょう！ 慣れてきたら、掲示板なんかを置くと、「+闇掲示板の管理人+」になれます。やったね。

さて、いかがだったでしょうか。かなーり基礎的なところを噛み砕いて書いたので、少しでも知識がある人にはだいぶ物足りなかったかなと思います。ただ、超初心者の方が感じているような「難しそう、怖そう、闇の技術」みたいなものではない、ということが、少しでも感じてもらえていたらな、とも思います。

犯罪でばかり目立つ匿名化技術ですが、正しく使えば不当な検閲にも対抗できたり、利用者のプライバシー保護の助けになるような技術です。是非正しく使って、より良い人生を送ってください。ではでは！！

～何もしていないのに壊れました～

起動しなくなったUbuntuを直した話

学部/回生	情報理工学部 無能情報コース 3回生
名前	るりと



こんにちは、るりとです。webやら機械学習やら色々やってきた結果特徴も無い内定も無い猿エンジニアになってしまいました。Twitterやってます：
@ruriro0125

動かないPC

電源を付けるとUbuntuが起動する筈がGrub shellが起動するようになってました。何故なのか...

ubuntu本体は生きていたので、/boot/grub/grub.cfgから起動。すると画面一杯に赤色のnvidia persistence daemon failedという文字が...どうやらnvidia-driverが壊れていると判断。

Nvidiaのドライバを再導入する

やることは簡単でnvidia周りを全てpurgeして再導入するだけ。再導入すれば上手く起動出来ると思ってました。そして走らないapt-get update...

Wifiが動作しない

wifiモジュールが刺さっているpcなのにwifiに接続が出来ない状態に。wifiモジュール自体は認識していたので😞。原因はrfkillによってwifiがブロックされてました。ブロックを解除するとwifiに接続出来るようになりました。そしてnvidiaのドライバも導入出来ました。やったぜ。

やはりGrub shellが起動する

ドライバを直せば正常にubuntuが起動するかと思ってましたがそんなことありませんでした。調べたところEFIパーティション側に存在するgrub.cfgが破損していることが分かったのでEFIパーティションをマウントしてubuntu側の/boot/grub/grub.cfgを移植するという荒業をやったところ直りました。やったぜ。

最後に

いかがでしたか？脳死してubuntuを再インストールした方が断然速かった。もう二度とやりたくないです。

MCEditを使ってマイクラでお家を建てよう！ 超入門

学部/回生 情報理工学部知能情報コース 4回生

名前 伊藤聡子(いとさと) Twitter : @it0sat0



マイクラでお家を建てたい！！

そんな野望をみなさん一度は持ったことがあるのではないのでしょうか？ 私はあります。**マイクラ**と**Minecraft**と言えば、世界トップレベルの知名度を誇るゲーム。一軒建てるくらいであればコツコツ手を動かせばいいものですが、人間とは欲深いもの。「もっと家を建てたい！」「いっそ、街を作りたい！」そうなるものです。なりませんかな？ なりますよね。そんな愚かな人間の願いをササッと叶えてくれるものが、**MCEdit**なのです！ Python2で書いたプログラムをfilterとし、実行することでワールドに変化を加えるものです。それでは、早速張り切っていきましょう！

1.環境構築

使うものはこちらの4つだけ！

- ・ **WindowsPC**(性能は高いだけ高ければいい)……MacだとOSのせいかバグる大変
- ・ **Anaconda2**……Python2の仮想環境立てるのに使う
- ・ **Python2**……懐かしきプログラミング言語
- ・ **MCEdit**……プログラムをマイクラの世界に適応するのに使用

環境構築に関しても細かく説明したかったのですが、ページ数の都合上残念ながら割愛します。無念！ 以下の公式wikiの指示にしたがって進めてください！

<https://github.com/mcgreentn/GDMC/wiki/The-GDMC-Framework> (全文英語)

途中、モジュールと呼ばれる追加コンテンツのようなものを入れるように指示されますが、いろいろバージョンの相性が合わない問題が出てくる……ということ。エラー出やすいモジュールやそれらの入れたらいいバージョン、入れ方はなどなどは以下の表とコマンドを参照あれ。

コマンド名 install モジュール名==バージョン

もし、MCEditの起動しても、マウス・キーボード操作ができない場合であれば、

pip install pygame==1.9.6

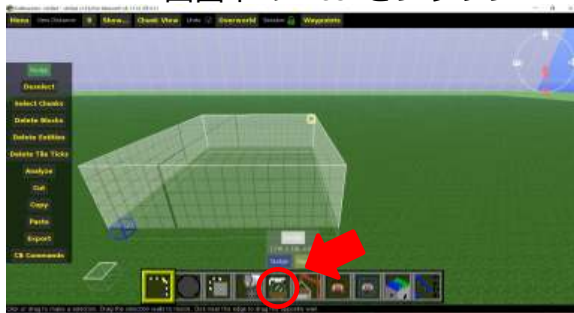
と入力して、pygameをバージョン指定して入れ直すと大体動きます！ いや～簡単ですね！

コマンド名	バージョン	モジュール名
conda	2.2.3	matplotlib
pip	1.9.6	pygame
pip	3.4	ftputil

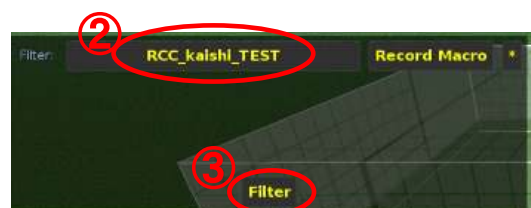
2.MCEditの基本操作

とりあえずMCEditを起動して、**Create New World**から**World Type**を**Superflat**にして、無限に平坦なワールドを生成しましょう。移動操作は、マイクラと同じです！

①filterの実行範囲を指定し、画面下のfilterをクリック



②filterを指定して、③filter実行 (ここでは、RCC_kaishi_TESTを選択しています)



この3ステップでfilter実行は完了です！ 試運転などでよくやる動作なので、覚えておきましょう！ では、肝心のfilterの開発へ→

3.filter開発

それでは本番。Python2によるfilter開発の始まりです。MCEditで使用するfilterは、環境構築の際にダウンロードした**GDMC-master**というフォルダの中にある**stock-filters**というフォルダで管理しています。

まずは、このフォルダ内にお好みの名前で**pythonファイル**を作りましょう。それを好みのエディタで開いてください。そしてそして、以下のサンプルコードにご注目！

```
1  #!/usr/bin/python
2  # -*- coding: UTF-8 -*-
3  import utilityFunctions as utilityFunctions #必要
4  from pymclevel import alphaMaterials, MCSchematic, MCLevel, BoundingBox #必要
5  from mcplatform import * #必要
6
7  displayName = "RCC_kaishi_TEST" #MCEdit上のfilter一覧に表示される名前を設定
8
9  def perform(level, box, options): #この中に生成プログラムを書いていく
10
11     (width, height, depth) = utilityFunctions.getBoxSize(box) #選択した範囲の横幅、縦幅、奥行幅を呼び出し
12
13     #選択した範囲に水をギッシリ生成する
14     for i in range(width):
15         for j in range(height):
16             for k in range(depth):
17                 utilityFunctions.setBlock(level, (79,0), box.minx + i, box.miny + j, box.minz + k)
```

17行目が、ブロックを生成するためのコードになります。基本的に、

utilityFunctions.setBlock(level, (ID,DataValue), x座標, y座標, z座標)

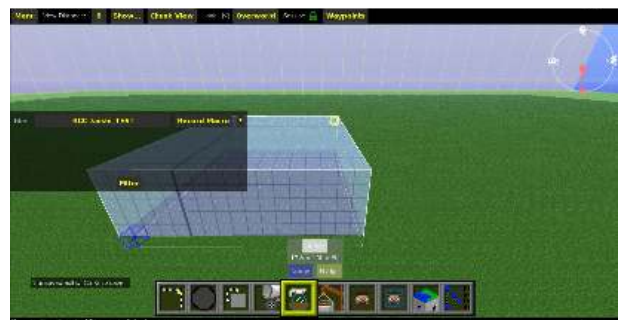
この書き方をします。上画像の17行目では、(79,0)としているIDとDataValueとは、マイクラ内で生成できるブロック・アイテムそれぞれに付けられた管理番号を指しています。そのほかのブロック・アイテムの番号に関しては、以下のURLをご参照あれ！

https://www.digminecraft.com/lists/item_id_list_pc_1_12.php (全文英語)

このfilterを実行した結果が右の画像です。プログラムのbox.minxやbox.minyというのは、指定した範囲のxやyそれぞれの最小座標を指しています。maxも存在していますので、その時々に応じて使ってみてください～。

MCEditはかなり重いソフトなので、「あれ、生成されていない」と、思った時はぜひ Minecraft1.12.2で加工済みのワールドを開いてみてください。大体生成されています。

さて、ここまでわかってしまえば、あとは想像力を働かせるだけ。forとifを乱用して、気合と根性と時間があれば、家を作ること街を作ること可能！ ちなみに、GDMCという名前で集落生成filterの世界大会があります。(http://gendesignmc.engineering.nyu.edu/) 毎年夏に開催されているので、極めまくってチャレンジするのもありよりのありじゃないでしょうか？



5.最後に

超超超入門部分だけの記事でしたが、いかがだったでしょうか？ この記事を書こうと思ったのは、実はGDMCに参加してたからなんですね！ 研究室で同期とタッグを組み、ボスからの最高のサポートのヒィヒィ開発していました。素晴らしい夏の思い出です。そんな我々、チームICE_JITが開発したものが右の画像のもの！(研究室から掲載許可済) 総合3位、美学分野1位のfilterです。和風でいい感じじゃないですか？

ハア～RCCで過ごした4年間、楽しかったなあ。もう卒業です。悲しい。会誌を書くことはないでしょう。というか、そうしないと留年になっちゃう。グッバイ会誌！ さらばRCC！ お前たちのことは忘れないぜ！



出先で VRChat がやりたい！



学部/回生	VRCC(ブイアールシー) 老害局長 (一応)4回生 情理 SN
名前	VR kanon-tencho @kanon_k4 k0ui1.jp

てんちょです。いきなりですが世界中の人々、誰もが1度は思ったことがあるだろう、

美少女になりたい！ と。

これは、必死に（22歳のおっさんが）環境構築して夢を現実にする夢物語である。

はじめに: VRChatの世界に迷い込んだ

2020年度、RCCでは新型コロナウイルス感染症拡大の影響をもちに受け、3月下旬から一切部室（大学内の活動拠点）が使えなくなってしまいました。

RCCのメンバーにも会えない時に、とあるお兄さんから教えてもらったのが、「VRChat」。最初はデスクトップモードで遊んでいましたが、5月に入ると、どうしてもVRヘッドセットが欲しくなり、Oculus Questを買ってしまいました。

※VRChatに関しては、他の人が寄稿してくれることを祈って省略します。（記事なかったら@taken_0114って方のTwitterを開いたら履修できます）



5月に買った初代Questと10月に新しく買ったQuest2

準備物

- ・一人に一台ある、Oculus Quest / Quest 2
- ・一家に二台はある、高性能ゲーミングPC
- ・大学講義には必須！ iPad Pro
- ・ポート開放ができる、NATルータ
- ・1Gbps契約の超高速(安定)回線
- (・出先もそれなりの高速回線)



自宅サーバ勢にはおすすめYamahaのRTXシリーズ

それぞれの環境構築



PCに導入したアプリアイコン群

まず、通常のVRゲームを動作させる環境を作りましょう。ここではStramVR上でVRChatが動作できる環境が構築されています。

次からが難題ですね、Oculus QuestにはOculus Linkと呼ばれる、PCとQuestをUSBケーブルで繋いでPC向けVRゲームができる公式機能がありますが、それを無線化していきます。

今回は、「Virtual Desktop」というアプリを使って無線化していきます。

※Questのアプリストアで1,990円で販売されています。

PCとQuest間の具体的な設定方法に関しては、割愛しますが、PCソフトであるSideQuestを使用して、QuestのVirtual Desktop上でSteamVRが遊べる状態にしていきます。

最後に、ローカル環境で動作が確認出来たら、ルータのポート開放を設定します。
(TCPの"38810","38820","38830","38840"をPCに向けて設定)

お好みで外からPCの画面を確認できるTeamViewerをインストールして構築完了です。



実際にVirtual Desktopを使って会誌を書く図

実際に外出先から遊んでみる

環境が構築出来たら、超高速回線が引かれているお友達の家に行って、実際にリモートでVRChatを起動してみよう!!!

準備物にiPadがあるのは、この時TeamViewerを使って自宅のPC画面が確認できるから、便利ですね！
これでOculus Quest1つで、まるで自宅にいるかのように、他人の家でVRChatが楽しめます！

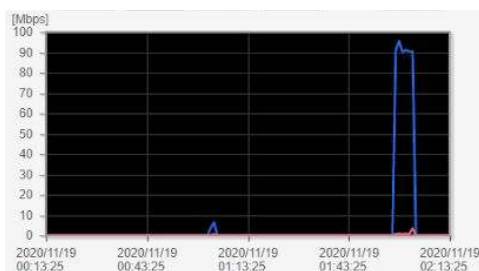
(そもそもVRだから自宅関係ないなんてマサカリx)



iPadのTeamViewerで接続先のPCを確認しながらリモートVRChat

どの程度の回線速度が必要なのか

無線VRって思ったより帯域が必要になります。静止コンテンツとかならいいですが、VRChatはワールドを動き回ったり、他人と会話する必要があるため、余裕をもったネットワーク環境を用意することをおすすめします。色々検証しましたが、自宅のPCと外出先のQuestの通信が40Mbps ~ 100Mbps程度出ないとともにVR出来ません。マンションだと厳しいかも。



最高品質設定では100Mbps程度の帯域が必要となった

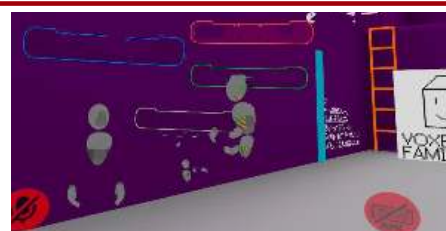


検証環境では上下共に800Mbpsを超える回線を使用した

おまけ: なぜPCのVRにこだわるのか

Questってスタンドアロンで動くから便利なんじゃねーの？って思った、その君。正論で何一つ間違ってます。ただ、Quest版のVRChatには2点ほど難点があり、「対応アバター以外はロボットで表示される」「カメラ機能が使えない」と、まあ不便で使いにくい。なので、意地でもPC版のVRChatを使いたいと思ってるのですね、はい。

※VRChatにはRCCのバチル部室がOBの手によって爆誕し、「VRCC」として現役会員やOB達が活動しています。見学はいつでもどうぞ。



こんな感じでロボットアバターがうろちょろしている

VR空間でUXデザインをした話

学部/回生 理工学部機械工学科 4回生

名前 taken(@taken_0114)



去年の12月末にOculus Questを買ってからというもの、VRChatにハマりまして、今年の4月からそこで毎週#VRChat関西弁集会というイベントを主催しています。(RCC民も何人か誘ってVRCCが発足してたりする。)

会場となるワールドをUnityで自作しているのですが、今回そのワールド制作でUX向上を目指してこだわった点を紹介したいと思います。

■入り口■

リスポーン地点には会場の案内板と無言ユーザー向けの花瓶型筆記具を設置しています。アルコール消毒液はVR空間では本来必要のないものですが、初参加ユーザーに対して受付でアルコール消毒を促すことで、会話のきっかけを生み出す狙いがあります。



■トイレ■

トイレは本来VR空間に必要なものですが、カレンダー、アバターのお色直し用の鏡、防音壁を設置することで、現実世界と同じ用途でトイレを利用できるという狙いがあります。



■食堂車■

VRChatではワールドやイベントに関係なく立ち話が多くなってしまいがちですが、適切な半径と高さの円卓を置くことで、円卓を囲んだ会話に誘導する狙いがあります。周囲にあるオセロ盤やマガジンラックは、参加者同士の会話を生み出す狙いがあります。



■ステージ■

ゲストの登壇やイベント用にステージとセンターマイクを設置しています、ステージは雑壇を兼ね備えた段差があるため、集合写真の撮影場所としても利用できます。



■ポスター■

VRChat内にはイベントやワールドのポスターが多々あるのですが、そういったポスターの中で一番目立つポスターを作ろうと考え、思いついたデザインです。

まず「関西」の二文字でギャラリーをポスターに注目させて、470文字あるイベント経緯の説明文でギャラリーをポスターの近くに引き寄せ、濃い文字の部分でイベントの概要を説明する狙いがあります。

紙面ではなかなか伝わりづらいですが、実際VR空間の中ではかなり目立ちます。

2019年の12月にOculus Questを買って、当初は全く別の目的でQuestを買ったはずが何気なくインストールしたVRChatにのめり込んでしまって、そのままの流れでわいわい楽しんたたら、ある日VRChatって関西弁を使う機会が少ないってことに気づいて、なんでだろうって理由を考えた結果VRChatは初対面の人と敬語で話す機会が多いけど敬語と関西弁って結構相性が悪いから、思っきり関西弁を話せるっていう状態になるのは難しいんだなって、でも何と云ってなんかないかも「ザ・なにわ」みたいな毎週開催イベントがもしあったとしてもそれもなんか違うなって思ってたある日、居酒屋の世界で呑んでたらカウンターに関西弁を使ってる人がいたから思い切って関西弁で話してみたら思いのほか楽しくて、「もっと関西人がVRChatでありのまま関西弁を話せるようになればいいのに。」って話になって、そこで2020年の4月に関西弁を話すフレンドを集めて#VRChat関西弁集会を開催してみたら結構いるんな方が参加してくださって、関西弁を勉強してみたって外国の人にも来ていたしたので、もう関西人だけじゃなくて非関西圏の人・無言の人・外国の人も参加可能にしようって方向性を変えて、そうして運営していると参加者の方に「昔は関西に住んでたけど就職してから関西弁を話す機会がなくて、今回関西弁を久しぶりにたくさん話せて嬉しかった。」とか言ってもらえるようになって嬉しくて、おかげさまで20回も開催させて頂いて、21回目からは少し方向性を変えて集会を隠れ家から飛行船に大規模改装しました。イベントに興味ある方はぜひ木曜22時にtaken_0114にjoinしてください。

ワールドとポスターにこだわったかいあってか、最近はイベントが盛況でありがたいお気持ちです。

VRChat、飽きさえ来なければ楽しいコンテンツなので皆さんも是非。



