

会誌2021



立命館
大学
会誌
2021

まえがき

冊子を手に取っていただきありがとうございます、
編集者のlufeと申します。RCCとして二年ぶりの出版です。

立命館コンピュータクラブ（以下、RCC）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ほとんど部室が使えず、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、この会誌に関わった方々に感謝申し上げます。

2021年12月

会誌担当者 lufe

目次

まえがき

1章 イントロダクション	5
1.1 立命館コンピュータクラブとは	5
1.2 活動内容	5
1.3 イベントの開催	5
1.4 普段の活動	6
1.5 活動拠点	6
1.6 入会について	6
1.7 アクセス	6
1.8 年間行事予定	7
2章 前期プロジェクト活動報告	8
2.1 DTM班.....	9
2.2 LT・プレゼン班	11
2.3 OS班	15
2.4 Web班	18
2.5 スロット班	22
3章 会員コラム	25
3.1 1回生	26
3.2 2回生	38
3.3 3回生	62
3.4 4回生	86
奥付	88

1章

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

今年は8月頃に、ハッカソンと言われるチームで特定のテーマに対して期間内でアプリケーションやサービスを開発して、その成果を競い合うといったイベントを行いました。今年からはオンライン開催を行って技術や交流をしました。テーマは「極限」であり、意見やアイデアを出すのが大変ではありましたか、面白い作品が多くでした。

(rcc_hack: http://www.rcc.ritsumei.ac.jp/2021/0902_12077/)

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3000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_rcc
Website:www.rcc.ritsumei.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・制作
5月	前期活動開始	前期プロジェクト活動のスタート
8月	ハッカソン	2泊3日の開発イベント
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
2月	ハッカソン	2泊3日の開発イベント
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

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 DTM班

音楽に関わる知識や技術を学び、オリジナル曲の製作を目標に活動を行いました。

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。

2.3 OS班

OS自作およびその拡張を行って、OSの処理内容に関する理解を深めることと、個人のOSを開発することを目標に活動を行いました。

2.4 Web班

Webの仕組みや技術についての知見と実装力を身につけ、個人制作物を開発することを目標に活動を行いました。

2.5 スロット班

オリジナルのスロットを作成する過程を通じてチーム開発でのプロセス、効果的なUIの設計などのテクニックを学ぶことを目標に活動を行いました。

DTM班

平井 栄太
情報理工学部 4回生

堀田 隆成
情報理工学部 3回生

駒谷 亮叡
理工学部 2回生

山口 蒼太
情報理工学部 1回生

齋藤 龍也
情報理工学部 3回生

新藤 尚輝
情報理工学部 3回生

田尻 聖奈
情報理工学部 2回生

大野 直哉
情報理工学部 1回生

宇佐 基史
理工学部 3回生

西村 雅貴
情報理工学部 2回生

武知 佳冴
理工学部 2回生

1. 活動概要

DTM班は音楽に関する知識を学び、オリジナル曲の製作を目指して活動した。具体的には、有志による講義や課題としての耳コピの実施などを行った。

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、コード、各種楽器やその打ち込み手法を正確に聞き取り、再現することを指す。DAWの操作に慣れ、音感や打ち込みの感覚を得るために行った。課題曲を指定し、全員が各自で作業をした後、相互に聞き合いアドバイスを行うことで理解の向上に努めた。

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・コードなどの理論を学習した。

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、拍子やテンポの説明を行った。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5度圈表の説明も行い、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、曲の制作に関する座学が中心であったため、後期では実際に曲を製作することを重視して活動を行っていく。また、難易度の高く説明を省いていた音楽理論や、ミキシング、シンセサイザーなどDAW上の技術についても学ぶ予定である。

4. 今後の目標

学園祭に向けてCDを作成することが目標に、作曲技術の向上を図る。

LT・プレゼン班

宇佐 基史
理工学部 3回生

奥川 莞多
情報理工学部 3回生

坪倉 奏太
情報理工学部 4回生

西見 元希
情報理工学部 4回生

文責：宇佐基史

1.はじめに

情報系の大学生における情報共有の為のアウトプットとして、ライトニングトーク(以下、LT)という5分間のプレゼンテーション形態を採用するという事は一般的である。“聴衆を想定した、極力感覚ではない、論理性に基づく有効なプレゼン技術又はスライド作成技術の獲得”を最終的な目標として各班員の発信力向上に努めた。特に、班内における正当性の基準を個人の裁量ではなく後述の文献より抜粋し定式化することに重きをおいた。

2.活動内容

班員の規模から、知見の共有を主とした活動に留まった。後期に班員どうしのブラッシュアップを組み込んだ形で再度同等のプロジェクトを開催する予定である。

2.1.参考群

以下に当活動において参考として用いた主なものを選定理由と共に記す。このほかにも技術の裏付けとしての実用例などに用いた参考もあるがそれらは量的な問題や厳密な趣旨にそぐわないものも含まれる為省略する。

2.1.1 ニックバーリー プrezent白熱教室

前編 2014年7月18日（金）23:00～23:54（NHK Eテレ）

選定理由 著者経歴によって選定した。過去4回のオリンピック招致を獲得したしているという経歴があり、近代的且つグローバルに通用するプレゼン技法の観点で信頼性があると判断した。また、内容の汎用性の高さに企画立案者が感銘を受けた点も上げられる。本プロジェクト班発足の動機でもある。

2.1.2 なるほどデザイン〈目で見て楽しむ新しいデザインの本〉

2015/7/31 筒井 美希（著）

選定理由 RCCを含む立命館大学情報系サークルにおけるデザインの参考図書として最も採用されていると判断したため。スライドの制作における技術の参考に採用した。本書の特徴として、豊富な実例が用意されていることが上げられる。

2.1.3 分かりやすい表現の技術 「分かりやすい説明」の技術 最強のプレゼンテーション15のルール (ブルーバックス) 新書

2002/10/23 藤沢 晃治 (著)

選定理由 内容の一般性の高さと"聴衆の想定"の観点で『なるほどデザイン』で言及されていない部分を補う内容を取り扱っていると判断したため。いくつかの大学では教科書として取り扱われている例もある。文字情報が多くなるほどデザインに比べ読みやすさは劣るが後述する当プロジェクト班で最も重要な視点として位置づけられる"聴衆の想定"の観点から参考に相応しい内容であると判断した。

2.1.4 How to sound smart in your TEDx Talk | Will Stephen

選定理由 文字情報で伝えることに限界のある。話し方や間の取り方といった部分のプレゼン技術の参考として取り上げた。舞台もTEDという比較的世界に開かれた場であり一般性や学術性の高さが求められる場であり、格式が求められる場合の参考としても有意性が認められた。直接的に取り扱われている内容も驚くほど薄く、当てはめる情報によって汎用が聞きやすい点もある。単純に内容としてもおもしろいのも選定理由の一つである。

2.1.5 The Design of Everyday Things: Revised and Expanded Edition 2013/11/5 英語版 Don Norman (著)

選定理由 正当性の根拠として一般性や事例数以外に心理学や人間工学と言った観点を取り入れるうえで、この本が結果的に生んだ社会への影響力の大きさから選定した。

2.2付録

当プロジェクト班において、すべてのプレゼン技術は"聴衆を如何に想定するか"を原理としていると定義した。趣旨、趣向を問わない範囲のプリミティブな内容が主である。次ページより総覧としてプロジェクト班を通して取り扱ったプレゼンの技法や注意点をまとめたものを掲載する。実際の班の活動では事例、根拠ともに紹介したが、要素だけを端的にまとめる以下のことおりである。また、実際の活動ではスライド作成において有用なツールの紹介や画像データ形式などの初步的な内容も取り扱ったが総覧では省く。

3.さいごに

付録は箇条書きから成る為、本誌読者の方々に内容が十分に伝わらないかもしない。しかもしも気になる知見があるようであれば是非、参考資料群の中から該当の知見を見つけ、より深い理解を得られたい。この資料とは違う、見るだけで面白い、与えられる知見を体言するかのような表現物であることを重視して参考資料を選定した。

正直、本誌より参考資料の方が面白い!!

2.1.4、2.1.2とか特にね!!

- 聴衆の想定
 - 内容理解に必要な予備知識の定義
 - 視線の動線の円滑化
 - 情報の優先度の明確化
 - 共通認識の定義
 - ユーモアの成立
 - 文化的タブーの排除
 - 慣用句の有意性
 - 時代性の考慮
 - リッチデザイン・スキューモフィズムの排除
 - 締め方の妥当性の検討
 - コロナ禍的配慮
 - 配色の再考察
 - 異なる通信環境下の公平性の担保
 - スライドの余白の再検討
 - ググらせるなどの実験
 - ※資料価値との両立
 - 聴衆の思考余地の利用
 - 妄想を掻き立てる言葉選び
 - 時間の確保
 - 実験,体験
- 全体構成
 - 3項目からなる要点
 - 三角関数的周期性
 - 要点の反復
- スライド構成
 - ページ数の記載
 - 目次
 - 初期段階での提示
 - 項目毎の進行具合の提示
 - 結びにおける総括としての目次の再提示
 - 注目点の決定(またその手法の妥当性)
 - 変色
 - 原色の採用
 - 地の色との反転
 - 拡大
 - 余白による孤立
 - 姿勢変更
 - アニメーション ※非推奨
 - 積極的画像表現
 - 媒体的制約の有意性の注意
 - 正確性の担保の注意
 - 伝達速度の確保
 - 言語制約の撤廃
 - 配色
 - グラデーションの排除
 - 原色の排除 ※注目点を除く
 - カラーパレットの採用
 - 4色以下の基本色の設定

- 必要情報以外の取捨選択
 - メタ情報の形式化
 - 聴衆の基礎知識の利用
 - アフォーダンス
 - 対応付け
 - 新情報の5個以下制限
 - 箇条書きの利用
 - 非文化
 - 型の明確化
 - 1行
 - 行頭
 - 同一単語の排除
- 音声情報
 - 情報量
 - 日本語→300文字/分
 - 非母国語→訳文200字/分
 - 情報の差別化
 - トーン
 - 間
- 再現性の向上
 - 練習
 - 本番環境の再現
 - 発声を伴う練習
 - 理想として本番時間の60倍の練習時間
 - デバイスのエラー確認
 - 緊張対策
 - 練習
 - ルーティンの体得
 - 練習
 - 練習
 - 練習
 - 練習
- 理系的配慮
 - ソースコードの視認性の確保
 - フローチャートによる代用
 - 想定された聴衆のスキルのより厳格な想定
 - 科学的根拠の提示
 - 数値表現
 - グラフによるビジュアル化
 - 配色の注意
 - 立体化の検討
 - 種別の選択
 - 円 →割合
 - 棒 →絶対値差
 - 折れ線 →時間推移
 - 表によるビジュアル化
 - 参考資料
 - 該当スライド下部の出展記載
 - 論文形式
 - 質疑対策
 - ページ番号の表示
 - 末尾の参考総覧

OS班

きょうすけ
情報理工学部 2回生

lufe
情報理工学部 2回生

クマさん
理工学部 2回生

Jin
情報理工学部 2回生

3^5
情報理工学部 4回生

1 活動概要

この班は書籍「ゼロからのOS自作入門」を教科書としてOS自作およびその拡張を行い、OSの処理内容に関する理解を深めることを目標としている。OS自作の経験は効率的なアプリケーション制作の探求やシステム障害の原因調査などに役立つことが期待される。

活動は主に教科書の内容解説を行い、実装は各自活動時間外に行うこととした。資料はScrapbox(<https://scrapbox.io/rcc-project2021-OS/>)にまとめ、班員と共有した。

2 各章の学習内容

2.1 第0章 OSって個人で作れるの?

OSとは制御プログラム、言語処理プログラム、サービスプログラムなどのプログラムを含むソフトウェアでWindows、MacOS、Linuxなどの巨大OSはソースコードが数千万行規模になる。しかし、機能を必要最低限に落とせばソースコード1万行程度のOSも作れる。

OS自作には既存のものを拡張するやり方と全部自分で作るやり方が存在する。前半期では必要最低限の機能を実装したMikanOSを一から作り、後半ではその機能を各自で拡張することに重きを置く。

2.2 第1章 PCの仕組みとハローワールド

バイナリエディタを用いて実行ファイルをバイナリを打ち込み作成した。作成した実行ファイルを用いて画面にHello Worldの文字列を表示させた。次に、C言語でClangとLLDを用いてコンパイル、リンクし、実行ファイルを作成した。最後にqemuというエミュレータを用いて画面上にHello Worldの文字列を表示させた。

2.3 第2章 EDKII入門とメモリマップ

EDKIIというUEFI BIOSまたはUEFI BIOS上で動くアプリケーション用の開発キットでHello、Worldを実装した。EDKIIには環境変数設定用のスクリプト(edksetup.sh)やビルド、ツールチェーンの設定用ファイル(target.txt、tools_def.txt)、他のプログラムから利用される基本ライブラリ(MdePkg)などを含んでいる。Hello、Worldの実装にあたっては1章で実装したものを書き直すという形をとった。具体的にはEDKの基本ライブラリをincludeして、構造体や関数を流用した。こうすることでより簡潔に記述することができた。またメモリマップについても学習した。

基本的なコンピュータはプログラム内蔵方式という、メインメモリに書かれた命令をCPUが実行していく方式をとっている。もし既にメインメモリの使用している部分に上書きするとプログラムは正常に動作しないため、メインメモリのどこにどんな情報が書かれているかを管理する必要がある。メモリマップにはそういう情報が書かれており、メモリマップを参考にしてメインメモリに情報を書き込んでいく。

2.4 第3章 画面表示の練習とブートローダ

OSを起動できるようにするためにカーネルとそれを呼び出すためのブートローダを作成した。ブートローダはUEFIアプリとして作成し、ブートローダはUEFI形式に縛られず自由に開発できる様にするためにELFバイナリとして分けて開発した。

2.5 第4章 ピクセル描画とmake入門

makeというコンパイルやリンクを自動化するツールを用いてbuildを自動化した。自動化を行なったことにより開発効率を向上することができた。

次に、ピクセルを描画する処理の実装を行なった。また、メモリの大きさを計算する処理にバグがあり、LOADセグメントのファイル上のサイズとメモリ上のサイズが異なるためLOADセグメントを正しくメモリ上に読み込まれるようにした。

2.6 第5章 文字表示とコンソールクラス

3章、4章で習得した画面をピクセル単位で塗りつぶす技術を用いて、文字の形をした図形を描写した。このために塗りつぶす部分を1、塗りつぶさない部分を0とした8*16の配列(フォント)を用意した。文字が描写できるようになったので簡単なコンソールを自作した。この時点では外部入力を受け付ける機構はまだないので出力専用のコンソールとなる。コンソールにメッセージを次々に表示させるために改行とスクロール機能を実装した。改行は文字を表示させるy座標をずらすことで実装し、スクロールは画面を塗りつぶして先程まで表示していた文字を1行ずつずらすことで実装した。

2.7 第6章 マウス入力とPCI

先程までの章をもとにマウスカーソルの絵を二次元配列を利用して描画した。また、マスカーソル以外の画面上に現れるものに関しては二次元配列に情報を格納することで実現した。このままではマウスカーソルはただの絵でしかないので、動かすことができない。そこでPCIバスに接続されているデバイスからxHCを用いてUSBバスからマウスを探し初期化を行うことでマウスからデータを受信できるようにしマウスカーソルを動かせるようにした。なお、USBドライバーについては複雑なため断念した。

3 活動で得られたこと

まずOSというものがどのように実行されているのかといったOSの基礎知識の確保である。次にOSはどのような作成手順を踏まれるのか等のOS開発の手順と経験である。プログラムの写経や、独自の改変をすることでよりOS開発の経験によって得ることができたものである。

4 反省点

前期における活動は6回に分けて行った。問題点は以下の2つである。1つ目はOS班では実装を終了時間が遅くなる見込みがあったためプロジェクト活動内にいれなかつた。その結果、開発する時間が少なかったため、全班員の開発が間に合わなかつた。2つ目は当初の進行よりも遅れているという点である。その結果、前期活動を行った内容の説明が不十分になってしまい、後期の残りの内容に回すことになったという点があつた。

5 今後の展望

前期の活動において行われた本の輪読や写経、独自の改変を通して班員の技術力や、OSに関する知識を深めることができた。後期の活動では、さらにOS開発に関する知識を深めることや、独自の改変を施したOS開発することによって班員の技術力を高めること、OSの開発の楽しさを学ぶ予定である。OS開発のさらなる進展や技術力向上を目指し今後の活動を行っていきたい。

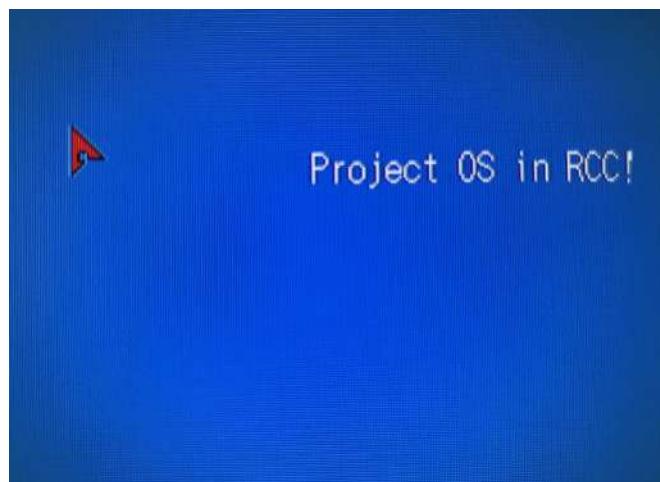
6 班員による成果物

6-1 仕様

MikanOSを独自のカーソルの表示や、出力する文字列の場所を改良した。

6-2 実装

ArchLinux上でVirtualboxとVagrantを用いて仮想環境を作成し、Ubuntuを立ち上げた。Ubuntu上で開発を行い、内部でqemuというエミュレータを用いてOSの実行環境を構築した。これを使えば誰でも簡単に手元のMikanOSを立ち上げることができる。



6-3 成果物に関する反省と感想

6章マウスカーソルを動かす実装ができなかつた。しかし、初めてのArchLinux上で仮想環境を作成し、VagrantとVirtualboxで開発環境を整えるといった経験を得ることができた。加えて誰でも簡単に手元のMikanOSを立ち上げができるといった環境構築を自動化するプロセスにも挑戦した。単にOS開発を学ぶ以上に技術力向上と知識を得ることができた。

Web 班

星名 藍乃介 情報理工学部 3回生	hane 情報理工学部 1回生	澤谷 祐樹 情報理工学部 1回生
竹中 萌 経営学部 1回生	本多 峰之 情報理工学部 1回生	husky 情報理工学部 2回生
kawai 情報理工学部 2回生	sno2wman 情報理工学部 2回生	lufe 情報理工学部 2回生
きょうすけ 情報理工学部 2回生	クマさん 理工学部 2回生	仁 情報理工学部 2回生
FKD 情報理工学部 3回生	OK 情報理工学部 3回生	阿部 健太朗 情報理工学部 3回生
宇佐 基史 情報理工学部 3回生	中尾 龍矢 情報理工学部 3回生	やぎちゃん 情報理工学部 3回生

1. 活動概要

Web 班は、Web の仕組みや構成技術についての知見と実装力を身につけることを目標とし、班員間で Web サービスや開発手法について共有しながら、個人制作物を開発するプロジェクトである。

2. 活動内容

活動前半では、Web 開発時に有用なリファレンスや教材の共有、初学者のためのチュートリアルや World Wide Web の仕組みについての共有を行った。なお、活動成果の詳細は外部 Web サービス Scrapbox (<https://scrapbox.io/rcc-web2021/>) でまとめられているので、本報告書で掲載しきれなかった内容に関してはそちらを参照されたい。上記の共有事項以外にも、班員による Web 開発に関する知見がまとめられている。

活動後半では、班員による個人制作物の開発を行った。

3. プロジェクトスケジュール

- 第1回 Scrapbox 企画の説明、教材紹介、環境構築
- 第2回 目標成果物の決定、Web の仕組み超入門、HTML/CSS/Javascript チュートリアル
- 第3回 成果物開発開始、Web フロントエンド技術の歴史、バックエンドチュートリアル
- 第4回 成果物開発、Web フロントエンドチュートリアル
- 第5～7回 成果物開発

4. 班員の制作物

班員の個人制作物とその概要について記載する。より詳しい情報は RCC 公式 Web サイトの 2021 年度前期プロジェクト活動報告書でまとめられている。

http://www.rcc.ritsumei.ac.jp/pr/wp-content/uploads/2021/08/2021-1_Project-Web.pdf

1. クロスサイトスクリプティング (hane)
 - クロスサイトスクリプティング脆弱性のある Web アプリケーションを自作し、自身で XSS 攻撃を行うことで理解を深めた。
2. ポートフォリオサイト (澤谷 祐樹)
 - 自身のポートフォリオである。筆者自身についてのページ、普段利用している discord のサーバーへのリンク、Twitter にリンクした画像や絵を並べたページを作った。
 - <https://kaze8888.github.io/kazeportfolio/>
3. BTS ファンサイト (竹中 萌)
 - 韓国の 7 人組男性ヒップホップグループである BTS のメンバーについて紹介した Web サイトを制作した。
4. チャットアプリ (本多 峰之)
 - Udemy で公開されている「React 初学者用のチャットアプリ開発」の解説動画を視聴し、仕組みを調べながら開発を進めた。最終的には、様々な職種の人がマッチングできるサービスを作り、コワーキングスペースの中で新たな価値のあるビジネスを創造したいと考えている。
5. Numer0n (husky)
 - ヌメロンという数字当てゲームを JavaScript で制作した。
 - <https://husky21st.github.io/numeron/>
6. ECサイト (kawai)
 - 個人で作った Web サービス、モバイルアプリ、ブログアカウントなどを売買できるサービス
7. henken.club (sno2wman)
 - 偏見本棚という概念があり、これをソーシャル Web サービスとしてプラットフォームを提供できれば面白そうと思い開発している。十分なデータが集まれば、ある程度の傾向を掴んで精度の高いレコメンドなども行えると期待している。
 - <https://github.com/henken-club>

8. lufe.site (lufe)
 - 自身の Web サイトである。4つのページで構成されており、経歴、興味分野、SNS や github 等のアカウントを知ることができる。自分自身を世に発信する、見える制作物を作成したいという動機から制作に至った。
 - <https://lufe.jp/>
9. 購入申請 API (クマさん)
 - 会計局の既存のシステムの保守が難しいため、その更新を目的として今回の API を実装した。実装に当たって保守管理のしやすさを優先した。機能としては備品購入申請の受付、書籍注文、月次決算情報の自動作成といったものを備えている。なお、一部機能は未実装である。
 - https://github.com/CityBear3/apply_system
10. Tetris made with React (FKD)
 - React を用いて Tetris を作成した。
 - <https://fukada6280.github.io/react-hooks-tetris/>
11. ポートフォリオサイト (OK)
 - Next.js や Tailwind CSS を使用してポートフォリオサイトを制作した。
 - <https://okgwknt.github.io/>
12. RCC部室予約サイト (阿部 健太朗)
 - 2021年8月現在 RCC の部室は入室不可だが、大学から対面活動が許可された場合には人数制限をかけた上で利用が可能となる。そこで、いつ、誰が部室を利用するのか予定を管理するサービスを開発した。会員は Google カレンダーに予定を追加する要領で、予約の追加・編集・削除を行い、制限人数に応じた抽選結果が Slack に通知される。
 - <https://crms.abelab.dev/login>
13. 基礎文法解釈の過程の残骸の GitHubPages での公開 (宇佐 基史)
 - 参考にしたのは [MDN Web Docs](#) というサイトである、特筆に値するものでもないが、HTML/CSS において基本的なコマンドと思われるものを単純に羅列し、Web 上に公開するまでを目標とした。筆者の学習過程の軌跡である。
 - <https://thetaleisalsowhite.github.io/>

14. 立命館大学交響楽第一回オペラ公演公開までのカウントダウンの掲載（宇佐 基史）
- 立命館大学交響楽団の方式 HP は管理において Jimdo という Web ホスティングサービスを用いているが、一部対応外の機能においては HTML に直接書き込む形で実装が可能である。カウントダウンタイマーの実装を経ることで基本的な JavaScript の仕様の理解を目指した。
 - <https://www.ruso60.com/>
15. ダウンジングサイト（中尾 龍矢）
- 画面幅を取得する JavaScript 関数や HTML の canvas を用いて、ブラウザ上でダウンジングゲームができるサイトを制作した。
 - <https://downsing.netlify.app/>
16. Web ワードクラウド（星名 藍乃介）
- 文章を可視化する手法として、ワードクラウドと呼ばれる、頻出語を頻度に比例する大きさで並べて図示する方法がある。今回はブラウザオンリーでワードクラウド画像を生成し表示する Web サイトを開発した。
 - <https://ran350.github.io/web-wordcloud/>
17. onakasuita.city（やぎちゃん）
- onakasuita.city は、空腹であるという意味の「おなかすいた」という文に大阪府に存在する市である「吹田市」をかけた「おなか吹田市」という文を、「.city」トップレベルドメインを用いたタイトルと同名のドメイン名を使用した Web サイトである。
 - <https://onakasuita.city/>

4. 総括

Web 開発初心者の班員も多くいたが、前章で挙げたように HTML/CSS や canvas、React といった技術の基礎文法を習得し実装する力を身につけることができた。教科書やドキュメントを読み進めるだけではなく、実際に手を動かしながら学ぶことで、Web 開発の流れやリファレンスの参照方法、デバッグの方法などについての理解も深めることができたと考える。

また、[Scrapbox](#) で Web 開発時に便利なツールやサービス、拡張機能、リファレンスなどをまとめていたが、本プロジェクト活動に留まらず、今後も有効活用していきたい。

スロット班

堀田 隆成
情報理工学部 3回生

阿部 健太郎
情報理工学部 3回生

八木田 祐悟
情報理工学部 3回生

深田 紘希
情報理工学部 3回生

桐井 優実
情報理工学部 3回生

田邊 雄士
情報理工学部 3回生

1. 活動概要

スロット班はオリジナルのスロットを作成する過程を通じて企画、要件の整理、設計、開発などのチーム開発でのプロセス、効果的なUIの設計などのテクニックを学ぶことを目標としたプロジェクトである。また本活動では話し合いを重視して活動を行う方針を立てた。

2. 活動内容

週に一度以上の活動を目標とし、特定の曜日などを定めずに活動を行った。活動の進行は以下のように行った。

1. 環境構築、スロットについて学ぶ、メンバーの役割決め
2. Unityなどの使い方の学習
3. 各自で開発を行う
4. スロットで遊ぶ

3. スロット制作

スロットを作成するにあたり、Dropbox Paperを用いて必要な機能、欲しい機能、工程、クラス図などを書き出した。

要件定義では機能要件として以下の事項を定義した。

- ゲーム性
- リール
- 筐体構成
- 図柄

3.1. ゲーム性

作成するスロットのゲーム性として、一般的にノーマルタイプと呼ばれるものを想定した。ノーマルタイプとは、7揃いなどのボーナスによりメダルを増やすタイプのスロットのことを指す。ボーナスに当選するまで緩やかにメダル枚数が減少するという特徴をもち、シンプルなゲーム性が特徴である。

3.2. リール

スロットの回転部をリールという。リールは一つにつき10種類以下、21の図柄を持つ。一般的なスロットはリールの回転が始まる時点でリールの止まる部分があらかじめ決まるという機能があり、機能を実現するためにリールは4コマまで回転方向に滑るといった機能を持つ。

3.3. 筐体構成

筐体は三つのリールを持ち、それに対応した三つのボタン、リールを回転させるレバーを持つ。

3.4. 図柄

図柄では図柄の並びによって完成する役について、またそのデザインについての定義を行なった。成立する可能性のある役としては

- FullHD 捱い
- ビッグボーナス (Seven Seven Seven)
- レギュラーボーナス (Seven Seven Bar)
- cherry 捱い
- WaterMelon 捱い
- Bell 捱い
- Replay 捱い
- ハズレ

図柄のデザインは次章のデザインで述べる。

4. デザイン

スロットの遊技機本体のデザインは、昔ながらのクラシックな回胴式遊技機にRCC要素を取り入れる形で行った。実装しやすく資料も多く存在していることから、このデザインを採用した。また縦長になりすぎないように心がけた。

デザインの作成にはFigmaを用いた。FigmaはWeb上でUIデザインをすることができるツールである。複数人でのリアルタイム編集が可能であり、複数人が1つの作品を手がける本プロジェクトにおいて適当な選択であった。



図1. 筐体デザイン

図柄のデザインとしてSeven、Full-HD、RubyCherry、JS、Tapimon、RCCBAR、ReactReplay、あたまのわるいひとの8種類を作成した。図型は色と形と価値の3点のうち、少なくとも2点を重複しないよう作成した。リールは素早く動作するため視認性が重要である。

さらに遊び要素として RCC 会員の素材やプログラミング言語やライブラリといった技術的な図柄を積極的に取り入れた。



図2.図柄

5. 実装

リール制御では各リールに与えられた図柄の並びを格納した配列を参照しランダムで初期配置を決める。その後レバーをクリックした後に 0~65536 の間でランダムな値を生成し、その値により成立する役を受けとり、リールに対応したボタンを押すと次にその図柄が止まるまでリールを回転させることで任意の役を成立させる。

コイン枚数の管理としてリール制御のコードで、制御とは別に盤面でどの役が成立したか判別する関数を作成し、対応した役を返す。

演出として本機種は FullHD 削い、ビッグボーナス、レギュラーボーナスが成立するゲームでは RCC のロゴが発光し利用者にボーナス当選を告知する。前述しているように成立する役を受け取ることでランプの発光を管理する。またレバー、ボタンを押した際に音源が再生される。

6. 機種概要

役にごとの確率、払い出し枚数は以下の表のようになっている。また機械割は 113% となっている。

成立役	確率	払い出し枚数
中段チェリー後 FullHD	1/2	458
中段チェリー後 BB	1/2	312
BB 単独成立	80/65536	312
BB チェリー成立後	100/65536	
BB スイカ成立後	50/65536	
RB 単独成立	56/65536	108
RB チェリー後成立	100/65536	
RB スイカ後成立	100/65536	
中段チェリー	26/65536	4
リプレイ	4553/65536	
ベル	6553/65536	9
弱チェリー	5625/65536	2
スイカ	2850/65536	9

7. 今後の展望

今回作ったスロットはノーマルタイプと呼ばれるシンプルなものであるため、今後はいわゆる A+rt 機や AT 機を作りたいと考えている。演出も大変シンプルなものとなったため、今後は複雑な演出によるボーナス当選の示唆やリーチ目、それを実現するためのリール制御を実装したい。また今回の経験を生かし、パチンコ遊技機などの制作にもつなげたいと考えている。

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あります。

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、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。

1961年～1980年のRCC

学部/回生 情報理工学部 SN 一回生

名前 羽田秀平



・はじめに

私が初めてRCCの部室に入った時、はじめに目にしたものは4、5台のPCや工具・3Dプリンター・プログラミングの本・ピアノ(?)などここには書ききれないくらい多種多様な備品が並んでいました。とても魅力的だったのですべての備品を確認せずにはいられず、課題そっちのけで1つ1つ確認していると備品棚の下に年代が書かれた段ボールがあり、中には1961年～と書かれた段ボールを見つけました。中に入っていたものは日誌やOBの自己紹介カードがまとめられたものや学園祭のパンフレットが入っていました。せっかくなので私が書く会誌の内容は当時の時代背景と昔のRCCにしました。

・1960年代

文献から確認できたRCCの活動していた年は1961年で、この時のRCCは物性論研究会という名前で活動していました。そして当時のOB・OGが会誌を初めて作成し、永久保存することを決めたみたいです。1960年代で起こった歴史的な事柄を調べると60年代は冷戦の時期で、ちょうどベルリンの壁が建設されて崩壊するまでの時代でした。また月面着陸もこの時期です。日本では高度経済成長期のど真ん中です。この時に活動していたOBの方達は現在80歳くらいなのでスケールの大きさを感じました。この頃の活動場所は現在の衣笠キャンパスに位置した場所にあるX線実験室が主な活動場所で、活動内容は今みたいにPCが普及していないので、X線に関する本の輪読やベクトル解析・結晶炉の制作・半導体の研究といった今の活動内容とは違ったものでした。1962年の会誌にはテレビに使われたトランジスタの話が会報として載せられています。1962年から関西物理科学生懇談会(KBK)にも参加し、京大生や神大生・市大(現ハム大)の方々と交流していました。また60年代は学生運動が盛んでしたがこれに関する文献は見つかりませんでした。

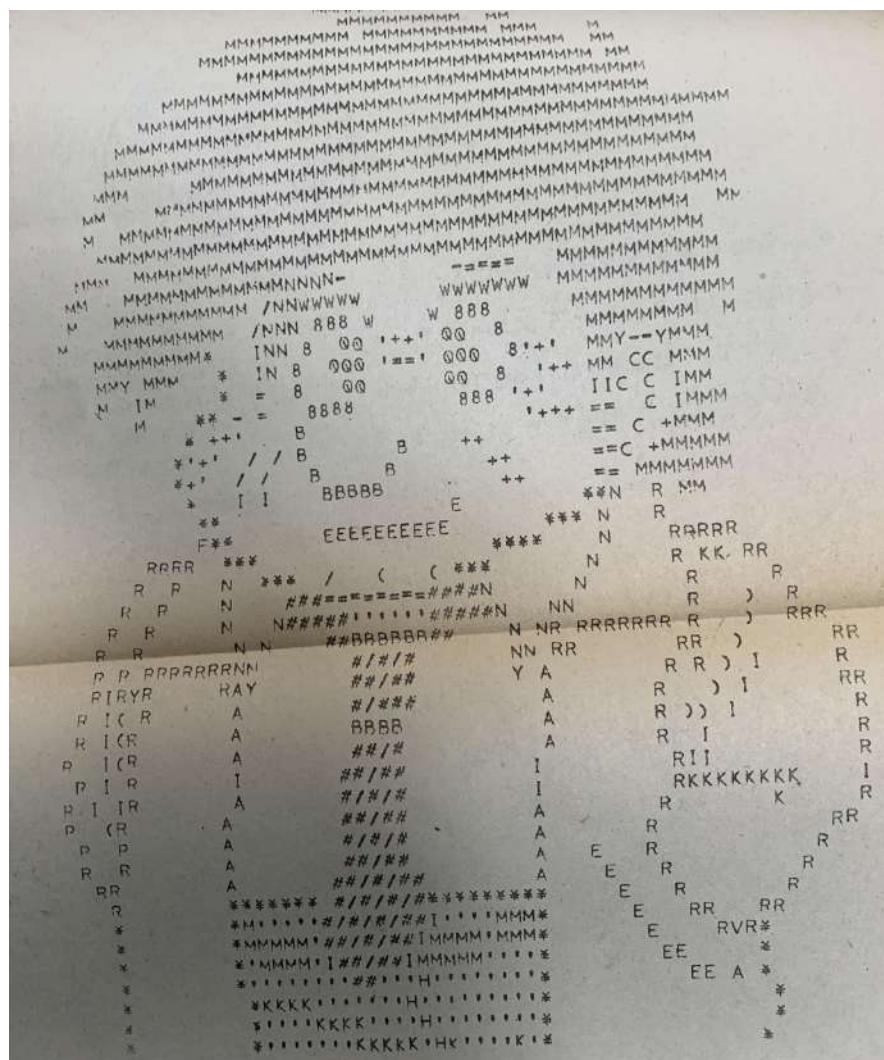
・1970年代

1976年に初めてIT関係の話が会誌に掲載されました。ソフトウェアという題名で電子計算機の構成やプログラムの処理手順・ファイル・プログラミング言語・OS・CPU・プログラミング言語についての記述がありました。

とくにプログラミング言語の項目では、FORTRANやCOBOL・ALGOL・PL/Iについての解説がありました。1977年には命令とアドレスについての説明があり、この時期から今の活動内容にちかいものになってきてました。また当時のOBの方々は詩や歌を自作したものを掲載されていました。ビ○ト○ズの解散を憂いた方もみられます。1978年から現在も続いている自己紹介カードが保管されていて、特に中○み○きさんやア○サ・ク○ス○イ・山○○恵・キヤ○○○ーズ・宇○○艦○○ト・郷ひ○○など今でもよく知られている人を自己紹介カードの趣味の欄などに記入している人が散見されました。1978年にはAA(アスキーアート)を載せている人がいました。

・さいごに

高度経済成長期など時代の出来事によって人の明るさが変わったりするところや部活運営の難しさについて見ることができました。



へへへ！お金を分析だ

学部/回生 情報理工学部・SAコース1回生

名前 Tucker



はじめに

AIとか機械学習とかデータサイエンスとか、いろいろなIT用語がSNSやネットニュースで話題になっています。いまでは中学教育のカリキュラムに統計学や確立、データ処理などの技術が加わるケースも全国的にみれば珍しくない状況になってきています。

そこで私は、Pythonによって簡単に実現できるデータサイエンスのコードを調べ実践してみました。なかでも今回はお金について（株式）のPythonの利用法の超超超基本を紹介します。

Pythonって？？

Pythonを知らない人のために。Pythonはオランダで開発されたプログラミング言語で、計算やデータ処理に特化しAIや機械学習に用いられています。またその記述の簡単さから、初学者からでも始めやすく、多くのユーザーが利用しています。

データ分析！！

Pythonには沢山のライブラリが用意されています。そのなかで日経平均株価を取得するライブラリ、データ処理をサポートするライブラリ、グラフを表示するライブラリ、インポートして準備を行います。

```
!pip install pandas_datareader
```

```
from pandas_datareader import data
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

これで実行すると準備が整いました。ここまでくれば後は簡単。

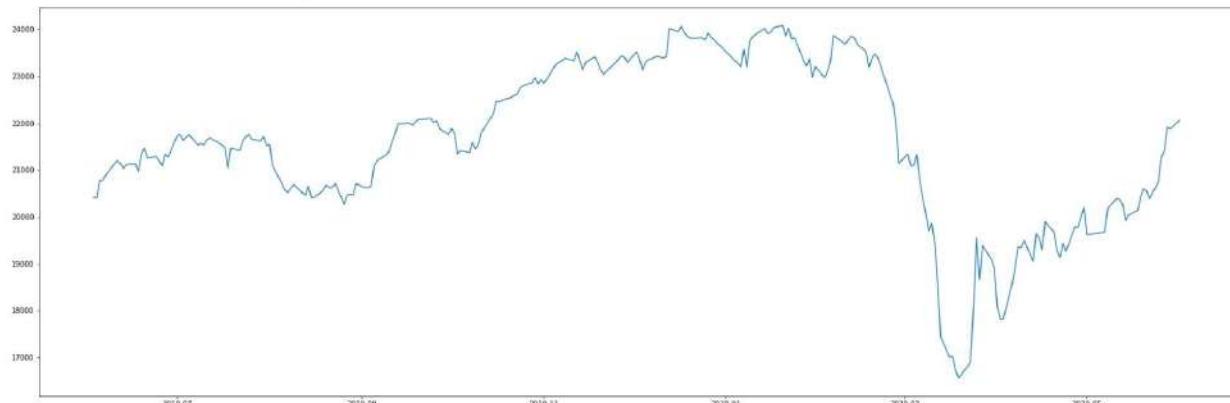
```
start = '2019-06-01'
end = '2020-06-01'

df = data.DataReader('^N225', 'yahoo', start, end)
//任意関数 = date.DataReader('銘柄', 'yahoo', '始まり期間', '終わり期間')
```

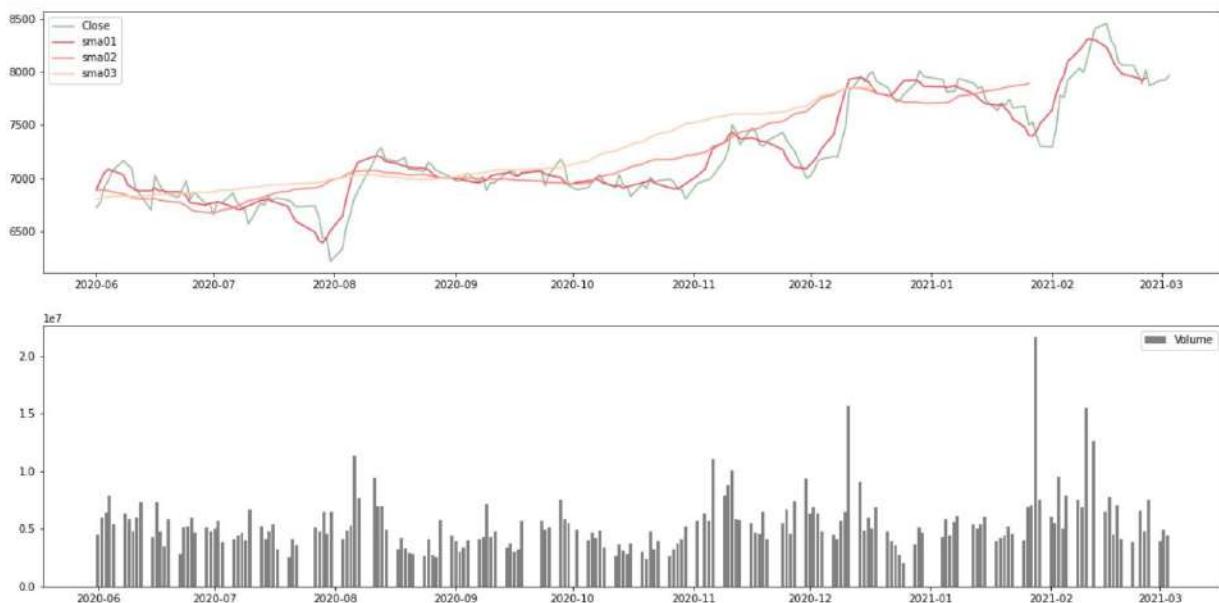
これで関数dfに1年分のデータが取得されました。データの表示方法はdf.head(10)で上位10種類が表示されます。

株の分析にはいろいろな方法があります。はじめは中でもグラフを作る過程を話します。

```
date=df.index  
price=df['Adj Close']  
plt.figure(figsize(38, 10))  
plt.plot(date,price)  
  
[  
@  
matplotlib.lines.Line2D at 0x156a7d2ef10>]
```



このようにたった4行でこんなグラフを表示進事ができます。又重ねたり、棒グラフを表示させたり、色を変えたりラベルうをつけたり、様々な分析ができます。



今回は、余り説明できなったが、ローソク足やボリンジャーバンドなど株式投資の分析には欠かせない要素も沢山あります。そのためまだまだ勉強していろいろなことがPythonができるようになりたいと思います。

幾何学模様を作ろう！

学部/回生	情報理工学部知能情報コース一回生
名前	大野直哉



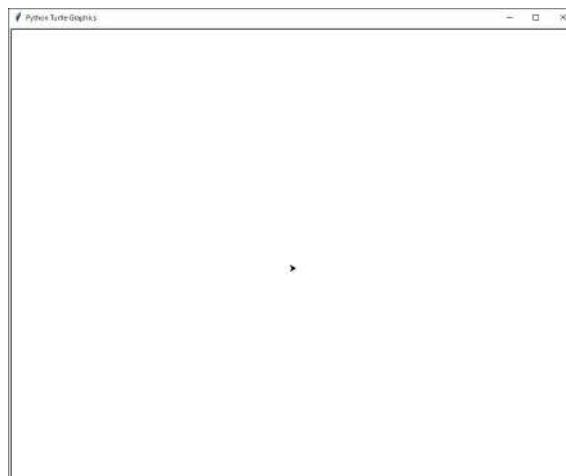
こんにちは！

幾何学模様っていいですよね。皆様は自分の思うままに幾何学模様を作りたくなる衝動に駆られませんか？しかし幾何学模様を作ろうと思ってもどうやって描けばいいのかわからなくて途方に暮れてしまいますが！Pythonにはそうした問題を簡単に解決できるライブラリがあります！

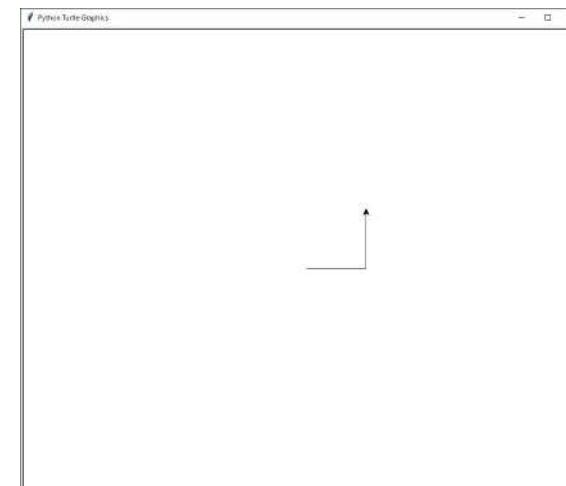
それが「Turtle」です。TurtleはPythonのGUIライブラリの一つであり、Pythonに標準でついているため簡単に実行できます。GUIライブラリと聞くと複雑な命令があり難しいと思うかもしれません。Turtleにはそんなものありません。Turtleは、直線を描く、直線を曲げる、弧を描く、といった線の描写に特化していて、他のGUIライブラリと比べて命令の数が少なく簡単です。

実際に実行してみましょう！

このように初期状態では画面の中心に矢印が表示されており、矢印の前後方向へ直線を描画することが出来ます。

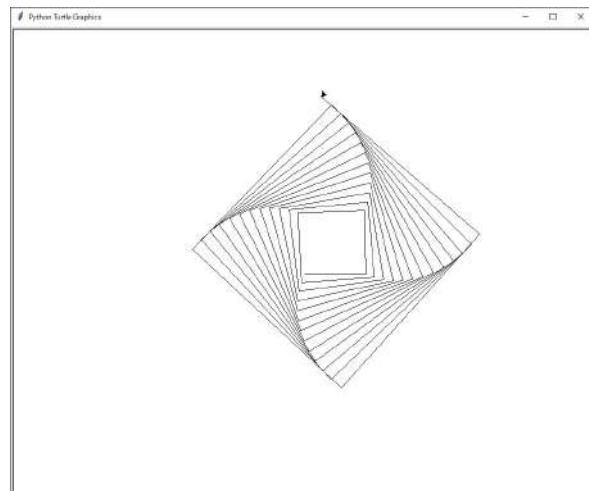


例えば前へ100進み、左に90度曲がってさらに前へ100進むという命令をするとこのような直線が現れます。

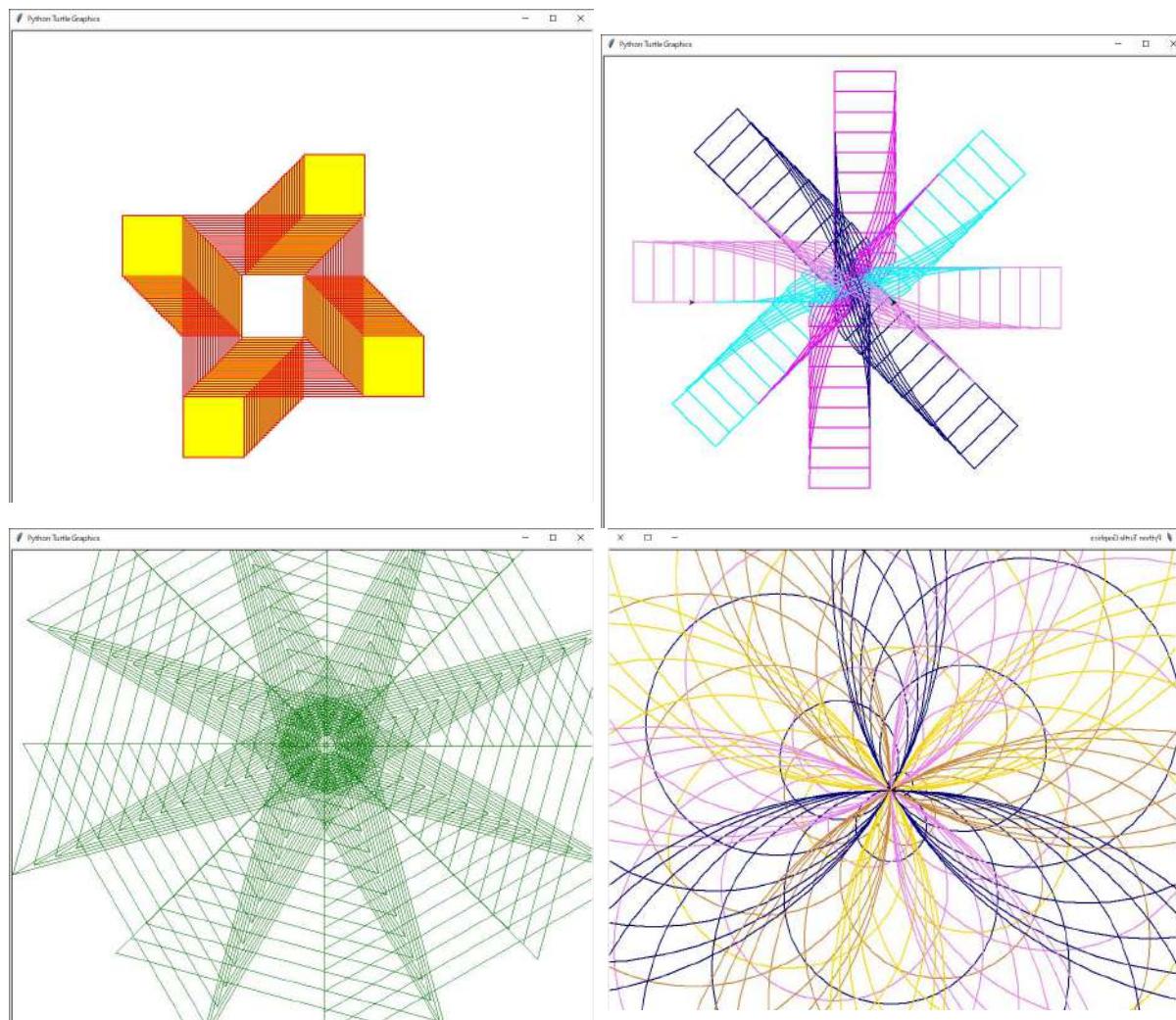


では次に前に前に100進んで91度左に曲がる命令を50回繰返し、また繰り返すたびに進む長さが1ずつ増えていく処理をしましょう。

そうするとこのような複雑な模様が現れました。このように線の描写と繰り返しを組み合わせることで簡単に幾何学的な模様を作ることが出来ます。



さらに、Turtleでは、線や線で囲まれた部分に色を付けることが出来て、模様を色鮮やかに彩ることが出来ます。



これであなたも自分で幾何学模様を作ってみましょう。

RGBを使いたい

学部/回生 情報理工学部 画像音 1回生

名前 swtn

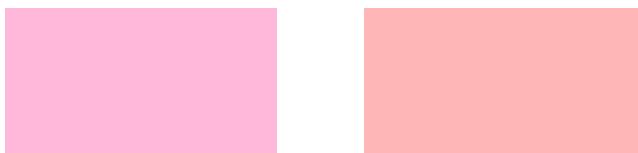


こんにちは。RGBを使いますでしょうか。私はあまり使ったことがないです。cygwinとかprocessingで画像を表示するときにつかったくらいだと思います。もしかしたらあるところで沢山使うかもしれません。

普段自分が色を指定するときは、カラーコードが多いような気がします。例えばHTML,CSSで色を指定するときとか。CSSなどではwhiteとか入力しても色が変わるのは便利です。

それはそうとRGBを使って何か書いてみたいなと思った結果、似たような濃淡？コントラストの色をザーッと表示するようなものがつくれそうだなと思いました。それを作ってみようかなと思います。この前先輩方から色を選ぶ際にとても使えるサイトを教えて下さいました。ある色を選ぶと補色や同系色が表示されるめちゃくちゃ便利なサイトでした。そのサイトがあればこんな事する必要はないと思います。

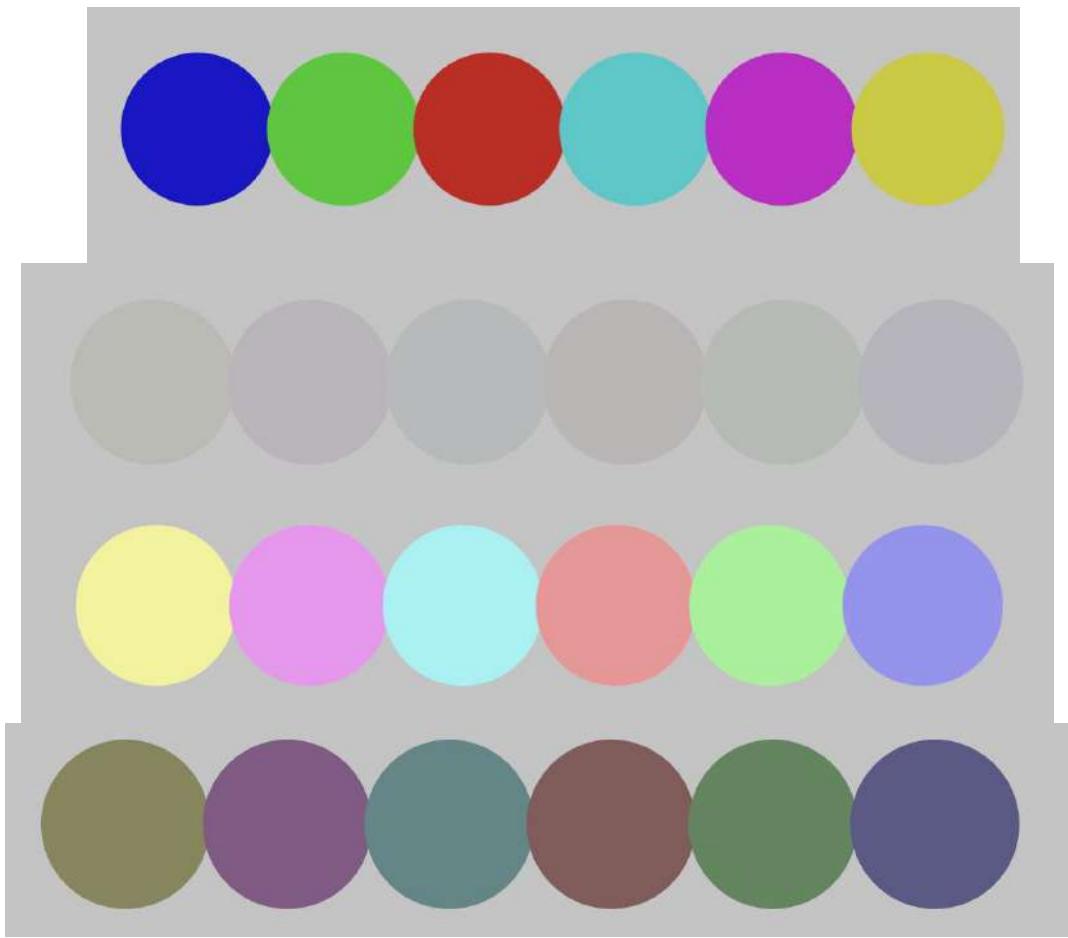
processingを使って作ってみようと思います。



この二つのRGBはそれぞれ左が(255,183,219)、右が(255,183,183)です。心なしか、顔文字の色に似てるような気がしますが、この二つの値を見ると二つの255,183という数字は共通しているので、二つの値を固定させつつ、残りの一つの値だけ変えれば色が作れそうな気がします。

変数を2つ用意し、そのうちの1つをRGBのどれか二つの値にし、残りの1つを余ったRGBの値にしました。多分組み合わせは6通りあるので6色の色が出るようにしました。

実行した様子が以下の写真です。



違う6色が作れたと思います。

変数の数字は乱数なので実行するたびに色が変わるはずです。
これを作ることによる意義はあまりないかもしれません、変数の
値を変えずに色々な色が作れるのは良いですよね。

ランダムで色が生成されるので気分に合わせて色をスポットでスポ
スポして絵の配色に使ってみるのも良いかもしれません。

MIDIパッドのすゝめ

学部/回生 理工学部ロボティクス学科 1回生

名前 山口 蒼太



MIDIパッドというコントローラーを皆様ご存じでしょうか？パソコンで音楽を作る際のツールの1つで、カスタマイズ性が高く、とても奥が深いツールになります。意外とその奥深さを知らない方が多いと思い、紹介してみることにします。

MIDIパッドって？

パソコンにつなげて使用するボタンがたくさんある板です。ボタンをポチポチして演奏します。以下のような見た目をしていて、基本的には音楽制作に使う機材になります。



引用: amazon.co.jp

どうやって演奏するの？

主に「ワンショット」と「ピアノ亜種」の2つあります。(名付けは適当です)

1.ワンショット

光るボタンに好きな音を割り振って、好きなタイミングで押して曲を作っていきます。音はドラムのように1度ボタンを押せばその楽器が叩かれるものでも、楽曲や声の素材をぶつ切りにしたものでもなんでも登録できます。MIDIパッドの最も主流な使い方です。

2.ピアノ亜種

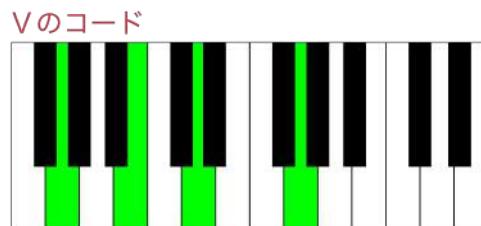
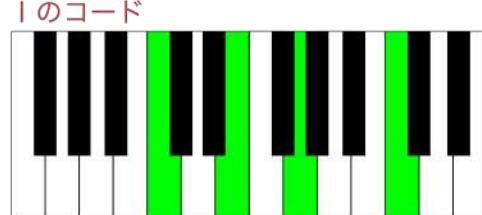
パッドでもピアノを演奏することができます。

パッドって鍵盤の形に並んでないじゃん

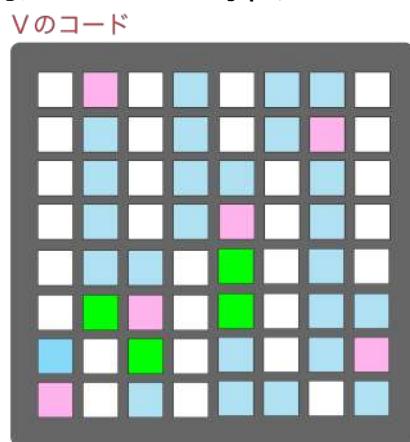
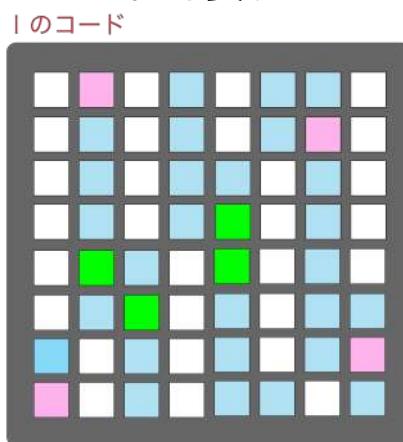
→逆にそのほうが弾きやすいんです！！パッドはボタンの並びの形を覚えるだけで和音を演奏できますし、転調だってとってもかんたんです。

ex.)お辞儀の「ジャーン ジャーン ジャーン」

これは I → V → I のコードでピアノで表すと以下のようにになります



これをパッドで表すとこんな感じ。(緑のところが押すボタン)



左右の I と V のコードで、パッドの押しているボタンが違うのがわかるでしょうか？鍵盤側は複雑に移動してるのでに対し、パッド側は同じ形のまま下に1つ下がっただけです！このように、コードを弾くのには、鍵盤だと複雑になりますが、パッドだととてもかんたんになります。

・ MIDIパッドの応用

MIDI信号を使えばいろんなソフトをMIDIパッドで操作できるようになります。もちろんMIDIキーボードでもできますが、パッドはLEDで好きな色に光らせることができるので、視覚的に、感覚的にわかりやすいコントローラとなります。複雑なキーボードショートカット、マウスのクリックではやりにくい操作等々、自分の好きな機能を割り振って、どんどん効率化しましょう！

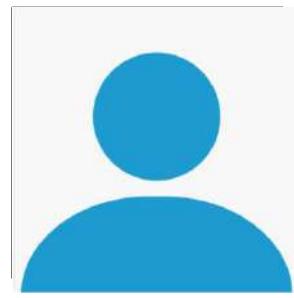
・ おわりに

いかがだったでしょうか？文章では伝わりづらい部分も多々あります。いろんな方がYoutube等に動画を投稿していらっしゃるので、それをご覧になると、さらにパッドの魅力が伝わると思います。僕もまだまだパッドは初心者なので、いっちょまえにライブパフォーマンスができるくらいになりたいですね。

ARP Spoofingのやりかた

学部/回生 理工 ロボティクス 1回生

名前 daiki



ARP Spoofingとは

ARP(IPv4のネットワーク層のアドレス解決プロトコル、IPアドレスからMACアドレスを求める)のキャッシュをSpoofing(なりすまし)し、データを傍受したり、システムを変更、停止させることができます。ネットワークのトラブルシューティングや中間者攻撃の際に用いられる。

!!!!!! 注意!!!!!!

以下の内容を試す場合は必ず自身で準備した検証環境で行って下さい！公衆で実行すると不正アクセス禁止法などに違反します！

1.準備

OSがKali Linuxの攻撃者端末(VMも可)とターゲット端末(OSは何でもよい)を用意し、Kali linuxに[dsniff]というパッケージをインストールする。

2.各アドレスを確認する

攻撃者端末でターゲットとルータのIPアドレスとMACアドレスを確認する。

3.IP転送機能を有効化する

Kali linuxの管理者権限で以下のコマンドを実行する。

```
# echo 1 > /proc/sys/net/ipv4/ip_forward  
# cat /proc/sys/net/ipv4/ip_forward
```

4.ARPキャッシュを改ざんする

管理者権限のターミナルを2つ起動して、以下のコマンドを実行する。

```
# arpspoof -i <インターフェース名> -t <ターゲット端末のIPアドレス>  
<書き換えたいたいMACアドレスに対応するIPアドレス>
```

例 右図の場合

```
# arpspoof -i eth0 -t IP-1 IP-2
```

これ↑を一方のターミナルで

```
# arpspoof -l eth0 -t IP-2 IP-1
```

さらに、これ↑をもう一方のターミナルで実行する。

これでARP Spoofingをしている状態になります。

5.終了

arpspoofコマンドはCtrl+Cでキャンセルし、

```
#echo 0 > /proc/sys/net/ipv4/ip_forward
```

以上のコマンドでIP転送機能を無効する。

	ターゲット	ルーター
IPアドレス	IP-1	IP-2

Webサイトを作りたい！

学部/回生 経営学部 1回生

名前 竹中 萌



初めまして、竹中萌です。私は経営学部で、RCCの拠点とするキャンパスとはちがう大阪の茨木キャンパスに通っていますが、コンピュータに詳しくなりたいと思いこのサークルに所属しています。ということで、まだまだ無知なので以下の内容はとても優しいです(о^_^о)是非見ていくってくださいね！

たくさん的人がよく目にする、色々なWebサイト...私もWebサイトを作つてみたいな~と思い、ある本を参考に制作してみました！（完成したものは著作権の問題により掲載できません。申し訳ありません。）

まず初めに、参考に使つた本は『1冊ですべて身につくHTML&CSSとWebデザイン入門講座』です。これを読み始める前の私は、HTMLは聞いてことがあり高校生のときに授業で少し触れたことがある程度だったのですが、CSSに関しては、聞いたことすらなく何をするために使うものなのか全く知りませんでした。ここで簡単に説明しておくと、HTMLとはHyperText Markup Languageの略で、Webページを作成するために開発された言語です。私たちがよく閲覧するWebページのほとんどにこれが使われています。次にCSSとは、Cascading Style Sheetsの略で、主にWebページの装飾をするために使用します。つまりHTMLでタイトルやコンテンツなどを字で表して、その字の色やサイズなどの飾り付けの部分をCSSで工夫してより見やすいものにするというわけです。本題に戻りますが、私が使つた参考書ではHTMLとCSSの基本的な内容が学べます。1から手順が書いてあるので、初めはとりあえずその書いてある通りに作業を進めていくだけで大丈夫なのでそんなにとても難しいということはありません。他にも、色のパターンだったり、Webデザインの良い例・悪い例、Webサイトの仕組みまで説明してくれていて、HTMLやCSS以外のことでも学ぶことができるので、Webについて知りたかったり、Webサイトを作りたいけど難しそうでかなり抵抗があるという方にオススメです☆☆☆

是非これを機に、Webサイトを試しに作つてみてください♪(‘▽’)これを目にしてくれた人がコンピュータに興味を持ってくれることを願っています！

回帰分析の基礎

学部/回生	システムアーキテクトコース 2回生
名前	フェリー



回帰分析とは、統計学の分野では、説明変数がどれほど目的変数に影響を与えること予測する統計の手法です。これを聞いたらさっぱりわかりませんね。

例えば、 $y=f(x)$ が与えられたとしましょう。説明変数とは、 x のことで、目的変数は y のことです。もっと簡単にいえば、 x は結果 y が起こるための原因です。

回帰分析の中では、様々な手法がありますが、最も代表的な例は单回帰分析と呼ばれるもので、次のように定義されます。

$$y = ax + b \quad \dots \quad ①$$

これは高校でよく出てくる1次変数ですね！すごく簡単でいいね！①の式を見ると、 a と b は定数で、それぞれ傾きと切片と呼ばれています。この二つの定数を求めるには、最小二乗法という手法がよく使われます。

しかし、最小二乗法を使うには、分析するデータの偏差平方和 (S_{xx})、 y の偏差積和 (S_{xy}) と平均が必要です。傾きと切片は次の式になります。

$$b = \frac{S_{xy}}{S_{xx}} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} = \frac{\sum (x_i y_i) - \frac{(\sum x_i)(\sum y_i)}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}}$$

$$a = \bar{y} - bx = \frac{1}{n}(\sum y_i - b \sum x_i)$$

一方的にいろいろ言われるのあまり面白くないので、例を使って説明します。

例えば、5人の生徒の数学と英語の点数、そしてそれぞれの平均等が書いている表があると考えましょう。その表のデータを上の式に入れて、回帰直線を完成しましょう。

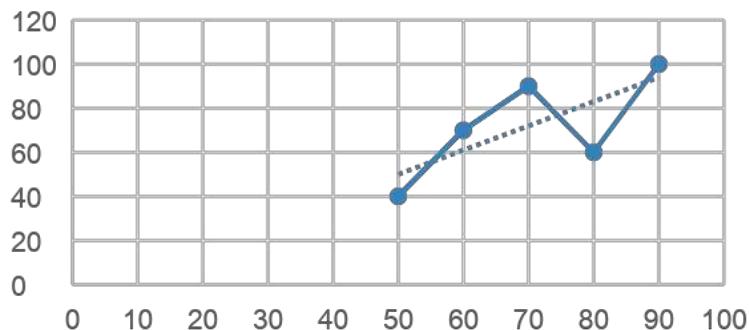
	数学(x)	英語(y)		xy	
Aさん	50	40	2500	2000	1600
Bさん	60	70	3600	4200	4900
Cさん	70	90	4900	6300	8100
Dさん	80	60	6400	4800	3600
Eさん	90	100	8100	9000	10000
合計	350	360	25500	26300	28200

$$b = \frac{S_{xy}}{S_{xx}} = \frac{\sum(x_i y_i) - \frac{(\sum x_i)(\sum y_i)}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}} = \frac{26300 - \frac{350 * 360}{5}}{25500 - \frac{350^2}{5}} = \frac{1100}{1000} = 1.1$$

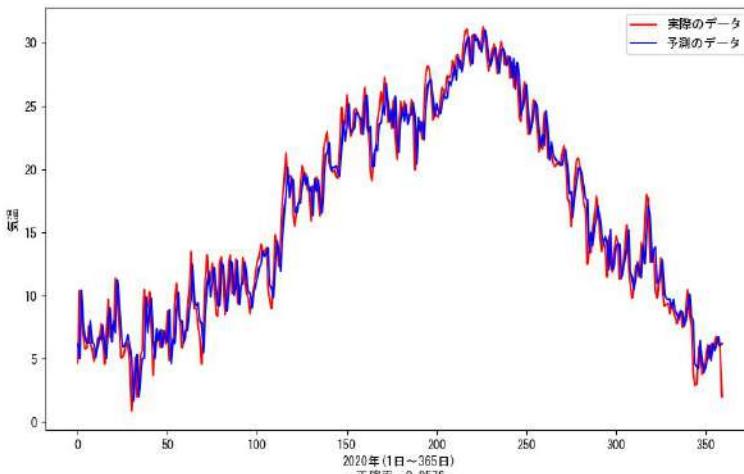
$$a = \bar{y} - b\bar{x} = \frac{1}{n} \left(\sum y_i - b \sum x_i \right) = \frac{1}{5} (360 - 1.1 * 350) = -5$$

英語(y)

よって、回帰直線は
 $y = -5 + 1.1x$ になっ
て、グラフは右のよ
うになります。



回帰直線はマーケティング、経済学、統計学やビッグデータ等によく使われています。僕は、回帰分析を使って、AIに2012年から2019年までの全国の気温データを学習させ、2020年の気温を予測し実際のデータとどれだけ一致しているかを計算してくれるプログラムを書きました。正解率は95.76%になりました。



追記：

統計に興味のある人は、「確率と統計」（1回生）、「多変量解析」（2回生）の講義を受講したほうがいいと思います

ブラウザゲームを簡単に作ろう

学部/回生	情報理工学部 知能情報 2回生
名前	husky



ゲーム

それは寿命と引き換えに一時の虚無感を忘れさせてくれるもの

今日、世の中にはさまざまなゲームがあります。その中でも、インターネットに繋がっているスマホかパソコンがあればすぐに遊べるのがブラウザゲームです。



一昔前まではブラウザゲームといえばFlashばかりでしたが、サポートが終了した後Unity WebGLを始めとする様々なツールが出てきました。今回紹介するのは、学習コストが低く WebGLの知識もいらすと2Dゲームが作れる PixiJSというものです。

実際にPixiJSを使ったものとして、これは自分がJSと TSの勉強をしながら2週間程で作ったクリックゲームです。

サイトの構築はNext.jsに任せていますが、ゲームの核となる部分はPixiJSを使っています。シーンごとに画面を作ることができ、まるで普通のゲームを作っているかのような感覚でブラウザゲームを作ることができます。

例えば、赤い円を作りたいときは、



```
let akamaru = new PIXI.Graphics()
    .beginFill(0xff0000)          //色
    .drawEllipse(0, 0, 30, 30)    //(中心のx、中心のy、幅、高さ)
    .endFill();
app.stage.addChild(akamaru);    //stageに追加
```

と直感的に描くことができます。



PixiJSをオススメする理由は他にもあります。

➤ ほとんどのブラウザで動作する

PixiJSはWebGL2.0に対応していますが、もしブラウザ側が非対応ならWebGL1.0に、それも非対応ならcanvas2Dに勝手にフォールバックしてくれます。これによってブラウザ環境を考えずにゲームを作ることができます。

➤ 他ライブラリと比べて軽量

先程にもあった通り、WebGLにも対応しているためパーティクルが多いゲームやテクスチャ数が多いゲームなどでは、高パフォーマンスで描画されます。動きが多いアクションゲームやタイミングが重要となるリズムゲーム、では大きなメリットとなります。

➤ アニメーションライブラリと互換性が高い

自分のゲームではアニメーションに非常にパフォーマンスが高いGSAPを使っていますが、実はGSAP内にPixiPluginがありPixi内のオブジェクトを自由に動かすことができます。コードも直感的に描くことができて簡単に開発することができます。



➤ オープンソースで現在もアップデートが続いている

フロントエンドのライブラリは入れ替わりが激しく流行り廃れがよく起りますが、PixiJSはコミュニティも活発でよくバグ修正などが行われているので長年使い続けることができます。

PixiJSのHPに行ってもらえると分かりますが、GoogleやMcDonald's、DisneyなどもPixiJSを使っているようなので、ゲームに限らず、自分のWebサイトにオシャレな背景をつけたり、アニメーションを多用したいときなども是非使ってみてください。

電波系男子

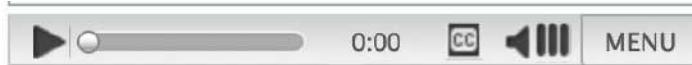
学部/回生 情報理工学部・SNコース 2回生

名前 Jin



みなさま、電車で耳がキーンとしたり、なんか音が聞こえたような聞こえなかったような気がしたことがたくさんあるかと思います。なぜなら、私たちの周りでは日々謎の存在が謎電波を通じて私たちに話しかけてくるからですが、本日はそれらの信号を解析して、込められた意味を明らかにしていきます。

まずはサイエンス誌の上を行く Wikipedia から PSK のサンプルファイルをダウンロードします。



[PSK31_sample.ogg](#) (Ogg Vorbis sound file, length 16 s, 16 kbps)

ファイル形式が .ogg というよく知らない物になっているので、Python に対応できるライブラリがあり、かつ分かりやすい wav ファイルにオンライン変換ツールで変換しましょう。変換時には細かい調整ができるので、注意が必要です。今回の例題の場合は右のようになるので、変換する時と、プログラムを書く時の値に気をつけましょう。

次はプログラミングの番ですまずはファイルを読み込み解析したいので wave のライブラリを使って音声ファイルを開きますが、私たち

```
import wave
import numpy as np
fname='rtty3s.wav' # should be specify the filename.
smp= 8000          # Sampling Rate
FQm= smp/914.0     # Mark Frequency 914Hz
FQs= smp/1086.0    # Space Frequency 1086Hz
wind= 32           # windows size Integer
waveFile = wave.open(fname, 'r')
```

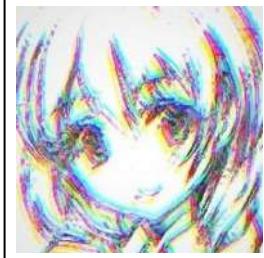
には怪電波の音にしか聞こえない雑音が、以下のような式で 0 と 1 のペアに変換されることが分かります。これらを BitWidth などの値を持って解析し、VARICODE に照らし合わせて変換すれば。。。

```
q=[];i=[]
for j in range(waveFile.getnframes()):
    buf = waveFile.readframes(1)
    q.append((buf[0]-128)*np.sin(np.pi*2.0/FQ*j))
    i.append((buf[0]-128)*np.cos(np.pi*2.0/FQ*j))
    print(int(sum(q)>0),int(sum(i)>0),sep=",")
    if j>wind:q.pop(0);i.pop(0)
```

あら不思議！いつの間にかただの

怪奇な音が "Welcome to Wikipedia, the free encyclopedia that anyone can edit" という美しい文章に！今後不意に変な音が聞こえた時はこれを頭に入れて即変換して大事なメッセージを聞き逃さないようにしましょう！

1970年にGitHubの草を生やす裏技！



学部/回生	情報理工 2回 SA
名前	kouryo4bonin

この記事は

<https://zenn.dev/sno2wman/articles/53a6b255178ad0> を一部修正した。



適当なプロジェクトを選びます、なければ作成してください。
GitHubならプロフィール用のREADMEリポジトリとかでいいでしょう。 (`<username>/<username>`)

便利のために最初に空コミットを入れていますが、不要ならしなくてもいいです。

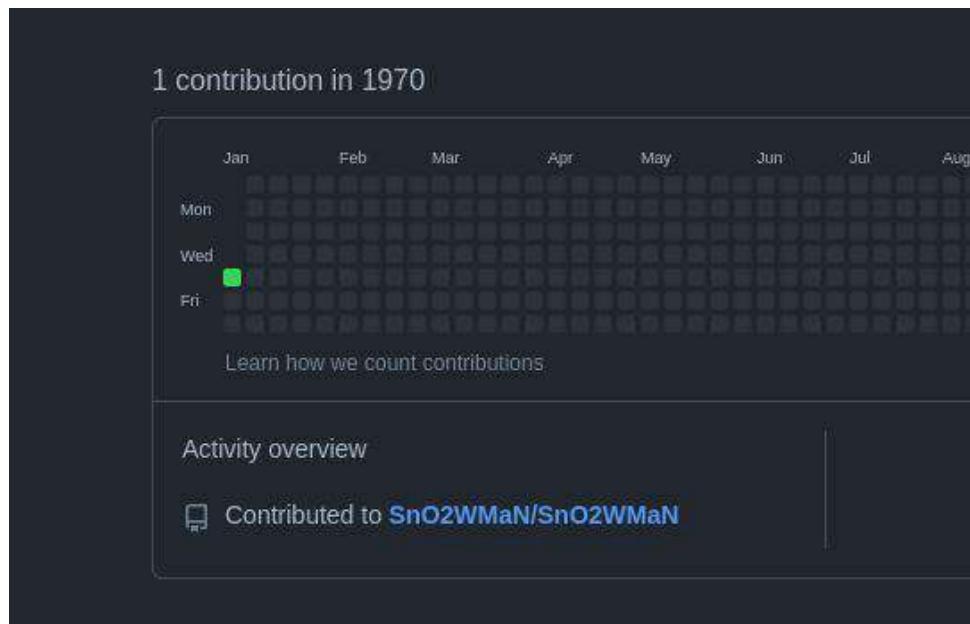
```
``shell
git init
git commit --allow-empty -m "Init"
echo '' >>
README.md
git add .
git commit -m "ワーオ!"
``
```

過去を改変します。

```
``shell
git rebase -i HEAD~1
# pick->editに変える

git commit --amend --date="$(date -u -d "@0")"
git rebase --continue
git rebase HEAD~1 --committer-date-is-author-date
``
```

あとはこれをGitHubへ…



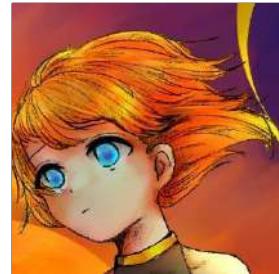
結果

1970年にGitHubに草を生やせる！

関連スレッド

- [実際の様子](<https://github.com/SnO2WMaN/SnO2WMaN/commit/5dd6de7afa3c18da7c1e3abdf7276f2f69c115d9>)
- [gitのコミット時間を変更する](<https://qiita.com/wMETAw/items/8444bba258083bd678c8>)

一週間で曲を生成する



学部/回生

情報理工学部・情報理工学科 SN二回生

名前

TZ

ネタがまじで思いつかないTZです。音楽って情報系なのかちょっと微妙なのですがRCCにはDTMプロジェクトがあるので許されると思います。

~一週間で曲を生成する~

①歌詞とメロディを同時に召喚する

歌詞は大体一音ごとに16分は早め、基本8分、ゆっくりめで4分という印象があります(テンポが120くらい)、かつ1-2小節ごとに1フレーズで休符が入る感じ。音楽は4進数?らしいので2小節の塊を繰り返して4小節とかそれをさらに繰り返すとかで思いついたメロディを再利用しまくります。メロディは厳密なものではなくて、前の音からめっちゃ上がるとかちょい下がるとかの感覚のがやりやすい(気がする)

②コードを割り当てる

いわゆる和音です。強い人はメロディに合うおしゃれコードを割り当てられるらしいのですが、コードわからんの民なのでおすすめコードとか言われているコードを持ってきます。C->F->G->Cとかいうやつです。4つのコードを1小節1つずつで割り当てる&4回繰り返してAブロック16小節決定!みたいな。それでメロディをコードに当てはめます。Cならドミソ。でも足りないのでドとミの間にレを挟む(経過音)とか、ドミドミドミの反復横跳びは上がりすぎなんじゃって時にドレドレドレにするとか(刺繍音)して調整します。合いそうになければ歌詞ごと変えます。

③編曲をする

(1)ドラムを入れる

パターンをそのままぶちこみます。Aメロはキックとスネアだけ、サビはクラップ入るしスネアの数も増える、みたいな感じでブロックごとに音数や楽器数を変えるとそれっぽいらしい。

(2)ベースを入れる

初心者大好きコードのルート弾き。(C=ドミソならド)、八分刻みをとりあえず打って適当に音消して休符にしたりオクターブで弾いたり。

(3) バッキングを入れる

ピアノとかギターとかシンセの良さげな音で②のコードを弾きなおします。4分刻みでちょっと音の長さ変えるとか、間にメロディイっぽいの挟むとかで頑張ります。

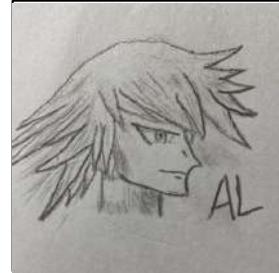
ミキシング？ そんなものは知らん

～完～

コードよりもVFXいいで！

学部/回生 物理科学科 2回生

名前 アル



VFXとは

visual effectの略名。FXはeffectって読めるからこう書くん
だぞ。ちなみに僕が考えたみたいになっているがマジだから
ね。

最近のトレンドになりつつある

ゲームやPCのクオリティが上がってきたためその演出を派手
にできるVFXが注目を浴びてきた。

VFXってどんなもの？

3DCGに近いものです。ゲームで技出したら斬撃が見えたり燃
えたりするでしょ？あれです。

え？ 実物が見たい？ 紙媒体でどうしろと？ 気になるなら
YouTubeでググれ！

どんなソフトを使えばいいの？

After Effect、UE4のカスケード、unityのShuriken particle
あたりを使えばいいです。でも大学生はAfter Effectが一番
おすすめです。

VFXの魅力

とにかく派手でかっこいい。自分がかつて想像していた厨二病
ごっこが完全再現できる。VFXができたら間違いなく3DCG、2DCG、
絵、脚本にも力を入れたくなる。ので、さらにやりたい事が増える。
VFXができると就職に有利になる。2021年現在各企業はVFXデザイ
ナーを募集を強化してますから。切られたとしても、転職先には映
画、CG、アニメーション、演劇業界なども必要として
いるから問題なし。

みんな、VFXやろーぜ！ VFX勢が少なすぎて僕は非常に困惑中だ
ぞ！

Unityの開発環境を整える

学部/回生 情報理工学部 SAコース 2回生

名前 きょうすけ



WindowsにおいてUnityの開発環境を整えるときに自分がしたこと
を書きます。以下のことを目標としました。

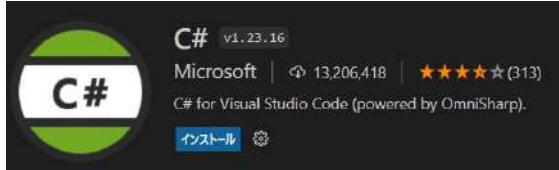
- VSCodeでのC#スクリプト編集
- Git管理

VSCodeを使えるようにする

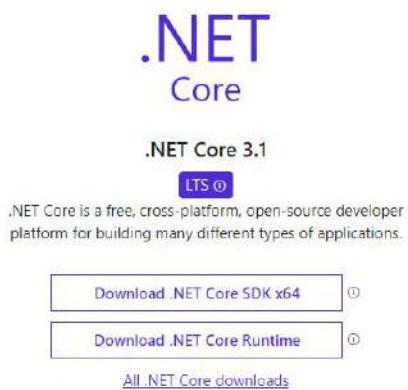
自分は普段、コードを書くときはVSCodeを使用しているので、スクリプト編集もVSCodeで行えるようにします。

まず以下のものをインストールして下さい。

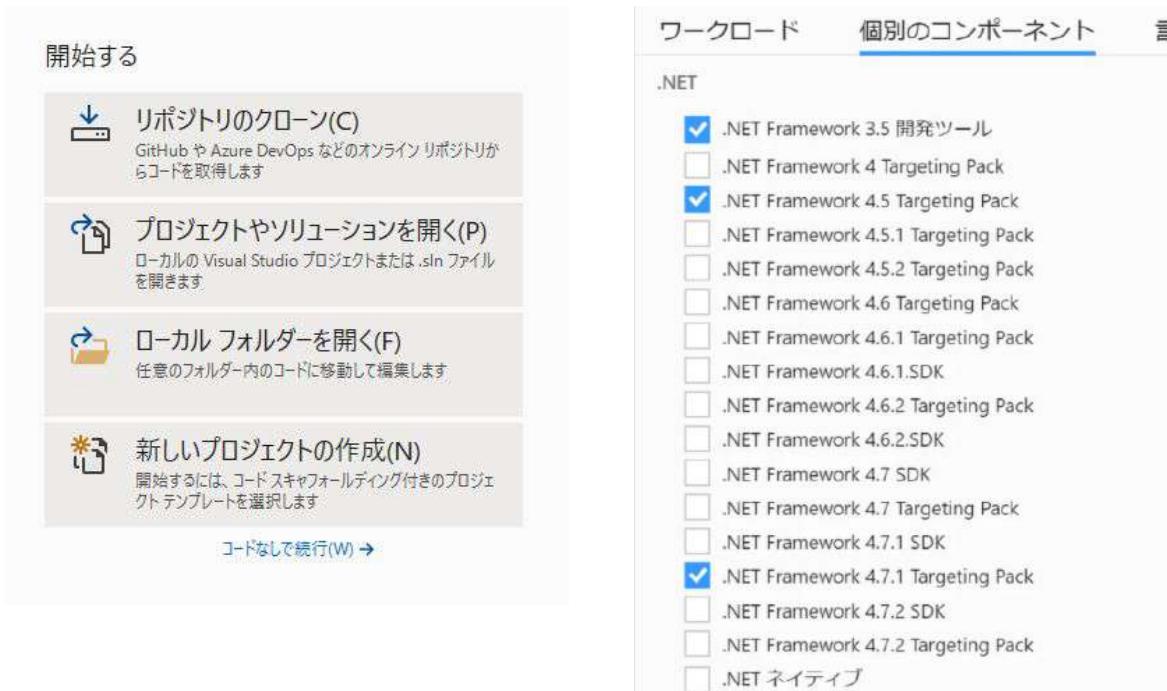
- VSCode
 - <https://code.visualstudio.com/>
- VSCodeの拡張機能
 - C#
 - Debugger for Unity



- Mono
 - <https://www.mono-project.com/download/stable/#download-win>
 - Unityのコンパイラです。
- .NET
 - <https://dotnet.microsoft.com/download/>
 - .NET Coreをインストールしてください。

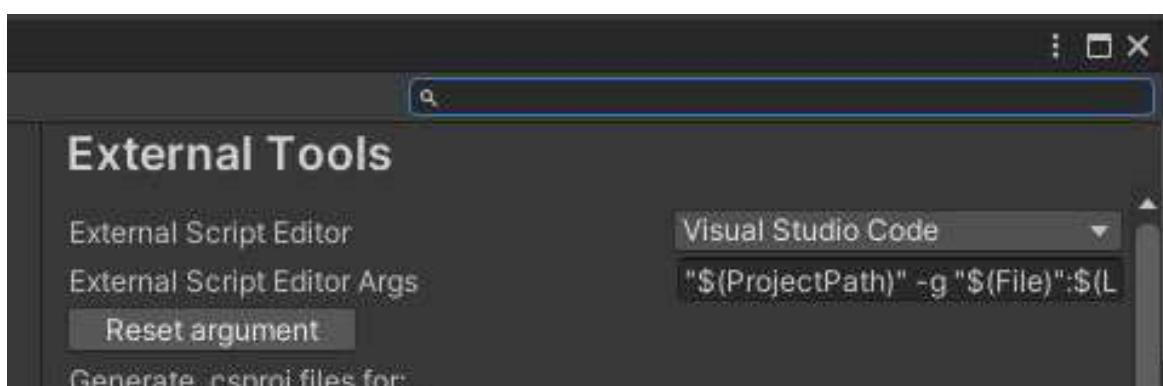


- Visual Studio
 - 後述のものを入れるために必要です。
- .NET Framework 4.5 Targeting Pack
 - Visual Studio経由で入れます。
 - Visual Studioをコードなしで続行し、ツール->ツールと機能を取得をクリック。
 - 個別のコンポーネントから.NET Framework Targeting Packをインストールします。



あとはUnityの標準エディターをCodeに設定すれば終わりです。

- UnityでEdit->Preferences->External Toolsを開き ExternalScriptEditorをVisualStudioCodeにします。
- 初めはBrowseでCode.exeを選択する必要があります。
 - %AppData%のLocal/Programs/Microsoft VS Codeに入ってると思います。



Git管理

Gitの操作環境はCUI、GUI人それぞれかと思いますが、共通して
gitignore
が必要になります。

- gitignoreは環境依存ファイルなどをバージョン管理から外すための設定ファイルです。
- Unity用の.gitignoreはGitHubの公式リポジトリを参考(kopipe)して下さい。
 - <https://github.com/github/gitignore/blob/master/Unity.gitignore>

Gitの操作環境に関してですが、自分はCLIを使い慣れてるのでコマンドぽちぽちしますが、GUIで操作したい人はGitHub for Unityを使えばいいと思います。

- <https://unity.github.com/>
- gitignoreを追加云々も全部やってくれるらしい。

gRPC超入門

学部/回生 理工学部物理科学科 2回生

名前 クマさん



みなさん、gRPCって知っていますか？



gRPCは"Remote Procedure Call"の略称で、日本語では「遠隔関数呼び出し」と呼ばれ、主にマイクロサービスのシステム間の通信で用いられているフレームワークです。gRPCを用いて作られたAPIの特徴はREST APIと違い、サーバー側とクライアント側のコードが一一で対応しており、一貫した処理を書くことが可能であり、protoファイルをもとに自動生成されるコードを持ちいるため、非常に開発効率が高いという特徴を持っています。さて、今回はGoを用いた簡単なgRPCサーバーとクライアントの実装方法の一部を紹介してみようと思います。

まずは下記のようなprotoファイルにサービスを定義します。

```
1 syntax = "proto3";
2
3 option go_package = "kaishi/proto";
4
5 package proto;
6
7 service GreetService {
8     rpc Order (GreetRequest) returns (GreetResponse) {}
9 }
10
11 message GreetRequest {
12     string name = 1;
13 }
14
15 message GreetResponse {
16     string greet = 1;
17 }
```

次に下記のコマンドでサーバーとクライアントの実装で使うファイルを生成します。

```
protoc --go_out=. --go_opt=paths=source_relative \
--go-grpc_out=. --go-grpc_opt=paths=source_relative ./greet.proto
```

生成されたファイルに実装されているインターフェースや関数、構造体を用いてGreetServiceのGreet methodを実装していきます。そして、サーバーの実装し、clientからリクエストを送ると下記のようなレスポンスが帰ってきます。

```
> grpcurl -plaintext -d '{"name":"John"}' localhost:58080 proto.GreetService.Greet
{
  "greet": "Hello, John"
}
```

なお、今回はクライアントを実装せずにgrpcurlというツールを使ってリクエストを送信しています。

また、サーバーのコードを記載していませんが、実装は簡単なのでぜひ、試してみてください。

参考文献

<https://grpc.io>

耳コピをしよう

学部/回生 理工学部2回生

名前 こま



こんにちは。

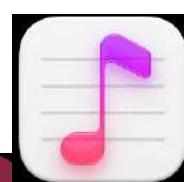
今日は耳コピをしていこうと思います。

手順としては、大まかにコード進行とリズムを打ち込み→メロディーを打ち込み→その他もろもろという感じで進めていきます。

僕みたいに「コード進行なんかよくわからん」という人には以下の便利なアイテムがあります。

便利アイテム

- ChordWiki : コード進行が乗ってるサイト。有名な曲なら大体ある。（他にUフレットや楽器.meなどのサイトもある）
- Capo 3 : 自動でコードを解析してくれるアプリ。
再生速度を変えたりすることもできる。
- Vocal Extractor : ボーカルと伴奏を分離してくれる。

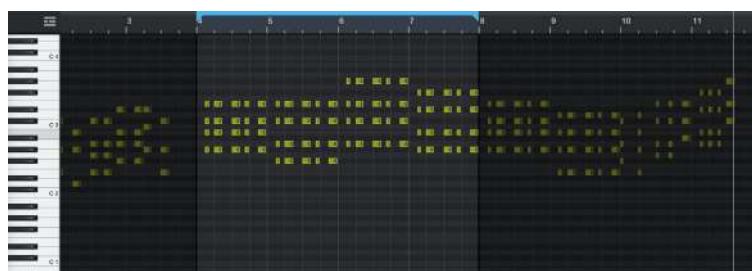


耳コピする曲：恋は渾沌の隸也

①まずドラムを適当に入れます



②エレピを打ち込む



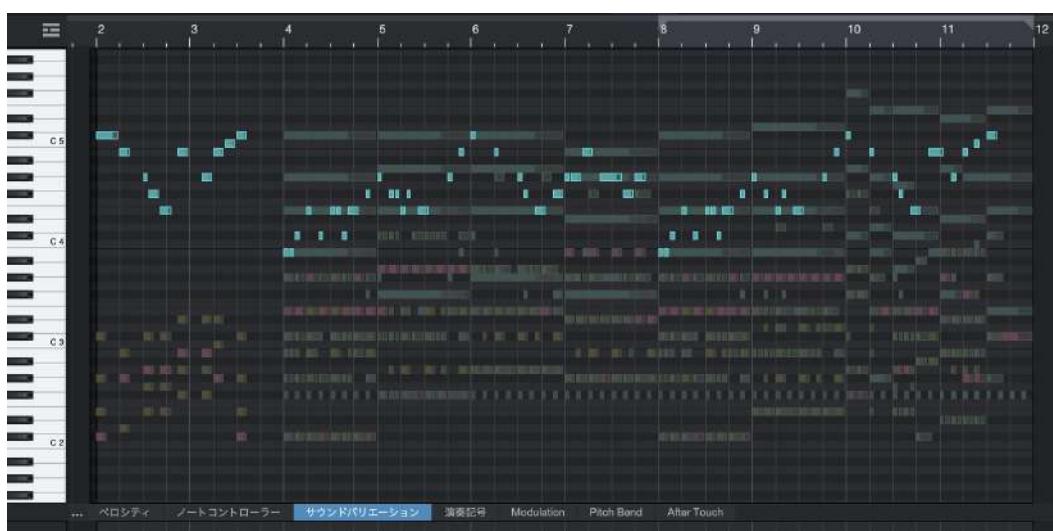
③トランペット



④ベース



⑤その他ゴチャゴチャ



完成！（イントロだけ）

<https://soundcloud.com/user-322859010/ggywqb9qumjf?si=16cad176cb9b41108fb126d5234c43d5>

Ae初心者によるPV制作

学部/回生 情報理工学部 知能情報コース 2回生

名前 つぼ



■Aeってなんぞ？

AfterEffects略してAe。Adobeのサービスの1つでアニメーション制作やモーショングラフィックス制作など主に映像制作に用いられるソフトウェアのことです。正式名称長すぎてとても毎回書いてられないでAeと略します。



■どれぐらい初心者なのか

Youtubeに投稿されているチュートリアルを趣味として何度か作ってみた程度。基礎操作すら怪しい。

■如何にそれっぽく見せるか

クオリティの高いものを作りたいといっても短時間で制作するからには限度というものがある。なのでネットに実践方法が公開されていて（つまり簡単）かつかっこよく見えるアニメーションに目をつけた。そして最終的に採用したのは以下の3つ。

- ・ヌルとカメラ機能による3Dの活用
- ・エクスプレッションを用いた自動化
- ・ロゴアニメーション

それぞれやったことを述べる。



- Camera Setting -

1. ヌルとカメラ機能による3Dの活用

AeではX軸Y軸の平面だけでなくZ軸にも画像やテキストを配置出来る。そしてカメラ機能を用いることで3次元上をぐるぐる視点移動させることが出来る。ヌルは自分自身深く理解していないのだが、基本的には既存のレイヤーと親子関係を結んでアニメーションをつけることが出来る空のレイヤーだと思ってくれたらいいです。

正面から見たり



横から見たり



これを近づけたり動かしたりしていい感じに

2. エクスプレッションを用いた自動化

ただカメラを動かすだけでは物足りないと判断し、背景の一部として文字をゆらゆらと動かそう！と決断。しかし手動で動かそうとするとフレーム単位でキーフレームを打つてしかも文字ごとに調整しないといけない。そんなことはとてもじゃないけどやってられない。ということでエクスプレッションという機能を用いて自動化しちゃおうという訳です。今回は不規則な動きが欲しかったので「wiggle」というものを用了。

3. ロゴアニメーション

ロゴのアニメーションはかっこいいと思えるものの作り方がYoutubeにたくさん公開されている。なので今回は「ダストマンTips」さんのチュートリアルを参考にして制作した。

ただそれだけではオリジナルに欠けると思い、Aの上に王冠がぴょんぴょんと跳ねるアニメーションを追加した。



Youtube動画リンク(7s)
<https://youtu.be/93n1HgzA3eQ>

おわりに

正直この依頼受けて後悔してた瞬間もあったけど全力でもがいて調べればなんやかんや耐えるってことを学んだ。3D機能でカメラぐるんぐるんさせるの楽しかった。ぐるんぐるんぐるんぐるん。

～～完成したもの～～

Youtube動画リンク(50s)
<https://youtu.be/X7AmygDe6KU>

参考文献等

-ロゴアニメーション/チュートリアル
<https://youtu.be/oK5PNGdIdpQ>

-ダストマンTips
<https://www.youtube.com/c/ダストマンTips>

-「wiggle」の解説
<https://btuber.jp/after-effects-wiggle-expression/>

VRChat環境構築入門

学部/回生 情報理工学部

名前 lufe



VRChatにはまつてもう1年になります。最近は始めより入っている時間が少なくなりましたが、やはりVRChatは私の癒しであり、メンタル回復には欠かせません。ここではVRChatの環境構築を行う手順を書こうと思います。1から環境構築をしますので誰でもできます！ここでVRChatの説明は他の記事に任せます。

※今回の環境としましてはWindowsパソコンです。

まずStreamというPCゲームのプラットフォームをダウンロードします。

そこでVRChatをダウンロードしましょう！



次にヘッドマウントディスプレイ(HMD)を買いましょう！まずは一番コスパがよいOculus Quest 2を買うことをお勧めします！

お金がある場合は他のViveやValveも
おすすめです。あとでもう1度出てくるので
覚えておいてください。



次にBoothで自分の好きなアバターを
買いましょう！自分で作ってもいいです。愛着がわいて、自分に似
合った子を見つけましょう！これであなたもVRChatの人間です！

中で楽しんでいると手の指が滑らかに動いている人が
いると思います。そこで指を動かしてみたい！って思うと思いま
す。

そこで先ほど紹介したValve indexと呼ばれるHMDがあり、その
HMDは指トラッキングがコントローラーについているので通常よりも
安く指トラッキングができます！

BOOTH



今度は足が動いている方が見つかると思います。次は足を動かしたい！と思った人も多いでしょう！最近ではいろいろとトラッカーがたくさん出始めたのでどれかってもいいと思います。私はTsundra Trackerというものを買いましたが安くてお手軽なのはやはりトラッカーとか、ユニモーションがおすすめです。



最後にVRChatをやっていると声をたくさん出す機会が増えます。その時に近所とか、親とかに声がうるさいと怒られたりすることがあると思います。その時は防音壁を購入しましょう！これで環境構築は終わりです。

もし、valveトラッカー等でトラッキングが頻繁に飛んでしまうといった状況の際は、追加でベースステーションを購入するといったことをおすすめします！

これで大体の環境構築は終了しました。最後に自分の心をVRChat内のアバターに吹き込んで全力でVRChatを楽しみましょう！そしたら自然にファントムセンスと呼ばれるVR感覚というのも芽生えてくると思います！ぜひともVRChatの環境構築をして楽しんでください！環境構築お疲れさまでした！よいVRChatライフを！

製品一覧

Steam(<https://store.steampowered.com/>)

VRchat(<https://hello.vrchat.com/>)

Oculus Quest(<https://www.oculus.com/>)

Booth(<https://booth.pm/>)

Valve Index(<https://www.valvesoftware.com/ja/index>)

HARITORA

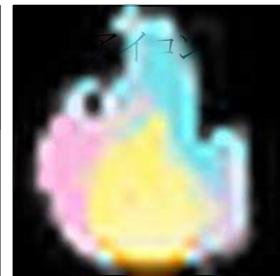
(https://ja.shiftall.net/archives/product_page/haritorax/)

Unimotion(<https://www.uni-motion.com/>)

Notionを使ってみた

学部/回生 情報理工学部

名前 lufe



最近話題に上がっていたNotionといわれるWebサービスを触ってみたという話。



・ Notionとは

Notionとは簡単に言うとWikiやタスク管理、メモといったものを一つのWebサービスで行えるようにするようなものである。Notionはいろいろなことができるが、自分でテンプレートを構築してそこからWikiやタスク管理。メモを書いていくといったものである。簡単に言えば、たくさんの部品があり、それを組み合わせることによって自分仕様のわかりやすいUIを目指す。

・ 使い始めようと思った理由

最近急速に人気が向上しているNotionというアプリケーションを教えてもらったため、一度は使ってみようと思ったから使用してみた。用途としては、タスク管理と個人技術ブログやメモが一つのアプリケーションで管理できるということが魅力的であった。そのため今までメモとしていたScrapboxからNotionへ移行しようと思った。ここでScrapboxのここがダメポイントとしてはScrapboxの独自記法がある。その為マークダウンでかけるNotionがいいと思った。

・ 使ってみての感想

率直にいうと大変だった。Notionの説明にも記載したが、自分でテンプレートを作成するというものである。自分の思うようにテンプレートを作成できず、もともとあるテンプレートをみても構成がよくわからなかった。私には早すぎたのだと感じた。一から自分でUIを設定している気分になり、これなら自分でブログ兼、メモのアプリケーションを作成した方がいいと思ってしまった。しかし、他のサービスからAPI等で情報を容易に持って来れることに感動したが、私はまだScrapboxの方が向いているということで、当分Scrapboxからは離れないと思う。

・**最後に**

学生や教職員の方はac.jpドメインのメールアドレスで登録することによって無料でパーソナルProプランを無料でしようすることができる。ぜひ、一度は使ってみることをお勧めする。一度使ってみたいとどんなものかわからない。一回は試しに使ってみること言うことが大事な気がする。自分に合ったWikiやタスク管理、メモツールになるかも。

Stripe API

学部/回生 理工学部・機械工学2回生

名前 けんしん



stripeとは

決済代行サービスです（類似サービスとしてはpaypalがあります）
paypal違うが言ころは、ユーザーがstripeのアカウントがなくても支払ができるということです。

具体的に何ができるかというと、stripeのAPIを使うことで、自身のサービスの決済処理を担ってくれます！

つまり、ユーザーにお金を請求してその代金を受け取りたい、「買い切り」「サブスク」などのプランを設定して請求したいといったことが可能になります。

また、初期費用、月額費用もかかりません。手数料は一回の決済に3.6%と安く導入することができます。

例：

一回買い切りの

必須パラメータは

請求する相手の顧

客IDと料金の2つ

だけで実装ができ

ます。

サブスクリプショ

ンも請求する相手の顧

客IDと登録したサービスプランのIDのみで実装が可能です。

こんな感じで楽に実装ができます。また、テストのライブラリも豊富にあり、130種類以上の通貨に対応しているので、グローバル展開が円滑に進めることができます。

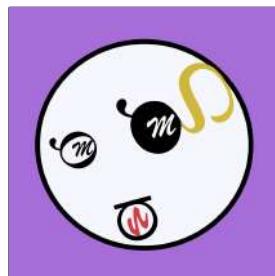
皆さんも決済機能をつける際は、ぜひStripeを！

```
def self.one_time_charge(price:, customer_id:)
  Stripe::Charge.create({
    amount: price,
    currency: "jpy", # 請求通貨は円で固定していますが変更可能
    customer: customer_id,
  })
end
```

```
def self.subscribe(customer_id:, plan_id:)
  Stripe::Subscription.create({
    customer: customer_id,
    items: [{ plan: plan_id }],
  })
end
```

TouchDesigner

学部/回生	情報理工 学部・実世界情報コース3回生
名前	momonga



夏休み最終日にTouchDesignerとかいうのをつかまされたmomongaです。

今回は色々TouchDesignerを触って便利なこと、面白いことがわかつってきたのでつらつらと紹介していきます。

まずTouchDesignerとは？

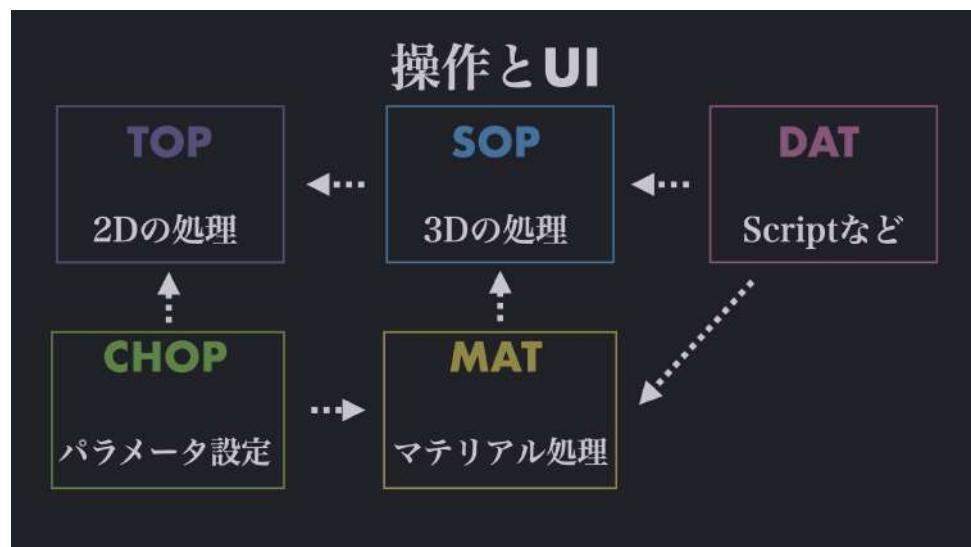
TouchDesignerってなんぞやって人が結構多いと思うので、まずはその説明。近年インタラクティブアート界隈で有名になってるソフトで最近だと主にmvとかに使用されたりします。

mvだけなら、ただの動画作成ソフトやCG作成ソフトで作れます
が、TouchDesignerの売りはどこでしょう？

僕が現時点では思うのはリアルタイムな処理に強い点です。mvに使われていると言ったので動画作成ソフトの延長と思われた方もいると思いますが、このソフトの使用目的はあくまでアートというところにあります。インタラクティブアートとはアートを見てる人の動きや反応に合わせて変化するジャンルのアートです。そのため様々なセンサーとの相性が良く作られています。

TouchDesignerの操作

TouchDesignerはoperatorと呼ばれるノードみたいなものを組み合わせていくことで作品を作り上げていきます。下にあるのはTouchDesignerのoperatorの種類です。2D・3Dの処理からScript(Python)の処理まで色々できます。（*OPはoperatorの略です）



で、作品

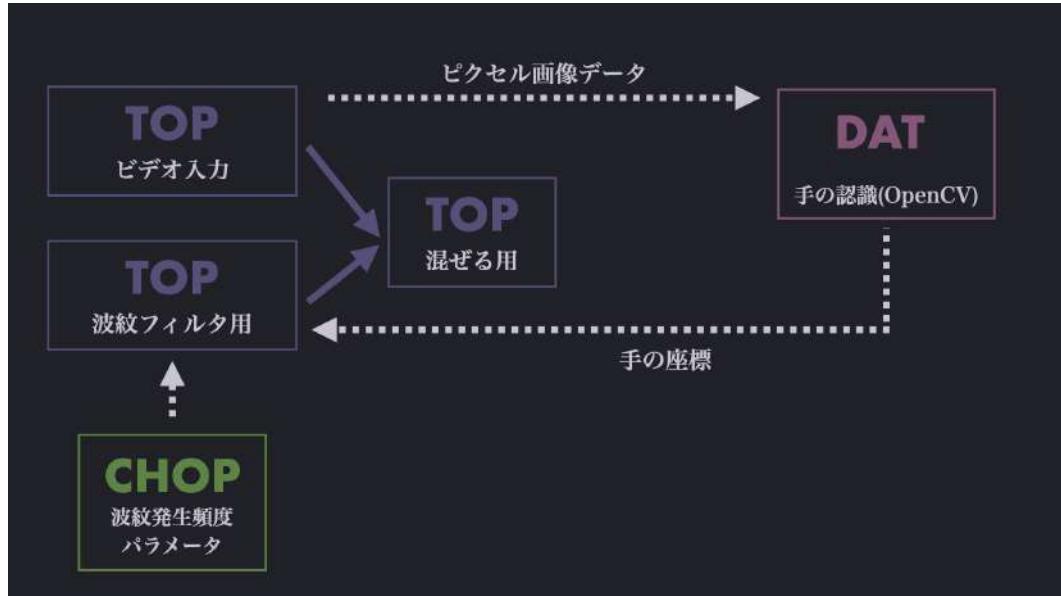
まあここまで調べたら大抵出てくることなので、おめえなに作つてん？を紹介していきます。

https://drive.google.com/file/d/13qVhg9m9fe-By_rvxjeKWcp4uN52aBX/view?usp=sharing

こんな感じで手を認識させてそこから波紋を出すような作品を作つてみました。

TouchDesignerのTOPとDATを使用して作りました。

構成



波紋のエフェクトはTouchDesignerの2Dのcircleを使用して表現しました。circleのオブジェクトはRGBAのデータを設定することができます。今回は、alphaの値を0.99あたりに設定して画像を屈折させています。circleの半径をCHOPで変化させることで波紋が広がっていく表現をしています。

DATではPythonでOpenCVを使って手の認識をさせています。ビデオ入力でとってきた画像情報をDATに投げて手を認識した時の座標を返すようにプログラムを書きました。

これをcircleオブジェクトの中心位置になるように設定することで手から波紋が出ている作品を実現しました。

手の認識はOpenCVの精度によるので反応しない時もあるのでそこが問題点です。また、波紋がスクリーン端で反射するような処理まで加えていきたいです。さらにその波紋と干渉したらよりリアルになると思います。

まだまだoperatorのパズルのように組み合わせていく作り方をうまく使いこなせていません。これからもっと頑張っていきたいです。

コレクション・フレームワーク

学部/回生 情報理工学部3回生

名前 TNB



最近富竹ボーナスしか引けないTNBです。

アイコンはある人がマーしてるやつですが著作権的にやばいっぽいので空白です。

何をかけばいいのか迷ったので研でちょうど発表するコレクションフレームワークについて書いてみます。

そもそもコレクションフレームワークってなんだよって話ですが、Javaのコレクションをあれこれするライブラリのことです。

(java.utilのパッケージ内にあるらしいです)

コレクションとは？

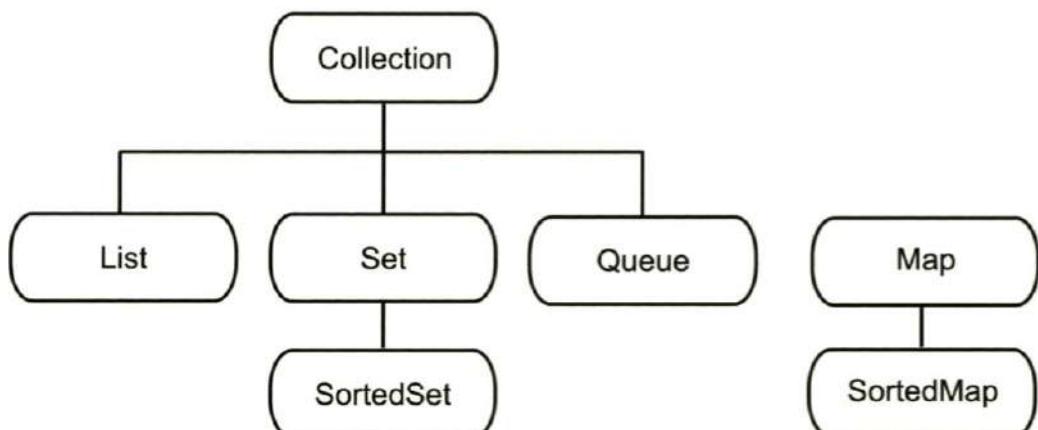
コレクションとは、配列の上位互換みたいなもので、配列との主な違いは要素数が変更可能であること、キーの値から検索が可能であることなどが挙げられます。そのコレクションをソートしたり、あれこれするのがコレクションフレームワークです。

コレクションフレームワーク

コレクションフレームワークの主なインターフェイスは

- Collection (コレクションの基本機能)
- List (整数のインデックスをつけた要素の列)
- Set (重複を許さんぞ！)
- SortedSet (昇順でソートされたSet)
- Queue (先頭から消えて、末尾から追加される)
- Map (キーと値のペア)
- SortedMap (キーの昇順のMap)

各継承関係(機能を引き継いでいること)は下図のとおり



CollectionをList,Set,Queueが継承していてさらにSetをSortedSetがMapをSortedMapが継承しています。

最後に

これ以上細かく説明するととんでもないことになってしまいそうなので最後にちょっとした小ネタを書いて終わろうと思います。今のJavaのコレクションはジェネリック(型情報を持ってること)ですが、昔はそうではなかったようで、色々な型情報のない要素を一つのコレクションに詰め込みました。

色々なものを詰め込める方が便利っぽく見えますが、コンパイルするときに型検査というものが十分にできなかったり、コレクションから要素を取り出した後に型変換を行わなければいけなかったり、取り出してもその要素が何かよくわからなくなってしまったりと、問題が多々あったようでジェネリックという概念が追加されました。



ですが、昔のJavaとの互換性を保つために、今のJavaでも型情報をつけなくていいようにできるらしいです。

今後の展望としてはJava使ってなんか作れたらいいなあ～程度に考えています。

ここまでざっと書いてきましたが、最後まで読んでいただきありがとうございました。

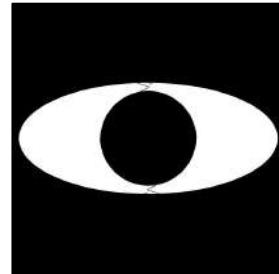
参考文献

立木秀樹、有賀妙子 (2017)
「すべての人のためのJavaプログラミング第3版」
共立出版

はんだづけの話をしよう

学部/回生 理工学部 ロボティクス学科

名前 USA



柄にもなくちゃんと為になることを書きたい。そして、多分學問的に RCCの電子工作力は僕が担っていかなくては行かないはず...(全然それに見合う先輩でいらっしゃいませんが。)せっかく大学で対面の許可が出てき始めたので、自分の為にも冬休みにはんだ付け練習会を開きたいという意思表明も込めてはんだ付けの基本的な復習を記して行きます!!(詳しくは後述するけど、購入申請もしちゃったし...)

まず、必要なものからその1、はんだごて。



何が無くともこれがなくちゃ始まらない。図は、スタンドに収納された状態のデス。持つていいのは青色の部分だけ。でないと火傷しちゃうよ!!このスタンドはその2、クリーナーとして機能するスポンジ付きです。濡らして使いましょう。

濡れ雑巾が僕のお気に入りデス。部室のスタンドはワイヤタイプのクリーナーが標準装備です。ついでに書きますがその3、**十分な電圧**も大事な要因です。熱エネルギーというのはなかなか燃費が悪いもので、、あまり考えてないとすぐにブレーカーが落ちかねません。続きまして...

その4、はんだ。まあ当然ですわな。鉛とスズを主とした合金です。



融点が低く、導体である。と。これを用いて電気の道筋を繋いで行くのがはんだ付けというわけです。極端な話、はんだごてとはんだと通電したいものがあればはんだ付けは可能なんです。

一応その5、**消化用品。**

これは最後の必須用品としてあげさせてください。火傷時の為に流水が確保できることも大事。やっぱり重要なので。さて、ココからは+@のものになってきます。が、「私、失敗しないので」的なドクターXみたいな人以外は実質必須も同様のアイテムを加えていきます。



その1 はんだ吸い取り線。付けすぎたはんだを滲み込ませることに特化した編み込まれた銅線。購入申請しました。後述する失敗例を解決するときに大活躍します。



+@その2はんだ吸収器。用途は前述の吸い取り線と同じ。前のは使えば使うほど無くなる事で使えなくなるけど、こっちは経年劣化によって使えなくなる。購入申請しました。前任のRCCはんだマスター(僕が勝手にそういうことにしている)は自前のものを部室に持ってきていた…そしてその人はそもそも失敗が少なかったのでこれらの道具は必要そうとして無かったのでおそらく部室に無くてもこまらなかつたという…しかし我々は違う!!!技術の間口を広げたくて二つとも購入申請をしました。清き一票を!!



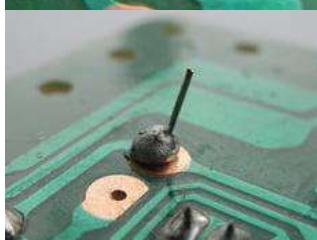
続いて…+@その3ツールクリッパー。これは一回この便利さを知ってしまうと手放せなくなる系の超便利ツール。はんだ付けをする想を考えてみてください。。。(...考えた???)右手にコテ、左手にはんだ。じゃ、新たに付け足したい銅線は誰が持つの??場合によっては光の当たり具合が悪い時どうやって基盤の角度を傾けるの??そもそも久しぶりのはんだ付け中ではなんだを持つ手が震えて危ない!!それらの問題を一手に解決します。購入申請しました。正直、一定以上の複雑さを持つはんだ付けを行う場合は実質必須ツールです。

さて、ここからはよくあるはんだの失敗例を。対処法やこうならないための注意点とともに記します。こうなっても+@アイテムがあれば幾らか修正も効く。だから、清き一票を!!



目玉はんだ

これははんだに導線がついてない。はんだ初心者あるあるの代名詞。**はんだを沿わせる前に導線は十分温に温めておくべし**コレ大事。



イモはんだ

今度は基盤に熱が足りてないパターン。**はんだを沿わせる前に基盤も十分温に温めておくべし**コレ大事。



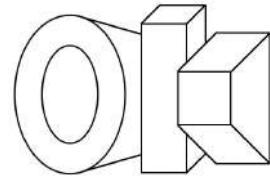
ブリッジ

はんだ多すぎのパターン。十分溶かして、+@アイテムで吸い取りましょう。。熱し過ぎるとデバイスを傷めるので注意。。

本当はフラックスの話もするべきなんだけど、区切りが良いのでここまでとします。

参考：[電子工作のコツ/はんだ付け | 村田製作所技術記事](#)

自作をはじめよう



学部/回生	情報理工学部3回生
名前	OK(okgwknt)

はじめに

今回は、自作について書きたいと思います。ここでいう自作は模倣するという感じがございますかね。自作をすることのメリットは、対象物の仕組みや機能について学ぶことができます。では、様々な自作について紹介していきます。

自作OS

OSは、ユーザに対してプログラムを実行するための環境を提供します。今年の初めあたりに、みかん本なるものが出ました。ほとんどの人が30日では終わらない分量です。今年度RCCでは、前述の本を参考にOS班が走っていますね。

自作ブラウザ

ブラウザは、Webサーバに接続するためのソフトウェアです。今年のセキュリティキャンプで、自作ブラウザを扱っているところがありましたね。資料も公開されているのでぜひやってみましょう。

自作Webサーバ

Webサーバは、HTTPで送られたリクエストに対して情報をレスポンスする役割を持ったソフトウェアです。作るためにWebの知識が必要となります。ブラウザと両方作って自作Webを作ってみましょう。

自作シェル

シェルは、UNIXなどのOSで使用されるコマンドインタプリタです。昨年度のC言語班では、自作コマンドをそれぞれ作り、統合し、自作シェルを成果物としました。

おわりに

この他にも、自作プログラミング言語、自作CPUや自作仮想マシンなどあります。自作したものは、完璧に作ることは難しく、本物には敵いません。本物は膨大なコード量です。なので実際に使うことはできないかもしれません。しかし、自作は勉強になります。普段目の行かないところにも知識が広がるでしょう。といっておきながら、筆者はこのうち2,3個しかやっていません。今後やる予定ですが... 皆さんも自作をはじめてみませんか。

なんにもおもいつかん

学部/回生 情報理工学部・実世界情報コース 3回生

名前 しんどう



なんにも思いつかないので、書きたいことを記述する。

その1

「キャラクターのデザインとかで気に入ってるやつについて語るだけ」

キャラクターのデザインは重要である。見た目とかもそうだが、設定などを凝っているものも多い。そこで、個人的に面白かったキャラクターについて、独断と偏見を多分に加えながら語りたい。

- ・「アン・イシュワルダ」(モンスターハンターワールド：アイスボーン)

とても特徴的な見た目をしている。このキャラクターは、物語のおおとりを務めるキャラクターである。そのため、同ゲームの他のキャラクターと違う仕様がある。目がプレイヤーではなく、カメラを常にのぞき込んでくるのである。正直言って怖い。一方で、このようなキャラクターは類を見ない。また、その仕組みもおそらくは単純なものと考えられる。そのため、独創的でとてもおもしろいデザインのキャラクターくなっている。また、ゲーム内で「大いなる存在」と呼ばれているが、慣れればカモである。

- ・「N-WGIX/V」(ACVD)

とにかくかっこいい。第二形態で一部発光するのがいい。過去作のキャラクターのオマージュではあるが、面影は残しつつ全く別のキャラクターを作りあげている。合体して戦闘機みたいになるのもいい。男の子の口マンを詰め込んだキャラクターである。

- ・「星の娘、エーブリエタース」(BloodBorne)

度肝を抜かれるデザインしている。ドラゴンと軟体動物と人間をミキサーで混ぜてぶちまけたかのような見た目をしている。初めて見たときは、「ナニコレ…」としかならない。しかし、よくよく見ると座っている女性に見えてくる。作品のテーマとも合っている素晴らしいキャラクターデザインである。

その2

「Ray-MMDって知ってる？」

Ray-MMDは、最近使い始めたMMD用のシェーダーエフェクト(?)である。面白かったので紹介しようと思う。このエフェクトはRuiさんが作成したものであり、無料で配布されている。このエフェクトを導入することで、MMDのデフォルトのシェーディングでは、不可能な処理を行えるようになる。そのため、Ray-MMDを導入する前と導入した後では作れる映像が全然違う。それを画像にして示す。この画像を作成するために、「名無しさんC作 ホワイトグリント」のモデルをお借りした。

図1の画像がRay-MMDを導入する前にMMDで出力した通常の画像である。MMDはトーンシェーディングを行っているので、アニメ調の影がついている。

図2は導入直後の画像である。図1と比べてだいぶ影が濃くなっていることが一目でわかる。そのため、自身の手で光源の強さなどを調整する方法がRay-MMDにはある。

前述の方法で調整したあとの画像が図3である。図1と比較してもらわると一目瞭然である。特にわかりやすい部分はメインカメラ(青いところ)である。通常、MMDで発光させることはできないのだが、Ray-MMDを導入することで、図3のように発光させることができる。それ以外にも、光源位置の詳細な設定なども行えるため表現の幅がかなり広がる(と思う)。

その性能をより詳しく知りたい場合、Ray-MMDを用いた多くの映像作品がYouTubeに投稿されているので、ぜひチェックしてもらいたい。

基本的な使用方法は以下のURLを参考にしていただきたい。

<https://czpanel.com/lecture/mmd/mme/ray-mmd-introduction/>

また、挫折していくなくても以下のURLのサイトは是非とも参照してもらいたい。

<https://ecliuth.gitbook.io/ec-ray-mmd-tutorials/articles-for-those-who-give-up-ray-mmd>

普通に作れるもののクオリティが上がるるので参考していただきたい。



図1 デフォルト



図2 導入直後



図3 調整後

その3

「Twitterはじめました」

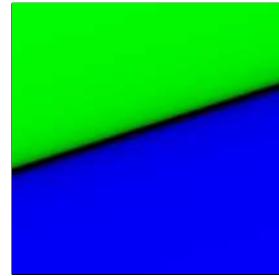
今までSNSというものを忌避してきたが、みんながすなるSNSなるものを私もはじめてみる事にした。正直な感想だが、「みんな、やさしい…(*'ω` *)」。そこには、私の思い描いていた地獄のようなSNSは存在していなかった。殺伐とした世界など存在していなかった。おそらく、そういう界隈だというのもあるのだろう。誠に感謝である。だが、「推しを推すためだけ」のアカウントなので、絶対に教えない。

絶対にだ。いいね？

UnityとJavaScript

学部/回生 情報理工 学部・SAコース 三回生

名前 たつや



初めに

HackUで使用した技術についてここに記します。

まず、Unityとはゲームエンジンの一つで、ゲームを簡単に作るための物です。Unityで作られたゲームはexe形式等に出力されますが、webGLに出力することもできます。そこでこのwebGLはその名の通り、webで動きます。ということはJavaScriptで色々いじる事は出来ないかということでいじっていました。

使ったもの

- Unity
- JavaScript
- firebase

したこと

まず、firebaseにアカウント登録した後、プロジェクトにStorageとAuthenticationを追加し、Authenticationにテキトーにメアドを登録しておきます。その後Unityでプロジェクトを作成し、Assetの中にJavaScriptのプログラムファイルを用意します。用意する場所は決められていて、Asset/Plugins/ の中に、拡張形式を.jslibで用意しておきます。そうすることでwebGLを出力する過程で一緒にコンパイルしてくれます。

ここからはプログラムコードについて書いていきます。

基本形は

```
var Plugin = { ここに変数宣言や関数の定義を書いていく };
```

```
mergeInto(LibraryManager.library, Plugin);
```

となります。変数の宣言は、

```
$変数名: {
```

辺数の内容、主にjson形式で書いていきます

```
}
```

となり、関数の定義は、

```
関数名: function(引数){
```

```
}
```

となります。

ちなみにC#からjavascriptの関数の定義は、

```
[DllImport("__Internal")]
    private static extern void 関数名();
```

で、呼び出しは通常の関数と同じです。

ちなみにC#からjavascriptの関数の定義は、

```
[DllImport("__Internal")]
    private static extern void 関数名();
```

で、呼び出しは通常の関数と同じです。

ちなみに、C#側からアクセスされない関数や変数はコンパイル時に削除されます。それでも必要な場合は

```
autoAddDeps(Plugin, '変数名 or 関数名');
```

と書く必要があります。

ここからはいよいよjavascriptからfirebaseへの通信に関するプログラムについて解説していきます。

まず使う変数について書いておきます。

```
conf: {
```

```
    apiKey: "...",
    authDomain: "...",
    databaseURL: "https://...",
    projectId: "...",
    storageBucket: "...",
    messagingSenderId: "...",
    appId: "...",
    measurementId: "..."
```

```
},
```

```
email: "HackU@gmail.com",
```

```
pass: "abcdefg",
```

```
db: "",
```

```
st: "",
```

```
once: "false" //過去にアクセスしたかどうかのフラグ
```

```
}
```

firebaseにアクセスするにはまずログイン処理を行わなければなりません。ログイン処理には

```
firebase.auth().signInWithEmailAndPassword(params.email,
params.pass); //メールアドレスとパスワードでログインする
でアクセスします。
```

そして、必要な初期化処理として

```
Initialize: function() {  
  if(params.once){  
    firebase.initializeApp(params.conf); //初期化  
    params.db = firebase.firestore(); //firestoreのアクセス口を  
    取得  
    params.st = firebase.storage().ref(); //storageのアクセス口  
    を取得  
    params.once = false; //フラグをfalseにする  
  }  
}
```

を行います。そしたらここからがデータの取得になります。

```
Get: function(col、 doc){  
  params.db.collection(PtoS(col)).doc(PtoS(doc)).get();//データ  
  を取得  
}  
Set: function(){  
  params.db.collection("pics").doc("name").set(add); //データを  
  送信  
}
```

これによってデータの送信が行えます。

このような感じでUntiyとデータベースの間でのやり取りが出来ま
す。

苦手を学習するタイピングソフト



学部/回生	情報理工学部・実世界コース 3回生
名前	FKD (@fkd_hideaway)

私などの初学者はプログラミング以前にタイピング速度を向上して、全体の作業スピードを向上させるほうが結果的に効率が良さそうです。

しかし、タイピングソフトには本質的に大きな問題をはらんでいます。それはプレイヤーが出題される文章を覚えてしまうというものです。これは収録単語数の少なめな寿司打やe-typingでしばしば発生します。

またタイピングはプレイしていると苦手な運指と得意な運指とに分かれていきます。ある程度苦手な運指は共通していますが、キー配列を変えているタイパー や個々人の苦手は存在するため、得意不得意はそれぞれと言えるでしょう。

例) 苦手な運指：一喜一憂 (ikkiitiyuu)
得意な運指：社会見学 (syakaikengaku)

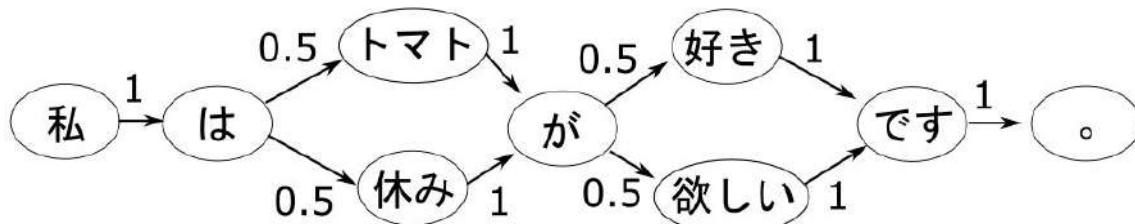
そこで出題側が文章を生成することによって覚えてしまう問題を解決し、さらにタイパーが比較的苦手な文章を生成することで、効率の良いタイピング練習を目的とする文章生成ソフトを作りました。思いつきです。

- 目次
1. N階マルコフ連鎖で文章生成
 2. タイプミスをN階マルコフ連鎖でモデル化
 3. 苦手度を順位付け
 - (4. 出題)

1. N階マルコフ連鎖で文章生成

N階マルコフ連鎖とは、次の状態が現在を含めた過去N個の状態履歴に依存して決まる確率過程のことです。（Wikipediaより）

マルコフ性とは未来の挙動が現在の値だけで決定することをいいます
が、マルコフ連鎖は過去も含めて現在の状態と定義しています。



画像出典元：<https://omedstu.jimdofree.com/>

これを利用して文章を生成します。Tweet generatorもこれを使って文章生成していますね。他にもRNNやLSTMを使う方法などがありますが、順位付けの際にもN階マルコフ連鎖を利用するため、ついでに文章生成もこの手法を使っています。

ノードは単語、エッジを遷移確率とするモデルを構築して文章を生成します。

適当にITについて説明している記事を拝借して、5つ文章生成した結果が以下となります。このように特定の文章を元にすることでタイピングに出る文章のテーマを指定することができます。

経済産業省の通商白書によると、eコマースの市場規模は99.8兆円、全体の10%を占めており最も大きいです。
深刻な人手不足に悩まされている分野・業界を10選紹介します。

しかし、その効果が出ているとはまだ言い切れない状況です。

また、間に挟む階層が多いほど、下の階層に属する利益が減っていきます。

X-Tech（クロステック）には様々な種類があります。

使用させていただいた記事：

<https://unistyleinc.com/techniques/993>

2. タイプミスをN階マルコフ連鎖でモデル化

ノードをカタカナ1文字、エッジをタイプミス数としたN階マルコフモデルを生成します。タイプミス数は実際にタイピングソフトをプレイしてもらってデータを収集します。生成した文章をこのモデルに通すと、プレイヤーがその文章を通してどれだけミスをするか暫定的に順位付けすることができます。これを仮にタイプミス推定モデルと呼びます。

3. 苦手度を順位付け

生成された文章をカタカナに変換した後、タイプミス推定モデルに通してミスする度合いを推定します。文章は多めに生成しておいて、ここで上位になった文章を実際に出題します。

例えば、タイプミス推定モデル($N=2$)内の `{("イ", "ッ") : ["キ", ...]}` という辞書型配列のvalueに出現することの多いカタカナは、ミスの度合いが高いと判定します。

今後の展望

N階マルコフなので、文章生成モデルとタイプミス推定モデルそれぞれについてNを決めてあげる必要があります。とりあえずそれぞれ $N=3$ 、 $N=2$ くらいで定義してあげるとそれっぽく動作します。

出題側に作問する時間が必要なので、ゲーム開始前にモデルを生成・保持しておく工夫が必要そうです。

問題点はタイプミス推定モデルを生成できるだけの、ある程度の数タイピングゲームをこなす必要がある点です。

思いつきでの実装なのでよりよい方法があると思います。深層学習を使ったらもっと良い成果が出そうですね。こうしたら良さそうみたいなものがあればぜひ聞きたいです。

参考文献: <https://qiita.com/k-jimon/items/f02fae75e853a9c02127>

	Scrapbox ノススメ	
学部/回生	情理SA 3回生	
名前	やぎちゃん (@ygkn35034)	

RCC では 2021年秋学期から、引き継ぎなどの会内での情報共有や、プロジェクトや勉強会といった活動の成果の管理に Scrapbox を用いることになりました。本稿では、RCC で Scrapbox を使用することになった経緯と、Scrapbox の特徴や使い方について述べます。他サークルへの Scrapbox の導入や、RCC 会員の使用の参考になれば幸いです。

RCC での情報共有の課題

RCC では、以前より会内での情報共有では Wiki や Google ドキュメントを使用して、一部のプロジェクトや有志の勉強などで Scrapbox を使用していました。もともと Wiki や Google ドライブが RCC 公式で運用されており、そこにより手軽な情報共有手段として Scrapbox を一部の会員が有志で使い始めたようです。しかし、これらを同時に使うにつれ、以下のような問題が発生していました。

- 様々な媒体に情報が分散してしまっていた
- 普段の活動で利用することが少なくなった Wiki の情報が更新されず、古い情報が残ってしまっていた
- 有志での運用であったため、プロジェクト活動の班ごとに Scrapbox のプロジェクトが建てられており、プロジェクト活動の班をまたいだ情報共有ができていなかった

これらの問題を解決するために、RCC 公式で Scrapbox を使用する案が出ました。RCC では、会内サービスなど非公開情報用と、活動の成果などの公開情報用の2つの Scrapbox プロジェクトを RCC 公式で管理することになりました。

執筆時現在、公開 Scrapbox プロジェクト（図1）[1]は運用開始からまだ間がなく、プロジェクト活動を行っていないため、ページ数がわずか 21 ページしかありません。非公開 Scrapbox プロジェクトには会内サービスやコロナ禍においての部室利用についての説明が記載されています。



図1：執筆時のRCC の公開 Scrapbox プロジェクト

Scrapbox とは

Scrapbox は「単純で強力なWiki」（公式ヘルプ「Scrapboxチュートリアルスライド」[2]より）です。Scrapbox では、チームなどが持つプロジェクトの中に 1 つのトピックごとにページを作り、ページ間をリンクさせていくことで情報を整理します。単純で手軽な編集機能と、強力なリンク機能が特徴です。慶應義塾大学湘南藤沢キャンパスの増井俊之研究室で開発、使用されている Gazz という Wiki が Scrapbox の前身です。立命館内では、以下のような Scrapbox の活用事例があります。

- 情報理工学部セキュリティ・ネットワークコースの講義「プログラミング言語」の受講生向けに授業関連情報を提供する「立命館大学情報理工学部セキュリティ・ネットワークコース プログラミング言語サポートページ」[3]
- 教員向けの資料である「立命館大学オンライン授業リソース」[4]
- 立命館大学情報理工学部プロジェクト団体 RiG++ のサークル内で情報共有するためのプロジェクト（非公開）

Scrapbox の基本的な使い方

先述したように、Scrapbox はプロジェクト内のページ間をリンクさせることによって情報を整理します。それぞれのページを記述する主な方法は、アウトラインエディタのようなスペースやタブを用いた箇条書きのインデントと、[] で囲むとその中身に応じてリンク、画像、アイコンなどを挿入できるブラケティングです。それぞれのページは、フォルダのような階層構造で整理するのではなく、すべてフラットな関係で配置します。他にも強調、コード、ハッシュタグ記法など様々な機能がありますが、以上だけでも Scrapbox は活用できます。詳しくはヘルプの「Scrapboxの使い方」[5]をご覧ください。特にチュートリアル[6]は Scrapbox の雰囲気の把握に最適で、一度試してみることをおすすめします。

Scrapbox の設計思想

Scrapbox は、Word や Google ドキュメントなどのような文書作成ソフトと比べて装飾に関する機能が少なかつたり、Markdown と比べて見出しなどの文章を構造化する機能が少なかつたりと、物足りない印象を受けるかもしれません。また、Markdown などの一般的に使われている記法ではなく独自の記法を用いていることからとっつきにくい印象を受けるかもしれません。しかし、これらは機能を少なく、記法をシンプルにすることで思考をスムーズに記述、整理できるようにする工夫なのです。

また、ページを階層構造ではなく、リンクを用いて整理することも他の情報共有ツールではありません見られない特徴です。これによって新しく人や情報が増えても情報を追加、編集、検索しやすくなります。

このように Scrapbox は一定の設計思想を持っており、それらを知ることは Scrapbox を使いこなすことに繋がります。

Scrapboxをグループでどう使っていくか[7]、階層整理型Wikiはスケールしない[8]、ScrapboxとMarkdownの思想的な違い[9]などのページもご覧ください。

Scrapbox のカスタマイズ

Scrapbox はとてもハッカブルなソフトウェアでもあります。Scrapbox には外部プログラムと連携できる API[10] や、UserScript[11]、UserCSS[12] と呼ばれる、JavaScript や CSS で Scrapbox を拡張する機能があります。どちらも上級者向け機能であり、Scrapbox の内部仕様の変更で急に動かなくなる可能性がありますが、ハッカー心をくすぐる面白い機能です。RCC の Scrapbox プロジェクトでは UserCSS を用いたカスタマイズをしており、RCC らしい印象を与える見た目にカスタマイズされています（先述図1、図2）。

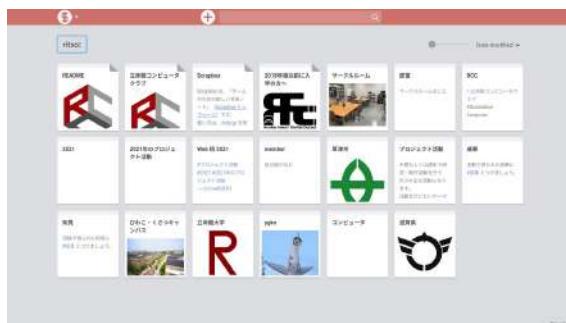


図2：カスタマイズ前の RCC の公開 Scrapbox プロジェクト
(UserCSS を消した状態、Red テーマ)

おわりに

本稿では、Scrapbox を RCC で使用する経緯と Scrapbox の概要について述べました。Scrapbox の魅力が少しでも伝わったのではないかでしょうか。Scrapbox はその自由さとハッカブルさといった特徴が RCC とマッチすると考えています。Scrapbox で RCC の情報共有が加速し、活動がさらに活発になることを願って、筆をおくこととします。

- [1]: <https://scrapbox.io/ritscc/>
 - [2]: <https://scrapbox.io/help-jp/Scrapboxチュートリアルスライド>
 - [3]: <https://scrapbox.io/Ritsumei-ISE-SN/>
 - [4]: <https://scrapbox.io/rits-ict/>
 - [5]: <https://scrapbox.io/help-jp/Scrapboxの使い方>
 - [6]: <https://scrapbox.io/help-jp/Scrapboxチュートリアルスライド>
 - [7]: <https://scrapbox.io/shokai/Scrapboxをグループでどう使っていくか？>
 - [8]: <https://scrapbox.io/shokai/階層整理型WiKiはスケールしない>
 - [9]: <https://scrapbox.io/rakusai/ScrapboxとMarkdownの思想的な違い>
 - [10]: <https://scrapbox.io/help-jp/API>
 - [11]: <https://scrapbox.io/help-jp/UserScript>
 - [12]: <https://scrapbox.io/help-jp/UserCSS>

3DS改造してみた

学部/回生	情報理工学部 SAコース 3回生
名前	ゆーみ



！注意！

この記事はゲーム機本体の改造行為や、犯罪行為を助長するものではありません。ゲーム機の改造行為自体は違法ではありません。しかし改造したゲーム機を用いて、オンラインゲームでチート行為を行うことや、ソフトのデータを違法にダウンロードすることなどは**犯罪**です。絶対にやめましょう。

あくまで個人利用の範囲で口論を感じて楽しめましょう！

3DSさん！！好きです！！！！ずっとずっと改造したいって思ってまた！！！

っていうノリで改造しました。用意したものはこちらの2点。



Nintendo 3DS (初期型)

「いつか改造するんだ...！」って思って中古で買つといた
(動作確認済み, キズ汚れあり, 色指定なし)

5

SDカード 32GB

SEIYUで売ってた800円くらいのやつ。

おうちにあるならそれの中身きれいにして使ってもいい。

64GBだとダメらしい(なんでなんだろ?)FAT32っていう

フォーマットじゃないとダメらしいけど、だいたいの

SDカードはこの規格を満たしてるのでもんまり気にしなくてOK。

(詳しくは「SDカード FAT32」で検索)



参考にしたのはこちらのサイトです。写真入りでとてもわかりやすかったです。

kaiの気まぐれblog

【3DS改造】「8月最新ver11.15.0対応！」3DS & New3DSにCFWを導入する方法

【Seedminer】【BannerBomb3 + Fredtool】

<https://k-a-i.blog/games/3dsbannerbomb3/>

※2021年10月現在の3DSの最新バージョンは2021年7月に配信された11.15.0なので、こちらのサイトはこのバージョンでの改造方法を解説したものになります。

(ちなみに3DSは2020年9月に生産が終了しているので、今後のバージョンアップデートはそんなにたくさん無いんじゃないかなーと思います)

基本的にはサイトにしたがってファイルをダウンロードして、SDカードにファイルをまちがえないように配置していくだけです。

そもそも3DSの改造ってなにするの

今回3DSに導入したのは、CFWとそれに付随するいくつかのソフトです。CFWというのは、私みたいに「3DSを改造したい!!!!」って思った人が**1番最初にいれる必要のあるソフト**です。CFWの導入の仕方にはいくつか方法があるようですが、3DSのバージョンアップデートで**対策されて使えなくなった手法もたくさん存在**します。サイトに記載されているのは「Fredtool」を使用するものです。こちらの手法自体は古くからあるそうですが、**最新のアップデート(11.15.0)でも対策されていないのでまだ使える！！**っていうすごいやつです。

多分Fredさんがつくった。Fredさんすごい。

たのしかったこと

- フレンドコードをつかってDSiWareの脆弱性をつくところ

DSiWareとは、Nintendo3DSの一つ前の世代であるDSiでダウンロードしたソフトを3DSでも遊べるようにするためのものですが、データ管理画面での不正な操作を防ぐための仕組みとして暗号化キーが使われています。

早い話が**暗号化キーを解読できれば不正な操作ができる**わけです。当たり前ですが暗号化キーは普通、解読できないようになっています。

ですが、この暗号化キーの前半部分に**フレンドコードが使われている**ということが分かっているので、**解読できてしまうらしいです** (DSiWareの脆弱性)

SeedminerというWebツールに自分のフレンドコードを教えてあげると**暗号化キーを計算してくれます。**



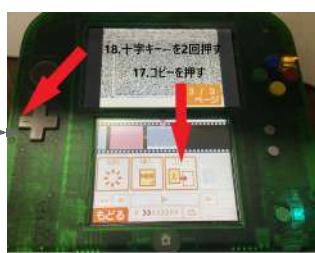
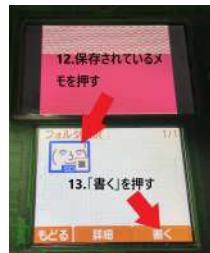
←Seedminerくん
とフレンドコード
でフレンド登録し
たところ



←クラッシュさせ
たところ
(参考サイトより
引用。写真撮って
なかった)

- うごメモからb9sToolが立ち上がったところ

DSiWareの脆弱性をついた後に、必要なソフトをパソコンでSDカードに入れてからちょっと設定をいじってDSソフト用のインターネット設定を開くと、**急にうごくメモ帳（略してうごメモ）が起動します**。通常と違う挙動なので、この時点でちょっと感動したんですが、うごメモの中にはメモのページをコピー＆ペーストした瞬間に画面がうごメモからb9sToolに切り替わったところで「私が！やりたかったのは！！こういうこと！！」ってなりました。全然仕組みわかんない。超たのしい。



参考サイトより引用

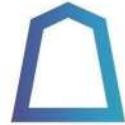


実際の画面

今後はこの3DSをさらに魔改造してカスタマイズしていく予定です！

JSF*ckに学ぶJavaScriptの変態仕様

学部/回生	情報理工学部SNコース / 3回生
名前	うんちゃん

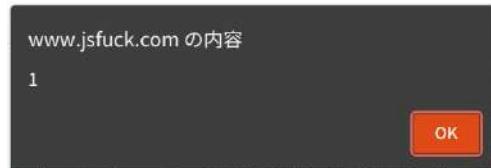


こんにちは。うんちやんです。突然ですが皆さん…

信じてください。以下のテキストは、アスキーアートでも、Brainf*ckでも、文字化けしている訳ではありません。

これは、JavaScript プログラム です。

いや、ほんとうに。試しにChrome dev コンソールで実行させてみる（※1）と下図のように出力されます。つまり、このテキストは「`alert(1)`」を実行する JavaScript（以下、JS）プログラムという訳です。もっと正確に言うと、これは、JS と等価のまま使用文字列を一部に制限した **JSF*ck** プログラムです。



■ JSF*ck とは ■

JSF*ck は、任意の JS プログラムを [] () + ! の6種の文字のみを用いて表現する難解プログラミング言語です。冒頭のテキストに Brainf*ck みが感じられたかもしれません、JSF*ck のネーミングは Brainf*ck を参考につけられたそうです。ただ、コンパイラやインタプリタを必要とする Brainf*ck とは異なり、JSF*ck は JS と等価であるためブラウザや Node.js 上で動作させることができます。

(※1) ブラウザでの場合、Content-Security-Policy で 'unsafe-eval' が指定されたWebサイトで実行する必要があります。(例: isfuck.com)

■ JS から JSF*ck への変換 ■

さて、ここからが本題です。どうすれば `[]() + !` のみで任意のJS プログラムを表現できるかを見ていきます。方針は次の通りです。

1. 空配列を生成
2. 加算演算子+ や 否定演算子! で暗黙の型変換や計算を行い値を表現
3. 値を文字列に変換し、ソースコードを表現
4. eval("コード") して実行

キーワードは**暗黙の型変換**です。例えば、JS では String 型から Number 型へは `Number("1")` として明示的に変換できますが、`+ "1"` としても同等に評価されます。このテクニックを駆使すると様々な値を以下のように表現できます。

型	表現	評価結果	説明
Boolean 型	<code>![]</code>	false	<code>Boolean([]) → true</code> に!で否定演算
Number 型	<code>+[]</code>	0	<code>Number([]) → 0</code>
	<code>+!![]</code>	1	<code>+true → 1</code>
	<code>!![]+!![]</code>	2	<code>true+true → 2</code>
String 型	<code>[+!![]]+[+[]]</code>	"10"	<code>[1]+[0] → "10"</code>
	<code>![]+[]</code>	"false"	<code>Boolean → String</code>
	<code>["a"]+["b"]</code>	"ab"	
undefined	<code>[] [[]]</code>	undefined	未定義のプロパティ

また、オブジェクトのプロパティは `obj["propName"]` でアクセスできるので、例えば、`[].filter` は `[]["filter"]` と表現できます。

次に、任意の文字列生成について考えてみます。例えば、"a" は "false" の2文字目なので `"false"[1]`、つまり `(![]+[])[+!![]]` と表現できます。JS*ck では、このような方法で任意のアルファベットを表現できることが示されています。

以上で、様々な「値」を文字列として表現できるようになったので、最後に、文字列を実行形式に変換してみます。`eval("コード")` と同等な、関数の `constructor` を上書きして即実行させる方法で実現します。

`[]["filter"]["constructor"]("コード")()`

■ おわりに ■

JSの暗黙の型変換というアレな仕様を駆使して変換を試みた訳ですね。ちなみに JS→JSF*ck の自動変換は jsfuck.com でできますので是非。

ページフォルトについて

学部/回生 情理SN3回生

名前 堀田隆成



こんにちは、書くことがないので今日は研究室のB3課題でやった内容についてでも書こうと思います。

そもそも課題としてシステムコールを作成するというものだったので、自分はシステムコールでプロセスのページフォルト数を取得するというのを作りました。

そもそもページフォルトについて詳しくない人も多いと思うので軽く解説します。

プロセスを実行するには必要になるページがメモリ上に配置される必要がある→メモリ上にない時に**ページフォルト**と呼ばれる割り込みが発生する。

それらのうち必要なページを2次記憶からメモリに配置するのをページインといい、また空きページがないときに空きを作るためにメモリから2次記憶に転送されることをページアウトという。

詳しくは情理であれば2回生から履修できるOSの授業でなっています。

そんなページフォルトなのですが実際にページフォルトが発生するところを見るために、事前にメモリ領域を確保するため大きな配列を要求するプログラムをバックグラウンドで複数実行させその上でページフォルトが発生するかを確認しました。(ちなみにここは配列だとスタックに確保されるからグローバル変数とかmalloc使ったほうがよかつたらしい。)結果としては無事にページフォルトが発生するところが見れました。このことについて研究室でプレゼンすると、もっと簡単にページフォルトが発生するプログラムを教えていただいたのでそちらも紹介しようと思います。(この発表やってて教授に泣かされたらしいね)

- malloc関数を使う

malloc関数は中身を初期化しなければ実際にはメモリ領域を確保しないという性質があるためそれを利用し、後から確保したメモリ領域にアクセスすることでページフォルトを発生させるという方法です。

この方法を聞いた時目から鱗が落ちる気持ちでした。

(注意実際にテスト書いてないから勘違いしてるところあるかも。違ってたらごめん)

ソフトウェアテストを始めよう

学部/回生	情報理工学部・実世界情報コース3回生
名前	Averak (@averak_jp)



はじめまして、3回生の健太朗です。元々ソフトウェア設計とテスト技法が好きだったので、最近になってスピリチュアルエンジニアリングを始めたところ、JPHACKSの関東ブロックで優勝できました。

1. テストの魔力

テスト（検証）とは、ソフトウェアが正しく動作しているかを検証し、プロダクション環境が故障するリスクを軽減するテクニックの一つです。洗練されたテストは正しく動作することを保証するだけでなく、ソフトウェアに高い保守性を与えます（テストコードとプロダクトコードが密結合して保守性が下がるアンチパターンもありますが、）。

とまあ、よくあるコピペのような魅力語りをしたわけですが、私は去年からテストを書き始めて、今ではパソコンでも統合テストを書くほどテストを信頼しています。ソフトウェアにバグはつきものなのだから、可能な限り早期発見できるよう明日から実践してみませんか？

2. テストの種類

テストは粒度に応じて分類されています（文化によって分類方法が変わることもあります）。粒度の大きなものはコストがかかるので、粒度の小さなものから順に検証していくのが一般的な戦略です。

1. 単体テスト
 - メソッドやクラスなどのコンポーネント単位のテスト
 - ホワイトボックステスト（内部仕様のテスト）
1. 統合テスト
 - 複数コンポーネントを組み合わせた機能単位のテスト
 - APIごとのテストなど
 - ブラックボックステスト（外部仕様のテスト）
1. E2Eテスト
 - システム全体が正しく動作することを確認するテスト
 - 自動化 or 手作業で実施

3. テストを始めよう

前述したものから、今回は最もシンプルな単体テストを扱います。ここではPythonで実装したメソッドに対し、任意の入力に対して正しい出力が得られるかを検証していきましょう。xUnit（単体テストフレームワークの総称）にはPython標準のunittestを採用します。

※ テストコード自体の品質を評価する指標に“カバレッジ”がありますが、この紹介はまたの機会に。

例として、2値の最大公約数を求めるプログラムを作成しましょう。
int型の引数a1、a2を取り、int型を返す関数gcpを実装します。

1. まずは実装してみる

```
def gcd(a1: int, a2: int) -> int:  
    return math.gcd(a1, a2)
```

1. テストを書く

```
assertEqual(8, gcd(24, 32))  
assertEqual(5, gcd(30, 5))  
assertEqual(1, gcd(72, 35))
```

これで完成です。`assertEqual`は2つの引数の値が一致するかを検証し、一致しない場合にはその詳細を出力する役割を持ちます。テストケースの設計方法にはコツがありますが、とりあえず不安がなくなるまでテストするのが良いでしょう。ただし、愚直にテストを増やすとコストが増すので慣れが必要かも知れません。（カバレッジや境界値分析、同値分割などがポイントです）

4. テスト駆動開発

「設計がテストを駆動する」なんて格言もありますが、テストを書いてみて設計の違和感に気づくこともあります。このような設計ミスの早期発見やテストを書くストレスを軽減させるテクニックの一つに、テスト駆動開発（TDD）というものがあります。前述した例では実装→テストの順にコーディングしていましたが、TDDはこの順番が逆転します。先にテストを書いて、passするまで実装を続けるのが特徴です。

TDDの目標は「動作する綺麗なコード」です。愚直にテストを書きまくれば機能性は保証されますが、テストの実施にかかるコストの増大や保守性の低下がネックになります。テストを先に書くことに違和感を感じるかも知れませんが、慣れればこのメリットに気づけるはずです。TDDは実装フェーズからストレスを排除し、動作する綺麗なコードを実現します。テストの重要性に気づいた方は、是非試してください。

5. おわりに

テスト書くのは嫌ですと？

学生証番号の話

学部/回生 情理4回生 SNコース Cysec研

名前 Yukimura|木村 悠生|@yuki12017825



おはこんばんちは！Yukimuraです。今回は学生証番号の話をします。
そう、立命生なら折に触れて入力させられるアレです。11桁のアレ。
今回は、そんな11桁のアレの秘密について、赤裸々に記していきます。

※悪用厳禁です。偽の学生証番号を作ることもできますが、
兎に角、悪用厳禁です

さて、早速ですが学生証番号について、分解してみていきましょう！
分解してみていきましょう、ということはつまり分解して見ることができる、ということですが、これら辺まではもうわかってる方もいるで
しょう

2 6 0 0 1 8 0 8 9 2 - 2

学部 学科 入学年度 学科通し番号 チェックデジット
以上の通りです。この後の展開が読めた方は他ページへどうぞ！

…というわけで、後の展開が読めなかつた皆さんと分かったけど読んでくださる皆さんに向けて、頭から説明していこうと思います。

最初の2桁は学部です。例えば情理生はみんな最初が26ですが、理工学部・文学部などの大規模学部は一つの学部に複数の番号が割り振られていることがあるようです。一覧は下表の通り。

複数の番号が当てられている学部については、学科ごとに振り分けられているようです
あと何故か24番が欠番。新しい学部を作る予定だったのか現行で作る予定があるのか何なのか…とにかく欠番です。

次に、3,4桁目の学科についてです。これは、学部によって様々ですので詳細については

11	法学部
12	経済学部
13	経営学部
14	産業社会学部
15	国際関係学部
16	文学部
17	文学部
18	政策科学部
19	映像学部
20	総合心理学部
21	理工学部
22	理工学部
23	理工学部
24	(欠番)
25	食マネジメント学部
26	情報理工学部
27	生命科学部
28	薬学部
29	スポーツ健康科学部

触れませんが、一学部一学科の学部については、00か11である場合が多いようです。ちなみに情理の情報理工学科は00。ISSEも00です。

続いて5,6行目。ここは入学年度です。2020年度入学なら20、2018年度入学なら18。これ、立命館があと100年続いたらどうするつもりなんでしょうね…

7~10行目は学科内での通し番号です。名前順に学科内で通し番号が振られています。同じような苗字の人でもこの数字の大きさが違うと、学科規模の差を感じるところですね！そしてここで、1頁目に例示した学生証番号が偽物だということがわかるでしょう。情理は一学年に800人もいないので…

そして11桁目がチェックデジットです。チェックデジットとは何ぞやという方のためにWikipediaに聞いてみると、"符号の入力誤りなどを検出するために、元の符号に付加される数字のこと。"とあります。

一定の計算式で計算されているもので、これで学生証番号のミスを検知できる、という大変素晴らしいのですが、逆に計算式さえ分かってしまえば"それっぽい"学生証番号を作ることができます。また最後の一桁だけ忘れちゃったときにも大助かり！！

そんな立命館大学学生証番号のチェックデジット計算式は、こちら！

$$k=11-(ax5+bx4+cx3+dx2+ex7 +fx6+gx5+hx4+ix3+jx2)mod11$$

(学生証番号は「abcdefghijkl-k」の形式であるとし、k=10,11のときk=0として扱う)

簡単に言えば、所定の数を各行に掛けて、その総和を11で割って余りを算出し、11からその数を引いた答えがチェックデジットだということです！簡単ですね！！暗記しておけば学生証の入力ミスチェックが超簡単にできます！え？四則演算とか余りとか難しい？そうですか…

…そんなことに脳ミソ使ってらんない皆さんのために、学生証の正誤が簡単にチェックできるwebサイトを作りました！下に貼ってあるQRコードからアクセスできますので、遊んでみてください。



さていかがだったでしょうか？？(テンプレ)

なんとなく知ってても、学部の各番号やチェックデジット計算式までは触れた人は少ないんじゃないかな？

と思って今回のテーマを選びました。学生証を見るたびに、「学生証番号にはこんな話があったなあ」と思いだしてくれると嬉しいです！

それではまたいつか！！

RCC会誌 2021

— 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用いて行われます。全体に向けて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えのないものであったか、強く肌で感じこととなりました。

約2年弱の時を経てサークルルームが利用再開し、今後はコロナ対策に細心の注意を払いながら、対面活動を再開させていく方針です。

ぜひ見学にお越しください。

2021年12月5日 初版第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

