







目次

P.3 ……SLP の活動

P.6 ……学生の製作物

P.19 ……アドベントカレンダー



香川大学 KBIT

学生プログラミング研究所 SLP



Tel : 087-864-2284
 Web : <http://poulenc.eng.kagawa-u.ac.jp/Pub/KBIT/>
 Mail : slp.kbit@gmail.com
 所在 : 香川大学 創造工学部 1号館 9F演習室
 Twitter : @2021Slp

KBITの概要

KBITは、電子情報通信学会四国支部
 香川大学スチューデントブランチです。
 香川情報技術学生支部の略称です。
 下部組織 SLP (学生プログラミング研究所)が
 メインで活動を行っています。

年表

年	出来事
1998	香川大学工学部開校 富永研究室での自主ゼミ活動開始
2003	SLP発足(自主ゼミの発展的改称) ICPCアジア大会(会津大学)出場
2004	米国BotBall大会(サンノゼ)出場
2005	ICPCアジア大会(ノウル)出場
2006	CESAスチューデントゲーム大会参加 ETロボコン関西予選出場～
2007	KBIT発足 (SLPの上位組織として)
2010	EPOCHコンテスト 本戦 5位 ゲーム学会 企画セッション 参加
2011	新部室完成 セブキャン 参加
2012	ETロボコン 中四国予選5位
2015	OSC徳島出展

Webページ



プロジェクトページ



年間活動表 (2015)

月	活動
04	新入生勧誘 情報処理技術者試験 花見大会
05	歓迎会BBQ
06	危機管理コンテスト
07	ACM-ICPCコンテスト
08	オープンキャンパス 高大連携のLEGO講座 企業訪問 サークル旅行
09	夏季合宿勉強会 ETロボコン SJCIEE 学会発表 / 支部交流
10	外部講師による講演会 情報処理技術者試験
11	オープンキャンパス 科学体験フェスティバル OSC徳島
12	SECCON2015オンライン予選 忘年会
01	新年会(先生宅)
02	送別会
03	春季合宿勉強会



シェルスクリプトマガジンで連載開始

「香川大学SLPからお届け!」という記事を、シェルスクリプトマガジンのvol.32号から連載しています。

SLPメンバの技術的な取り組みについて、紹介しています。

これまでに、Sensuによるサーバ監視、

Go言語によるWeb アプリ開発、

Reactによるネイティブアプリ開発、

Swiftによるシェルスクリプト・

iOSゲームアプリ製作、CTFの問題解説
についての記事を掲載しています。



ロボットコンテスト

組込み技術者向けの国内大会「ETロボコン」に、8年連続して出場しています。この大会では、LEGO NXTを用いてライントレースを行い、走行タイムを競います。さらに、設計の志向をUMLを用いて表現し、正確さと可視性を評価します。参加チームは毎年増えており、2011年度からは中四国地区が開設されました。



ETRコース図

ETロボコン: <http://www.etrobo.jp/2015/>

ETロボコンの参加者は、組込系の企業の若手、大学や高専の学生です。

SLPは、2015年度には、Lコースを完走し、全国で91位となりました。

二輪で倒立振子を行いながら走行するマシンを用い、規定のコースをライントレースしながら走行します。走行タイムだけでなく、各所に設置された難所(破線、ツインループ、トレジャーハント)の攻略によって、ボーナスポイントを取得できます。2輪倒立振子の制御は難しく、ただ安定して倒立させるだけでも大変でした。



科学体験フェスティバル

小中学生を対象とした、科学や理科実験の面白さを伝える「科学体験フェスティバル」に毎年参加しています。

LEGOを使ったプログラミング体験や、テンセグリティと呼ばれる安定した構造を持つ物体の作成体験などを行っています。誰かに教えるという体験と違う年代に接する機会として認識しています。



LEGOプログラミング体験



テンセグリティ作り

LEGO ロボットのプログラミング教室

- どんな実験なの？
コンピュータでロボットを動かすには、その動き方のルールを、順番に決める必要があります。これをプログラムと言います。プログラムは、誰かに教えるように書くことができます。
- 用意するもの
コンピュータ（こちらで用意します）
LEGO ロボットキット（こちらで用意します）
- 実験のしかたとコツ
レゴマインドストームは、レゴブロックで組み立てられたロボットをコンピュータでコントロールするものです。写真の白い小さい箱が「脳」です。プログラムは、パソコンで作成し、ケーブルで「脳」に送ります。プログラムは、命令の順番を書きプログラムを組み立てて、かんたんに作ることもできます。ロボットの部品には、モーターやギアなどの動力、向きに変わったことや色や音を感じるセンサーなどがあります。書き出したプログラムを送ると、いろいろなロボットの動きと、プログラムのしくみを理解しよう。
- 気をつけよう
レゴには小さな部品や、とがった部品、とても早く動く部品などがあります。落ちた部品は拾って片づけてください。怪我にはご注意ください。

定期的な小中高校生を対象として、LEGOを使ったプログラミング教室を開催しています。

LEGOロボットを動かすためのプログラムを製作することを通じて、論理的に考える力を身につけてもらうことを目的としています。また、プログラミングや数学の楽しさを伝えています。

LEGO教室

LT

アプリの説明 その2



- 簡単なアニメーション
- スコア = 移動距離
- アイテムは2種類
 - ・ 加速アイテム
 - ・ 減速アイテム

今作っているもの「Battle Ship(仮)」

内容：自艦艇を移動させながら砲撃で敵艦を倒す。画面の任意の場所をタッチすることで、砲撃を放物線状に発射する。



Androidゲーム開発のこれから

- 現在絶賛開発中！！
- 横スクロールアクションゲーム(マリオ的な)



個人的な取り組み

Unreal Engine 4を使ったゲーム作りに挑戦した



SLPでは毎年、新入生も含めて自己紹介LTを行います。その他、参加したインターンシップやイベント、学習した技術についてのLTも行っています。



OS を製作している話

香川大学 創造工学部 2年 横山ひなた

お初にお目にかかります。SLP 所属 2 回生の横山と申します。過去の作品について記事にしている方が多いかと思うのですが、私は現在開発中のものについて文章にしていきたいと思います。初記事のため拙い部分も多いかと思いますが、最後までお読みいただければ幸いです。

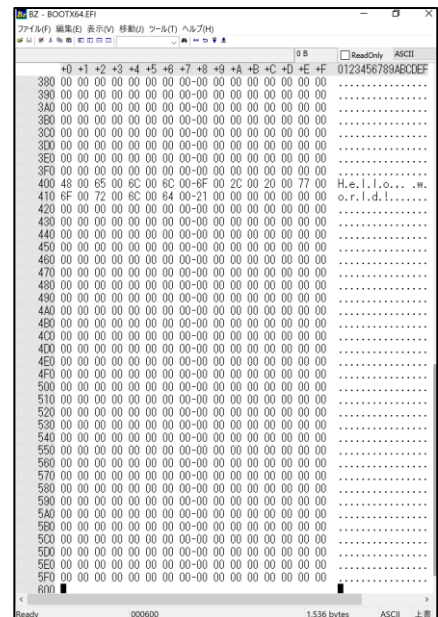
まず、そもそも何を作っているのかという話なのですが、『ゼロからの OS 自作入門』(内田公太著, マイナビ出版, 2021)を参考に自作 OS に取り組んでおります。「なぜ OS を作ろうと思ったのか?」ということに対しては、OS を題材とした授業を受けて興味を持ったため、調べていた結果自作 OS というものを知って取り組んでみようと思いついたからです。参考にした本が最近出版されていたこともあります。

ともあれ、いざやってみようとしたときに開発環境が整っていなかったため、いろいろダウンロード & 設定を行いました。今回開発に利用している OS は windows10 です。WSL2 上の Linux を用いています。また、試験用パソコンを準備できなかったため、VcXsrv を用いて試験を行っています。

開発手順は本の通りといっても過言ではないため、個人的に印象に残ったところでも書こうかと思えます。今までテキストエディタしか触ったことがなかったため、バイナリエディタを用いて書くということが凄く新鮮でした。用いたエディタは「Binary Editor Bz」です。機械語は説明のみでしか見たことがなかったため、実際に体感することは、今後とても役立つように思えました。

(Binary Editor Bz を用いて「Hello, world!」と表示する→)

今後の展望としては、まずは完成。そして、アドベントカレンダーの時に作った色が虹色に変わるペイント機能の実装を考えています。また、制作した OS を基本 OS としたパソコンでいろいろ動かしてみたいと思っています。



まとめとしては、OS を作っていること。今後の抱負として、完成後もさらなる機能を追加していきたいです。

ここまで長々とお付き合いありがとうございました。



Kawa-Rhythm

香川大学 創造工学部 3年 石上 椋一

1. 導入

今回は、Kawa-Rhythm という Web アプリを制作しました。この Web アプリは Twitter、Youtube、Amazon、Pixiv の 4 つのメディアを取り扱えるものとなっています。この Web アプリを制作しようと思ったきっかけは 2 つ存在します。

1 つ目は、ブラウザの操作に興味を持ったからです。私は、この制作物とは別で画像解析の勉強を行っています。画像解析を行う上でスクレイピング技術を使用し、画像収集を行っていました。そこで、ブラウザの操作に興味を持ち、それに関する制作物を制作したいと感じたのが 1 つ目のきっかけです。

2 つ目のきっかけは、使用するメディアを 1 つにまとめたいと感じたからです。私はよく夜まで Twitter をしたり(ツイ廃)、Pixiv で画像をよく漁ったりしています。しかし、メディアを変えるたびにブラウザのタブを切り替えて、一回一回ログインや検索をし直すのがめんどくさいと感じていました。よって、このような Web アプリを制作すれば解決できると考えました。

以上、2 つの理由からこの Web アプリを制作しようと思いました。

2. 使用言語、技術

- Ruby
- Rails
- Watir
- Chrome driver
- Twitter API

3. 仕様

現段階では、ローカル環境でのみ使用することができます。各メディア用にログインボタンと検索ボタンを作成し、Twitter 用にはツイートボタンを実装しています。ログインボタンを押した場合、Google Chrome が自動で開かれ、選択したメディアの画面に移動し、ID とパスワードが自動で入力されます。検索ボタンを押した場合は、上記と同じくログイン処理が行われ、テキストボックスに入力された単語で検索を行います。ツイートボタンを押した場合、テキストボックスに入力されたメッセージがツイートされます。

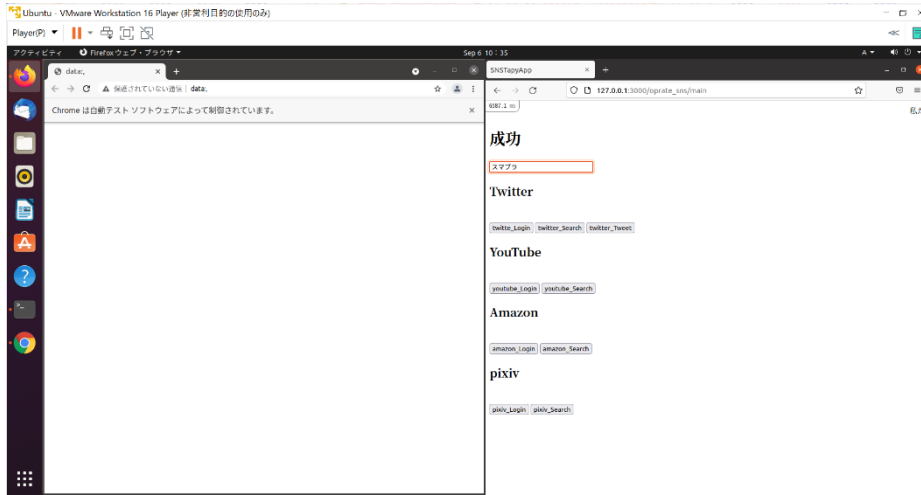


図 1 入力画面

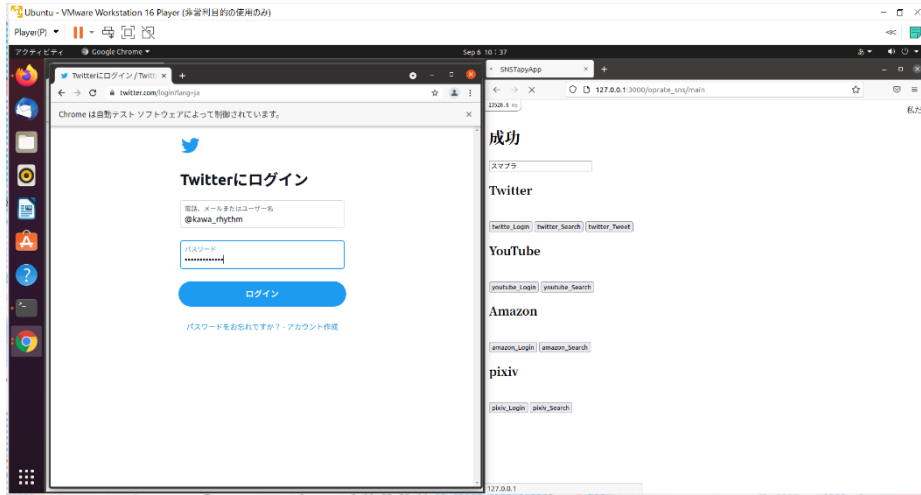


図 2 ログイン画面

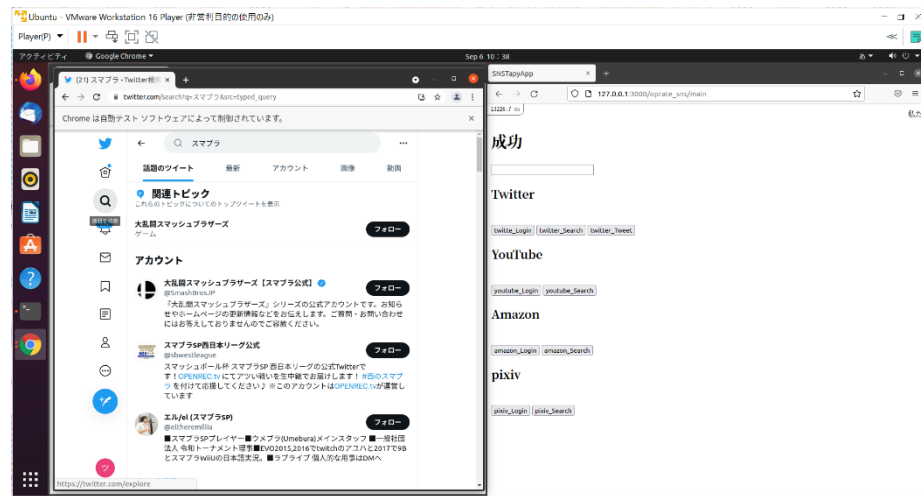


図 3 検索画面



4. 今後

現在は、各メディアに対して別々に操作を行っていますが、複数のメディアを組み合わせたいと思っています。例えば、Pixivにある画像をツイートしたい場合、自動保存、作者やURLの取得、Twitterの自動ログイン、自動ツイートといった流れを行えるようにしたいと考えています。さらに、難しいかもしれませんが、Amazonで商品をカートに入れる作業も行えたら良いなと思っています。また、現在は4つのメディアしか取り扱っていませんが、FacebookやInstagramなど取り扱うメディアを増やしていきたいです。

5. 参考文献

<http://watir.com/>

<https://github.com/watir/watir-rails>

<https://www.seleniumqref.com/>

<http://kakedashi-xx.com:25214/index.php/2021/05/27/post-2533/>



Linux コマンド学習基盤 Web アプリケーション

香川大学 創造工学部 3年 三枝泰士

導入

開発者にとって、Linux コマンドを扱う機会はなきにしもあらずだと思うが、どのコマンドをどのように使えば何ができるのかを学ぶ機会は、実際に触ってみる他にはほとんど無いだろう。そこで、誰でも手軽にシェル芸 Linux コマンドを学習できるようにするため、Web インタフェースを通したコマンドの実行を可能にした。

悪意のあるコマンドを防止するため、gVisor を利用したサンドボックス環境を利用している。

使用技術

- Docker
- gVisor
- Python

概説

問題とそれに使用するファイルが用意されており、問題の解答が算出されるコマンドを入力し送信することで、正誤と実行結果が表示されるものとなっている。

実装は、Docker コンテナとして、ASGI Web サーバー(FastAPI)、Nginx、postgresql が構成されている。Web サーバーに、コマンドを実行するリクエストが届くと、新たに Alpine Linux のコンテナが gVisor を使って作成され、コマンドを実際に実行する。実行結果は Web サーバーに返却され、ブラウザに反映される。



esrever

テキストを反転して表示せよ

[No.1 Question File](#)

```
cat q_1.txt | ...
```

EXECUTE

g/re/p

DISTRIB_CODENAMEの値を取り出せ

[No.2 Question File](#)

```
cat q_2.txt | grep 'CODENAME'
```

EXECUTE

#

hogeという文字列を、md5でハッシュ化せよ。例として、ファイルは、piyoという文字列をmd5でハッシュ化したものである。

[No.3 Question File](#)

```
cat q_3.txt | ...
```

EXECUTE

Correct!

```
DISTRIB_CODENAME=focal
```

工夫点

gVisor を利用している。gVisor とは、Google が提供しているコンテナランタイムで、ユーザー空間にカーネルを用意することで、ホストマシンのカーネルを不正なシステムコールから保護することができる。これを使うことによって、コンテナを利用しつつ、他のコンテナに影響を与えないようにしている。また、リソースを消費させないように、プロセス数やメモリ、実行時間等の制限をかけている。



問題点

上記の通り、リソース制限を設けてはいるが、ストレージの制限はできていない。実行後のコンテナは即座に消去されるものの、大量のファイルを作成された場合、多くのストレージが浪費されてしまうだろう。対策としては、ホストマシンでの `quota` の設定や、`docker` の `--storage-opt` オプションを使用することがある。しかし、ホストマシンの設定の変更はホストマシンを汚すことになるため、やりたくない。また、`docker` のオプションを使うにも、`docker` のストレージドライバを変更する必要があり、変更先の `devicemapper` は非推奨で、将来的に削除される予定である。さらに、`dm.basesize`(最小コンテナサイズ) が 1GB より小さくできない。もっといいやり方が欲しいところである。

今後について

今後の機能追加として、問題に点数を設け、ユーザーごとの進捗やランキング機能があれば、競技のような使い方ができ、面白い学習ができると考えている。

そのため、問題への点数設定、ユーザーの作成とログイン機能、ランキング集計とそのグラフィカルな表示機能の追加ができればいいだろう。また、現在は問題数が 3 問しかないため、問題の充実も必要である。

最終的にはサークルで使用できる品質にできればいいなと思っている。



クリーンアーキテクチャとJWT を用いて アプリケーションを作ってみた

香川大学 創造工学部 3年 樋口史弥

目次

- アプリケーションの仕様
- 使用技術
- クリーンアーキテクチャって？
 - 実際に使ってみたメリット・デメリット
- JWT って？
 - JWT が生成される流れ
- 最後に

アプリケーションの仕様

今回作成したアプリケーションは、ユーザ名とパスワードを含んだリクエストを受けるとユーザ名とパスワードのペアが正しいか検証し、正しい場合はトークンを返すアプリケーションです。ソースコードは以下のリンクにあります。

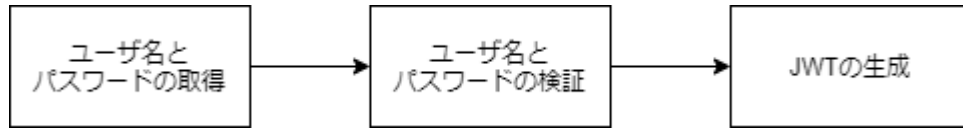
<https://github.com/higuruchi/certification-app>

以下は実行時のデモです。

```
$ curl -w '%n' 'http://<IP Address>:<Port>/' --data 'name=higuruchi&password=pass' -XPOST  
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhZG1pbWV3ZG11ZSswZXhwIjozNjMxNjgxODk3ZQ.YEW_rN5T2rQyferN2kt2V3up3fW5N030jIDphkzXx7JU"}  
$ curl -w '%n' 'http://<IP Address>:<Port>/' --data 'name=higuruchi&password=faultpass' -XPOST  
{"message":"unauthorized"}
```



処理の流れ s

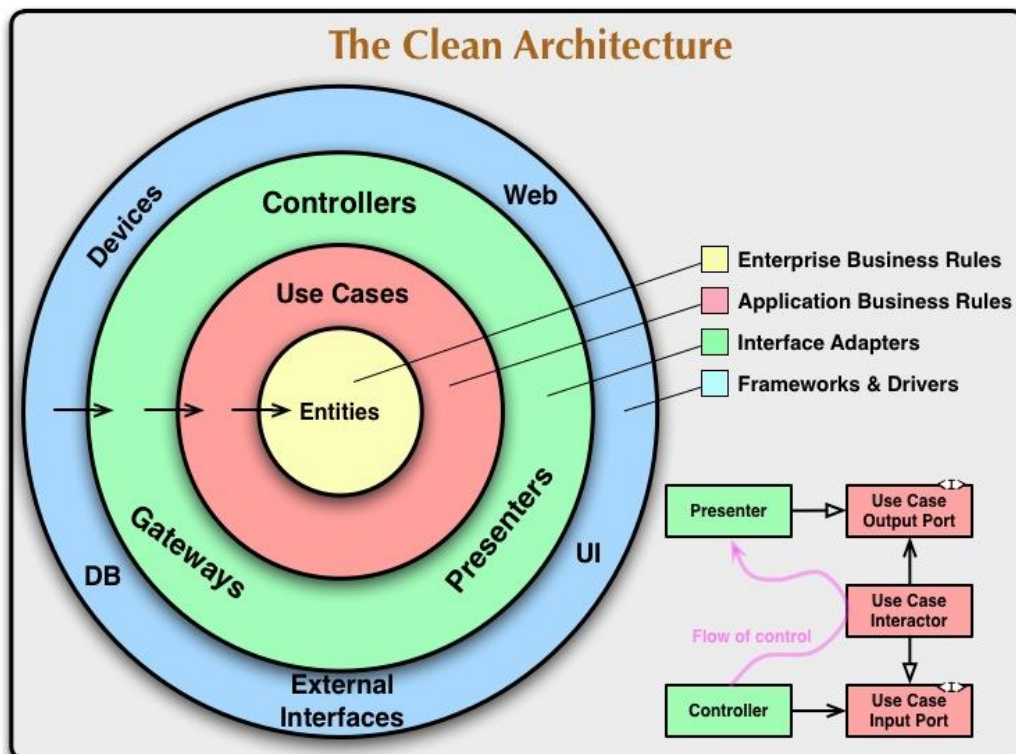


使用技術

- Go 1.17
 - Echo <https://echo.labstack.com/>
 - Wire <https://github.com/google/wire>
- MySQL Ver 8.0.26

クリーンアーキテクチャって？

クリーンアーキテクチャとは下図の通り External Interface ・ Interface Adapter ・ Usecase ・ Entity の 4 層になっており、それぞれの層を分離し、依存の流れを外から中の一方向にすることを目的としたアーキテクチャです。





それぞれの層の解説

- Enterprise Business Rules
アプリケーションのビジネスロジックを表現する Entity が所属するレイヤー
- Application Business Rules
Entity が持つ振る舞いを表現するためのレイヤー
Entity に所属するオブジェクトと協調しユースケースを達成する
- Interface Adapter
Frameworks & Drivers の層から来たデータを内側の層で扱えるデータの変換にする役割を果たす
- Frameworks & Drivers
データベースの操作などの外部との連携の役割を果たす

実際に使ってみたメリット・デメリット

実際にクリーンアーキテクチャを利用したアプリケーションを作ってみて感じたメリット、デメリットは以下になります。

メリット

- 内側の層は外側の層がどのような実装になっているかなどを気にしなくてよい実装になっているため、データベースなどのミドルウェアを変更した場合でも最小限の変更で済ませることができる
- どの層にどのような処理が書いてあるかがわかりやすいため、目的のコードを探す手間を省くことができる

デメリット

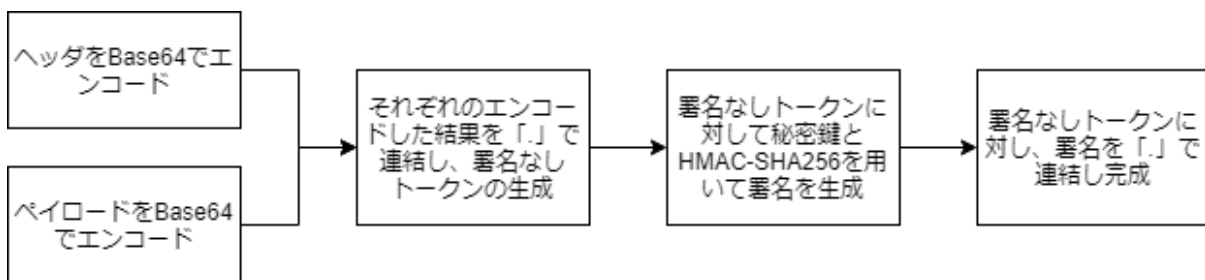
- 依存関係を解消するためのインターフェースを実装しなければならず、プログラム全体が複雑になり、ファイルやソースコードの行数も増加することがある



JWT って？

JWT とは JSON Web Token の略称で、属性情報を JSON データ構造で表現したトークンの仕様です。ヘッダ、ペイロード、署名の3つの要素から構成され、トークンを書き換えた場合、署名と照合することにより書き換えを検出することができます。

JWT が生成される流れ



最後に

今回のアプリケーションはとても小さく、クリーンアーキテクチャのメリットよりもデメリットのほうが大きくなってしまいかもかもしれませんが、チーム開発をする際や、保守性などが大切になってくる場合、それぞれの層が分離され疎結合になっているためインターフェースさえ決まっていれば同時に進めることができ、外部との連携方法を変える際にも最小限の変更に抑えることができます。これを機会に皆さんもクリーンアーキテクチャを学んでみてはどうでしょうか。



VR で犬と遊んでみた

香川大学 創造工学部 4年 重松亜夢

こんにちは！重松亜夢です。最近実家にもあまり帰れていないため、実家の犬が恋しいです。そこで、犬と遊べる VR ゲームを作り始めました。ソースコードはこちら

(<https://github.com/Amakuchisan/DogPartyGame>)です。

現在実装している機能は、ボールを投げると、犬がボールを追いかけるという、一般的な遊びです。まだまだ未完成ですが、作成途中のゲームの雰囲気を、ざっくりとお伝えできればと思います！

犬のモデルの Asset について

犬のモデルには、こちらの Asset を利用しています

(<https://assetstore.unity.com/packages/3d/characters/animals/lowpoly-dogs-pack-164065>)。



このパックには、7種類の犬と、65個のアニメーションが含まれています。例えば、「座る、走る、ジャンプする、寝転がる」などの動きを設定することができます。

実装

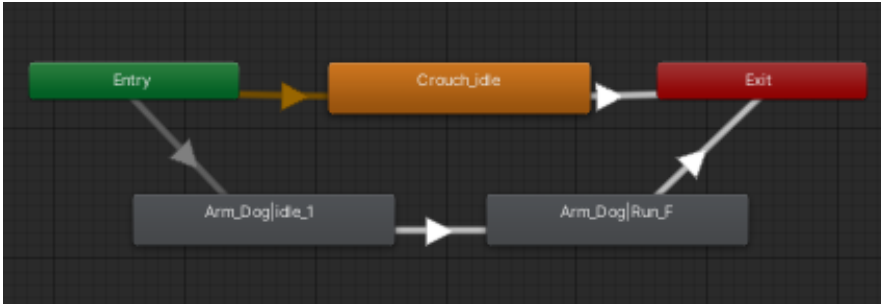
Terrain の機能を使って、地面を作成し、草を生やし、木を植え、犬(レトリバー)を1匹配置しました。そして、弾性のあるボールを作成し、犬の横に置きました。

また、犬の正面に、プレイヤーを配置します。この時点で既に犬をかわいく感じます。

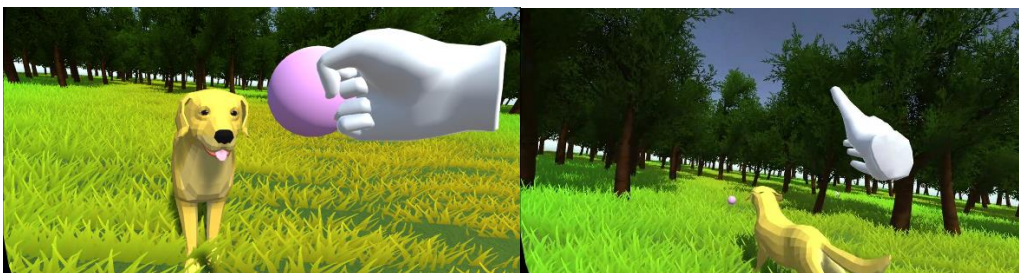




アニメーションの設定は、下図のようになっています。通常の状態では、Entry から始まり、図の上側(待機状態: Crouch_idle)の遷移を行います。ボールをプレイヤーが持つと、Entry から、図の左下(状態状態:idle_1)へ遷移します。そして、その状態でプレイヤーがボールを投げると、ボールを追いかけて走る状態(Run_F)へと遷移します。



上記のアニメーションに遷移するように作成した結果、下の画像のようになりました。ボールを持つと、画像左のように、犬がボールを見つめてきます。その状態でボールを投げると、右のようにボールを追いかけてくれます。追いかけている様子がとてもかわいく、気に入っています。動くともっとかわいいです。



まとめと今後の予定

このように、犬がボールを追いかけて、その犬をプレイヤーが追いかけるだけの簡単な機能を実装しています。本当は、もう一つ、特技を習得させる予定だったのですが、原稿の締め切りに間に合いませんでした。読者の方は、ぜひ、今後実装されるであろう機能を予想してみてください。また、こんな機能あったら楽しそう！というアドバイスも頂けると嬉しいです！

最後に、現在、ボールを追いかけてはくれますが、まだ投げたボールをくわえて持ってくることはできないため、ボールをくわえて持ってきてくれるようにしたいです。これから、もっと遊びのバリエーションを増やしていくことを目指しています！



悪魔の数字サイト

URL : <https://cd-donki-gorira.hatenablog.com/entry/2020/12/01/031107>

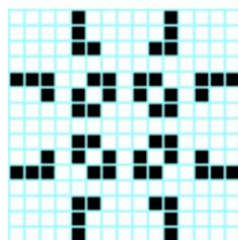
概要 : Ruby on Rails を使用し、Webアプリを制作。
誕生日から、その人の運勢を占うサイトである。



JavaScriptでライフゲーム制作

URL : <https://yassi.hatenablog.com/entry/2020/12/02/000000>

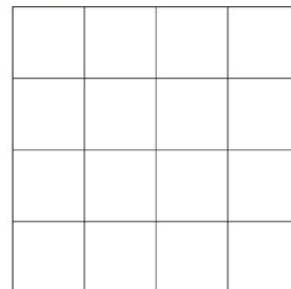
概要 : 初めに、基盤の大きさを決定し、ドット絵を描く。
そこから、シミュレーターを起動させる。



LifeGame

バブル入力(デフォルト) | バブル数決定

AICheck | HelpGif | PLAY





Pythonでメール送信

URL : https://qiita.com/yasu17_21/items/6761e679ffeca4650829

概要 : PythonでGmailのメールを送信するコードの作成。
自身のメールサーバーから他のサーバーにメールを送る。

```

import smtplib

def prompt(prompt):
    return input(prompt).strip()

fromaddr = prompt("From: ")
toaddr = prompt("To: ").split()
print("Enter message, end with ^D (Unix) or ^Z (Windows).")

# Add the From: and To: headers at the start
msg = ("From: %s\r\nTo: %s\r\n\r\n" % (fromaddr, " ".join(toaddr)))
while True:
    try:
        line = input()
        except EOFError:
            break
        if not line:
            break
        msg = msg + line

print("Message length is", len(msg))

server = smtplib.SMTP('localhost')
server.set_debuglevel(1)
server.sendmail(fromaddr, toaddr, msg)
server.quit()

```

```

server.ehlo()
server.starttls()
server.login('自分のGmailのアドレスを入力', 'パスワードを入力')

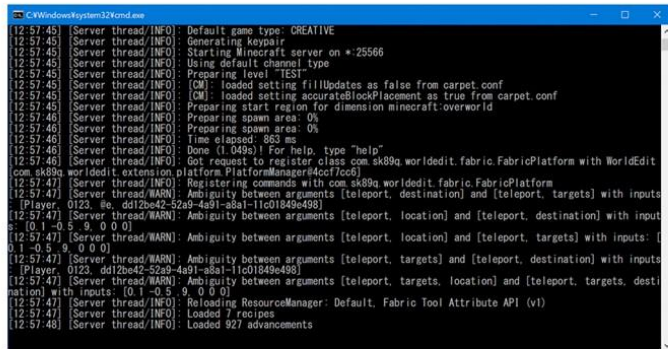
server.set_debuglevel(1)
server.sendmail(fromaddr, toaddr, msg)
server.quit()

```

Discord botを作成

URL : https://qiita.com/Gomatamago_/items/e38f50d764cfc2dc20b6

概要 : DiscordのAPIを使用し、Pythonで作成。
botからサーバーの起動と停止を行えるものを作成。

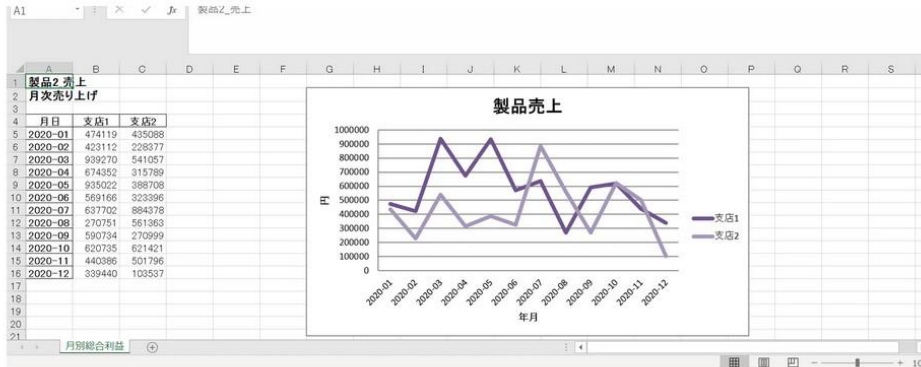




PythonでExcelを動かしてみよう

URL : <https://qiita.com/g75hca/items/3724e7609e70cfdcf8bb>

概要 : PythonでExcel操作を行えるコードを作成。
データ分割およびグラフ作成の機能を実装。



C言語で時間管理

URL : https://qiita.com/ayumimi_0929/items/6af4c7b39a9dc4ae832a

概要 : C言語でタイマーとストップウォッチを作成。

タイマーを何秒後にセットしますか？ 3

時間です！

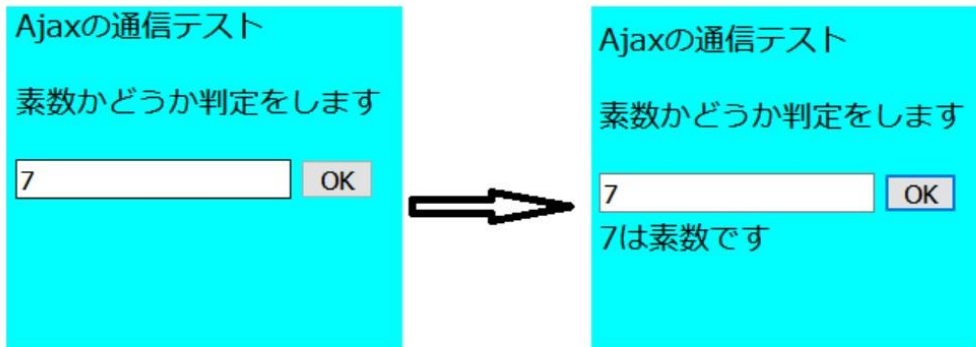
ストップウォッチを開始します。
止めるときはEnterキーを押してください。

1.821秒です。



Ajaxについて調べてみた

URL : <https://higuruchi.hatenablog.com/entry/2020/12/18/205657>
概要 : jQueryとPHPを使用し、簡単にAjaxを実装。



ビット演算の勉強

URL : https://qiita.com/SSK_Ayagawa/items/544a277101948f8307bb

概要 : 競プロを機に、ビット演算についてまとめた。
扱う例として、C++のbitsetクラスを紹介。

```

#include <iostream>
#include <vector>
#include <bitset>
using namespace std;

int main(){
    int n, k;
    cin >> n >> k;
    vector<int> a(n);
    for(int i = 0; i < n; i++) cin >> a[i];

    bool ans = false;

    // 素因数を求め、約数kになるものを求める
    for(int i = 1; i <= n; i++) {
        if(a[i] % i == 0) {
            bitset<30> b(a[i]); // a[i]のビット列を生成
            if(b.count() == k) ans = true;
        }
    }

    if(ans) cout << "Yes" << endl;
    else cout << "No" << endl;

    return 0;
}

```

ビット演算	bitsetの演算子	つかいかた
AND	&	変数 1 & 変数 2
OR		変数 1 変数 2
XOR	^	変数 1 ^ 変数 2
NOT	~	~ 変数
左シフト	<<	変数 << シフトするビット数
右シフト	>>	変数 >> シフトするビット数



便利だったもの紹介

URL : <https://i13302.github.io/30808.html>

概要 : GitHub Actionsと Dockerfileを用いたLaTeXの論文執筆テンプレートを紹介。

✖ master push	master
master push #1: by i13302	
✔ Merge pull request #1 from i13302/Chap/...	develop
push build #9: Commit fe4cc42 pushed by i13302	
✔ typo修正	Chap/1
push build #8: Commit e3ddf29 pushed by i13302	
✖ 改行を追加	Chap/1
push build #7: Commit f380bc2 pushed by i13302	

Neural Network Consoleを使った機械学習

URL : <https://c-r-t-h1235.hatenablog.com/entry/2020/12/07/174000>

概要 : サンプルプロジェクトとしてはじめからついてくる、
画像認識の最も簡単なプロジェクトを実行。

The screenshot shows the Neural Network Console interface. On the left is a sidebar with navigation options: 'ダッシュボード', 'プロジェクト', 'データセット', 'ジョブ履歴', 'サンプルプロジェクト', '公開プロジェクト', '公開API', and 'サービス設定'. The main content area is titled '目的から探す > 画像認識' and has tabs for '目的から探す', 'テクニックから探す', 'データセットから探す', and '人気'. Below the tabs, there are two project cards. The first card, 'tutorial.basics.01_logistic_regression', is highlighted with a red box and shows a diagram of a neural network with layers labeled 'Input', 'Affine', 'Sigmoid', and 'BinaryCrossEntropy'. The second card, 'tutorial.basics.02_binary_cnn', shows a diagram of a convolutional neural network. Both cards have 'Cloudで実行' and 'Download' buttons.

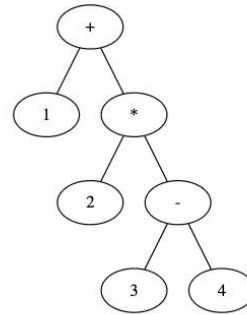


日本語プログラミングを自作

URL : <https://qiita.com/hirany/items/6d1462b0b27d7490d67e>

概要 : 日本語プログラミング言語、ogengoを自作、紹介。

```
def w_separator(line):
    tagger = mecab.Tagger()
    words = tagger.parse(line).split('\n')
    wordslist = []
    code = ""
    for v in words:
        w = v.split()
        if w == ['EOS']:
            break
        word = {'value': w[0], 'kind': w[1].split(',')[0]}
        wordslist.append(word)
    for i in range(len(wordslist)-2, -1, -1):
        if wordslist[i+1]['value'] in reserved:
            continue
        if wordslist[i+1]['kind'] == '名詞' and wordslist[i]['kind'] == '名詞':
            wordslist[i]['value'] += wordslist[i+1]['value']
            wordslist.pop(i+1)
    for v in wordslist:
        code += v["value"] + ' '
    return code
```



GASとSlackを接続

URL : <https://qiita.com/gacky35/items/e131fcff6dda58ad298e>

概要 : Google スプレッドシートと Google Calendar を Slack に連携。
Google Apps Script(GAS) でプログラムを作成。

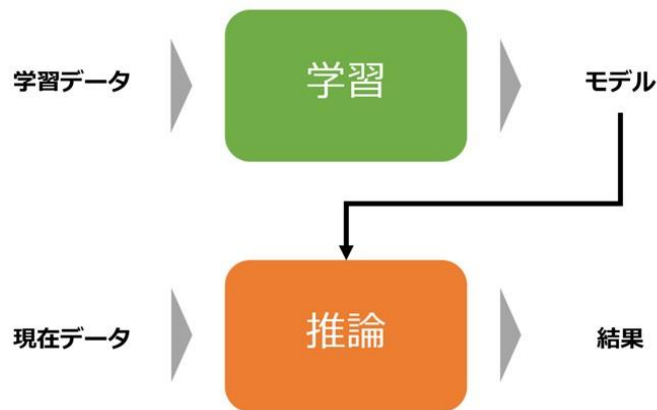




機械学習のおおまかな流れを理解

URL : <https://noleff.hatenablog.com/entry/2020/12/13/033102>

概要 : 機械学習のおおまかな流れに焦点を当て、まとめた。



BasicPlayerを使って音楽プレイヤーを作成

URL : <https://qiita.com/kakukaku-15/items/abb7ee011b7cf4aafc34>

概要 : BasicPlayer APIを使用して、音楽プレイヤーを作成。
VSCodeの拡張機能“Java Extension Pack”を使用して開発。

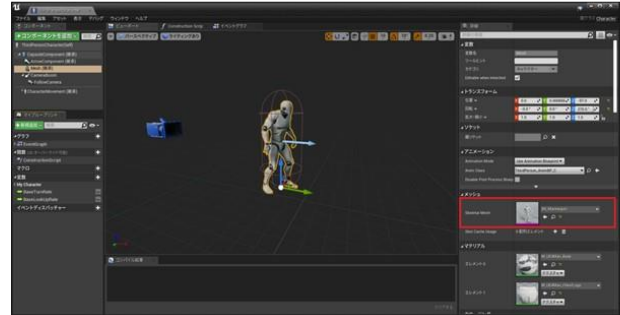




UnrealEngine4を使って 簡単な3Dゲーム作成

URL : <https://neokiex.hatenablog.com/entry/2020/12/15/152224>

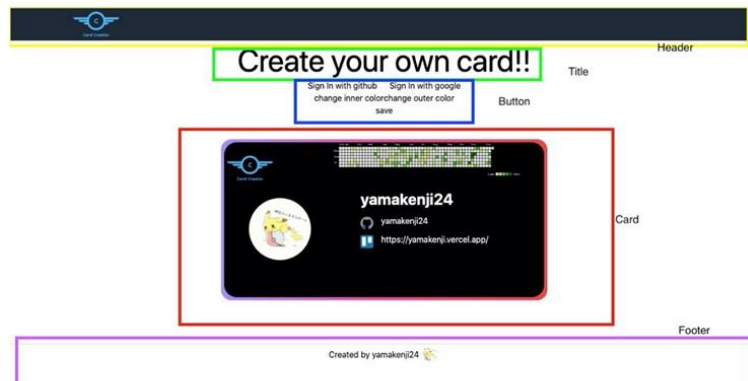
概要 : プロジェクトの作成から、プレイヤー、マップ、エネミーの作成。



Next.jsとTailwindを利用した エンジニア名刺アプリ

URL : <https://zenn.dev/yamakenji24/articles/2921714d4b69d0>

概要 : GitHub認証とGoogle認証を用いてプロフィールデータを取得、表示するアプリを実装。

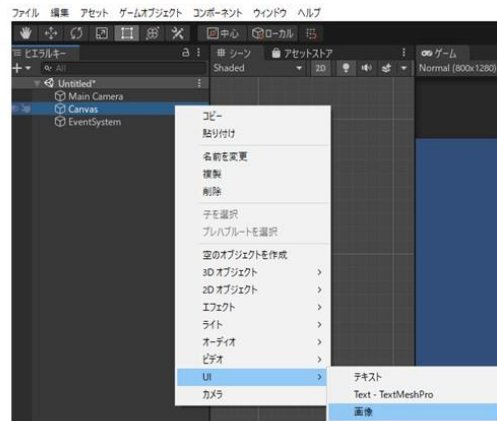




Unityで脱出ゲームを作成

URL : <https://qiita.com/4Hceo/private/1af736f7a56afc2541b1>

概要 : Android向けの脱出ゲームを作成。



フルサービスリゾルバを可視化

URL : <https://qiita.com/pikapika2/items/84d9203d09d07442c1de>

概要 : フルサービスリゾルバの可視化。
JSで作成し、コード解説を行った。

