KGW UNIV

香川大学

S L P 2022

目次

香川大学 KBIT

学生プログラミング研究所 SLP

Tel ：087-864-2284

Web：https://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：@KbitSlp

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(ソウル)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.

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

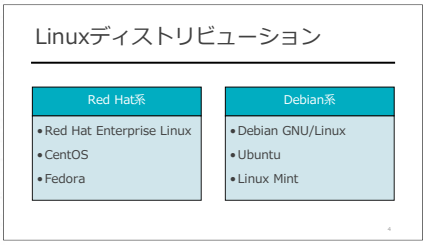
テンセグリティ作り

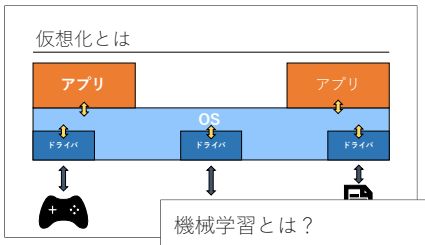
定期的に小中高校生を対象として、LEGOを
使ったプログラミング教室を開催しています。

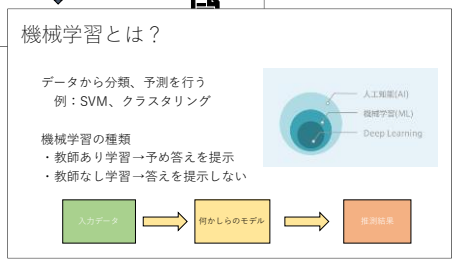
LEGOロボットを動かすためのプログラムを
製作することを通じて、論理的に考える力を
身につけてもらうことを目的としています。
また、プログラミングや数学の楽しさを伝え
ています。

LEGO教室

LT



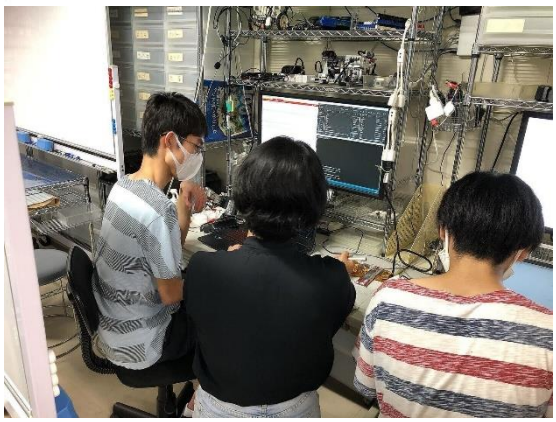




SLP では毎年、新入生も含めて自己紹介 LT を行います。
その他、参加したインターンシップやイベント、学習した技術についての LT も行っています。

ICPC

ICPC（国際大学対抗プログラミングコンテスト）に参加しました。
３人一組のチームを作り、チームで協力しプログラミングと問題解決の能力を競いました。



複数ある問題のどこにどれだけ時間をかけるかなど、各チームで工夫して取り掛かっていました。

技術書展

技術書展 13 にサークルの書籍「Technologics Vol.1」を出展しました。
技術書展マーケットでオンライン販売しています。
https://techbookfest.org/product/paU7Rsg13FdKzd62n0wWvy?productVariantID=r7bL8g9czxd02P2tRQT43K

<目次>

サークルメンバーそれぞれが、興味のある技術について自由に記事を書いています。
今回は、サーバ構築、画像認識、ブロックチェーン、ゲーム、LaTeX 等々、総勢１７名が参加した大ボリュームの内容です。

簡易バ美肉

香川大学 創造工学部２年 谷﨑勇太

1. はじめに

皆さん、バ美肉をしてみたいと思ったことはありませんか。バ美肉をご存じない方もいるかもしれないので説明します。２D、３D のキャラクターを自身の動きにリンクさせ配信している人を Vtuber と呼びます。その中でも、美少女のアバターを纏うことを、バーチャル美少女受肉（バ美肉）といいます。

Zoo などでの顔出しって恥ずかしいですよね。そういったときにも使用できるシステムになっています。

2. 使用するもの

- Python 3.10.2
- OpenCV
- Dlib
- PIL
- イラスト

3. 仕様

カメラで取得した画像を編集し表示するシステムです。編集内容は、顔の位置にイラストを重ねるというものです。また、目の開閉や笑顔の状態に応じたイラストを表示するようになっています。

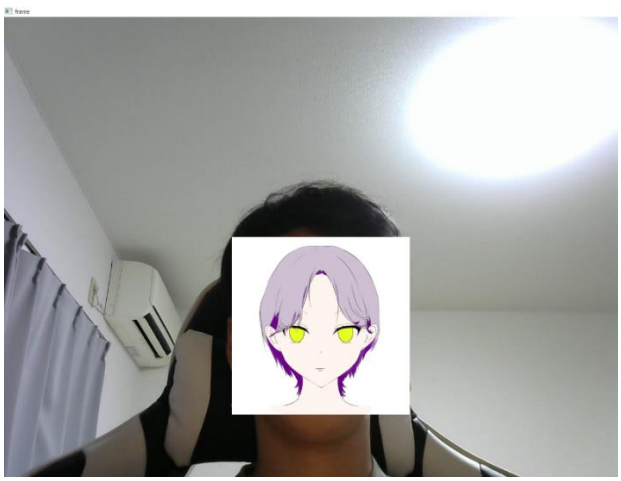


図 1 目が開いているとき

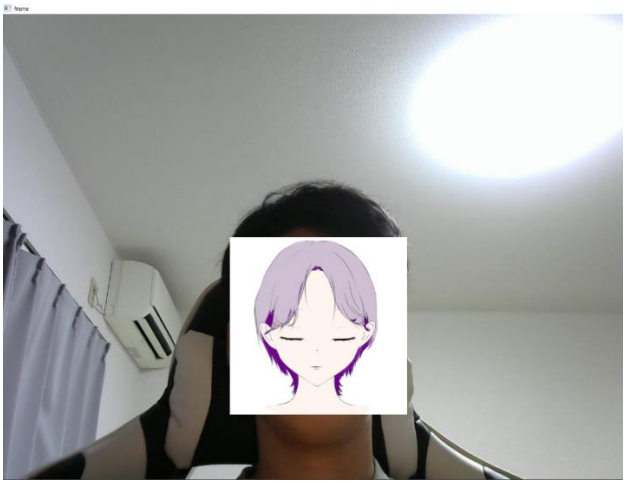


図 2 目を閉じているとき

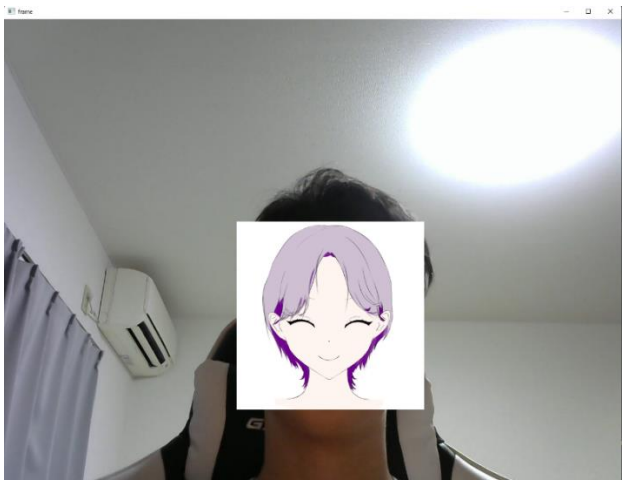


図 3 笑顔

4. 目の開閉判定

目の開閉判定は Dlib で検出されるポイントの距離を利用しています。

図 4 Dlib で検出されるポイント

図４では Dlib で検出されるもののうち、目の付近のポイントになります。
３７番と４０番の距離と、３８番と４２番、３９番と４１番の距離の比率で目の開閉を判定しています。

5. 笑顔の判定

笑顔の判定には唇の両端の距離で判定しています。通常時よりも口角が上がると笑顔判定になるようにしています。

6. イラストの表示

OpenCV を使用して顔の位置情報を取得し、その位置にイラストを重ねて表示するようにしています。各判定で別々のイラストを表示することにより、表情が現実と同じになるようにしています。

7. 最後に

現段階では表情が少ないため、表情のパターンを追加することを考えています。
このシステム単体ではパソコン上に顔をイラストに置き換えた画像が表示されるだけですが、VirtualCam というプラグインを使うことで Zoom などでの使用ができるようになります。詳しくは、[2]を見てください。
コードは GitHub に載せています。良ければ見に来てください。
https://github.com/tus18/vtuber

8. 参考文献

[1]https://qiita.com/satsukimain/items/ddc3e0f5bcf88f47e8cf
[2]https://loumo.jp/archives/24912

Google カレンダーに提出物や出席の予定を自動追加

香川大学 創造工学部 2 年 坂東恭幸

はじめに

皆さんは大学の提出物をちゃんと出していますか。GPA に関わることなので出すに越したことはありませんが、ついうっかり存在を忘れていたなんてこともあります。今回はそれを防ぐために Google カレンダーに提出物や出席の予定を自動追加するシステムを制作しました。

準備したもの

・Edge
・Moodle（大学のサイト）
・Google App Script（JavaScript）
・Google Calendar

手順

1. Moodle のカレンダーからソースを取得する
2. 取得したソースから必要な情報を抽出する
3. 抽出した情報を元に Google カレンダーにイベントを作成する

1. 大学のサイトにログインして必要となるページのソースを取得

ログイン処理を行うために、その流れを知っておく必要があります。今回はログイン処理の仕組みを解説することがメインではないため割愛しますが、やることとしては Web ブラウザが行っている通信の内容を GAS に実装して再現するといった感じです。

まず、ブラウザの開発者ツールからネットワークを開いた状態でログインを行います。開発者ツールの画面からリクエストやレスポンスの通信内容が確認できるため、これをもとに以下の a から c までのプログラムを書いてしまいます。

a.　ログイン画面表示のリクエスト

まずはログイン処理のリクエストに必要な Cookie を取得するため、ログイン画面表示のリクエストを行い、そのレスポンスから Cookie を取得します。ログイン処理のリクエストに必要な Cookie は全て必要というわけではなく、特定の情報だけで OK でした。今回、必要な Cookie は“MoodleSession”のみであったため、リクエストヘッダーの“Set-Cookie”から値を取得しました。

```
// ログインページを開く(GET) <200>
response = UrlFetchApp.fetch('https://kadai-moodle.kagawa-u.ac.jp/login/index.php');
```

b.　ログイン処理のリクエスト

レスポンスは POST となっており、Payload に”username”、 “password”、 “logintoken” がありました。“logintoken”は先ほどのログイン画面表示のレスポンスの HTML タグ内にあったため、cheerio と呼ばれる、スレイピングなどで利用される GAS 特有のライブラリを使って値を取得しました。では先ほど取得した Cookie を含め、これらをもとにログイン処理を実装していきます。

ここでの注意点として、1 つ目はブラウザからのアクセスであると偽装するために、送信元のユーザーエージェントを指定しておくこと、2 つ目はリダイレクト処理を行ってしまうと取得される情報がログイン処理のレスポンスではなく、ログイン画面のレスポンスになってしまうため“followRedirects”は false にしておきます。

```
// ログインフォーム送信(POST) <303>
headers = {
  'cookie': moodleSession,
  'user-agent': user_agent
}
payload = {
  'logintoken': logintoken,
  'username': id,
  'password': pass
}
options = {
  'method': 'post',
  'headers': headers,
  'payload': payload,
  'followRedirects': false
}
response = UrlFetchApp.fetch('https://kadai-moodle.kagawa-u.ac.jp/login/index.php', options);
```

無事、レスポンスが返ってきました。次のカレンダー画面表示のリクエストに必要な Cookie を取得したいのですが、レスポンスヘッダーを確認すると“Set-Cookie”が 2 つありました。そのため、その 2 つの中に“MoodleSession”があるかどうかを解析し、取得しました。

c.　カレンダー画面表示のリクエスト

最後にカレンダー画面表示のリクエストヘッダーにログイン処理後の Cookie とユーザーエージェントを指定し、GET メソッドでレスポンスを行います。URL はカレンダーのページを指定しておきます。

```
// 直近イベントのカレンダーのページ(GET) <200>
headers = {
  'cookie': moodleSession,
  'user-agent': user_agent,
}
options = {
  'method': 'get',
  'headers': headers,
}
response = UrlFetchApp.fetch('https://kadai-moodle.kagawa-u.ac.jp/calendar/view.php?view=upcoming&course=1',
options);
```

これでようやくカレンダーのソースページを取得できました。

2. 取得したソースを元に必要な情報を抽出し、整理する

必要な情報は先ほども登場した cheerio で取得していきます。抽出する情報はイベントの概要、説明、期限、教科名、URL などがあります。また Google カレンダーに自動追加を行うときにイベントの重複確認を行う必要があるため、運よくソース内にあった ID も抽出しておきます。抽出が終わったら、Google カレンダーにきれいに落とし込めるように、抽出した情報を結合などして、最終的にはタイトル、時間、色、説明といった感じに整理します。このように分けた理由としては次の節で後述しています。

3. Google カレンダーに追加する

Google カレンダーに追加する前に重複確認を行っておきます。Google カレンダーのタイトルを取得し、タイトル内にある ID と Moodle から取得した ID を比較し、同じであればスキップ、そうでなければ追加していきます。次に、先ほど整理したイベントの情報を以下のコードのように書いていきます。これを実行することで、右の図のように Google カレンダーとして追加できました。あとは GAS のトリガーで一日一回、更新する時間を指定しておくことで今後自動的に提出物のイベントが追加されます。

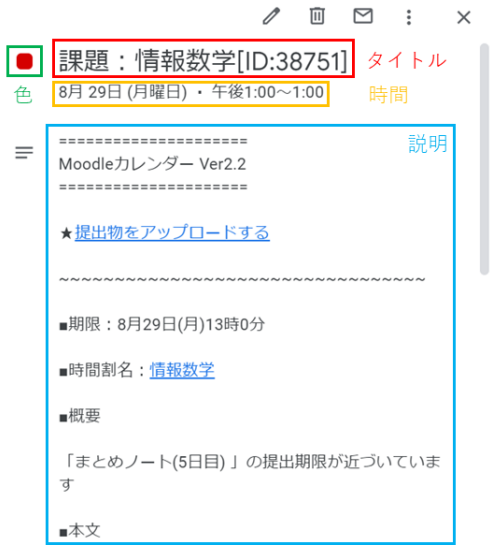
```
// タイトルを指定
let title = titles + '[ID:' + id + ']';

// 時間を指定
let startTime;
if (color === 2) {
  startTime = new Date((due - 5400) * 1000);
} else {
  startTime = new Date(due * 1000);
}
let endTime = new Date(due * 1000);

// イベントの説明
let options = {description:content}

// イベントを追加
let event = calendar.createEvent(title, startTime, endTime, options);

// 追加したイベントの色を指定
event.setColor(color);
```



最後に

今回作ったカレンダー自動化は、便利すぎて非常に満足しています。Google カレンダーはスマホにもインストールでき、締め切りの 1 時間前に通知が来たり、今後の予定を一目で確認したりできるため、このシステムを開発してから提出物の出し忘れがありません。欠点としては導入方法が少しめんどくさいため、今後はだれでもできるように、アプリ化して簡単に導入できるようにしていきたいです。

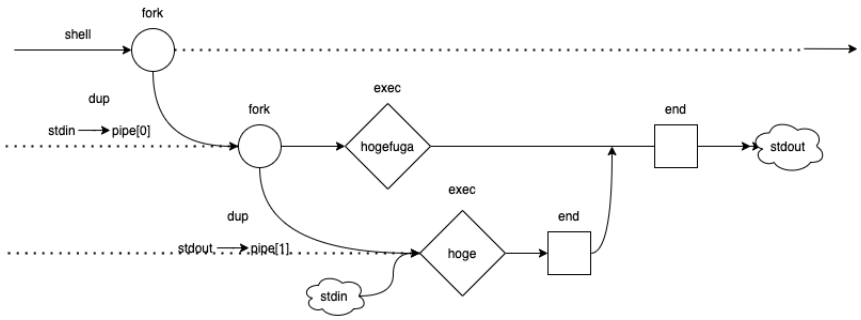
参考文献

https://developers.google.com/calendar/api/v3/reference

自作シェル「sijimi」を作ってみた

URL：https://zenn.dev/higuruchi/articles/142b613eb1e650

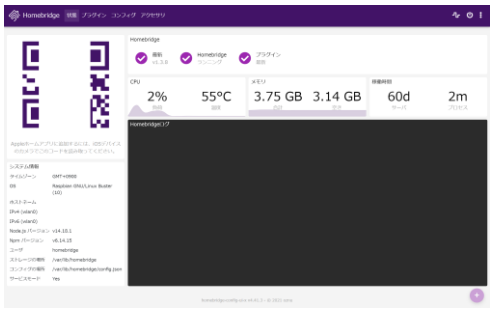
概要：sijimiに実装されている機能である、コマンド、パイプ、リダイレクトの説明。

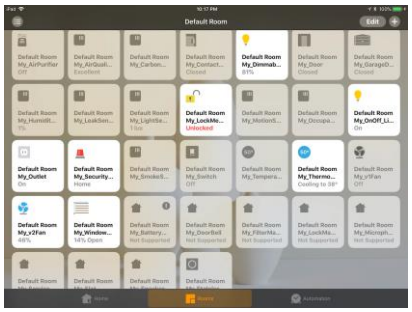


スマホで家電操作

URL：https://qiita.com/Gomatamago_/items/8585dfdd38a93e41cecf

概要：HomeBridgeとスマートリモコンを使って家の家電を操作する方法。





rubyでアプリつくる

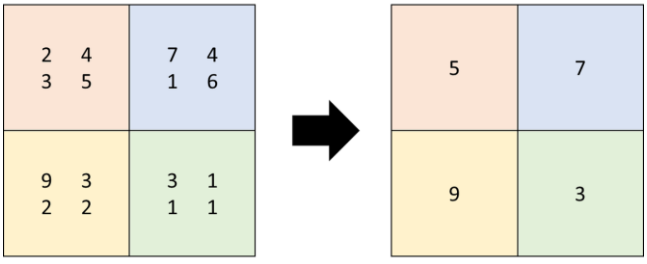
URL：https://qiita.com/hishibiro/private/2c6a9e891ea0c750f9b0

概要：Ruby on Railsの設定について説明。

リア充を学習

URL：https://cd-donki-gorira.hatenablog.com/entry/2021/12/04/194637?_ga=2.38574053.751180675.1638610243-359943590.1630585542

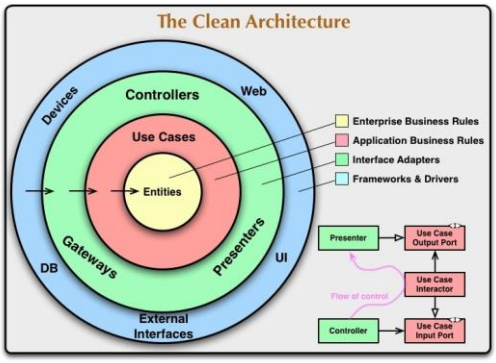
概要：画像解析の勉強がてら、リア充の学習してみた。

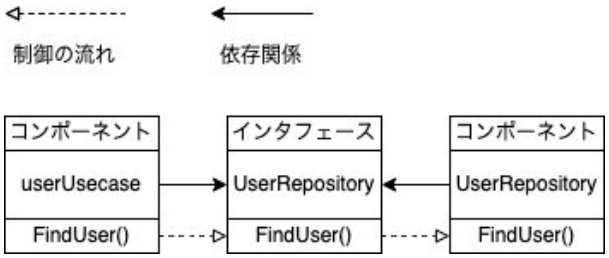


クリーンアーキテクチャと依存関係逆転の法則

URL：https://zenn.dev/higuruchi/articles/58a766398b9336

概要：クリーンアーキテクチャと依存関係逆転の法則について説明。

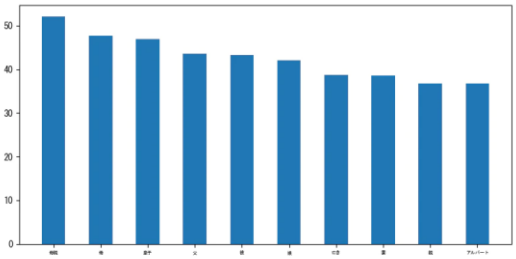


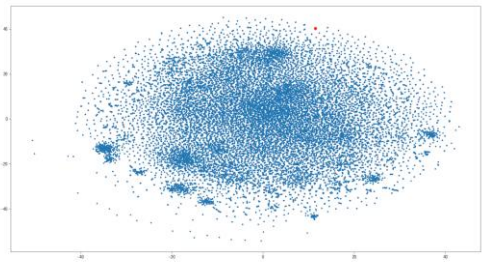


word2vec実装

URL：https://qiita.com/g75hca/items/507a557f10d6133a699a

概要：Word2vecによる分散表現を可視化。





ブログ開設しました

URL：https://blog.yamakenji.com/blog/first-post

概要：Remixを用いて、自身のブログを構築する。

```ts
export function getAllTags(): string[] {
  const blogs = getAllBlogs();
  const tags = blogs.map((blog) => blog.tags).flat();
  return Array.from(new Set(tags));
}

export function getAllCategories(): string[] {
  const blogs = getAllBlogs();
  const categories = blogs.map((blog) => blog.category);
  return Array.from(new Set(categories));
}
```

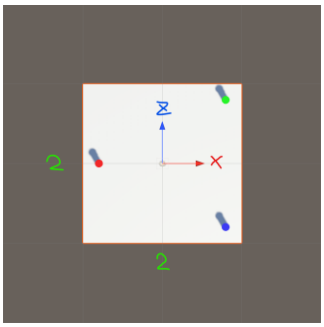
```ts
export const loader: LoaderFunction = async (content: any) => {
  const category = content.params.slug;
  const blogs = await getBlogsByCategory(category);
  const data: LoaderData = { blogs, category };

  return json(data, {
    headers: {
      'Cache-Control': 'public, max-age=60 s-maxage=60',
    },
  });
};
```

Quest2で移動した軌跡を取りました

URL：https://cnaan.hatenablog.com/entry/2021/12/10/070000

概要：VR空間で歩いた軌跡のログをとって見てみる。

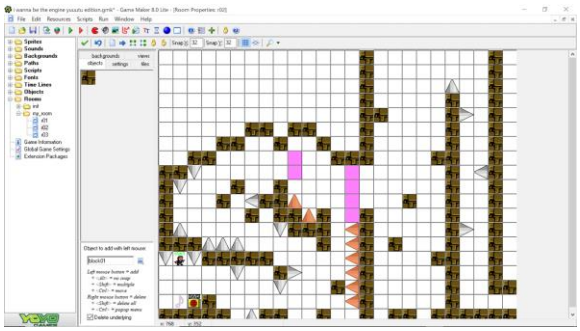


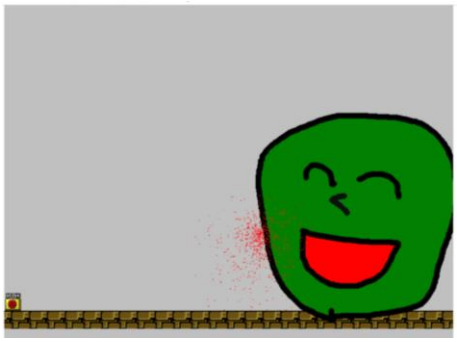
	A	B	C	D	E	F	G	H
1	time	position.x	position.y	position.z	forward.x	forward.y	forward.z	
2	2021/12/10 1:28:21	0	0	0	0	0	1	
3	2021/12/10 1:28:23	0	1.01	2.24E-17	0.2826	-0.32046	0.904125	
4	2021/12/10 1:28:25	0	1.01	2.24E-17	0.171217	0.053636	0.983772	
5	2021/12/10 1:28:27	0	1.01	2.24E-17	0.089512	-0.01999	0.995785	
6	2021/12/10 1:28:29	0	1.01	2.24E-17	0.042343	-0.16633	0.98516	
7	2021/12/10 1:28:31	1.85462	1.01	0.767159	0.679065	0.048145	0.732497	
8	2021/12/10 1:28:33	6.855697	1.01	6.990535	0.637785	0.055431	0.768217	
9	2021/12/10 1:28:35	3.863449	1.01	7.815657	-0.99323	0.06281	-0.09773	
10	2021/12/10 1:28:37	-2.41715	1.01	7.131356	-0.68577	0.046085	-0.72636	
11	2021/12/10 1:28:39	-6.38219	1.01	2.022796	-0.68829	0.062602	-0.72273	
12	2021/12/10 1:28:41	-6.19519	1.01	-4.84214	0.73847	0.062452	-0.67139	
13	2021/12/10 1:28:43	-0.56586	1.01	-6.92826	0.66638	0.062374	0.742999	
14	2021/12/10 1:28:45	6.276159	1.01	-8.10253	0.988202	0.126646	0.086127	
15	2021/12/10 1:28:47	6.617395	1.01	-8.11422	-0.18202	0.162468	0.969781	
16								

ゲームエンジンとプログラミングでゲーム作成

URL：https://qiita.com/asn407/items/329345e5c943b149d2d2

概要：プログラミングでアイワナを作ろう！

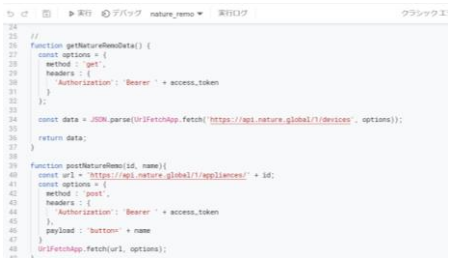


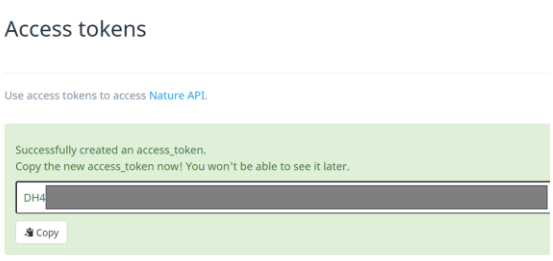


GASでNatureRemoのデータ取得や信号の送信

URL：https://qiita.com/tsukasa0220/private/0b0ec63b9e26614d4f3a

概要：GASを使ってNatureRemoのAPIにアクセスし、センサー情報の取得や赤外線信号の送信。

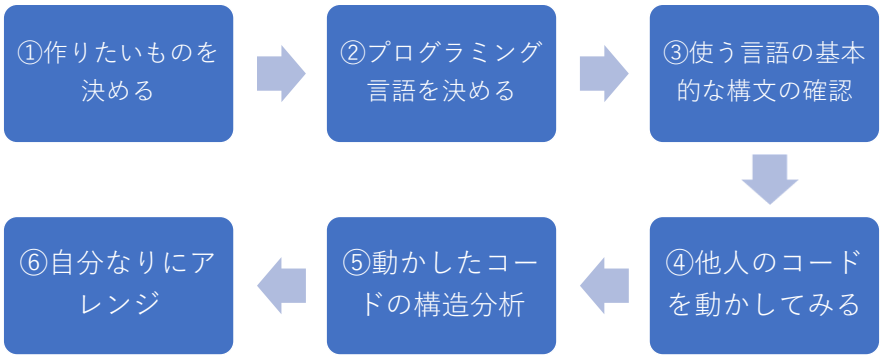




プログラミングのおすすめの勉強法

URL：https://qiita.com/g75hca/items/eaa08ebbc0b1af896396

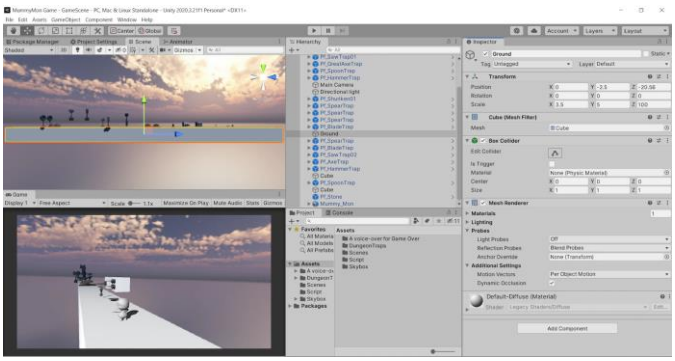
概要： プログラミングのおすすめの勉強法について説明。



Unityで3Dゲームを作りました

URL：https://qiita.com/tus18/items/e8e3c61a096404cab5a6

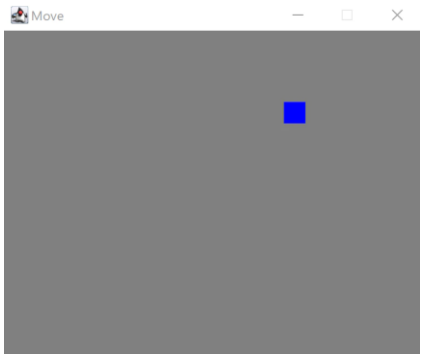
概要：Unityで、障害物をよける簡単な３Dゲームの作成方法。

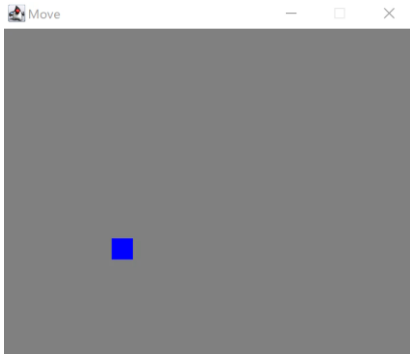


キーボード入力で動く四角形

URL：https://qiita.com/yuukichi1397/items/eeab4a6b3ebd2ebacf3c

概要：Javaを使って画面に映した図形をキーボードからの入力で動かす処理を作成。





WordPressで役立つHTMLタグ

URL：https://qiita.com/sujimi/items/984fdb763280a6dc71bf

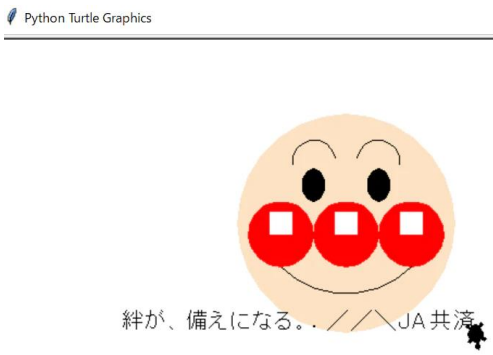
概要：WordPressでよく使うHTMLタグについてまとめてみた。

タグ	意味	属性	意味
meta	その文書に関する情報（メタ情報）を指定 必ず〜間に記述	name属性	メタデータの名前を指定
		content属性	メタデータの値などを指定
		charset属性	HTML文書の文字エンコーディングを指定
title	その文書に文書にタイトルを付ける		
link	関連する文書ファイルを指定する	href属性	リンクする外部リソースのURL（必須属性）
		rel属性	現在のファイルとリンク先の外部ファイルとの関係性を表すキーワード（必須属性）

機械学習で動物の判別

URL：https://qiita.com/nagotta/items/c79adf6a95c9b9110121

概要：pythonの標準ライブラリであるturtleを使用してイラストを描画する。



簡単な対話botを作りました

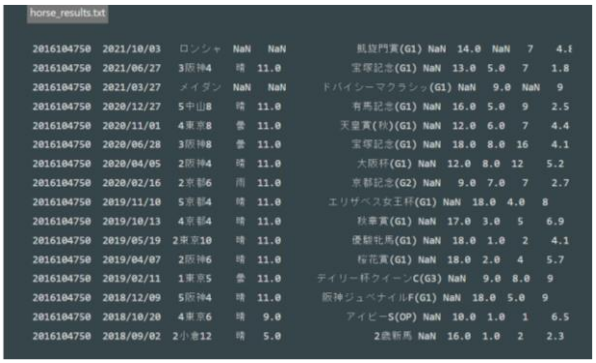
URL：https://qiita.com/tom140425/items/5a118407d930e0a7406d

概要：Python学習のチュートリアルとして簡単な対話botを作成。

有馬記念を全力で予想する

URL：https://qiita.com/SSK_Ayagawa/private/952c2c4ace101d8aa374

概要：Pythonを用いて、2021年有馬記念を予想する。

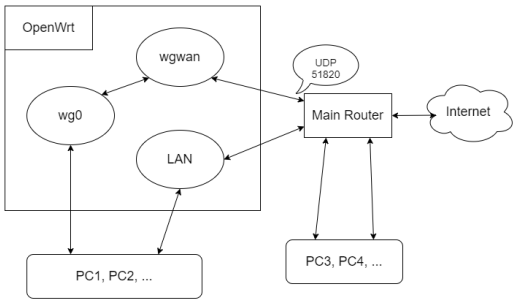

horse_results.txt

2016104750	2021/10/03	ロンシャ	NaN	NaN	紙旋門賞(G1)	NaN	14.0	NaN	7	4.1
2016104750	2021/06/27	3阪神4	晴	11.0	宝塚記念(G1)	NaN	13.0	5.0	7	1.8
2016104750	2021/03/27	メイダン	NaN	NaN	ドバイシーマクラシック(G1)	NaN	9.0	NaN	9	
2016104750	2020/12/27	5中山8	晴	11.0	有馬記念(G1)	NaN	16.0	5.0	9	2.5
2016104750	2020/11/01	4東京8	曇	11.0	天皇賞(秋)(G1)	NaN	12.0	6.0	7	4.4
2016104750	2020/06/28	3阪神8	曇	11.0	宝塚記念(G1)	NaN	18.0	8.0	16	4.1
2016104750	2020/04/05	2阪神4	晴	11.0	大阪杯(G1)	NaN	12.0	8.0	12	5.2
2016104750	2020/02/16	2京都6	雨	11.0	京都記念(G2)	NaN	9.0	7.0	7	2.7
2016104750	2019/11/10	5京都4	晴	11.0	エリザベス女王杯(G1)	NaN	18.0	4.0	8	
2016104750	2019/10/13	4京都4	晴	11.0	秋華賞(G1)	NaN	17.0	3.0	5	6.9
2016104750	2019/05/19	2東京10	晴	11.0	優駿牝馬(G1)	NaN	18.0	1.0	2	4.1
2016104750	2019/04/07	2阪神6	晴	11.0	桜花賞(G1)	NaN	18.0	2.0	4	5.7
2016104750	2019/02/11	1東京5	曇	11.0	デイリー杯クイーンC(G3)	NaN	9.0	8.0	9	
2016104750	2018/12/09	5阪神4	晴	11.0	阪神ジュベナイルF(G1)	NaN	18.0	5.0	9	
2016104750	2018/10/20	4東京6	晴	9.0	アイビーS(OP)	NaN	10.0	1.0	1	6.5
2016104750	2018/09/02	2小倉12	晴	5.0	2歳新馬	NaN	16.0	1.0	2	2.3

Rustでシェル書く

URL：https://yassi.hatenablog.com/entry/2021/12/25/011106

概要：OpenWrtをスイッチ化してWireGuardを入れた。



指パッチンカメラを作ってみた

URL：https://qiita.com/Gomatamago_/items/afc908e22a227e765f77

概要：Swiftを用いて、指パッチンでシャッターが切れるカメラを作成。