Teach Yourself KMC

独習KMC vol.15

京大マイコンクラブ著

目次

KMC の最近の活動 **1**

新入生プロジェクト **1**
お絵かき練習プロジェクト 2019 (ykn510) 1
競技プログラミング練習会 Normal2019 (laft) 1

開発プロジェクトやその他勉強会 **3**
ガベコレを作る (hatsusato) 3
きっと弾幕ゲー (kitsan) 4
チョークを投げるゲーム (hashi210) 4

部員のコラム **6**

Slack のスラッシュコマンドをつくる (utgwkk) **6**
ごあいさつ 6
スラッシュコマンドについて 6
今回作るスラッシュコマンド 6
作る前に、スラッシュコマンドの仕組み 7
先に完成したもの 7
完成するまでの道のり 7
Slack Bot との比較 9
いかがでしたか 9
おまけ：正規表現にマッチする文字列を生成する仕組み 10

あとがき **11**

あとがき **11**

新入生プロジェクト

お絵かき練習プロジェクト 2019

ykn510

こんにちは、お絵かき練習プロジェクト 2019 担当の ykn510 です。

KMC ではお絵かき練習プロジェクトと称し、初心者から経験者まであつまってデジタルイラストの技術を学んだり実際に手を動かして互いにアドバイスし合ったりしています。

後期はいろんな上回生を特別講師としてゲストに呼んで、前期よりも一層踏み込んだ理論の話をしていただきました。僕だけでは話しきれないような上級テクニックなども教えていただくことができ、さらなるスキルアップができたと思います。

春休みはプロジェクトの延長として、地力をつけることに重点をおいてデッサンやクロッキーなどをしてきました。こういった練習は一人でやろうとするとかなり心理的ハードルが高い反面、継続的にやることが何より大事なので、これからも部員全員で機会を共有していけると良いですね。

来年度もお絵かき練習プロジェクトは引き続き開かれる予定ですので、イラストを描いてみたい、もっと上手くなってみたいという人は是非 KMC に足を運んでみてください。

競技プログラミング練習会 Normal2019

laft

競技プログラミング練習会 Normal2019 を担当している KMC2 回生の laft です。

前期から引き続き、同期の aotsuki と二人で担当しています。

このプロジェクトでは、プログラミング未経験者を含む競技プログラミング初心者・中級者に向けて競技プログラミングに頻出のアルゴリズムやデータ構造を紹介して、その後にバーチャルコンテストを開催し、学んだばかりのアルゴリズムやデータ構造を用いて実際に問題を解くということを行っています。

知らない人向けに簡単に説明しておくと、競技プログラミングとはプログラミングによって問題を解く競技のことです。「AtCoder」というサイトが国内では最も有名で、オンラインで簡単に始められるのでよければ調べてみてください。

前期では私と aotsuki が参加者に教える形で前半の講座パートを行っていましたが、後期では参加者にも同じように自分で興味のあるアルゴリズムについて調べ、前で講座をしてもらいました。

講座をしようと思うと、教えるアルゴリズムの本質的な部分をきちんと理解しておく必要があり、講座の準備だけでもかなり勉強になります。参加者にとっても良い経験になったのではないでしょうか。

来年度は競技プログラミング練習会 Normal2020 を行います。来年度は私の担当ではありませんが、私の後輩が初心者向けに手厚く教えてくれると思いますので、興味がある方はぜひ参加してみてください。

競技プログラミング経験者向けに競技プログラミング練習会 Advanced2020 も同時に行いますので、経験者の方も奮ってご参加ください。

開発プロジェクトやその他勉強会

ガベコレを作る

hatsusato

発起人の hatsusato です。このプロジェクトは『ガベージコレクション』（翔泳社）[*1] という本を輪読して、さまざまなガベージコレクション（以下 GC）アルゴリズムを学ぶ勉強会になります。GC を扱った日本語の書籍として、他に『ガベージコレクションのアルゴリズムと実装』（秀和システム）がありますが、あちらは基本のアルゴリズムと具体的な実装から構成されるのに対して、本書はより専門的なアルゴリズムを多く含んでおり、GC を研究しようとする大学院生が読むような密度となっています[*2]。本書の訳者の一人が KMC OB であるという縁もあり、寄贈してもらった本も使って勉強会を進めています。

プロジェクトは 2019 年 4 月から始まり、月に 1 度集まって担当者が担当分の本の内容のまとめを発表する輪読形式で読み進めています。担当者は読んだ内容のまとめを Scrapbox[*3] に共有しておき、当日の発表で分かりづらかった部分を参加者全員で Scrapbox 上でリアルタイムに編集しながら理解を深めています。参加者は全員 B4 以上という比較的年齢層の高いメンツですが、毎回和気藹々と楽しく読み進めています。

これまでに、マークスイープ GC・コピー GC・参照カウント GC・世代別 GC をベースとして、さまざまな亜種・変種の GC アルゴリズムも勉強してきました。これからまもなく並行・並列 GC アルゴリズムを扱う章に突入するので、今からますます先を読むのが楽しみになっています。プロジェクト開始当初は毎回 1 章のペースで進んでいましたが、最近は 1 章あたりの分量が多くなってペースが落ちてきているので、このままでは 2 年で読み終われないのではないかという危惧もありますが[*4]。プロジェクト名が表すように、本来は GC アルゴリズムを勉強してそれを実装することを目標にしていましたが、参加者がみな忙しいためになかなか実装まで手が回っていません。それでも、いずれはこのプロジェクトで学んださまざまな GC アルゴリズムを実装してみたいと考えています。

[*1] 副題：自動的メモリ管理を構成する理論と実装。URL: `https://www.seshop.com/product/detail/20755`, ISBN: 9784798147413.

[*2] 残念ながらそれほど分かり易いわけではないので、原著や引用論文を参照する覚悟が必要かもしれません。

[*3] こちら `https://scrapbox.io/garbage-collection/` からまとめを閲覧することができます。

[*4] 2 年で読み終われないと参加者の多くが卒業してしまいます。

きっと弾幕ゲー

kitsan

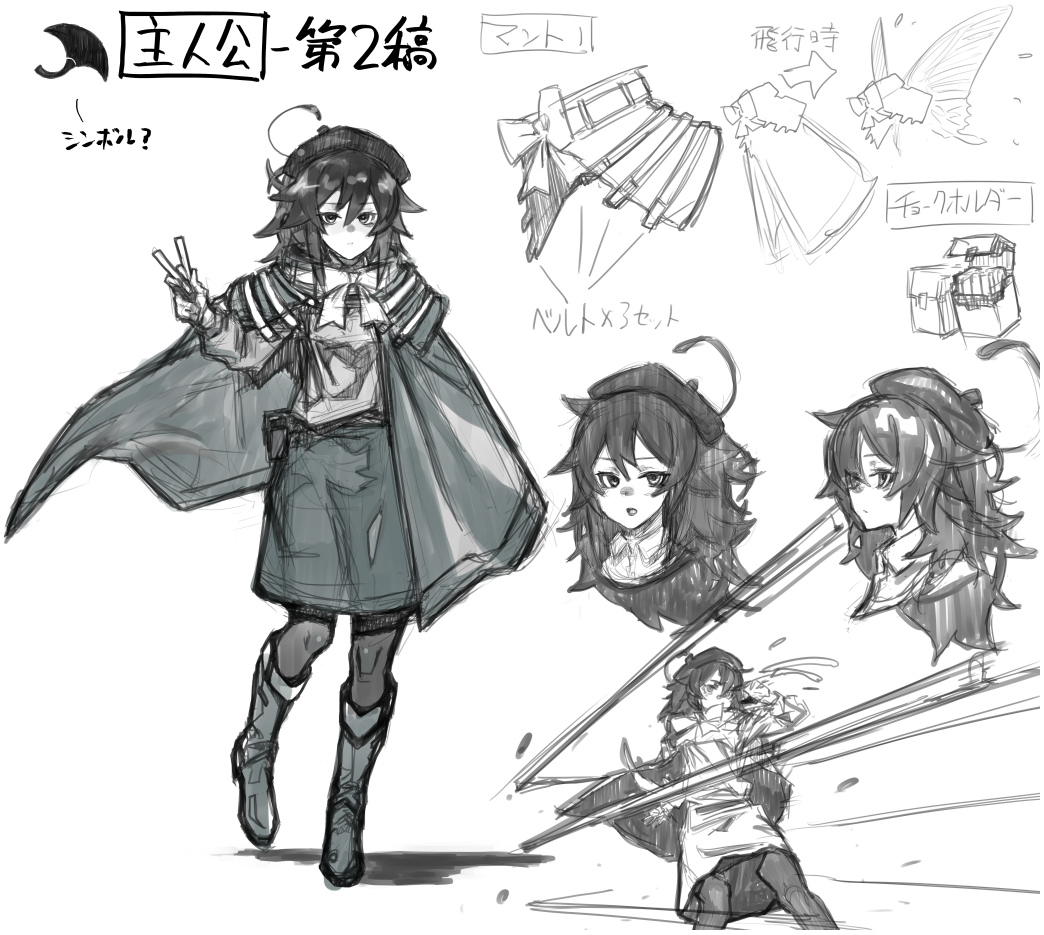
こんにちは。このプロジェクトの担当者の kitsan です。このプロジェクトは名前通り「必ず弾幕ゲーを完成させるぞ！」という強い意志を持って作られたゲーム作成プロジェクトです。

現在私たちは今年 10 月リリース予定のゲーム「東方蒼月闘」の完成に向けて日々取り組んでいます。このゲームは東方 Project の二次創作ゲームで、ボス側をプレイして弾幕を張るという新しいシステムを導入した対戦型弾幕ゲームです。数ある弾幕パターンを組み合わせて自機を追い詰めていくという、戦略性の高い試合が楽しめます。また、魅力的なキャラクターたちのストーリーモードなども実装予定です。今年 10 月の東方紅楼夢、Steam でのリリースまでぜひお待ちください。

チョークを投げるゲーム

hashi210

ディレクターの hashi210 です。このプロジェクトでは、10 人程度のメンバーが集まってゲームを制作しています。ゲームのジャンルは 2D 横スクロールアクションで、Unity を用いて開発しています。昨年の 12 月から活動を開始し、今年の 2 月から春休みを使って本格的に制作を進めてきました。キャラクターとアクションのカッコよさに魅力になるようなゲームを目指しています。C99 や今年の京大の 11 月祭では体験版が出せるようにしたいですね。続報を待て！



主人公のデザインアートワーク

Slack のスラッシュコマンドをつくる

utgwkk

ごあいさつ

こんにちは、KMC 6 回生の utgwkk です。普段はお酒を飲んで暮らしています。

本日はみなさまに、Slack のスラッシュコマンドの作り方についてお伝えします。もし、Slack についてご存知ないのであれば、一度公式サイトをごらんください[*1]。

スラッシュコマンドについて

スラッシュコマンド (Slash Commands) とは、その名の通り/（スラッシュ）で始まるコマンドによって、いろいろな機能を使うことができる、という Slack の機能です。Slack を使ったことがあるのであれば、 /remind や /leave のようなコマンドを使ったことがあると思います。これらのコマンドのようなものを自作することができる、と思ってもらえればひとまず大丈夫です。

スラッシュコマンドの例としては、以下のようなものがあります。

/github　GitHub のリポジトリへの commit や push などの通知設定を変える

/giphy　GIPHY[*2]という Web サービスで、指定された語句で検索して出てきた GIF アニメーション画像を投稿する

今回作るスラッシュコマンド

今回は、図 1 のような、与えられた正規表現にマッチする文字列をランダムに生成して投稿するスラッシュコマンドを作ろうと思います。

[*1] `https://slack.com/intl/ja-jp/`

[*2] `https://giphy.com/`

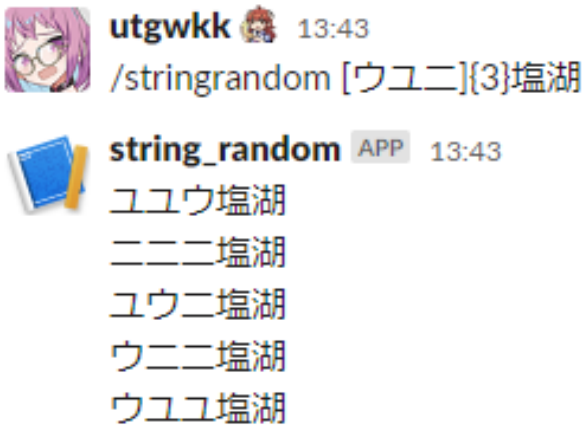


図 1: /stringrandom スラッシュコマンドの例

作る前に、スラッシュコマンドの仕組み

本質だけを抽出すると、スラッシュコマンドを実行すると、スラッシュコマンドごとに指定されたサーバーに対して、Slack のサーバーから HTTP POST リクエストが飛ばされる、という仕組みになっています。つまり、Web アプリケーションを実装したことがあるのであれば、もうスラッシュコマンドを作ることができる、と言ってよいです。

先に完成したもの

完成したものがこちらになります。Python で書いてあります。Web アプリケーションフレームワークとして Flask を使いました。わたしが開発速度重視で Web サービスを作るときは Python と Flask の組み合わせを使うことが多いです。

`https://github.com/utgwkk/slack-string-random-slash-command`

完成するまでの道のり

スラッシュコマンドを動かすサーバーを用意する

スラッシュコマンドは Web アプリケーションと同じ、ということは、スラッシュコマンドを動かすためのサーバーが必要です。適当に VPS を借りたり Heroku を使ったりしましょう。今回は無料で動くことに目をつけて Heroku を使うことにします[*3]。

[*3] ちなみに無料枠の範囲だと 24 時間サーバーを稼動させることができません。

アプリケーションを登録する

`https://api.slack.com/apps` で、これから作るスラッシュコマンドアプリケーションを登録します。名前はいい感じに決めましょう。今回は string random という名前で、開発に使うワークスペースとして KMC のものを使うことにします。

次に、どういう機能を作るつもりか聞かれるので、Slash Commands を選びます。続いて、図 2 のように、どのようなスラッシュコマンドを作るか聞かれるのでここも入力していきます。Request URL には、用意したサーバーの URL を入れます。

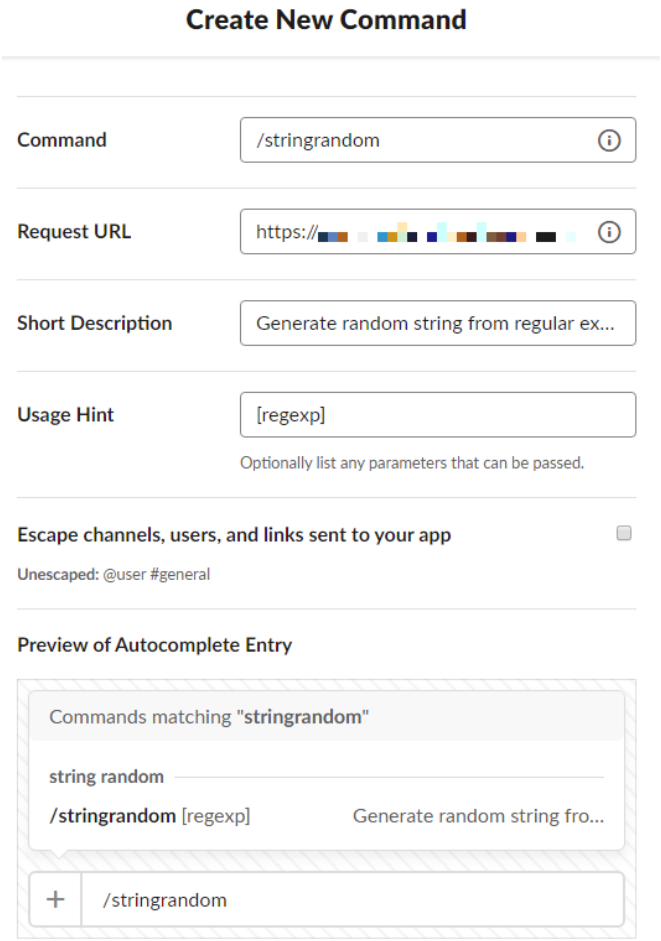


図 2: Create New Command

ここまでできたら、Install your app to your workspace から、スラッシュコマンドを開発用ワークスペースで有効化してみましょう。これでスラッシュコマンドが実行できるようになり

ます。

レスポンスを返す

/stringrandom というエンドポイントに POST リクエストが来たら、リクエストの text パラメーターから正規表現を受け取ります。正規表現からランダムな文字列を 5 つ生成したら、生成した文字列をレスポンスに詰めて返します。

ついに完成

完成したので遊びまくります。

Slack Bot との比較

スラッシュコマンドと Slack Bot は、どちらも発言をすると何らかの機能が使える、という点で似ているのですが、利点は以下のようなところにあると思います。

Web アプリケーションとして実装できる　Slack Bot は、Slack が提供する API に沿って実装したプログラムを常時起動する、という雰囲気になります。一方でスラッシュコマンドは、Web アプリケーションそのものといってよいので、Web 系の開発に使えるライブラリがそのまま流用できます。

常時起動しなくてよい　Slack Bot が常に発言を受け付けて反応を返すには、Slack Bot を常時起動しておく必要があります。これに対してスラッシュコマンドでは、Slack からの HTTP リクエストに対してレスポンスを返すことさえできればよいので、Web サーバーを常時起動しておく必要はありません。たとえば、AWS の API Gateway + Lambda や GCP の Cloud Functions のような FaaS[*4]を使い、サーバーを用意することなくスラッシュコマンドを実装することができます。

Bot ユーザーを invite せずに使える　Slack Bot に反応してほしい場合、チャンネルに Bot ユーザーがいる必要があります。スラッシュコマンドはチャンネルに関係なく、発言しさえすれば使えます。

いかがでしたか

スラッシュコマンドは思ったよりも簡単に実装できて、Web サービス開発の経験がそのまま活かせるのでおすすめです。おもしろいスラッシュコマンドを作ってみんなで遊びましょう。

[*4] Function as a Service

おまけ：正規表現にマッチする文字列を生成する仕組み

この節はスラッシュコマンドとまったく関係ありません。

正規表現の用途といえば、ふつうは文字列とマッチするかどうか判定するといったものが思い浮かべられると思います。/stringrandom では逆に、与えられた正規表現にマッチする文字列を生成しています。これはどのような仕組みになっているのでしょう。

正規表現にマッチする文字列をランダムに生成するために、Xeger[*5]というライブラリを使いました。このライブラリでは、まず正規表現を表す文字列をパースして、Python の正規表現エンジンが解釈できるようにコンパイルしています。コンパイルされると、プログラミング言語の抽象構文木のように、正規表現の構文木のような構造に変換されます。ちょうど抽象構文木をインタプリタが解釈するように、正規表現の構文木をなめつつランダムな文字列を生成しています。

たとえば、`[ウユニ]{3} 塩湖` のような正規表現の変換先を Lisp の S 式のように表すと以下のようになります。

```
((repeat (choice (list ウ ユ ニ)) 3) . (string 塩湖))
```

`repeat` が来たら、ウ・ユ・ニのいずれかの文字をランダムにひとつ取ることを 3 回繰り返し、「塩湖」と結合する、のように読めますね。

[*5] https://github.com/crdoconnor/xeger

あとがき

■**utgwkk**　携帯動画変換君で変換した動画を PSP で見まくっていたことを思い出しました。

■**polaris**　実は記事を書くつもりだった……。

■**nona7**　記事を書かないのに校正校閲がうるさい。

独習 KMC vol.15

2020 年 12 月吉日　初版発行

著作・発行　京大マイコンクラブ
表紙&裏表紙デザイン　murayama

メールアドレス　info@kmc.gr.jp
Web　https://www.kmc.gr.jp/

落丁・乱丁の際は在庫がある限りお取り替えいたします。上記のメールアドレスまでご連絡ください。

独習KMC
vol.15

■利用分類
部誌
■製品分類
KMC
■レベル
初級　中級　上級

Kyoto University Micro Computer Club

https://www.kmc.gr.jp