

command line による editor 操作の習熟プログラムの開発

情報科学科 27014533 和田創熙

1 研究の動機

最初は西谷によって開発された shunkuntype(ターミナル上で実行するタイピングソフト)の再開発をテーマにしていたが、タイピングに特化したソフトを開発しても同じようなものが Web 上に大量に配布されており、それ以外¹の付加価値を付けた²ソフトを開発しようと考えた。特に editor 操作に関しては

ツールはプログラマ自身の手の延長である。これは他のどのようなソフトウェアツールよりも Editor に対して当てはまる。テキストはプログラミングにおける最も基本的な生素材なので、できる限り簡単に操作できる必要があり、一つの強力な editor の習熟は作業の効率化に欠かせない [1]。

と記述されている。さらに、西谷研究室ではタイピング、Ruby 言語、Emacs による editor 操作、CUI 操作の習熟が作業効率に非常に大きな影響を与えするため習熟を勧めている。そこでこれらの習熟を目的としたソフトを開発しようと考えた。

2 editor_learner の概要

ソースコード³は Ruby 入門の教科書を参考になっている [2]。

2.1 initialize

editor_learner を動作させた時、最初に自動的に動く部分である。基本的に作業を行うためのファイルの作成、gem で install した editor_learner のパスを格納したインスタンス変数の作成がメインである。

2.2 random_check

開始から終了までの動作の概要は以下の通り、

- 用意されたソースコードから 1 つ選ばれ、question.rb にコピーされ、新しいターミナルが開かれる。
- そこで question.rb の内容を answer.rb に写経する。
- 写経し終わると前のターミナルに戻り”check”とコマンドラインで入力する。
- 正しければ終了、正しくなければ間違った箇所のみが表示され、再度確認、入力を行い正誤判定の繰り返しをする。

2.3 sequential_check

開始から終了までの動作の概要は以下の通り、

- chapter と問題番号を引数として入力。その問題が q.rb

にコピーされる。

- その後新しいターミナルが開かれる。
- 写経し終わった後の動作は 2.2 の 4 番からは random_check と同じ動作である。

3 他のソフトとの比較

表 1 人気タイピングソフトと editor_learner の利点と欠点とユーザーインターフェース。

	利点	欠点	UI
editor_learner	editor操作やキーバインドなどの習熟が可能	間違った箇所がわかりにくい。	CUI
PTYPING	豊富なプログラム単語の入力可能	プログラムの実行、プログラムの入力が不可能。	CUI
e-typing	間違いの多かった箇所などがわかる。	同じ内容のものが多くて慣れてしまう。	GUI
寿司打	GUIにより、モチベーションが継続されやすい。	ローマ字入力のみなので約物の入力が一切ない。	GUI

表 1 は人気タイピングソフトと editor_learner の利点と欠点とユーザーインターフェースを表にしたものである。

CUI に近い PTYPING が人気の理由はコードに関する単語を入力できるという点であった。それに加えて editor_learner は editor 操作やコードの入力、実行まで可能になっている。

GUI ベースのソフトには継続性に関しては劣る。しかし CUI ベースではキーバインド使用により作業の高速化、効率化が期待される。editor_learner はプログラマ向けの機能を多数有しているソフトであると言える。

4 終わりに

editor_learner は {} や () など約物の入力やカーソル移動、ファイルの開閉、保存などの CUI 操作を全てキーバインドで行う。その結果、editor_learner がいかにプログラマ向けのソフトであり、ソフトの目的に沿った技術の向上が期待される。

参考文献

- [1] Andrew Hunt, David Thomas, 「達人プログラマー」, (オーム社, 2016 年)。
- [2] 伊藤淳一, 「プロを目指す人のための Ruby 入門」, (技術評論社, 2017 年)