

## シェルコマンド学習支援ゲームの開発

佐藤将真†

西田誠幸†

† 拓殖大学工学部情報工学科

### 1 はじめに

Unix や Linux の端末におけるシェルコマンドラインインタフェースは、情報系の学生や IT 技術者が、計算機上でファイル操作するための基盤となるツールの 1 つである。この学習を支援するソフトウェアの 1 つに、ゲーム Terminus[1] がある。このゲームは、シェルコマンド初学者を対象にファイル操作を行うシェルコマンドを紹介し、コマンドの基本的な使い方についての学習を支援する。しかし、学習者が使い慣れたエクスプローラや Finder などの GUI 環境を離れて CUI 環境を活用ようになるには、シェルコマンドの便利さを実感してもらうことやコマンド入力を訓練する場を提供するなどの別の学習支援も必要である。

本稿では、シェルコマンドを学んだ経験がある学習者を対象に我々が開発した、CUI 環境の活用を促すためのシェルコマンド学習支援ゲームについて述べる。

### 2 問題の分析と開発方針

学習支援ゲームを開発するにあたってファイル操作のための GUI 環境と CUI 環境について比較、考察したことを述べる。

Windows や Mac OS などの普及とコンピュータリテラシ教育により、エクスプローラや Finder を通じた GUI によるファイル操作に習熟した人が多くいる。これらのツールではマウスなどのポインタデバイスを用いており、ファイル操作の一部はアイコンのドラッグやポップアップメニューの選択などにより実行可能である。これに対して CUI 環境では主にキーボードによる文字入力が必要で、すべてのファイル操作はコマンドにより実行されるため、コマンドを記憶しておくことが求められる。また、ファイル操作の結果をすばやく得るには、キーボードによる文字入力の手速も求められる。逆に CUI 環境の利点の 1 つとして、同じようなファイル処理を繰り返したり、パイプライン等によりコマンドを組み合わせることが可能であり、GUI 環境では少ないステップで実現するのが難しい高度なファイル操作を行うのに適している。

考察を受けて、次の開発方針に基づいてゲームを開発することとした。すなわち、

1. ファイル操作を行う時間に制限を設ける
2. 基本的なファイル操作を要求する問題に加えて、コマンドの繰り返しや組み合わせを要求するような比較的難しい問題も用意する。
3. Unix や Linux の標準的な端末エミュレータとシェルを用いる。

補足説明をする。1 の制限時間を設けることで、ゲームクリアのために、問題を見てからコマンドを探すのではなく、学習者にコマンドを記憶しておくことを求める。また、コマンド入力の手速向上を促す。2 の比較的難しい問題を比較的短い時間でクリアすることを通じて、学習者に対して GUI 環境にはない CUI 環境の利点を知ってもらうことを狙う。

### 3 シェルコマンド学習支援ゲーム

#### 3.1 本ゲームの設定

仮想的なハッカーが学習者の持つ実際のディレクトリとファイルに対して、改変や削除などの攻撃を行うという設定をとる。繰り返し行われるハッカーの攻撃に対して、学習者はコマンド操作で設定時間内にすばやくファイルを修復することが求められる。

#### 3.2 ゲームの流れ

ゲームは複数のステージで構成される。1 つのステージは仮想ハッカーから提示される問題文と画像、制限時間、クリア判定ルーチンで構成される。ゲームプログラムの処理と合わせて、ゲームの進行について説明する。

1. 学習者のホームディレクトリのファイルを、ゲーム用ディレクトリとして複製する
2. 複数のステージの中から一つを選択する
3. ゲーム用ディレクトリ内のファイルに対してステージに登録されたハッカーの攻撃を行う

Learning Support of Shell Commands by a Game

†Shoma SATO, Seikoh NISHITA

†Dept. of Engineering, Takushoku University  
193-0985, Hachioji, Japan

4. ウィンドウ上にハッカーの画像と問題文，制限時間を表示する
5. 定期的にディレクトリを監視して，ステージクリア，ゲームオーバー，プレイ中の状態を判定する
6. 学習者のファイル操作によってクリア条件を満たしたらステージクリアとなり，2 の処理に戻る

すべてのステージをクリアする、もしくは制限時間を超えてもクリア条件を満たさなければゲーム終了とする。図1にゲームプログラムの処理を示す。また、問題文と正解のコマンドの例を図2に示す。

ステージを設計するには学習者に使用させたいコマンドを選ぶ必要があるが、これらのコマンドはLinux標準教科書 [2] の2章と3章から選択した。

### 3.3 ステージクリア判定

本ゲームの各ステージにおけるクリア判定は、ゲーム用ディレクトリのファイル構成を検査することにより実現している。従って各ステージの問題文では学習者に対して、最終的にファイル構成を更新するような処理を指示する必要がある、ステージによっては不自然な問題となってしまう。また、各ステージで想定する正解のコマンド列以外のコマンドを指定した場合にも、ゲームクリアと誤って判定されてしまう恐れがある。ステージクリアの誤判定を解決するには、ゲーム専用の端末エミュレータやコマンド実行履歴を使用する方法等が考えられるが、これらは今後の課題である。

### 3.4 評価実験

基本的なシェルコマンドを学んだことがある大学生7名に対して、シェルコマンド学習支援ゲームを使用してもらい、その後、ゲームの難易度(4段階)と今後シェルコマンドを使用したいかどうか(はい/いいえ)を問う

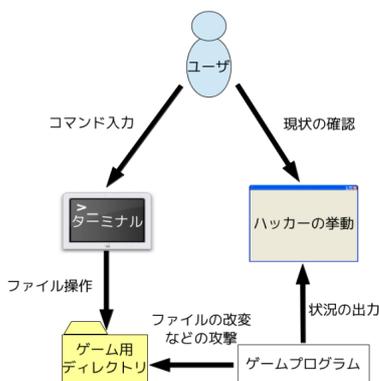


図1: ゲームの流れ図

問題文 1 : 20 秒後に Game ディレクトリ直下にある data.txt を削除する。(data.txt を複製して data.txt.bak を作成しよう。)

回答例 :

```
$ ls
$ cp data.txt data.txt.bak
```

問題文 2 : どこかのディレクトリにあった sample.java を別のファイル名に変更したゾ!

回答例 :

```
$ find . -type f -mmin -1
$ mv foo/hoge.dat foo/sample.dat
```

図2: 問題と回答の例

アンケートに回答してもらった。なお、この7名のうち3名は普段からシェルコマンドを使っており、残りの4名はほとんど使っていない人たちである。ゲーム難易度については「難しい」「やや難しい」と回答した人が6名であった。後でインタビューしたところ、コマンドを思い出している間に制限時間を超えてゲームオーバーになるという意見が多く寄せられた。また、今後シェルコマンドを使用したいかどうかとの質問には、「はい」と回答した人が6名いた。この結果からシェルコマンド学習支援ゲームについて肯定的な評価が得られた。

## 4 おわりに

本稿では、シェルコマンドラインによるファイル操作の活用を促すことを目的としたシェルコマンド学習支援ゲームについて述べた。ゲーム開発後の評価実験の結果、比較的肯定的な評価が得られた。今後の課題としては、学習により効果的な問題文の検討、より正確なステージクリア判定方法の考案などがあげられる。

## 参考文献

- [1] MEng '13Team: Michele Pratusевич '13. Terminus, 2013. <http://web.mit.edu/mprat/Public/web/Terminus/Web/main.html>.
- [2] 特定非営利活動法人エルピーアイジャパン. Linux標準教科書 (ver.1.1.1), 2010.