

タイピング速度向上支援ソフトの作成

～高速タイピングを目指して～

プログラミング演習 D グループ：後藤 和宏

Kazuhiro GOTO

1 はじめに

タイピング速度を向上させるためには練習が必要である。その練習を面白く、楽しくサポートするためにタイピングソフトのシステムを実装した。本タイピングソフトでは、ユーザは各個人のタイピングレベルに応じて 4 段階のレベルを設定することができ、各々のレベルにあった練習をすることができる。また、ただ入力するだけの単純なタイピングソフトとは異なり、ゲーム感覚で練習をすることができる。

2 システム概要

Fig.1 に本プログラム全体のオートマトンの状態遷移図を示す。

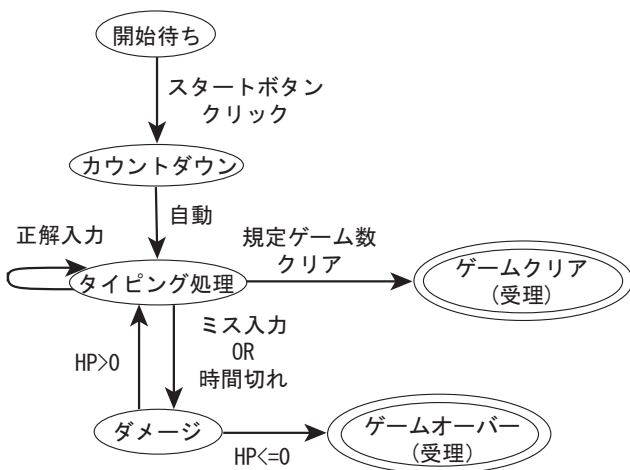


Fig. 1 状態遷移図

以下に Fig.1 に示した状態遷移図について詳しく説明する。

- 開始待ち
開始待ち状態である。ここで難易度を選択し、スタートボタンをクリックするとカウントダウンへと遷移する。
- カウントダウン
ゲーム開始までのカウントダウンをする状態である。画面上には『3』『2』『1』『READY』『GO!!』と順に表示される。ここではユーザが動作することはない、ゲーム開始に備える。

- タイピング処理

タイピング処理状態である。ここでは、画面上に示された文字列を入力する。文字列を入力し、それが正解ならば、次の文字列の入力待ち状態である入力待ちへと遷移する。また、入力が不正解、または時間切れならばダメージ状態へと遷移する。規定ゲーム数を消化するとゲームクリアへと遷移する。タイピング処理を詳しく説明したものを 3.2 節の Fig.2 に示す。

- ダメージ

ダメージを受ける状態である。入力待ちで不正解、または時間切れであった場合に遷移する。ダメージによりヒットポイント (以後 HP とする) が減少し、0 以下になった場合ゲームオーバーへと遷移する。HP が 0 より大きければ入力待ちへと遷移する。

- ゲームオーバー

ゲームオーバーとして受理される。ダメージを受けて、HP が 0 以下になるとゲーム終了となり、画面に『Game Over』と表示される。

- ゲームクリア

ゲームクリアとして受理される。ダメージを受けながらも、HP が 0 より大きければゲームは続き、規定ゲーム数をクリアするとゲームクリアとなる。

3 タイピング処理

タイピング処理には、いろいろなタイピング処理がある。例えば、タイプされた文字を一括して識別するもの。また、一文字ずつ正誤判定をしていくものなどがある。今回は、後者のタイピング処理を実装した。以下に作成したタイピングソフトのタイピング処理について説明する。

3.1 タイピング処理概要

タイピング処理を実装するにあたって、リアルタイムに正誤判定するため、またすべての入力方式に対応するために様々なルールを設定した。タイピング処理の順序を以下に述べ、それぞれのルールについて詳しく説明する。

3.1.1 タイピング処理の順序

タイピング処理の順序は以下の通りである。

1. 入力されたキーを 3.1.2 項の 6 つのパターンとマッチングし登録
2. パターンの組み合わせを 3.1.3 項のルールと比較
3. 途中 3.1.4 項に示すようなエラー入力があると、入力は消去される
4. ルール通りの入力があると文字を文字データベースから出力
5. 出力された文字と入力すべき文字を比較
6. 文字が一致すれば次の文字へ
7. 不一致ならば再入力

3.1.2 パターン

3.1.1 項に示したパターンについて説明する。作成した 6 つのパターンを以下に示す。なお、可子音、左子音、NOTEXPECT、PIN は造語である。

- 母音
a, i, u, e, o
- 子音
母音以外のアルファベット
- 可子音
子音が 2 つ続いている後ろのアルファベット
例：by, ch, xt, sh など
- 左子音
同じ子音が続いている後ろのアルファベット
例：ss, kk, bb など
- NOTEXPECT
作成したタイピングソフトでは使用しないキー
例：@, /, ; など
- PIN
1 つの入力で受理されるキー
例：1, 2, 3, I, S, D, L など

3.1.3 文字データベースから文字を出力するルール

3.1.1 項に示した文字が出力されるルールは以下の 6 通りである。

- 母音
例：a あ
- 子音, 母音
例：si し

- 子音, 可子音, 母音
例：sya シャ
- 子音, 左子音, 母音
例：kka っか
- 子音, 左子音, 可子音, 母音
例：bbya っびゃ
- PIN
例：I I, 7 7

3.1.4 エラー入力

入力が消去されるエラー入力の例を以下に示す。

- 子音, 子音
例：tr
- 子音, 可子音, 子音
例：tyt
- 子音, 左子音, 子音
例：ssk
- 子音, 左子音, 可子音, 子音
例：sshya
- NOTEXPECT
例：@
- 子音, PIN
例：s7

3.2 タイピング処理のオートマトン

Fig.2 にタイピング処理のオートマトンの状態遷移図を示す。

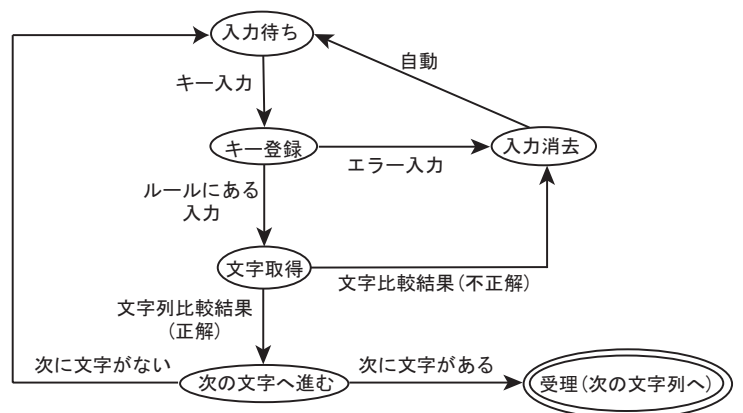


Fig. 2 タイピング処理の状態遷移図

4 プログラムの実行結果

実際に作成したプログラムについて説明する．Fig.3 にプレイ中の画面を示す．

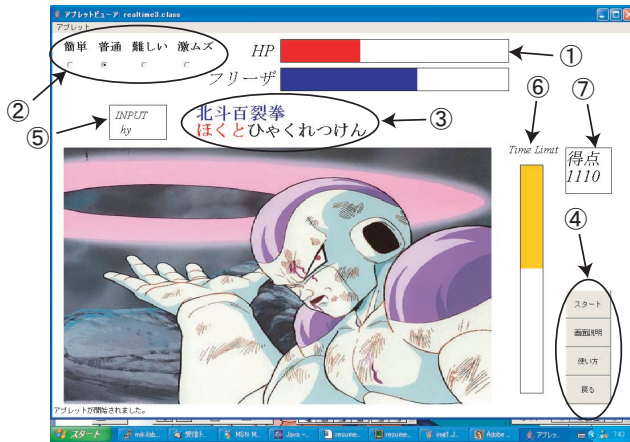


Fig. 3 ゲーム画面説明

以下にゲーム画面を詳しく説明する．

上のゲージはプレイヤーの HP の残量を示している．ダメージを受けると減少していく．下のゲージは、ボスの HP の残量を示している．

選択する難易度によって、制限時間の減少する速度が変化する．

ここに表示される文字列を入力する．正解するとその文字が赤色に変化する．

スタートボタンをクリックすることによって、ゲームを開始することができる．画面説明、使い方ボタンをクリックすると、ゲームについての説明が表示される．

入力中のキーが表示される．

文字列ごとに制限時間が異なっており、さらに難易度ごとに減少する速度が変化する．このゲージがなくなるまでに、入力すべき文字列を入力できなければ、時間切れとなりダメージを受ける．簡単な制限時間を 1 とした場合の難易度別の制限時間の比率を Table1 に示す．

Table 1 難易度別ゲージの減少速度

難易度	ゲージの減少速度
簡単	1
普通	0.8
難しい	0.44
激ムズ	0.3

Table1 は、普通は簡単な 0.8 倍、難しいは簡単な 0.44 倍、激ムズは簡単な 0.3 倍の制限時間であることを示している．

5 動作実験

ゲーム開始からの動作実験を行う．動作実験の様子を Fig.4 に示す．



Fig. 4 動作実験の様子

1. 開始待ち状態で、難易度の選択を“簡単”、“普通”、“難しい”、及び“激ムズ”の中から選択する．この難易度によって、制限時間が異なる．ここでは、“普通”を選択した．
2. スタートボタンをクリックすることによってカウントダウンが始まった．
3. カウントダウンが終了すると、入力する文字列が表示され、タイピングがスタートした．
4. 文字列を入力し、一文字ずつ正誤判定が行われた．入力すべき文字以外の文字の入力、または、エラー入力をするダメージを受けた．また、制限時間内に文字列の入力ができなかった場合もダメージを受けた．ダメージを受けると、画面中央上部の HP が減少した．ここでの入力する文字列は“こうもんをよじのぼる”であった．
5. 正解入力により、1つの文字列をすべて入力した場合、得点が加算された．
6. 本実験では、最終的にボスを倒し、ゲームクリアした．その画面を Fig.5 に示す．

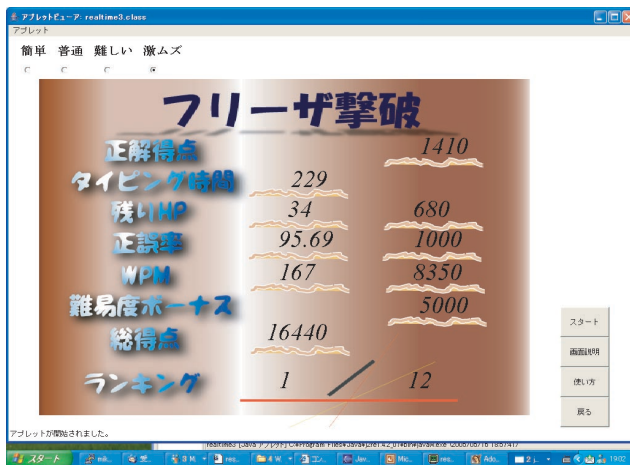


Fig. 5 ゲームクリア画面

Fig.5 中のそれぞれのオプションの意味を以下に示す。

- 正解得点
ゲーム中、1つの文字列を時間切れまでに、正しく入力すると30点加算。
- タイピング時間
実際に、キー入力を行っている時間を表示。
- 残りHP
ゲームクリア時に残っているプレイヤーのHPの残量を表示。残りHP × 30を点数として加算。
- 正誤率
正しくキー入力された率を算出。95%以上で点数加算。
- WPM
WPM(Words Per Minute: 1分あたりの単語数)とは1分間に何文字の文字(ひらがな)を正しく入力したかを示す値である。ここではその値を表示している。WPM × 50を点数として加算。
- 難易度ボーナス
難易度別にクリアボーナスとして、簡単は300点、普通は500点、難しいは1500点、激ムズは5000点加算される。
- 総得点
すべての点数が加算された得点が表示される。
- ランキング
総得点のランキングが表示される。

6 実験結果

Fig.4の動作実験の様子より、HPが減少している様子が確認できる。そして、文字を入力している間に、制

限時間の減少もプログラム通り実行されている様子が確認できた。また、一文字ごとに入力があっていれば文字が赤色に変化し、まちがっていれば入力が消去される、ルール以外の入力は消去されるタイピング処理も正しく実行できた。ボスの画面でも、ボスのHPの減少が正しく行われ、ボスのHPによって画面の遷移が正しく行われた。Fig.5のボスを倒した後の画面でも、正解得点、タイピング時間、残りHP、正誤率、WPM、難易度別ボーナス、総得点およびランキングが正しく表示された。

7 まとめ

本報告ではタイピング速度を向上するための支援ソフトの作成を行った。動作実験を行ったところ、タイピング処理、時間制御、画面遷移およびオプションの表示ともに適切な動作を行うことが確かめられた。以上より、本演習で作成したタイピングソフトは、ユーザが楽しくタイピングを練習するために有効なシステムであると考えられる。

参考文献

- 1) Java プログラミング BlackBook, Steven Holzner, インプレス, 2001 Jan 1
- 2) 新 Java 言語入門, 林 晴比古, ソフトバンクパブリッシング, 2002 Mar 30