

4.5 ゲームの仕様(3)

前回までに二つのゲームを作った。今回は以下の機能をもつユーザーインターフェイス実装しよう。

- コマンドによる機能の切り替え

先回のままでは、ユーザーが思い通りの機能呼び出せない。ユーザーが必要な機能呼び出すための仕組みが必要である。呼び出しの方法として、GUI(Graphical User Interface)を使ったもの、コマンドをユーザーに入力させるものなどがあるが、今回は比較的実装が簡単なコマンド入力の方法をとることとしよう。

今回の開発対象となる機能は以下の関数により実現する。必要なヘッダファイルをインクルードすること¹。

- void menu()
 - 以下の変数を宣言、初期化せよ。
 - * DATA型の要素をN個もつ配列として利用するためのポインタdataArrayを用意し、適切に初期化せよ。
 - * タイピングの問題数をint型の変数nで表す。また、DataReaderの戻り値を保持する²int型の変数successを用意する。
 - * int型の変数retを初期値1で宣言しておく。(retは下で述べるループを抜けるか判定する変数)
 - * キーボードから読み込んだ入力1文字用のchar型の変数chを用意する。
 - * その他、必要な変数は適宜、宣言・初期化すること。
 - menu> という文字列を画面にprintf関数を使って表示する。
 - 変数retが1である間、以下の処理を繰り返す。
 - * getchar関数を実行し³、戻り値を変数chにセットする。
 - * 変数chにセットされた一文字はアルファベットであり、これがコマンドとなる。qがクイズゲーム、tがタイピングゲーム、eがプログラム終了である。そのため、chの値に応じて以下の処理を行う(switch文を使うこと)。

¹使用している関数が定義されているヘッダファイルはすべてインクルードすることが必要である

²typing.txtの読み取りの成否を表す

³text

- ・ chの値が\n （改行文字）である場合、何もしない(getcharが読み込んだ改行文字の除去を取り除くための対策)
- ・ chの値がqの場合、input関数を使ってクイズゲームを実行し、その後、menu> という文字列を画面にprintf関数を使って表示する。
- ・ chの値がtの場合、DataReader関数を実行して成功したら、typing関数を使ってタイピングゲームを実行する。その後、menu>という文字列を画面にprintf関数を使って表示する。
- ・ chの値がeの場合、変数retの値を0にセットする（ループを抜けることになる）。
- ・ これらに当てはまらない場合、画面に

```
invalid command.
と表示する。
```

－ 利用したポインタを解放する。

以上を実装せよ。ヘッダファイルmenu.hは以下の通りとし、

```
#ifndef MENU_H_
#define MENU_H_
#include "quiz.h"
#include "typing.h"

void menu();
#endif
```

menu関数の実装は、ファイルmenu.cに行うこととする。なお、以下のプログラム（ファイル名をmenu.driver.c、実行ファイル名をmenuとせよ）に上記のプログラムをリンクして実行せよ。

```
#include "menu.h"

int main(int argc, char **argv){
    menu();
}
```

4.6 レポートについて（重要）

第4回から第6回までで作ったquiz.c、typing.c、menu.cのプログラムソース、これらに対応するヘッダファイル、menuを実行した時の入出力を示し（最低限、すべてのコマンドは実行すること）、また苦勞した点、工夫した点、考察などをレポートしてまとめ、提出せよ。レポートの締め切りは第7回の授業とする（開始時に提出）。レポートはWordもしくはTeXで作成し、PDF形式に変換したものを提出してもらう。加えて、ソースプログラムquiz.c、typing.c、menu.cも提出を求める。紙によるレポートの提出は不要である。

なお、レポート作成時に他の受講者と相談することは奨励するが、他の受講生および過去のレポートからのプログラムソースおよびレポートのデッドコピーの疑いがあるものは点をつけない。各自、レポートは自力で作成すること。加えて、プログラムソースや入出力のみで、苦勞した点、工夫した点、考察等がないものについても減点する。必ず、これらについても記述すること。