

7.5 もっともっとゲームらしくする

前回までで、宝物を落とすモンスターとそれを拾う自キャラを実現した。また、点数と面を画面に表示する機能もつけた。

ただし、ゲームとしてはもう一味欲しいところである。それに、ゲームオーバーになったりqを押してゲームをやめた時に得点の情報も消えてしまう。これをなんとかしよう。ますますくどいが、必ずncurses.hをインクルードし、コンパイル時にはリンクすること。以下は、今回実現する機能である。

- 最高得点の保存
- 最高得点のロード
- ゲームの改良

今回実装する関数は以下のとおりである。

- `int getHighScore(int highScore, int score)`
 - もし、scoreがhighScoreよりも大きければscoreを、scoreがhighScore以下であればhighScoreを返す。
- `int loadHighScore()`
 - ファイル `highscore.txt` からfscanf関数を用いて点数の数値を読み取り、その値を返す。ただし、`highscore.txt` ファイルには点数の数値のみが記されているものとする。
 - ファイルがないなど、読み取りにエラーが発生した場合は0を返す。
- `void saveHighScore(int score);`
 - ファイル `highscore.txt` に、fprintf関数を用いて点数の数値を書きだす。
- `void highScoreDisp(int highScore);`
 - highScoreの値を1行目の真ん中から二列ヨコ(COLS/2+2)にhighScoreの値をHI SCORE : Hのように画面に表示せよ (HがhighScoreの値)。

今回は、先回のredraw2.hに加え、以下のhighscore.hを用いる。

```
#ifndef HIGHSCORE_H_
#define HIGHSCORE_H_
#include "redraw2.h"

int getHighScore(int highScore, int score);
int loadHighScore();
void saveHighScore(int score);
void highScoreDisp(int highScore);
#endif /* HIGHSCORE_H_ */
```

ここで宣言されている関数の実装はファイルhighscore.cにて行うこととする。

7.6 main関数の修正

これにともない、先回の課題で提示したncurses-game2.cにおけるmain関数を以下のように変更する¹。

- getHighScore, loadHighScore, saveHighScore, highScoreDispの各関数を用いて正常にハイスコアを表示、保存、更新できるようにせよ。
- さらにこのゲームをより楽しめるものにするための工夫を行え。

このようにしてできたプログラムのファイルをncurses-game3.cとせよ。必要なファイル、ライブラリとともにコンパイル・リンクせよ。コンパイルしてできる実行ファイルの名前はncurses-test3とする。

以上を作成し、実行してみよ。

7.7 最終レポート

レポートには以下を記載すること。

- 今回作成したredraw2.c、highscore.c、ncurses-game3.cの各ソースファイル

¹先回作成したredraw2.cには変更を加えない。

- 実行画面のキャプチャ（プレイしている途中のものでよい）
- 各ソースファイルについての考察・苦労した点（なるべく詳しく書くこと）

考察および苦労した点についての記載がない場合は減点する。また、上記の仕様に加えて、必ずプログラムの改善・工夫を行うこと。改善したり工夫した点についての記述も必ず加えること（工夫の程度によって加点する）。

第14回の本演習の終了時から一週間以内に提出すること。いままでのレポート同様、ソースファイルとレポートのPDFファイルの提出を求める。Scombで提出すること。病気等の特段の理由がない場合、この締め切りを過ぎた場合はレポートを受け取らない。締め切り厳守でレポートを作成すること。