

## 7.1 もっとゲームらしくする

前回用いたncursesライブラリが提供している関数を用いて次の機能を実装しよう。くどいが、必ずncurses.hをインクルードすること。

- 自キャラ位置の初期化
- 自キャラの表示・キーボードによる移動
- 当たり判定
- 点数表示（面表示、得点表示、残りカウント表示）
- ゲームオーバー時の表示

なお、前回、モンスターをMにしたので、

自キャラを@で表す

ことにしよう。

先回はモンスターがひとりでふらふらしていたわけだが、今回は自キャラを加えたいし、キーボードで自由に動かしたい。そしてモンスターが落とした財宝を拾うと面クリアだ。ただし、面が進むと拾わなければならない財宝の数は一つずつふえるぞ。

そのための関数を以下に用意する。

- void initPlayerLocation(POSITION \*player);
  - playerの現在位置に半角スペースを表示<sup>1</sup>。
  - playerのxとyの値をともに5にセットする。
  - playerの新しい位置に半角の@（アットマーク）を表示する。
- void getPlayerLocation(POSITION \*player, int key);

getch関数を事前に実行し、入力されたキーの値を引数keyにとることを前提とする。

  - keyの値<sup>2</sup>によって以下のようにplayerのxおよびyを変更する。  
switch文を活用せよ。

---

<sup>1</sup>mvprintw関数を使おう。

<sup>2</sup>ncurses.hで定義されているものを使う。

- \* keyがKEY\_LEFTであるとき、playerのxの値が0より大きい時、xの値から1を引く。
- \* keyがKEY\_RIGHTであるとき、playerのxの値が(COLS-1)より小さい時、xの値に1を加える。
- \* keyがKEY\_UPであるとき、playerのyの値が0より大きい時、yの値から1を引く。
- \* keyがKEY\_DOWNであるとき、playerのyの値が(LINES-1)よりも小さいとき、yの値に1を加える。
- \* defaultでは (keyが上記以外の場合)、何もしない。

- void getMonsterLocation(POSITION \*monster, POSITION \*treasure, int \*numTreasure);

先回のgetMonsterLocation関数を改造して以下のような処理とせよ。

- int型のdx、dyを宣言する。getMonsterLocation関数を実行するたびに、dxとdyにそれぞれ、-1,0,1のいずれかの値をランダムに代入する。
- int型のorigX、origYを宣言する。それぞれ現在のmonsterのx座標、y座標をこれらにセットする。
- int型のdiceを宣言する。diceはランダムに0から9までの値をとることとする。
- monsterのxの値が0よりも大きくかつdxが負である場合、または、monsterのxの値が COLS-1より小さくかつdxが正である場合、monsterのxの値にdxを加える。
- monsterのyの値が0よりも大きくかつdyが負である場合、または、monsterのyの値が LINES-1より小さくかつdyが正である場合、monsterのyの値にdyを加える。
- \*numTreasureが1でかつmonsterのxとyおよびtreasureのxとyがそれぞれ等しい時<sup>3</sup>、monsterのxとyにorigX、origYをセットする。
- もし、diceが0に等しく、\*numTreasureの値が0に等しいとき、かつ、上の処理によりmonsterが移動した場合（つまりmonsterのxもしくはyが変化した場合）、 treasureのxとyにorigX、origYを

---

<sup>3</sup>つまり、treasureのある場所にmonsterがいるとき。

セットし、`*numTreasure`の値を1にセットし、今セットされた`treasure`の場所にTの文字を表示する。

- `int getTreasure(POSITION *player, POSITION *treasure, int *numTreasure);`
  - `*numTreasure`が1でかつ`player`の`x`と`y`および`treasure`の`x`と`y`がそれぞれ等しい時<sup>4</sup>、`treasure`の位置に半角スペースを表示し、`*numTreasure`を0にセットしたのち1を返す。それ以外の時には0を返す。
- `int encounter(POSITION *player, POSITION *monster);`
  - `player`の`x`と`y`および`monster`の`x`と`y`がそれぞれ等しい時<sup>5</sup>、1を返し、それ以外の時には0を返す。
- `void initMonsterLocation(POSITION *monster);`
  - `monster`の現在位置に半角スペースを表示。
  - `monster`の`x`をCOLSの半分、`y`をLINESの半分にセットする。
  - `monster`の新しい位置にMを表示する。
- `void initTreasureLocation(POSITION *treasure, int *numTreasure);`
  - `treasure`の現在位置に半角スペースを表示。
  - `*numTreasure`の値に0をセットする。
- `char gameOver();`
  - 画面の中央に“GAME OVER”と表示する<sup>6</sup>。
  - `ncurses`ライブラリが提供する`refresh`関数を実行する。
  - `sleep`関数<sup>7</sup>を使い、2秒停止する。
  - `char`型の値 ‘q’ を戻り値として返す。

---

<sup>4</sup>つまり、`treasure`のある場所に`player`がいるとき。

<sup>5</sup>つまり、同じ場所に`player`と`monster`がいるとき。

<sup>6</sup>位置に注意。COLSとLINESをうまく使おう。

<sup>7</sup>`sleep(2)`で2秒停止する。

以上をredraw2.cに実装せよ。なお、今回は、以下のredraw2.hを用いる。MAX\_COUNTは得点の初期値であり、時間に応じた値がこれから引かれて点数となる。

```
#ifndef REDRAW2_H_
#define REDRAW2_H_

#define MAX_COUNT 100

struct position{
    int x;
    int y;
};
typedef struct position POSITION;

char gameOver();
int getTreasure(POSITION *player, POSITION *treasure, int *numTreasure);
int encounter(POSITION *player, POSITION *monster);
void initPlayerLocation(POSITION *player);
void initMonsterLocation(POSITION *monster);
void initTreasureLocation(POSITION *treasure, int *numTreasure);
void getPlayerLocation(POSITION *player, int ch);
void getMonsterLocation(POSITION *monster,
                        POSITION *treasure, int *numTreasure);

#endif /* REDRAW2_H_ */
```

## 7.2 main関数

以上で実装した関数を使ったmain関数を以下のように用意した。

```
#include <ncurses.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include "redraw2.h"

int main(int argc, char** argv){
```

```
/* declaration of variables */
POSITION *player=(POSITION *)malloc(sizeof(POSITION));
POSITION *monster=(POSITION *)malloc(sizeof(POSITION));
POSITION *treasure=(POSITION *)malloc(sizeof(POSITION));
int cnt=0;
int ch;
int i;
int stage=0;
int score=0;
int numTreasure;
int clear=0;

/* initialization of rand function */
srand((unsigned)time(NULL));

/* initialization of ncurses */
initscr();
noecho();
cbreak();
curs_set(0);

keypad(stdscr, true);
wtimeout(stdscr, 500);

/* initialization of variables */
player->x=5;
player->y=5;
monster->x=COLS/2;
monster->y=LINES/2;
treasure->x=-1;
treasure->y=-1;
numTreasure=0;

/* First, let's read key. */
ch=getch();

while(ch!='q'){
    mvprintw(1,2,"stage : %d", (stage+1));
```

```

mvprintw(2,2,"score : %d", score);
mvprintw(3,2,"count : %3d", MAX_COUNT-cnt);
cnt++;
/* move monster */
mvprintw(monster->y,monster->x," ");
getMonsterLocation(monster, treasure, &numTreasure);
mvprintw(monster->y,monster->x,"M");

/* move player */
mvprintw(player->y,player->x," ");
getPlayerLocation(player, ch);
mvprintw(player->y, player->x,"@");
refresh();

/* judgement */
clear+=getTreasure(player,treasure, &numTreasure);
if(cnt==MAX_COUNT){
    /* could not get treasures */
    ch=gameOver();
}else if(encounter(player,monster)){
    /* caught by monster */
    ch=gameOver();
}else if(clear==stage+1){
    /* stage clear */
    mvprintw(4,2,"Gotcha!");
    refresh();
    sleep(2);
    mvprintw(4,2,"      ");
    stage++;
    score+=MAX_COUNT-cnt;
    cnt=0;
    clear=0;
    initPlayerLocation(player);
    initMonsterLocation(monster);
    initTreasureLocation(treasure, &numTreasure);
}else{
    ch=getch();
}

```

```
    }  
    endwin();  
    return 0;  
}
```

このプログラムのファイルをncurses-game2.cとし、redraw2.cとともにコンパイルしてできる実行ファイルの名前はncurses-game2とする。

以上を作成し、実行してみよ。