COMP0204: Introduction to Programming for Robotics and AI

Assessment 6: Robot Safe Maneuver Game

Course lead: Dr Sophia Bano

MEng Robotics and AI

UCL Computer Science

Tutorial from Lab 8 – Exercise 1

- Draw a rectangle using ncurses

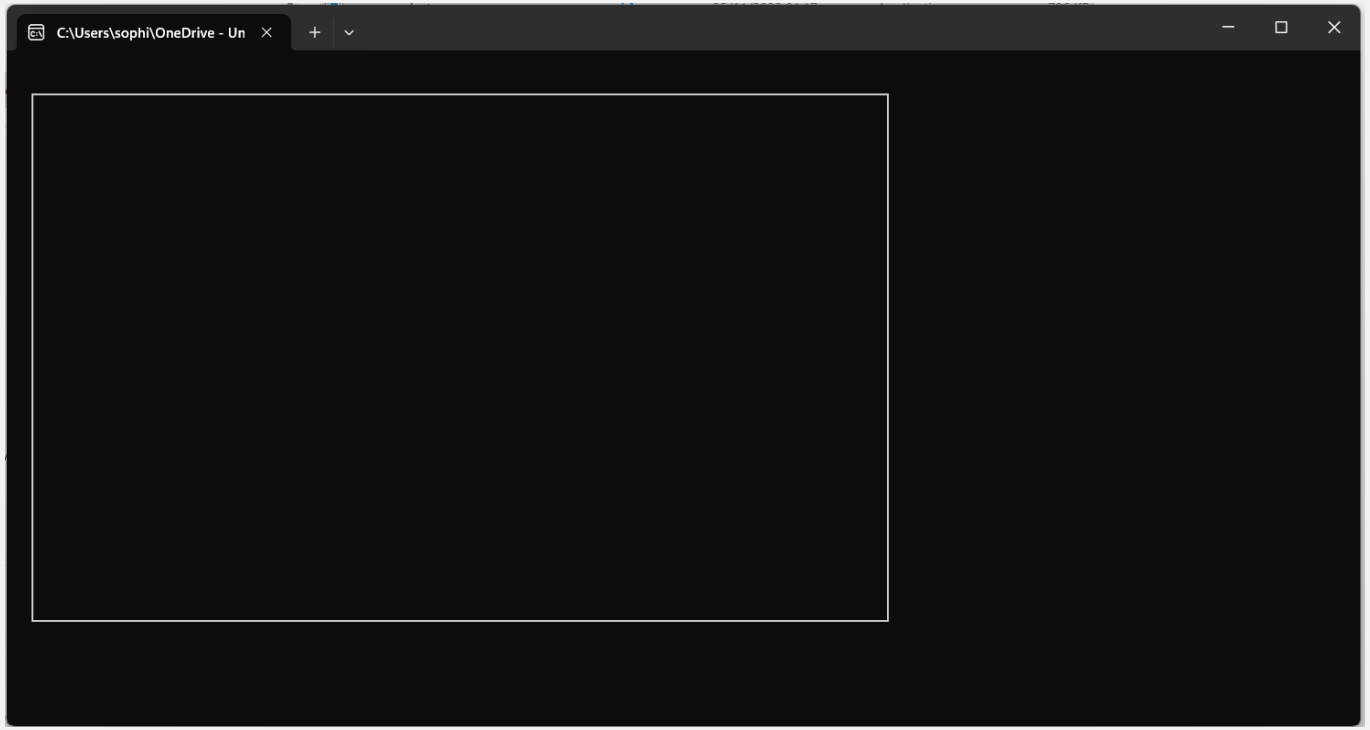
Hint: Use move and addch

```
move(y, x);
addch(ACS_VLINE);
```

Or

movaddch

```
mvaddch(1, 1, ACS_VLINE);
```



Taking arrow keys from the keypad

```c
#include <ncurses/ncurses.h>
int main()
{   int ch;
    initscr();
    raw();
    keypad(stdscr, TRUE);
    noecho();

    printw("Type any character to see it in bold\n");
    while (1)
    {   ch = getch();
        if (ch == KEY_LEFT)
            printw("Left arrow is pressed\n");
        else if (ch == KEY_RIGHT)
            printw("Right arrow is pressed\n");
        else if (ch == KEY_UP)
            printw("Up arrow is pressed\n");
        else if (ch == KEY_DOWN)
            printw("Down arrow is pressed\n");
        // if ESC is pressed, end the program
        else if (ch == 27)
            break;
    }
    //getch();
    endwin();
    return 0;
}
```

Tutorial from Lab 8 – Exercise 2

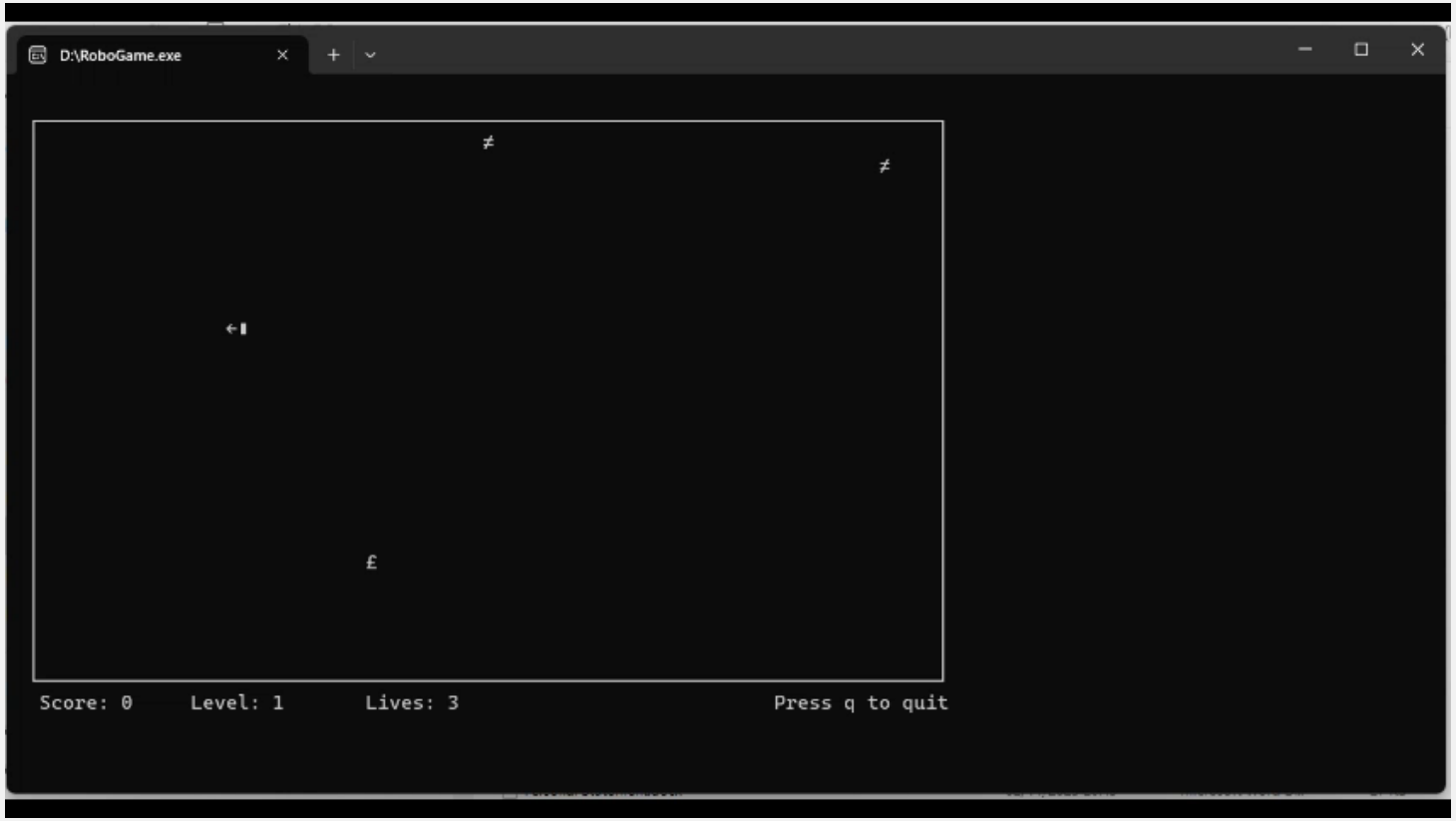
- In the same code as of Exercise 1, now add a character ‘R’ in your favorable colour at the centre of the rectangle, then use the arrow keys to move the ‘R’ up, down, left and right.

- **Hint:** Keep note on the current location of ‘R’ and increase/decrement its coordinates based on up, down, left, right key press.

Background

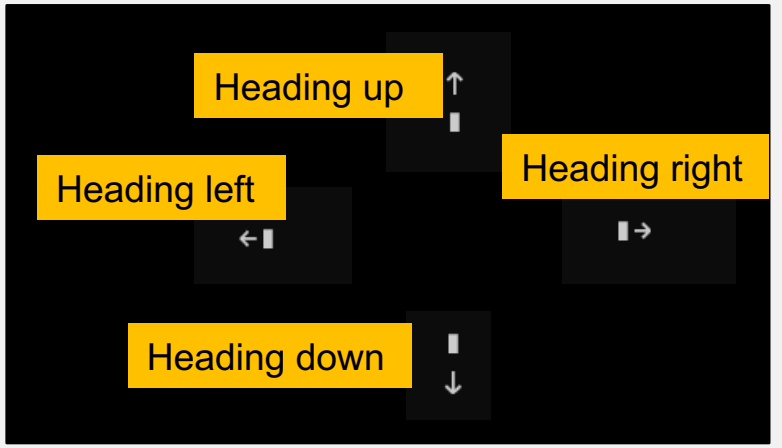
- You are manually operating a robot which is operating at a disaster site in a room.
  - The robot is required to rescue people without hitting the danger locations.
  - Beware: the danger locations keep on growing.

IMPORTANT: Use <ncurses.h> for the game development. No other GUI library is permitted. Feel free to use other standard libraries, like time.h, string.h, etc (as needed)



Simplest Demo

Assessment 6 (20 marks)



Part 1

- **Draw the game screen** (rectangle defining the room).
- **Draw the robot** which is made of atleast 2 characters (one representing the heading and one representing robot body)
- **Move the robot with constant speed** using left, right, top and bottom arrows.
- **Implement boundary conditions** for the room walls. If, the robot hits any wall, the robot losing a life. The robot has three lives. The game should End if all lives are lost.
- If user press ‘q’ then the game should End.

Assessment 6 (20 marks)

Part 2

- **Draw a person** to rescue at random location (represented by £ in my example)
- **Draw two danger locations** at random (represented as ≠ in my example). Every time a person is rescued (it gets a score of +10), the **danger locations increases by two**.
- When your **robot hits a danger location** or walls, it loses a life.
- When **five people are rescued**, the game moves to **next level**. Increase the speed of the robot to increase complexity. Continue to increase speed at 5, 10, 15,… rescues.
- **Display score, level, and lives**.

Assessment 6 (20 marks)

Part 3

- Part 1 and 2 give the essential requirement of the robot.

- **BE CREATIVE:** You can add additional features to your game as you desire.

Example:

- Having separate game screen (room layout with obstacles at each level).
- Adding colors to your game.
- Adding intro and ending game screen.
- Adding leadboard (asking user to enter their name and keep a record on username and scores)
- ……

Assessment 6 (20 marks)

Submission

You are required to **submit the source file (.c) and executable as a single zip file** via Moodle.

(**indicate as comment on the top of your .c file the operating system used**)

Make sure your code is well commented, variable and function names are indicative of the task they are performing.

Deadline: 6th Dec 2023 (11:59pm)

Assessment 6 (20 marks)

Marking will be based on functionality of each Part, and also inperson viva.

Vivas are scheduled on 11th December during the lab session.

Inperson viva is mandatory so make sure you attend the lab session as notake will be scheduled.

During the viva, you will be requested to download the submitted program, compile and run. And explain your code.

Note: Be remaindered that UCL has strict policy on plagiarism
https://www.ucl.ac.uk/students/exams-and-assessments/academic-integrity

Assessment 6 - Operating System Info