

**TECHNICAL UNIVERSITY IN KOSICE**  
**FACULTY OF ELECTRICAL ENGINEERING AND**  
**INFORMATICS**

**PROBLEM SET 6: CURSES**  
**THE CONSOLE GAME - SATELA**

**Kostiantyn Chernov**

**First year**

**2020/2021**

**C1**

## **CONTENT**

<b>1. TASK'S GOALS</b>	<b>3</b>
<b>2. SOLUTION PROPOSING</b>	<b>3</b>
<b>3. USING OF THE PROGRAM</b>	<b>6</b>
<b>4. CONCLUSIONS</b>	<b>8</b>

## ***1.Task's goals***

Use the ncurses library to create any program (game, presentation, or other artwork).

Together, the project must cover at least 3 of the following goals:

- work with colors;
- keyboard control (without Enter confirmation);
- multiple levels;
- working with time (with time the program changes);
- working with command-line arguments;
- working with files.

The project must be more complex than the examples and its level must be enough.

## ***2.Solution proposing***

To our goals perfectly fits the game called “Guitar Hero” or as I want to call it – “Satela”. It is a 2d game, where, from top to the bottom are moving plates with different speed as game continue. The main goal of the game is to achieve as many score as possible by hitting on specific buttons when plate approaches bottom edge. Example of that game provided in the image you can see below.



Image 2.1. Guitar Hero game

The game concept is already satisfying the first goal of the project, which is a two-dimensional world. To realize all features of the game, was decided to create it through a C library called “ncurses”. This library provides all needed functions, which allow us, to paint on the terminal window on Linux.

So, tasks which were decided to complete, became:

- work with colors;
- keyboard control (without Enter confirmation);
- multiple levels;
- working with time (with time the program changes).

The “ncurses” library provides the functionality of continues input and color changing. We can create different levels or maps to make our game more difficult and various. Thus, we also work with time. The main game process is going on in certain refresh period. With every refresh plates are moving.

Library functions used:

- `mvprintw()` – put symbol or text on certain coordinates;
- `napms()` – interrupt execution of the game on certain period;
- `refresh()` – paint all changing made by `mvprintw()` function;
- `clear()` – clear a terminal screen;
- `getch()` – return pressed key value;
- `attron()` – apply new color for painting;
- `attroff()` – disable color pair.

Own function created:

- `make_zone_of_game(char lvl[])` – draw game field where all action is depicted;
- `check_d/f/k/l (const int i, int map[][4])` – when d/f/k/l is clicked checks if there is a plate on appropriate position.
- `game(int default_map[][4])` – main game loop, ends when approached last plate in map. In this loop is processing user input;
- `chose_map(int map[][4], char lvl[])` – shows available maps, and when chosen reads its content;
- `start_game()` – starts game. Clears console, chose map, draw zone, and starting main game loop.
- `menu()` – displays menu for user where he/she can start or quit the game;

- `end_menu(int default_map[][4],char lvl[])` – shows end menu, where user can restart game, choose another map or quit the game.

### ***3. Using of the program***

When we have the source code of the game, we can make execution file with help of the command `gcc ps6.c -Werror -Wall -Incurses`

When you already have an execution file, you can execute it by enter “./filename” in command line. Then you’ll see the game window, with couple of options, you can Start game(input “s” button) or quit game(input “q” button). If user chosen start the game, he or she can load a map. There are only 2 map for now. Finally, when map is loaded, game started. Main game process is depicted bellow.



Image 3.1. Main game window

Now, by pressing on keyboard, especially on d/f/k/l buttons when plate is inside of area before keys, user will achieve more and more points. If user clicks on button when there is no plate in zone points decrease. Image also provided bellow.

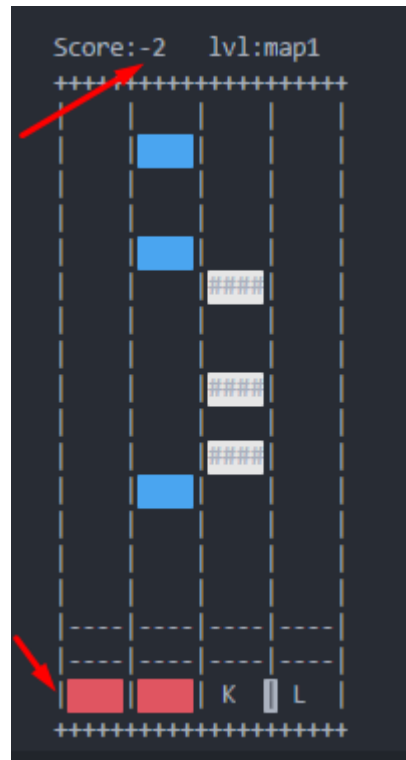


Image 3.2. User misses plates

When player lose, he/she will see end menu and can choose another map or restart game, or quit.

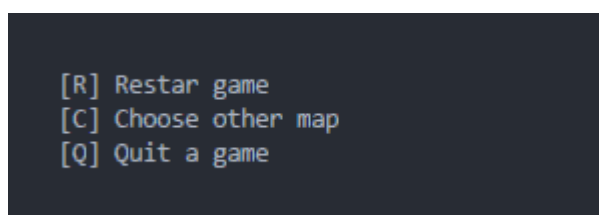


Image 3.3 End menu

## ***4. Conclusions***

There are couple possible improvements as more maps, speeding up plates, add feature of collecting coins from single game and then buy improvements for player as score multiplier or freeze.

To conclusions, would be great to say, that all set goals were satisfied. Were improved knowledge of C language, especially its library “ncurses”. Made game, turned out very sophisticated, easy to use and funny.