

TECHNICAL UNIVERSITY IN KOSICE
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATICS

PROBLEM SET 6: CURSES
THE CONSOLE GAME – STICKFIGHT

CONTENT

1. TASK'S GOALS	3
2. SOLUTION PROPOSING	3
3. USING OF THE PROGRAM	5
4. CONCLUSIONS	9

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 “Stick fight”. It’s a 2d game, where, in the center is standing stickman and from sides to the center is moving enemies. The main goal of the stickman is to defeat as many enemies as he can. Example of that game provided in the image you can see below.

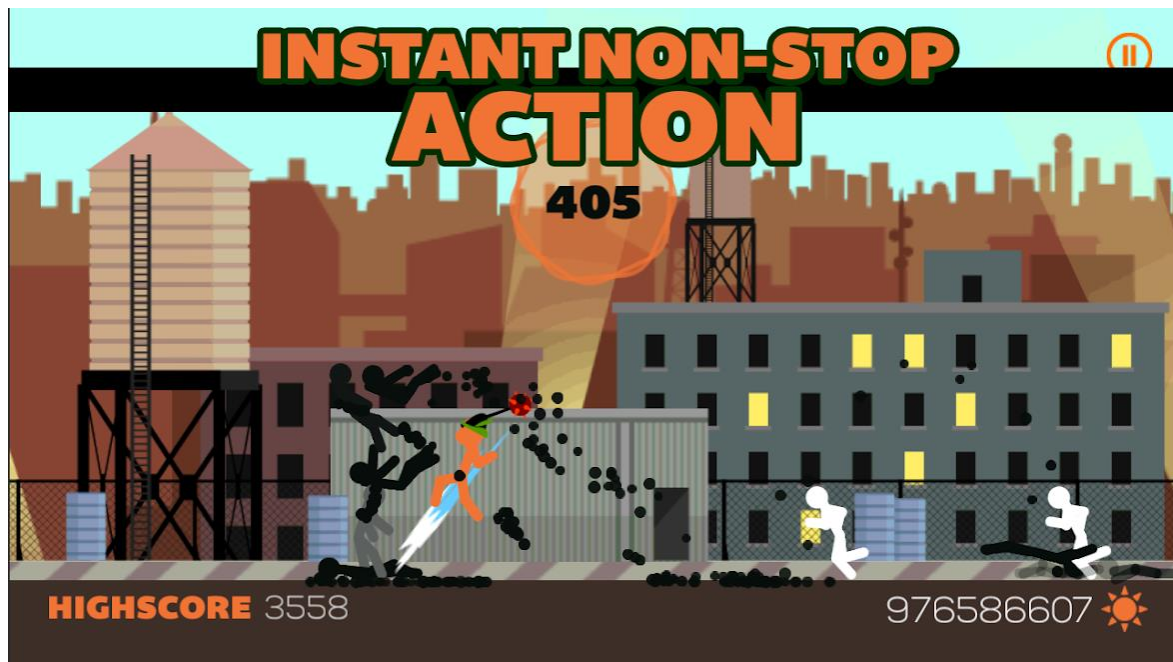


Image 2.1. Stick fight game

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, it 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. The game difficulty changing is going to be made by speeding up of enemies. Thus, we also work with time. The main game process is going on in certain refresh period. By defeating more and more enemies, a refresh period will speed up.

Library functions used:

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

Own function created:

- `draw_sickman(struct Stickman obj)` – draw stickman with provided coordinates in input structure;
- `draw_enemies(char enemy[])` – draw enemy with provided coordinates in input array;
- `stick_right(struct Stickman obj, struct timespec t, int size, int enemys[size])` – draw the enemies killing process to the right side, and return 1 or 0 if enemy was killed;
- `stick_left(struct Stickman obj, struct timespec t, int size, int enemys[size])` – draw the enemies killing process to the right left, and return 1 or 0 if enemy was killed;

3. Using of the program

When we have the source code of the game, we can make execution file with help of the command “make”. This command will work only if you Makefile. The content of the Makefile you can see in the image bellow.

```
M Makefile
1  # variables
2  CC=gcc
3  CFLAGS=-std=c11 -Wall -Werror
4  LDLIBS= -lm -lcurses
5  OUTPUT=$@
6
7  # targets
8  %: %.c
9      $(CC) $(CFLAGS) $@.c $(LDLIBS) -o $(OUTPUT)
10
```

Image 3.1. Content of Makefile

When you already have an execution file, you can execute it by enter “./filename” in command line. Then you’ll see the game window, image provided below.

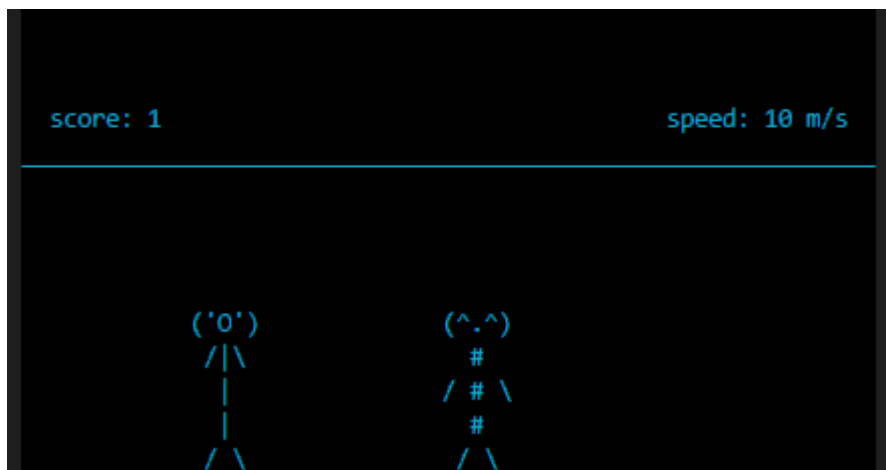


Image 3.2. Main window

Now, by pressing left and right arrows, stickman will be killing enemies by slashing them of his sword. Image also provided bellow.

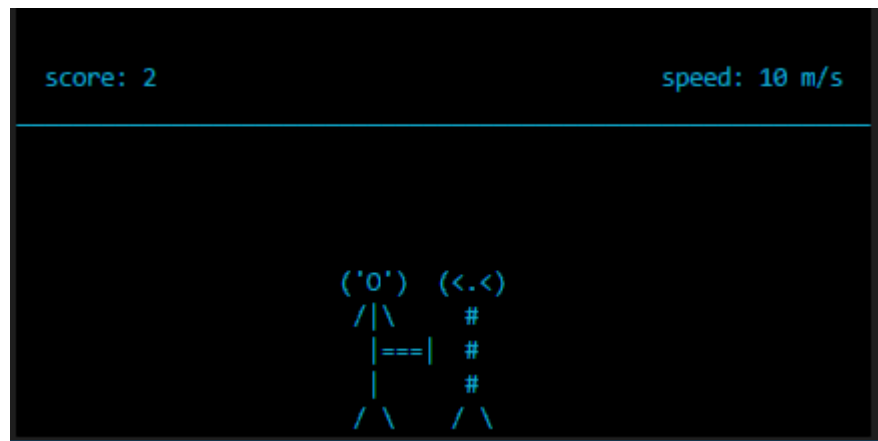


Image 3.3. Stickman kills enemies

Every 5 enemies killed, game difficulty is going to be harder. Player will see this with the acceleration of enemies and a color change. Image provided below.

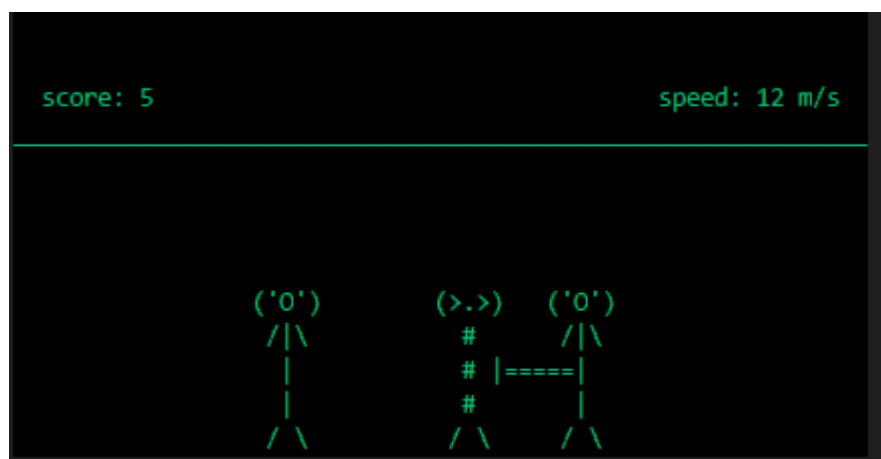


Image 3.4. Highest difficulty

When player lose, he/she will see the total score reached above. On this, game is over.



Image 3.5. Game over

4. Conclusions

There are couple possible improvements as more difficulty levels, more enemies' variety, add feature of collecting coins from single battle and then buy improvements for stickman.

There is a better way to make this game more responsible, is rewrite it using threads, but it will bring along many other problems and will take more time.

To conclusions, would be great to say, that all set goals were satisfied. Were improved knowledge of C language, especially it's library "ncurses". The made game, turned out very sophisticated, easy to use and funny.