2022-12-19
Author: Robert Okuła
robert.okula@pg.edu.pl

1 Basics of the Atari game Spy Hunter

In this project student is expected to write desktop application of an Atari game called **Spy Hunter**. The object of the game is to drive down roads and destroy other (enemy) vehicles, using the weapons that are available for the player. The project is based on the Atari 2600 version of the game. More detailed information is available on the Wikipedia. The example of the gameplay is available on Youtube.

2 General guidelines of the project

The assignment is accompanied by a sample project (template) in which the following functionality is implemented:

- calculating the increment of time, which allows to track the flow;
- displaying the graphics in BMP format;
- drawing a pixel, line, rectangle;
- text display.

The template uses "graphics" library SDL2 (2.0.7) – http://www.libsdl.org/. It is included in the startup project and does not need to be obtained (downloaded) separately.
The compilation is preformed using the command (on a 32-bit Linux operating system):
```
g++ -O2 -I./SDL-2.0.7/include -L. -o main main.cpp -lm -lSDL2 -lpthread -ldl -lrt
```
or (on a 64-bit Linux operating system):
```
g++ -O2 -I./SDL-2.0.7/include -L. -o main main.cpp -lm -lSDL2x64 -lpthread -ldl -lrt
```
In order to successfully build the template, in the same directory where the `main.cpp` file is located, additional resources should be places:

- bitmaps for drawing (e.g. `cs8x8.bmp`, `eti.bmp`). Notice the size of the letters in the filenames!
- `libSDL2.a` (`libSDL2x64.a` on a 64-bit system);
- `sdl` directory.

The project includes scripts that can be used for compilation (comp in 32-bit environment and comp64 in 64-bit environment).

The grading of the project may take place using the environment chosen by the student. There are two options:

- Linux – the student is required to check that the program compiles properly and run under the distribution installed on computers in the laboratory before the grading;
- Windows – using the MS Visual C++ environment available in the laboratory.

2.1 Program controls

The key controls should utilized in a specified way:

- `arrows`: moving the player in given direction;
- `esc`: quit the program;
- `n`: start a new game;
- `s`: save the game state;
- `l`: load the game state;

- `p`: pause/continue;
- `space`: shooting;
- `f`: finish the game.

2.2 Mandatory requirements (5 points)

All elements stated here have to be implemented. Lack of any of the following elements results in gaining 0 points from the project:

(a) The game board has to be presented in a aestethic and ergonomic way. The upper space of the window should contain the name, surname and index number of the author.

(b) Displaying the elapsed time and score during gameplay. Both values are reset when the new game is started.

(c) The basic functionality of the game: the movement, the road form (with collisions). The movement is immediate – responding to the events.

(d) Supporting `esc` and `n` controls according to the Section 2.1.

(e) The game should also keep the score according to the mechanics of the game described in the linked article.

(f) The right bottom corner of the game screen should contain the letters of implemented elements.

2.3 Optional requirements (11 points)

The tasks should be implemented according to the article linked in the Section 1 and additional materials:

(g) (**2 pt.**) *Saving and restoring game state.* Save and load game from file (keys `s` and `l`). Saved game is identified by the time of the save (date, hour, minute and second). When loading, the list of saved games is displayed and the user choosed the position from the list in a way chosen by the author.

(h) (**1 pt.**) *Pausing the game.* The player should be able to pause the game and then be able to return back to the play.

(i) (**2 pt.**) *Enemies and other cars.* The gameplay should include enemies that the player can chase, that can attack and overrun the player, but also regular, non-enemy cars that when destroyed halt the score counter for a while.

(j) (**1 pt.**) *Shooting.* The player can shoot both enemy and non-enemy cars, this might impact the score in a way described in the article.

(k) (**1 pt.**) *Forcing out of the road.* The player can force off the road both enemy and non-enemy cars, this might impact the score in a way described in the article.

(l) (**1 pt.**) *Getting some cars.* For a short period of time in the beginning the player should have the unlimited amount of cars (even if their car is destoyed, the one appears immediately). However, after that period, the number of cars should be limited and the new ones should be obtained according to the article. When the number of available cars is zero, the game should be over and a screen that informs about it should be displayed.

(m) (**1 pt.**) *Power-up.* Another weapon might appear on the road (This is a small simplification – no Weapons Van described in the article) and when collected (by riding on them) it temporarily (limited ammo) changes the way the shooting works (for example it allows for longer-range shooting).

(n) (**2 pt.**) *Keeping the score.* Store and display the best results' list. Single result consist of the number of points and the time of play. The player has ability to append to the list when exiting the game. the list is limited only by the memory size (dynamic memory allocation). Best results' list should be persistent (i.e. stored in a file and available between the launches). The menu allows you to display the list sorted by points (option `p`) and by time (option `t`).

2.4 Additional requirements (3 points)

Attention: the following requirements are graded only if all previous points (a)-(n) are implemented **for the overall amount of 16 points**.

(o) (**2 pt.**) *Two-player game.* The author should suggest the controls and the gameplay for two players (cooperation or player v. player?).

(p) (**1 pt.**) *Other power-ups.* The author should suggest another power-ups that can be beneficial to the gameplay.

2.5 Final remarks

- The use of the STL libary is prohibited.
- Compiling and launching the project is a required to get any points. The student must check whether the computer in the laboratory on which they are going to present it is equipped with the appropriate software.
- The project must be implemented by the student themselves. The student must be able to explain each element of the code and apply minor changed on demand.
- The game should be playable and easy to use.