Gaspard MATIC

Julien SEGONNE

IN 104 report

Tetris project

Our project aims to code a Tetris game in C programming language using the SDL library. We began with the structure of the code : we already had the programs .h and .c, the name of the functions with the variables, the definition of the different structures and variables, some filled functions and the makefile. In addition, we had to install the SDL library in order to get the graphic and keyboard functions to fill the program. Finally, we worked with the git environment to be able to work in group and separately.

There are some points of the project we think relevant and we want to highlight. Firstly, we had problems to display graphics from SDL : on the BYOD, we couldn't display the window and the playfield we coded, so we decided to move on the virtual computer with Ubuntu. This solution allowed us to have a look on what we were coding and it turned out to be a better support to code because it allowed us not only to work at school, on the BYOD, but also from a remote internet network. Even if we lost some time trying to fix the problem on the BYOD, it finally made us move forward very quickly because we could work during the holydays while checking by compilating and testing if there was no problem. Then we were able to code. One of the key point of our program was the way we handled the playfield : we used an array in which each slot represented a block (not the pixel, the entire block) of the playfield by its colour. We coded the function *draw_block* which took the coordinates (x,y) and the colour of the block as variables and drew the block on the window. In this way, each lapse of time the array was covered and all the blocks were drawn. The current tetromino isn't drawn until it reaches the ground, it is modelized by a structure containing the four coordinates (x,y) of the blocks and the colour. This way to treat the falling tetromino avoids a problem when we calculate if the future tetromino has a legal position in tetris.c. Then, we coded the movement of the falling tetromino : the translations left, right and down were quite easy to

implement but the rotation was more difficult. We had to stop coding for a bit and think : to make the tetromino rotate we applied mathematically a matrix of rotation on the coordinates, it gave the coordinates (-y,x) for each block of the tetromino. However, it brought us a new issue : for a not centred tetromino with the centre block in (15,30) for exemple, it gave (-30,15) so the tetromino is going far away. We found a solution : we centred the tetromino on (0,0), applied the matrix and then replaced the tetromino. That solution was the better we found, it appears in the function *updateTetris*.

Finally, we added two functionalities that we coded from scratch. We first added a hard mode to make the game more difficult and to change the style. We defined a global boolean named *hardMode* in the programs, when the key SPACE is pressed, its value changed to enter or leave that mode. Then some functions like *draw_Block* depend on the value of *hardMode*, in that function all the colours shift to the red. In addition, the speed of the game increases namely the tetromino falls, moves and rotates faster. To finish the beauty of our game, we added an original song during the game. We chose to install the extension SDL_mixer to deal with a mp3 file and to have a good sound quality. We first downloaded the song and added it to our folder. Then, in the function *init_graphics*, we initialized the library and loaded the song to play it after. We would have preferred to create a function fully dedicated to the audio but we lacked time, so we put it in a function already called and which as a .h file. But there was a problem when we compilated : the functions of SDL_mixer weren't recognized. So we added the command *-lSDL2_mixer* in the makefile in the compilating lines to link the library so that the computer is aware we used that extension. Finally, with more time we would have coded a functionality to replace the music with a faster one when the player switch to the hard mode to make the game more stressful and we would have added more sound effects like when a line disappears or when the game is over.

To conclude, this project taught us both computing skills (on SDL principally but also on the way we think our code) and general skills. We learnt how to use the git environment, we faced time problems and deadlines, and we developed our team working and our communication. Finally, we managed by ourselves by beginning with almost nothing.