Jeu de Taquin

Jimmy Teillard - Université Gustave Eiffel

November 21, 2020

- Introduction
- How to use
- Architecture
- Implementation choices
- What I learnt

Introduction

The "Jeu de Taquin" is a mathematical game that consists in reassembling a scramble board by sliding pieces one by one.

For that, we take an image, and remove a portion of it, then scramble the board so that it is actually solvable in the end.

How to use

1. Compile the program:
   On a *nix system, preferably a Linux distribution, go in the root folder of the projet, and run `make`. Make sure the libMLV is installed on your system. If you wish to clean the folder, run `make rmproper`.

2. Run the program:
   In this same folder, run `./taquin` to launch the game.

3. Navigate between the menus until starting the game.
   You'll find yourself facing a board that will be scrambled.

4. Solve the puzzle using your arrow keys to move the pieces around.

Architecture

In an idea to follow a MVC (Model-View-Controler) architecture, my project is structured as is. I have two main modules : * model * viewctrl

The goal is to properly separate the backend from the front end to avoid confusion and allow for easier debuging, while avoiding to mix MLV's functions everywhere.

In this case, I fused the View and the Controler into one, since the actual view here are the MLV functions themselves. Also, in the case of games, it isn't that uncommon to use a modified version of MVC (such as MVVM).

`main.c`'s goal is simply to declare important variables such as the board, and launch the game properly.

I separated every file in different semantic folders (src, include, bin) to avoid clustering the root folder of the projet.

Implementation choices

As said before, the choice of MVC is something that I used because it allows me to better organize myself ; and as such, let everyone read my code properly. There's also the fact that I'm used to it.

In the game, the board is represented with a struct that contains squares, which themselves are also structs that only contain coordinates. I chose to implement the board like that just because it is probably the best compromise between clarity and simpleness. A regular two dimensional array of integers would have actually sufficed, but, to me, it's still important to have a well structured code that is readable at first glance even without reading comments.

The way I scramble the board is very simple : I just do random moves at the start of the game so that the board is always solvable. I know there are actual algorithms to initialize a board already scramble AND it to be solvable, but, in this context, where it's not very costly, I chose the first solution…and eh… it makes for a nice animation at the start !

What I learnt

First and foremost, I've never used libMLV even once before this project, since I didn't spend my last year here at UGE ; so this is a tool that I learnt to use on the go.
Though I must say that learning very quickly a tool by myself is not something new to me. The real difficulty was installing the library to be honest.

The project itself didn't allow me to learn much, but it actually probably helped me a lot in applying advises my teachers told me regarding the C language and it's conventions.