BEN HAMIDA Ghassen

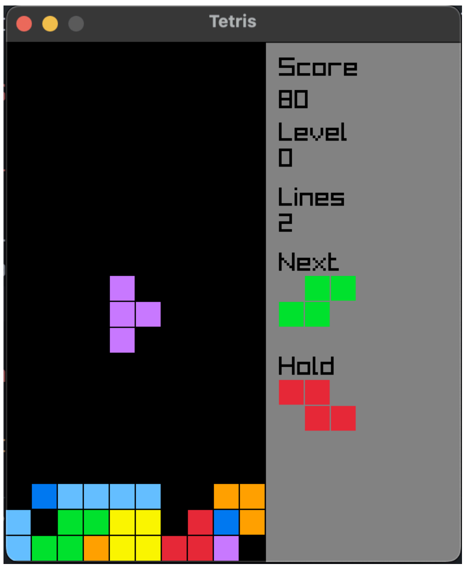
ENSAE 2024-2025

C++ Project Report

Tetris

Description of the Project

Tetris is a classic tile-matching puzzle video game originally developed by Russian game designer Alexey Pajitnov in 1984. The game consists of different geometric shapes called *tetrominoes*, which are made up of four blocks. The player's goal is to manipulate these falling shapes to fit them into a grid, with the aim of forming complete horizontal lines. Once a line is filled, it disappears, and the player earns points. The game ends when the stack of blocks reaches the top of the screen, preventing any more tetrominoes from falling.



This project is an implementation of the classic Tetris game using **C++** and the **Raylib** graphics library. The goal of the project was to create a functional Tetris game retro-themed, complete with user input, and various game mechanics. Many game features have been implemented in this project including the basic rules of Tetris (right and left movement, rotation), holding, hard drop, soft drop, lock delay, speed increase with the level reached as well as line clearing and a scoring system.

Architecture

There are many libraries and frameworks to choose from for graphic display, but Raylib was chosen as the library for Tetris game development due to its simplicity and intuitive usage. It is also a cross-platform, open-source library. In the words of the creator of Raylib: "A simple and easy-to-use library to enjoy video games programming."

At the beginning, as I was experimenting with Raylib, I started by implementing only the I Tetromino falling down with a basic display. After that, I added the other Tetrominoes. From there, I continued adding a feature at each step and testing it: rotation, collision, locking pieces to the board, the bag system for Tetromino generation, line clearing, and other advanced Tetris mechanics introduced in the "New Tetris," such as hard drop, displaying the next Tetromino, the holding feature, and lock delay—unlocking new possibilities in the game. Additionally, a scoring system and a falling speed that increases with the level were implemented.

Initially, I intended to follow a **Model-View-Controller** (**MVC**) architecture. However, I decided to focus first on better understanding the game and Raylib. Later, when I tried to apply MVC architecture to the project in its final state, it proved not to be difficult due to the well established object-oriented design already in place.

The project features Tetromino (a base class for the 7 different Tetrominoes) and Board models, along with some helper classes (struct). There are views for each model, as well as a SidePanel view for displaying information about the score, level, next Tetromino, and hold functionality. Given the relatively small size of the application, a single controller handles user input and manages the connection between the data in the models and their respective views.

Problems, critics and solutions :

There is no official documentation for Raylib. To learn it, I referred to examples on its official website and some GitHub repositories. I learned it quickly because of its intuitive approach.
However, this simplicity comes at a cost. If detailed customization of the interface is required, it becomes quite complicated, and opting for another

library or framework like SDL2, which offers a low-level API, might be a better choice—though its learning curve is steeper.

Possible improvements

- Add sound effects.
- Add a scoreboard.
- Allow setting the desired level (and speed) from the beginning.

These features can be implemented easily due to the well established architecture of the project; it's just a matter of time.

Installation Guide

1. Clone the repository from https://github.com/yourusername/tetris.git.
2. Install Raylib and Make.
3. Compile the files.
4. Run the application.

Check the README file for more details.

How to play

The tetrominoes start falling down automatically. You can:

- Move them right or left using the right and left arrow keys, respectively.
- Rotate the piece clockwise using the up arrow.
- Drop it faster (soft drop) using the down arrow.
- Perform a hard drop with the space bar.
- Hold a piece by pressing the C key.

You increase your score by forming full horizontal lines. The points earned increase exponentially with the number of lines cleared simultaneously. Thanks to lock delay, you can still move the tetromino for 0.5 seconds after a collision.