DR. MINIX – TILES LAB

2LEIC02_3 LCOM 24/25

Developed by:

Ana Santos up201905129
João Júnior up202306719
Mário Pereira up202304965
Vasco Sá up202306731

Table of Contents

1. What was our goal? What is our application? 3

2. How did we structure the project? 4

3. What devices did we use and for what purpose? 5

4. What are the differentiating features of our project? 5

1. What was our goal? What is our application?

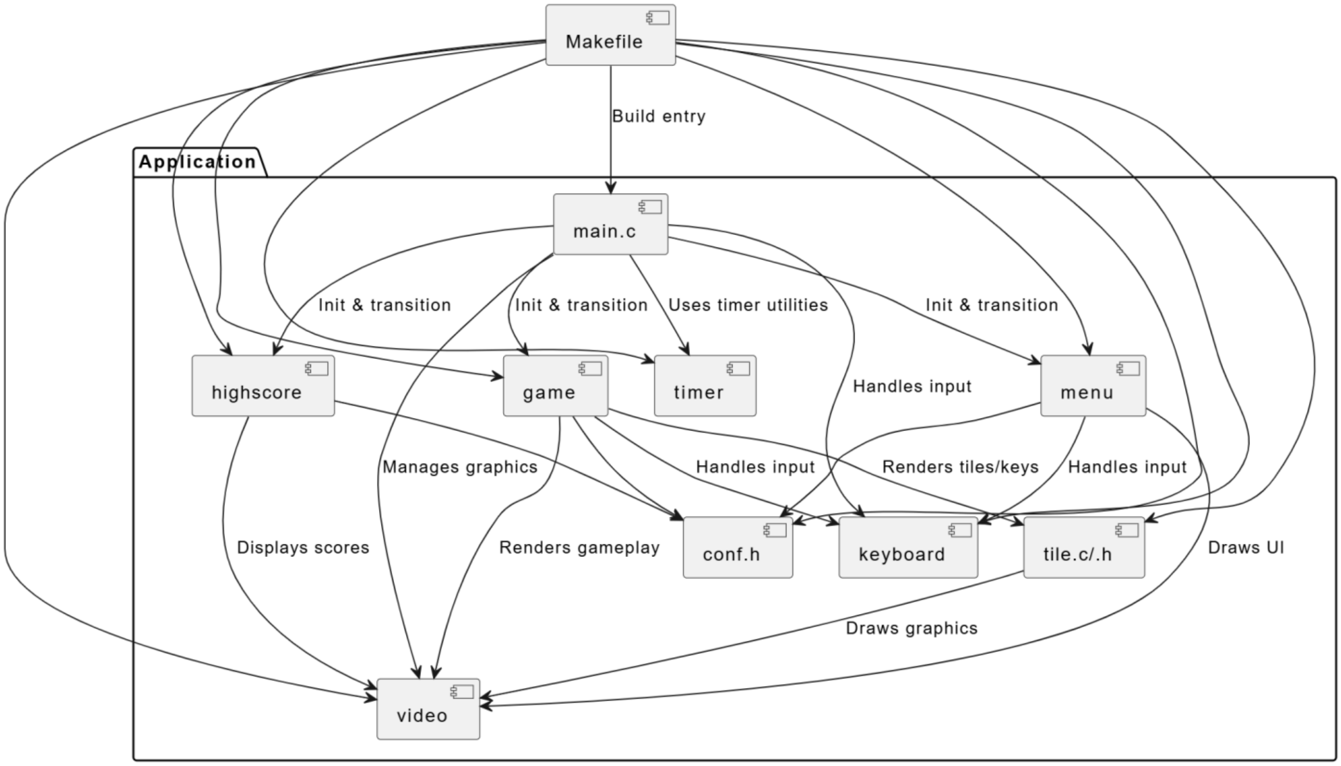
We developed Dr.Minix – Tiles LAB, a graphical rhythm-based game application for the MINIX operating system as part of the LCOM (Laboratório de Computadores) course with the goal of deepening our understanding of low-level programming, device interaction and graphics programming. The game includes a main menu with options, gameplay, and a highscore system.

Objectives included:

- Learning hardware interaction: We wanted to gain experience with device drivers by directly programming the keyboard controller, timer, and graphics card using C and MINIX system calls.
- Developing a graphical user interface: The application features a custom graphical interface, including a main menu, options menu, game screen and a highscore screen, all rendered using VBE (VESA BIOS Extensions) graphics mode.
- Making gameplay feel live: We designed the game logic to react instantly to user input and timer events to make everything feel smooth and interactive.
- Keeping the code organized and easy to work with: by splitting the project into modules (like menu, graphics, and game logic) we wanted to make the code easy to understand and simple to update in the future.
- Creating a fun experience: Our main goal was to make the game enjoyable, with features like two tile types, different difficulties and highscores that keep players engaged.

2. How did we structure the project?

The project is organized into several distinct modules, each responsible for a specific aspect of the application:



Architecture image generated by GPT-4.1

- main.c

  Initializes the system, manages the main loop and transitions between menu, game, and highscore screens.
- timer/

  Handles the timer interrupt and time-related utilities.
- keyboard/

  Contains the keyboard controller (KBC) and handles inputs, including scancode processing.
- video/

  Manages graphics and video mode, draws primitives and handles framebuffer operations.
- menu/

  Main menu and options menu logic, including drawing and user interaction.

- game/

  Renders core game logic, state management, and gameplay.
- highscore/

  Manages and displays the highscore.
- tile.c / tile.h

  Handle the logic and drawing of tiles and piano keys, including their appearance, movement and interaction in the rhythm gameplay.
- conf.h

  Project configuration constants.
- Makefile

  Specifies how to compile the project, which files to include, and what libraries to use. Follows Minix LCOM standards and connects with the LCF, XPM, and LM libraries.

3. What devices did we use and for what purpose?

- Keyboard Controller (i8042/KBC): For user input and game controls.
- Timer (i8254): For managing game timing, animations, and periodic events.
- Graphics (VBE): For rendering the game, menus, and highscores in graphical mode

4. What are the differentiating features of our project?

Our project stands out because it gives players a lot of control over their experience, all from a single, interactive options menu. Users can switch between dark and light modes as well as change the screen resolution making the whole interface instantly adapt to their choices.

You can also pick between two tile types and set the difficulty level, which directly changes how fast the game runs, how often notes appear, and how many points you can score. All these settings are easy to adjust, and the menu highlights your current selection.