MSc Data Science
Extended Research Project Plan

Student name	Anke Du
Student number	11539096
Project Title	Game Boy Emulator Development
Supervisor	Bernardo Magri
Project partner (if applicable)	

Non-technical project summary (up to 150 words)

This project entails the creation of a software emulator—an application that simulates the functionality of physical hardware devices solely through software. The emulator will be constructed to simulate the core functions of a basic handheld gaming console, encompassing its processor unit, display system, audio channels, and user input methods.

The primary elements to be replicated encompass the system's processor, graphics rendering, audio channels, and user input. The project's emphasis lies not on the Game Boy itself, but on the design and implementation of a working emulator that accurately replicates a genuine hardware device.

The project provides practical experience in systems programming and low-level architecture, addressing subjects such as instruction decoding, memory mapping, temporal synchronization, and components integration. It seeks to illustrate how intricate physical systems can be accurately and modularly represented in software.

The Aim of the project and up to four Research questions or Objectives

Aim:

To develop and execute a modular software emulator that can replicate the internal architecture of an actual hardware system. The Nintendo Game Boy exemplifies fundamental principles in systems development, architectural modelling, and emulation validity.

Objectives:

- Accurately implement the target system’s CPU instruction set, memory model, and interrupt logic.
- Recreate the graphical and audio subsystems to produce faithful visual and sound output.
- Design input handling and a minimal user interface to support real software execution.
- Evaluate the emulator’s performance and correctness through structured testing using reference ROMs.

Description of Methodology to be used (up to 400 words)

This project will utilise a bottom-up implementation technique in order to develop a software emulator which provides a representation of the internal architecture of a real hardware system. The Game Boy will serve as a practical target for emulation due to its well-documented hardware design and relatively simple architecture.

The development process will start with the central processing unit, which will include the implementation of the Sharp LR35902 instruction set, registers, memory access modes, and interrupt mechanisms. Standard test ROMs and community-verified debugging tools (like BGB and SameBoy) will be used throughout the entire process.

For graphics module, this project will imitate an image rendering process similar to that of the Game Boy display, which encompasses the creation of dynamic sprites and background tiles, the implementation of colour palettes, and the advancement of scrolling functionalities. Also, the system employed a scanline-based rendering technique to generate the image sequentially, line by line, in synchronisation with the console's timing. A framebuffer will be utilised to aggregate rendered pixels for presentation on a contemporary display.

The audio system will recreate the Game Boy Color’s four sound channels: two square wave oscillators, one programmable wave channel, and one noise generator. These will be generated using waveforms and synced with system timing. Real-time audio output is not necessary for basic functions, but it improves emulation and user satisfaction.

User input will be coupled with emulator virtual controllers, which will imitate button pushes using keyboard events. Input values will modify memory-mapped registers in the same manner as the original hardware processed controller input. Timing accuracy and debouncing techniques will be assessed to guarantee responsiveness throughout gameplay.

Upon the completion and testing of all individual components, they will be integrated into a unified emulator application. The accuracy of the emulator will be assessed using common test ROMs that validate memory operations, interrupt management, and instruction execution. Its compatibility with graphics, audio, and input control will also be evaluated by comparing its runtime performance to well-known, accurate emulators. Performance will be evaluated based on temporal consistency, CPU usage, and frame rendering speed. Visual and audio outputs will be evaluated as necessary to ensure hardware fidelity. Optional features like save states, debugging tools, or frame stepping could be looked into, depending on time and progress.

Project timetable up to 250 words (and a diagram which can be added as a separate page)

The project will be completed in approximately 19 weeks, starting in April 2025, and will be divided into three phases: development of the main modules, integration and evaluation, and final delivery preparation.

Stage 1 (Week 1-7) begins with CPU emulation, then graphics rendering, input handling, and a basic ROM-loading interface. These components provide the basic layers required for a functional emulator and are built step-by-step, with each step reliant on the completion of the previous one.

Stage 2 (weeks 8–13) aims to integrate the system as a whole. Performance will be assessed and improved after audio emulation is constructed and synchronised with CPU timing. To ensure correctness and consistency, all subsystems will be verified against test ROMs.

Stage 3 (Week 14-18) emphasises documentation, optional feature improvements (such save states), and last submission. Early August will see completion of the project report and demonstration video in line with the general submission date.

Planned without overlap and following a strict dependency chain, all tasks ensure controlled and traceable development. Graphically complementing this timetable was a Gantt chart with week-level granularity and dependency arrows.

	Task Name	Duration	Start	ETA	Timeline (Unit: Week, starting from 2025/4/1)
					1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1	CPU Emulation	21 days	2025/4/1	2025/4/21	
2	Graphics Rendering	14 days	2025/4/22	2025/5/5	
3	Input Handling	8 days	2025/5/6	2025/5/13	
4	Basic UI & ROM Loader	8 days	2025/5/14	2025/5/21	
5	System Integration	8 days	2025/5/22	2025/5/29	
6	Audio Emulation	8 days	2025/5/30	2025/6/6	
7	Performance Testing	8 days	2025/6/7	2025/6/14	
8	Evaluation & Bug Fixing	15 days	2025/6/15	2025/6/29	
9	UI Polish & Optional Features	11 days	2025/6/30	2025/7/10	
10	Final Report Drafting	14 days	2025/7/11	2025/7/24	
11	Video Preparation	8 days	2025/7/25	2025/8/1	
12	Final Submission Packaging	7 days	2025/8/2	2025/8/8	

Describe any risks to the project succeeding and how you will manage them (150 words)

A primary risk in this project is the technical intricacy of low-level hardware emulation, especially with the CPU, memory interactions, and timing-sensitive graphical output. To address this, development will adhere to a modular framework with unit testing conducted after each milestone. Any inconsistencies will be identified and rectified inside each module prior to proceeding.

Another concern is scope creep, wherein extra features such as save states or debugging tools may detract time and focus from the primary functionality of the emulator. The implementation of optional features will be postponed until the core system is stable and on track.

Ultimately, evaluating emulation accuracy is intrinsically challenging in the absence of a definitive benchmark. The test output will be compared to established emulator behaviour and test logs for management purposes. All differences will be recorded and utilised to inform further enhancements.