A logo for a university

AI-generated content may be incorrect.

Cart-iologist: Functional Specification

For submission to SETU

Author: Stuart Rossiter

Student Number: C00284845

Course: BSc (Hons.) Software Development

Supervisor: Joseph Kehoe

Contents

[1. Introduction 3](#_Toc210586136)

[1.1 Target Users 3](#_Toc210586137)

[1.2 Technologies 3](#_Toc210586138)

[2. Functional Requirements 4](#_Toc210586139)

[2.1 Core Requirements 4](#_Toc210586140)

[2.2 Non-Core Requirements 4](#_Toc210586141)

[3. Context Diagram and Use Cases 5](#_Toc210586142)

[3.1 Context Diagram 5](#_Toc210586143)

[3.2 Use Case Diagram 5](#_Toc210586144)

[3.3 Brief Use Cases 5](#_Toc210586145)

[3.3.1 Dump Save 5](#_Toc210586146)

[3.3.2 Restore Save 5](#_Toc210586147)

[3.3.3 Swap Cartridge Port 6](#_Toc210586148)

[3.3.4 Insert Cartridge 6](#_Toc210586149)

[3.3.5 Remove Cartridge 6](#_Toc210586150)

[3.3.6 Check Health 6](#_Toc210586151)

[4. Metrics 7](#_Toc210586152)

# Introduction

The Cart-iologist is a diagnostic tool designed to check the overall health of retro game cartridges, and manage their saves through a GUI based PC program. Designed for the purpose of retro game preservation, the Cart-iologist will use an Arduino MEGA 2560 to interface with the cartridges, through cartridge port modules. These modules will be able to be swapped in and out, allowing for a wide variety of cartridge types, and future expansion.

## Target Users

The Cart-iologist will be designed for retro gaming enthusiasts and preservationists. Retro gaming enthusiasts would want to ensure that their video games remain playable, and would be interested in backing up their save games in case something does go wrong. Preservationists keep all sorts of cartridges in working order, preserving the hardware with the software, so they would make use of the Cart-iologist’s diagnostic capabilities.

## Technologies

To serve as an interface between the cartridge port modules and the PC, a microcontroller will be required, in this case an Arduino. Due to the number of pins required for cartridge ports such as the Game Boy Advance, or the Nintendo 64, an Arduino MEGA 2560 would be ideal, due to the sheer number of digital I/O pins on the board. Alternate boards would be usable, but would require latches or multiplexing. For the GUI based PC program, Python would be sufficient, as it can interface with the Arduino directly over serial, and the Tkinter library provides GUI functionality. Various electronic components will also be required, listed below:

* 4x 8 bit latches, for stabilisation and dealing with the N64 cartridges high speeds.
* 2x 8 bit transceivers, for 5v to 3.3v logic conversion.
* USB-C port. The Arduino has a max mA output, so a separate USB will be used to power the cartridges.

# Functional Requirements

The functional requirements for the Cart-iologist are separated into 2 groups: Core and Non-Core.

## 2.1 Core Requirements

* Users should be able to run diagnostic tests on inserted cartridges.
* Users should be able to dump the save data from their cartridges.
* Save Data should be able to be flashed back to the cartridges.
* Users should be able to view and manage their various save files from the GUI program.
* Cartridge ports should be able to be removed, and swapped out with other cartridge port modules.

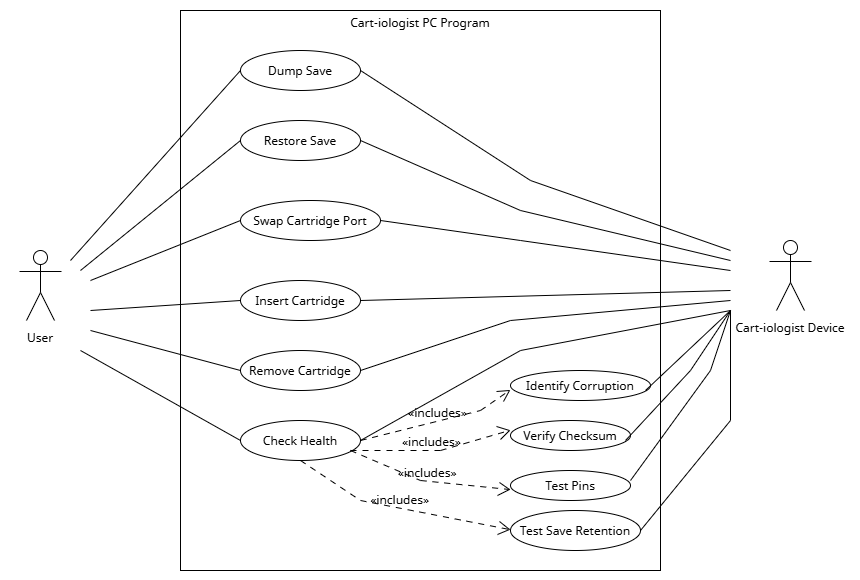
## 2.2 Non-Core Requirements

# Context Diagram and Use Cases

## Context Diagram

To be added.

## Use Case Diagram



## Brief Use Cases

### Dump Save

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to dump the save file from the inserted cartridge. The user selects “Save Manager” from the sidebar. From there, the user selects the “Dump Save Data” option. The use case ends when the save data has been dumped from the cartridge.

### Restore Save

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to restore a dumped save to the inserted cartridge. The user selects “Save Manager” from the sidebar. From there, they select the desired save, and press “Restore Save Data”. The use case ends when the save data has been restored to the cartridge.

### Swap Cartridge Port

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to swap the inserted cartridge port in the Cart-iologist device with a different one. The user selects “Port Swap” from the sidebar, and clicks “Eject Port”. Once prompted that the port is safe to remove, the user removes the current port module, and inserts a new one. This use case ends when the device detects the new module.

### Insert Cartridge

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to insert a new cartridge into the Cart-iologist Device. The user inserts the cartridge into the inserted port module. This use case ends when the program recognizes that a cartridge has been inserted.

### Remove Cartridge

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to remove the current cartridge from the Cart-iologist Device. The user removes the cartridge from the inserted port module. This use case ends when the program recognizes that no cartridge is inserted.

### Check Health

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to check the health of the inserted cartridge. The user selects “Health Check” from the sidebar. The user then selects the tests they wish to run, and clicks “Run Test(s)”. This use case ends when the results of the tests are returned to the user.

### Identify Corruption

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to identify any corrupted data within a cartridge/memory card. The user selects “Health Check” from the sidebar. The user then selects “Identify Corrupted Blocks”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Verify Checksum

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to verify the checksum of a cartridge ROM against a known dump. The user selects “Health Check” from the sidebar. The user then selects “Verify Checksum”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### Test Pins

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to verify the checksum of a cartridge ROM against a known dump. The user selects “Health Check” from the sidebar. The user then selects “Verify Checksum”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

### 3.3.10 Test Save Retention

Actors: User, Cart-iologist Device

Description: This use case begins when the user wishes to test the save retention of the inserted cartridge. The user selects “Health Check” from the sidebar. The user then selects “Test Save Retention”, and clicks “Run Test(s)”. This use case ends when the result of the test is returned to the user.

# Metrics