Pokémon Randomizer

BACS 485 - Final Paper

Simon Warga, Owen Wamsley, Justin Moehlenpah, Touchsada Jan On, Michael Saenz

Glossary:

Pokémon FireRed: This is a game in a series of games called Pokémon, developed by Gamefreak and published by Nintendo. This game was developed for the Game Boy Advanced, a handheld video game console also put out by Nintendo.

ROM: ROM is an acronym for Read-Only Memory. It refers to computer memory chips containing permanent or semi-permanent data.

- In our case, this is a game file ripped from a GBA cartridge and placed onto the computer for editing and playing in an emulator.

Emulator: An Emulator is a piece of software that is meant to emulate and act as the hardware for a video game console such as a Game Boy Advanced.

- The emulator we will be using is mGBA which is an open-source Game Boy Advanced emulator

Pokémon-TM: TM is an acronym for Technical Machine. It refers to an in-game CD that a trainer can give to a Pokémon to teach it a new move. They can only be used once.

Item: An item is an object in Pokémon that a trainer can pick up and use in some manner

<u>Section I Planning and Project Management - Pokémon Randomizer:</u>

Description: A Pokémon randomizer will give a user a unique experience playing Pokémon. We will be creating a randomizer for Pokémon FireRed to be used on an emulator (mGBA). This randomizer will allow the user to customize features of the game such as wild Pokémon appearances and different starter Pokémon. Along with many more customizable options, a user will be able to play Pokémon FireRed in their own creative way.

Simon Warga (Project Manager and Business Analyst)
- Keeping contact with sponsor, and coordinating project and R&D

Owen Wamsley (Software Developer)
- Implement randomization elements and save the changes to the game file.

Michael Saenz (Network Architect and Security Engineer)
- Understanding encryption and decryption of the ROM

Justin Moehlenpah (Web Developer and Database Architect)
- Responsibility: Implementation of randomization elements

Touchsada Jan On (UI/UX Designer)
- Responsibility: Building GUI interface of each randomization feature

<u>Project / System Request</u>

Sponsor - Richard Luna (Simon's friend)
Need - creating extension for entertainment
<u>Requirements:</u>
Functional
1. User should be able to interact with a GUI to modify settings
2. User should be able to select components of game to randomize
3. Randomized Elements:
    a. Starter Pokémon
    b. Wild Pokémon, including event Pokémon
    c. All other in-game Pokémon, including trades and prizes
    d. Trainer Pokémon
    e. Trainer titles (like 'Lass' or 'Pokemaniac')
    f. Items found on the map, including hidden items
    g. TM attacks
    h. Pokémon-TM compatibility
    i. Pokémon themselves, including types, stats, palettes, abilities, wild hold items, and movesets
4. Text Document stating randomized elements

Non-functional
5. Randomization shouldn't slow down the game

6. Randomization should be completed before the startup of the game

Value - System allows for a user to modify elements of a game to their liking, creating a personalized experience that could be utilized for live-streaming, video content, or other personal uses. This system directly impacts the gaming industry; it allows for diehard fans of a game to experience something different that viewers would find entertaining.

i. Tangible - user able to modify the game introduce creativity, which can lead to tangible profit
ii. Intangible - Helps the user become more creative

Constraints:

- Pokémon game file must be provided by user
- Offline functionality

Level of Effort (LOE):

Small: 4 hrs
Medium: 10 hrs
Large: 20 hrs
Extra Large: 80 hrs

User Stories

- As a user, I want to be able to randomize Pokémon Fire Red through a GUI so I can easily change my randomizer settings. (**Medium**)

- As a user, I want to be able to customize what I randomize so I can customize my game experience to my tastes. (**Small**)

- As a user/player, I want to be able to randomize the game's starter Pokémon and their item so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize the games wild Pokémon encounters so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize the games NPC trainers and their Pokémon so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize the game's rewards for defeating NPC trainers so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize the game's items found on the map so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize Pokémon's TM compatibility so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize Pokémon's types, stats, palettes, abilities, wild hold items, and move sets so I can get a unique gaming experience. (**Medium - Large**)

Tasks

- Learn how to edit Pokémon Fire Red's game file. (**Large**)

  Total Hours: 104 hours

Pokémon Randomizer Executive Summary

The feasibility analysis for the Pokémon Randomizer Project has been done with a detailed feasibility study. The highlights of the feasibility analysis are as follows:

Technical Feasibility Analysis

The Pokémon Randomizer Project is technically feasible, although there are some risks.

Pokémon Randomizer regarding familiarity with a user and the application is moderate

- The front-end and back-end developers will be using Java programming language to implement the code and build the application.
- The implementation contained some unfamiliarity regarding modifying the Pokémon game file and its database and accessing the ROM file type, where the game will be executed from.
- Business user involvement will be essential because the developers will need the specific characteristics/features when modifying the game in order to accurately implement the application

The risk regarding the familiarity of technology using Pokémon Randomizer is low

- Front-end and back-end developers have extensive knowledge regarding the implementation between the GUI component and the logical interface of the game using a Java IDE from the available open-source software.
- The application will be implemented as an executable file, therefore the technology used to execute the application file will be similar to the device that the user uses to execute the game file. (i.e. local PC machine)

The project size is considered to be moderate

- The project scope will consist of building the GUI component that interconnects with the backend logic when the user decides to randomize features in the Pokémon game. (i.e. customizing the game's victory rewards)

- The project team will likely consist of 2 teams of 2-3 people

The compatibility with Pokémon Randomizer existing technical infrastructure is good

- Since the application will be an executable Java file, the application will be able to run on existing technology infrastructure that every user has available, such as their personal computer.

Economic Feasibility
The entirety of this project is meant to be nonprofit. Our sponsor is not funding this with any actual capital and the only costs we are going to incur will be intangible and thus would be very difficult to quantify. One example of an intangible cost would be the time spent learning how to modify the ROM and implement it into Java scripting.

Organizational Feasibility
From an organizational perspective, this project has a moderately high risk.

- **Top management support:** The top executive of the company strongly support the project
- **Project champion:** Richard Luna (Simon's friend) is passionate about the project and delivers a clear requirement that proposes benefits to the overall organization's goal as he shares the functionalities with his peers and his social community.
- **Organizational management:** Overall, organizational management such as IT support and software development managers support the idea of the *Pokémon Randomizer Project.* However, the customer services department may experience an increased amount of user traffic as the application is new and can be overwhelming for novice users.
- **System Users:** The user will expect a user interface that is clear and simple as they navigate throughout the application. The visual hierarchy and Gestalt principles such as the concept of proximity will be used, so the user can scan the feature and accomplish their goals quickly. The logic will accurately reflect the functionality of the GUI components, so the user can randomize the Pokémon game without stress and frustration. The project will require regular feedback from the user in order to improve each iteration and perfected the overall project before release.

Additional Comments

- The *Pokémon Randomizer Project* will allow the organization to extend the gaming experience for gamers and reach new audiences. As the gaming industry grows, there will be an increasing number of gamers who would like to customize the game that they like and regain the new and more exciting experience.

- To enhance the acceptance of the new Pokémon randomizer application, the organization will focus on the gamers as the main group of audience. These groups will spread the use of the application through their peers and social group, which leads to benefit the organization's overall goals.

Project Plan

Approach: We will take an iterative approach, where we add in features after getting the main program with basic functions added in. This will allow us to remain flexible if we need to change our scope during this project if we did not estimate time accurately. We will test the software after each implementation of randomization elements.

Tasks and Timeline (WBS): This project will span the duration of the Spring semester of 2022. Dates are subject to change

1. Create file opener and editor (1/10-1/31)
2. Create Main Application interface (2/1-2/11)
3. Prioritize randomization elements (2/11-2/13)
4. Create Pokémon database (2/14-2/24)
5. Create starter Pokémon randomizer (2/25-3/10)
6. Implement Wild Pokémon spawn randomization (3/11-3/18)
7. Implement Field Item Randomization (3/19-3/26)
8. Implement Pokémon Stat randomization (3/27-4/4)
9. Implement Pokémon move randomization (4/5-4/10)
    a. Pokémon move compatibility randomization.
10. Project testing and cleaning (4/11-5/1)

<u>Section II Analysis:</u>

Overview: A user will insert the ROM into the randomizer, then have the ability to modify the game their own way. They will be provided with a tabular GUI where they can change components of the game. The user will then save this new randomized ROM to the preexisting ROM. They can then open an emulator appropriate to the Pokémon FireRed game, and open the new ROM and begin playing.

<u>Requirements Interview Notes:</u>

- Randomize Pokémon abilities, location stats, and items
- Most important features:
    - Playing the game again in a different way
    - New and unsuspecting

        - Randomize a lot
        - Focus on replayability
            - Stats and Abilities randomizing
            - Change Type for the Pokémon?
        - Randomize what Pokémon shows up where
    - What do you expect to have to do to use the randomizer
        - Tabular
        - Give info for each option
        - Options for detailed customization
        - Main tabs functions with the main elements
    - Any type, anywhere
    - Option for dark or light mode
    - **Most important feature:** be able to use the entire list of Pokémon and be able to randomize EVERY Pokémon
    - **Lowest priority:** UI looks and feel
    - Functional Requirements rank (Most to Least priority)
        - Pokémon themselves, including types, stats, palettes, abilities, wild hold items, and movesets
        - Wild Pokémon, including event Pokémon
        - Starter Pokémon
        - Trainer Pokémon
        - All other in-game Pokémon, including trades and prizes
        - Items found on the map, including hidden items
        - TM attacks
        - Pokémon-TM compatibility
        - Trainer titles (like 'Lass' or 'Pokemaniac')
- UI must have buttons
- Should create a copy of the ROM before randomizing so that it keeps the original
- Insert your own seed
- Set a default path for saving ROMs
- Add the funny monkey maybe? (Easter Egg)
    - if the user configures to a very specific set of settings, the funny monkey appears

<u>Requirements:</u>

Functional Requirements (Prioritized by the Client)

1. The user should be able to select components of the game to randomize

    a. The user should be able to see the Pokémon's types, stats, palettes, abilities, wild hold items, and movesets
    b. The user should be able to pick their own wild Pokémon
    c. The user should be able to pick their own starter Pokémon
    d. The user should be able to pick what Pokémon trainer character they want to use
    e. The user should be able to pick what prizes they can win
    f. The user should be able to choose what item they can pick up on the map
    g. The user should be able to choose what item is hidden and can be picked up on the map
    h. The user should be able to choose the TM attacks to train their Pokémon
    i. The user should be able to choose the TM compatibility for the Pokémon
    j. The user should be able to change the name of the Pokémon trainer that was chosen.
    k. The user should be able to randomize what Pokémon show up where throughout the game

2. The user should be able to see the list of all of the Pokémon
    a. The user should be able to randomize every Pokémon on the list
3. The user should be able to interact with a GUI to modify settings
    a. The user should be able to modify the GUI setting to a bright or dark theme
4. The summary text of what component the user had picked to be randomized
    a. The user should be able to see what components they have randomized in the GUI text output box.
5. The user will provide a Pokémon Red ROM file for the system to use.
6. The system will use a hex editor to edit the ROM file to make a patch.
7. The system will edit the ROM file based on user randomizer selections.
8. The system will patch the newly created patch into the ROM.
9. The user should be able to see a set default path for saving the modified ROM file.
10. The system will create a new patched ROM file based on the user's selected randomizer settings.
11. The system will create a copy of the unedited ROM file before editing the file.
12. The system will assign a seed variable to each created ROM file.
13. The user can enter a seed value to replicate a randomized ROM.
14. The GUI should have tabs to break apart randomizer elements.
15. The GUI should have options for dark mode and light mode.

Non-functional Requirements

1. **Operational**
    a. The application should be an executable file, so the user can run the file on any computer
2. **Performance**

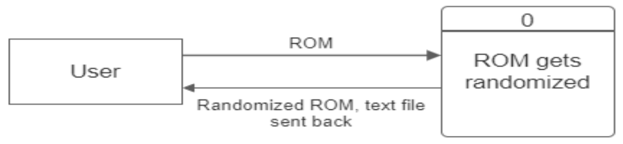
   a. The file should execute in less than 5 second
   b. The interaction between the user and the application should be instantly responsive and have no delay.
   c. The application should be closed and stop running when the user closes the application
3. **Cultural and Political**
   a. The application will be built using Java and any necessary script plugins such as a hex editor

<u>User Stories:</u>

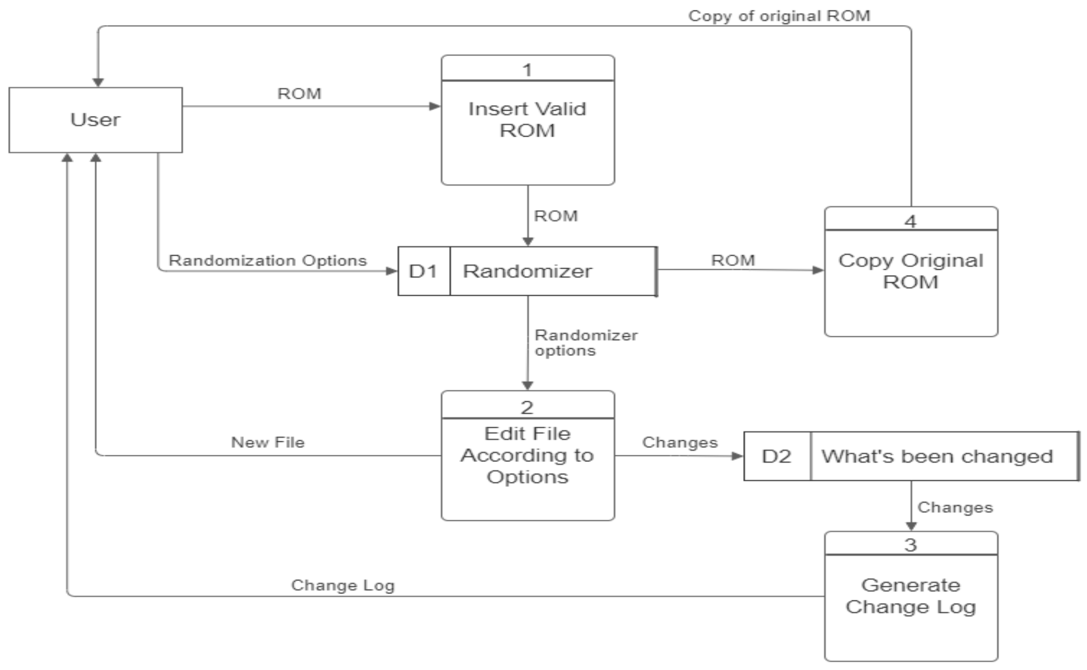
- As a user, I want to be able to randomize Pokémon Fire Red through a GUI so I can easily change my randomizer settings. (**Medium**)
  - The GUI must have buttons
  - The GUI should have tabs to break the randomizer into sections
  - The GUI should have dark and light mode
- As a user, I want to be able to customize what I randomize so I can customize my game experience to my tastes. (**Small**)
  - The GUI will have selectable radio buttons to select which elements to randomize.
- As a user/player, I want to be able to randomize the game's starter Pokémon and their item so I can get a unique gaming experience. (**Medium**)
  - Have the option to have any type of Pokémon be a starter
  - Have the option to have only 3 stage Pokémon be a started (only Pokémon with two evolutions)
- As a user/player, I want to be able to randomize the games wild Pokémon encounters so I can get a unique gaming experience. (**Medium**)
  - Have option to do fully random, 1-to-1 global, and 1-to-1 area
- As a user/player, I want to be able to randomize the games NPC trainers and their Pokémon so I can get a unique gaming experience. (**Medium**)
- As a user/player, I want to be able to randomize the game's rewards for defeating NPC trainers so I can get a unique gaming experience. (**Medium**)

- As a user/player, I want to be able to randomize the game's items found on the map so I can get a unique gaming experience. (**Medium**)
    - Should be able to enable/disable key item randomization
- As a user/player, I want to be able to randomize Pokémon's TM compatibility so I can get a unique gaming experience. (**Medium**)
- As a user/player, I want to be able to randomize Pokémon's types, stats, palettes, abilities, wild hold items, and move sets so I can get a unique gaming experience. (**Medium - Large**)
- As a user I want to be able to enter and get a seed for my randomized game so I can

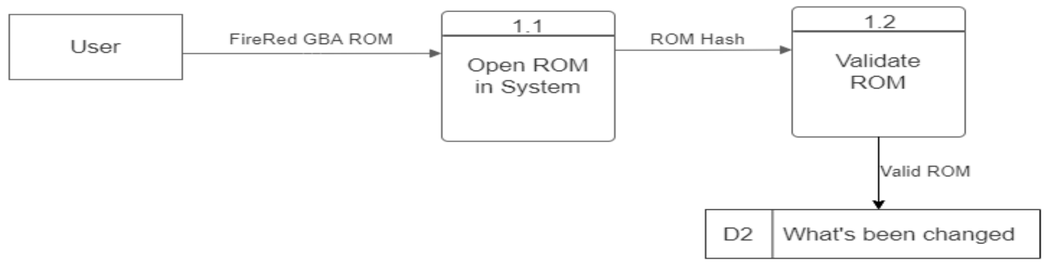
CONTEXT DIAGRAM



LEVEL 0



LEVEL 1



Data Dictionary and E/R Diagram

Our system won't be using any database structure since all of the data is stored within the ROM file itself and will not need to be stored by us or the system. In the next section however, we created a physical data model modeling how we believe the data looks internally.

Section III Design Narrative:

General Steps:

Create desktop applications to house modules.

Create randomization driver code (Executes correct randomization modules, and creates a copy of rom.)

Create different randomization modules and tie them into the desktop application.

We plan to release parts of this project one at a time, with the exception of creating the desktop application, driver code, and the first randomization module. These will most likely get done in the first part of the project. Once that is done, we can build one module at a time and implement it into the main project.

The first step we will take to build our project is building the desktop application and GUI that will house the different randomization modules. This will be a simple application. This will be built using Java Swing, which will allow us to build a flexible lightweight GUI. This application will allow users to select their game file, enter an optional randomization seed, and select their chosen randomization modules.

Along with building this main application, driver code will have to be developed. This will handle all other code being run. First, this code will create a copy of the game file, so that the original version of the game is protected. Then, depending on what modules the user chooses, the driver code will call other randomization modules and allow them to edit the game file. This code will also pass along a randomization seed if provided by the user. The driver code must also create a text file report of what seed was used, and what elements of the game are now randomized.

The first randomization module implemented will be to randomize the starter Pokémon that show up at the beginning of Pokémon FireRed. Usually, these Pokémon are always the same. This module will allow for choosing if the Pokémon must be basic Pokémon, or may also be more powerful evolutions. Once the setting is chosen, a random Pokémon will be chosen, from a list of Pokémon in the game. Once a Pokémon is chosen, the game's code will be edited so that the starter Pokémon's pointers now point to the randomly chosen Pokémon. For this module to work, we will have to create a file of all Pokémon, their pointer values, and what generation they are.

The second randomization module to be implemented will be randomizing the game's wild Pokémon encounters. This will require a list of where the encounterable Pokémon are stored, and randomly putting pointers to other Pokémon in those spots.

Once the first few modules are finished, we will continue to go down prioritized user story list one at a time. Once these first few most important modules are in place, adding more modules should be easier. We intend to add as many of these as we can, since there are many different features that we can eventually add into this program. The amount of these modules we will implement will depend on how long the first parts take.

Technologies to be Used:

Java: Java will be used for this project because it is well known among the team members. There are no special constraints for this project, so it is a simple choice to use Java. Java is also widely used in the game emulation community for creating different emulators

IntelliJ IDE: Intellij will be used as our main IDE, as it is the environment most of the members of the group know the best and feel most confident using. This IDE is an open source software development platform that everyone can install on their local machine and easily gain access to.

Java Swing: Java Swing will be used to make a simple and lightweight desktop application. The application GUI does not need to be complex, and will mostly contain buttons and checkboxes.

Data Storage: For this project we will use a simple file system. There is no need for a database because the data being stored is static, and not very complex. Most data being stored will be pointers for certain pieces of data, like where starter Pokémon are stored. Also being stored will be values within the domain for a certain part of the project. Since both of these pieces of stored data are simple and static, storing data in files should be fine.

Operating System: This project will be represented by an executable file that can be run on any operating system.

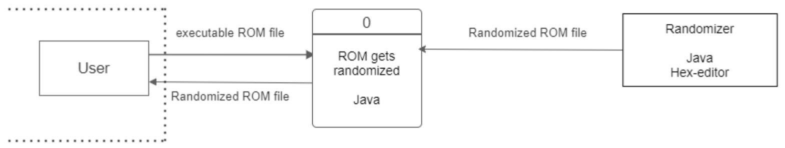
Discord: We will be using Discord to help communicate with each other when we can’t meet face-to-face. We will also use this as an informal method of keeping track of our work progress and communicating about assignments

Trello: Trello will also be used as a more formal platform to keep track of work progress and assign new work.

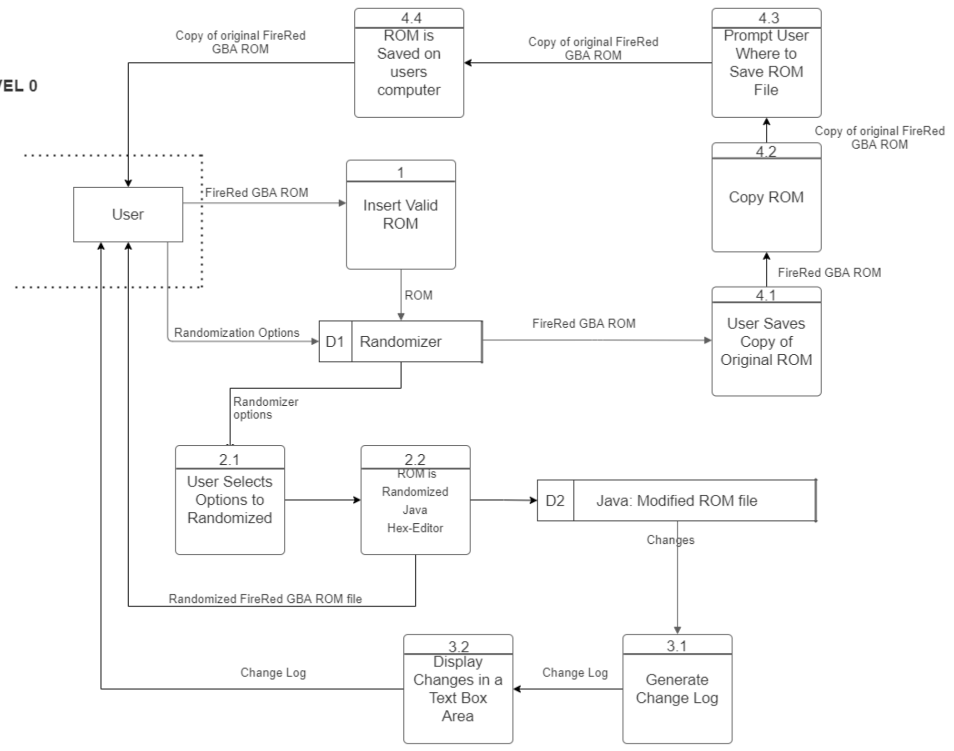
Miscellaneous Software: We will be programming this with a specific emulator in mind for using and testing the modified ROM file. The emulator we are testing this with will be mGBA and the ROM file will be a US version of Pokémon FireRed.

Physical Process Models

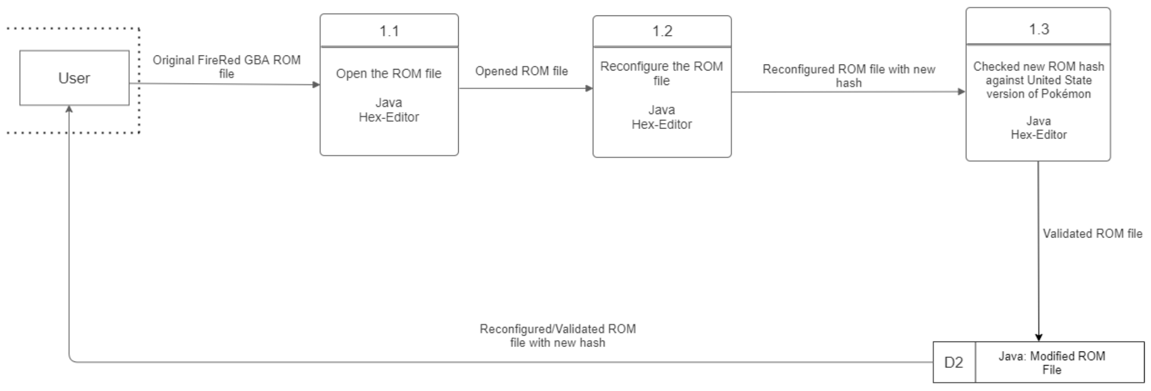
CONTEXT DIAGRAM



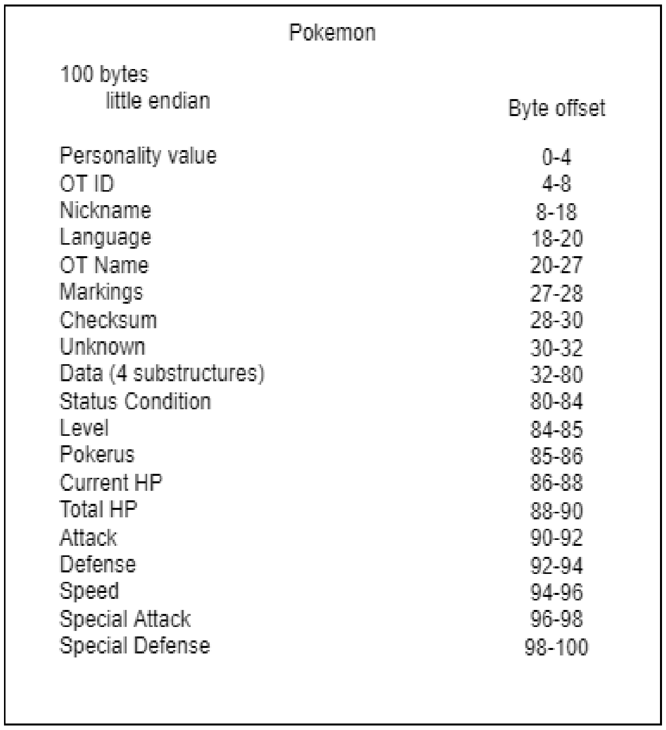
LEVEL 0


Copy of original FireRed GBA ROM
4.4
ROM is Saved on users computer
Copy of original FireRed GBA ROM
4.3
Prompt User Where to Save ROM File
Copy of original FireRed GBA ROM
4.2
Copy ROM
FireRed GBA ROM
4.1
User Saves Copy of Original ROM
User
FireRed GBA ROM
1
Insert Valid ROM
ROM
Randomization Options
D1
Randomizer
FireRed GBA ROM
Randomizer options
2.1
User Selects Options to Randomized
2.2
ROM is Randomized
Java
Hex-Editor
D2
Java: Modified ROM file
Changes
Randomized FireRed GBA ROM file
3.2
Display Changes in a Text Box Area
Change Log
Change Log
3.1
Generate Change Log

LEVEL 1


User
Original FireRed GBA ROM file
1.1
Open the ROM file
Java
Hex-Editor
Opened ROM file
1.2
Reconfigure the ROM file
Java
Hex-Editor
Reconfigured ROM file with new hash
1.3
Checked new ROM hash against United State version of Pokémon
Java
Hex-Editor
Validated ROM file
D2
Java: Modified ROM File
Reconfigured/Validated ROM file with new hash

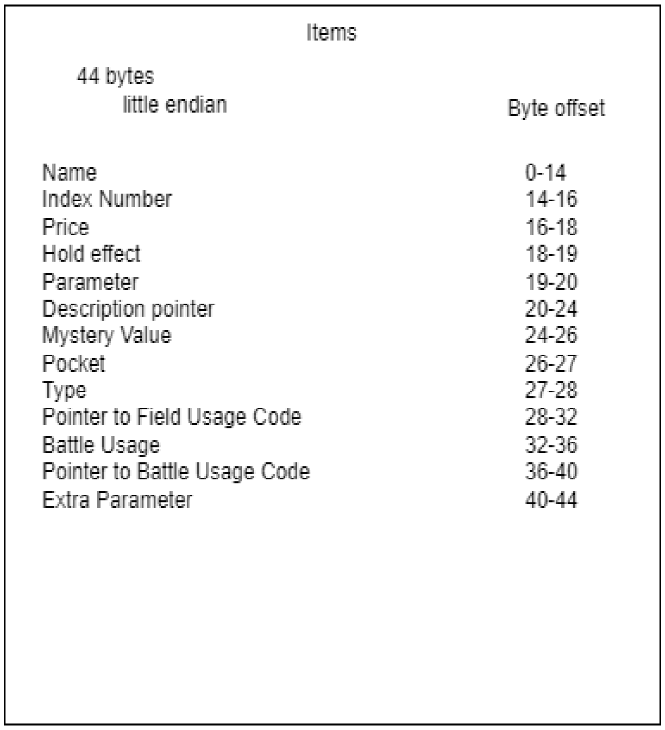
Physical Data Models



Pokemon

100 bytes
little endian

	Byte offset
Personality value	0-4
OT ID	4-8
Nickname	8-18
Language	18-20
OT Name	20-27
Markings	27-28
Checksum	28-30
Unknown	30-32
Data (4 substructures)	32-80
Status Condition	80-84
Level	84-85
Pokerus	85-86
Current HP	86-88
Total HP	88-90
Attack	90-92
Defense	92-94
Speed	94-96
Special Attack	96-98
Special Defense	98-100



Items

44 bytes
little endian

	Byte offset
Name	0-14
Index Number	14-16
Price	16-18
Hold effect	18-19
Parameter	19-20
Description pointer	20-24
Mystery Value	24-26
Pocket	26-27
Type	27-28
Pointer to Field Usage Code	28-32
Battle Usage	32-36
Pointer to Battle Usage Code	36-40
Extra Parameter	40-44

<u>Section IV Implementation:</u>

Source Control: https://github.com/Bigboss0912/Iron_Toddlers-BACS487

Work Assignments: Simon, our project manager, will be assigning work, using Trello.

- **Justin:** Will be working mostly on the coding and scripting for the back-end of the system
- **Michael:** Will also be focused on back-end coding and scripting
- **Owen:** Third of us focused on back-end development

- **Touch:** Will be focused on the front-end development, specifically fine tuning the UI and the referential documentation built in to the UI
- **Simon:** Will be focused on testing and creating the user guide as the project proceeds

Each member will also be assisting in other aspects, but the tasks listed above are the main focuses for each member.

Testing: We will be doing a mix of unit testing, system testing, and acceptance testing. The unit testing will allow us to see as we code if our modules are working correctly. Next, system testing will allow us to make sure that all of our modules work together as a whole. Finally, acceptance testing will be done to make sure the module works correctly within the entire system. We can check to see that the game is truly randomized. We will need to test to see that each module works correctly alone, and with other modules as well.

Documentation: The application will have referential documentation built into the UI, meaning that there will be explanations for each button and function within the application. That way, if a user isn't sure what a module does, they can quickly find out. We will also create simple procedural documentation. This will be a walkthrough, so a user can quickly start the program, and create a custom randomized version of the game.