OOP POKEMON PROJECT

This is the documentation for the Pokémon programming project, as part of the A-level computer science course (OCR). This is NOT my final submission and is being used as preparation and for feedback preceding the final programming project, which will have NO feedback from teachers, as per specification requirements

Adam Carrington; Candidate number: 1529; Centre number: 18339.

Table of Contents

[1.0 - INTRODUCTION 2](#_gjdgxs)

[1.1.0 – Pokémon, trading cards and video games. 2](#_30j0zll)

[1.1.1 – The video games. 3](#_1fob9te)

[2.0 – ANALYSIS 4](#_3znysh7)

[2.1.0 – Requirements 4](#_2et92p0)

[2.2.0 – Success criteria 4](#_3dy6vkm)

[2.3.0 – Stakeholders 4](#_1t3h5sf)

[2.4.0 – Research 4](#_4d34og8)

[3.0 – DESIGN 4](#_2s8eyo1)

[3.1.0 – Class Diagrams 4](#_17dp8vu)

[3.2.0 – Decomposition Diagrams 4](#_3rdcrjn)

[3.3.0 - Flowchart 4](#_26in1rg)

[4.0 – DEVELOPMENT 5](#_lnxbz9)

[5.0 – TESTING 5](#_35nkun2)

[5.1.0 – TEST TABLES 5](#_1ksv4uv)

[5.2.0 – Evidence 5](#_44sinio)

[6.0 – EVALUATION 5](#_2jxsxqh)

[7.0 – REFERENCES / BIBLIOGRAPHY 5](#_z337ya)

# 1.0 - INTRODUCTION

This introductory section into my project will attempt to outline, in as much detail and sophistication as possible, the Pokémon trading card game and video game(s), which will become the foundation for the project itself.

To reiterate what has been previously stated on the front page, this document and any code packaged with it is NOT for submission to exam boards.

## 1.1.0 – Pokémon, trading cards and video games.

This section of the introduction aims to outline and detail the premises and gameplay in the Pokémon trading card game, and video games (not including game-specific nuances).

### 1.1.1 – The video games.

The Pokémon games are a staple in gaming culture, and have been popular for decades, since its initial release in 1996, it has gone through many iterations and updates. Including spinoffs such as Pokémon Go and Pokémon Coliseum.

I have decided to focus on the original Pokémon games, entitled “Pokémon Red Version” and “Pokémon Blue Version,” as to not include any nuances and quirks from other games.

The Pokémon games in their most fundamental state are like a game of Top Trumps, in which both games compare statistics between each other to determine who wins. In Pokémon however, each player first chooses a Pokémon to use. Afterwards they all choose a move to use, and a target to act upon, comparing their “speed” stat to determine who plays first. Each move has its own properties and can act offensively or defensively depending on the type of move, and the outcome of using that move. An offensive move can provide both damage to an opposing Pokémon and defensive properties to the attacking Pokémon. After several turns occur, a player’s Pokémon may run out of HP, or “health points,” indicating the end of the game. The winning Pokémon is the Pokémon, which is still able to take a turn, the losing Pokémon being considered “fainted.”

## 1.1.2 – Object Oriented Programming

Object Oriented Programming, or OOP for short, is a style of programming which utilises ‘classes” to emulate a real-life object. Each class contains attributes and methods. Attributes are the same as variables in functional programming, however attributes are representations of features that the object has. In contrast, a method is an action that the object can carry out; for example, a person can have the attributes of ‘hair colour,’ and ‘eye colour,’ and the methods ‘move()’ and ‘talk().’ Take note of the usage of brackets after methods, these indicate that the method is a function of the object, and not an attribute. Attributes do not use brackets, because they simply store information for use in that class, or other classes through what is known as ‘inheritance.’

The syntax in OOP varies by language, however the theory behind it stays constant. Some languages do not have OOP functionalities, languages such as C, Fortran and COBOL.

# 2.0 – ANALYSIS

### 2.1.0 – Requirements

In this project, I will design, develop and produce a program which plays Pokémon in the style of the trading card game. It will be in text form and will contain the following features (subject to change):

* A main menu, which will contain:
  + Login menu
  + Settings management
    - Text speed, etc.
  + Quit game
  + Team management (See stats, etc.)
* The Pokémon game itself, which works the same as the trading card game

### 2.2.0 – Success criteria

|  |  |  |
| --- | --- | --- |
| Criteria | Importance | Completed |
| Main menu |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

### 2.3.0 – Stakeholders

There are likely to be few stakeholders in regards to this project, due to it being Pokémon, there will be a small group of people who may be interested for that aspect, however there may also be a larger group of people interested in the text-based aspects. If I were to publish this project on a website such as GitHub, there would be a very high chance that Nintendo either attempts to DMCA strike the repository, or I get a nice Cease and Desist letter from them asking me to remove it or go to court. As you could probably tell from that, Nintendo doesn’t like people recreating their games, especially under the licensed names. An example of this going wrong would be jdh, who wrote an operating system to only play Tetris([Tetris OS](https://www.youtube.com/watch?v=FaILnmUYS_U&t=708s)), while the game of Tetris is free to write and distribute, using the trademarked name will end badly for most people, because the Tetris Company will likely remove it from published webpages or, like Nintendo themselves, send you a Cease and Desist letter. Other than those setbacks, there is likely to be some fans of the series or text-based RPGs who would be interested. Some programmers may also be interested, most likely in the written code, rather than the running game.

### 2.4.0 – Research

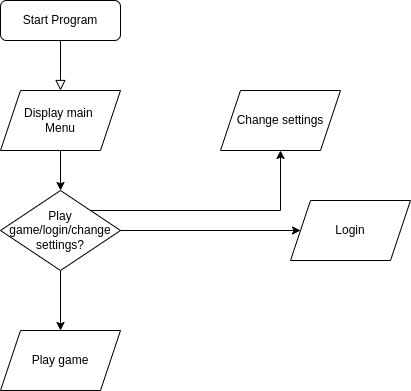
The research for this project was in most respects, simple. Most of the information that I have needed could be found in the Pokémon games, and through prior knowledge. However, that does not mean I did no research, in fact, I spent quite a lot of time watching modded playthroughs of the video games([I modded Pokémon platinum to RANDOMLY teleport me - PointCrow](https://www.youtube.com/watch?v=5Dmn-ni79AU), [Shining Pearl but every Pokémon evolves randomly each level - PointCrow](https://www.youtube.com/watch?v=5v7e4aLH3SA&t=2044s)). I too played a modded playthrough of Pokémon platinum, in which teleports are random. Although these are not original versions of the games, they still hold true to the original concept of Pokémon, due to the battling aspect of the games not changing.

# 3.0 – DESIGN

## 3.1.0 – Class Diagrams

## 3.2.0 – Decomposition Diagrams

## 3.3.0 – Flowchart



## 3.4.0 – UI Design

Being a text based application, the UI will mainly consist of ASCII text, this includes but is not limited to:

* characters a-z, lowercase and uppercase
* numbers 0-9 in any combination required
* special characters:
  + !”£$%^&\*()\_+-=/,.<>?;’:@[]{}#~¬`|\

The usage of these characters will be used to create an environment that a user can easily access without issues; however, an incompatibility with fonts can occur, due to some fonts not containing the required characters, however, this could be displayed to the user before a menu appears, to alert them to change fonts, so that the issue does not affect gameplay. Another solution is to utilise the minimum amount of special characters. Characters a-z and numbers 0-9 will be fine though, unless a different language is used, then an issue will not occur unless the language used does not use characters a-z or numbers 0-9. This will be an issue that is difficult to fix, unless translated versions are released.

## 3.5.0 – Algorithms

## 3.6.0 – Test plans

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test num | Criteria | Test Data | Type | Reason | Expected | Actual |
| 1 | Create Account |  |  |  |  |  |
| 2 | Login to account |  |  |  |  |  |
| 3 | Change login details |  |  |  |  |  |
| 4 | View pokemon |  |  |  |  |  |
| 5 | Change pokemon name |  |  |  |  |  |
| 6 | Play game |  |  |  |  |  |
| 7 | Save stats into file |  |  |  |  |  |

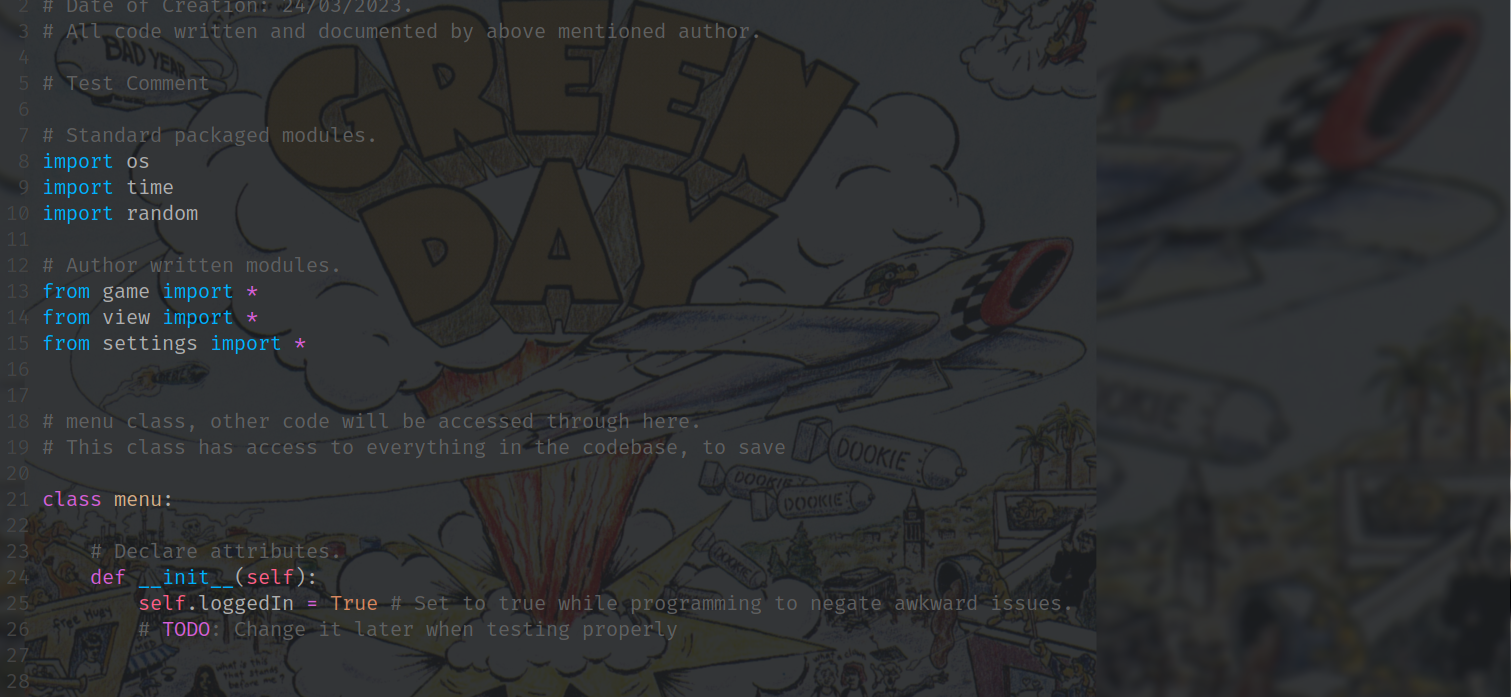
# 4.0 – DEVELOPMENT

This section outlines and, where necessary, develops on areas of development.

## 4.1 – Main menu, settings.

Writing the menu was very simple and not time consuming at all.

The code for the menu is shown below(You may have to zoom in):



The only attribute required is loggedIn, which will be set as false by default, it is currently set to true to avoid issues, because I have no login system written.

Creating a settings menu was more difficult, as I had to decide on the best way to save settings and edit them. As most other programs/games use, I decided to use a JSON file, as it uses a format that is easily understandable, and is essentially a dictionary, this means I can import the file as a dictionary without any weird formatting, edit the dictionary as nescessary, and write changes back to the json. By using a json file, you can also simply open the file and edit settings in a text editor, rather than having to run the game.

## 4.2 - Game

Writing the main game is difficult, so far(as of 17/04/2023) I have written code to generate a random team, and some stats, however, it is all not complete, I still need to write the code and find any issues, which won’t be extremely time consuming, however, finding the right syntax will take some time, because there will be at least 200+, if not more lines to write. The main issue is working with a large JSON file containing all moves, and another JSON file containing all pokemon and their stats, this is the bulk of the problem and will take some research and reading of Python’s documentation into how I can implement them effectively.

# 5.0 – TESTING

## 5.1.0 – TEST TABLES

## 5.2.0 – Evidence

# 6.0 – EVALUATION

# 7.0 – REFERENCES / BIBLIOGRAPHY