**Contents**

[I. Introduction 1](#_Toc151000652)

[II. Project Overview 1](#_Toc151000653)

[III. Reflection 2](#_Toc151000654)

[IV. Future improvements 3](#_Toc151000655)

[V. Conclusion 3](#_Toc151000656)

# I. Introduction

After a period of research and exploration into the "Online Coding Challenge Platform for Beginners" project, I have successfully developed a foundational programming platform tailored for newcomers. The source code is written in Python using Pygame library. Currently, the project ensures the essential features required for a standard 2D-RPG game. The project is developed based on image assets from the "Stardew Valley Game Clone with Python and Pygame" - <https://www.youtube.com/watch?v=R9apl6B_ZgI&t=1975s>, (freeCodeCamp.org) and some other free assets sourced from Pinterest.

The project runs smoothly; however, it still lacks many features to provide a good gaming experience (fighting features, creatures, menu). The graphics of some features are still simple.

# II. Project Overview

Spoj, w3school, codeforce are platforms that facilitate easier access for readers to assess their abilities or create learning motivation through tests, scores, or ranks. Is there a way to make everything even more engaging? A visual perspective from data visualization has inspired a new idea for building a coding challenge platform - an RPG game with a storyline, logic, integrated with supplementary animated elements for somewhat dry content like algorithms and data structures, with the expectation of adding more interest and motivation for users. Now, users' knowledge is evaluated in a livelier way: through a character's stat sheet.

The project is implemented in the form of a basic 2D RPG game, with the following features:

- Player: The character controlled by the user (using arrow keys).

Map: Constructed using Tiled software. From open assets, the software allows editing and exporting the Encoded Map as an XML file.

- Level: Due to time constraints in development, currently, there are only a few basic knowledge sections for the game levels (consisting of 12 multiple-choice questions in the areas of data structure, algorithm, and a quick test – questions requiring users to solve problems, attach files, input – provide accurate output). The map has a size of 30x30 tiles, with each tile consisting of 64 pixels.

- Quiz and Fight: Player's stats increase gradually after each quiz. These stats will assist in levels with increasing difficulty and vice versa – players need to take quizzes to replenish stamina points (a stat that allows players to use combat features). This will balance the educational and entertaining aspects of the game.

- Collision and Interaction: The project uses basic mathematical concepts and statements to handle "collisions" and interactions between two objects. This feature is clearly demonstrated in character movement and quiz solving.

- Visualization and Camera: Objects are layered to determine their drawing order (e.g., the ground layer will be prioritized for drawing first, the main layer – character is drawn on top). To optimize pixel rendering, layers are continuously generated around the character and shifted as the character moves.

- Story, Details, Instructions: The project currently lacks these specific and detailed features. However, the development and addition of these features are entirely feasible in the future.

# III. Reflection

Code, researching source code, practicing, testing, adding new features, testing, fixing bugs is an iterative process throughout the project. For me, the debugging process provides valuable experiences and insights into understanding how a program operates. Frameworks or libraries with high applicability are advantageous, but this can become a limitation if overused (not optimized for specific project requirements). The project uses a high-level language (Python) for ease of readability and accessibility, but resource control is challenging due to the flexibility of this programming language. A notable issue arises when trying to draw image surfaces with the Pygame library on a large-sized map, causing unstable fps. This is a clear concern and optimization area for future development of the project.

Throughout the project implementation, it can be observed that most of the architecture is built upon basic programming constructs: loops, conditions, and string manipulation. The most challenging aspect during this process is linking classes (object-oriented programming). The structure and programming style are crucial in this process, requiring a general mindset to efficiently reuse properties from the parent class. Creating logical dictionaries also greatly supports code clarity and conciseness.

Furthermore, path-related issues need thorough checking to ensure the project runs smoothly. In this project, I use the os library to retrieve the current direct path, combined with indirect paths to mitigate these risks.

Delta time (a parameter adjusting the fps – frames per second) combined with logical object arrangement contributes to creating more depth in gameplay. This factor is also used for setting the start and end times of animations, as well as the character's speed.

The Map and Combo box are two features that clearly demonstrate the effective use of data types. XML is utilized in creating the Map, showcasing its flexibility: easy to read, understand, use, and customize properties according to the specified needs. By creating XML from Tiled – Pixel points are encoded into a value corresponding to 64 pixel points (relative position). Drawing upon basic knowledge of vectors, issues related to collision or interaction are resolved.

The development process of the Combo box is truly intriguing when constructing it with a stack of 3 fixed slots. By saving the Combo configuration in the Settings file, the Combo box efficiently checks the order of key presses to use character skills corresponding to the status attributes (described more comprehensively in the video).

# IV. Future improvements

There are numerous directions to continue developing the project – both in terms of gameplay and test creation, synthesis, and knowledge statistics. Specifically, additional elements could be introduced such as creating more storyline, character backgrounds, increasing difficulty, diversifying game levels, enhancing interaction between characters and players (linking to health or fitness apps to boost corresponding character attributes). This will be the key to maintaining interest and learning motivation for players. The selection of knowledge to be conveyed needs to be clearer, categorized, and suggestions about the knowledge aspects players need to supplement should be provided if quiz test performance falls below a certain threshold.

# V. Conclusion

Game programming is a perfect experience for a first project due to its popularity and diversity. The flexible application of data structures, algorithms, and software architecture spans across all levels, depending on the programmer's knowledge and creativity. This is also a part of the reason why I evolved the initial online coding challenge platform into the current project (Online Coding RPG) – to put into practice a broader range of learned concepts.