Creating a game using “pygame” library.

Georgii Rukhlov
ČVUT–FIT
rukhlgeo@fit.cvut.cz

January 7, 2024

1 Introduction

My objective was to develop a game featuring real-time exploration and turn-based combat, utilizing the Pygame library.

2 Methods / algorithms

I employed the Pygame library to render the game on the screen, implementing a modular approach by dividing the game into separate modules, each dedicated to a specific function. One notable challenge I encountered was the creation of the game map. To address this, I discovered a solution in the form of the ’Tiled’ application. This tool enabled me to design a map by placing specific tiles and then export the design as ’.csv’ files.

In order to manage game entities effectively, I designed classes for tiles, enemies, and players. These classes were then grouped to facilitate collision detection and trigger battles, particularly when a collision occurred with an enemy. While the implementation of real-time exploration proved relatively straightforward, the development of a turn-based combat system presented challenges and required more time than anticipated. The resulting system, although functional, did not achieve the level of smoothness I aspired to.

Furthermore, I faced difficulties with the efficiency of sprite handling. Splitting sprite sets into individual images for each sprite proved to be a time-consuming process. This aspect of the project became an area for improvement, and I recognized the need to explore more efficient methods for managing sprite sets to enhance development efficiency.

3 Results

Real-Time Exploration:

The real-time exploration aspect of the game was successfully implemented and provided players with a seamless and engaging experience. Players were able to navigate through the game world in real-time, interacting with the environment and exploring different areas. Turn-Based Combat System:

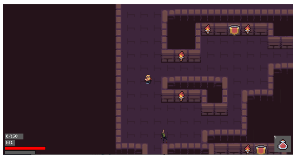

Figure 1: Real-time exploration

The development of the turn-based combat system posed challenges, and while functional, the system did not achieve the desired level of smoothness. Future iterations may benefit from optimizations and refinements to enhance the overall gaming experience during combat sequences.

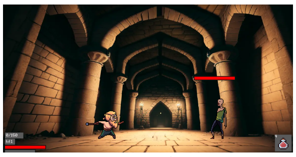

Figure 2: combat system

The integration of ’Tiled’ for map creation proved to be a practical solution, streamlining the process of designing diverse game environments. Drawing maps with tiles and importing them as .csv files not only enhanced efficiency but also contributed to the visual appeal of the game.

4 Conclusion

Working on this project has been a valuable learning experience, offering insights into game development from scratch and enhancing my understanding of gaming mechanics.

The code’s flexibility allows other users to customize the game by importing their own map ’.csv’ files and adding unique sprites. This project serves as a versatile tool for users to create their own games using the provided mechanics.

While the turn-based combat is functional, there is room for improvement. Enhancements, such as new features and better animations, could elevate the overall gaming experience.

References

[1] pygame community. pygame documentation. online, 2023. : https://www.pygame.org/docs.