Comparison with Design Specifications

Design Requirement	**Implementation**
<u>Concept/Gameplay</u> The user will navigate a player character through a 5x5 grid to reach an exit. The game will be turn based and the player can interact with other objects on the grid. On reaching the exit, the player enters the next level. Every 10 levels the background changes, with level 100 having a unique background. The user can upgrade their character's stats in a shop, during the run.	Concept remains the same. Other objects the player interacts with includes enemies and chests. Each turn, the player makes one move and each enemy makes one move. Background/world changes every 3 levels instead, as 10 levels is too long. Shop was implemented exactly as intended, however the user can now also purchase and upgrade character abilities.
<u>Components of Interface</u> Main Menu Game Screen Shop Screen Optional - Leaderboard - Collectible item screen	All original game screens were added. A leaderboard screen was added to supplement the endless gamemode. A guide was also added, to assist users in understanding the game and controls. Collectible items were not added at all, due to time constraints and their lack of importance to the main game.
<u>User Interaction</u> Game stats on the main menu. When the character is clicked, it is selected. When adjacent tiles are clicked or arrow keys are used, and movement is verified, the character is moved to the selected location. When other tiles are clicked, an information tab will appear. When pause or shop are clicked, their respective screens will appear.	No significant changes here. Character is moved through mouse clicks only however, as keyboard movement was too complex to implement in the time given. Information about other tiles appears on the side menu, when right clicked. Pause and shop menus both appear when their icons are clicked, however they can also be accessed through keyboard shortcuts, to improve ergonomics of the interface.

Basic stats will be shown on game screen. Sound output will be controlled by users own device.	Basic stats will be displayed for every tile on the grid. Sound output can be disabled from the pause menu, but volume is controlled by the user's own device.
<u>Code Structure</u> - Functions for main operations - Classes for similar objects - Loops for repetitions	All tiles on the grid share the same class, as they all have the same properties. Most of the code aside from the main grid loop is stored in functions and called when necessary
<u>Language</u> Python will be used for compatibility with all devices after initial setup. System will attempt to auto install libraries	Language has not changed. System attempts to auto install pygame
<u>Ergonomics</u> Menu screens (pause/shop) can be accessed with shortcuts. Tiles will be large enough to be easily clicked on by the user. User can use arrow keys or WASD to move player character	Shortcuts were implemented, (O to open shop and P or ESC to open pause). The game grid opens at a decent size on startup, and can be resized at any time. Arrow key and WASD movement was not implemented due to its complexity and time constraints
<u>Quality Assurance</u> To achieve quality assurance criteria, the following factors will be considered: - Reduce repetitions of code - Simple design/interface - Continuous testing of the code - Independent functions, whitespace and code comments	All quality assurance factors were considered and implemented when developing this system.

Summary

The project mostly meets the original design specifications, with only a few minor components of the original plan being left out, either to save time or improve on the original plan, such as the secondary arrow key input system and the reduced level count.

Testing Levels

Unit/Module

As most blocks of code were in their own respective functions, this level of testing was frequently used to ensure code worked as intended. As the program was being built, one function was created at a time, and tested individually to ensure that it worked as intended. Eg, to test the appearance of the shop menu, the shop function was called separately to observe the positioning of items on the shop menu.

Program

The entire program was run to test how well functions worked together, and whether each one returned the correct data or changed the required global variables. Eg. This type of testing was used to check the relationship of two functions in an earlier prototype. The first function determined the location of player movement, and the second animated movement. However, the animation moved differently to the intended location, revealing that there is an error with the values returned by the location determiner function.

System

The program was tested on the school computer and worked as intended. Parts of it were also tested on repl.it (as repl.it did not have python 3.10 when the code was tested, the entire program was not able to be tested), and the tested components also worked as intended.

Test Data

Testing Mouse Input

- Main Menu
    - When buttons are not clicked, program does nothing
    - When keys are pressed, nothing happens (except when ESC is pressed, game is closed)

        - When buttons are clicked, the user is taken to the correct screen
    - Game Screen
        - Side Menu
            - Shortcuts work as intended, and Shop and Pause menu work on click
            - When ability icon is clicked:
                - If ability has been triggered already, nothing happens
                - If there are no ability uses left, nothing happens
                - Otherwise, ability is activated
        - When right clicking on cards to test whether the information panel activates, an issue was discovered
            - Program currently detects all mouse clicks in all instances (including scroll wheel), not just left click
            - Adjusted program, so that only the intended button is detected
        - Testing gameplay
            - When an enemy is clicked, the player correctly attacks it
            - When an empty tile is clicked, the player moves to the empty tile
            - When tiles outside of the player's range are clicked, the input is ignored
            - If the player themselves is clicked a second time, the move is cancelled.
    - Pause Menu
        - When testing the options on pause menu, they were not being activated correctly
        - Used pygame.draw.rect() to find the dimensions of the menu options
        - Found a problem, where different dimensions were being used for the displayed option and detecting mouse click
    - Shop menu
        - When attempting to close shop menu, the menu had to be clicked multiple times before it would close.
        - After adding print() flags, the shop() function was found to be indented onto the wrong line, causing it to be part of another loop

<u>Testing Keyboard Input</u>

- Game Screen
    - When the shortcut keys are pressed, the correct menu opens.
- Shop Screen

- When a non number key is typed, it is ignored
- When numbers or backspace is typed, the bulk buy counter is updated respectively
- The 3 digit limit for the counter works as intended

Logging Tests

All major tests of the project were logged in the logbook, and issues were noted when they occurred. Solutions to these issues were also written, when the issues were solved. Minor tests and problems where only a few lines were added, or the problem took a very short time to fix were not logged.