**Maze Game- Documentation**

A logo with blue and red text

Description automatically generated

Name= Shovit Adhikari

Course= Cybersecurity and Digital Forensic (sec A)

Level4

Project work= 2D MAZE GAME

Submitted to: Aashish Acharya

* THE GAME OVERVIEW

This is the 2D graphical maze game made using the help of SFML and visual code editor. In this game the player navigates through a maze, collecting the collectable items, avoiding the enemy and tries to reach the exit point. The game features are as follows:

* A main menu with options for players to either start the game, continue to the previous one or quit or exit it.
* Maze with collectibles items and enemies.
* Game state saving/loading
* Level based progression
* HOW TO PLAY THE GAME
* To move the player around use the W, A, S, D button to navigate and move around the map.
* Make sure to avoid the enemy (X’) and collect the collectable items (\*’) and reach the exit door (E’).
* On the main menu press 1 to start the new game or 2 to load the previously saved file or press ESC button to quit the game
* The game automatically saves when you complete a level or when the game is over

**Maze Design**

* The maze is a 15x15 grid represented by a 2D character array.
* The symbol that are used here are

o 'P' - Player

o 'X' - Enemy

o '\*' - Collectibles (10 points each)

o 'E' - Exit Door (50 points)

o '█' - Wall

o ' ' - Empty space

**FILE INPUT/OUTPUT**

Saved game file: savegame.txt

It stores the

* Level number
* Score made
* Number of moves
* Coordinate of the player and enemy
* Full maze layout

**Required files**

* Arial.ttf (for the text and fonts)
* Savegame.txt (auto created when saving the game)

**Features of the game**

* Level-based design
* Basic enemy random movement
* Collison detection
* Save and load the progress
* Simple SFML UI and text rendering

**GAME FLOW**

* 1. Start menu
* Choose to start a new game or continue or quit

2. Game loop

\* Handle movement and interaction

\* Update enemy position

\* Redraw the window every frame

3. save/exit

\* Game save the progress on game over or level completion.

**Code Structure**

**1. Entity Class**

* Base class for all moving objects (Player, Enemy).
* Stores position (x, y) and symbol.
* placeOn() method puts the entity in the maze.

**2. Player Class**

* Inherits from Entity.
* Handles movement based on keyboard input.
* Updates score on collecting '\*', ends game on collision with 'X', completes level on reaching 'E'.

**3. Enemy Class**

* Inherits from Entity.
* Moves randomly in available directions.
* Kills the player if they occupy the same position.

**4. Game Class**

Handles the entire game logic:

* generateMaze() & generateMaze2() generate levels with predefined layouts.
* saveGame() stores level, score, moves, player/enemy positions, and maze layout in a file.
* loadGame() reads the saved data and resumes the previous session.
* draw() renders all elements (tiles, player, enemy, collectibles, exit) in a graphical window.

**5. Main Function**

* Displays main menu.
* Starts game loop.
* Processes player input.
* Updates enemy movement.
* Manages game over, level completion, and display updates.