

**Project Report**

**Data Structures Spring-2024**

**Section: CS-D**

**Members:**

Wajahat Ullah Khan 22I-0776

Momin Munir 22I-0854

Ahmad Aqeel 22I-1134

**Labyrinth Enchanted Explorer Game**

**Overview**

This game involves navigating through a labyrinth while collecting items and avoiding enemies. The player's objective is to reach the exit while managing their inventory, avoiding obstacles, and dealing with enemy encounters.

**Functionality**

1. **Start Game Function**: The **startGame()** function initializes the game environment, including the game window, maze layout, player, enemies, items, and scoring system.
2. **Inventory Display**: The game displays the items collected in the inventory, including keys, swords, shovels, maps, poison, and telephones.
3. **Game Loop**: The game operates within a continuous loop, checking for player input, updating game state, and rendering the scene accordingly.
4. **Collision Detection**: Collision detection mechanisms are implemented to handle interactions between the player, walls, enemies, and items. This ensures that the player cannot move through walls, collects items upon contact, and loses lives upon encountering enemies without a sword.
5. **Enemy Behavior**: Four enemies move within the maze, each with a predefined pattern of movement. If the player collides with an enemy without a sword, they lose a life.
6. **Win/Lose Conditions**: The game ends when the player either successfully reaches the exit with a shovel or loses all lives. Upon winning or losing, appropriate messages are displayed, and the game window closes upon pressing the space bar.

**Inventory System**

* The inventory system allows the player to collect and manage items found within the maze.
* Each item collected contributes to the player's score and may have specific effects within the game (e.g., keys unlock doors, swords defeat enemies).
* Pressing the "I" key opens a separate window displaying the current inventory items.

Work Distribution:

* Wajahat Ullah Khan (22i-0776):
  + Implemented the core game mechanics, including player movement, collision detection with walls and items, interaction with enemies, and win/lose conditions.
  + Managed the main game loop and window management using SFML.
  + Integrated the inventory system into the game, allowing players to interact with items and manage their inventory during gameplay.
* Momin Munir (22i-0854):
  + Implemented the maze generation algorithm and logic for creating random maze layouts.
  + Ensured that the generated mazes were suitable for gameplay, with clear paths, walls, and item placements.
  + Collaborated with other team members to integrate the maze generation system into the game and ensure that it interacted correctly with other game elements.
  + Implemented the AVL tree data structure for managing the inventory system, including insertion, removal, and searching operations.
  + Designed and implemented the AVL tree's integration with the inventory system to efficiently store and manage inventory items.
* Ahmad Aqeel (22i-1134):
  + Designed and implemented the inventory system architecture, including data structures and algorithms for managing inventory items.
  + Implemented the functionality for adding items to the inventory when players collect them in the game.
  + Implemented the logic for removing items from the inventory when they are used or consumed by the player.
  + Handled the user interface aspects of the inventory system, including opening the inventory window and displaying inventory items.