

## CL2001: Data Structures (Spring 2024)

### Semester Project

**Project groups:** This project can be done within a group of **three (3) students**. There is no restriction on the selection of group members. Students are allowed to make groups according to their preferences.

**Plagiarism:** -100% marks in the project if any significant part of the project is found plagiarized. A code is considered plagiarized if more than 20% code is not your own work. Ensure original work with proper attribution to avoid penalties.

#### Submission Guidelines:

1. All submissions **MUST** be made using the Google form. Solutions sent to the emails will not be graded. To avoid last-minute problems (unavailability of the Internet, load shedding, Internet down, etc.), you are strongly advised to start working on the project from day one.
2. Submit a report of your project that must include:
  - a. **Work division** between members
  - b. Brief explanation of your code.
3. Combine all your work (solution folder) in one .zip file after performing “Clean Solution”. Submit the zip file on Google Forms within the given deadline. If only .cpp file is submitted, it will not be considered for evaluation. **The naming convention for the zip file is rollno1\_rollno2\_rollno3.zip.**

**Note: Zip folder must contain source files(.cpp/.h) and project report only.**

**Deadline:** The deadline for project submission is **May 2nd, 2024 at 1:00 PM**. No late submissions will be accepted or graded.

## **Project Title: "Enchanted Labyrinth Explorer"**

**Scenario:** Welcome to the mystical realm of Arcadia, where an ancient labyrinth lies shrouded in mystery and danger. Known as the Enchanted Labyrinth, this intricate maze is said to conceal untold treasures guarded by magical creatures and hidden traps. Brave adventurers are called upon to embark on a quest to navigate through the labyrinth's twisting passages, employing wit, courage, and strategic thinking to overcome its challenges and claim its riches.

### **Tasks:**

#### **1. Implement AVL Tree for Inventory Management:**

The objective is to design an AVL tree data structure that can efficiently manage the inventory of treasures and artifacts procured by the adventurer in the Enchanted Labyrinth. The AVL tree must facilitate hassle-free insertion, deletion, and retrieval of items, while also ensuring a well-organized and balanced storage system for the adventurer's valuable loot.

#### **2. Design Maze Generation Algorithm:** Develop an algorithm that will dynamically generate the layout of the Enchanted Labyrinth. The maze should be generated procedurally and feature randomized paths, dead-ends, and obstacles to provide a one-of-a-kind and engaging experience for every adventurer. Incorporate the AVL tree to effectively store information regarding the locations and categories of the treasures concealed inside the maze.

#### **3. Implement the Shortest Pathfinding Algorithm:** To help adventurers navigate through the Enchanted Labyrinth, it is recommended to use a shortest pathfinding algorithm, such as Dijkstra's algorithm or A\* search. This algorithm will calculate the most efficient route from the entrance to the exit of the maze, taking into account the complex layout of the labyrinth and any obstacles encountered along the way.

#### **4. Integrate Enemy Encounters and Combat Mechanics:** Introduce mystical creatures and magical guardians inside the Enchanted Labyrinth to challenge adventurers during their quest. Implement a combat system that enables adventurers to participate in strategic battles with these foes, utilizing their collected treasures and abilities to overcome each encounter. Utilize the AVL tree to manage the adventurer's inventory of weapons, potions, and other combat-related items.

#### **5. Reward System Enhancement:** To make the Enchanted Labyrinth more exciting for adventurers, it is suggested to improve the reward system. This can be done by adding valuable treasures and artifacts that adventurers can obtain as they progress through the

maze. The addition of rare and legendary items that offer unique abilities and bonuses would motivate adventurers to explore deeper into the labyrinth in search of greater rewards. To keep track of the collected treasures and their attributes, the AVL tree can be used to store the information

6. User Interface Refinement: Design an immersive user interface for the Enchanted Labyrinth Provide adventurers with real-time feedback on their progress, inventory, and encounters within the maze, enhancing the overall gaming experience.
7. Testing and Optimization: It is important to thoroughly test the game to identify and fix any bugs, glitches, or performance issues. Additionally, the codebase and algorithms should be optimized for efficiency and scalability to ensure smooth gameplay and responsive controls on different devices and platforms.

The program should be written in C++ and should be well-documented and easy to understand

**Bonus:**

1. Add Sounds
2. Random Generation of Maze Each time

**Useful Links:**

Download SFML: <https://www.sfml-dev.org/download/sfml/2.6.1/>

SFML Setup: <https://youtu.be/pgvIJ-Zr9Ys?si=gH4AebmA5nQEfJdy>

SFML Shapes: <https://www.sfml-dev.org/tutorials/2.0/graphics-shape.php>

Good Luck 😊