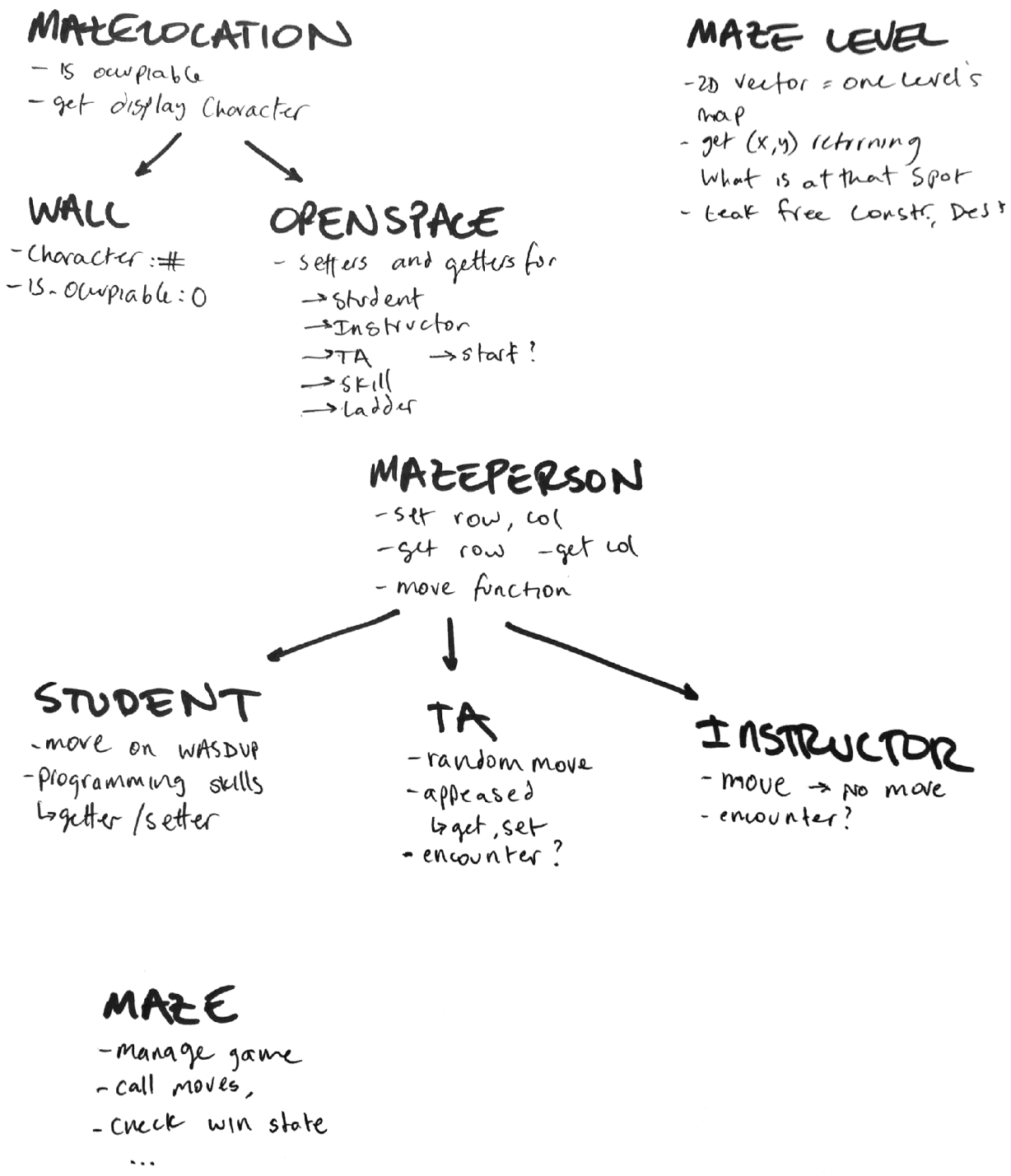
# Program Design & Testing Document for Final Program
# Lyell C Read

**Problem Statement**

This problem/assignment is asking me to construct a C++ program that:

- Does not segfault
- Compiles on FLIP
- Does not leak memory
- Allows the user to play ‘Escape from CS162’ - a maze game.
- Checks arguments and plays from a map in a text file.
- Uses the required classes, inheritance and polymorphism (below).
- This program:
    - Allows the user to choose where to move, and only permits this move if there is not an enemy there.
    - Manages the “special events” related to TAs and Instructors, as well as ladders, start.
    - Moves the TA’s randomly, and keeps track of their level of appeased-ness
    - Manages the player’s programming skills
    - Manages encounters with TA’s
    - Resets level or game on certain failures (see README.md)
    - Asks the user if they want to play again.
- I assume
    - The text file will be laid out per spec, as that will not be checked.
    - The user has knowledge of the game.

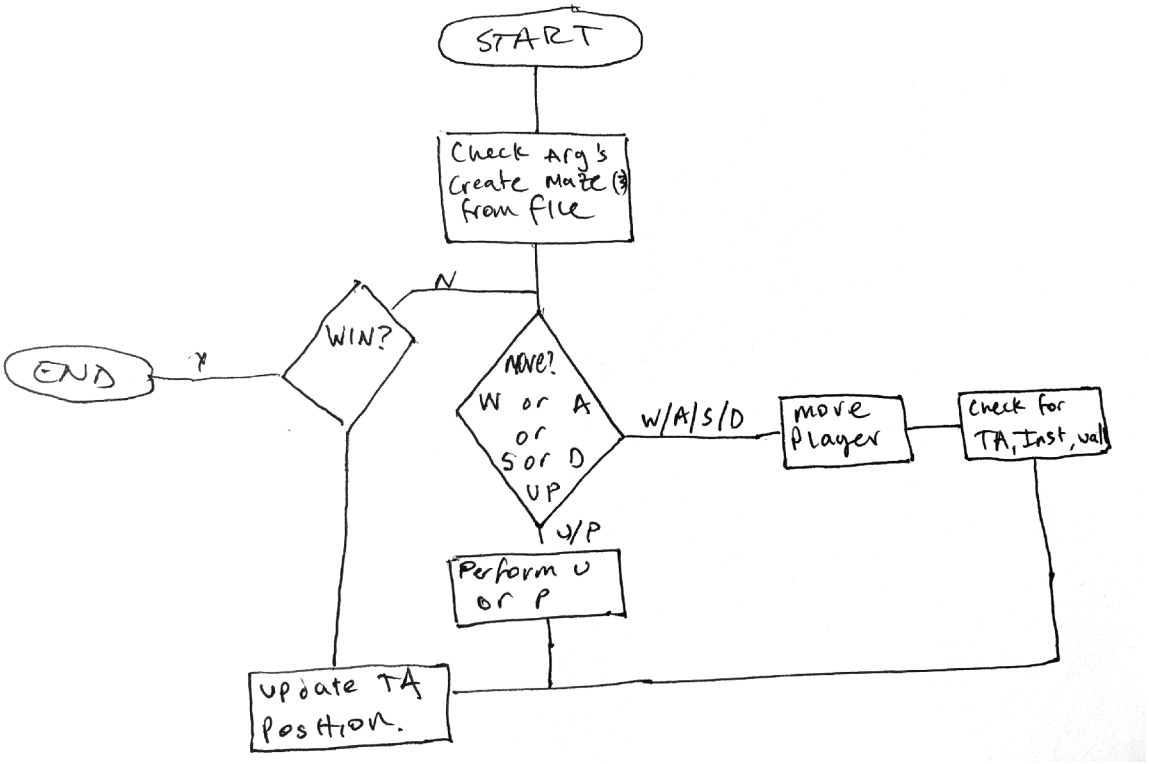
**Classes/ Inheritance Structure**




**Understanding the Problem**

As described in the requirements, the problem asks me to create a program that can play a nice game of ‘Escape from CS162’ using classes and inheritance and polymorphism. The game will be neatly organized, and follow the typical rules of the game listed in README.md, on GitHub. The class structure and member functions for this game are listed above.

**Program Flow (Diagram)**




**Pre- and Post- Conditions**

Format:

- Class::
    - Member Function Name
        - Pre
        - Post

- MazeLocation::
    - Is_occupiable
        - Member variable is_occupiable is defined
        - Returned.
    - Get_display_character
        - Member variable display_character is defined
        - Returned

- MazeLocation::Wall::
    - See MazeLocation::
- MazeLocation::OpenSpace::
    - Set_student
        - None
        - Variable set
    - Set_instructor
        - None
        - Variable set
    - Set_TA
        - None
        - Variable set
    - Set_cell_is_skill
        - None
        - Variable set
    - Set_cell_is_ladder
        - None
        - Variable set
    - get_student
        - Variable is defined
        - returned
    - get_instructor
        - Variable is defined
        - returned
    - get_ta
        - Variable is defined
        - returned
    - get_cell_is_skill
        - Variable is defined
        - returned
    - get_cell_is_ladder
        - Variable is defined
        - returned
- MazeLevel::
    - Constructor
        - File passed in is valid
        - Vectors for each new level stored dynamically
    - Destructor
        - Vector pointers are valid
        - Vectors are deleted, all memory for this function is freed.
    - Get_x_y
        - Vectors are defined
        - Returned

- MazePerson::
    - Set row
        - Row passed
        - Row set
    - Set col
        - Col passed
        - Col set
    - Move
        - None
        - None
- MazePerson::Student::
    - Move
        - None
        - Character code is returned.
    - Get_programming_skills
        - Programming skills var is defined
        - Returned
    - Set_programming_skills
        - Programming skills is passed, and var is defined
        - Set.
- MazePerson::TA::
    - Move
        - None
        - None
    - Get_appeasal
        - Appeasal timer is defined
        - Returned
    - Set_appeasal
        - Appeasal is defined and passed
        - Set.
- MazePerson::Instructor::
    - Move
        - None
        - None
- Maze::
    - Manage_game
        - Game started
        - Game ended

**Data Verification**

Checking user input for (“W” | ”A” | ”S” | ”D” | ”U” | ”P”)

| Input | What Should Happen | Does This Happen |
|---|---|---|
| “A” | Go Left | |
| “1” | Error - please enter again | |
| * SKE)@<#>C | Error - Please enter again | |
| “” | Nothing happens | |
| “ “ | Error - please enter again | |
| “K” | Error - Please enter again | |
| “WAS” | Go Up | |
| “ L : “ : :S” | Go Down (first characters ignored. | |

Checking for play again input (1 | 0)

| Input | What Should Happen | Does This Happen |
|---|---|---|
| 1 | Play Again | |
| 0 | Stop, and leave. | |
| A | Error - please enter again | |
| : | Error - please enter again | |
| 88 | Error - out of range. Please enter again | |
| []asdfpjaps0023ioru-9asudfj | Error - please enter again | |
| “” | Error - please enter again. | |