

|  |
| --- |
| SNAKES AND LADDERS |
|  |
| November 2  DS ASSIGNMENT 2  Authored by: NOMAN AZIZ (18I-1561) F |



# DATA STRUCTURES CREATED

|  |
| --- |
| * Array Based Queue Class * Players Struct * Node Struct * RollStack Class * Tile Struct * Board Class * Game Class |

IMPLEMENTATION

* I used *Array Based Queue Class* to store player turn sequence, it has various functions in it and important ones are
  + Void reverse() : In it I reverse the queue when needed in game (4 sixes)
  + Void removefront() : It is used to reduce queue size and remove the first element, it is used when a player wins the game, he will be removed from the turn queue.
* I used *Players Struct* to store the Players Position
* I used *Node Struct* to create a Dice Roll Stack
* I used *RollStack Class* to implement Dice Roll Operations since when a 6 comes dice must be rolled another time so it pushed into stack, its important functions are
  + Void push\_roll(int val) : It pushes a dice value in the stack
  + Void reverse() : It reverses the Stack, used in Board Class to Assign Tiles Snakes Positions since snake positions are stored in reverse in stack when read from map.txt
  + Void pop() : It pops the top value in stack
  + Void Empty() : It empties the stack and is used when next player turn is encountered.
* I used *Tile Struct* to create a 4D Linked List used in Board Class, it contains up, down, left and right tile pointers and two bool variables of snake and ladder to determine whether the tile contains snake head or ladder tail and also it contain an int index to store the index of tile.
* I used *Board Class* to create a 10X10 Game Board by linked list. It contain Start and End Tile pointers which points Tile 1 and 100 respectively, its main functions are
  + Void initialize() : It creates 100 Tiles using linked list, at this point it does not specify the ladders and snakes present in board.
  + Void setSnakesLadders(ifstream &input) : This function reads from a given file and implement snakes and ladders locations. At first, it reads snakes and ladders symbols from file and pushes their locations in two stacks (snakes and ladders) respectively. Then from previously created 100 Tiles Board, I used the stack values to assign snakes and ladders locations.
* Finally, I used *Game Class* to implement the main game, it contains variables of turnqueue, dicerolls, Ludo, players, noofplayers and currplayer from previously created classes. Its main functions are
  + Constructor: It calls Board Class Functions to initialize the board.
  + Int TraverseBoard() : This function receives a current position and diceroll stack, I pop values from stack and add them in current position one by one, if snake or ladder is encountered in this process, then current position changes its value to where that snake or ladder leads using Tile indexes.
  + Void RollDice() : This function is used for changing player turn queue, reversing player turnqueue and determining the entry of players in Board.
  + Void Play() : This function is used for all the simulations and display, and calling all the functions required in this game systematically.