Lecture 3

Linked Lists 2 (Application)

Dr. Yusuf H. Sahin
Istanbul Technical University

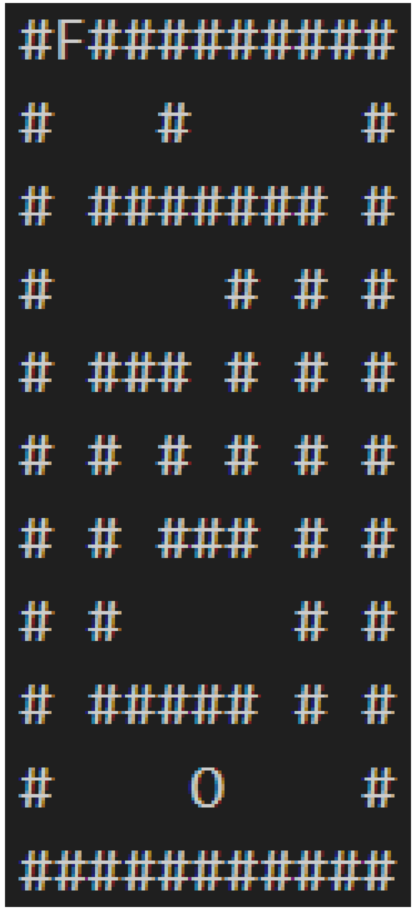
sahinyu@itu.edu.tr

Maze Excape

- In our maze escape game, the player (**O**) tries to reach the exit(**F**) by checking the empty tiles from four directions.

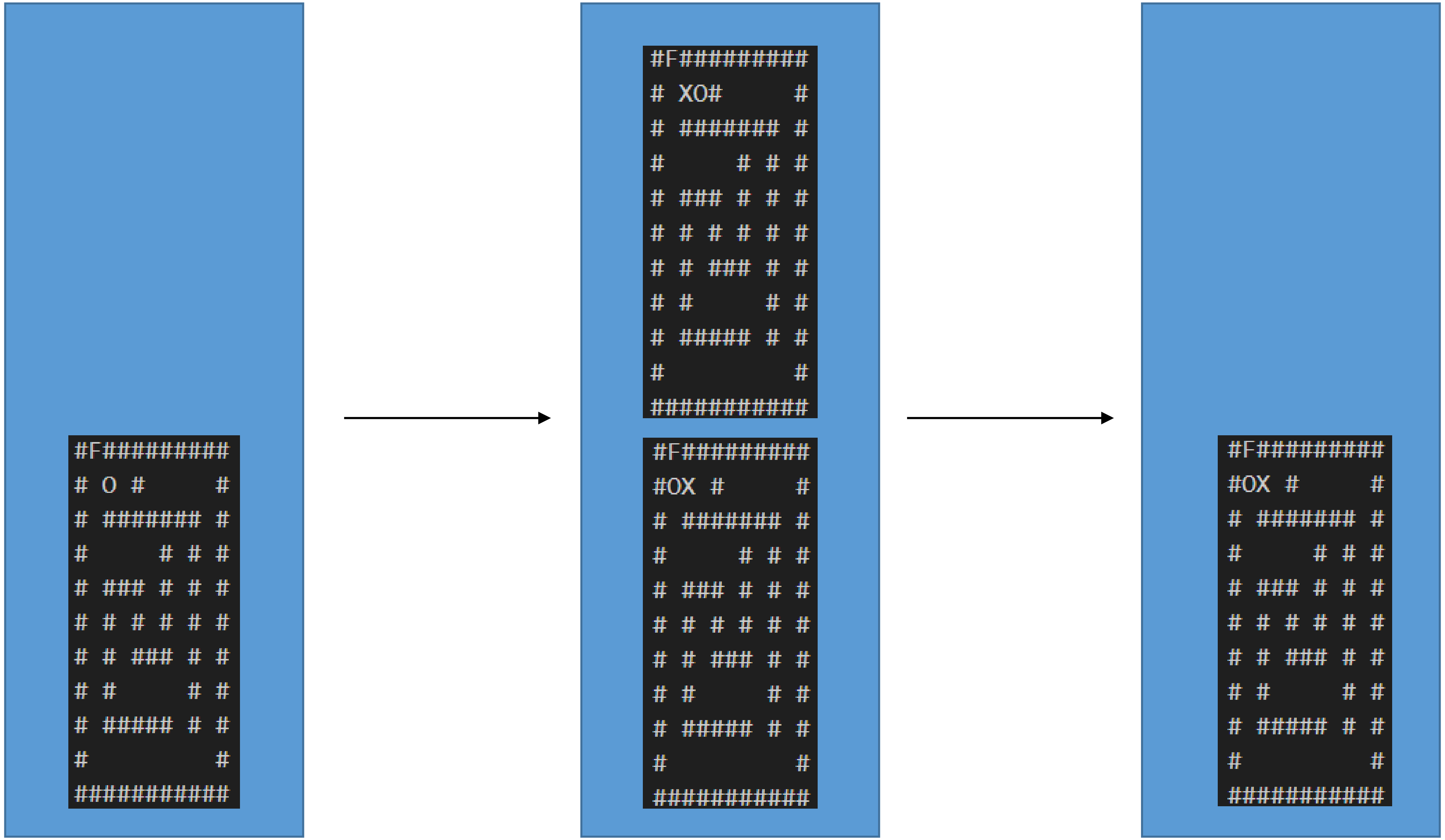
- There are so many trials and errors. To find the exact solution, stack usage is one of the most efficient options.

    - Push all the options (labyrinth states) to the stack.
    - If you have reached a dead end select the top state from the stack.



```
#F#########
#   #     #
# ####### #
#     # # #
# ### # # #
# # # # # #
# # ### # #
# #     # #
# ##### # #
#    O    #
###########
```

Maze Excape



```
#F#########
# O #     #
# ####### #
#     # # #
# ### # # #
# # # # # #
# # ### # #
# #     # #
# ##### # #
#         #
###########
```

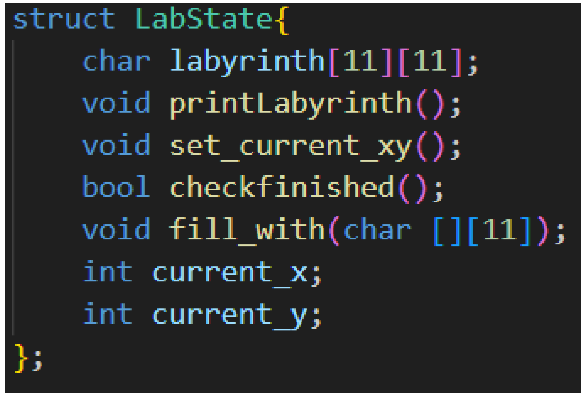
```
#F#########
# XO#     #
# ####### #
#     # # #
# ### # # #
# # # # # #
# # ### # #
# #     # #
# ##### # #
#         #
###########
```

```
#F#########
#OX #     #
# ####### #
#     # # #
# ### # # #
# # # # # #
# # ### # #
# #     # #
# ##### # #
#         #
###########
```

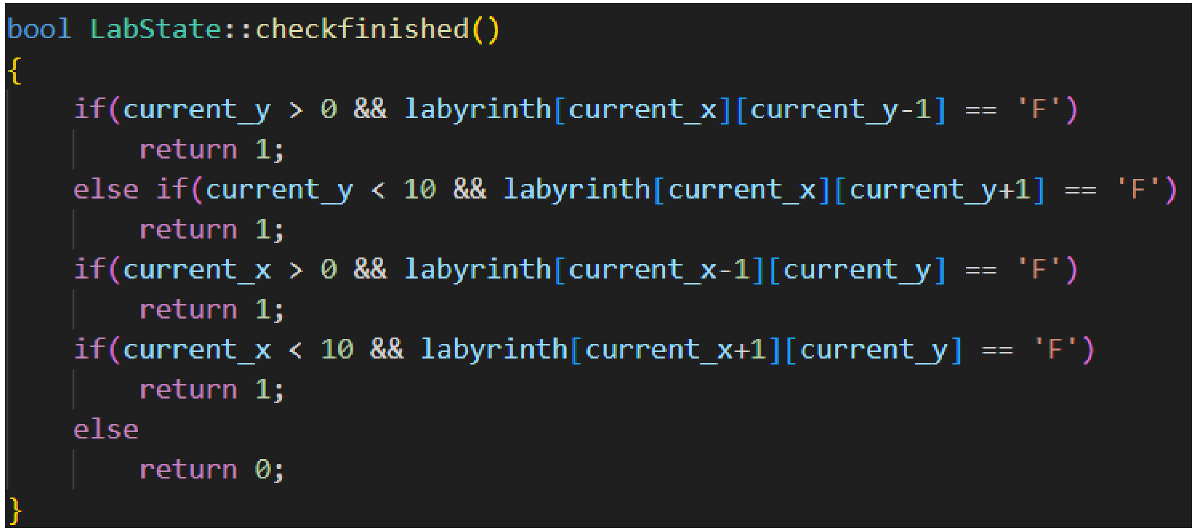
```
#F#########
#OX #     #
# ####### #
#     # # #
# ### # # #
# # # # # #
# # ### # #
# #     # #
# ##### # #
#         #
###########
```

Maze Excape

- In the skeleton code, a stack definition using linked lists, and the LabState data structure to store and manage the labyrinth state is given.



```
struct LabState{
    char labyrinth[11][11];
    void printLabyrinth();
    void set_current_xy();
    bool checkfinished();
    void fill_with(char [][11]);
    int current_x;
    int current_y;
};
```



```
bool LabState::checkfinished()
{
    if(current_y > 0 && labyrinth[current_x][current_y-1] == 'F')
        return 1;
    else if(current_y < 10 && labyrinth[current_x][current_y+1] == 'F')
        return 1;
    if(current_x > 0 && labyrinth[current_x-1][current_y] == 'F')
        return 1;
    if(current_x < 10 && labyrinth[current_x+1][current_y] == 'F')
        return 1;
    else
        return 0;
}
```

- **Task:** Escape from the maze using stack!