

CSE 102 Spring 2025 – Computer Programming Assignment 2

Due on March 19, 2025 at 23:59

Create a simple game in C where a player must reach a door on a 2D board. The game follows these constraints:

- **Only scalar variables** (no arrays, structs, or pointers)
- **Only conditional statements** (if, else, and else if)
- **File I/O** to store and update the game state
- **No loops** (iteration is done by multiple program executions)
- **No functions** (everything in main)

Game Rules:

1. The game operates on a **2D grid** of size width × height.
2. The game state is stored in game_state.txt and contains:
width height playerX playerY doorX doorY
3. If the file does **not exist**, a new board is generated.
4. The player moves using W (up), A (left), S (down), or D (right).
5. The code checks:
 - **If the input is valid** (only W, A, S, D allowed).
 - **If the player is hitting a wall** (out-of-bounds moves are rejected).
 - **If the player has reached the door** (game won).
6. If the player wins, a **congratulations message** is displayed, and a new board is generated.

Example Game Execution:

Initial run:

```
Generating a new board...
Enter the width and height of board:
5 5
Width: 5, Height: 5 Player: (2,2), Door: (4,4)
```

Second run:

Enter move (WASD): D
Player moves right to (3,2)
Game continues...

Run 3:

Enter move (WASD): S
Player moves down to (3,3)
Game continues...

Run 4:

Enter move (WASD): D
Player moves right to (4,3)
Game continues...

Run 5:

Enter move (WASD): S
Player moves down to (4,4)
Congratulations! You escaped!
Generating a new board...

Requirements:

1. **File Handling [30 pts]**
 - Reads from game_state.txt if it exists.
 - If the file doesn't exist, it creates a new game.
 - Updates the game state after every move.
2. **Scalar Variables Only**
 - No arrays, no loops, no functions.
3. **Game Persistence [70 pts]**
 - The game progresses with each execution.
 - The player moves closer to the door, and the game ends when they reach it.
 - After winning, a new board is generated.

IMPORTANT NOTES:

- Submit your homework as a zip file named as your student id (StudentID.zip) and this file should include:
 - YourStudentID.c file
 - A reports containing the screenshots of running code and generated outputs.
- Programs with compilation errors will get 0.
- Do not use any library other than `stdio.h`. You are only allowed `time.h` for generating random numbers.
- The output format must be as given, do not change it.
- Compile your work with given command "`gcc --ansi your_program.c -o your_program`".
- For any questions and problems use Teams page of the course.