# COSE322 System Programming Assignment #2

## Instructions

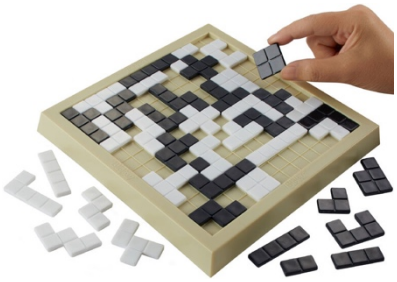
- **Deadline: November 17th, 2024, by 11:59 PM.**
- Carefully read the problem description and write the corresponding code.
- Create a Cargo package for the problem, with the following package name:
  - Problem_1
- Write a report explaining how you solved the problem. The report is limited to four A4 pages, and it should include the following contents.
  - **Feature Definition**: Break down the features required to meet the requirements as finely as possible (e.g., board creation and display, random block generation, block rotation, block placement, win/loss determination, etc.).
  - **Implementation Strategy**: Outline specific strategies to address the above requirements. No need to describe source code.
- Save the report as a PDF file, named in the following format:
  - Report_[Your student ID].pdf (e.g., **Report_2024210046.pdf**)
- Compress your Cargo package and report into a single .zip file and submit it on Blackboard. The .zip file should be named as follows:
  - Assignment2_[Your student ID].zip (e.g., Assignment2_ 2024210046.zip)
- In summary, your .zip file should include the following file tree structure

```
Assignment2_2024210046/
├── Report_2042210046.pdf
└── Problem_1/
```

- **Important:** If the submission format (file name, type, compression, etc.) is not followed, points will be deducted without exception
- **Assignment requirements:** Your program code must satisfy the following requirements.
  - **R1.** Your program code should be implemented with *src/main.rs* and *src/lib.rs*. The main logic of the program should be included in *lib.rs*, and main.rs should simply call the code defined in lib.rs.
  - **R2.** lib.rs should contain more than five test functions. It's up to you to decide which functions or methods to test.
  - **R3.** Lib.rs should utilize at least one closure and one iterator. It's up to you to decide where and how to use these features.

## Problem 1. Blokus

Implement a game where two players take turns filling blocks on the game board. The player who can no longer place a block loses the game.
Please refer to the board game in the example picture as a reference.



- The game board consists of a 9x9 grid.
- Each player receives a randomly generated block (P1: O, P2: @) on their turn.
- Each player can choose to place the block at a specified coordinate (input: 1 3) or rotate it counterclockwise (input: O).
- Blocks can only be placed in empty spaces on the board.
- If a player can no longer place their received block, that player loses, and the game ends.

```
   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|_|_|_|_|_|_|_|_|
2 |_|_|_|_|_|_|_|_|_|
3 |_|_|_|_|_|_|_|_|_|
4 |_|_|_|_|_|_|_|_|_|
5 |_|_|_|_|_|_|_|_|_|
6 |_|_|_|_|_|_|_|_|_|
7 |_|_|_|_|_|_|_|_|_|
8 |_|_|_|_|_|_|_|_|_|
9 |_|_|_|_|_|_|_|_|_|

P1's block:

 _ _ _
|_|O|_|
|O|O|O|
|_|_|_|

Put your block (r c) or Rotate (0): 1 1

   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|O|_|_|_|_|_|_|_|
2 |O|O|O|_|_|_|_|_|_|
3 |_|_|_|_|_|_|_|_|_|
4 |_|_|_|_|_|_|_|_|_|
5 |_|_|_|_|_|_|_|_|_|
6 |_|_|_|_|_|_|_|_|_|
7 |_|_|_|_|_|_|_|_|_|
8 |_|_|_|_|_|_|_|_|_|
9 |_|_|_|_|_|_|_|_|_|

P2's block:

 _ _ _
|_|_|@|
|@|@|@|
|@|_|_|

Put your block (r c) or Rotate (0): 0

P2's block:
```

```
 _ _ _
|@|@|_|
|_|@|_|
|_|@|@|

Put your block (r c) or Rotate (0): 1 3

   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|O|@|@|_|_|_|_|_|
2 |O|O|O|@|_|_|_|_|_|
3 |_|_|_|@|@|_|_|_|_|
4 |_|_|_|_|_|_|_|_|_|
5 |_|_|_|_|_|_|_|_|_|
6 |_|_|_|_|_|_|_|_|_|
7 |_|_|_|_|_|_|_|_|_|
8 |_|_|_|_|_|_|_|_|_|
9 |_|_|_|_|_|_|_|_|_|

.....
```

- Configure the game board as described above, allowing each player to place their assigned block.
- Randomly generate one of the 14 block types illustrated below for each turn. Ensure that various shapes of blocks are generated within a 3x3 grid range.

```
 _ _ _
|O|O|O|
|_|_|_|
|_|_|_|
 _ _ _
|O|O|O|
|O|_|_|
|_|_|_|
 _ _ _
|O|O|O|
|_|_|O|
|_|_|_|
 _ _ _
|O|O|O|
|_|O|_|
|_|_|_|
 _ _ _
|_|O|O|
|O|O|_|
|_|_|_|
 _ _ _
|O|O|_|
|_|O|O|
|_|_|_|
 _ _ _
|O|O|_|
|O|O|_|
|_|_|_|
 _ _ _
|O|_|_|
|O|O|_|
|O|O|_|
```

```
 _ _ _
|_|O|O|
|O|O|_|
|O|_|_|

 _ _ _
|O|O|_|
|_|O|O|
|_|O|_|

 _ _ _
|_|O|O|
|O|O|_|
|_|O|_|

 _ _ _
|_|_|O|
|O|O|O|
|O|_|_|

 _ _ _
|O|_|_|
|O|O|O|
|_|_|O|

 _ _ _
|_|O|_|
|O|O|O|
|_|O|_|
```

- **Block Placement**: When the user inputs a row and column coordinate, place the generated block at that position. In this case, the top-left corner of the generated block serves as the placement reference point. Refer to the example below.

```
P1's block:

 _ _ _
|X|O|_|
|O|O|O|
|_|O|_|

Put your block (r c) or Rotate (0): 2 3

   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|_|_|_|_|_|_|_|_|
2 |_|_|X|O|_|_|_|_|_|
3 |_|_|O|O|O|_|_|_|_|
4 |_|_|_|O|_|_|_|_|_|
5 |_|_|_|_|_|_|_|_|_|
6 |_|_|_|_|_|_|_|_|_|
7 |_|_|_|_|_|_|_|_|_|
8 |_|_|_|_|_|_|_|_|_|
9 |_|_|_|_|_|_|_|_|_|

The X marks are shown for reference purposes. The X mark indicates the
starting point for inputting the block placement coordinates.
```

- **Block Generation/Placement**: Display P1's blocks as O and P2's blocks as @ (refer to the game description above). When placing blocks, ensure that new blocks cannot be placed in spaces that are already occupied (marked as O or @). If a player selects a space where the block cannot be placed, display an error message, allowing the player to choose a different location.

```
   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|_|_|_|_|_|_|_|_|
2 |_|_|_|O|_|_|_|_|_|
3 |_|_|O|O|O|_|_|_|_|
4 |_|_|_|O|_|_|_|_|_|
5 |_|_|_|_|_|_|_|_|_|
6 |_|_|_|_|_|_|_|_|_|
7 |_|_|_|_|_|_|_|_|_|
8 |_|_|_|_|_|_|_|_|_|
9 |_|_|_|_|_|_|_|_|_|

P2's block:

 _ _ _
|@|@|@|
|_|_|_|
|_|_|_|

Put your block (r c) or Rotate (0): 4 3
P2 is not able to put the block into (4,3).

P2's block:

 _ _ _
|@|@|@|
|_|_|_|
|_|_|_|

Put your block (r c) or Rotate (0):
```

- **Block Rotation**: If the player inputs 0, rotate the block 90 degrees counterclockwise around the center of the 3x3 grid. Refer to the example below. The rotation can be performed multiple times.

```
P2's block:

 _ _ _
|_|_|@|
|@|@|@|
|@|_|_|

Put your block (r c) or Rotate (0): 0

P2's block:

 _ _ _
|@|@|_|
|_|@|_|
|_|@|@|

Put your block (r c) or Rotate (0):
```

```
P1's block:

 _ _ _
|0|0|0|
|_|_|_|
|_|_|_|

Put your block (r c) or Rotate (0): 0

P1's block:

 _ _ _
|0|_|_|
|0|_|_|
|0|_|_|

Put your block (r c) or Rotate (0):
```

- **Win / Loss Determination**
  - If a player receives a block but cannot place it on the board, they lose the game.
  - After considering all possible placements, including rotations, determine if the block can be placed.
  - If it is confirmed that the block can no longer be placed, display a win/loss message and end the game.

```
   1 2 3 4 5 6 7 8 9
   _ _ _ _ _ _ _ _ _
1 |_|0|_|_|@|_|0|@|@|
2 |0|0|0|_|@|@|0|@|@|
3 |_|0|_|@|@|0|0|0|_|
4 |0|0|0|0|0|0|_|0|0|
5 |0|@|_|0|@|_|_|0|_|
6 |@|@|@|@|@|@|0|_|_|
7 |0|@|0|0|@|0|0|0|@|
8 |0|0|0|0|@|@|@|@|@|
9 |@|@|@|0|0|_|@|@|@|

P1's block:

 _ _ _
|_|0|_|
|0|0|0|
|_|0|_|

P1 fails to put the block. P2 wins!
```