

# SOS Game with Parallel Programming

## Introduction

In this project, you will implement the SOS game and use parallel programming to reduce the time it takes to calculate the winner on larger grids. The SOS game is a two-player game played on a grid, where players take turns to place either an 'S' or an 'O' on the grid. The goal is to create a sequence of 'SOS' either horizontally, vertically, or diagonally. The player with the most 'SOS' sequences at the end of the game wins.

## Implementation

You will implement the SOS game using Python. You will also use the multiprocessing library to parallelize the calculation of the winner on larger grids. Further you will also implement a computer opponent.

## Game Rules

- The game is played on a square grid (size = 8).
- Players take turns to place either an 'S' or an 'O' on an empty cell of the grid.
- The first player to create a sequence of 'SOS' either horizontally, vertically, or diagonally wins.
- If the grid is filled and no player has won, the game is a draw.

	S						
					O		
			S				
		O			S		
	S						
	O						
	S	S			S	O	S

## GUI (BONUS) 3 ABS

The GUI will have the following elements:

- A grid to display the game state.
- Two buttons to place 'S' or 'O'.
- A label to display the current player.
- A label to display the winner or draw message.
- A button to reset the game.

You can use the PySimpleGUI or tkinter library to create the GUI.

## Parallel Programming

You will parallelize the calculation of the winner using the multiprocessing library. You will divide the grid into equal parts and assign each part to a separate process to calculate the number of 'SOS' sequences in that part. Once all the processes have finished, you will combine the results to determine the winner.

## Deliverables

The following are the deliverables for this project:

1. Python code for the SOS game with GUI and parallel programming.
2. A report documenting the design and implementation of the project.
  1. Mention parallel parts of the project
  2. Screenshots
3. A video to demonstrate the game and the parallel programming.

## Grading

The project will be graded based on the following criteria:

1. Correctness of the game rules and GUI.
2. Correctness of the parallel programming.
3. Efficiency of the parallel programming in reducing the time for larger grids.
4. Clarity and organization of the report.
5. Quality and clarity of the presentation.