

პროგრამირების აბსტრაქციები

სემინარის ამოცანები #9

თამაში Tic-tac-toe

ჩვენი მიზანია თამაში "Tic-tac-toe"-ს ამომხსნელი პროგრამა დავწეროთ. ანუ გვჭირდება ისეთი ფუნქცია, რომელსაც დაფაზე არსებულ სიტუაციას გადავცემთ და ეს პროგრამა გვიპასუხებს ორივე მოთამაშემ რომ ოპტიმალურად ითამაშოს რა შედეგით დამთავრდება თამაში.

ა) ამისთვის ჯერ შემოვიღოთ დაფის აღმწერი სტრუქტურა.

```
#include "grid.h"
using namespace std;

enum position {
    cross = 1,
    neither = 0,
    nought = -1
};

struct boardT {
    Grid<position> grid;
    position currentPlayer;
};
```

ბ) ახლა დავწეროთ ფუნქცია, რომელიც შეამოწმებს მოცემულ დაფაზე თამაში დამთავრებულია თუ არა და თუ დამთავრებულია ცალკე მისამართით გადაცემულ არგუმენტში ჩაწერს ვის სასარგებლოდ დამთავრდა თამაში.

```
bool doesPlayerWin(boardT& board, position player) {
    for (int i = 0; i < 3; i++) {
        bool isRowFull = true;
        bool isColumnFull = true;
        for (int j = 0; j < 3; j++) {
            if (board.grid[i][j] != player)
                isRowFull = false;
            if (board.grid[j][i] != player)
                isColumnFull = false;
        }
        if (isRowFull || isColumnFull)
            return true;
    }

    if (board.grid[0][0] == player
        && board.grid[1][1] == player
        && board.grid[2][2] == player)
        return true;
    if (board.grid[0][2] == player
        && board.grid[1][1] == player
        && board.grid[2][0] == player)
        return true;

    return false;
}

bool isGameOver(boardT& board, position& gameResult) {
    if (doesPlayerWin(board, cross)) {
        gameResult = cross;
        return true;
    }

    if (doesPlayerWin(board, nought)) {
        gameResult = nought;
        return true;
    }

    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (board.grid[i][j] == neither)
                return false;
        }
    }
    gameResult = neither;
    return true;
}
```

გ) ახლა კი დავწეროთ ფუნქცია რომელიც შეამოწმებს მოცემული დაფიდან დაწყებული ორივე მოთამაშემ რომ სწორად ითამაშოს როგორ დამთავრდება თამაში.

```
position gameFinalResult(boardT& board) {
    position gameResult;
    if (isGameOver(board, gameResult))
        return gameResult;

    position otherPlayer;
    if (board.currentPlayer == cross)
        otherPlayer = nought;
    else
        otherPlayer = cross;

    gameResult = otherPlayer;
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            if (board.grid[i][j] == neither) {
                boardT nextBoard = board;
                nextBoard.grid[i][j] = board.currentPlayer;
                nextBoard.currentPlayer = otherPlayer;
                position currentResult = gameFinalResult(nextBoard);
                if (board.currentPlayer == cross) {
                    if (currentResult > gameResult)
                        gameResult = currentResult;
                } else {
                    if (currentResult < gameResult)
                        gameResult = currentResult;
                }
            }
        }
    }
    return gameResult;
}
```