Context

Embarking on your journey as a new game studio, you're immersing yourselves in the software lifecycle by initiating your first project: de developing, testing, and iterating upon a digital rendition of the timeless game, Tic-Tac-Toe. This endeavor not only showcases your team prowess but also underscores your commitment to mastering the intricacies of the software development process.

User Stories

- As a player, I want to be able to make a move by choosing an empty square, so that I can place my symbol on the board.
- As a player, I want to be able to see the current state of the game, so that I can keep track of the moves made by both myself and opponent.
- As a player, I want to be notified when the game has ended in a win, loss or draw, so that I can see the result of the game.
- As a player, I want to be able to start a new game after the current game has ended, so that I can play again.
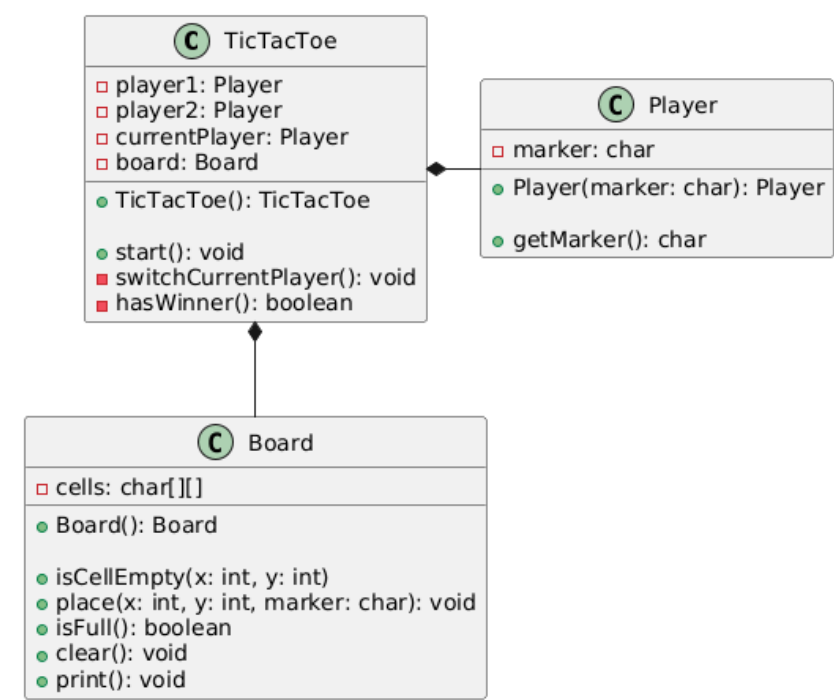
Tasks

- Create a GitHub repository (public) ✓
- Create a GitHub project and link it to the created repository ✓
- Transfer the user stories to a new Kanban board in the project ✓
- Give every user story a unique id ✓
- Create a new Java Maven project ✓
- Initialize a new git repository ✓
- Set the GitHub repository as a remote repository ✓
- Commit and push your blank Java project to the main branch ✓
- Protect the main branch (only allow pull requests) ✓
- Start working on the user stories ✓
- Create a new feature branch for every user story and name it correctly
- Merge the branch after completion via pull request
- Make sure the CI/CD tests your code before merging
- Always advance the user stories on the Kanban board
- The main branch should "deploy" the application via CI/CD (make the jar downloadable)

↑

?

Class Diagram

C TicTacToe

- player1: Player
- player2: Player
- currentPlayer: Player
- board: Board

- TicTacToe(): TicTacToe

- start(): void
- switchCurrentPlayer(): void
- hasWinner(): boolean

C Player

- marker: char

- Player(marker: char): Player

- getMarker(): char

C Board

- cells: char[][]

- Board(): Board

- isCellEmpty(x: int, y: int)
- place(x: int, y: int, marker: char): void
- isFull(): boolean
- clear(): void
- print(): void

Example Output

```
Current Player: X
 ───────
| | | |
| | | |
| | | |
 ───────
row (0-2): 1 (human input)
column (0-2): 1 (human input)
Current Player: O
 ───────
| | | |
| |X| |
| | | |
 ───────
rom (0-2): 1 (human input)
column (0-2): 0 (human input)
Current Player: X
 ───────
| | | |
|O|X| |
| | | |
 ───────
```

Last modified: Wednesday, 9 August 2023, 3:44 PM

↑

?