Design report

The goal is to develop the most epic game of Tic-tac-toe you have ever played.

The team decided to make web based project and made a class diagram using brainstorming. The class diagram (see following) was used as a guideline for the development.

|  |
| --- |
| Player |
|  |
| name()  wins()  losses()  human()  computer() |

|  |
| --- |
| Board |
|  |
| moves()  status()  empty() //Til að taka stöðuna |

|  |
| --- |
| GameEngine |
| + Board board  + Player player |
| moveCounter()  wins()  draw() // teiknar í GUI |

|  |
| --- |
| TicTacToe |
| + GameEngine game |
|  |

The final class diagram looks like the following tables:

|  |
| --- |
| Board |
| - board: char[] |
| + void randomPlace(char value)  + void place(int cellIndex, char value)  + boolean isFull()  + int emptyCount()  - emtpyCells()  + string toString() |

|  |
| --- |
| Cell |
| + x: int  + y: int  + symbol: Character |
| + Cell(int x, int y, Character symbol)  + boolean onBoard(int y) |

|  |
| --- |
| Game |
| # Board board |
| + Game()  + void computerMove()  + void playerMove(int cellIndex)  + void customMove(int cellIndex, char value)  + string winner()  + boolean isDone() |

|  |
| --- |
| Player |
| - name: string  - wins: int  - losses: int |
| + player() |

|  |
| --- |
| WebGame |
|  |
| + webGame()  + String render() |