

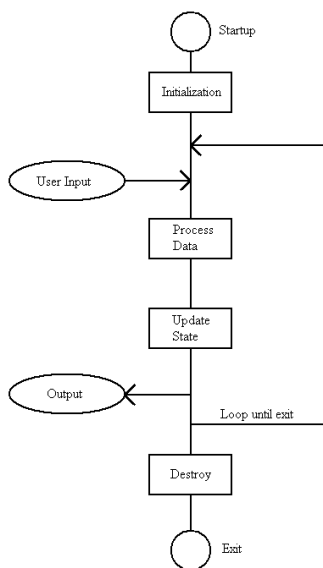
Läxa

I den här uppgiften ska vi bygga klart tictactoe (luffarschack) för att ha ett färdigt spel redo att spela vars lösning vi går igenom på fredag, samt lite annat också. Vi ska nu använda (switchPlayer) och (getUserInput) från förra läxan. (drawBoard) från förra läxan behöver vi dock göra om. Vi vill nu skapa en gameloop också som tar input från användare, uppdaterar spelvärlden och renderar (ritar ut) spelvärlden.

Gameloop

I ett spel använder man alltid en gameloop som itererar över 4 huvudsakliga uppgifter

- * ta användarinput (getUserInput från förra läxan)
- * uppdatera spelvärldens konfiguration (updateBoard)
- * kontrollera om någon har vunnit (gameOver)
- * outputa (rendera/rita upp) den nya spelvärlden, (drawBoard)



Instruktioner

1. Börja med att skapa 2 globala variabler

* **let player = "X";**

* **const board = [[null, null, null], [null, null, null], [null, null, null]]**

2. Förutom (getUserInput) och (switchPlayer) från förra läxan ska vi nu skapa ytterligare 3 funktioner:

* **function updateBoard(board, position, player) { }**

Denna funktion ska uppdatera board på positionen med player värdet så att t.ex

updateBoard(board, 8, "X") ;

console.log(board)

=>

| | | |
|--|---|--|
| | | |
| | | |
| | X | |

```
updateBoard(board, 3, "O");
```

```
console.log(board);
```

=>

| | | |
|--|---|---|
| | | O |
| | | |
| | X | |

För att updatara board behöver ni plocka fram rätt rad och rätt kolumn i den raden. Efter en board är en array av arrayer behöver ni ändra board på följande sätt: `board[0][2] = "O"` för att placera O i det översta högra hörnet.

*** function gameOver(board) { }**

Denna funktion ska kontrollera om någon har vunnit spelet. Funktionen tar board variabeln som input och kontrollerar om någon rad har samma tecken ifylld så att t.ex

```
console.log(board);
```

=>

| | | |
|---|---|---|
| | | O |
| | O | |
| O | | |

```
console.log(gameOver(board));
```

=> true

```
console.log(board);
```

=>

| | | |
|---|---|---|
| X | | O |
| O | X | |
| O | X | O |

```
console.log(gameOver(board));
```

=> false

Enklast är om ni kontrollerar alla 8 möjligheter som finns för att vinna spelet, dvs kolla om någon av de horisontella linjerna, vertikala linjerna eller de 2 diagonala linjerna har samma tecken i alla positioner. För att t.ex kontrollera om översta raden har samma tecken i alla positioner så kan vi kontrollera `board[0][0] === board[0][1] && board[0][1] === board[0][2]`

*** function drawBoard(board) { }**

Denna funktion används för att rita upp spelplanen och kan definieras som följande:

```

function drawBoard(board) {
  let textBoard = "_____\n";
  for (let row = 0; row < board.length; row++) {
    textBoard += "|";
    for (let col = 0; col < board[row].length; col++) {
      if (board[row][col] === null) {
        textBoard += "_|";
      } else {
        textBoard += `${board[row][col]}|`;
      }
    }
    textBoard += "\n";
  }
  console.log(textBoard);
}

```

Ni kan kopiera koden, men se till att testa den och förstå vad koden gör, exempelvis med hjälp av debuggern i konsollen och sätta några breakpoints på lämpliga ställen.

3. När vi har definierat dessa 3 funktioner återstår bara att skapa gameloopen som bör vara en for loop som kör max 9 iterationer. I varje iteration bör loopen

```

* player = switchPlayer(player); // för att byta spelare
* const position = getUserInput(); // för att spelare ska göra val av ruta
* updateBoard(board, position, player); // för att uppdatera spelplan
* if (gameOver(board)) {
  break; // avbryt loop i förtid om någon har vunnit
}
* drawBoard(board); // för att rendera den aktuella spelplan

```

Med dessa ändringar bör vi nu ha ett fungerande spel som kan spelas.

Lycka till!