

Laboration, Tic-Tac-Toe steg 2 (4) – Validering och initering

Laborationerna i kursen består i att ni i fyra delsteg skall bygga spelet Tic-Tac-Toe (tre i rad).

För att komma igång i detta steg skall ni fortsätta på er lösning från steg 1.

I denna kurs kommer ni använda ett autorättningsystem för att kontrollera om ni uppfyller delstegens krav. Ni kan hitta autorättningsystemet i Canvas eller här: <https://informatik-webbkurser.hotell.kau.se/JSAutoFeedback/>

När ni lämnar in laborationen så kräver vi att ni lämnar in hela projektet (alla filer) samt en bild på autorättningsystemet där det står att ni inte har några fel i koden.

Er uppgift i detta andra delsteg är att i filen **JSFunctions.js** skriva kod för att validering inmatad data(Nicknames och valda färger) utifrån kraven nedan. När valideringen är klar skall ni även skriva kod för att initiera (starta upp) spelet.

Ni skall använda er av unobtrusive JavaScript och `addEventListener()` för händelsehantering.

OBS. Ni skall kommentera all kod i er lösning (ej HTML- eller CSS-kod, enbart er JS-fil)

- Lägg en lyssnare för händelsen "load". Vid händelsen skall följande ske:
 - Anropa metoden "initGlobalObject" i `oGameData`
 - Lägg till klassen *d-none* på elementet med id `game-area`
 - Lägg en lyssnare på "Starta spelet!"-knappen som lyssnar efter klick-händelsen.
- När knappen klickas skall funktionen `validateForm` anropas.
- I `validateForm` skall följande kontrolleras med hjälp av undantagshantering (try-catch):
 - Nickname (för båda spelarna) måste vara minst 5 tecken långt
 - Nickname för spelare 1 och spelare 2 får inte vara lika
 - Valda färger får **INTE** vara vit, svart eller lika för båda spelarna.

Om något av ovanstående inte är uppfyllt skall ett fel-meddelande visas i p-elementet med `id="errorMsg"`.

- Om villkoren ovan är uppfyllda, gör följande:
 - Anropa metoden "initiateGame".
- Definera (skapa) metoden ovan (initiateGame). Metoden tar inga invärden och har inget returvärde.
- Lägg till kod för följande i funktionen ovan:
 - Göm formuläret genom att lägga till klassen "d-none" på formulärelementet.
 - Visa spelplanen genom att ta bort klassen "d-none" på elementet med id "game-area"
 - Ta bort textinnehållet i elementet med id "errorMsg".
 - Spara Spelare 1's namn i objektet oGameData (oGameData.nickNamePlayerOne)
 - Spara Spelare 2's namn i objektet oGameData (oGameData.nickNamePlayerTwo)
 - Spara Spelare 1's färg i objektet oGameData (oGameData.colorPlayerOne)
 - Spara Spelare 2's färg i objektet oGameData (oGameData.colorPlayerTwo)
 - Töm spelplanen genom att använda querySelector för att hämta ut alla td-element på planen och sedan loopa igenom dem och sätta textContent="" samt bakgrundsfärg till vit.
 - Deklarera de lokala variablerna "playerChar" och "playerName"
 - Slumpa fram vilken spelare som skall göra första draget genom att:
 - Slumpa fram ett tal mellan 0 och 1
 - Om talet är mindre än 0.5 sätt:
 - playerChar = oGameData.playerOne
 - playerName = oGameData.nickNamePlayerOne
 - oGameData.currentPlayer = oGameData.playerOne
 - Om talet är större eller lika med 0.5 sätt:
 - playerChar = oGameData.playerTwo
 - playerName = oGameData.nickNamePlayerTwo
 - oGameData.currentPlayer = oGameData.playerTwo
 - Ändra texten i h1-elementet som ligger i div-elementet med klassen "jumbotron" till "Aktuell spelare är XXX" där XXX är nickname på aktuell spelare.

TicTacToe

Ange nickname för spelare 2!

Nickname spelare 1:

Bertil

Färg spelare 1:



Nickname spelare 2:

nickname

Färg spelare 2:



Starta spelet!

Aktuell spelare är Ulla (O)

