

Labb 4 – Interaktivitet II

Multimedia 7.5 hp VT-14

Introduktion

I denna laboration kommer vi att arbeta mer med javascript för att skapa interaktivitet. Tanken är nu att vi fördjupar oss i alla delarna – javascript, html såväl som css. När man jobbar med nya tekniker är det ofta troligt att man behöver stanna upp och söka information, så var inte rädd för att aktivt använda internet till att leta information.

Inlämning

Din inlämning ska bestå av en .zip-fil eller .rar-fil (inga andra komprimeringsformat är tillåtna!) innehållandes följande (med följande struktur):

- labb2_fornamn_efternamn.zip
 - uppgift1 (mapp)
 - * grid.js
 - * index.html
 - * grid.css
 - uppgift2 (mapp)
 - * randomizer.js
 - * index.html
 - uppgift3 (mapp)
 - * menu.js
 - * menu.css
 - * index.html

Kodstandard

- Ingen Javascript eller CSS ska placeras “inline” i något htmdokument. All javascript-kod skall alltså skrivas i .js-filer och css i .css-filer.
- All kod ska vara korrekt indenterad! (Läs mer om indentering på [htmlhunden](#) om du är osäker)

Uppgifter

Nedan följer uppgifterna som resulterar i inlämningarna ovan.

Uppgift 1 Rutnät av bilder

I denna uppgift skall ni skapa en sida som har ett rutnät/schackbräde av bilder. Bilder finns att hämta [här](#) och HÄR OSV.

1. Börja med att skapa en `.html`-fil
2. Skapa sedan en `.js`-fil där du kommer att skapa metoder som du behöver.
3. Skapa sedan en `.css`-fil där all formatering skall finnas!

Krav

- Rutnätet skall göras dynamiskt, det är alltså inte ok att skapa ett rutnät som är av en hårdkodad storlek. Således kan det t.ex. vara bra att abstrahera ditt program till en metod som du förslagsvis kallar `drawGrid` och som tar två siffror: `rows` och `cols`. Ditt rutnät ska alltså kunna skapas som `2x2` såväl som `20x30`.
- Rutnätet ska alltid täcka 100% av fönstrets bredd och 100% av fönstrets höjd.

Uppgift 2 Spelet Memory

Denna uppgift går ut på att skapa en enklare variant av spelet memory. Spelplanen består av ett antal bilder, i par. Spelet går ut på att man ska välja två bilder, om de två bilderna är likadana så får man ta bort dem ifrån spelplanen. När alla bilder är borta ifrån spelplanen är spelet över och man har vunnit!

I vanliga fall spelar man med uppochnedåtvända bilder, för att spelet förstås inte ska vara busenkelt. Detta kommer vi att strunta i när vi bygger vårt eget Memory. Alla brickor i ditt memory-spel ska alltså hela tiden vara "uppåtvända". Man ser alltså alltid alla bilder, och spelet går helt enkelt ut på att klicka på par.

Krav

- Din spelplan måste vara minst `3x2`
- När en korrekt matchning är gjord ska båda brickorna tas bort
- När en inkorrekt matching görs ska ett meddelande visas för användaren

- När spelet är slut ska ett meddelande visas för användaren
- De bilder du använder måste vara av rimlig storlek, både i dimensioner och i “tyngd”.

Förslag på tillvägagångssätt

1. Skapa en HTML, en CSS, en JS-fil, och “koppla ihop” dem.
2. Skriv ut dina bilder i din HTML.
3. Hitta ett sätt att identifiera par, t.ex. genom ID:n och/eller `class`:er.
4. Fundera över vidare steg som behöver tas och bryt ned i delproblem.