

Lab 2: Sänka skepp (Battleship) i javascript

Överblick

Den här labben har som mål att göra dig bekväm med att använda javascript, HTML och CSS. I den här labben så ska du göra en partiell (eller komplett) implementation av spelet Sänka skepp. Det finns många versioner av spelet och du bör välja en av de vanligare versionerna.

Specifika krav

1. Spelplanens storlek måste vara genererad och får inte hårdkodas som ett 9x9-grid, använd en variabel istället.
2. Det ska vara enkelt att ändra storleken på spelplanen (det ska räcka att ändra värdet av en variabel på ett ställe i programkoden)
3. HTML-taggen <table> får inte missbrukas för layout av spelplanens olika delar. Tables ska endast användas för tabulering av data, [inte för layout](#).
4. Programmet ska fungera i både Firefox och Chrome.
5. Använd "use strict" i hela koden.
6. Använd ett linting-tool (Det får konfigureras valfritt).
7. Använd ÅTMINSTONE 5 olika typer av CSS attribut, CSS-koden ska vara i separat fil från HTML- och JavaScript koden. Fokus är på att det ska vara lärorikt och roligt [snarare än vackert](http://www.themostamazingwebsiteontheinternet.com/). (<http://www.themostamazingwebsiteontheinternet.com/>).

Definitioner

"Ruta": Godtyckligt klickbart HTML-objekt av ditt val.

"Skepp": Ruta som är visuellt skilt från de andra klickbara HTML-objekten när det klickas på.

Uppgift

Du ska implementera spelet Sänka Skepp. Spelet antar att det är två spelare vid samma dator.

Spelet består av två faser:

1. Spelarna placerar sina skepp och då är spelarna ärliga och tittar bort när den andra spelaren placerar sina skepp.
2. Spelarna skjuter ned varandras skepp.

Det är dock inte ett krav att båda gör sin fas ett innan fas två. Uppgiften kan förenklas till att någon skjuter klart innan hen placerat ut sina skepp. Alltså: Spelare 1 placerar skepp, spelare 2 skjuter. Spelare 2 placerar skepp, spelare 1 skjuter. Detta får ni implementera valfritt.

Utplacering av skepp på spelplanen ska ske genom att man klickar på en ruta, vill spelaren ångra sin placering så ska man kunna klicka på det redan utplacerade skeppet för att ta bort den. Det ska finnas en start-knapp som startar spelet/går vidare till nästa fas.

Bas-versionen av spelet kommer inte att kolla att skeppens form är rätt, detta lämnas till bonusuppgiften. Det ska finnas en mekanism för att spelarna ska kunna hålla koll på:

- Antalet placerade skepp.
- Antalet skepp man sänkt.
- Antalet skott man avfyrat.

Det ska också finnas en knapp som rensar brädet som man kan använda när man har spelat klart.

Uppgiften är ganska fri i sin form, men spelet måste vara spelbart och ni får inte hitta på regler godtyckligt utan måste utgå från Wikipedia eller annan källa för regler.

Tips:

När du debuggar använd dig av konsolen för att skriva spårutskrifter.

För att öppna konsolen på Firefox, tryck Ctrl + shift + k.

https://developer.mozilla.org/en/docs/Tools/Web_Console.

För att öppna den i Chrome, tryck, Ctrl + Shift + j.

<https://developer.chrome.com/devtools/docs/console#using-the-console-api>

Du kan inspektera element i de flesta webbläsare för att få mer förståelse för vad som faktiskt händer.

References

- Tips & trix: <https://modernweb.com/45-useful-javascript-tips-tricks-and-best-practices/>
- En tutorial: [W3schools tutorial](#)
- Strict mode: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode

Frågor

1. Vad är skillnaden mellan == och ===? Vad är "best practice" gällande vilken man ska använda?
2. Vad gör "use strict"?

Bonusuppgift X2

Modifiera din kod så att det bara går att placera ut skepp som är på rätt form. Med andra ord, de är raka och av längd 2, 3, 3, 4 och 5 (eller vad för regler du har använt).

Frågor:

Har javascript klasser? Är prototypes klasser? **Motivera ditt svar.**

Vad är callbacks i Javascript?

Hur används callbacks asynkront?