

Projekt: Brief

OBS: Projektet är nytt för HT24. Det kan finnas något fel i dokumentationen eller någon ofullständig beskrivning. Säg gärna till om du hittar något (erik.pineiro@mau.se). Eventuella ändringar till projektet kommer via anslag på Canvas.

Intro

Man kan få G eller VG i projektet.

För G: Lös minst 5 av de 6 föreslagna problemen (sidorna).

För VG: Bygg en sida där man kan spela Blackjack. Eller en sida där man kan spela Fyra i rad.

Oavsett vad du väljer att göra så måste du redovisa koden du lämnar in.

Det går bra att lämna in filerna och avvakta med redovisning. Informera gärna oss via mejl om du föredrar att inte redovisa i januari. **Notera dock att momentet inte är avklarat (och inga hp registreras) tills du har blivit godkänd i redovisningen.**

Det går bra att lämna in både G-lösning och VG-lösning. Med andra ord: Det går bra att lämna in en G-lösning och efter det sätta sig ner och jobba med en VG-lösning. Om du fixar det kan du sen lämna in VG-lösningen. Vi kommer alltid att utgå från senast inlämnad zip-fil. Skulle det visa sig att VG-inlämningen innehåller fel så kan du alltid falla tillbaka på G-inlämningen.

G eller VG spelar egentligen väldigt lite roll för din framtida professionella karriär. Men VG-problemen är lite mer komplexa och kommer att hjälpa dig bli bättre utvecklare.

G eller VG kan dock spela roll – tror jag – när man tillsätter platserna för studier utomlands.

ChatGPT och annan hjälp

Som vanligt är det helt ok att använda ChatGPT, koda tillsammans med andra eller fråga någon vän som kan programmera.

Glöm dock inte att:

- (1) Det är du som sökte till denna utbildning, som vill få något ut av den och som har planer på att öppna nya möjligheter i arbetsmarknaden.
- (2) Du behöver redovisa koden som du lämnar in.

Problem

Det finns sex föreslagna problem (eller sidor) i mappen "sidor".

De beskrivs i videorna. Readme-filerna innehåller för det mesta väldigt lite info. Detta eftersom det sidorna ska göra är ganska uppenbart genom att kolla på videon.

Du ska lösa minst fem av problemen. Vid sidan om dessa fyra ska du även koda "hemsidan" med länkarna till problemen.

Sidorna ska se ut som på videorna. Du har dock fria händer vad gäller storlekar, färger och typsnitt.

Som vanligt förväntas du kunna redovisa all kod du lämnar in. Men det är inte säkert att vi hinner prata om alla problem under den korta tiden vi har.

VIKTIGT:

- A. Det får inte förekomma repeterad CSS-kod. CSS-kod som gäller för alla sidor måste placeras i en fil (common.css) som alla HTML-filer länkar till.
- B. Det får inte förekomma repeterad JS-kod. Jag har identifierat det som är gemensamt för alla sidor (förutom hemsidan) och som måste kodas i funktioner i en JS-fil som alla sidor länkar till (common.js):
 - A. Fylla griden med slumpmässiga siffror
 - B. Returnera en slumpmässig siffra
 - C. Skapa home-länken

Blackjack

Regler för Blackjack hittar du här:

https://sv.wikipedia.org/wiki/Black_Jack

Du behöver inte implementera Split eller Över/Under.

Du kan skapa en Blackjack för bara en spelare (plus banken).

I korthet: Spelaren startar med en viss fast belopp att satsa med. Vid varje omgång kan hen satsa så mycket hen vill av pengarna hen har kvar. Banken har alltid tillräckligt med pengar att ge spelaren om hen vinner. Om spelaren gör slut på sina pengar så avslutas spelet med relevant meddelande.

Spelet måste se " snyggt " ut, men det finns inga formella krav annat än korten måste inkludera färgikoner (klöver, ruter, etc).

Fyra i rad

Alltså detta spel: https://sv.wikipedia.org/wiki/Fyra_i_rad (kallas Connect4 på engelska).

Två spelare ska kunna spela, självklart på samma dator (i DU3 ska vi lära oss hur man spelar från två olika datorer, kopplade över en server). Exakt hur du löser hur man lägger brickorna är upp till dig men brickorna ska läggas som i spelet: uppifrån, och så "åker de ner" till sin plats.

Webbsidan måste själv upptäcka, och meddela, att en spelare har fyra i rad och vunnit. Brickorna som skapar fyra-i-rad ska då markeras. Efter det ska man kunna spela igen.

Webbsidan ska också upptäcka, och meddela, att matchen blev oavgjort: om ingen spelare har vunnit och det inte finns plats för brickor i brädan.

Extra saker du kan koda för att imponera på läraren:

- Webbsidan håller koll på hur många ggr varje spelare har vunnit.
- Spelare kan välja färg på sina brickor.
- Klurigt! Sidan upptäcker att ingen spelare kan vinna spelet, även om det finns plats för fler brickor på "brädan".
- Annat som du själv kan komma på

Spelet måste se " snyggt " ut, men det finns inga formella krav.

Allmänna krav (gäller G och VG)

- I. Du ska skapa en Github-repository och använda den i utvecklingen. Minst 20 meningsfulla commits.
 - A. Det ska finnas en länk på din sida som länkar till repon. Säkerställ att den är publik.

- II. Du ska publicera din sida på webben. Det ska finnas en tydligt placerat länk som länkar till din sida på webben.
- III. Du måste använda minst ett typsnitt från Google Fonts, och det måste ske via @font-face.
- IV. Zip-filen som du lämnar in måste vara mindre än 400Kb. Ladda gärna ner den från Github.
 - A. 400Kb inkluderar typsnittsfilerna.
 - B. Ta inte med några filer än just de som behövs för webbplatsen.
 - C. Om du gör någon av VG-problemen och använder någon större bild så går det bra att komma upp till 600Kb för hela zip-filen.
- V. Det får inte förekomma några kommentarer i koden.