

Arbeidskrav PGR102

Høsten 2025

Introduksjon til oppgaven

- ★ Utleveres 3. Oktober
- ★ Innleveringsfrist 24. Oktober kl 10:00
- ★ Arbeidskravet leveres individuelt eller i grupper opp til 3 personer
- ★ Arbeidskravet må bestås for å kunne ta eksamen i faget
- ★ Alle hjelpe midler er tillatt. Kopiering av kode fra andre eller AI er ikke tillatt
- ★ Det oppfordres til å spørre studentassisterter og foreleser om hjelp, ta gjerne kontakt via Canvas eller Mattermost

Praktisk informasjon

Angående bruk av AI og KI-botten som er lagt ut via Canvas:

- Klipp-og-lim er ikke godkjent
- Kan brukes for å få tips og hjelpe deg i riktig retning
- Lag for eksempelvis en sjekkliste med underpunkter, og få den til å hjelpe deg med hvert underpunkt individuelt, og ikke hele oppgaven på en gang, da blir det fort rot!
- Du **MÅ** forstå all kode du selv har skrevet

Angående navngivning på variabler og funksjon bør disse være selv-forklarende og være på engelsk, hvor kommentarer i koden din kan godt være på norsk

Kodebesvarelse

I denne oppgaven får du en mappe med ferdiglaget HTML og CSS-kode. Du skal ikke røre den ferdiglagde koden for å løse arbeidskravet, du skal kun kode Javascript for å legge til ”muskler” i programmet. I script-taggen ligger det et array og et objekt som inneholder data for karakterene. Disse dataene skal du bruke gjennom oppgaven. Du trenger ikke bruke all ferdiglaget kode eller data om du ikke ser det nødvendig i din besvarelse, her er det mange måter å løse oppgavene på. Løs de på din måte! Du kan lage en egen Javascript-fil om du vil, men det er ikke nødvendig.

Arbeidskravets kodebesvarelse består av ”**Påkrevde funksjonaliteter**”, ”**Påkrevde teknikker**” og ”**Tilleggsfunksjonalitet**”. Pass på at du leser gjennom disse nøyne før levering og sjekk at du har fått med deg alt som kreves.



Påkrevd funksjonalitet

1. Når brukeren trykker på en av heltene, skal helten angripe dragen

- Det skal komme en alert hvor det står:
"(Heltens name) har gjort (heltens damage) skade på (Dragens name)!"
- Dragens liv (currentHP-variabelen) skal bli trukket med heltens damage, avhengig av hvilken helt den ble angrepet av.
- Dragens currentHP blir oppdatert

Påkrevde teknikker: alert, function, click-event, manipulering av array-objekter

2. Dragen skal gjøre et motangrep umiddelbart når den blir angrepet av en helt.

- Når dragen blir angrepet, skal dragen angripe en tilfeldig helt tilbake.
Det er altså $\frac{1}{3}$ sjanse for hver helt å bli angrepet.
- Det skal komme en ny alert hvor det står:
“(Dragens name) har angrepet (Heltens name)!”
- Helten som ble angrepet sitt liv (currentHP) skal bli trukket med dragens damage.
- Det er en tilleggsfunksjonalitet (se lenger ned i oppgavesettet) å ta stilling til hva som skjer dersom dragen angriper en død helt.

Påkrevde teknikker: alert, function, manipulering av array-objekter, Math.floor, Math.random

3. Spillet er tapt! 😞

- Dersom alle heltene er døde og dragen har en currentHP over 0, er spillet tapt.
- En alert skal dukke opp hvor det står:
“Spillet er tapt! (Dragens name) har vunnet!”

Påkrevde teknikker: alert, function, .remove, if-else eller array-metode

4. Spillet er vunnet! 🎉

- Dersom dragens currentHP blir 0 eller mindre og ikke alle heltene er døde, har helten(e) vunnet spillet!
- En alert skal dukke opp:
“Gratulerer, du har vunnet spillet!”

Påkrevde teknikker: alert, function, .remove, if-else/switch/array-metode

Tilleggsfunksjonalitet

Grupper på 1 person: *Velg minst 1 tilleggsfunksjonalitet*

Grupper på 2 personer: *Velg minst 2 tilleggsfunksjonaliteter*

Grupper på 3 personer: *Velg minst 3 tilleggsfunksjonaliteter*

Du/Dere står fritt til å velge de(n) tilleggsfunksjonaliteten du/dere vil, avhengig av hvor mange dere er på gruppen deres.

1. En helt er død!

- Dersom en helt sin currentHP blir 0 eller mindre, skal helten(e) ikke kunne bli angrepet av dragen siden de er døde (oppgave 2). De døde heltene forsvinner også fra nettleseren. Dersom dragen er død, skal dragen forsvinne fra nettleseren.

Forslag til teknikker:

Oppdater alive: true/false avhengig av om helten er død eller ikke, deretter filtrer arrayet slik at dragen kun vurderer å angripe levende helter. Se på .remove for fjerning av HTML-elementer fra nettleseren.

2. Henriette Healer

- Henriette Healer gjør damage på dragen og healer sine heltevenner med 100HP hver gang hun trykkes på!
- Det skal dukke opp en liten flaske med healthPotion i nettleseren når hun klikkes på, se i assets-mappen.
- Hun kan ikke heale over maxHP, eller heale døde helter.

3. Navnene og livene til karakterene skal vises frem og oppdateres i nettleseren

- I p-taggene hvor det står "Heltens navn"/"Dragens navn" skal du vise frem de riktige navnene fra arrayet
- I p-taggene som viser livet til heltene og dragen (f.eks. 200 / 200 HP) skal du vise frem riktig informasjon for hver karakter fra arrayet slik: "(currentHP / maxHP) HP"
- currentHP i nettleseren skal oppdateres når karakteren tar damage

4. Lifebar-farge oppdateres!

- Den grønne life-baren blir mindre og mindre avhengig av karakteren sin currentHP.

5. Damage Boost!

- Hvis brukeren trykker på knappen "D" får alle helter en Damage Boost på 10%! (Heltene gjør 10% mer skade)
- Det skal komme opp en tekst på nettleseren som sier fra at Damage Boost er aktivert

6. setTimeout for at angrepet fra dragen tar tid

- Bruk setTimeout (https://www.w3schools.com/jsref/met_win_settimeout.asp) for at angrepet fra dragen ikke kommer umiddelbart, men kommer først etter 2-3 sekunder. Det skal ikke være mulig å klikke på heltene i venteperioden.

7. Spillet er vunnet/Spillet er tapt

- I stedet for bare en alert som gir en melding når spillet er tapt eller vunnet, lag en div-tag og en h1-tag i Javascript og gi de styling ved å gi de klassenavn i Javascript og vis dette i nettleseren. De skal ha forskjellig styling avhengig av om spillet ble vunnet eller tapt.

Lykke til!