

Deleksamen 1-4 studentnummer: 143262 / Yahye Abdi Ahmed

Introduksjon

Deleksamen 1:

-kommentar i denne filen som beskriver hvordan kodene fungerer. Da jeg leverte denne deleksamen, forstå jeg ikke helt hvordan kode delene hang sammen.

Deleksamen 2:

-ting som blir fikset: restart knapp visning når spill er uavgjort, restart knapp forsvinner når spiller er.

-fikser der riktig spiller vinner. Problem med setInterval() fikset med if statement

-fargen på utskrift når spiller vinner mister gule fargen ved restart av spill.

Deleksamen 3:

-teller startet feil fra 6sec, skal telle ned fra 60sec.

Deleksamen 4:

- Lese oppgave i canvas
- Html og CSS
- Tegnet 2 spiller
- Player1 og player2 beveger seg
- Player1 og player2 vegg check
- Lage 2 ball, 1 ball for hver spiller
- Spiller og ball kollisjon skjekk: tok langtid, vanskelig
- Ball1 og ball2 skytting: vanskelig, fikk endel hjelp med det.
- Spiller liv tap: bugg der spiller liv gikk ned konstant, fikset etter tilbakemelding fra medstudent.
- Gameover funksjonalitet for spill. Lagde 2, en gameover for spiller1 og en for spiller2
- Restart knapp

Deleksamen 1

Hvordan kode i oppgaven fungerer

Html/CSS:

I body tag er det en div element med id "mainframe" og annen div i den med id "background". I CSS er #mainframe div stilsett med 100 view hight/width størrelse med rød backgrunnsfarge. Fargen byttet til grønn i script. #background div har 80/80 view hight/width størrelse. #background div har bakgrunnsbilde url med verdi "start-cat.jpg. Dette bildet blir byttet med annen script når alle feilene er rettet. Siden #background div ligger i #mainframe div, blir grønnfarge vises på området mellom de to divene. #mainframe div har ikke z-index verdi, så vi ser ikke hele grønn backgrunnsfargen for #mainframe. #background div dekker 80vw/vh av skjermen.

Script:

Script manipulerer de to divene. For #background div er 8 variabler som brukt og 1 variabel for #mainframe div.

Mainframe div:

Først har vi 2 variabel som hver inneholder del av navnet til riktig bildet som string. Bildet navn er "error-hero".

Variabel examtype har string verdi "error" og variabel studenType har string verdi "hero". Disse variablene brukes i siste if statement for byttet #background div bakgrunnsbilde bildet url verdi.

Variabel maxScore har verdi 1337 og variabel currentScore har verdi 330. For å gjøre disse to variablene lik, er det 3 variabler til.

Variabel addedScore er verdi currentScore(330) * 4 som er 1320. Og variabel missingScore har string verdi "17". Summen av disse to variabler blir 132017 med console log og lagret i tredje variabelen som currentScore variabel som redeklarerert igjen med sum verdien. Verdien er string selv om det ser ut som nummer i console log. Console log typeof missingScore viser string og addedScore nummer. Summen viser number17, nummer og string som er 17.

Først if statement kjører hvis maxScore = 1337 og det er sant. I if er det variabel successImage som konkatenerer variabel examtype("error"), bindestrek og variabel "hero". Variabel successImage verdi blir da "error-hero". Det er filnavnet til bildet som katte bakgrunnsbildet blir byttet til.

Andre if statement på linje 67 har i parameter at currentScore variabel er lik maxScore variabel og dette stemmer. Begge variablene har sum 1337, men de er ikke samme datatype. maxScore er nummer datatype og currentScore streng datatype. Siden parameter i if har bare en =, så blir det kjekket at begge variablene er samme datatype. Derfor er denne if sant.

I if blir bakgrunnsbilde i barne div med id "background" byttet ved å hente div document.getElementById() metode. Style.backgroundUmage er css selektor, med .style har vi tilgang til alt i css i javascript. Her blir background byttet med url som tar imot variabel successImage, som har "error-hero" som verdi og ".jpg" blir konkatenerert for å fullføre riktig fil navn for bildet navnet "error-hero.jpg".

Bakgrunnsfarge fargen foreldre div med id "mainFrame" blir byttet sty.backgroundcolor med streng som har "rgb" konkatenerert med variabel green green som har streng verdi "0,128,0", 0 i rød, 128 i grønn og 0 i blå. Dette gir grønnfarge bakgrunn for foreldre div.

Deleksamen 2

- I linje 264, restart knapp vises når spillet er uavgjort også og ikke bare når en spiller har vunnet.
- Riktig spiller vinner: linje 343. Jeg hadde et problem der feil spiller vant hele tiden. Sammen med par medstudenter prøvde vi å finne løsning, men ingen av oss klarte det i flere dager helt til en av oss løste med bruk av en variabel og if statement. Måten koden fungerer er forklart i kommentar der koden ligger.
- Linje 403, Bakgrunnsfarge for spillervunnet fikset. Fargen for innholdet div med id "playerturn" blir gult bakgrunnsbilde fjernet i restartGame funksjon. Dette har jeg gjort med backgroundColor verdi tom streng.
- I linje 405, restart Knappen fikset når spiller har vunnet ved å gi knappen css stil display none.

Deleksamen 3

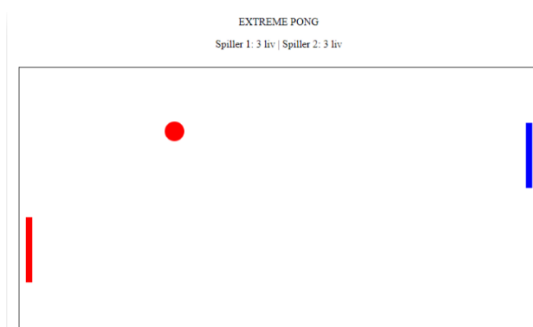
- Linje 455, verdi på variabel med navn count blir byttet 6 to 60. Dette er sekunder for spill varighet. Jeg hadde på 6sek for lettere jobbe med restart knappen og glemte å bytte før innlevering da vi hadde deleksamen3.

Deleksamen 4

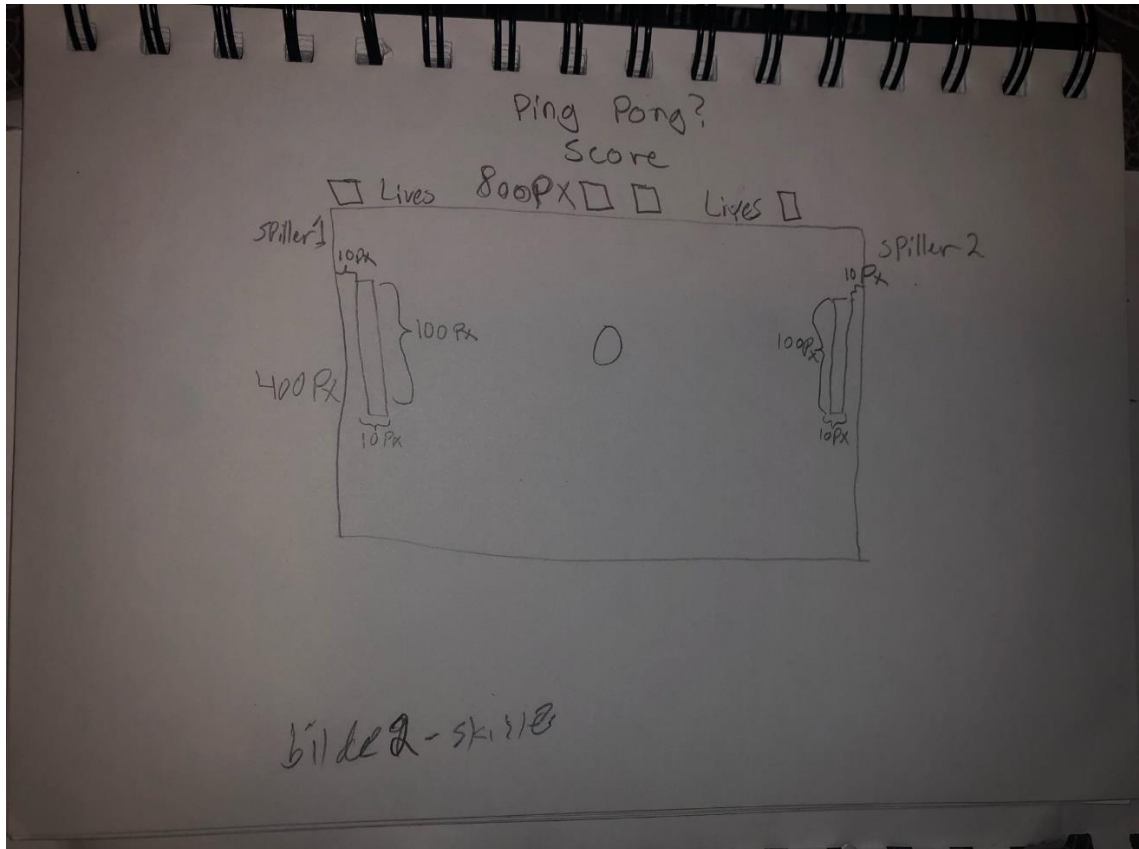
Start:

Det første jeg gjorde var å lese oppgaven i canvas og se gjennom spill utvalg. Jeg valgte nesten med en pong spillet, men jeg hadde misforstått hva slags pong spill det var. Jeg trodde det var som ping-pong der man skal holde ballen i live. Jeg fant ut senere at spillere skulle skyte ball på hverandre og at det er meningen at spillere skal unngå å bli truffet.

Html og CSS:



- Bilde1: skisse fra selv oppgaven

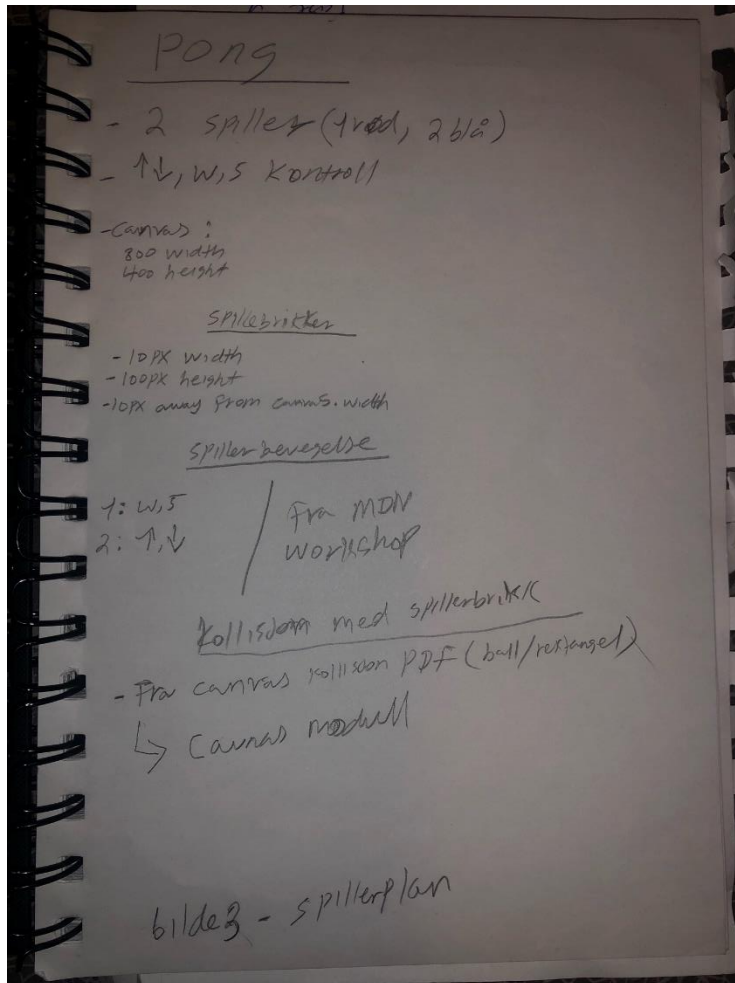


- Bilde2: Papir og blyant av skissen med måling. Noe ting er forandret som hvor spiller liv ligger

Jeg har brukt disse skissene for å lage spill vindu som vises på skjermen i pc, mobil, ipad.

Html og css er veldig basic med div container med elementer i html og flexbox i CSS. Font familie Rye har jeg importert i css-filen.

Tegne 2 spiller og ball:



- Bilde3: spillerplan - plan for spiller.

Jeg har tegnet player1 og player2 firekant, en rød ball på canvas. Måten 2d elementene er tegnet ligger kommentert ferdig i siste js-filen. Gamle js fil i starten ligger "testingScript" mappen og heter js.js. Der kan vi 2 spiller og en ball mellom. Det ingen animasjon i denne gamle filen.

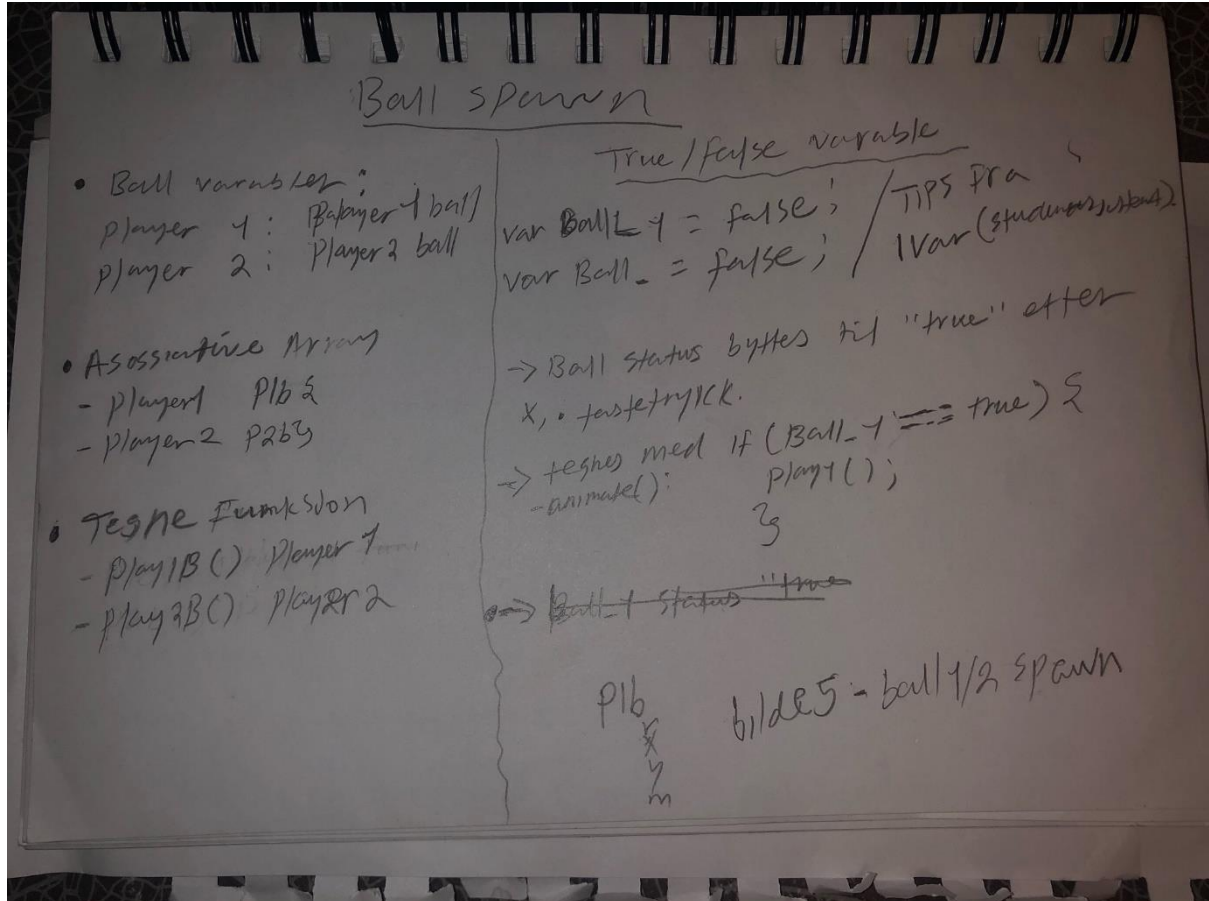
Player1 og player2 animasjon og vegg check:

- Bilde4: kollisjon-tries prøvelser. Andre ifer er brukt i koden. Men hoved poeng er å kjekke krasj test for ballen i baksiden av canvas kanten. Det ser samme for venstre siden når player2 skyter til ballen sin til venstre.

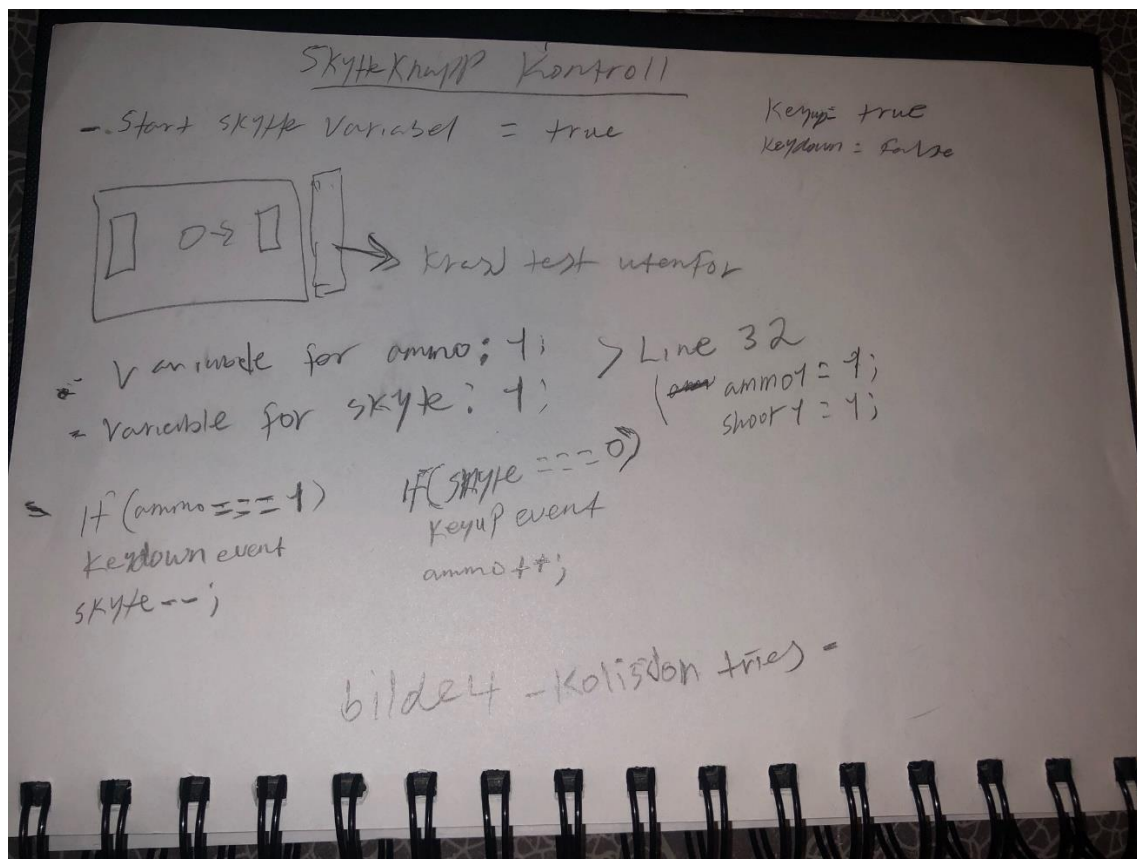
Her har jeg lagd bevegelse for 1 spiller og gjøre samme for andre spiller etter. Kodebitene er for spiller 1 bevegelse er kommentert ferdig i siste js-filen og hvor inspirasjon for kodene er hentet fra som kommentar også. Gamle js-filen for denne delen ligger i "j3copy.js" i testingScript mappen. Først lagde jeg tastatur hendelse lytter for keydown, keyup med hver sin lytter funksjon. Player1 og player2 har ikke felles tastatur addEventListener og funksjoner, så hver spiller har egen keyup event med sin

funksjon og keydown event med sin funksjon. Bevegelsene og krasj test med y-akse topp og bunn skjer update funksjon, animasjon skjer i animate funksjon.

Lage 2 ball, 1 ball for hver spiller:



- Bilde4: ball1/2 spawn. Denne spawn for ball uten hensyn på at den skytes fra spiller sin midtpunkt, altså for at x og punktum tast i tastatur fungerer og ballen skytes ut.



- Bilde5: kollisjon-tries prøvelser. Andre ifer er brukt i koden. Men hoved poeng er å kjekke krasj test for ballen i baksiden av canvas kanten. Det ser samme for venstre siden når player2 skyter til ballen sin til venstre.

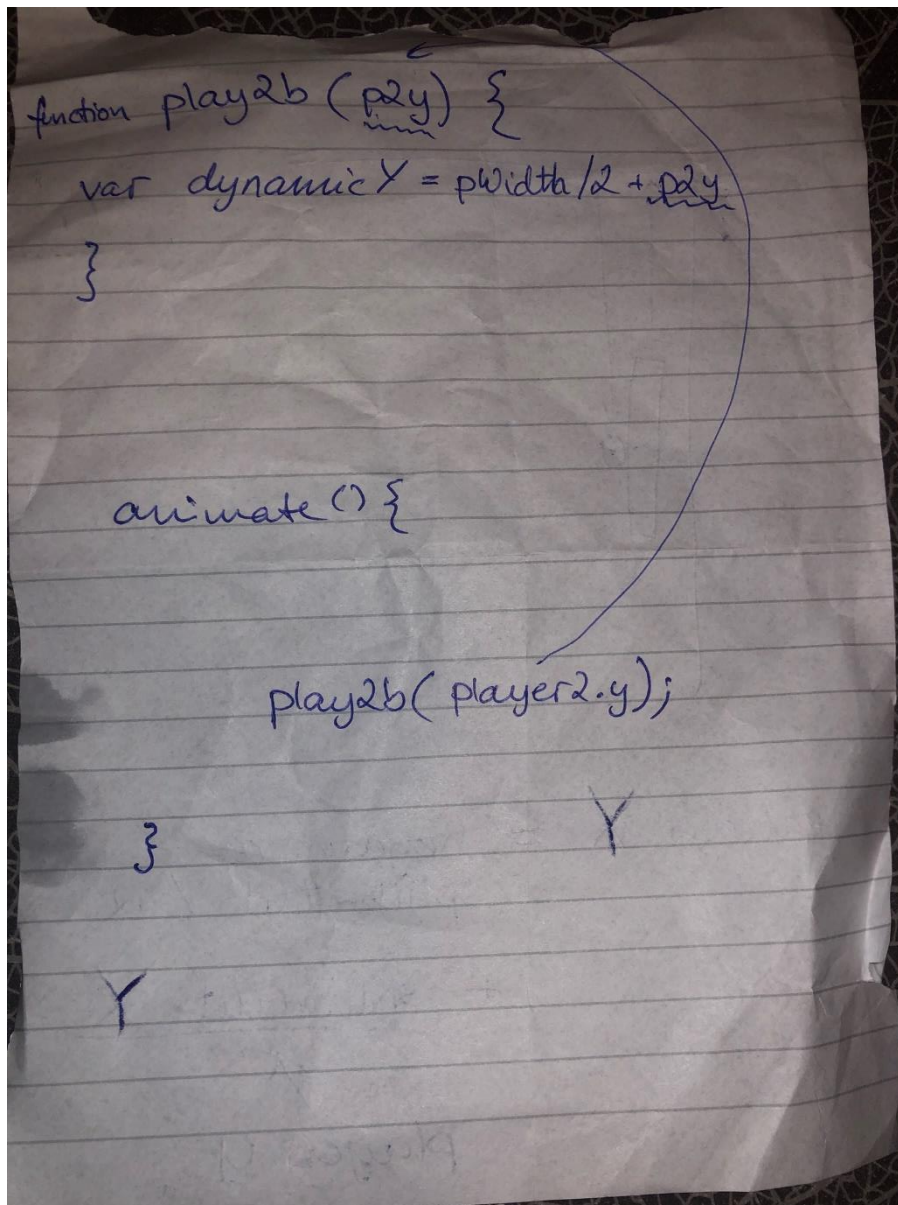
Kollisjon formel for ball krasj med spiller firekant:

https://hiof.instructure.com/courses/5112/pages/kollisjonsdeteksjon?module_item_id=136953

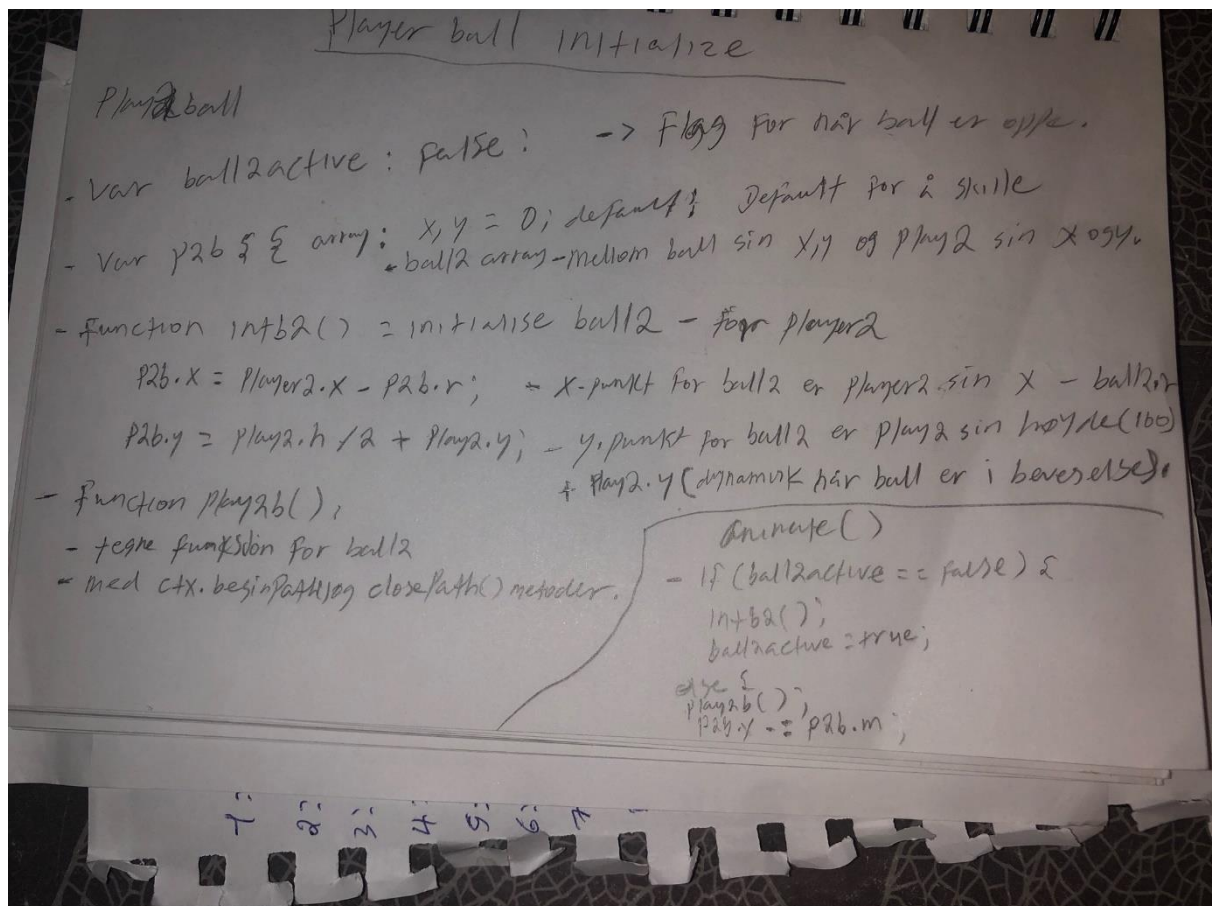
Denne delen av oppgaven hadde god del utfordring med. Det ligger i kommentaren i sist script for løsning. Gamle js-fil for denne delen er blant annet "js3mess.js" i testingScript mappe. I "js3-singelballs-working" skytes ballene ut bare fra midten i canvas høyde og bare en gang for hver spiller.

Ball1 og ball2 skytting: vanskelig del, fikk endel hjelp med det:

Jeg brukte ganske mye tid på å prøve på denne delen og satt fast egentlig.



- Bilde6: Marius. I veiledningstime fikk jeg tips fra lærer om å bruke parameter til å få ballen å spawne i midten av spilleren som skyter den ut, men jeg fikk et problem der ballen bevegde seg med spillers bevegelse i y-aksen.



- Bilde7: ballint: Det ble løst med bruk av initialize funksjon for skille mellom spiller og ball y-akse koordinater etter stund forsøk med studentassistent.

Jeg fikk da god hjelp med en av ballene, ball 2 for player2. Den ligger i gamle js-filen

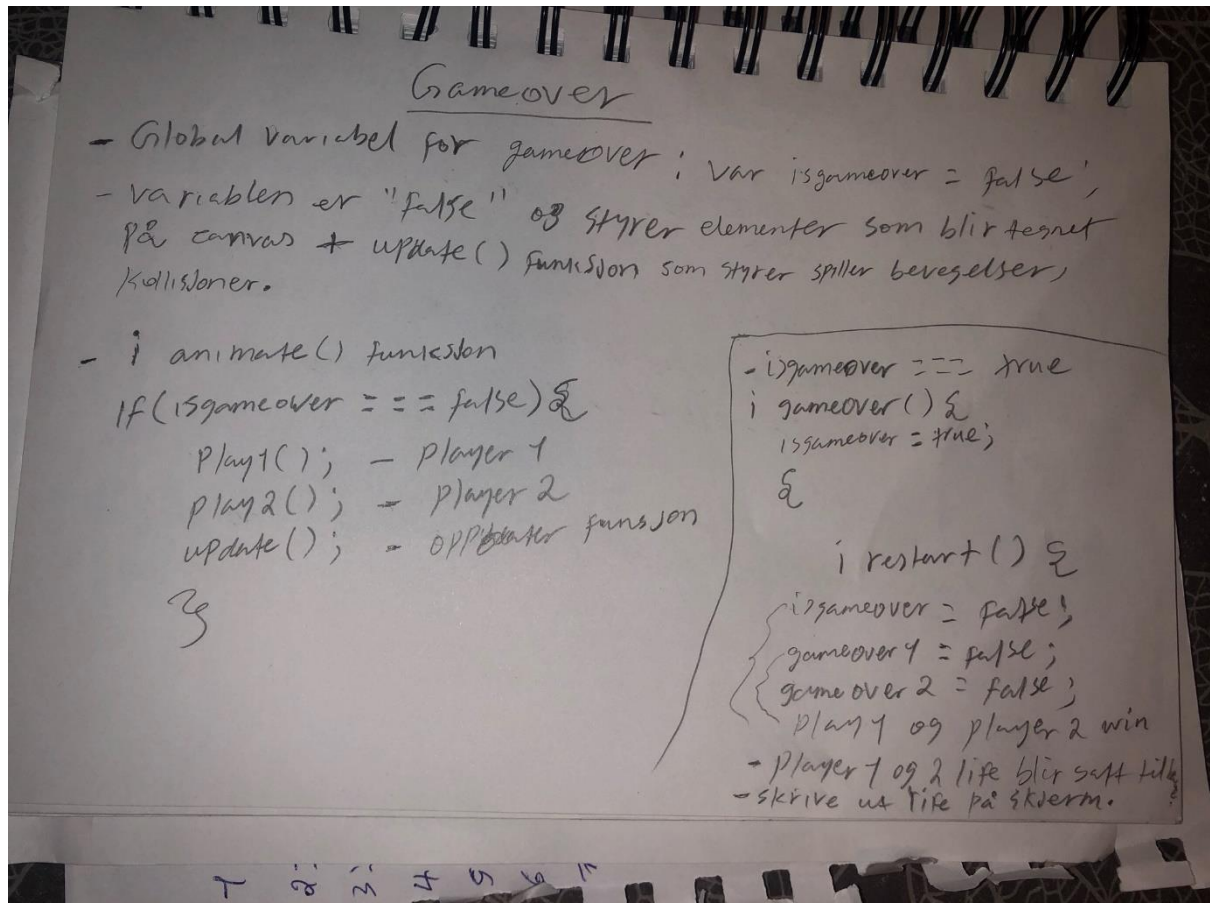
"js3Ball2ServeralShotsOk.js" i testingScript mappe. Begge ballene fungerer riktig i gamle js-filen

"js3both2BallWorking.js". Måten kode bitene er satt opp er kommentert i den siste js-filen, "js3.js".

Spiller liv tap: bug der spiller liv gikk ned konstant:

Når ballen krasjet med spillet, så fortsatte truffet spillet å miste liv, men jeg fant løsning etter kommentar fra medstudent som sa ballen sin x og y-akse tegnepunkt blir igjen på krasjpunktet. Så jeg tok ball sin x og y-akse tegnepunkt tilbake vekk fra krasjpunktet for både ball1 og ball2. Det løste spiller liv buggen.

Gameover funksjonalitet:



- Bilde8: gameover. Strukturen for omtrent sånn.

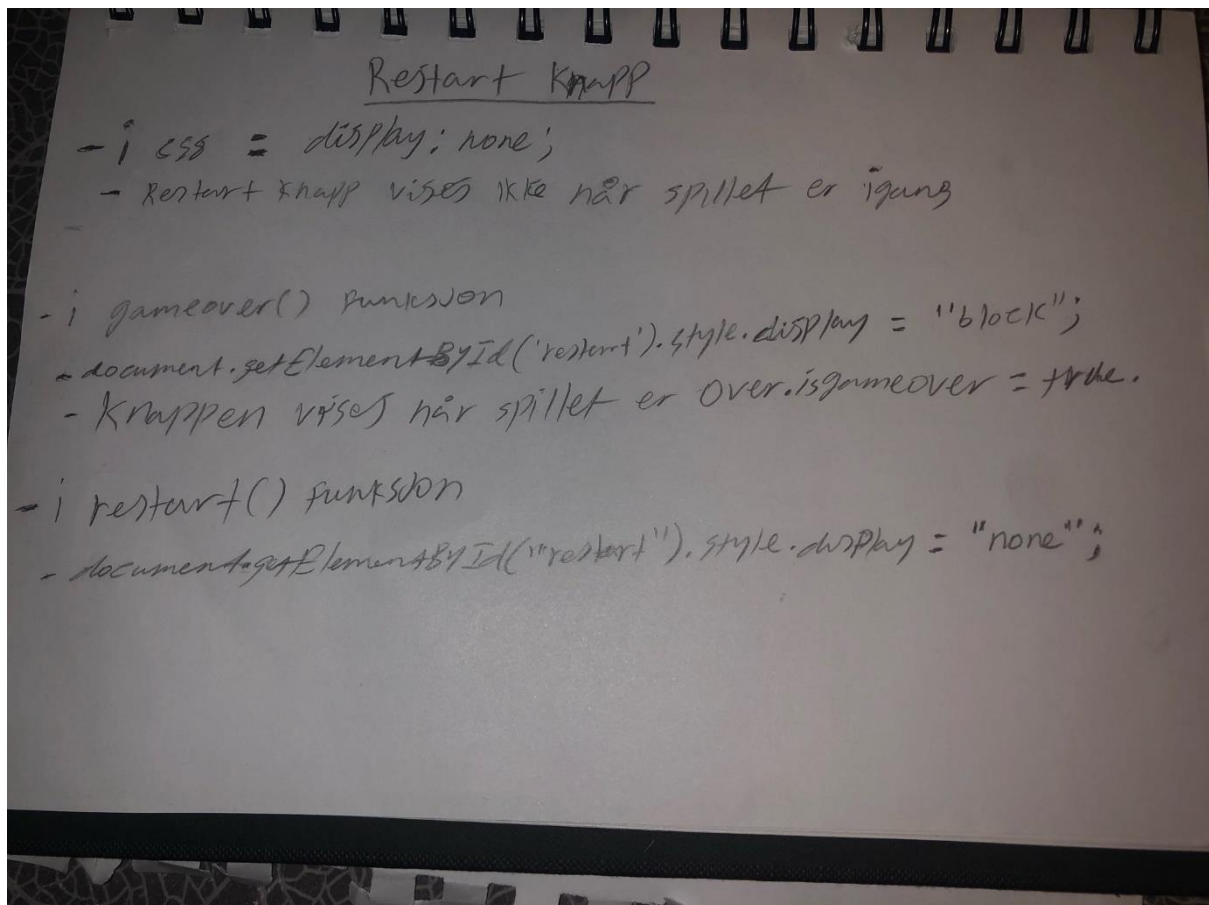
Jeg brukte boolean som er inspirert av ball, ball2 boolean variabler og lagde egen variabel for gameover satt til false med navn isgameover. Denne variabelen brukes i animate funksjon ved if statement der player1, player2 og update() funksjon kjører kun når isgameover variabel er false. Når isgameover er true, så tegnes ikke spiller elementer og update funksjon kjører ikke. Isgameover blir true når en spiller har mistet alle 3 liv.

Hver spiller har egen mini game over kjekk. Player1 har gameover1 boolean som kjører når player1 vinner. Player2 har gameover2 boolean variabel som kjører når player2 vinner. gameover1 kjører og blir true når player2 har 0 liv igjen, gameover2 kjører og blir true når player1 har 0 liv igjen. In animate funksjon, er det 2 if der en som kjekker om gameover1 er true og den andre som kjekker at

gameover2 er true. I hver av 2 if er det fillText path og gameover() funksjon kjører. If kjører hvis kondisjon er sant.

Når gameover() funksjon kjører, så blir isgameover variabel true og da blir alle 2d elementer på canvas fjernet siden player1, player2 og update funksjon kjører kun når isgameover er false.

Restart knapp:



- Bilde 9: restart knapp kode struktur. Dette er samme opplegg som deleksamen2 og deleksamen3.

I CSS har jeg gitt knappen, a-tag med id "restart" display none. Så knappen vises når spillet starter først gang.

I gameover() funksjon blir restart knappen hentet med document.getElementById og får display block. Knappen vises når det er gameover sånn at vi har mulighet til å klikke på knappen og starte spillet på nytt. I restart() funksjon blir knappen hentet igjen document.getElementById og får display

none. Knappen forsvinner igjen når spillet er restartet, men den dukke opp igjen når en spiller har vunnet og det blir gameover igjen.