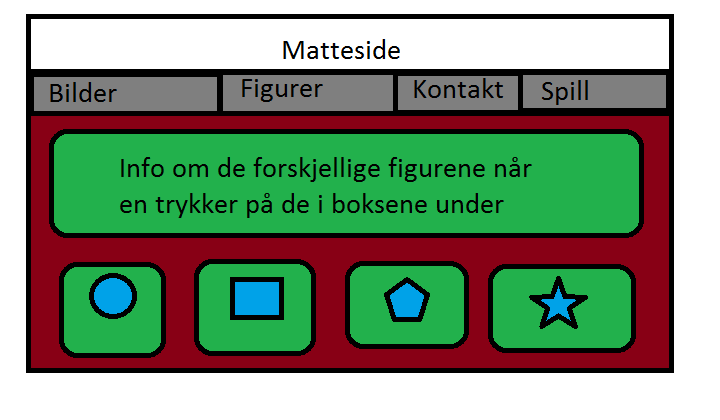
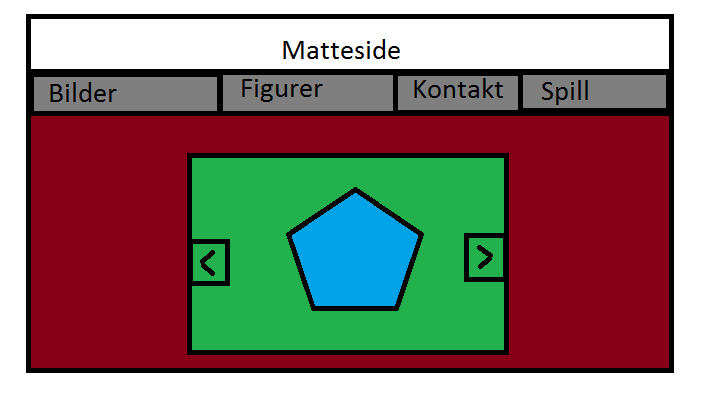
Informasjonsteknologi innlevering mattematikkside

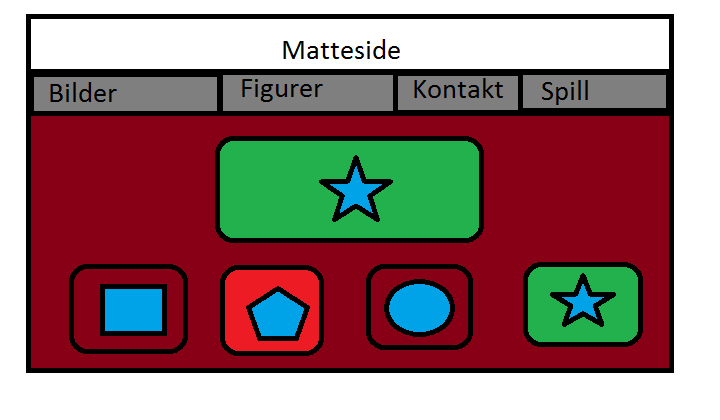
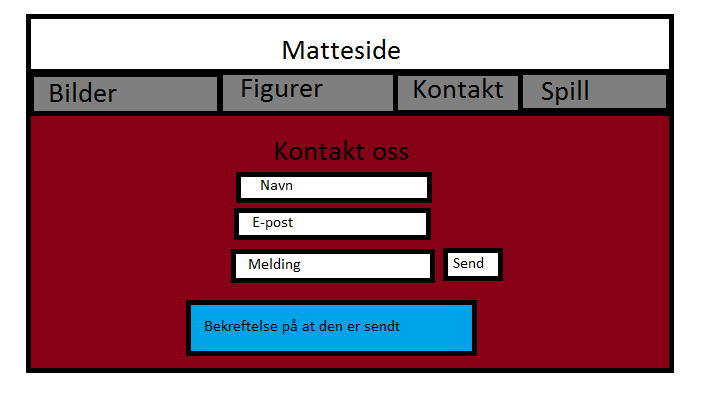
# Wireframe:

Dette er hvordan jeg ser for meg at nettsiden skal være



Figur 2 På siden figurer skal man få opp en liste med alle figurene. Når en trykker på en av dem skal det komme opp info om dem i den første tekstboken (div)

Figur 1 Det første brukeren møter når en går inn på nettsiden. Her skal det være et slideshow med forskjellige bilder av geometriske figurer



Figur 4 Her er figurspillet. Først må man legge inn navn for å starte spillet. Når man er ferdig kommer det opp en liste med poengsummer over andre

Figur 3 Inne på siden kontakter skal det være mulighet for å kontakte eierne av siden. Man MÅ legge inn navn og e-post for å sende en melding. Når man trykker send får man en bekreftelse

# Use case for siden figurspill:

1. Starter opp nettsiden figurspill.html
2. Får opp input der bruker skal skrive inn navn for å spille
3. Når brukeren trykker på start fjerner programmet inputelementet og knappen og starter opp spillet
4. Brukeren får opp en stor tilfeldig figur og fire mindre figurer under (se figur4, bilde).
5. Brukeren skal da trykke på den lille figuren som er lik den store
6. Trykker brukeren feil blir bakgrunnen rød og den rister, trykker brukeren riktig blir bakgrunnen grønn. Når brukeren har riktig kommer det også inn nye figurer
7. Dette kan brukeren fortsette med helt til tiden på 10 sekunder har gått ut. For hver riktige får man 10 nye sekunder, tar brukeren feil får man to mindre sekunder.
8. Sår spilleren er ferdig kommer det opp en liste over alle de som har spilt spillet før med deres poengsum og feil.

# Pseudokode figurspill

1. Når trykker på knappen start sendes navn og funksjonene start() og nedtelling() starter.
2. Funksjonen start():
   1. Funksjonen start() velger ut fire forskjellige figurer fra liste og putter de inn i listen utTrekteFigurer
   2. Så velger den tilfeldig en av figurene og setter den som riktig (lagrer den som tilfeldig)
   3. Blander rekken for at alt skal være tilfeldig
   4. Skriver ut det valgte bildet i diven sporsmal
   5. Skriver ut alle figurene i utTrekteFIgurer i diven svar
   6. Når bruker trykker på figur kjører funksjonen sjekkSvar() med iden til bildet som parameter
   7. Hvis iden er det samme som variabelen tilfeldig (den riktige figuren) setter man bakgrunnen til diven den er i til grønn og spiller av lyd. Er de ikke like setter den bakgrunnen til rød og rister
3. Funksjonen nedtelling():
   1. For hver gang nedtelling kjører skal man ha 1. sekund mindre
   2. Er tiden mere enn 0 skal programmet skrive tiden i tidInfo
   3. Er tiden ikke mere enn 0 skal man ikke starte på nytt og innholdet i divene skal tømmes for at man ikke skal kunne fortsette. Så skal programmet kjøre funksjonene lagrePoeng () med navn, antall poeng og feil og hentePoeng();
4. Funksjonen lagrePoeng()
   1. Startet med å sjekke om det er lagret noe som heter poengsom i nettleseren fra før av.
      1. Hvis ja, skal en hente rekken (array) med navn og poeng og gjøre den om til javascript for så å legge til enda et objekt med navn, poeng og feil.
      2. Hvis nei, så skal den legge til et objekt med navn, poeng og feil.
   2. Så gjøre om rekken med objekter til JSON og lagre den
5. Funksjonen hentePoeng()
   1. Henter rekken fra nettleser
   2. Gjør den om til Javascript
   3. Sorterer den med funksjonen sorter()
   4. Skriver ut alle rekordene
   5. Skrier ut en knapp med teksten «Spill igjen med samme navn»
   6. Når bruker trykker på denne kjører funksjonen startIgjen() som starter spillet igjen, med samme navn
6. Funksjonen sorter()
   1. Denne funksjonen returnerer en sortert rekke basert på poengsummen

# Flytdiagram:

