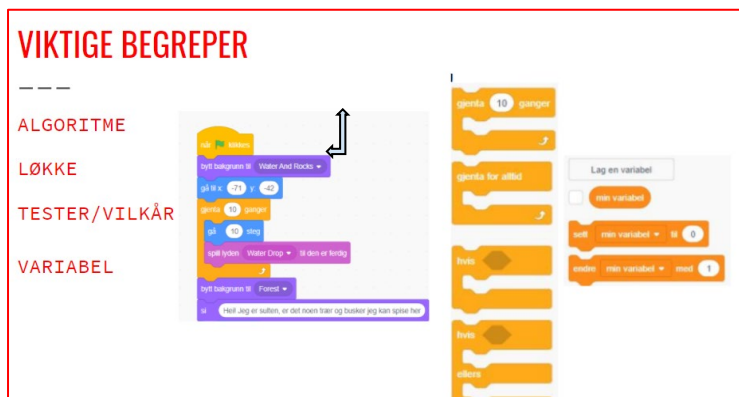


Forslag til undervisningsplan for lærer

Del 1: Animasjon

1. Oppstart: Tankekart for aktivering av forkunnskaper: hva er programmering – hva brukes det til i dagens samfunn?
2. Introduksjon av Scratch (for lærerne: for tips se introduksjonsvideoen «Scratch med Ingrid og Anna»).
3. Elevene lager sine egne animasjoner.
4. Når elevene har arbeidet en stund med egne animasjoner, kan man gå igjennom viktige begreper (for tips se del 2 av introduksjonsvideo) ved å forklare begrepene ved å bruke eksempler fra elevens arbeid eller introduksjon.



Kode laget i animasjon – brukt til å forklare

5. Elevene presenterer animasjonen og koden de har laget for en læringspartner og gir hverandre tips til videre arbeid
6. Elevene arbeider videre med animasjonen sin.

Eksempel på oppdrag for del 1

(Elevene har tidligere arbeidet med økologi og hvordan dyr og planter er tilpasset omgivelsene de lever i).

Oppdrag 1

Lag en animasjon om dyret som dere skrev om i naturfag:

1. Animasjonen skal inneholder minst to ulike figurer
2. Animasjonen skal inneholder minst to ulike bakgrunner
3. Figurene i animasjonen skal bevege seg
4. Figurene i animasjonen skal snakke til hverandre

SKAPE – TESTE – FORBEDRE :)

Del 2: Spill


1. Oppstart: kort gjennomgang av hvordan man kan styre figurene og hvordan man lager en variabel (se del 2 av instruksjonsvideo).
2. Elevene lager sine egne spill.
3. Når elevene har arbeidet en stund med egne spill, kan man gå igjennom noen flere begreper i programmering ved å bruke eksempler fra kodene man gjennomgikk i punkt 1 eller vise eksempler fra elevenes koder. Det fungerer godt at man lar elevene arbeide med sine spill og animasjoner og heller gir tips og råd etter hvert som elevene får mer kunnskaper, altså knytter teorien til elevenes praktiske erfaringer.
4. Forslag til underveisvurdering etter at elevene har arbeidet med sine spill i 1 til 2 timer:
Elevene presenterer i grupper på ca. 4 sine spill (og animasjoner) og koder for hverandre og lærer. De andre gruppene sitter og arbeider med sine egne spill, mens lærer snakker med én gruppe av gangen. På denne måten lærer elevene av hverandre og lærer får sett hvor elevene er i prosessen. Siden det i læreplanen er fokus på at algoritmer skal skapes, testet og forbedres, er det viktig å legge til rette for at elevene presenterer hvordan de har programmert (altså at man ser på kodene til elevene) og ikke bare hvordan spill og animasjon ser ut. Da kan man også diskutere fordeler og ulemper ved å kode på ulike måter.

Her kan man også legge inn forslag til utvidelser hvis man har elever som har kommet langt:

- man kan legge inn minus poeng i spillet
 - man kan legge inn GAME OVER ved bestemt poengsum
 - man kan legge inn ulike hastigheter på figurer eller objekter, slik at spillet blir vanskeligere
 - elevene kan komme med tips og innspill til hverandre
5. Eleven får forbedre sine spill.
 6. Eleven kan levere inn spill og animasjoner til vurdering.

Eksempel på hvordan oppdrag 2 kan se ut:

OPPDRA 2



LAG ET SPILL

- der du får **poeng**
- i programmeringen skal du lage en **algoritme**, som har med **løkker, tester og variabel**.

1. Alle skal lage en plan (pseudokode) for spillet sitt.
2. Vis fram plan til lærer
3. Deretter kan du starte å programmere

Feilsøking

Viktig for begge undervisningsoppleggene er å oppmuntre elevene til å prøve å finne feil selv. Ofte skyldes feil at programmet kun gjør det man sier det skal gjøre. Hvis man for eksempel ikke skriver i koden at poengene skal starte på null ved en ny runde i spillet, vil det ikke gjøre det.

Hjelp gjerne elevene i dette arbeidet ved å snakke om algoritmisk tenkning og at man kan vise noen koder, der elevene skal finne feilen.