○ Lærarrettleiing - Straffespark

PÅ BOKMÅL | TIL OPPGAVE | LAST NED PDF

Om oppgåva

I denne oppgåva skal elevane lage eit enkelt fotballspel, der dei skal prøve å score på så mange straffespark som mogleg. Denne oppgåva eignar seg godt som ein fyrste introduksjon til Scratch.

Oppgåva passar til:

Fag: Matematikk, programmering.

Anbefalte trinn: 3.-10. trinn.

Tema: Koordinatar, objektorientert programmering, løkker, variablar, testar.

Tidsbruk: Dobbelttime eller meir.

Kompetansemål

- ☐ **Matematikk, 4. trinn**: lese av, plassere og beskrive posisjonar i rutenett, på kart og i koordinatsystem, både med og utan digitale verktøy
- ☐ **Programmering, 10. trinn**: bruke grunnleggjande prinsipp i programmering, slik som løkker, testar, variablar, funksjonar og enkel brukarinteraksjon

Forslag til læringsmål

- ☐ Elevane kan bruke matematiske omgrep til å forklare figurane og ballen sin posisjon i koordinatsystemet (bakgrunnen).
- ☐ Elevane kan forklare korleis løkker, testar og variablar fungerer, og kvifor dei er omsynsmessige å bruke i denne oppgåva.

Forslag til vurderingskriterium

- ☐ Eleven syner middels måloppnåing ved å fullføre oppgåva.
- ☐ Eleven syner høg måloppnåing ved å vidareutvikle eigen kode basert på oppgåva, til dømes ved å gjere ein eller fleire av variasjonane under.
- ☐ Dette er ei oppgåve der elevane fint kan prøve kvarandre sine spel og vurdere kvarandre.

Føresetnader og utstyr

- [ ] **Føresetnader**: Ingen, fin introduksjon til Scratch.
- [ ] **Utstyr**: Datamaskiner med Scratch installert. Eventuelt kan elevane bruke Scratch i nettlesaren viss dei har ein brukar (eller registrerer seg) på scratch.mit.edu/. Elevane kan gjerne jobbe to og to saman.

Framgangsmåte

Her finn du tips, erfaringar og utfordringar til dei ulike stega i oppgåva. Klikk her for å sjå oppgåveteksten.

Generelt

- [ ] I denne oppgåva må elevane halde styr på tre figurar i tillegg til scena, og passe på at kvart skript blir koda på riktig stad. Ved nøye med at skripta ligg på riktig figur, som beskrive i oppgåva.

Steg 2: Me sparkar ballen

- [ ] Katten skyt ballen før ein klikkar på den, eller den må gå fleire steg før den når fram til ballen. Viss det skjer bør ein flytte på kor `Leo` og `Ball` blir plassert ved å endre på `gå til x: () y: ()`-klossane. Viss problemet er at katten må gå fleire steg kan ein òg endre på kor langt `Leo går` når han blir klikka på. Viss elevane allereie kan litt om koordinatsystemet er det ei fin øving å tenke på kva koordinatar ein må endre for å flytte figurane. Alternativt kan ein flytte på figurane ved å klikke og drag, og så sjå koordinatane øvst til høgre i skriptvindauget.

Steg 4: Målmannen reddar!

- [ ] I dette steget jobbar me vidare med skriptet som vart skrive på `Ball` i steg 2. Pass på at elevane ikkje lagar to ulike skript. Viss dei gjer det vil effekten stort sett vere at ballen beveger seg fortare enn normalt, fordi begge skripta flyttar ballen.

Steg 5: Fyrstemann til 10!

- [ ] Her er det mange små skript som startar på meldingane `Mål` og `Redning`. Pass på at dei blir lagt på riktig figur. Det tyder at `Ball` og `Målmann` har skript med `stopp [andre skript i figuren v] :: control`, `Leo` har skript der han `seier` noko, og Scena har skript som tel `(Mål)` og `(Redningar)`.

  Viss elevane vill at `Ball` eller `Målmann` skal seie noko kan det vere utfordrande på grunn av `stopp`-klossen. Ei mogleg løysing er som følgjer:

  ```
  når eg får meldinga [Redning v]
  sei [Hurra, eg redda!]
  stopp [andre skript i figuren v] :: control
  ```

  Det er viktig å *ikkje* bruke `sei [Hallo!] i (2) sekund` sidan klossen vil gjere at `Ball` eller `Målmann` ikkje sluttar å bevege seg før etter 2 sekund. For at snakkebobla skal bli borte kan ein bruke ein `sei []`-kloss (utan tekst). Den kan bli lagt øvst i `når @greenFlag vert trykt på`- eller `når eg får meldinga [Nytt spark v]`-skriptet.
- [ ] For enkelheits skuld set me aldri retninga på `Ball` i dette prosjektet. Sidan ballen aldri endrar retning - den går alltid horisontalt frå venstre mot høgre - er det sjeldan eit problem. Men viss elevane har endra retning på `Ball` slik at den sprett på skrå over skjermen må retninga tilbakestillast. Det gjer ein enkklast ved å klikke på klossen `peik i retning (90 v)` (eller ved å leggje denne klossen øvst i `Ball` sitt hovudskript).

Variasjonar

Viss elevane allereie er komfortable med Scratch er dette prosjektet eit godt høve til å prate om korleis ein gir ulike figurar unik oppførsel ved å gi dei ulike skript.

Eit viktig konsept i Scratch er at ein koder ved å beskrive eigenskapane (utsjånad, posisjon, retning osb.) og oppførselen (skript) til figurar. På fagspråket kallar me det **objektorientert programmering** (meir presist er Scratch *prototypeorientert programmering*, men skilnaden er ikkje relevant her). Dette virkar så naturleg at elevane sjeldan bevisst tenker på det, og samstundes skapar det sjeldan problem.

Dette er eit introduksjonsprosjekt, og elevane blir leia ganske detaljert gjennom korleis spelet programmerast. Det er framleis rom for ein del kreativitet. Elevane kan gjerne oppfordrast til å

- [ ] velje eigne figurar og bakgrunnar. `Leo` må ikkje vere ein katt, og det har blitt scora mål med andre ting enn fotballar.
- [ ] eksperimentere med hastigheita til `Ball` og `Målmann`. Ved å endre på tala i `gå () steg`-klossane vil figurane flytte seg saktare eller raskare. Det er nyttig læring å teste effekten av slike endringar, og å observere korleis vanskegraden i spelet forandrar seg (sjå boksen **Endre farta** på slutten av steg 4).
- [ ] forandre på tekstane i snakkeboblane til `Leo` eller tekstane som visast når ein vinn eller tapar i spelet.
- [ ] leggje på passande lydeffektar. Det blir nemnt i oppgåva på slutten av steg 5, men om elevane har litt erfaring med Scratch frå før kan dei gjerne gjere det undervegs i programmeringa.
- [ ] eksperimentere med objektorientert programmering:

- [ ] Start eit nytt Scratch-prosjekt ved å klikke `Programmering` frå hovudsida eller `Ny` i `Fil`-menyen.
- [ ] Legg til ein ekstra figur - til dømes `Bat1` - slik at det er to figurar i prosjektet. Dra dei rundt på scena slik at figurane er i kvart sitt hjørne.
- [ ] Spør elevane korleis dei vil kode at katten beveger seg mot den andre figuren. Spesielt må du passe på at dei er bevisst kva figur som må programmerast (katten). Spør om det same kan bli programmert ved å leggje eit skript på den andre figuren (svaret er nei, fordi det er katten som beveger seg).

  ```
  for alltid
      peik mot [Bat1 v]
      gå (10) steg
  slutt
  ```

- [ ] Korleis kan me programmere at flaggermusa rømmer frå katten når katten tek (rører) den? Igjen, kva figur må me programmere? Kanskje begge? *Me må programmere flaggermusa sidan den rømmer (oppførsel).* Her treng me ikkje noko nytt program for katten så lenge den ikkje reagerer på at den rører flaggermusa (*ingen ny oppførsel å beskrive*).

  Det er mange måter å skrive kode for at flaggermusa rømmer. Det følgjande er eit døme (hugs at koden høyrer til flaggermusa):

  ```
  for alltid
      vent til <rører [Sprite1 v]>
      gli (0.2) sekund til x: (tilfeldig tal frå (-240) til (240)) y: (tilfeldig tal frå (-180) til (180))
  slutt
  ```

- [ ] Spør elevane om dei kan tenke seg ein annan måte (enn objektorientert) å programmere på. Altså ein der ein ikkje knyttar skripta til figurane.

  Eitt døme på ein anna type programmering er **imperativ programmering** der programma blir skrive som ein serie kommandoar utan at me skil mellom kva figur som får kommandoen. I eit slikt språk ville dei to skripta over bli skrive som *eitt* skript om lag som dette (alle desse klossane eksisterer ikkje i Scratch):

  ```
  for alltid
      flytt [katten v] mot [flaggermusa v] :: motion
      viss <[katten v] rører [flaggermusa v] :: sensing>
          flytt [flaggermusa v] til x: (tilfeldig tal frå (-240) til (240)) y: (tilfeldig tal frå (-180) til (180)) :: motion
      slutt
  slutt
  ```

  Du kan gjerne vise koden til elevane. I tillegg til at det berre er eitt skript, kva andre skilnader ser dei? *Den andre store skilnaden er at ein alltid må fortelje kva figur som skal utføre kommandoane. Det er underforstått i Scratch.*

Eksterne ressursar

- [ ] Førebels ingen eksterne ressursar...