Lærarrettleiing - Straffespark

Skrevet av: Stein Olav Romslo

Kurs: Scratch

Tema: Blokkbasert, Spill

Fag: Matematikk, Programmering

Klassetrinn: 1.-4. klasse, 5.-7. klasse, 8.-10. klasse

Om oppgåva

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





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

☐ Før	esetnader: Ingen, fin introduksjon til Scratch.
Scra	tyr: Datamaskiner med Scratch installert. Eventuelt kan elevane bruke atch i nettlesaren viss dei har ein brukar (eller registrerer seg) på atch.mit.edu/ (https://scratch.mit.edu/). Elevane kan gjerne jobbe to og to nan.

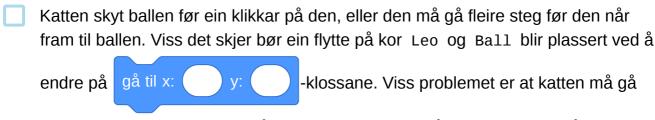
Framgangsmåte

Her finn du tips, erfaringar og utfordringar til dei ulike stega i oppgåva. Klikk her for å sjå oppgåveteksten. (../straffespark/straffespark_nn.html)

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



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!

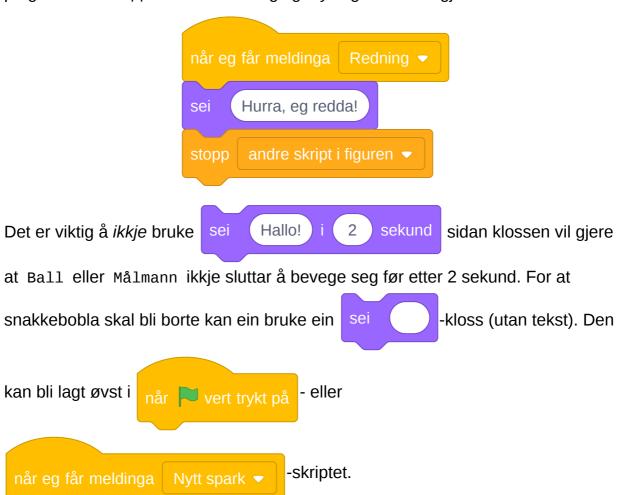
I dette steget jobbar me vidare med skriptet som vart skrive på Ball i steg 2.

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 , 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:



For enkelheits skuld set me aldri re	etninga 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 eleva	ane har endra retning på Ball slik at den sprett
på skrå over skjermen må retninga	a tilbakestillast. Det gjer ein enkklast ved å klikke
på klossen peik i retning 90 ▼	(eller ved å leggje denne klossen øvst i Ball
sitt hovudskript).	

Variasjonar

ein vinn eller tapar i spelet.

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

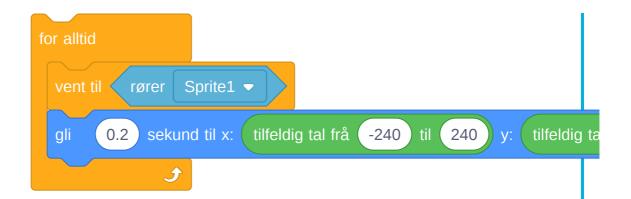
velje eigne figurar og bakgrunnar. Leo må ikkje vere ein katt, og det har blitt scora mål med andre ting enn fotballar.
eksperimentere med hastigheita til Ball og Målmann. Ved å endre på tala gå steg -klossane vil figurane flytte seg saktare eller raskare. Det er nyttig læring å teste effekten av slike 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

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 ▼ gå 10 steg	
	Korleis kan me programmere at flaggermusa rømmer frå katten når katten tek (rører) den? Igjen, kva figur må me programmere? Kanskje begge? <i>Me må programmere flaggermusa sidan den rømmer (oppførsel).</i> Her treng me ikkje noko nytt program for katten så lenge	

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

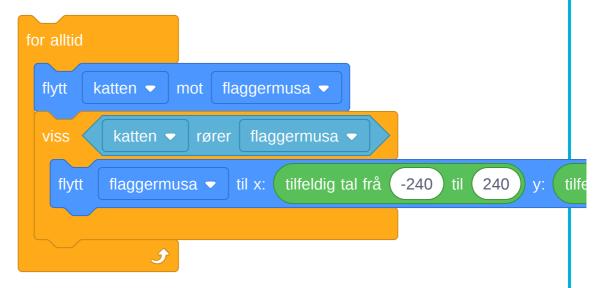
beskrive).

den ikkje reagerer på at den rører flaggermusa (ingen ny oppførsel å



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):



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...

Lisens: CC BY-SA 4.0 (http://creativecommons.org/licenses/by-sa/4.0/deed)