

Taraldsen, Myhra

Programmering med Spehroballer

Innledning

Fra skolestart i 2020 får skolen nye læreplaner som skal møte nye kompetansebehov i et stadig mer digitalisert samfunn. God forståelse for teknologiens muligheter og begrensninger er viktig for å kunne bruke og skape nye digitale verktøy (Nygård, 2018). Programmering og algoritmisk tenkning blir i fagfornyelsen en del av matematikkfaget, og vi mener derfor at målet bør være at elevene skal kunne bruke programmering som metode for å løse matematiske utfordringer. For at innføringen av programmering i matematikkfaget skal være på fagets premisser, må det utarbeides gode undervisningsopplegg som legger til rette for at elevene kan lære matematikk gjennom bruk av programmering.

I denne artikkelen presenterer vi et undervisningsopplegg der elever på 5. trinn får en innføring i programmering mens de jobber med temaet måling i matematikk. Undervisningsopplegget ble gjennomført av lærer Guro på hennes 5. trinn våren 2019.

Lene Hayden Taraldsen

Høgskulen på Vestlandet
lene.hayden.taraldsen@hvl.no

Kristin Sæterdahl Myhra

Høgskulen på Vestlandet
k.s.myhra@hvl.no

Programmering og algoritmisk tenkning

For å kommunisere med et dataprogram må tanker sorteres som en logisk rekke av operasjoner og formuleres på et språk som dataprogrammet forstår. Gjøvik og Torkildsen (2019) problematiserer bruken av ulike termer knyttet til begrepet *algoritmisk tenkning*. I denne artikkelen velger vi i likhet med Gjøvik og Torkildsen (2019, s. 32) å se på begrepet algoritmisk tenkning synonymt med det engelske begrepet «Computational Thinking», ut ifra Bocconi, Chiocciariello og Earp (2018, s. 7) sin definisjon: «Computational thinking is regarded as a thought process entailed in designing solutions that can be executed by a computer, a human, or a combination of both». Vi følger Nygård (2018) og definerer *programmering* som hele prosessen fra problemanalyse til planlegging av hvordan selve koden skal struktureres, men inkluderer også selve kodingen som en del av programmeringen. Koden forteller dataprogrammet hvilke oppgaver som skal løses ved hjelp av det programmeringsspråket som dataprogrammet forstår. Noen programmeringsspråk tilbyr ferdige blokker med kommandoer. Ved *blokkbasert programmering* kan man sette sammen instruksjoner ved hjelp av blokkene, det er ikke behov for egenskrevet tekst. Vi mener disse blokkene forenkler kodingen fordi man ikke trenger å lære det underliggende tekstbaserte språket slik man må når man arbeider med *tekstbasert program-*

mering. I dataprogrammer som baserer seg på tekst, må teksten skrives i sekvenser som blir satt sammen til instruksjoner. Dette gir ofte større frihet og flere muligheter, men kan oppleves mer krevende fordi man må kunne programmeringsspråkets semantikk og syntaks. I forslaget til ny læreplan er det ikke spesifisert om elevene skal arbeide med blokk- eller tekstprogrammering (Utdanningsdirektoratet, 2019).

Programmering ved hjelp av Spheroballer

Det finnes et stort utvalg av verktøy som kan brukes i programmeringsundervisning for barn og unge. For dette undervisningsopplegget har vi brukt et klassesett med Sphero BOLT-baller og iPader med gratis-appen Sphero Edu vist i figur 1. En Spheroball er en app-styrt, programmerbar robotball der elevene kan kode gjennom tegning, blokkbasert programmering eller tekstprogrammering (JavaScript). Ballene er utstyrt med sensorer som gir brukeren informasjon om ballens bevegelser på en enkel måte. Spheroballene kobles til appen via Bluetooth og har en rekkevidde på 30 meter. Vi valgte dette verktøyet fordi Spheroballen er en fysisk gjenstand som elever kanskje gjenkjenner fra sin egen lek med ball. I tillegg til det visuelle ytre gir arbeid med Spheroball også mulighet til å utvikle elevers kompetanse fra tegning til blokkprogrammering og videre med tekstprogrammering.



Figur 1: Sphere BOLT-ball og appen Sphero Edu på Ipad. Hentet fra <https://www.apple.com/no/shop/product/HMA12ZM/A/sphero-bolt-robot>

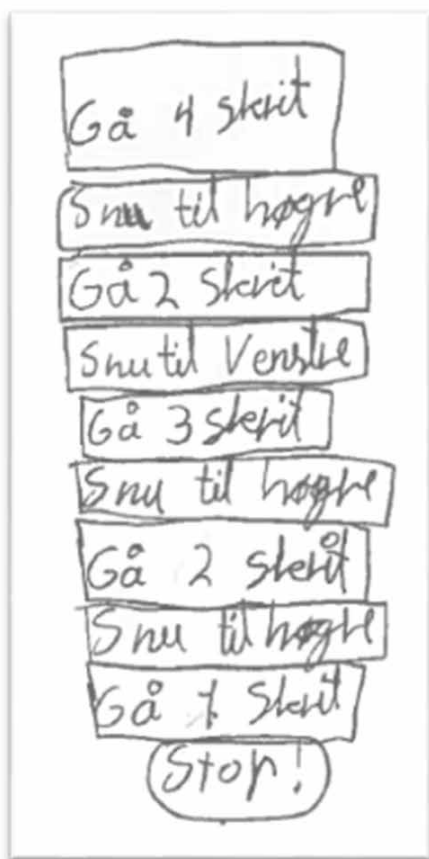
Beskrivelse av gjennomføring på 5. trinn

Vi kom i kontakt med lærer Guro som aldri har prøvd programmering verken privat eller i jobbsammenheng. På det første møtet med Guro presenterte vi et forslag til undervisningsopplegg og lot henne teste Spheroballene. Deretter tilpasset vi undervisningsopplegget (vedlegg 1) til 5. trinn i fellesskap. Vi ble enige om at Guro selv skulle undervise elevene, og at vi ble med som observatører.

Noen uker senere møtte vi lærer Guro på skolen der hun jobber, og brukte tiden før undervisningen til å snakke litt om hennes forventninger og hva hun tenkte om å skulle undervise i programmering for første gang. Hun fortalte blant annet at hun syntes det var vanskelig med programmering i begynnelsen, men at det ble lettere å forstå når hun fikk prøvd Spheroballene selv. Videre uttrykte hun sine tanker om elevenes første møte med programmering: «Om jeg får det til, ja, så klarer de det. De er jo mye flinkere enn meg. De tar dette veldig mye enklere enn noen av oss. Det gjør de! Så er det noen som ikke tar det like lett, men så slenger de seg på ... , og så får de det til, tror jeg.» Da skoleklokka ringte, fulgte vi med Guro til klasserommet.

1. økt

Fokus for økta var å introdusere elevene for begrepene algoritmisk tenkning og programmering. Guro startet med å spørre elevene om de visste hva programmering var. Mens noen elever nikket og kom med innspill, var andre stille. Guro og elevene pratet sammen for å finne en felles forståelse av begrepet. Mens *programmering* ble introdusert gjennom samtaler, fikk elevene møte *algoritmisk tenkning* gjennom en praktisk oppgave. Oppgaven var å skrive ned en instruks, instruksene skulle beskrive en rute gjennom klasserommet. Ruten skulle unngå pulter og andre fysiske hindringer, og elevene skrev sine ferdige instruksjoner ned med blyant og papir. En medelev skulle deretter følge den detaljerte instruksjonen for å finne den riktige



Figur 2: En veibeskrivelse gjennom klasserommet.

veien mellom pultene i klasserommet. I etterkant ble oppgaven oppsummert felles i klassen.

I elevinstruksen i figur 2 kan vi se at eleven har presentert instruksen som innrammede blokker med delinstrukser. Eleven har brukt verbene «gå» og «snu» i kombinasjon med lengdebeskrivelsen «skritt» og retningsbeskrivelsene «høyre» og «venstre». Til slutt avsluttet elevene med «stopp». Vi la merke til at elevene brukte hverdagspråk i sine beskrivelser, noe som kan være utfordrende for en mottaker som krever mer presise formuleringer.

2. økt

Det har gått et døgn siden første økt. De 22 elevene på 5. trinn var tydelig spente på å få prøve ut Spheroballene. Guro hadde valgt et klasserom

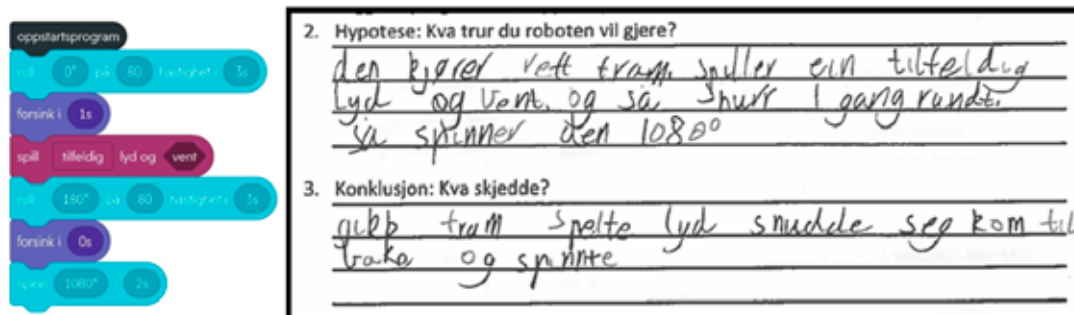
med stor gulvplass, og elevene satt fint på rekke mens Guro introduserte økta og forklarte hvilke regler som gjaldt, se figur 3. Elevene ble delt inn i par, og med litt hjelp fikk alle koblet sammen riktig Spheroball og iPad.



Figur 3: Guro introduserer økta og presenterer reglene.

Den første øvelsen var en *testoppgave* hvor elevene skulle utforske en ferdigskrevet blokk-programmert kode med instruksjoner til hvordan Spheroballen skulle bevege seg, se figur 4. Hensikten med oppgaven var å la elevene bli kjent med noen av de ferdigskrevne blokkene de kom til å møte i appen Sphero Edu, og se hvordan hver blokk representerer et steg i instruksjonen. Elevene fikk først skrive ned hva de trodde roboten ville gjøre dersom den fikk instruksjonen. Deretter kodet elevene den ferdige instruksjonen i appen og ga instruksjonen til Spheroballen for å undersøke hvordan den reagerte på instruksjonen.

I elevsvaret vist i figur 4 formulerte elevene at roboten skulle bevege seg fremover, spille en tilfeldig lyd, snurre rundt en gang, vente og deretter gå inn i en spinnbevegelse. Vi la merke til at elevene formulerte vendingen på 180° som at robotballen ville snurre en gang rundt og lot være å beskrive Spheroballens rettlinjede bevegelse mellom lydavspillingen og forsinkelsen. Da elevene selv prøvde ut ballen, oppdaget de at 180° tilsvarte at Spheroballen endret retning, og at den deretter beveget seg tilbake til utgangspunktet. Fra denne elevbesvarelsen leser vi at elevene enten mangler forståelse for hvor stor rotasjon 180° er, eller at de ikke formulerer seg presist. Begge er relevante innslag som kan skape gode samtaler i klasserommet.



Figur 4: Til venstre: ferdigskrevet kode elevene fikk utdelt i testoppgaven. Til høyre: et elevsvar.

I oppgave 1 skulle elevene bli kjent med forholdet mellom (og enheter for) vei, fart og tid i tilknytning til Spheroballen. De fikk også i oppgave å sammenlikne sensorens målinger med egne fysiske målinger av distansen ballen beveget seg. Målet var å få ballen til å bevege seg nøyaktig 1 meter. For å få ballen til å rulle brukte elevene bevegelsesblokken *rull*, som de kjente fra testoppgaven. I denne blokken implementeres bevegelse som en hastighetsvektor som virker i en angitt tidsperiode. Det er viktig at læreren er oppmerksom på at formuleringen i bevegelsesblokken ikke er presis i bruken av begrepene *hastighet* og *fart*. Det er også viktig å poengtere at fart oppgis uten enhet, mens enhetene til både retning [$^{\circ}$] og tid [s] er gitt.

For å løse oppgaven prøvde elevene seg frem ved å endre på parameterne tid [s] og fart i bevegelsesblokken. Selv om bevegelsesblokken ikke kan ta inn informasjon om avstand, kan avstanden Spheroballen har beveget seg, leses av i appen. Elevene kunne derfor bruke både måleteip og sensordata i appen til å avgjøre hvilke parameterverdier som fikk ballen til å rulle nøyaktig 1 meter (figur 5). Elevene var ivrige og diskuterte temaet måling, for hvor langt er egentlig 1 meter? En diskusjon mellom to elever startet da de hadde limt måleteip på gulvet. Måleteipen markerer cm fra 0 til 100, og denne skalaen repeteres i teiprullen elevene fikk utdelt. Utfordringen i dette tilfellet var at



Figur 5: Konsentrert arbeid med målingsoppgaven

teiprullen ikke startet på null, men på 60 cm. Elevene diskuterte om de kunne starte på 60 cm, og hvor skulle de i så fall stoppe? Elevene oppdaget også at måledata oppgitt i appen skilte seg fra målinger gjort fysisk i klasserommet, og diskuterte ivrig hva som kunne være årsaken til avviket mellom målinger gjort med de ulike måle metodene. Til slutt fikk de i oppgave å beregne Spheroballens fart i enhetene cm/s og km/t ved hjelp av avstandsmålinger i appen og tiden elevene hadde implementert i bevegelsesblokken.



Oppgave 3: Labyrint

Me skal ha en liten konkurranse med Spheroballene.

Oppgava går ut på å få ballen til å gå gjennom ei gitt løype på kortast mogleg tid.

Løypen vil se slik ut:

Lag eit program slik at ballen kan komme seg gjennom labyrinten.

Det er ikkje lov å sitte ved løypa å prøve seg fram.

Lag koden → test → gå tilbake for å gjere justeringar.

- Kvifor vart det slik?
- Vart det sånn me hadde tenkt?
- Har me vore presise nok?
- Kva har me gløymt å kommunisera?

Figur 6: Til venstre: elevene gjør seg klare til å sende Spheroballen gjennom labyrinten. Til høyre: oppgaveteksten.

I oppgave 2 skulle elevene programmere Spheroballen til å tegne et kvadrat med 1 meter lange sider. Her ble elevene utfordret på kjennskap til kvadratets egenskaper for å programmere, i tillegg til at de måtte benytte informasjonen om vei, fart og tid som de fant i oppgave 1, for å få riktig størrelse på kvadratet. Guro pratet med elevene underveis, og det var tydelig at de klarte å se for seg hvordan Spheroballen skulle bevege seg, men likevel fikk mange av gruppene problemer med å få ballen til å bevege seg slik de har tenkt. En av utfordring elevene fikk, var at Spheroballen fortsatte rett frem i fire lengder i stedet for å rotere 90° i hjørnene. Guro så da behovet for å snakke med elevene om hva de ulike blokkene faktisk betydde, slik at elevene fikk formulert sine tanker om bevegelse riktig. Elevene jobbet i noe ulikt tempo, og selv om noen kunne hatt behov for mer tid på oppgave 2, ønsket Guro at alle skulle begynne på den siste utfordringen, oppgave 3, på samme tid.

I oppgave 3 var utfordringen å få Spheroballen til å rulle gjennom en fastlagt labyrint på gulvet. Oppgaven ble fremstilt som en konkurranse. Labyrinten var skissert opp på oppgavearket med nøyaktige mål på løypa, se figur 6. For å fokusere på oppgaven la elev-

ene Spheroballen til siden mens de formulerte koden i appen. Da elevene var ferdige med koden sin, fikk de Spheroballen tilbake for å gjennomføre et testforsøk på gulvet, og de fleste fant ut at de måtte gjøre justeringer før de var klare for konkurranse. Det var tydelig spenning i rommet, og elevene jobbet intenst med oppgaven. Guro markerte labyrinten med teip på gulvet i klasserommet, og elevene stilte seg opp, se figur 6. Alle parene fikk sende Spheroballen gjennom labyrinten, og mange oppdaget fort at de ikke hadde vært nøyaktige nok. Noen elever hadde ikke tatt seg tid til å lese av målene fra skissen, mens andre hadde slurvet med beregningene sine. Guro valgte å ikke si så mye til elevene underveis, men veiledet dem med kommentarer som «Hva kan dere endre?» og «Hva tenker dere nå?». Da den første gruppen mestret utfordringen og fikk ballen til å smyge seg igjennom labyrinten, ble det full jubel i klasserommet. Elevene jobbet iherdig med kodene sine, og mens noen var opptatt av å perfektionere koden, hadde andre fullt fokus på å få ballen gjennom labyrinten.

Tilbake på lærerrommet var Guro fornøyd: «Alt fungerte som det skulle! Opplegget fungerte godt, elevene oppførte seg bra, og det var ingen

tekniske problemer.» Guro fortalte at hun hadde vært litt spent på hvor nøyaktige elevene kom til å være: «... de måtte være fokuserte og nøyaktige. Det har de veldig godt av å lære. Jeg ser i matematikkboka at det er mange som slurver og ikke er nøyaktige, ikke er presise. Men her er det tydelig at når de slurver, så gjør ikke robotballen det de vil at den skal gjøre.»

Programmering i fremtidens klasserom

Når programmering og algoritmisk tenkning kommer inn i matematikkfaget fra høsten 2020, er dette nytt for mange lærere. Guro er positiv til utviklingen av matematikkfaget, men er usikker på sin egen kompetanse når det gjelder programmering: «Nei, jeg kan jo ingenting! Utenom det som jeg har tilegnet meg nå da, men det gikk jo bra, selv om jeg i utgangspunktet ikke kunne noe. Jeg er veldig glad for at jeg turte å bare hoppe i det!»

For 5.-klassingene som deltok i undervisningen, var det i likhet med Guro første møte med programmering og algoritmisk tenkning. Hovedmålet med undervisningsopplegget var å la elevene opparbeide forståelse for begrepene algoritmisk tenkning og programmering gjennom lek og utforskning, som samsvarer til foreslått kompetansemål etter 2. trinn i matematikk, som er å *lage og følge regler og trinnvise instruksjoner i lek og spel*. Gjøvik og Torkildsen (2019) påpeker at programmering ikke nødvendigvis trenger å foregå på en datamaskin, noe vi erfarte i første økt med elevene. Videre ønsket vi at elevene skulle få en forståelse av programmering som en metode ved å skrive algoritmer for å få Spheroballen til å følge ønsket instruks. Dette er i tråd med det foreslåtte kompetansemålet etter 5. trinn i matematikk, som sier at elevene skal *lage algoritmar med bruk av variablar, vilkår og lykkjer og programmere desse* (Utdanningsdirektoratet, 2019).

For at programmering skal bidra til en positiv utvikling av matematikkfaget, må det utarbeides gode undervisningsopplegg som lar elevene

bruke programmering som metode for å løse matematiske utfordringer. Under gjennomføringen av undervisningsopplegget observerte vi at elevene møtte utfordringer som ledet frem til gode matematikkfaglige samtaler om måling og bevegelse. Det var tydelig av Spheroballen som verktøy bidro til at programmering ble oppfattet som mindre abstrakt, og at elevene engasjerte seg i matematikken de behøvde for å gjennomføre oppgavene. Et eksempel verdt å fremheve var noen elevers utfordring med manglende nøyaktighet. Det var tydelig at elevene virkelig ønsket å finne ut hvilke regnefeil de hadde gjort, slik at de kunne lage et program som fikk Spheroballen gjennom labyrinten i oppgave 3. Videre mener vi også at programmering i seg selv kan være en kilde til motivasjon for arbeid med matematikk. Fra undervisningsøkta så vi at elevene jobbet konsentrert, og da Spheroballen endelig krysset mållinjen i oppgave 3, var det full jubel.

Referanser

- Apple (2019). *Sphero BOLT Robot*. Hentet 07.10.19 fra: <https://www.apple.com/no/shop/product/HMA12Z-M/A/sphero-bolt-robot>
- Bocconi, S., Chiocciariello, A. & Earp, J. (2018). *The Nordic approach to introducing computational thinking and programming in compulsory education. Report prepared for the Nordic@BETT2018 Steering Group*. Hentet fra <https://www.itd.cnr.it/doc/CompuThinkNordic.pdf>
- Gjøvik, Ø. & Torkildsen, H. A. (2019). Algoritmisk tenkning. *Tangenten – tidsskrift for matematikkundervisning*, 30(3), 31–37.
- Nygård, K. (2018). *Programmering i skolen. Hvordan komme i gang?* Oslo: Pedlex
- Utdanningsdirektoratet (2019a). *Algoritmisk tenkning*. Hentet 16.09.19 fra: <https://www.udir.no/kvalitet-og-kompetanse/profesjonsfaglig-digital-kompetanse/algoritmisk-tenkning/>
- Utdanningsdirektoratet (2019b). *Læreplan i matematikk fellesfag 1.–10. trinn*. Hentet 20.09.19 fra: <https://hoering.udir.no/Hoering/v2/343?notatId=686>