

Matematikklærerstudenters erfaringer med spillprogrammering i PICO-8

Kjetil Liestøl Nielsen

Institutt for Matematikk og Naturfag, Universitetet i Sørøst-Norge

Artikkelen beskriver en pilotutprøving av spillprogrammering med PICO-8 for studenter i matematikk på grunnskolelærerutdanning 5-10 (GLU 5-10). Målet med utprøvingen var å se hvordan studentene opplevde arbeid med spillutvikling med tekstbasert programmering. Et stort flertall av studentene opplevde prosjektet som morsomt, spennende, lærerikt, nyttig og givende, samtidig som at det også kunne være frustrerende, vanskelig og tidkrevende. Noen studenter rapporterte også at de tok prosjektet med seg inn i praksis for å engasjere sine elever. Artikkelen diskuterer disse resultatene i lys av videre utprøvinger på lærerutdanning og muligheter for bruk av spillprogrammering i matematikkemnet i grunnskolen.

Introduksjon

I 2020 ble det innført ny læreplan i grunnskolen hvor programmering har en stor rolle. UDIR argumenterer for innføringen av programmering ved at det kan knyttes til "21. century skills" og algoritmisk tenking, og for at elever skal få en økt forståelse for hvordan vårt digitale samfunn fungerer (Sevik 2016). I matematikk har programmering i skolesammenheng blant annet tidligere vist å kunne bedre læringsutbytte i geometri og økt motivasjon til å lære matematikk (Forström & Kaufmann, 2018). Elevene i norsk skole begynner tidlig med programmering i LK20 og skal allerede etter andre trinn kunne tenke algoritmisk i form av regler og stegvise instruksjoner, og etter fjerde trinn kunne lage algoritmer og bruke programmeringskonsepter som variabler, vilkår og løkker. Selv om dagens elever har relativt lite erfaring med programmering, vil en ungdomsskolelærer om noen år møte elever som har programmert og tenkt algoritmisk siden småtrinnet. Dette kommer til å sette store krav til læreres kompetanse i programmering og programmeringsdidaktikk i årene som kommer, og det er studier som viser at det er et behov for mer opplæring av norske lærere i programmering (Kravik et al., 2022; Andersen, 2022). Dette er ikke en unik utfordring for Norge da flere internasjonale studier viser også behov for mer opplæring av lærere i programmering (Hijon-Neira et al., 2017; Corradini et al., 2017; Misfeldt et al., 2019; Attard & Busitill, 2020).

Selv om programmering generelt har vist å kunne føre til økt motivasjon hos elever (Forström & Kaufmann, 2018), kan også mangel på motivasjon og interesse hos elever være en utfordring når programmering skal innføres i undervisningen (Andersen, 2022). Programmering kan for noen oppleves som vanskelig (Bennedsen & Caspersen, 2008) og det er også ikke en automatikk i at fokus på programmering bidrar til bedre læring og motivasjon i matematikkfaget (Forström & Kaufmann, 2018). Forström og Kaufmann (2018) argumenterer derfor at det trengs mer forskning på ulike måter å implementere programmering.

Et område i bruk av programmering som har hatt lite fokus i norsk sammenheng, er læring av programmering og fag gjennom utvikling av små dataspill. Spillutvikling har vist å kunne brukes til å lære programmeringskonsepter (Doerschuk et al., 2013), fremme kreativ tenking (Bulut et al., 2022) og fremme ferdigheter i problemløsning (Dekhane et al., 2013). Det har også vist å kunne øke elevens motivasjon og holdninger til matematikk og hjelpe elever aktivere deres refleksjon rundt hverdagslige matematiske erfaringer (Ke, 2014). Bruk av tilfeldigheter er ofte vanlig i spill, og en studie har også vist at utvikling av spill kunne gi elever en økt forståelse for sannsynlighet (Akpınar & Asland, 2015). Spillutvikling åpner også mange muligheter for tverrfaglig arbeid (Masuch & Nacke, 2004), noe som gjør det spesielt relevant for LK20 med sitt fokus på tverrfaglighet og tverrfaglige temaer.

Å innføre spillprogrammering og spillutvikling i undervisning krever derimot en dypere kompetanse hos lærere enn generell kompetanse i programmering, slik som kompetanse i spilllogikk og spillrelaterte programmeringskonsepter (f.eks. kollisjons- og bevegelseskode). Innenfor spillprogrammering er det i tillegg flere begreper som kan misforstås av elever (Johnson, 2017) og det kan være utfordrende å veilede elevene til å tenke enkelt nok og holde omfanget lavt i spill de ønsker å utvikle (Masuch & Nacke, 2004). De potensielle fordelene med bruk av spillprogrammering i grunnskolen, åpner derimot for spørsmål hvorvidt spillprogrammering og spillutvikling burde være en del av lærerutdanningen, spesielt for kommende matematikklærere da programmering er en stor del av læreplanen i matematikk.

I denne artikkelen beskrives en pilot-utprøving av spillprogrammering på grunnskolelærerutdanningen med tekstbasert programmering i PICO-8, en fantasikonsoll designet som en nybegynnervennlig måte å lære programmering og spillutvikling. Målet med utprøvingen er å undersøke hvordan lærerstudenter i matematikk opplever spillprogrammering, hvorvidt de får dette til, og hvorvidt de mener dette er passende for elever i grunnskolen. Erfaringer fra en slik utprøving kan brukes til å vurdere muligheter for å implementere slike prosjekter i både grunnskolelærerutdanningen og grunnskolen. Selv om denne studien fokuserer på lærerstudenter i høyere utdanning, vil refleksjonene og diskusjonene rundt implementering av spillprogrammering i matematikkfaget også være relevant for lærere i grunnskole og videregående som har interesse for bruk av programmering i skolen. Mot slutten av artikkelen vil muligheter for implementering i skole bli diskutert.

Metode

Bakgrunn

Pilotprosjektet ble utført høsten 2022 i et matematikkemne studenter som går Grunnskolelærerutdanning 5-10 tar i sitt tredje semester ved campus Notodden på Universitetet i Sørøst-Norge (USN). Valg av riktig verktøy er viktig i et slikt prosjekt og det er en fordel å velge verktøy som er enkle å bruke for å skape en lav inngangsterskel (Ke, 2014). Samtidig argumenterer DiSessa (1997) for at programmeringsverktøy man bruker i utdanning må være såpass fleksible at de ikke holder igjen faglig sterke elever/studentene. Studentene hadde litt erfaring med programmering i Python fra tidligere emner, og et logisk valg kunne dermed vært å bruke Python som programmeringsspråk i prosjektet. Python har flere bibliotek som kan brukes til spillprogrammering, f.eks. PyGame. Dette hadde derimot ført til at studentene måtte bruke ekstra programvare til f.eks. lagring av grafikk i tillegg til at studentene måtte organisert

alle filer i prosjektet selv. Det ble vurdert at dette ville skape en høy inngangsterskel til prosjektet for studentene og at deres tidligere erfaring med programmering i Python, som var begrenset, ikke ville gjøre opp for den høye inngangsterskelen. Siden hovedfokuset i utprøvingen var på selve spilldesignet og programmeringsdelen, ble dette alternativet valgt bort.

For å holde en lav inngangsterskel, kunne et naturlig valg vært å bruke et blokkprogrammeringsspråk som f.eks. Scratch (Mladenović et al., 2016). Dette er et språk som har blitt brukt i tidligere studier hvor elever har laget små spill (se f.eks. Mladenović et al., 2016; Akpinar & Aslan, 2015). Blokkprogrammering har derimot også ulemper. I en studie av Attard og Busutill (2020) følte lærerne i studiet at det var enklere å gjøre endringer med tekstbaserte språk og i matematikk var det også enklere å skrive matematiske formler når man brukte tekstbasert programmering. I en studie av Weintropp og Wilensky (2015) følte elevene i studien at selv om blokkprogrammering var lettere å komme i gang med, følt det også mindre autentisk ut og ikke ut som "ekte" programmering. Følelsen av å jobbe med ekte programmering kom først når elevene jobbet med tekstbasert programmering. Selv om det var lettere å komme i gang med blokkbasert programmering, følte elevene samtidig at det tok lengre tid å lage programmer og at blokkprogrammering fort ble mer forvirrende og uoversiktlig når kompleksiteten av programmet økte (Weintropp & Wilensky, 2015). Tekstbasert programmering ble derfor ansett som det mest egnede for dette prosjektet, både for grunnene nevnt over og fordi det er viktig at lærerstudenter i matematikk på lærerstudenter på 5-10 har en solid forståelse for tekstbasert programmering. For å holde en lav inngangsterskel samtidig som at verktøyet fokuserer på tekstbasert programmering og har nok fleksibilitet til å kunne holde en høy takhøyde, falt valget på PICO-8.

PICO-8

PICO-8 er en såkalt "fantasikonsoll", noe Ritzl (2021) beskriver på følgende måte: "Fantasy consoles simulate the restricted hardware of an old system packaged into a user friendly experience with integrated tools for asset creation and game logic programming". Spillene man kan lage i PICO-8 har en oppløsning på 128x128 piksler med 16 tilgjengelig farger. På lik linje med gamle spillkonsoller som Game Boy og Nintendo Entertainment System (NES), består grafikken av "sprites", dvs. små bilder på 8x8 piksler som man kan bruke som byggeklosser til spillkarakter, objekter i spillet og bakgrunn. I tillegg til sprites, har man også kommandoer for å tegne geometriske former som linjer, sirkler og rektangler.

Fantasikonsoller som PICO-8 skiller fra andre spillutviklerverktøy som f.eks. Game Maker og Unity ved at alle verktøy er laget for å føles som en integrert del av den fiktive konsollen. Når man starter PICO-8 blir man møtt av en oppstartsekvens og en kommandolinje hvor man kan starte spill man har lagret eller åpne utviklerverktøyene. PICO-8 har en egen editor for å skrive kode og lage grafikk (se figur 1), samt en editor for å lage bakgrunn/kart, lydeffekter og musikk. PICO-8 har både en betalingsversjon som kan installeres på Mac, Windows og Linux i tillegg til en gratis utdanningsversjon som kjøres i nettleseren (www.pico-8-edu.com). De fleste studentene i dette prosjektet brukte gratisversjon.



Figur 1. Skjerm bilde av kode- og grafikk-editoren i PICO-8.

Det kan virke motsigende å velge et verktøy som er designet for å ha såpass mange restriksjoner når det tidligere ble argumentert for at programmeringsmiljøet som brukes ikke skal holde sterke elever og studenter igjen. Det er derimot flere grunner til valget av et verktøy med disse begrensningene. Det første er fokuset på en lav inngangsterskel, spesielt på områder som lærerstudenter i matematikk sannsynligvis ikke er like erfarne med, slik som programvare for å lage grafikk. De fleste vil i tillegg klare å lage noe grafikk når grafikken består av små bilder på 8x8 piksler og 16 farger. Restriksjoner kan derimot være også et nyttig verktøy for å fremme kreativitet (Dahl & Moreau, 2007). En lav inngangsterskel kan gi en følelse av kompetanse, noe som er en viktig faktor for å ha en positiv opplevelse av å jobbe med restriksjoner (Dahl & Moreau, 2007). Samtidig er det viktig at restriksjonene ikke forhindrer en følelse av autonomi (Dahl & Moreau, 2007). PICO-8 legger opp til dette ved å fokusere på tekstbasert programmering hvor man har stor fleksibilitet i hvordan man designer spillene. Programmering i PICO-8 utføres i Lua, et programmeringsspråk som brukes som skriptspråk («scripting language») i profesjonelle programmer f.eks. Adobe Lightroom og Davinci Resolve, i tillegg til å være et mye brukt skriptspråk innenfor spillutvikling. Å bruke et programmeringsspråk som brukes aktivt i dagens teknologi, kan, tatt opplevelsene til elevene i Weintrop og Wilensky (2015) i betraktning, muligens gi en økt følelse av legitimitet. Samtidig er det et programmeringsspråk med en enkel syntaks som gjør det passende for nybegynnere.

Undervisningsopplegget

For å gi studentene opplæring i PICO-8, ble det laget fire læringsvideoer hvor de to første fungerte som en introduksjon til PICO-8 og Lua, samt oppfriskning av programmeringskonsepter som variabler, løkker og if-setninger. I de to siste videoene ble et lite spill programmert fra bunn. Studentene fikk i oppgave å arbeide seg gjennom disse videoene før de begynte på arbeidet med spillet, ala et omvendt undervisningsopplegg. Studenter har ofte varierende forkunnskaper i programmering og et omvendt undervisningsopplegg blitt foreslått som en gunstig måte å kunne tilpasse opplæringen i sitt tempo (Attard & Busutill, 2020).

Spillet som ble laget i video 3 og 4 dannet også en slags mal som studentene kunne bruke for inspirasjon, samt få eksempel på viktige programmeringsmessige elementer i spill slik som game-loop, bevegelseskode, kollisjonskode osv. Å ta utgangspunkt i ferdig spill-kode er foreslått av Ke (2014) som en måte å minske kompleksiteten ved bruk av spillutvikling og spillprogrammering i skole. PRIMM-modellen (Predict, Run, Investigate, Modify, Make) anbefaler også å ta utgangspunkt i eksisterende kode før elever og studenter lager sin egen (Sentance et al., 2019). Før studentene begynte på arbeidet med utviklingen av sitt spill, fikk de oppgaver hvor de svarte på programmeringsspørsmål rundt PICO-8 relatert til videoene for å sjekke sin forståelse, samt hvor de måtte gjøre justeringer i koden til spillet som ble utviklet i video 3 og 4 for å endre egenskaper i spillet (f.eks. endre hastigheten til spilleren og legge til ny funksjonalitet).

For å få inn et fokus på tverrfaglighet, var et av kravene til spillet som skulle utvikles at tema i spillet måtte relateres til minst ett av de tverrfaglige temaene i LK20, "folkehelse og livsmestring", "demokrati og medborgerskap" og/eller "bærekraftig utvikling". Selv om bærekraftig utvikling ikke er en del av læreplanen i matematikk, ble dette likevel tatt med da det er et viktig punkt i læreplanen i de andre fagene og siden LK20 legger stor vekt på tverrfaglig arbeid. Studentene ble oppfordret til å se på læreplaner for andre skolefag, og hvordan de tverrfaglige temaene blir beskrevet der, for å få inspirasjon til hvordan de kunne fått inn aspekter fra disse i temaet de fokuserte på i spillet de skulle utvikle. Utviklingen av spillene var et prosjekt som gikk over hele semesteret, men det ble lagt opp til egne dager hvor mye av utviklingen foregikk. Figur 2 viser skjermbilde fra to av spillene utviklet av studentene. I spillet til venstre må spilleren rømme fra en flodbølge som oppstod pga. global oppvarming, mens i spillet til høyre må spilleren samle opp plast i havet før livet i havet dør. To andre grupper hadde også spill som handlet om søppel i havet, mens den siste gruppen fokuserte på sortering av søppel i sitt spill.



Figur 2. Skjermbilder fra to av spillene utviklet av studentene med PICO-8 (gjengitt med tillatelse).

Datainnsamling

For å få et innblikk i studentenes erfaringer med prosjektet, ble et anonymt spørreskjema med både lukkede og åpne spørsmål utdelt ved semesterslutt. I tillegg til spørreskjema, vil det også presenteres erfaringer som kom frem i personlig kommunikasjon med noen studenter. Her vil

det også i ett tilfelle presenteres en anekdote fra en student som deltok i en utprøving i 2023, altså fra en utprøving året etter pilotutprøvingen, for å eksemplifisere bruk av PICO-8 i matematikkundervisning (utprøvingen i 2023 ble gjort på lik måte som 2022, med unntak av at studentene jobbet i grupper på 2 mens de i 2022-utprøvingen jobbet i grupper på 4 til 5). All personlig kommunikasjon er gjengitt med tillatelse fra de aktuelle studentene.

Resultater

Av de 21 studentene som fulgte emnet i 2022, var det 18 som svarte på spørreskjemaet (ca. 86% svarprosent). Studentene ble bedt om å gradere i hvilken grad ulike ord/setninger var passende for å beskrive deres opplevelse av prosjektet. Resultatene vises i tabell 1. Tabellen viser at de fleste studenter hadde en positiv opplevelse av prosjektet hvor de opplevde det som morsomt, spennende, lærerikt, nyttig og givende. Dette til tross for at de fleste av disse studentene også opplevde prosjektet som frustrerende, vanskelig og tidkrevende. Under er eksempel på sitater fra studenter på evalueringsskjemaet om positive opplevelser med prosjektet:

Veldig gøy, ganske vanskelig, men givende.

Syns dette var noe av det morsomste vi har gjort på skolen.

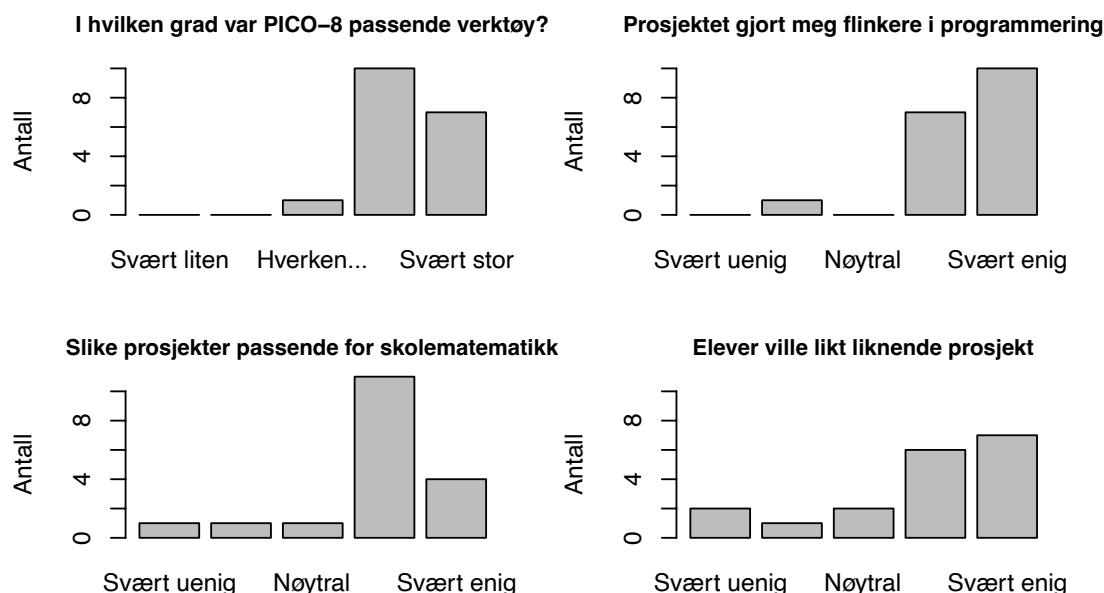
Synes dette er et viktig og bra prosjekt. Ingen tvil om at dette blir mere og mer relevant, og skal vi henge med i utviklingen (og ikke minst elevene vi skal undervise for) er vi nødt til å mestre dette.

Tabell 1. Spørsmål: «I hvilken grad vil du si at PICO8-prosjektet var». Tallene viser antall svar.

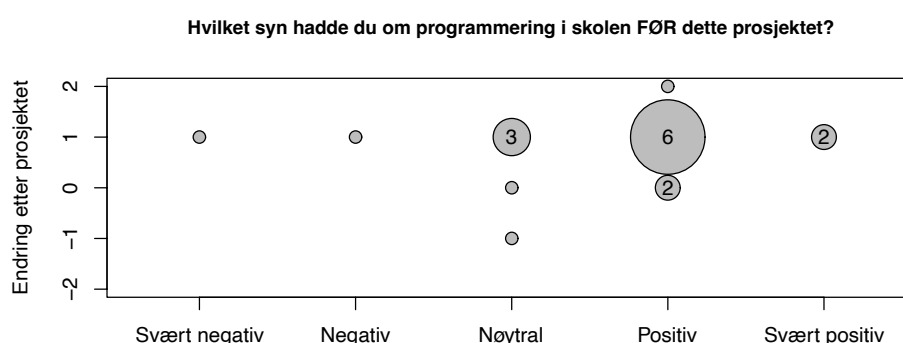
	Svært liten	Litt liten	Hverken stor eller liten	Stor	Svært stor
Morsomt	1	0	2	6	9
Frustrerende	0	1	3	10	4
Vanskelig	0	0	1	11	6
Givende	0	1	3	7	7
Spennende	1	0	1	10	6
Bortkastet tid	10	5	2	1	0
Lærerikt	0	1	0	8	9
Nyttig	0	1	2	8	7
Kjedelig	9	5	3	1	0
Tidkrevende	0	0	2	12	4

Studentene ble spurt hvorvidt de mente PICO-8 var et passende verktøy å bruke i et slikt prosjekt (figur 3). Én student svarte det mer nøytrale «Hverken stor eller liten grad», mens resten svarte enten «Stor» eller «Svært stor grad». Et flertall av studentene mente også at prosjektet har gjort dem flinkere i programmering (figur 3) og at det har hatt en positiv påvirkning på deres syn på bruk av programmering i skolen (figur 4). De fleste var også positive til hvordan et liknende prosjekt hadde passet inn i matematikkfaget i skolen (figur 3). Studentene var litt mer delt på hvorvidt de tenker elever i skolen ville likt et liknende prosjekt (figur 3), men flertallet av studentene mente dette var et prosjekt elever ville likt.

En student fortalte at hen hadde tatt med spillprosjektet inn i praksis til en klasse på 9. trinn hvor elevene hadde om ungt entreprenørskap og elevbedrift på skolen. Studenten brukte der spillet hen hadde vært med å lage som et eksempel til elevene hvordan man kan knytte elevbedrift opp mot FNs bærekraftsmål, til stort engasjement hos elevene, ifølge studenten. Studenten fortalte videre at flere elever sa de skulle ønske de hadde lært å programmere tidligere slik at de også kunne lage spill, og de spurte lærerstudenten om hen kunne hjelpe dem lage liknende spill. En av elevgruppene brukte også spillet som inspirasjon til sin elevbedrift.



Figur 3. Spørsmål på spørreskjemaet: «I hvilken grad var PICO8 et passende verktøy for et spillprogrammeringsprosjekt som dette?», «PICO8-prosjektet har gjort meg flinkere i programmering», «Prosjekter ala PICO8-prosjektet kan være passende i matematikkfaget i skolen», «Jeg tror elever i skolen ville likt å jobbe med et liknende spillprogrammeringsprosjekt». Ordlyden på svaralternativet i midten på det første spørsmålet var «Hverken stor eller liten».



Figur 4. «Hvilket syn hadde du om programmering i skolen FØR dette prosjektet» (x-akse) og «Har PICO8-prosjektet hatt noen påvirkning på ditt syn på programmering i skolen?» (y-akse). Tallene i sirklene viser antall svar. Sirkler uten tall hadde ett svar. Tallene på y-aksen går fra «Ja, jeg er svært mer positiv nå» (2) til «Ja, jeg er svært mer negativ nå» (-2). Tallet 0 betyr at de mener prosjektet ikke har endret deres syn.

En annen lærerstudent rapporterte også at hen hadde tatt frem spillprosjektet i sin praksis, her for å skape ro og engasjement i en urolig klasse. Under er et utdrag fra en epost jeg fikk fra studenten:

Jeg hadde hatt 2 helt forferdelige timer på rad, i en utfordrende klasse. Det var slåsskamper og snøballkasting og fotballspilling osv. innendørs, mens til en hver tid minst 3 jenter gråt eller hadde vondt i hodet... Jeg hadde brukt opp det meste av kreativitet da jeg kom på å si "vet dere at jeg har programmert et dataspill?" Umiddelbar stillhet i rommet. 24 stykker i kø som pent spurte om de kunne få prøve. De lagde seg konkurranse om å få flest poeng, og lærte seg selv og hverandre menyene. Gleden og overraskelsen var stor da en av gutta plutselig fikk "win-screen" 🎉 (Til MIN glede, så var det han det hadde gått aller mest utover resten av dagen - og nå fikk han helte-status). Flere av gutta ytret ønske om å få prøve å endre hastighet osv. på spillet, så jeg tenker jeg drar med meg noen ut i grupper neste gang vi er flere voksne :)

Det kom derimot også frem kritikk av prosjektet fra studentene, blant annet at gruppene kunne være noe store og at det kunne til tider være ujevn fordeling av hvem som faktisk utførte mesteparten av programmeringen. Et av spørsmålene på spørreskjemaet spurte hvorvidt studentene mente prosjektet burde videreføres til neste års studenter. Alle studenter utenom én student mente prosjektet burde videreføres.

Det var også en student fra utprøvingen i 2023 som tok prosjektet med seg ut i praksis, hvor studenten brukte PICO-8 til å innføre tekstbasert programmering til en 9-klasse som fra før kun hadde erfaring med blokkprogrammering. Studenten fokuserte derimot ikke direkte på spillutvikling i dette tilfellet, men brukte PICO-8 til å lage oppgaver rundt geometri og koordinatsystem. I PICO-8 er koordinatsystemet «opp-ned» slik at y-aksen peker nedover, og studenten la opp til refleksjoner og diskusjoner rundt koordinatsystemet og hvordan elevene måtte programmere for å endre størrelsen på geometriske figurer. Dette var til stor entusiasme hos både elever og matematikklæreren som vanligvis underviste faget, ifølge studenten selv. Studenten skrev følgende om matematikklæreren:

Hun var veldig glad og entusiastisk siden dette ikke var noe hun selv kunne undervise i og ønsket at vi kunne gjøre flere slik undervisningstimer sammen.

Diskusjon og konklusjon

Selv om lærerstudentene hadde lite erfaring med programmering fra tidligere, med unntak av litt programmering i Python fra tidligere emner, tyder resultatene på at de fleste hadde en positiv og lærerik opplevelse av prosjektet og at de ikke opplevde det som for avansert og overveldende. I Andersen (2022) uttrykte en av lærerne i studien at de mente det kunne bli utfordrende å gå fra enkel programmering med noen få linjer, til å sette dette sammen til en større struktur og mer sammensatt program. Dette er noe som kreves for å lage et spill, og resultatene fra denne utprøvinger viser at dette er mulig selv for studenter (og dermed også lærere) med lite programmeringserfaring fra tidligere. Det betyr ikke at det var et enkelt prosjekt for studentene, noe vi ser i resultatene hvor de fleste opplevde spillprogrammeringen som både vanskelig og frustrerende (tabell 1). Utholdenhet er en viktig del av en algoritmisk tenker og det er mulig at fokuset på utvikling av spill kan være en såpass motiverende faktor

at det gir studenter (og elever) nok pågangsmot til å holde ut selv når det oppleves som vanskelig.

De fleste studentene var også positive til hvordan et liknende prosjekt kunne være passende for matematikkundervisning i grunnskolen. Selv om historiene fra studentene som tok spillene sine ut i praksis er anekdotiske, viser det hvilket potensial spill og spillutvikling har for kunne skape engasjement hos elever. Man kan argumentere at det var selve spillene som var hovedfokuset hos elevene (og ikke selve utviklingen), men samtidig endte begge historier fra 2022-studentene med at elevene selv ønsket kunnskap om programmering for å lage egne spill. At spill kan skape økt motivasjon og interesse er også i tråd med tidligere studier nevnt tidligere (Ke, 2014; Dekhane et al., 2013). Det er flere måter å innføre et liknende prosjekt i skolesammenheng. Dersom man har klasser hvor elevene har lite erfaring med programmering, eller der man har lite tid til et slikt prosjekt, kan en mulighet være at elever tar utgangspunkt i ferdig spill-kode hvor elevene skal gjøre endringer i koden for å endre oppførsel i spillet. Som nevnt tidligere, har Ke (2014) argumentert for at å ta utgangspunkt i ferdig spill-kode, kan være en måte å minske kompleksiteten med spillutvikling og programmering. Dette kan man også gjøre i større prosjekter, f.eks. ved at elevene tar utgangspunkt i ferdig spill-kode som de bruker som et startpunkt for videre utvikling.

Anekdoten fra 2023-studenten viser også et eksempel på andre måter enn direkte spillutvikling hvor spillutviklingsverktøy som PICO-8 kan brukes i matematikkundervisning. Her ble PICO-8 og programmering brukt for å legge opp til refleksjoner og diskusjoner rundt geometri. Geometri er en vanlig innfallsvinkel til å lære tekstbasert programmering med Python og turtle-biblioteket (se f.eks. Kravik, 2020). Å koble programmeringsundervisningen til noe visuelt, slik som å lage figurer med turtle eller bildemanipulering, kan være motiverende i når elever lærer programmering (Araujo et al., 2018). Verktøy som PICO-8 har derimot en fordel ved at det er enkelt å utvide programmeringen fra å jobbe med geometriske former, slik man ville gjort med f.eks. turtle-biblioteket, til å inkludere elementer fra spillprogrammering slik at man kan få de motiverende og tverrfaglige fordelene med å jobbe med spill. I tillegg til kommandoer for å tegne geometriske figurer, har man i PICO-8 også mulighet for å lage pseudotilfeldige tall slik at det også er et mulig verktøy for å utforske egenskaper til sannsynlighet i matematikkundervisningen. Dette er også et element som kan kobles inn i spillprogrammeringen, f.eks., utforske hvordan oppførselen til et spill endrer seg dersom man endrer sannsynligheten for hvor ofte en hendelse skjer i spillet. Dersom man underviser på høyere trinn, man kan her f.eks. koble dette til forventningsverdier hvor elevene gjør utregninger på forhånd før de implementerer sannsynlighetene i spillene.

Mladenović et al. (2016) argumenterer for at programmering i matematikkfaget må fokusere på mer enn utelukkende å utvikle matematikk-logisk tenking og åpne for å legge til rette for "creativity enabling environments" hvor flere ferdigheter brukes for å være mer tiltrekkende for dagens digitale elever. Spillprogrammering passer godt inn i en slik beskrivelse og det er også mange muligheter for også skape tverrfaglig arbeid på tvers av fag da det å lage et spill kan kreve ferdigheter fra mange skolefag, som f.eks. matematikk (programmeringslogikk), naturfag (bevegelse og annen fysikk), kunst og håndverk (det visuelle uttrykket), norsk (fortellerteknikker i et interaktivt medium) og samfunn (tematikk i spillet).

Referanser

- Akpinar, Y., & Aslan, Ü. (2015). Supporting children's learning of probability through video game programming. *Journal of Educational Computing Research*, 53(2), 228-259.
- Andersen, V. T. (2022). *Utfordringer og erfaringer med programmering* (Master's thesis, University of South-Eastern Norway).
- Araujo, L. G. J., Bittencourt, R. A., & Santos, D. M. (2018, February). An analysis of a media-based approach to teach programming to middle school students. I *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 1005-1010).
- Attard, L., & Busuttil, L. (2020). Teacher perspectives on introducing programming constructs through coding mobile-based games to secondary school students. *Informatics in Education*, 19(4), 543-568.
- Bennedsen, J., & Caspersen, M. E. (2008). Exposing the programming process. In *Reflections on the Teaching of Programming* (pp. 6-16). Springer, Berlin, Heidelberg.
- Bulut, D., Samur, Y., & Cömert, Z. (2022). The effect of educational game design process on students' creativity. *Smart Learning Environments*, 9(1), 8.
- Corradini, I., Lodi, M., & Nardelli, E. (2017, August). Conceptions and misconceptions about computational thinking among Italian primary school teachers. In *Proceedings of the 2017 ACM conference on international computing education research* (pp. 136-144).
- Dahl, D. W., & Moreau, C. P. (2007). Thinking inside the box: Why consumers enjoy constrained creative experiences. *Journal of Marketing Research*, 44(3), 357-369.
- Dekhane, S., Xu, X., & Tsoi, M. Y. (2013). Mobile app development to increase student engagement and problem solving skills. *Journal of Information Systems Education*, 24(4), 299-308.
- DiSessa, A. A. (1997, August). Twenty reasons why you should use Boxer (instead of Logo). In *Learning & Exploring with Logo: Proceedings of the Sixth European Logo Conference, Budapest, Hungary* (pp. 7-27).
- Doerschuk, P., Juarez, V., Liu, J., Vincent, D., Doss, K., & Mann, J. (2013, October). Introducing programming concepts through video game creation. In *2013 IEEE Frontiers in Education Conference (FIE)* (pp. 523-529). IEEE.
- Forsström, S. E., & Kaufmann, O. T. (2018). A literature review exploring the use of programming in mathematics education. *International Journal of Learning, Teaching and Educational Research*, 17(12), 18-32.
- Hijón-Neira, R., Santacruz-Valencia, L., Pérez-Marín, D., & Gómez-Gómez, M. (2017, November). An analysis of the current situation of teaching programming in Primary Education. In *2017 International Symposium on Computers in Education (SIIE)* (pp. 1-6). IEEE.
- Johnson, C. (2017). Learning Basic Programming Concepts with Game Maker. *International Journal of Computer Science Education in Schools*, 1(2), n2.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & education*, 73, 26-39.
- Kravik, R. (2020). Programmering og geometri. *Tangenten – tidsskrift for matematikkundervisning*, 31(2), 7–11.

- Kravik, R., Berg, T. K., & Siddiq, F. (2022). Teachers' understanding of programming and computational thinking in primary education—A critical need for professional development. *Acta Didactica Norden*, 16(4), 23-sider.
- Masuch, M., & Nacke, L. (2004, November). Power and peril of teaching game programming. In *Proceedings of the 5th Game-On International Conference, S* (pp. 347-357).
- Misfeldt, M., Szabo, A., & Helenius, O. (2019, February). Surveying teachers' conception of programming as a mathematics topic following the implementation of a new mathematics curriculum. In *Eleventh Congress of the European Society for Research in Mathematics Education* (No. 13). Freudenthal Group; Freudenthal Institute; ERME.
- Mladenović, M., Krpan, D., & Mladenović, S. (2016). Introducing programming to elementary students novices by using game development in Python and Scratch. *EDULEARN16 Proceedings*, 1622-1629.
- Ritzl, Björn (9 March 2021). "Developer case study - From fantasy computer to Defold". Defold Foundation. Hentet 2 Februar 2023
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education*, 29(2-3), 136-176.
- Sevik 2016. Programmering i skolen. Hentet fra:
https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf
- Weintrop, D., & Wilensky, U. (2015, June). To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th international conference on interaction design and children* (pp. 199-208).