

Rapport – Visuell programmering for «ikke-programmerere»

Andreas B. Olaussen

Emil Berglund

Sebastian W. Thomsen

Høgskolen i Østfold

Emnekode: ITF10219

Ord: 1852

Rapport – Visuell programmering for «ikke-programmerere»

I denne rapporten retter vi fokuset vårt mot en nøye utforskning av visuell programmering som et middel for å klargjøre kodingens kompleksitet for en bredere gruppe enkeltpersoner, spesielt de som ikke identifiserer seg som erfarne "programmerere." Vår analyse starter med en grundig presentasjon av visuell programmering, med særlig vekt på å skille denne metoden fra den mer tradisjonelle tekstbaserte tilnærmingen. Denne innføringen danner grunnlaget for en omfattende diskusjon om fordeler, utfordringer og konkrete anvendelser som illustrerer hvordan visuell programmering kan være en bro til tekstbasert koding.

I rapporten skal vi også diskutere bruken av visuell programmering i skole- og utdanningssammenheng. Denne delen vil inkludere eksempler på hvordan visuelle verktøy og plattformer brukes i klasserommet for å hjelpe elever med å utvikle sine programmeringsferdigheter. Vi vil også presentere relevant forskning og studier.

Et viktig høydepunkt i rapporten vil være vår utforskning av de konkrete programmeringskonseptene som kan undervises ved hjelp av visuelle verktøy. Vi vil presentere og diskutere ulike eksempler fra barneskoler der visuell programmering har blitt brukt til å lære barn grunnleggende programmering. Dette gir en innsikt i hvordan visuell programmering kan hjelpe unge elever med å utvikle viktige ferdigheter for fremtiden.

Hva er visuell programmering

Visuell programmering er en type programmering hvor man bruker et grafisk brukergrensesnitt for å skrive eller lage kode. Ved hjelp av illustrasjoner og visuelle elementer kan man kode på en enklere, mer praktisk og raskere måte enn et tradisjonelt programmeringsspråk (Muscad, 2022).

Et typisk eksempel på et visuelt programmeringsspråk er «Scratch». Visuell programmering er typisk basert på et konsept som kalles for «drag-and-drop», noe som

Scratch tar fullt utbytte av. «Drag-and-drop» er igjen en type programmering der man kan dra og slippe ulike kommandoer som til sammen danner et fungerende program (Shala, 2022).

Scratch utnytter også noe som kalles for blokkprogrammering. Blokkprogrammering er egentlig veldig likt som «drag-and-drop», men går ut på at brukeren skal dra og slippe visuelle blokker som representerer kodeelementer. (Gilbert, 2022).

Scratch

Scratch har så vidt blitt nevnt og vil bli videre brukt i oppgaven. Scratch er som nevnt et blokkbasert programmeringsspråk som er utviklet for såkalte «ikke-programmerere» eller nybegynnere. Dette verktøyet har gjort programmering tilgjengelig for alle, uavhengig av alder eller erfaring. Det som gjør Scratch til en så fin begynnerplattform er fargekodene og de kategoriserte blokkene. Dette enkle, visuelle systemet gjør det mulig for brukere å bygge programmer ved å dra og slippe blokker som representerer handlinger og logikk (Statped, 2022)

Scratch gir en visuell tilnærming til programmering ved å la brukerne lage interaktive historier, animasjoner og spill ved hjelp av fargekodede blokker som er enkle å forstå og kombinere. Dette gjør det mulig for nybegynnere å utforske grunnleggende programmeringskonsepter som sekvenser, løkker og betingelser på en intuitiv måte. Det er i hovedsak derfor Scratch er så fin for målgruppen sin (Tarakki100, 2022).

Uansett om du er en barneskoleelever som utforsker grunnleggende programmering for første gang, en lærer som ønsker å introdusere programmering i klasserommet, eller en voksen som er nysgjerrig på å lære programmering uten forutgående erfaring, er Scratch en plattform som gir alle muligheten til å utvikle programmeringsferdigheter.

Forskjell på tekstbasert og visuell programmering

Det er allerede nevnt at visuell programmering er en type programmering som bruker grafiske elementer for å lage kode. I motsetning til visuell programmering, som bruker grafiske elementer, bruker tekstbasert programmering tekst for å lage kode. Tekstbasert programmering avhenger at brukeren faktisk skriver alt det hen vil oppnå med programmet, hvor man i visuell programmering som oftest drar inn det man ønsker for å oppnå det man vil.

Visuell programmering og tekstbasert programmering er utrolig forskjellig fra hverandre på mange områder. Ett av flere områder er enkelhet. Hvis man tar utgangspunkt i Scratch drar man ferdiglagde kodeblokker inn på et hoved-ark hvor man gjør finjusteringer. I tekstbasert programmering derimot må man skrive hver eneste linje. Dette fører oss til lesbarhet. Tar man utgangspunktet i en komplett nybegynner, vil den personen mest sannsynlig finne det lettere å forstå hva som foregår i et Scratch-program, i motsetning til et Java-program. I Scratch har man farger som deler inn hvilke funksjoner kodeblokken har, i tillegg til at det er veldig ortofont. Koden er med andre ord skrevet i nærheten til hvordan vi snakker, og er veldig oversiktlig grunnet fargebruken. Det samme kan ikke sies for tekstbasert programmering. I nesten alle tekstbaserte programmeringsspråk har man symboler, funksjoner og forkortelser som gjør læringskurven veldig lang og bratt, og ikke minst vanskeligere å forstå. I tillegg er fargebruken mindre kreativ, og i noen programmer har man ingen farger i det hele tatt (Statped, 2022)

Likevel trenger man tekstbasert programmering for å lage mesteparten av det vi bruker programmering til i dag. Nettsider lages ved hjelp av «HTML», «CSS» og «JavaScript», hvor alle de nevnte er tekstbaserte programmeringsspråk som kreves for å lage en fullfunksjonerende nettside. Alle apper vi har på telefonene våre er laget med tekstbaserte programmeringsspråk. Blokkbasert koding kan derfor være en måte å introdusere koding for «ikke-programmerere», men det er mye vanskeligere å forstå ekte kodingskonsepter og

syntaks når man i hovedsak drar og slipper bokser. Med tekstbasert koding får man en full læringsopplevelse som inneholder viktige kodekonsepter som man vil huske og kan bygge videre på. (Darvell, 2021)

Samlet sett viser dette hvordan det på ingen måte er enkelt for noen å starte rett på tekstbasert koding for første gang, men at visuell programmering er et fint sted å starte for ikke ikke-programmerere». Ettersom det er mer visuelt og intuitivt enn tekstbasert programmering er det lettere for nybegynnere å forstå hvordan programmering fungerer.

Fordeler og utfordringer med visuell programmering

Programmering kan for mange som ikke har noe særlig erfaring med det virke veldig overveldende. Det er mye tekst og fremmedord som er satt sammen på en måte man ikke har sett før som skal få datamaskinen til å utføre visse handlinger. Noe som kan gjøre det lettere å starte med programmering er visuell programmering. Det er flere grunner til at det er fordelaktig å bruke visuell programmering til å lære nybegynnere å programmere. For det første slipper man å forholde seg til de tradisjonelle teksttunge programmeringsspråkene. Ved hjelp av visuelle figurer, symboler og illustrasjoner vil man kunne løse ulike problemer og oppgaver. En annen grunn til at visuell programmering kan være en god måte å lære nybegynnere programmering på er at ved hjelp av de visuelle elementene vil det for mange være lettere å huske, i motsetning til et tekstbasert kodespråk (Muscad, 2022).

Det er dermed ikke sagt at det ikke finnes utfordringer og begrensninger ved denne tilnærmingen. En av begrensningene til visuelle programmeringsspråk er at det ikke er like kraftig og fritt som tradisjonelle tekstspråk. Dette gjør at om man ønsker å gjøre store oppgaver vil det etter hvert være nødvendig å benytte seg av et av disse tekstspråkene. Det vil også være vanskeligere å jobbe med og finne feil i koden i et visuelt programmeringsspråk enn i et tekstbasert programmeringsspråk om det blir for mye kode.

Det som er fint med visuell programmering er at det finnes flere ulike programmer og

tjenester for enhver aldersgruppe. Om man ønsker å benytte seg av visuell programmering til læring i en gruppe med 5. klassinger så finnes det tjenester som tilbyr dette, og andre tjenester som tilbyr mer komplekse problemløsninger til for eksempel videregående skoleelever (Sevik et al., 2016)

Praktiske eksempler

Når det kommer til hvordan man kan implementere programmering i skolen, vil de fleste tenke på fag som matematikk og naturfag der programmering er lagt inn som et konkret mål for fagene, men det er også muligheter i andre fag. “Elevene kan programmerere en interaktiv fortelling på norsk eller engelsk, bygge roboter som løser reelle eller konstruerte problemer, kode skrittellere til bruk i kroppsøving eller lage en kalkulator som kan konvertere mellom måleenheter til bruk i mat og helse” (Eidslott, 2021). Eidslott (2021) hevder også at det kan være lurt å finne små problemer og utfordringer fremfor store og kompliserte oppgaver, slik at det ikke blir for overveldende (Eidslott, 2021).

I Spania ble det gjennomført en studie over to år på fem forskjellige skoler der de analyserte 107 elever i femte- og sjetteklasse. Der brukte de blant annet Scratch til å gjennomføre ulike prosjekter. Ifølge Sáez-López gjorde det at de jobbet i prosjekter med visuell programmering at elevene var mer motiverte, entusiastiske og engasjerte. Det viste seg også at bruken av visuell programmering økte elevens kunnskap om programmeringskonsepter, logisk- og algoritmisk tenkning (Sáez-López et al., 2016).

Programmering i skole- og utdanningssammenheng ble startet opp i 2020, slik at elever skal kunne få en bredere forståelse innenfor flere ulike fag, og samtidig tilegne seg forkunnskaper innenfor programmerings teknologi (Vogt, 2021).

Grunnen til at dette faget har blitt en del av skolegangen handler om at det meste av hva vi omgir oss med er teknologi og består av koder. Programmering kan brukes til det aller meste og det er et godt verktøy innenfor for eksempel matematikk, naturfag og kunst. Å lære

seg programmering kan gi flere fordeler og være med på å øke logisk og kritisk tenkning.

Ifølge Statped (2021) er programmering i lærerplanen fra 5.klasse. De lærer hovedsakelig visuell programmering, eller blokkprogrammering som “Scratch”. Her lærer barna programmering med hjelp av spill og lek. Siden visuell programmering er enklere enn den tradisjonelle programmeringen som blir skrevet med tekst, er det enklere for barn å lære seg nybegynnerferdigheter innenfor programmering.

Ifølge Statped (2021) er visuell programmering, som for eksempel ved bruk av robot et godt tiltak for barn med lærevansker. De peker videre på at grunnen til dette kan være at barna får en slags eierskap til den visuelle programmeringen, da de er med på å skape noe.

Scherer et al., (2019) har utført en metaanalyse hvor de viser at programmering er assosiert med flere ulike kognitive fordeler. Funnene fra denne metaanalysen peker mot at programmering kan bidra til økt kreativ tenkning, bedre matematiske ferdigheter, økt resonneringsferdigheter og økt skoleprestasjoner. Elevene som fikk øvd på programmering, økte betydelig innenfor nevnte fordeler fremfor barna som ikke programmerte.

Skoler og lærere kan implementere visuell programmering i undervisningen ved å starte med enkle opplæringsverktøy slik som Scratch slik at elevene får grunnleggende forståelse for programmering, før de går videre til vanskeligere programmer slik som Python. Ved å introdusere visuelle programmering har de en bedre grunnforståelse for å lære seg andre programmeringsspråk senere i skolegangen.

Mange fagskoler og Høgskoler tilbyr videreutdanning til lærere innenfor programmering fra 1-7 trinn, hvor lærere kan lære seg programmering og hvordan de kan undervise elevene innenfor dette temaet.

Vi ser et klart potensiale for visuell programmering som et inngangspunkt for å engasjere studenter og ikke-programmerere. Det gir en lavere inngangsterskel for nybegynnere ved å tilby en visuell og intuitiv tilnærming til programmering. Likevel, som

påpekt i teksten, er det viktig å erkjenne begrensningene og behovet for tekstbasert programmering for å mestre avanserte konsepter. En balansert tilnærming som kombinerer visuell og tekstbasert koding, kan gi en solid læringsopplevelse. I utdanningen kan visuell programmering være et kraftig verktøy for å utvikle grunnleggende konsepter og øke interessen for koding blant studenter. Det er en nøkkel til å bygge et solid fundament for fremtidig teknologisk forståelse.

Litteraturliste:

Darvell, B. (2021). *Block-based Coding vs. Text-based Coding*.

<https://bsd.education/block-based-coding-vs-text-based-coding/>

Eidslott, H. (2021). *5 råd for å komme i gang med programmering i skolen*.

<https://www.utdanningsnytt.no/fagartikkel-pedagogikk-programmering/5-rad-for-a-komme-i-gang-med-programmering-i-skolen/282639>

Gilbert, J. (2022). *What is Block Coding?* CodeWizardsHQ.

<https://www.codewizardshq.com/what-is-block-coding/>

Muscad, O. (2022). *What Is Visual Programming & Why Is It Important?* - DataMyte.

<https://datamYTE.com/visual-programming/>

Sáez-López, J-M., González M. R., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*.

<https://doi.org/10.1016/j.compedu.2016.03.003>

Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational*

Psychology, 111(5), 764–792.

<https://doi.org/10.1037/edu0000314>

Selvik, K., m.fl. (2016). Programmering i skolen. *Notat fra Senter for IKT i utdanningen*.

https://www.udir.no/globalassets/filer/programmering_i_skolen.pdf

Shala, D. (2022). *Drag and Drop Coding: Visual Coding Without Programming Skills*.

<https://www.no-code-software.com/no-code/drag-and-drop-coding/>

Statped (2021, 01. Mars). *Programmering*. Statlig spesialpedagogisk tjeneste.

<https://www.statped.no/laringsressurser/teknologitema/programmering-for-barn-med-saerskilte-behov/programmering/programmeringssprak/>

Statped. (2022, 25. Januar). *Scratch*. Statlig spesialpedagogisk tjeneste.

<https://www.statped.no/laringsressurser/teknologitema/scratch/>

Tarakki100. (2022) *Introduction to Visual Programming Language*. (2021).

<https://www.geeksforgeeks.org/introduction-to-visual-programming-language/>

Vogt, Y. (2021, 01. Februar). *Programmering blir allemannseie i skolen*.

Forskningsmagasinet Apollon.

https://www.apollon.uio.no/artikler/2021/1_utdanning_programmering.html