

Institutt for lærerutdanning og pedagogikk

Koding som digital grublis

En kvalitativ studie om hvordan elevenes læringsstrategier påvirkes gjennom programmering.

—

Susanne Iversen

Masteroppgave i lærerutdanning for 1.-7. trinn mai 2015

Sammendrag

Dette prosjektet viser hvordan læringsstrategiene til elevene påvirkes gjennom programmering i Scratch. Elevene jobber problemløsende, men benytter få problemløsningsstrategier. Prosjektet går ut på å undersøke om, og hvordan elever på mellomtrinnet jobber problemløsende i programmering, og om de benytter seg av de kjente problemløsningsstrategiene.

Prosjektet er en kvalitativ studie hvor det er gjennomført fire ulike steg i innsamlingen med bruk av videoopptak. Steg én er en innføringstime for full klasse hvor de fikk innføring i programmet Scratch. Scratch er et programmeringsprogram og språk lagd for å lære barn programmering. Til de neste stegene blir det gjort et utvalg på fire elever. I steg to får de fire elevene en ny oppgave i Scratch som er problemløsende. Under steg tre regner elevene to tradisjonelle problemløsningsoppgaver i matematikken. Det siste og fjerde steget er et gruppeintervju med de fire elevene. På bakgrunn av dette blir elevenes problemløsende jobbing i programmeringen og matematikken sammenlignet.

Resultatene viser blant annet at elevene jobbet likt med strategier i både matematikken og programmeringen, og det kommer tydelig frem at elevene trenger mer opplæring i strategier og problemløsende arbeid. Spesielt virker ikke elevene vant med å vurdere sin egen læringsprosess. Med tanke på deling og samarbeidskultur er det mye bedre under programmeringen.

Nøkkelord: Problemløsning, problembasert læring, læringsstrategier, programmering, Scratch, problemløsningsstrategier, vurdering av læringsprosess

Forord

Snart er fem år på universitetet ferdig, og denne masteroppgaven setter punktum på min lærerutdanning. Mest av alt vil jeg takke mine to veiledere, Karin Rørnes og Geir Olaf Pettersen ved UiT Norges arktiske universitet, for all støtte, hjelp og innspill til denne oppgaven. Jeg er evig takknemlig for alle deres konstruktive tilbakemeldinger. Resultatet hadde ikke blitt det samme uten deres hjelp.

Jeg vil også takke læreren for å ha stilt sin klasse og sine elever disponibel for mitt arbeid i en travel hverdag. En spesiell takk til klassen, og de fire utvalgte elevene, uten dere ville ikke prosjektet vært gjennomførbart.

Videre vil jeg takke min søster Julianne, og hennes samboer Tor, for at dere viste meg og fikk meg interessert i programmering for barn. Dere skal også ha mange takk for all støtte og oppmuntrende ord, og ikke minst middager, under dette arbeidet.

Jeg vil også takke mine medstudenter for støtte og hjelp, og ikke minst for alle gode diskusjoner, men også latterfulle pauser, på «kontoret».

Til slutt vil jeg takke mamma og pappa for alle oppmuntrende ord, støtte og også faglige diskusjoner underveis i dette arbeidet.

Tromsø, mai 2015

Susanne Iversen

Innholdsfortegnelse

Sammendrag	i
Forord	iii
Innholdsfortegnelse	v
1.0 Innledning	1
1.1 Bakgrunn	1
1.2 Avgrensning og problemstilling	3
1.3 Disposisjon	4
2.0 Scratch	5
2.1 Lær Kidsa Koding og Kodeklubber.....	5
2.2 Scratch	5
3.0 Teorikapittel	9
3.1 Perspektiv på læring	9
3.1.1 Dataprogrammenes utvikling gjennom paradigmene	11
3.2 Problembasert læring.....	12
3.3 Problemløsning	13
3.4 Hva kjennetegner en problemløser?	14
3.5 Programmering	16
3.5.1 Problemløsning i programmering	17
3.6 Grunnleggende ferdigheter	18
4.0 Metodekapittel	21
4.1 Utvalg	21
4.1.1 Utvalg av klasse	21

4.1.2 Utvalg av elever	22
4.2 Kvalitativ metode	22
4.3 Arbeidets forskningsdesign	23
4.4 Datainnsamlingsmetode.....	24
4.4.1 Observasjon med videoobservasjon.....	24
4.4.2 Intervju – gruppeintervju	25
4.5 «Aksjonen» i klasserommet.....	27
4.5.1 Beskrivelse av utvalg	27
4.5.2 Analyse av elevoppgaver	28
4.5.3 Beskrivelse av innsamlingen.....	30
4.6 Analysemetode	32
4.7 Etske drøftelser og metodekritikk	33
4.7.1. Informasjon og samtykke.....	33
4.7.2 Etske overveielser	34
4.7.3 Metodekritikk.....	34
5.0 Analyse og diskusjon.....	37
5.1 Elevenes problemløsningsstrategier	37
5.1.4 Diskusjon rundt elevenes problemløsningsstrategier.....	43
5.2 Elevenes vurdering av egen læringsprosess	44
5.2.1 Diskusjon rundt elevenes vurdering av læringsprosessen.....	45
5.3 Elevenes deling, samarbeid og spørsmålsstilling	46
5.3.3 Diskusjon rundt elevenes samarbeids- og delingsvilje	47
5.4 Oppsummering og videre diskusjon	48
5.4.1 Presentasjon av data i en matrise	49
5.4.2 Refleksjoner	51

6.0 Konklusjon og videre arbeid	53
6.1 Videre arbeid	55

Litteraturliste.....	57
-----------------------------	-----------

Liste over vedlegg:

Vedlegg 1 Forklaring om programmering	
Vedlegg 2 Observasjonspunkter.....	
Vedlegg 3 Felix og Herbert- oppgave til steg 1	
Vedlegg 4 Spørsmål til elever	
Vedlegg 5 Godkjennelse fra NSD	
Vedlegg 6 Informasjonsskriv med svarslipp	

Liste over figurer

Figur 1 Bildeutsnitt av programmet Scratch (MIT, u.å.).....	6
Figur 2 Forklaring av ulike kategorier i Scratch (Wikipedia, 2015, egen oversettelse).....	7
Figur 3 Matematikkoppgave 1 (Alseth m.fl., 2006).....	29
Figur 4 Matematikkoppgave 2 (Matematikk.org)	30
Figur 5 Utsnitt av kode som Hilde og Fred forsøker. Eksisterende kode fra introduksjonsøkten til venstre, elevenes forsøk til høyre.	39
Figur 6 Utsnitt av kode som Bente og Fred forsøker. De legger inn nye brikker i den eksisterende koden fra introduksjonstimen.	40
Figur 7 Matrise av egnethet av strategi hos elevene i programmerings- og matematikkøkten	51

1.0 Innledning

1.1 Bakgrunn

Samfunnet i dag betegnes som et kunnskapssamfunn og den teknologiske utviklingen har gjort kunnskap tilgjengelig med noen tastetrykk og et søk. Skolen kan derfor ikke lengre satse på at «pugg og husk»-strategier er det som fremmer læring for elever som vokser opp i en digital tidsalder. Hvordan elevene tilegner seg kunnskap har derfor vært sentral i læringsforskningen de siste årene. Prosessen frem til målet, fremfor målet i seg selv, har blitt mer vektlagt. I 2013 ble det utnevnt et utvalg ledet av Sten Ludvigsen (kalt Ludvigsenutalget) som skal undersøke og komme med rapporter for hvordan fremtidens skole skal bli. Utvalget sier blant annet følgende om elevens læring: *«Blant annet viser forskningen at det å lære noe i dybden, reflektere rundt egen læring og bruke læringsstrategier fremmer elevenes læring og kan gi dem et grunnlag for å lære gjennom hele livet»* (NOU, 2014:31). Det å lære elevene hvordan en skal tilegne seg kunnskap er en viktig ferdighet elevene tar med seg for fremtiden. Å jobbe problemløsende er en måte å jobbe på for å «lære seg å lære», og for å få mer fokus på læringsprosessen. Det får elevene til å reflektere, undersøke, stille spørsmål og komme frem til en forståelse på egen hånd, eller sammen med andre (Bjørke, 2000). Dette gjør at elevene får utviklet sine læringsstrategier og organisert egen læring. Læreplanen *Kunnskapsløftet 2006* fremhever læringsstrategiene i prinsipper for opplæringen, og forklares slik:

«Dette er strategier for å planlegge, gjennomføre og vurdere eget arbeid for å nå nasjonalt fastsatte kompetansemål. Det innebærer også refleksjon over nyervervet kunnskap og anvendelse av den i nye situasjoner. Gode læringsstrategier fremmer elevenes motivasjon for læring og evne til å løse vanskelige oppgaver også i videre utdanning, arbeid eller fritid»
(Utdanningsdirektoratet, 2012¹)

Gjennom hele min skolegang har jeg vært interessert i, og selv likt problemløsningsoppgaver og problemløsende arbeid. I matematikken har det i de siste årene blitt mer oppmerksomhet omkring problemløsning fordi dette fremmer at matematikk ikke bare er ferdighetstrening

med regler, regning og algoritmer (Löwing og Kilborn, 2002). I tillegg til problemløsning har jeg alltid vært interessert og opptatt av digitale ferdigheter. Høsten 2013 ble jeg introdusert for programmering for barn gjennom bevegelsen «Lær Kidsa Koding». Jeg engasjerte meg i den lokale Kodeklubben i Tromsø, og lærte gjennom denne om programmeringsprogrammet og programmeringsspråket Scratch. Scratch er utarbeidet av Massachusetts Institute of Technology (MIT), og er lagd spesielt med tanke på at barn skal lære seg programmering. En nærmere forklaring av Scratch og Lær Kidsa Koding blir gitt i kapittel 2.0. Da jeg startet i Kodeklubben fikk jeg til stadighet høre flere positive aspekter ved å lære seg programmering. Det som oftest ble trukket frem var at elevene lærte seg ferdigheter innenfor problemløsning. Jeg erfarte selv på kodeklubben at programmeringen hadde potensiale til en ny måte å jobbe problemløsende på. Flere land, blant annet Finland, England og Danmark har, eller er på tur til, å innføre programmering i læreplanen. Dette økte min nysgjerrighet for hva elevene lærte av å programmere, annet enn det å skrive kode.

Digitale ferdigheter er en av de fem grunnleggende ferdighetene elevene skal inneha ved endt skolegang. Når Ludvigsenutvalget også viser til IKT og problemløsning som to av ti viktige kompetanser for fremtiden (NOU, 2014), bestemte jeg meg for å undersøke dette nærmere. I matematikken sies det i digitale ferdigheter som grunnleggende ferdigheter at elevene skal:

«...bruke digitale verktøy til læring gjennom spel, utforsking, visualisering og presentasjon. Det handlar òg om å kjenne til, bruke og vurdere digitale verktøy til berekningar, problemløysing, simulering og modellering».....«Vidare inneber det å bli stadig meir merksam på den nytten digitale verktøy har for læring i matematikkfaget».
(Utdanningsdirektoratet, 2013³)

Jeg ønsker gjennom dette arbeidet å se på hvordan elevene gjennom de digitale ferdighetene kan utvikle evne til problemløsning i matematikken, og om dette påvirker deres læringsstrategier, gjennom programmet Scratch.

I 2003-2004 kom Stortingsmelding 30 *Kultur for læring* ut (Meld. St. 30, 2003-2004). Stortingsmeldingen viste blant annet til at elevene ikke mestrer det «å lære», altså læringsstrategier for å tilegne seg ny kunnskap. I kjølvannet av læreplanreformen: *Kunnskapsløftet 2006* ble det også satt sterkere fokus på vurdering for læring (Dale, 2010). Intensjonen om at skolen og lærerne må bidra til at elevene tilegner seg bedre læringsstrategier er tydelig i planverket. Dette må derfor reflekteres i lærernes didaktiske

overveielser når de planlegger sin undervisning. Trude Slemmen (2011:63) sier at vurdering for læring skal være en planlagt prosess som læreren skal bruke til to formål:

- 1) Tilpasning av undervisning
- 2) Hjelp eleven til å justere sine egne læringsstrategier

Læreren har altså en viktig jobb for å hjelpe elevene på vei mot en ferdighet til å kunne vurdere sin egen læringsprosess for å kunne utvikle seg videre. Vurdering skal altså brukes for å fremme læring, og ikke som en kontroll for hva elevene har lært.

1.2 Avgrensning og problemstilling

Det jeg ønsker å undersøke gjennom dette prosjektet er hvordan, og om programmering er med på å påvirke læringsstrategiene til elevene. Jeg ønsker også å se om programmering kan legge til rette for problemløsning i matematikken ved bruk av digitale verktøy. Jeg ønsker å se om elevene gjennom de digitale ferdighetene kan få erfaring med problemløsning, og undersøke hvordan elevene jobber problemløsende under en programmeringsøkt. Jeg skal ikke se på *hva* de lærer, men på arbeidsprosessen deres og strategiene de benytter seg av. For å ha grunnlag for å si noen om hvordan de jobber problemløsende i programmeringen skal jeg sammenligne dette med måten de jobber i matematikken med de mer tradisjonelle problemløsningsoppgavene.

Jeg har dermed valgt følgende problemstilling:

Hvordan kan programmering i matematikken påvirke læringsstrategiene til elevene?

Jeg har i tillegg valgt følgende forskningsspørsmål:

- *Hvilke problemløsningsstrategier benytter elevene seg av?*
- *Hvordan jobber elevene problemløsende når de programmerer?*
- *Kan programmet Scratch benyttes som en problemløsende oppgave?*

Jeg ønsker å se om elevene bruker noen av de kjente strategiene fra matematikken, som blir presentert og forklart i kapittel 3.4. Etersom programmering ikke er formalisert i den norske skolen ønsker jeg også å undersøke om dette lar seg gjennomføre, og hva elevene synes om dette.

Ulike forskere har undersøkt kvantitativt om elevene blir bedre problemløsere ved å lære seg programmering, og har kommet frem til ulike svar (Fessakis m.fl., 2012,/Taylor m.fl.,

2010./Brown m.fl, u.å/Kalelioglu og Gülbahar, 2014). Jeg skal gjennomføre en kvalitativ aksjon hvor elever skal få lære programmeringsprogrammet og -språket Scratch. I etterkant skal jeg velge ut noen elever og undersøker hvilke problemløsningsstrategier de benytter seg av når de jobber i programmet og sammenligner strategiene når de jobber med matematikkoppgaver. Jeg skal til slutt ha et gruppeintervju med de utvalgte elevene om hva de syntes om programmering, og hvordan de tenkte når de jobbet. Jeg håper at jeg på bakgrunn av dette kan si noe om elevens læringsstrategier påvirkes gjennom programmeringen.

1.3 Disposisjon

Denne oppgaven har følgende tre hoveddeler; teori, metode, analyse og drøfting, i tillegg til innledning og avslutning. Oppgaven starter med en forklaring av programmet Scratch i kapittel 2.0. Dette fungerer også som en inngang til kapittel 3.0, teori, hvor jeg vil reflektere rundt læringsteori opp mot problemløsende arbeid. Videre i teorien presenterer jeg annen sentral teori som benyttes i oppgaven, blant annet om problemløsning og problembasert læring. Etter denne teoretiske bakgrunnen går jeg over til metodedelen i kapittel 4.0 der jeg forklarer og begrunner valg av innsamlingsmetoder, og beskriver design av oppgaven. I kapittel 5.0 presenterer jeg analyse og drøfting av datainnsamlingen. Konklusjon og videre arbeid presenteres i kapittel 6.0.

2.0 Scratch

Som en introduksjon til teorikapittelet vil programmet- og programmeringsspråket Scratch beskrives i dette kapittelet. Det vil også først bli gjort kort redegjørelse for Lær Kidsa Koding og Kodeklubber som har vært en inspirasjon for dette arbeidet.

2.1 Lær Kidsa Koding og Kodeklubber

Bevegelsen Lær Kidsa Koding (LKK) ble stiftet av Simen Sommerfeldt 27. februar 2013. Før dette hadde det vært flere ulike personer som hadde drevet med opplæring av programmering til barn, på andre arrangementer. Sommerfeldt tok initiativ til å samle alle i en egen felles bevegelse (Lær Kidsa Koding, u.å.). I begynnelsen var Kodeklubben, drevet av Helge Astad, og LKK to egne bevegelser, men de fant tidlig ut at de skulle samkjøre og samarbeide. *«Kodeklubben ble det som barna er med på, mens Lær Kidsa Koding er hele bevegelsen med støtte til skoler, og arbeid blant oss voksne»* (Lær Kidsa Koding, u.å.). Bevegelsen har utviklet seg til et stort frivillig nettverk bestående av lokale grupper, privatpersoner, biblioteker, myndigheter, skoler, vitensentre og bedrifter. Bevegelsen er inspirert av ulike internasjonale partnere som *«Computing at school»*, *«Code.org»*, *«Codeclub»* og *«CodeWeek»* (Lær Kidsa Koding, u.å.). Bevegelsene og arbeidet har ført til at programmering har blitt implementert i skolen i flere land. Bevegelsen i Norge startet for å lære barn og unge IT og programmering hvor målet er å øke kunnskapen, øke rekrutteringen til IT yrkene og hjelpe skoler rundt i landet til å implementere programmering. Det finnes derfor et eget lærernettverk innad i LKK som jobber spesifikt med oppgaver som kan brukes i klasserommet.

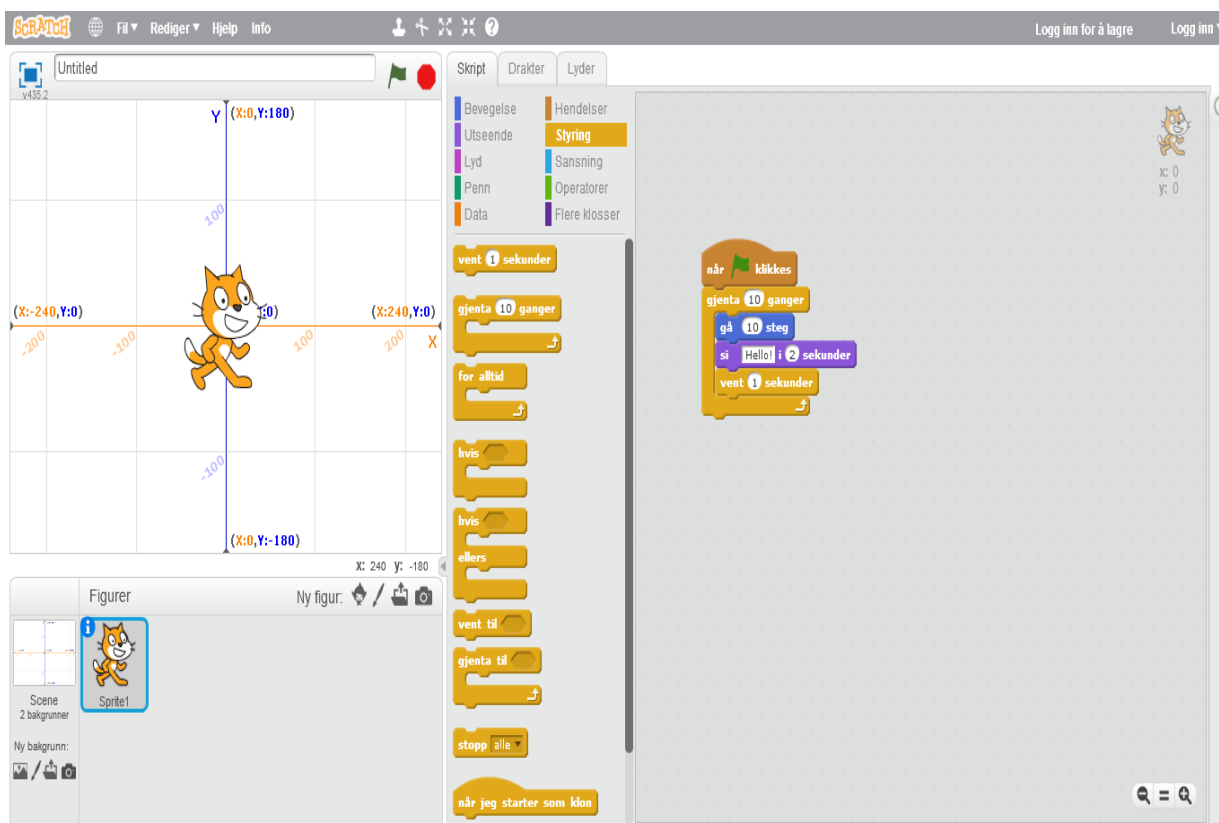
2.2 Scratch

Scratch er et gratis programmeringsspråk for nybegynnere i programmering. Programmet ble utviklet ved MIT Media Lab innenfor gruppen *«Lifelong Kindergarten Group»*. Disse begynte utviklingen av programmet i 2003, og i 2007 ble programmet gjort online etter å kun ha vært en nedlastet versjon. Fra 2007 kjørte programmet i nettleseren og ikke bare som nedlastet versjon. Det er likevel fortsatt mulig å laste det ned. I online versjonen ble det åpnet for å

kunne dele prosjekter med hverandre, og også muligheten for å videreutvikle eksisterende delte prosjekter.

Scratch ble lagd med utgangspunkt i at det ikke fantes en tilsvarende plattform tiltenkt for at barn skal lære programmering. Ideen var å utvikle et program som barn og ungdommer kunne utvikle egne animasjoner, grafikk og spill i (Kafai og Burke, 2014). En forklaring av vanskelige ord og begreper innenfor programmering og Scratch finnes i vedlegg 1, og er lagd som forklaring til elevene i innsamlingen av data.

Scratch har lagt vekt på en visuell innfallsvinkel på programmering med lav begynnerterstel. Programmet har likevel store muligheter. Scratch er bygd opp av brikker som man bruker for å lage koden. Det ligner veldig på å bygge LEGO-klosser, en velger en sammensetning av brikker som blir til en kode, men det må være samsvar mellom brikkene for at koden skal fungere. Instruksjonene må også være nøyaktige instruksjoner. Området der de ulike figurene beveger seg er bygd opp av koordinatsystemet, se figur 1. Dette gjør at en kan bestemme bevegelsene etter koordinater og grader. Visning av koordinatsystemet kan slås av og på.



Figur 1 Bildeutsnitt av programmet Scratch (MIT, u.å.)

For lettere å kunne navigere mellom kodebrikkene og i programmet er alle brikkene er lagt inn under ulike kategorier som har egen fargekode. Da blir det lettere å forstå hva de ulike

brikkene brukes til, og lettere å finne frem til dem. Figur 2 viser de ulike kategoriene og fargekodene med en kort forklaring hva de brukes til. For eksempel er den første kategorien *bevegelse*. Alle disse brikkene har blå farge, og det er de brikkene som bestemmer bevegelsen til figuren en programmerer. Den bytter i tillegg vinkler og retning.

Kategori	Forklaring av kategori	Kategori	Forklaring av kategori
Bevegelse	Beveger figurene og bytter vinkler og retning	Hendelser	Ulike brikker som får programmet til å starte.
Utseende	Kontrollerer utseende, drakter og tale	Styring	Brikker med «hvis» - og løkker.
Lyd	Spiller lydfiler	Sansing	Bestemmer hva som skjer hvis figuren er borti noe eller en farge
Penn	Tillater figuren å tegne (slik som LOGO-programmeringen er kjent for)	Operatorer	Matematiske og logiske operatorer
Data	Lager variabler, og bruk av disse	Flere klosser	Lager klosser og bruk av disse og flere tilleggsfunksjoner

Figur 2 Forklaring av ulike kategorier i Scratch (Wikipedia, 2015, egen oversettelse)

I figur 1 vises de ulike kodebrikkene til kategorien *styring*. Her ser man at kodebrikkene har løkker. Til venstre i figur 1, på *scriptområdet* ser en et eksempel på en kode. Når man skal bygge en kode velger en kodebrikker fra de ulike kategoriene. Disse kan kombineres som en vil, såfremt de passer med hverandre. Som i eksempelkoden fra figur 1 ser man en *Hendelse*-brikke kombinert med *bevegelse* og *utseende* som er lagt inni en løkke fra *styring*. På denne måten kan en lage mange ulike program gjennom programmering i Scratch.

3.0 Teorikapittel

“I think everybody in this country should learn how to program a computer because it teaches you how to think” (Steve Jobs).

I dette kapitlet blir sentral teori presentert. Sitatet ovenfor speiler tanken bak intensjon til Lær Kidsa Koding, og er et ofte brukt sitat i miljøet. Kapitlet starter med å reflektere rundt læringsteori, og viser også hvordan datamaskinen har utviklet seg gjennom de ulike paradigmen. Videre blir problembasert læring og problemløsning i matematikk presentert. Til slutt kommer en forklaring om programmering og hvordan en kan jobbe problemløsende i programmering.

3.1 Perspektiv på læring

Læring kan beskrives og forstås på ulike måter, men kan defineres som en erfaringsbasert prosess som varer gjennom hele livet (Befring, 2007). Mennesket endres og utvikles på en måte som påvirker adferd, kunnskap og holdninger. Denne læringsprosessen påvirker hvem vi blir, og kommer som et resultat både av planlagt og målrettet læringsinnsats i barnehage og skole, og de opplevelser og erfaringer vi får på andre arenaer. Skolen må ta høyde for at barn har et liv utenfor skolen som påvirker deres muligheter til å lykkes i skolen. En læringsteori sier noe om hvordan en oppfatter at mennesker tilegner seg læring, og kan hjelpe lærere til å forstå hvordan elever tenker. På denne måten kan lærere støtte elevene underveis i deres læringsprosess. Støtten læreren skal gi underveis ligger også i vurderingsforskriftene under § 3.2: *Undervegsvurdering skal brukast som ein reiskap i læreprosessen, som grunnlag for tilpassa opplæring og bidra til at eleven, lærlingen eller lære kandidaten aukar kompetansen sin i fag* (Lovdata, 2009).

Pedagogikken og synet på læring har endret seg gjennom historien ettersom flere har forsøkt å forklare læring som fenomen. Lenge var det behavioristiske teorier med stimuli og respons som rådet. På midten av 1900-tallet kom kognitivismen som en reaksjon på behaviorismen. Kognitivismen satte fokus på at det er eleven selv som konstruerer kunnskap og forståelse (Jensen og Aas, 2011). Jean Piaget er en viktig bidragsyter innenfor kognitivismen og konstruktivismen, og skiller mellom figurativ og operasjonell kunnskap. Den figurative kunnskapen er faktakunnskaper, mens den operasjonelle kunnskapen er prosessen hvor

elevene må assimilere eller akkomodere kunnskap (Jensen og Aas, 2011). Kognitivismen og konstruktivismen legger vekt på at kunnskapen konstrueres i det enkelte individ. De siste årene er det de sosiokulturelle læringsteoriene som har rådet i skolen og skoleforskningen. Det kan forklares ved at samfunnet har blitt mer komplekst og mangfoldig, og teorier om læring har flere aspekter ved seg enn hva de kognitive teoriene fremstiller (Krumsvik og Sjöblom, 2013).

Den sosiale konstruktivismen går ut på at mennesket konstruerer våre begreper i et sosialt og kulturelt miljø. Denne læringsteorien er en videreføring av konstruktivismen, og kom som en reaksjon på at den kognitive konstruktivismen ikke tar i betraktning at læring også foregår i et sosialt miljø. Konstruktivismen ser ikke på læring som en påfyllingsprosess, men at barnet selv må konstruere forståelsen sin gjennom læring og erkjennelse (Illeris, 2012:55). Den sosiale konstruktivismen får frem at læring skjer i lys av en kultur, og dermed kan forståelsen av, for eksempel, et begrep variere fra kultur til kultur. Sosial konstruktivisme skiller seg fra Piagets kognitive teorier ved at den vektlegger språk og sosial samhandling for å konstruere kunnskap, og dette må være satt i en kontekst. Lev Vygotsky er en viktig bidragsyter til sosial konstruktivisme og den sosiokulturelle læringsteorien. Han trekker frem språket som det viktigste medierende redskapet til læring (Krumsvik og Sjöblom, 2013). Gjennom språket får elevene uttrykt sine ideer, stilt spørsmål og diskutert. Som lærer vil en slik forståelse innebære å tilrettelegge for at elevene får samarbeide og samtale om det som skal læres før de går i gang med oppgaveløsning.

Imsen (2012) viser til at Vygotsky legger vekt på at læreren er en viktig medierende hjelper i elevenes læringsprosess. Læreren skal fungere som en støtte i elevenes arbeid, og hjelpe dem til å utnytte sin proksimale utviklingszone. Proksimal utviklingszone handler om den nærmeste utviklingssonen eller det rommet for læring som åpnes opp når eleven får hjelp til å strekke seg ut over det som elevene kan og klarer i utgangspunktet. Tilpasset opplæring må forstås som et samspill mellom individ og fellesskap. Medelever kan bidra til hverandres læring gjennom å dele kunnskap og/eller oppmuntre hverandre til å prøve seg på noe nytt.

I vurdering for læring skal læreren fungere som en «coach» for elevene gjennom å gi tilbakemeldinger, hjelpe til med teknikker og stille motiverende spørsmål som fremmer refleksjon (Slemmen, 2011). I kjølvannet av læreplanreformen, *Kunnskapsløftet 2006* ble det satt et sterkere fokus på vurdering forstått både som underveis- og sluttvurdering. Sammenhengen mellom læringsprosess og resultat ble understreket i de nye vurderingsforskriftene. I vurdering for læring ligger det at elevene selv skal være

medvirkende i egen læringsprosess. Gjennom programmet Scratch får elevene forhåpentligvis øvd seg på å vurdere sin læringsprosess, og læreren blir en viktig støtte og «coach» i denne prosessen.

Det konstruktivistiske perspektivet vil altså legge vekt på prosessen frem til et svar, og ikke bare svaret i seg selv. Det blir viktig at elevene opplever faget som meningsfullt og at de får metoder og strategier de kan benytte seg av når de skal løse oppgaver. Som lærer må man også være opptatt av kvaliteten på læringsmiljøet og relasjonene som de sosiokulturelle læringsteoriene bringer inn. Slik sett vil det sosiokulturelle inkludere det individuelle og det sosiale perspektivet på læring, og det er et slikt bredt perspektiv læreplanverket for grunnskolen har til grunn på læring i skolen.

3.1.1 Dataprogrammenes utvikling gjennom paradigmen

På samme måte som pedagogikken har endret seg gjennom ulike paradigmer, har også teknologien og dataprogrammene gått gjennom flere paradigmer på kort tid, fra å gå fra et behavioristisk til et mer sosiokulturelt syn på læring (Koschmann, 1996). Når det kom dataprogrammer lagd for skolen og læring på slutten av 1990-tallet, gikk de tilbake i tid i paradigmen. Programmene som var lagd til skolebruk var veldig behavioristiske med fokus på ett rett svar og direkte tilbakemelding. Oppgavene som ble presentert var veldig lite problemløsende. Etter hvert som teknologien utviklet seg litt ble det flere interaktive program som fokuserte på kognitiv læring, men som likevel var bygd opp med oppgaver og direkte tilbakemelding (Koschmann, 1996). I dag ser en at programmer som blir lagd for læring er mer fokusert på et sosialt konstruktivistisk læringssyn, selv om en fortsatt finner nettsider fra læreverk med oppgaver som er veldig behavioristisk oppbygd. Programmet Scratch er et eksempel på et program som bygger på et mer sosialkonstruktivistisk læringssyn. Teknologien og dataprogrammene har altså gått gjennom samme utvikling i de pedagogiske paradigmen. Når de pedagogiske programmene og nettsidene kom, hoppet de tilbake i tid, og dermed ble skolens bruk av datamaskiner og IKT hengende etter den gjeldende pedagogiske tenkningen som befant seg i skolen.

3.2 Problembasert læring

Å lære gjennom problembasert læring vil si å gå fra læreraktivitet til elevaktivitet, og å få mer fokus på læring kontra undervisning (Bjørke, 2000). En av de mest signifikante fordelene med problembasert læring er at denne arbeidsmåten fokuserer på flere muligheter for løsning, fremfor å fokusere på et hva som er rett svar. Dette kan hjelpe elevene til å mestre på nye områder som kanskje ikke er tilgjengelig gjennom de mer tradisjonelle arbeidsmåtene (Lambros, 2002:5). Grunntanken i problembasert læring er at «*læring gjennom problemløysing er meir effektiv for å utvikle brukbar kunnskap ein lese-, hugse-, gjenngi-modell...*» (Bjørke, 2000:22). En kan summere problembasert læring i tre hovedprinsipp (Pettersen, 2005:65):

1. Problembasering
2. Selvstyring og selvregulering
3. Gruppebasering

Gjennom disse tre hovedprinsippene ligger det at elevene skal utvikle problemløsningsstrategier og sin evne til problemløsning. Elevene må også være medvirkende i sin egen læringsprosess ved å vurdere og å reflektere rundt læringsprosessen gjennom selvstyring og selvregulering. Dette betyr ikke at læreren får frigjort tid ved at elevenes medvirkning blir større, læreren får en viktig jobb med å legge opp undervisning som underbygger dette, og for å støtte elevene i deres arbeid. Til slutt er prinsippet om at elevene skal arbeide i grupper slik at den skal fungere som en støtte for hverandres læringsarbeid. Gjennom gruppen får også elevene diskutert og reflektert sammen over løsningsstrategier. For å få dette til å fungere er det viktig at en som lærer også jobber med et godt klassemiljø. Problembasert læring skal altså gi elevene mulighet til selvstendig og gruppebasert arbeid og lære seg strategier for løsning som gjør at de er rustet for fremtiden og dens utfordringer (Bjørke, 2000:30).

Opgaver med flere innfallsvinkler er anbefalt for elever fra fjerde til sjette trinn, og dermed er problembasert jobbing å etterstrebe for denne aldersgruppen (Lambros, 2002:45). Det kan også være med på å holde på interessen ved at elevene får være med å utforske ulike løsningsforslag. Hensikten er at elevene skal styrke sammenhengen og kunnskapsoverføringen mellom teori og praksis. Mange lærere føler likevel at læreplanen ikke gir tid til dybdestudier (Hattie og Yates, 2014:80). Elevene trenger tid for å kunne knytte sammen teori med praksis og finne forbindelser mellom dem. «*Problembasert læring*

representerer et pedagogisk konsept for å utvikle læringsmiljøer som både fremmer og støtter dybdelæring» (Pettersen, 2005:49). En skiller mellom dybdelæring og overflatelæring i pedagogikken, hvor den ene ikke blir sett på som bedre enn den andre. De er begge nødvendige i en læringsprosess, men for å få en meningsfull, relevant og varig læring må en ha utstrakt bruk av dybdelæring.

3.3 Problemløsning

I matematikken har problemløsning fått mye oppmerksomhet de siste 20 årene. Tanken bak å innføre problemløsning i undervisningen er at matematikk ikke bare skal handle om ferdighetstrening, regler og regning av algoritmer. Undervisningen skal også hjelpe til å utvikle verktøy som kan hjelpe elever til å løse ulike oppgaver (Löwing og Kilborn, 2002:241). Innenfor matematikken jobber en problemløsende gjennom oppgaver som er såkalte matematiske problem. Et matematisk problem er når elevene ikke kan løse oppgaven med rutineferdigheter (Alseth, 1998). Problemet trenger en matematisk undersøkelse for å kunne besvares. Dette betyr også at et matematisk problem er relativt i forhold til hvem som løser det, ettersom en har ulik kompetanse. For eksempel vil et multiplikasjonsstykke med tosifrede tall være en problemløsningsoppgave for en elev på småtrinnet, mens det vil være en rutineoppgave for en elev på 7. trinn (Røsseland, 2005). Å undervise gjennom problemløsning betyr at elevene møter på oppgaver som utfordrer dem til å lære gjennom mer virkelighetsnære kontekster, problemer, situasjoner og modeller (Van De Walle m.fl., 2010).

Hensikten med at elevene skal jobbe med problemløsningsoppgaver er at de skal bli inspirert til arbeid og for å sette i gang tankeprosesser (Alseth og Røsseland, 2014). Det er ikke meningen at elevene skal lære seg ferdigheter i problemløsning kun for ferdighetens skyld, men at elevene gjennom disse ferdighetene får tatt i bruk sin matematiske kompetanse. En god problemløsningsoppgave setter søkelyset på utforskning og tenkning rundt prosessen frem til et svar. Problemløsning er ikke en ny arbeidsmetode i matematikken. I formålet til læreplanen i matematikken for grunnskolen står det:

«Matematisk kompetanse inneber å bruke problemløysing og modellering til å analysere og omforme eit problem til matematisk form, løyse det og vurdere kor gyldig løysinga er. Dette har òg språklege aspekt, som det å formidle, samtale om og resonnerer omkring idear.» (Utdanningsdirektoratet, 2013¹)

Problemløsning kan læres på tre måter, *for*, *om* eller *gjennom* (Van De Walle m.fl., 2010) Å lære *for* problemløsning er å lære kunnskapen og algoritmen først, før en begynner med problemløsningsoppgaver. Å lære *om* problemløsning er å lære seg hvordan en løser en problemløsningsoppgave, altså om prosessen og om strategier i problemløsning. Å lære *gjennom* problemløsning betyr at elevene lærer matematikk gjennom virkelighetsnære oppgaver, modeller, situasjoner og kontekster. Det er denne siste metoden som er aktuell i dagens læreplan.

3.4 Hva kjennetegner en problemløser?

Bjorklund (1990) skiller på to hovedgrupper av problemløsningsstrategier, «Forward search» og «problem reduction». Disse to hovedgruppene er ment for mer generelle problemer, og ikke spesifikt til matematikken, selv om de kan overføres til den. I «forward search» går strategien ut på å prøve ulike metoder eller fremgangsmåter for å komme frem til et svar på. Denne blir sett på som anstrengende og langsom. «Problem reduction» går ut på å lage seg delmål, og dele opp problemet slik at en klarer å finne en løsning. En må da se på hvor en står nå, og hva som må til for å komme frem til svaret (Bjorklund, 1990:25-26).

George Polya (1957) har definert fire steg for å løse problemløsningsoppgaver: *forstå problemet, legge en plan, gjennomføre planen og se tilbake og reflektere*. Disse stegene er ikke en lineær prosess. En må bevege seg frem og tilbake mellom stegene når en jobber med en problemløsende oppgave. Steg én omhandler det å forstå hva problemet er og innebærer. Elevene må forstå ord og begreper som brukes i oppgaven, og bør kunne peke ut hovedprinsippene i problemoppgaven. Gjennom dette steget er det viktig at elevene får aktivert forkunnskap på temaet de skal jobbe med. Her har læreren en viktig jobb med å stille spørsmål for å forsikre at elevene har forstått oppgaven. Steg to er å legge en plan for hvordan å komme frem til en løsning. Her må elevene vite hvilke utregninger, hvilke beregninger og sammensetninger en må gjøre for å komme frem til løsningen. Når de vet dette har de lagd seg en plan (Polya, 1957). Det er i dette steget elevene velger seg en strategi de skal benytte seg av for å finne en løsning, og det er viktig at elevene lager en plan som bygger på forståelse. Dette steget kan være utfordrende, og av og til må man prøve og feile før man får en ide. Som lærer bør en stille spørsmål som får elevene til å tenke på lignende problemer, eller identifisere viktige deler av oppgaven. Steg tre er å gjennomføre planen en lager i steg to. Det viktige i dette steget er å sjekke at en gjør rett, og at en ikke glemmer planen (Polya,

1957). Hvis elevene har lagd seg en god plan er lærerens jobb liten i dette steget. Likevel bør læreren stille spørsmål som får elevene til å sjekke at de ikke gjør noen feil. Det fjerde og siste steget er å se tilbake og reflektere rundt de stegene og arbeidet en har gått gjennom. Mange elever lukker boken eller går til neste oppgave med en gang de har funnet en løsning, men når de gjør dette mister de det Polya identifiserer som det viktigste steget (1957:15). Ved å reflektere over løsningen og veien til den, vil elevene kunne utvikle deres kunnskap og ferdighet til å løse problemoppgaver.

Ved å lære disse fire stegene vil elevene forbedre sin evne til å jobbe med problemløsningsoppgaver, ettersom de må vurdere og reflektere rundt deres valg og svar. Disse stegene gjelder også for mange ulike type problemløsningsoppgaver, fra de enkle problemene til de lengre og vanskeligere flersteps tekstoppgaver.

For at en problemløser skal lykkes med problemløsningsoppgaver er det fire kategorier som spiller inn. Disse er: 1) Forkunnskapen til problemløseren, 2) problemløserens kunnskap om problemløsningsstrategier, 3) problemløserens evne til å kontrollere og endre, og 4) problemløserens selvtillit rundt problemløsning og matematikk (Schoenfeld, 2013:11, egen oversettelse). Når elevene skal løse en problemløsningsoppgave kan de bruke ulike strategier for å komme frem til en løsning, eller de kan prøve og feile (Røsseland, 2005). Men elever som har høy problemløsningskompetanse vil også kunne velge mellom flere strategier, og kunne velge en god strategi for å løse problemer. Elevene står helt fritt til å velge en strategi de kan benytte seg av. Van De Walle m.fl. (2010:43) viser til noen vanlige strategier som er identifisert fra elever i grunnskolen:

- *Å tegne et bilde, «spille» det ut, eller bruke en modell:* elevene gjør ulike aktiviteter for å synliggjøre hva problemet er og hvordan en kan løse det.
- *Se etter mønster:* denne er spesielt viktig innenfor algebraiske oppgaver, og spiller en viktig rolle i det å lære elevene viktige ferdigheter.
- *Gjett og sjekk:* Strategien går ut på å forsøke noe, og se om du kan lære noe av forsøket.
- *Lage en tabell eller diagram:* tabeller eller diagrammer kan være med på hjelpe til å analysere og forme nye ideer.
- *Prøve ut en enklere form av problemet:* Det kan være en god ide å undersøke om en klarer å løse en enklere form av problemet, og overføre noe av den slik at man enklere kan analysere og forstå, og dermed klare oppgaven.

- *Lage en organisert liste:* Her vil en systematisk gå gjennom mulige utfall og kan vise et tall på muligheter eller bekrefte at alle muligheter er inkludert.
- *Skrive en formel:* Noen oppgaver kan settes inn i en formel med tall og symboler som gjør oppgaven lettere å løse.

Disse ulike metodene blir ikke brukt på alle trinn, for eksempel kan en elev på småtrinnet ha store vansker med å skrive en formel for å løse problemet. Elevene på barnetrinnet skal likevel helst ha utviklet noen problemløsningsstrategier, og ha en begynnende evne til å velge blant dem. Det er også viktig at en ikke forteller hvilken strategi de skal bruke, men heller trekke frem egnede strategier i oppsummeringen (Van De Walle m.fl., 2010). Når elevene finner en god strategi bør den trekkes frem og diskuteres i plenum. Problemløsningsstrategiene skal være til hjelp for å sortere og oppdage hvilke matematiske objekter og forhold som gir mening (Schoenfeld, 2013).

Det er flere aspekter som skiller en ekspert fra en nybegynner i problemløsning. Blant annet er evnen til å bevege seg gjennom de ulike stegene til Polya, og det å bruke disse aktivt et klart skille. Gode problemløsere bruker mye tid på det første steget til Polya, og stiller seg selv nøkkelspørsmål for å oppdage og forstå kjent og ukjent informasjon i oppgaven. De overvåker sin egen prosess, og retter på egne feil gjennom regningen (Chapin m.fl, 2014:97). Schoenfeld (2013) sier i tillegg at gode problemløsere har utholdenhet og endrer vei i samsvar med kontrollen av problemløsningsprosessen deres. På den andre siden vil en nybegynner haste seg gjennom de første stegene og starte rett på oppgaven uten en særlig plan. En nybegynner i problemløsning vil også mest sannsynlig bruke en undersøkelse hvor de har fokus på målet og jobber seg baklengs gjennom oppgaven (Sweller, 1988). En ekspert derimot vil starte med å velge seg en strategi og som gjennom denne vil klare å finne de ulike elementene oppgaven ber om. Hazzan m.fl. (2014:76) sier også at hvis ikke elevene lærer seg noen formelle instruksjoner på hvordan de skal løse en problemløsningsoppgave, vil selv den mest oppfinnsomme eleven benytte seg av en uproduktiv «gjett og sjekk» strategi.

3.5 Programmering

Programmering defineres som den prosessen med å skrive en kode som får et program eller en maskin til å fungere slik en vil. *«Programmering går ut på å sette opp en serie instruksjoner som styrer maskinen og som avgjør hvordan den skal reagere på inndata, inntastinger,*

musebevegelser og annet» (Rossen, 2009) Programmering er en ferdighet som er vanskelig å lære seg, men ved hjelp av rette strategier og verktøy kan barn lære seg konsepter og grunnleggende programmering (Saeli m.fl., 2010). Programmering i skolen har fått mye oppmerksomhet i det siste, men det er ikke første gang en snakker om læring gjennom programmering i et pedagogisk syn. Seymour Papert, som er en matematiker og programmerer, gjorde allerede på 70-tallet undersøkelser om hva elever på barnetrinnet lærte gjennom programmet han selv har vært med å lage – LOGO (Papert, 1980). Scratch innehar mange av de samme prinsippene som LOGO, og er inspirert fra dette arbeidet.

Kafai og Burke (2014) tar opp problemstillingen hvorfor barn bør lære programmering. Hovedfokuset deres ligger ikke på å lære kun programmering, men det å lære *«computational thinking»* gjennom programmeringen: *«needs to be thought of as computational participation because the computer programs that are being created, used, repurposed, and shared have become our social connections»* (Kafai og Burke, 2014:9). Når en skriver en kode, kommuniserer en ikke bare med datamaskinen, en kommuniserer med hele verden. Målet med å lære barn programmering er ikke at alle skal bli programmerere, men som Kafai og Burke (2014:135) sier det: *«we all need to understand code to examine digital designs and decisions constructively, creatively, and critically»*. Gjennom å lære seg programmering kan elevene utvikle kognitive ferdigheter, og også kulturelle og sosiale aspekter som de benytter i samarbeidet med andre. Gjennom programmering i Scratch får også elevene prosessert deres matematiske aktivitet gjennom et visuelt verktøy, selv om de ikke øver på et spesifikt matematisk emne (Taylor m.fl., 2010).

3.5.1 Problemløsning i programmering

I programmering er det mye problemløsning. Som i alle andre emner må en starte med å forstå oppgaven og problemet en skal løse, før en lager seg en plan. Når en løser en problemløsningsoppgave vil en gjøre det stegvis (som en algoritme) som til slutt løser problemet (Hazzan m.fl., 2014). I programmering blir det litt annerledes. Stegene vil ofte bli selve koden. *«The difficult stages, however, lie in between: how to move from the requirement understanding to the problem solution»* (Hazzan m.fl., 2014:77). En jobber problemløsende når man programmerer fordi man først må finne en løsning på problemet, for å så reflektere hvordan man skal gjøre om dette til en kode som datamaskinen forstår og klarer å gjennomføre (Papert, 1980). Det ligger muligheter i programmeringen for å lære seg gode

strategier for problemløsning, kreativitet og strategier for tenkning (Saeli m.fl., 2010). Programmering er et godt verktøy for å hjelpe elevene til å analysere, organisere, uttrykke og evaluere deres tanker i en klar og konsis måte gjennom problemløsningsprosessen (Fessakis m.fl., 2012). Dermed kan også et programmeringsmiljø være en god arena for å drive underveisvurdering, og der elevene selv er delaktige i prosessen.

Videre sier Hazzan m.fl. (2014) at en viktig del av å programmere er å kunne «debugge», altså drive med feilsøking og sjekke korrektheten, og deretter rette på eventuelle feil i koden. Det å sjekke koden, og ikke se på feilene som en negativ ting men heller lære av dette, vil alle som programmerer gjøre. Papert (1980) trakk nettopp dette frem, at det å lære elevene å «debugge» kan være med på å endre elevenes negative holdninger til feil.

The process of debugging is a normal part of the process of understanding a program. The programmer is encouraged to study the bug rather than forget the error. And in the Turtle context there is a good reason to study the bug. It will pay off. (Papert, 1980:61)

Ved å fokusere på at en feil er noe man kan lære av, har man en mulighet til å jobbe med holdningen til elevene rundt feil. Elevene må hele tiden drive med feilsøking og vurdere seg selv og sin læringsprosess.

3.6 Grunnleggende ferdigheter

Et av det mest avgjørende kjennetegnet av læreplanen *Kunnskapsløftet 2006* er prinsippet om å innføre og integrere fem grunnleggende ferdigheter i alle fag. Dette kan ses på som en nøkkel til å forstå overgangen fra industri- til kunnskapssamfunn som vi betegner dagens samfunn som (Dale, 2010:127). Det kan ses på som en nøkkel fordi det dannes en tanke om et integrert menneske som har med seg ferdigheter som er en forutsetning for å kunne lære og utvikle seg videre. De fem grunnleggende ferdigheter er digitale ferdigheter, muntlige ferdigheter, å kunne lese, å kunne regne og å kunne skrive (Utdanningsdirektoratet, 2012). De grunnleggende ferdighetene skal benyttes i alle fag, og brukes som hjelp til å oppnå kompetansemålene i de enkelte fagene, og på fagets premisser (Dale, 2010).

Regning som grunnleggende ferdigheter innebærer *å kunne anvende matematikk i ulike fag når det er relevant og på de ulike fagenes premisser* (Matematikksenteret, 2014). Det å bruke regning i alle fag vil blant annet si å lese av tabeller, regne forhold og bruke ulike metoder for

å løse problemer og vurdere gyldigheten av svaret de kommer frem til. Gjennom å benytte seg av regning som grunnleggende ferdighet skal altså elevene også få mer erfaring med problemløsning, gjennom å jobbe mot en helhetlig kompetanse rundt problemløsning.

Digitale ferdigheter som grunnleggende ferdigheter vil si å bruke digitale verktøy til og blant annet hente inn informasjon, løse praktiske oppgaver og skape digitale produkter. Ettersom den digitale utviklingen har vært stor de siste årene, har de også vært med på å utvikle og endre mange av læringsprinsippene, og *«derfor er digitale ferdigheter en naturlig del av grunnlaget for læringsarbeid både i og på tvers av faglige emner. Dette gir muligheter for nye læringsstrategier, men stiller også økte krav til dømmekraft»* (Utdanningsforbundet, 2012²). Som vist i innledningen skal de digitale ferdighetene som grunnleggende ferdighet i matematikkfaget spille en viktig rolle for å jobbe problemløsende i matematikkfaget.

Gjennom de digitale ferdighetene- og regning som grunnleggende ferdighet har man som lærer en mulighet til å jobbe mot at elevene skal utvikle seg problemløsende, både med tanke på strategier og arbeidsmåter, og ikke minst om læringsstrategier.

4.0 Metodekapittel

I dette kapitlet beskrives, reflekteres og begrunnes metodevalgene som er gjort for å samle inn datamaterialet. De er samlet inn gjennom videoopptak, lydopptak og feltnotater. Innsamlingen deles i fire ulike steg:

1. Introduksjon
2. Oppgaver i Scratch
3. Oppgaver i matematikk
4. Gruppeintervju

Første steg er en introduksjonstime som ble gjennomført i full klasse med 18 elever, hvor elevene lærte seg programmeringsprogrammet og språket Scratch. Jeg gjorde så et utvalg på fire elever for de neste stegene. Steg to jobbet elevene med oppgaver i det samme programmet. I steg tre jobbet elevene med matematikkoppgaver, og i fjerde og siste steg gjennomførte jeg et gruppeintervju med elevene. Jeg skal beskrive hvordan jeg har gjort utvalget av informantene, og hvordan jeg har gått frem for å samle inn data.

4.1 Utvalg

4.1.1 Utvalg av klasse

Når jeg skulle finne en klasse som jeg kunne gjennomføre forskningen i, sendte jeg mail til flere rektorer i Tromsøsskolene. Der presenterte jeg prosjektet og ba om å bli satt i kontakt med en lærer fra mellomtrinnet som kunne stille sin klasse disponibelt for et slikt opplegg. For at elevene skulle kunne si noe selv om hva de har lært og hva de syntes, ønsket jeg elever fra mellomtrinnet. En rektor ved en byskole satte meg i kontakt med en lærer for 6. trinn som var behjelpelig med å få gjennomført dette prosjektet. Etter et møte med læreren var datoer for gjennomføring satt.

4.1.2 Utvalg av elever

Jeg har valgt kun å ha fire elever til steg to til fire. Grunnen til dette er for å kunne gå i dybden hos noen elever, og klare å holde meg innen for rammene av oppgaven. Jeg vil ikke kunne generalisere noe fra prosjektet, men jeg vil kunne undersøke og ende opp med noen nye og mer presise problemstillinger som vil være grunnlag for videre utforskning av temaet. Utvalget ble gjort etter introduksjonstimen. Jeg ønsket å ha to jenter og to gutter slik at det ikke skulle bli en kjønnsdominans i utvalget. Kriteriene for utvalget var elever som virket motiverte, og som var middels til sterke elever i matematikk. Jeg ønsket dette slik at det ikke skulle være noen faktorer om faglig svakhet eller kjønnsforskjell jeg trengte å ta hensyn til i et så lite utvalg. I løpet av introduksjonstimen var det seks til syv elever som utmerket seg ved å være interessert, stille undrende spørsmål og virket motivert til arbeidet. I etterkant av timen samsnakk jeg med lærer, og valgte ut fire av disse elevene. Jeg har valgt å gi elevene fiktive navn for å holde deres anonymitet. Elevene blir videre henvist til som Hilde, Fred, Steinar og Bente. En nærmere beskrivelse av elevene blir presentert i kapittel 4.5.

4.2 Kvalitativ metode

For å få svar på mine forskningsspørsmål har jeg valgt å ha en kvalitativ metodologisk tilnærming. Et kvalitativt studie kjennetegnes ved at det er større fleksibilitet enn ved et kvantitativt studie. Gjennom en kvalitativ tilnærming ønsker en å gå i dybden på noen få informanter. Metoden egner seg når en skal utforske noe nytt eller noe har lite kjennskap til fra før av (Christoffersen og Johannesen, 2012). Som jeg presenterer i min problemstilling og mine forskningsspørsmål ønsker jeg ved denne oppgaven å undersøke hvordan læringsstrategiene til elevene blir påvirket, og hvilke problemløsningsstrategier elevene bruker når de løser Scratchoppgaver. For å kunne si noe om dette må jeg studere de i en programmeringssituasjon hvor jeg har fokus på å observere elevenes bruk av strategier og arbeidsmåter. Jeg vil også sammenligne dette med strategiene og arbeidsmåtene de benytter seg av ved løsning av matematikkoppgaver. Jeg skal undersøke dette ved å gjennomføre en økt hvor elevene blir introdusert til programmeringen og programmeringsspråket Scratch. I etterkant skal utvalgte elever jobbe med nye oppgaver i programmet, og deretter problemløsningsoppgaver i matematikk. For å kunne beskrive dette rikere ønsker jeg å intervjuer elevene slik at jeg kan ha med elevenes syn. Det er denne konteksten jeg er ute etter

å beskrive i dette arbeidet. Disse beskrivelsene og observasjonene ville jeg ikke kunne gjort gjennom et kvantitativt studie. Jeg kunne gjennomført en før- og ettertest på elevenes problemløsningskompetanse for å se om det har blitt bedre problemløsere etter de har lært seg programmering, men på denne måten hadde jeg ikke fått en dypere innsikt. Det er rike beskrivelser av elevenes jobbing og fra elevene selv som jeg ønsker å få tak i. «*Kvalitativ forskning studerer altså først og fremst den autentiske konteksten og/eller korleis informantar ser på dette*» (Krumsvik, 2014:21). Etersom jeg skal studere denne komplekse praksiskonteksten vil det være gunstig for meg å nytte meg av en kvalitativ metode.

4.3 Arbeidets forskningsdesign

Måten jeg har valgt å gå frem for å samle inn data kan sies å være innenfor rammen av aksjonsforskning ettersom jeg gjør en innovasjon. Jeg ønsker å komme med forslag til en endring eller variasjon i praksis. Aksjonsforskning kjennetegnes ved at det tar tak i et problem som erfares i praksis. Aksjonen kartlegger så problemets områder og prøver deretter en løsning som er arbeidet frem gjennom samarbeid, gjensidig læring og felles kompetanseutvikling (Brekke & Tiller, 2013). Men ettersom det er jeg som både planlegger og gjennomfører aksjonen, uten det nære samarbeid med læreren, er jeg ikke innenfor rammene av det klassiske aksjonsforskningsarbeidet hvor det er et forskende partnerskap. Jeg står alene i dette forskningsarbeidet, og vil ikke ha mulighet til en felles refleksjon rundt et opplegg slik som aksjonsforskning legger opp til.

Jeg ser også at det er fellestrekk til et casestudie ved at jeg studerer en gruppe som jeg henter inn mye informasjon over en kortere periode (Christoffersen og Johannessen, 2012). Men også her vil min nærhet til feltet være et problem. Jeg skal som lærerstudent ut å forske i egen praksis (ved at jeg fungerer som elevenes lærer de aktuelle timene). Wadel (2006:116) sier noe om forskning i eget feltarbeid: «*Det særskilte kan sies å være at en ikke bare er deltaker i sin egen forskningsprosess, men også at forskningsprosessen er basert på en selv som delaktig*». Min forskning baserer seg på at jeg som forsker gjennomfører en time, og gir ulike oppgaver til elevene. Etersom opplegget baserer seg på et program en må ha kjennskap til, ville det vært vanskelig å få læreren til denne klassen til å gjennomføre timen uten å ha gjennomgått programmet grundig med henne. Jeg er ute etter å endre praksis og kommer med et forslag til endringen. Som nevnt tidligere vil jeg også undersøke om det opplegget lar seg gjennomføre og derfor må jeg studere meg selv som lærer. Dermed har jeg noen fellestrekk

med feltarbeidet. Cato Wadel (2006) betegner feltarbeid som et forskningsarbeid der forskerne er ute i feltet og som oftest studerer folk i sine naturlige omgivelser. Forskning i egne erfaringer derimot ser en på seg selv i roller i forhold til andre, og forskeren selv er en sentral aktør (Wadel, 2006:114). Deltakende observasjon er den vanligste måten å samle inn data på.

Forskning innenfor humaniora skiller på tre ulike typer design; eksplorativt, deskriptivt og kausalt. Det eksplorativt designet har som mål å utforske, det deskriptive å beskrive og kausale ser på årsak og virkning. Jeg har et eksplorativt design ettersom jeg ikke har funnet noe tidligere forskning på dette feltet.

Det jeg ser er at i dette prosjektet er at jeg er nødt til å benytte meg av en miks av metoder, ettersom mitt design må fange dobbeltrollen jeg har som en forskende lærerstudent. Kanskje det vil være behov for en ny type forskningsdesign nå som en masterbasert lærerutdanning, og dermed krav om en forskende lærerstudent, trer frem – og etter hvert en forskende lærer som aktivt driver med forsknings- og utviklingsarbeid i egen klasse. Jeg velger å kalle mitt forskningsarbeid for et for et eksplorativt aksjonsforskningsprosjekt, hvor forskning på egne erfaringer er sentrale.

4.4 Datainnsamlingsmetode

Metoden jeg skal benytte meg av i studiet for å samle inn data er observasjon og intervju. Gjennom alle øktene jeg har gjennomført med elevene opptrer jeg både som lærer og som forsker, og derfor har jeg valgt å bruke videoopptak for å kunne sikre meg tilstrekkelig og god empiri. Dobbeltrollen som lærer og forsker må jeg være bevisst, og jeg skal etterstrebe det å skaffe meg en avstand fra datamaterialet slik at analysen ikke blir påvirket av dette (Postholm og Moen, 2011).

4.4.1 Observasjon med videoobservasjon

Ettersom min deltakelse i klasserommet var så stor, valgte jeg å bruke videoobservasjon som innsamlingsmetode. Da ble det frigjort tid slik at jeg kunne fungere som en lærer i klasserommet. Jeg hadde en deltagende observasjon når jeg var i klasserommet ettersom jeg tok noen feltnotater, men når jeg ser på økten fra filmkameraene i ettertid vil jeg kunne

observere elevene mer direkte. Dette krever at jeg klarer å løsrive meg fra nærheten som oppstår ved mitt nærvær i klasserommet. Ved å bruke videoobservasjon får jeg også mulighet til å fange opp både verbale og ikke-verbale ytringer som elevene har seg imellom (Postholm og Moen, 2011). Dette passet bra når elevene jobbet to og to sammen. På denne måten kunne jeg være sikker på å få med meg god data som jeg i ettertid kunne analysere. For at elevene ikke skulle bli opptatt av videoutstyret poengterte jeg at det ikke bruktes til noe annet enn at jeg skulle se på det i ettertid. Dette godtok elevene, og de så ut til å glemme videokameraene etter hvert.

Videoopptak er ikke en kopi av virkeligheten og det som skjer i klasserommet. Kameraets og mikrofonens plassering gir begrensninger i hva som blir registrert, og vil dermed bare være en representasjon av realiteten (Bjørndalen, 2009). Elevene som sitter i motsatt hjørne av mikrofonen kan oppfatte en helt annen samtale enn det mikrofonen registrerer. «*Generelt sett er problemene med både lyd- og bildeopptak større jo flere personer som er implisert i den pedagogiske situasjonen*» (Bjørndalen, 2009:71). I introduksjonstimen var det 18 elever som jeg skulle registrere gjennom to kameraer med mikrofon, mens de andre øktene var det tre eller fire elever. Det var mer utfordrende å samle god data gjennom to kameraer i nesten full klasse fremfor tre-fire elever på et grupperom.

Både underveis i øktene, og når datamaterialet har blitt studert i ettertid har jeg hatt noen faste punkt jeg har observert etter. For å kunne svare på min problemstilling har jeg hatt mye fokus på hvordan elevene har gått frem for å finne ut av, og svare på oppgavene de har fått. Jeg har også studert hvordan de selv forklarer fremgangsmåte både til meg og til medelever. Jeg har ikke lagd meg et observasjonsskjema, men har punkter over hva som skal observeres. Dette ligger som vedlegg 2.

4.4.2 Intervju – gruppeintervju

Et kvalitativt forskningsintervju er den mest brukte måten å samle inn data på innenfor kvalitativ metode (Christoffersen og Johannesen). Gjennom et intervju har en mulighet til å få fylldige og detaljerte beskrivelser, og intervjuet vil ha en form for struktur, og et klart formål. Jeg har valgt å gjennomføre et gruppeintervju med de fire utvalgte elevene, i tillegg til videoobservasjonene gjennom øktene. Jeg hadde i utgangspunktet tenkt å ha et fokusgruppeintervju med elevene, men etter overveielser om at dette kanskje var et vanskelig

tema for elevene å diskutere mellom hverandre, valgte jeg å ha et gruppeintervju. Jeg håpte likevel at elevene ville kommentere og utvikle en diskusjon så derfor gjennomførte jeg et gruppeintervju fremfor intervju med én og én. En utfordring med å gjennomføre et gruppeintervju kontra et intervju med én og én, vil være at de kan bli påvirket av hverandres svar. Den første som svarer kan gi et så styrende svar at de andre svarer likt, selv om de kanskje ville svart annerledes om de hadde blitt spurt spørsmålet alene. Derfor ønsker jeg også å filme intervjuet slik at jeg kanskje kan observere reaksjoner eller blikk som kan tyde på en slik feilkilde. Det er også utfordrende å intervju barn fordi de ikke i like stor grad som voksne har utviklet en reflekterende tankegang, og de kan ha problemer med å svare på noen spørsmål.

Gruppeintervjuet har jeg forberedt som et semistrukturert til strukturert intervju. Et semistrukturert intervju har en intervjuguide, men ikke alltid fastsatte spørsmål. Et strukturert intervju har klargjorte spørsmål, men det er åpne spørsmål slik at det ikke er fastsatte svaralternativ slik det er i strukturert intervju med faste svaralternativer (Christoffersen og Johannessen, 2012). Det kan være en fordel med en viss grad av struktur, fordi det ofte blir lettere og både gjennomføre intervjuet, men også å analysere det. Jeg hadde klargjorte spørsmål før intervjuet, men hadde ikke fastsatt rekkefølge på dem og hadde rom for å ta opp spørsmål som kom underveis. Jeg kunne da bevege meg fritt i intervjuguiden, og undersøke nærmere eller mer inngående i svarene informantene kommer med. Grunnen til at jeg valgte intervju er for å få frem informantenes erfaringer og oppfatninger på en god måte. Spørsmålene ligger som vedlegg 4.

Jeg gjennomførte også uformelle samtaler med elevene samtidig som de jobbet. Jeg stilte spørsmål underveis i øktene om hvordan de hadde tenkt, hvorfor de valgte å gjøre det slik eller hva de hadde gjort. Det er en form for et ustrukturert intervju, men er en naturlig dialog gjennom en undervisningstime for en lærer som fungerer som en veileder for elevene (Postholm og Moen, 2011).

Når jeg gjennomførte steg tre og fire i innsamlingen min sviktet lyden på videoopptaket. Slike ting kan skje når en holder på med teknisk utstyr. Jeg valgte derfor å gjennomføre intervjuet på nytt med en båndopptaker. Grunnen til at jeg nå benyttet meg av båndopptaker og ikke videoopptak, var fordi jeg ikke lengre hadde tilgang på videokamera. Men ettersom det kun var intervjuet jeg skulle gjennomføre på nytt så jeg det som dekkende nok å benytte meg av båndopptaker. Elevene fikk de samme spørsmålene som de gjorde under det første intervjuet.

Gjennom observasjonene vil min subjektivitet spille inn på filtreringen av datagrunnlaget. Når elevenes egne meninger kommer frem vil de også virke på hvordan jeg ser på prosessen i etterkant av intervjuet ved at deres meninger farger min subjektivitet, og skaper en interaksjon (Postholm og Moen, 2011:61). Interaksjonen mellom observasjonene og intervjuet kan gjøre at jeg får en større forståelse for læringsarbeidet som har foregått.

4.5 «Aksjonen» i klasserommet

4.5.1 Beskrivelse av utvalg

Klassen er en typisk norsk klasse med dyktige elever. De har ikke jobbet systematisk med problemløsningsstrategier i klassen, men læreren bruker noen problemløsningsoppgaver til noen temaer. Da må elevene selv finne ut av sammenhengen, og de går gjennom oppgaveløsningene i fellesskap til slutt. Læreren oppfordrer elevene til å måtte tenke og prøve selv, og gjerne tegne til oppgavene. Men elevene har ikke hatt gjennomgang av hvordan en går frem for å løse en problemløsningsoppgave.

Mine utvalgte elever, Bente, Fred, Hilde og Steinar er alle positive og læringsvillige. Steinar er faglig sterk i matematikk og er konsentrert når han jobber. Han kan bli stresset og frustrert av hindringer han møter på. Fred er middels- til sterk i matematikk. Han kan lett miste konsentrasjon og dermed jobbe mindre effektivt. Hilde er meget sterk i matematikk, og er ei trygg jente som virker reflektert for alderen. Hun jobber godt og konsentrert. Bente er middels sterk i matematikk. Hun er noe svakere på lesing, og er mer stille enn de andre fire elevene. Alle elevene var aktive på fritiden med ulike fritidsaktiviteter og fungerte godt sosialt i klassen.

4.5.2 Analyse av elevoppgaver

Før jeg går videre og beskriver innsamlingen av datamaterialet mitt velger jeg å presentere elevoppgavene. I alle øktene oppfordret jeg elevene til å diskutere mellom hverandre fremfor å spørre meg.

Til steg én brukte jeg en oppgave fra Kodeklubben, som blir brukt som introduksjonsoppgave til elevene der. Oppgaven ligger som vedlegg 3.

Til steg to var kriteriene for oppgaven at den skulle være problemløsende, og at skulle benytte seg av de kodebrikkene de hadde fått introdusert under steg én. Oppgaven skulle være en åpen oppgave. Elevene fikk i oppgave å videreutvikle oppgaven fra introduksjonsøkten ved at de skulle få katten Felix til å gå fortere og fortere for hvert sekund som gikk. Dette var et forslag fra en elev etter vi hadde hatt introduksjonsøkten, og jeg valgte å bruke den fordi elevene virket motivert til å finne ut av problemet. For å klare denne oppgaven, ved at katten skal øke fart hvert sekund, må elevene lage en «variabel» som de definerer for «fart», og velge at den skal øke for hvert sekund. Oppgaven er problemløsende ved at det er mulig å løse den på flere måter, men for å få denne økningen til å bli konstant må en bruke en «variabel». Det finnes likevel flere muligheter for hvordan en skal lage seg koden med bruk av variabelen. Elevene fikk presentert oppgaven muntlig. Jeg stilte også spørsmål underveis hva elevene hadde forsøkt og hvordan de tenkte.

Til steg tre, når elevene jobbet med problemløsningsoppgaver i matematikk valgte jeg to ulike oppgaver. Oppgave 1 (figur 3) er en oppgave hentet fra lærernes bok i læreverket Multi 5B (Alseth m.fl., 2006). Jeg valgte denne fordi oppgaven kan løses på flere måter, og kan tilpasses til elever på ulikt nivå. Det er også en åpen oppgave, og det er 8 ulike løsninger på oppgaven.

Oppgave 1

Noen venner skal på butikken. Alle kjøper det samme. Til sammen betaler de 36 kroner. Prisen på varer de kan kjøpe er: Sjokoladen koster 2 kr. Brus koster 5 kr. Hva bestilte de, og hvor mange var de?

Er det flere mulige løsninger?



Figur 3 Matematikkoppgave 1 (Alseth m.fl., 2006)

Oppgave 2 (figur 4) har jeg hentet fra Matematikk.org (u.å.). Denne valgte jeg fordi jeg ville se om elevene klarte å systematisere og sjekke opplysningene for å komme frem til rett svar. Systematisering og sjekking av korrekthet er viktig når en skal programmere, og derfor valgte jeg denne for å ha et godt grunnlag til å kunne sammenligne med programmeringen.

Oppgave 2

I en gate har vi husnumrene 2, 4, 6 og 8. I hvert av de 4 husene bor det en elev. Hver elev har sitt favorittfag og dessuten et eget kjæledyr.

- a) Eleven i nr.6 liker fysikk.
- b) Hunden er nabo til hesten.
- c) Ole liker matematikk og bor på den ene enden.
- d) Erik bor mellom hesteeieren og eleven som har fysikk som favorittfag
- e) Kristine bor ved siden av katteeieren.
- f) Eleven som bor lengst til høyre, har teknikk som favorittfag.
- g) Katten er nabo til eleven som liker norsk.

I hvilket hus bor Åse, og hvem eier apekatten?



Figur 4 Matematikkoppgave 2 (Matematikk.org)

4.5.3 Beskrivelse av innsamlingen

4.5.3.1 Introduksjonstimen, steg én

Timen ble gjennomført med 18 elever til stede. Jeg startet med å snakke om programmering, hva det var og hva man kunne bruke det til. Oppgaven elevene fikk ligger som nevnt som vedlegg 3. Jeg kunne fokusere på å tenke gjennom formuleringer til vanskelige begreper. Jeg utformet derfor et skjema med begreper som jeg skrev forklaringer til, som hver elev fikk i starten av økten, som nevnt i kapittelet 2.0 om Scratch. Dette ligger som vedlegg 1.

Etter at vi hadde snakket litt om programmering viste jeg internettsiden programmet ligger på, og forklarte kort hvordan man skulle skrive en kode. Elevene åpnet nettsiden og lagde seg en bruker. Når de hadde lagd seg en bruker, begynte de å kode. De fikk begynt, og de aller fleste var ferdige med steg én i oppgaven, før vi mistet tilgang til internettet 30-35 minutter inn i timen. Ettersom programmet kjører via nettleseren er en avhengig av internett for å få det til å

fungere. Tilgangen kom heller ikke tilbake i løpet av timen. Elevene fikk derfor i oppgave å programmere hverandre, ved å skrive en kode manuelt på papir som den andre etterpå skulle følge. Mens elevene skrev og spilte ut koden, koblet jeg min datamaskin til prosjektoren og ordnet klart en nedlastet versjon av Scratch som en ikke behøver internett for å fungere. Versjonen er gammel og har et litt annet utseende, men det er de samme prinsippene som gjelder. Elevene fikk så i oppgave å bytte på å lese opp instruksene, og bytte på å programmere via dette programmet fra min datamaskin. På denne måten fikk alle elevene enten lese eller programmere, og vi lagde spillet sammen. Alle elevene fikk sett hvordan programmet fungerte, og fikk vært med på å fullføre spillet.

4.5.3.2 Scratchoppgaver, steg to

Jeg startet innledningsvis med å si at vi skulle programmere i Scratch slik som forrige gang, men denne gangen uten oppgavearket med «fasit» på. Jeg la likevel frem oppgavearket fra forrige gang slik at elevene hadde mulighet til å se på dette om de hadde lyst.

De utvalgte elevene jobbet to og to, Fred sammen med Hilde og Bente sammen med Steinar. Vi jobbet i et grupperom tilknyttet klasserommet deres. Etter omkring 20 minutter gav jeg et hint, om at de måtte bruke noe de lærte forrige gang, ettersom elevene hadde jobbet seg fast.

Vi ble litt forsinket i starten blant annet av treg oppstart av datamaskiner, før de var logget inn på Scratch og klar til å starte. Det vil si at de hadde omkring 30-35 minutter å jobbe på. Jeg valgte derfor å la elevene få jobbe i denne tiden, også fordi de sakte men sikkert nærmet seg en løsning. Jeg hadde ikke lyst å bryte opp i håp om at elevene skulle huske hva en variabel kunne gjøre, og dermed forstå hva de måtte gjøre. Derfor ble det ikke tid til en samtale i etterkant av jobbingen om hvordan de hadde gått frem. Jeg som forsker måtte forholde meg til den tiden jeg hadde fått tildelt, og kunne ikke gå ut over denne. Jeg syntes prosessen elevene var inne i var så interessant at jeg valgte ikke å bryte denne opp.

4.5.3.3 Matematikkoppgaver, steg tre

Vi startet timen med å snakke kort om hva elevene hadde gjort forrige gang, og hva de syntes om det. Denne timen var Fred syk, og timen ble kun gjennomført med Bente, Hilde og Steinar, på samme grupperom som under steg to. Elevene begynte med oppgave 1. De fikk

velge om de skulle samarbeide eller jobbe selvstendig, men alle elevene jobbet selvstendig. Jeg la også frem noen blyanter de kunne bruke som hjelp om de ønsket, men ingen benyttet seg av dette. Alle elevene klarte å finne minst to ulike korrekte svar til oppgaven. I oppgave to var det kun to av tre elever som klarte den. Denne oppgaven krever ingen regning, men en må systematisere informasjon for å klare å plassere de ulike i rett hus.

Når elevene hadde gjort disse to oppgavene gjenstod det 15 minutter. Disse ble brukt til å samtale om hvordan elevene hadde kommet frem til svarene og hvordan de hadde tenkt. Vi hadde også intervju i denne økten, beskrevet i 4.5.3.4.

4.5.3.4 Gruppeintervju, steg fire

Til intervjuet, steg fire, har jeg brukt tid på å forme spørsmål som kan hjelpe meg til å få svar på min problemstilling og mine forskningsspørsmål. Spørsmålene ligger som vedlegg 4.

Intervjuet ble gjennomført to ganger, og andre gang med lydopptaker som jeg bruker som datamateriale. Ettersom jeg måtte inn en ekstra gang, og med en travel skolehverdag, fikk jeg avsatt tid i lunsjen, på 15 minutter. Intervjuet ble gjennomført på samme grupperom som tidligere, mens elevene spiste lunsj. Denne timen var Fred frisk, men nå var Hilde blitt syk. Steinar og Bente fikk spørsmålene for andre gang, og hadde dermed tenkt og svart på disse spørsmålene fra før av. For Fred derimot var spørsmålene helt nye. Elevene svarte godt for seg, og intervjuet gikk uten problemer.

4.6 Analysemetode

Datamaterialet mitt består av de fire stegene som beskrevet i kapittel 4.5.3, og som er samlet inn gjennom videoopptak, båndopptaker, egne feltnotater og elevbesvarelser i matematikkøkten. Siden jeg er ute etter å undersøke elevenes problemløsende arbeidsmåte i programmeringen var det økten rundt programmeringen jeg har tatt utgangspunkt i, og bygger det opp med situasjoner fra matematikken.

Som en start på analysen begynte jeg med å transkribere videofilmene, og leste gjennom disse for å bli kjent med datamaterialet. Jeg fortsatte deretter med den konstant komparative analysemetoden hvor jeg forsøkte å forenkle og strukturere datamaterialet mitt (Postholm, 2005).

Jeg begynte å kode materialet ut ifra prinsippene i problembasert læring. Jeg fant ut jeg måtte benyttet meg av *åpen koding* hvor jeg satte navn på ulike situasjoner i datamaterialet (Postholm, 2005). Jeg kom frem til ni ulike koder:

- Planlegging.
- Gjett og sjekk.
- Problemløsningsstrategier.
- Samarbeid.
- Spørsmål.
- Elevenes holdning.
- Feilsøking og vurdering.
- Skrivning av kode.
- Systematiseringsevne.

Arbeidet videre gikk til å kategorisere kodene slik at materialet ble håndterlig, og antall enheter reduseres (Postholm, 2005). Jeg strukturerte kodene og satte det i et nytt system, og gjennomførte da en *aksial koding* (Postholm, 2005). Jeg kom gjennom dette arbeidet frem til denne oppbygningen:

1. Elevenes problemløsningsstrategier
 - Forstå problemet
 - Planlegging og gjennomføring
 - Se tilbake og reflekter
2. Elevenes vurdering av læringsprosess
3. Elevenes deling, samarbeid og spørsmålsstilling

Videre fant jeg situasjoner i datamaterialet mitt som jeg vil ta opp og diskutere i kapittel 5.0 og som er bygd opp rundt kjernekategoriene og underkategoriene presentert her.

4.7 Etske drøftelser og metodekritikk

4.7.1. Informasjon og samtykke

Selv om jeg ikke samler inn personopplysninger må jeg ha godkjennelse fra Norsk samfunnsvitenskapelige datatjeneste AS (NSD) fordi jeg filmer elevene. Prosjektet ble derfor før oppstart meldt inn til NSD og godkjent 04.11.2014. Godkjennelsen finnes i vedlegg 5.

Foresatte i den aktuelle klassen har fått informasjonsskriv med svarlapp om godkjenning, se vedlegg 6. Alle foresatte godkjente prosjektet før oppstart.

4.7.2 Ethiske overveielser

Jeg som forsker må ivareta ulike forskningsetiske prinsipper som for eksempel anonymitet, selvbestemmelse hos informantene og å unngå skade ved å gi en feilaktig fremstilling av informantene (Christoffersen og Johannessen, 2012:47). Kvale og Brinkmann (2009) sier det vil være fullt av etiske og moralske utfordringer rundt et forskningsintervju, og fremhever presentasjon av informantenes svar som den største etiske utfordringen. Det menneskelige samspillet mellom meg som forsker og elevene som informanter, og også elevene seg i mellom, vil påvirke både meg og elevene. Ettersom jeg er ute etter elevenes beskrivelser av tanker rundt løsninger av oppgavene er dette et samspill jeg må ta etiske og analytiske hensyn til. Jeg må også sørge for ikke å krenke eller gi en feilaktig fremstilling av informantenes beskrivelse av deres tanker. Min beskrivelse av dataene jeg samler inn vil være en representasjon av virkeligheten, og ikke den faktiske virkeligheten. Representasjon av virkeligheten fungerer som bindeleddet mellom forskerens forforståelse, og virkeligheten (Christoffersen og Johannessen, 2012). Derfor vil min forkunnskap og forforståelse også påvirke hva jeg observerer og oppfatter, hvordan jeg fremstiller dette og tilslutt hvordan jeg tolker det.

4.7.3 Metodekritikk

At jeg kommer inn som forsker og observerer elever, og tar noen ut av klasserommet og gir dem nye oppgaver kan påvirke elevene ettersom de ikke er vant med en slik situasjon. Jeg har heller ikke et forhold til elevene fra tidligere, men forsøkte å legge til rett for at elevene skulle føle seg trygge ved at jeg gjennomførte introduksjonstimen i full klasse før jeg gjorde utvalget. Et problematisk aspekt ved mitt forskningsarbeid er min nærhet til feltet jeg studerer. Selv om det ikke er et felt jeg er i til vanlig, driver jeg som nevnt med en deltakende observasjon. Jeg som forsker vil ha en nærhet til mine informanter som igjen kan føre til en påvirkning. Wadel (2006:115) sier: *«i alle typer forskning er forskeren deltakende i sin egen kunnskapsvervelse både med hensyn til oppdagelser og validisering»*. Jeg må altså ta hensyn til at jeg som forsker påvirker min egen forskning. Det kan tenkes at elevene vil være

mer interessert i programmeringen fordi det er noe nytt, og det kommer inn en ukjent person som smitter over et engasjement og entusiasme enn om en av de «vanlige» lærerne hadde gjennomført timen. Når elevene ser at de blir filmet kan de også tro at de er med på noe viktig, og dermed kan gjøre en ekstra innsats i timen.

5.0 Analyse og diskusjon

Problemstillingen min er som nevnt: *Hvordan kan programmering i matematikken påvirke læringsstrategiene til elevene?* Jeg skal også se nærmere på hvordan problemløsningsstrategier elevene bruker, og hvordan de arbeider når de løser en problemløsende oppgave i Scratch. Jeg skal også se om programmet kan brukes som en digital problemløsningsoppgave. Derfor tar jeg utgangspunkt i programmeringsøkten, som beskrevet i kapittel 4.5.3.2, og det er denne økten jeg har hatt fokus på under analysearbeidet. For å ha grunnlag til å kunne svare på problemstillingen må jeg sammenligne Scratch- og matematikkøkten. Derfor vil jeg trekke inn situasjoner fra matematikkøkten, som beskrevet i kapittel 4.5.3.3, som belyser likheter og forskjeller av elevenes arbeidsmåter som kan være til hjelp til å svare på problemstillingen. Som jeg presenterte i kapittel 4.6 har jeg kodet datamaterialet mitt og kom gjennom dette arbeidet frem til noen trekk hos elevenes jobbing:

- Elevenes problemløsningsstrategier
- Elevenes vurdering av egen læringsprosess
- Elevenes deling-, samarbeid og spørsmålsstilling

Jeg har valgt å dele analysekapitlet inn etter disse tre trekkene og vil vise til situasjoner fra datamaterialet for å støtte under mine analyser. Jeg vil etter hvert delkapittel diskutere mine funn. Etter jeg har presentert analysen velger jeg å ha en avsluttende og sammenfattende diskusjon.

5.1 Elevenes problemløsningsstrategier

Når jeg skal beskrive hvordan elevene jobbet, og hvilke strategier som kom til uttrykk, har jeg har valgt å ta utgangspunkt i Polyas fire steg i problemløsningsprosessen; *forstå oppgaven, legge en plan, gjennomføre planen og se tilbake og reflekter*. Jeg deler derfor inn underkapitlet etter disse hvor jeg presenterer eksempler fra elevenes jobbing. Når jeg begynte å se etter hvilke problemløsningsstrategier elevene benyttet seg av, var dette mye vanskeligere enn jeg hadde forutsett. Det var vanskelig å se hvordan, og om elevene brukte noen gitte strategier, eller å få de til å forklare hvordan de tenkte. Jeg har valgt å presentere steg to, *legge en plan*, og steg tre, *gjennomføring*, sammen for blant annet å unngå gjentakelse.

5.1.1 Forstå problemet

Oppgaven i Scratch gikk ut på å få katten i spillet de lagde i introduksjonsøkten til å gå fortere og fortere for hvert sekund som gikk. I matematikkøkten gikk oppgave 1 ut på å finne ut hvor mange som hadde vært på butikken og hvor mye de hadde handlet for hver vare, og oppgave 2 gikk ut på å systematisere personer og deres interesser og dyr i korrekt hus etter gitte opplysninger.

Når elevene fikk oppgavene, både i Scratch- og matematikkøkten, begynte alle rett på oppgavene. De diskuterte ikke innholdet i den uttalte oppgaven, og diskuterte ikke begreper. Elevene virket til å forstå oppgavene og trengte ikke gå gjennom begreper eller identifisere hva spørsmålet eller problemet omhandlet. Det kan likevel diskuteres om elevene hadde klart å vise dette gjennom kontrollspørsmålene rundt forståelse av oppgavene. Eksempler på slike spørsmål er «Hva går oppgaven ut på?» og «Hva er den ukjente i oppgaven som må undersøkes?» (Polya, 1957). Jeg valgte ikke å stille slike spørsmål fordi jeg var opptatt etter å se etter strategiene elevene benyttet seg av. Jeg hadde også lyst å ha minst mulig innvirkning på elevenes jobbing slik at jeg ikke var en faktor jeg måtte ta med i betraktningen i analysen.

5.1.2 Legge en plan og gjennomføring

Under programmeringen la elevene frem og kom med forslag til hvordan de kunne løse oppgaven. Elevene arbeidet to og to, Hilde sammen med Fred, og Bente sammen med Steinar. Begge gruppene lagde seg to ulike planer, og dermed to ulike forsøk på en kode for hvordan de skulle løse oppgaven. Jeg startet hele programmeringsøkten ved å si:

«Dere må tenke og prøve å diskutere sammen hvordan dere skal få dette til, og prøve å lage en plan, og så forsøke det»

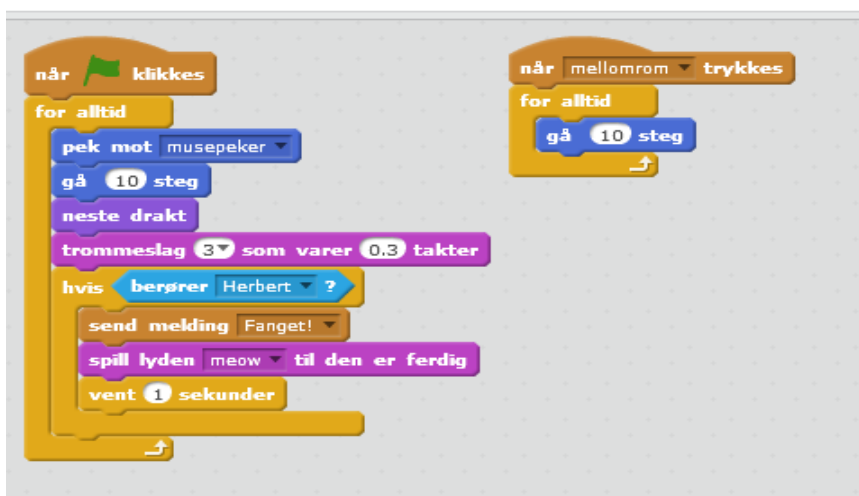
Elevene brukte ikke noe tid i starten til å diskutere hva de kunne gjøre, eller hvilken fremgangsmåte de skulle velge for å løse oppgaven. De undersøkte dette samtidig som de lagde seg en kode. I matematikken legger ikke elevene frem noen plan, men begynner også her rett på oppgavene.

Hilde startet i programmeringsøkten med å gå systematisk gjennom alle kategoriene før hun og Fred la frem en plan. Hun skaffet seg en oversikt over de ulike kategoriene, og så også nærmere på noen av brikkene i programmet Scratch. Fred ble som en følge av samarbeidet mellom dem med på denne systematiseringen, selv om han virket mer utålmodig med å

begynne å programmere. Denne systematikken hos Hilde kjente jeg igjen når hun skulle løse matematikkoppgavene. Hun gikk i matematikkøkten systematisk gjennom informasjonen i begge matematikkoppgavene, og strukturerte arbeidet videre ut i fra dette. I begge matematikkoppgavene klarte hun å systematisere ut de viktige opplysningene.

Gjennom den systematiserte gjennomgangen i programmet Scratch fant Hilde og Fred en kodebrikke som heter «når mellomrom trykkes», som er en hendelsesbrikke, forklart i kapittel 2.2. Dette var ikke en brikke vi hadde gjennomgått i introduksjonsøkten. De lagde seg en plan som omhandlet at de skulle ta i bruk denne brikken, ved at farten skulle økes når de trykte på mellomrom. Fred forklarer hva de tenker mens han lager koden:

Fred: «Men da kan vi si som dette: gå ti steg i.... og så gjør vi slik at den ligger under en «for alltid», gå 10 steg, og så finner vi en knapp som gjør at den øker farten. Hmm. Vi må se hva vi har. «For alltid» først».....«Når mellomrom trykkes går den «for alltid» 10 steg, så må vi finne en måte for å få det til å gå fortere». [Se figur 5]



Figur 5 Utsnitt av kode som Hilde og Fred forsøker. Eksisterende kode fra introduksjonsøkten til venstre, elevenes forsøk til høyre.

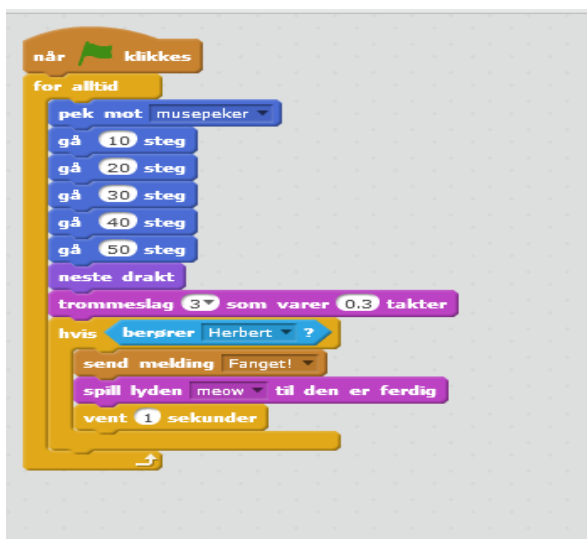
«For alltid» og «gå ti steg» er kodebrikker i Scratch-programmet. Når han sier knapp, mener han en kodebrikke. Hilde og Fred lager en ny kode ved siden av den eksisterende koden fra introduksjonstimen. De bruker en del tid på å lete etter denne «fortere» brikken, og prøver små varianter av planen de har lagd. Til slutt legger de fra seg denne planen. Hilde og Fred lager seg egentlig en plan hvor de må benytte en selvlaget *variabel* for å få til koden de forklarer. Elevene tenker ikke at de må lage denne variabelen selv, og mye tid går bort til å

lete etter «fortere» brikken. I matematikken er det å løse eller finne elementer i oppgaver en viktig del av det å løse problem. I alle problem er det å identifisere mangler eller den ukjente viktig. Elevene jeg har undersøkt gjorde ikke dette i matematikkoppgavene de fikk presentert, som kan skyldes at oppgavene antakelig ikke var veldig problemløsende for dem.

Steinar hadde forsøkt Scratch-programmet hjemme, og begynte rett på å kode når han og Bente hadde fått oppgaven. Han uttrykte ikke noe plan, men begynte rett på. Da forsøkte han, uten noen uttalt mening, å plassere noen «gå» brikker innen enn løkke «for alltid». Det kan tenkes at fordi han hadde sett nærmere på programmet hjemme hadde han ikke et behov for og gjennom kategoriene slik Hilde og Fred gjør. Denne mangelen på systematikken hos Steinar kjenner jeg igjen i matematikken. Steinar fikk problemer under den andre matematikkoppgaven ved at han ikke klarte å systematisere ut ifra opplysningene og dermed komme frem til rett svar.

Bente kommer etter hvert med forslag på en plan under programmeringsøkten:

«Forsøk å ta en brikke med 10 steg, kanskje det fungerer med mange. Når katten går, inni en for alltid, så må vi ha veldig mange, så kan den «vent»-brikken være den siste. Hvis vi tar mange, 10 av dem, så går han oppover, sånn 30 til 40 så kan vi gå oppover til hundre, og så når han har gått til hundre, det blir kanskje litt vanskelig». [Se figur 6]



Figur 6 Utsnitt av kode som Bente og Fred forsøker. De legger inn nye brikker i den eksisterende koden fra introduksjonstimen.

Hun uttrykker en plan hvor de bruker mange av «gå» brikkene for å få til oppgaven, og hvor de videreutvikler den eksisterende koden fra introduksjonstimen. De forsøker dette, men tror

ikke de får koden til å fungere og sletter den igjen. Planen hun legger frem krever mye jobb med mange brikker. Hvis de hadde benyttet en lavere tallverdi ville de sett at katten ville gått fortere og fortere, men ikke for hvert sekund og den ville til slutt stoppet på 50. De ville likevel fått en slags løsning på oppgaven.

Når hverken Hilde og Fred eller Bente og Steinar klarer å finne en brikke som heter «fortere», eller en brikke som får noe til å skje fortere, blir de frustrerte. De tenker at de har gjort noe feil, og stopper ikke opp og reflekterer rundt prosessen. De deler ikke problemet opp i mindre deler. Om elevene hadde identifisert at et av delproblemene var å finne ut hvordan de kunne lage en «fortere» brikke kunne de kanskje klart sammen å undersøke variabler nærmere. Elevene jobber i Scratch i rundt 25 minutter (av 40 totalt) og ser ut til å ha jobbet seg fast. De prøver lite nytt, og begynner å vise tegn på å gi opp ved at de blant annet begynner å spille det opprinnelige spillet fra introduksjonsøkten fremfor å lage kode. Jeg velger da å gi elevene et hint:

«Dere må bruke noe av det dere lærte forrige gang for å få dette til. Så noe dere gjorde forrige gang må dere lage en ny vri av»

Hilde og Fred begynner å se nærmere på den opprinnelige koden fra spillet. De leser gjennom oppgavearket for å se hva de lærte i introduksjonstimen. Elevene ser altså etter mønster i koden for å undersøke om de kan bruke noe av det i den nye koden de må lage. Men Hilde og Fred klarer ikke hente ut noe informasjon de kan bruke fra dette. Bente og Steinar undersøkte også den gamle koden, men ikke like systematisk og grundig som Hilde og Fred gjorde, som gjør at de går fort videre.

Hilde og Fred går inn på data-kategorien, og ser på den gamle variabelen de lagde i introduksjonsøkten. Her finner de noe de tror de kan bruke:

Hilde: «Endre «fart», jaaa, nei det går ikke! Nei det går ikke ann å skrive! Hvis det hadde vært endre til «fart»!»

Fred: «hvis det hadde vært endre «fart» med 1 hvert 5 sekund. Da kunne vi bare sette den under»

Hilde og Fred går bort fra variabler igjen når de ikke tror de kan bruke noe der, men nå undersøker Steinar og Bente dette nøyere.

Steinar: «ok, jeg må bare prøve. «Lag en ny variabel» [lager ny variabel og gir den navn «fart»]

Bente: *«nei vent litt, du må sette den inn her»* [peker på inni en «for alltid» i koden]

Steinar: *«nei, det blir ikke å gå. [sletter variabelen «fart»] Det er en variabel ikke noe annet»*

Jeg spør om elevene har testet koden de lagde, og oppfordrer dem til å gjøre dette før de sletter koden igjen. Bente og Steinar lager variabelen på nytt, og prøver enda en gang. Hilde og Fred hører at jeg sier dette, og velger og forsøke på nytt de også. Begge gruppene lager en kode hvor de bruker en selvlaget variabel kalt «fart». Ingen av gruppene klarer å lage en kode som er korrekt og som gjør at det fungerer, men nå begynner de å nærme seg. De nærmer seg fordi begge gruppene har lagd seg en variabel som de kaller fart, og bruker den i den eksisterende koden. Grunnet feil i koden fungerer den ikke som den skal. En feilsøking her kunne løst problemet for elevene. Tiden til denne økten går ut, og elevene får ikke tid til å gå nøyere gjennom kodene. Ingen klarte å fullføre oppgaven akkurat slik den ba om. I intervjuet sa blant annet Steinar hva som var vanskelig med oppgaven:

«Ja det va litt vanskelig for jeg vet ikke helt hvordan man blander blokker for at det skal bli en spesiell kode på en måte som får den til å gå fortere»

I matematikken uttrykker ikke noen av elevene noe plan for hvordan de skal løse oppgavene, og alle begynner med å regne i oppgave én, eller plassere opplysninger i oppgave to. Det kan være at oppgavene ikke var problemløsende for elevene. Selv om de ikke så hva svaret kunne bli på oppgaven én, begynte de alle rett på å regne i gangetabellen, og trengte hverken å lage seg en plan eller å modellere for å finne en fremgangsmåte. De noterte likevel noe nøkkelinformasjon på oppgavearket som for eksempel priser i oppgave én, eller informasjon i oppgave to. Alle elevene tenkte i gangetabellen for å finne frem til løsningen på den første matematikkoppgaven, og hadde altså en multiplikativ strategi. Jeg tror den eneste grunnen til at de måtte jobbe med oppgave én i en stund var fordi det var mange elementer i oppgaven å holde styr på. De begynte rett på med å prøve og feile med tall, forklart i kapittel 3.4. Alle elevene begynte med å tenke «hvor mange av hvert produkt for 36 kr». Etter hvert gikk elevene over til å tenke på «hvor mange personer», for eksempel hvis det var 4 personer, hva kunne hver av dem kjøpe.

5.1.3 Se tilbake og reflekter

Ettersom elevene ikke fant en løsning på oppgaven i programmeringen hadde de ikke mulighet til å se tilbake på hele prosessen fra de fikk problemet til svaret, og refleksjon rundt dette. Elevene reflekterte heller ikke når de hadde forsøkt å lage en kode. De undersøker ikke om det de har gjort gir mening og kan brukes slik eller i en annen form. De viser mangel på en slik refleksjon ved bare å slette koden og starte på nytt, uten å diskutere hva de forsøkte. Dette kommer jeg til tilbake til under kategorien *elevenes vurdering av egen læringsprosess* i kapittel 5.2.

I matematikkøkten viste heller ikke elevene noe vilje til å stoppe opp og reflektere rundt hva de hadde gjort. Når de var ferdig med oppgaven begynte de rett på neste oppgave eller bare satt og ventet til de andre var ferdige. Selv om elevene sjekker sine svar, noe jeg også kommer tilbake til under kapittel 5.2, går de ikke nøye gjennom sine valgte strategier for å overveie og undersøke om deres resultat og veien til svaret var god.

5.1.4 Diskusjon rundt elevenes problemløsningsstrategier

Som jeg viser i datamaterialet og gjennom analysen benytter elevene seg mest av «gjettt og sjekk» strategien, forklart i kapittel 3.4. Steinar uttrykte også selv «*jeg bare prøver, jeg vet ikke*» når han og Bente startet å lage kode. Hazzan m.fl. (2014:76) sier at elevene må lære seg noen formelle ferdigheter på hvordan en skal løse en problemløsningsoppgave, ellers vil selv den mest oppfinnsomme eleven bruke en uproduktiv «gjettt og sjekk» strategi. Blant annet sier også Chapin m.fl. (2014) at et kjennetegn ved en nybegynner innenfor problemløsning er at de bruker kort tid på Polyas første steg i problemløsningsprosessen. De starter rett på oppgaven, og uten å ha lagd seg noen gjennomtenkt plan. De reflekterer heller ikke om deres valg er effektiv og kan gi en løsning (Chapin m.fl., 2014). Disse trekkene på en nybegynner i problemløsning kjenner jeg igjen hos elevene når de jobber både i programmeringen og i matematikken. Elevene bruker særdeles kort tid på Polyas (1957) første steg. Når de videre lager seg kode er det som nevnt gjennom å gjette på et svar, og sjekke om den fungerer. Men de bruker ikke de erfaringene til å lage en bedre plan. Refleksjonen rundt hva som er forsøkt, hva de kan undersøke, hva som vil være unyttig å prøve er ikke tilstede. Elever som innehar en god kompetanse på problemløsning og problemløsningsstrategier skal kunne vurdere og reflektere rundt sine valg av strategi. Dette gjorde ikke elevene i noen grad. Elevene benytter

seg av det Bjorklund (1990) presenterer som «forward search». Elevene bruker flere metoder på koden for å få til en løsning, men reflekterer og vurderer ikke deres svar. Elevene tar heller ikke med seg noe informasjon fra den forrige utprøvde koden til deres neste forsøk som Bjorklund (1990) presiserer som en viktig del av det å planlegge en vei til løsning på problemet. Om elevene kunne klart å identifisere at det å kunne lage en «fortere» brikke var et delproblem, ville de kanskje uten mitt hint klart og kommet frem til variabler. De benytter seg altså ikke av «problem reduction».

Elevene bruker heller ikke tid i det siste steget til Polya, *se tilbake og reflekter*, som forklart i kapittel 3.4. Denne prosessen er det Polya (1957:15) understreker som en viktig prosess for nettopp det å få større kunnskap om en problemløsningsprosess og utvikle deres ferdigheter til å løse problemer.

Læreren til elevene sa at de ikke hadde jobbet med problemløsningsstrategier, og dette vist i måten elevene jobbet på. Slik jeg forstod det på læreren hadde de jobbet med problemløsning som lære *for* problemløsning som Van De Walle m.fl. (2010) viser til. Ved at de også ikke hadde jobbet med lære *om* problemløsning kan det være en grunn til at de nå ikke klarte oppgaven min som krevde å jobbe *gjennom* problemløsning. Det å programmere er å finne en løsning til problemet gjennom refleksjon og finne ut hvordan en skal skrive dette til en kode gjennom et programmeringsspråk (Saeli m.fl., 2010). Gjennom mitt datamateriale viser jeg at elevene ikke reflekterer og lager seg en plan ut fra denne, for hvordan de kan løse dette som en kode.

5.2 Elevenes vurdering av egen læringsprosess

Som Hazzan m.fl. (2014) sier er det å kunne «debugge» viktig når en skal programmere, og derfor har jeg valgt å ta dette med som et eget trekk ved elevene. Det å «debugge» er feilsøking av koden og vurdering av egen læringsprosess, og det er viktig for alle som programmerer. Som programmerer er en nødt til å sjekke korrektheten i koden de har lagd for å dobbeltsjekke om det har skjedd en feil, uansett for hvilket nivå en er på. Denne prosessen kan også ses på som Polyas (1957) siste steg, og derfor vil det være en viktig del i problemløsningsprosessen.

Når eleven lager en kode som ikke fungerer sletter de den med en gang uten å diskutere den. De sjekker ikke om noe av det de har gjort er rett, eller om det kan brukes. De fjerner alt og begynner på nytt, som dialogen viser:

Hilde: «*vi må ta bort den [en kode]*»

Jeg: *Prøv også å se når dere lager en kode, og ikke får den til å fungere, hva er det da som er feil med den koden?*

Hilde: «*men vi tok den bort...*»

I matematikken er det også viktig å kunne sjekke sitt eget svar og sine egne utregninger for å kunne være sikker på at man har gjort rett, og det er spesielt viktig i en problemløsningsprosess. Vurdering av gyldigheten til strategien, stegene en har tatt og korrektheten i svaret gjør at elevene blir mer bevisst sin egen læringsprosess (Van De Walle m.fl., 2010). Elevene sjekket sine svar i matematikkøkten, og her kom systematikken til Hilde frem igjen. Hun var den som gikk grundigst gjennom svarene og dobbeltsjekkete sine svar. Bente og Steinar gjorde ikke dette like grundig, men sjekket i større grad korrektheten i matematikken fremfor i programmeringen.

5.2.1 Diskusjon rundt elevenes vurdering av læringsprosessen

Et prinsipp innenfor problembasert læring er at elevene er ansvarlige og delaktige i egen læringsprosess (Lambros, 2002). Gjennom dette må også elevene reflektere og vurdere sin egen læringsprosess. Dette blir sett på som viktig innenfor problembasert læring, og viktig for å utvikle sin problemløsningskompetanse. Som jeg viser i datamaterialet viser ikke de elevene jeg har studert en kompetanse til å feilsøke koden og vurdere sin læringsprosess. Hazzan m.fl. (2014) sier at å lære seg å feilsøke koden er av de første tingene en må lære seg når en skal begynne å programmere fordi det fremmer elevenes evne til å programmere. Når man går gjennom en kode og undersøker den nærmere vil de samtidig lære seg aspektet rundt hvordan å programmere bedre, og en vil bygge forståelse i motsetning til kun å følge en oppskrift. Refleksjon rundt om koden bør feilsøkes og hva som eventuelt må rettes på i koden er viktig. Når elevene gjør dette reflekterer de på nytt rundt programmeringsprosessen og valgene de har tatt gjennom denne (Hazzan m.fl., 2014:86). Feilsøking er en ferdighet for å få i gang

tankemåter rundt valg, og gir en mulighet for å forstå vår egen læringsprosess. Det kan tenkes at det er urealistisk å forvente at elevene skal klare å feilsøke når de ikke har fått opplæring i det. Men elevene skal gjennom vurdering for læring lære seg å vurdere egen læringsprosess. Jeg valgte ikke å lære dem hvordan de skal feilsøke koder grunnet liten tid, men poengterte at det var viktig å sjekke koden man hadde lagd.

Gjennom mangelen på feilsøking hos elevene ser jeg at de heller ikke vurderer sin problemløsningsprosess og sine valg av strategier. Å arbeide gjennom problemløsning fremmer en slik vurdering av prosessen og valg av strategier (Van De Walle m.fl., 2010). Dette forteller meg at det er viktig som lærer å fokusere på å lære elevene disse ferdighetene. Kanskje en gjennom grunnleggende ferdigheter har mulighet til å fremme nettopp dette, ettersom man som lærer har en mulighet til å jobbe mot at elevene skal utvikle seg problemløsende, både med tanke på strategier og arbeidsmåter, og ikke minst om læringsstrategier.

5.3 Elevenes deling, samarbeid og spørsmålsstilling

Noe jeg oppdaget underveis i analysearbeidet var at det var forskjell i delingskulturen mellom de to øktene. Når elevene jobbet i matematikken jobbet elevene mye mer lukket, og virket mindre villig til å dele sine strategier og tankemåter. Elevene forklarte kun hva de hadde tenkt hvis *jeg* spurte. Men når elevene programmerte var det en mer åpen dialog rundt hva de hadde forsøkt, og alle elevene gav hverandre tips uavhengig om jeg hadde stilt dem spørsmål eller ikke. Dette kan skyldes at elevene slet like mye med oppgavene, og at det ikke var kun én person som ikke fikk til. De jobbet også to og to som gjør at elevene ikke føler at de står alene med det fulle ansvaret om en feil. Samarbeidet innad i gruppene gikk veldig bra. Likevel er det interessant at det er en slik klar forskjell mellom øktene.

Elevene stilte få spørsmål både til meg og til medelever under programmeringsøkten. I starten var elevene opptatt av om *jeg* hadde klart oppgaven for å forsikre seg om at det fantes en løsning på den. Resten av økten spør elevene noen spørsmål til hverandre som omhandlet valg av brikker og deres forsøk på løsning. Steinar spurte blant annet Hilde om hvor de fant noen brikker han hører henne snakke om.

Steinar: «*Hvor fant dere den?*»

Hilde sier til Steinar og Bente: «*Vi fant det under data, der går det an å endre poeng, men det går ikke an å endre til fart*»

Dette hjelper Steinar og Bente på veien mot å prøve ut mer med variabler.

Elevene stilte nærmest ingen spørsmål under matematikkøkten. Elevene stilte ikke spørsmål til hverandre om hvordan de hadde løst oppgaven eller hvordan de tenkte. Steinar spør ikke de andre hvordan de løste oppgave 2. Da hadde både Hilde og Bente fått til oppgaven, og han virket litt forlegen for at han ikke klarte oppgaven.

5.3.3 Diskusjon rundt elevenes samarbeids- og delingsvilje

Gjennom analysen ser jeg at elevene har en større vilje til samarbeid og felles utforskning i programmeringen fremfor i matematikken. Det at ingen av elevene klarer oppgaven i programmeringen kan gjøre at det ikke er like «flaut» å spørre hva de andre har forsøkt, som i matematikken. Når en av gruppene i programmeringen trodde de hadde klart oppgaven, og sa dette høyt, ble ikke de andre elevene forlegne, men spurte med en gang hva de hadde gjort. Til forskjell fra matematikkøkten delte medelevene dette villig. Slik hjalp de hverandre videre hele veien. Ved at elevene samarbeider og diskuterer under programmeringen og ikke under matematikken, viser at Scratch er et program som bygger på og fremhever sosiale læringsteorier. Samarbeid og diskusjon i grupper er en arbeidsmetode som benyttes og fremheves gjennom problembasert læring (Pettersen, 2005). Et gruppearbeid skal fungere som et redskap for læring, både for den enkelte og for gruppen. Van De Walle m.fl (2010) trekker også frem viktigheten ved diskusjon og samtale rundt strategier, noe en lettere får til gjennom gruppearbeid. De viser også til at det er lærerens jobb å stimulere til slik samtale og diskusjon ved oppfølgingsspørsmål. Chapin mfl. (2014) sier at nettopp problemløsningsstrategier er et godt grunnlag for å drive en faglig diskusjon i klasserommet. En kan da samtale om løsningsstrategier og dette igjen kan avdekke hull i tidligere lærte konsepter i matematikken, og elevene får brukt språket som et medierende redskap. Elevene har startet med en viktig del av det å utvikle sine problemløsningsstrategier ved at de deler og diskuterer mellom hverandre. Videre er det opp til læreren å styre diskusjonen inn på mer rene strategivalg.

Vil det si at en slik ny arbeidsmåte er en mulighet for mer og bedre samarbeid? Det kan tenkes at elevenes holdning i matematikkøkten er en holdning som har blitt medbrakt fra den tradisjonelle matematikkundervisningen. Det er de samme elevene som jobber i begge øktene

men har likevel en signifikant forskjell i holdninger mellom matematikkøkten og programmeringsøkten. Holdninger i matematikken blir påvirket av faktorer som motivasjon, selvfølelse, utholdenhet, vilje til å arbeide i matematikk og matematikk angst. I tillegg blir holdningene påvirket av menneskene rundt, medelever, lærer og foresatte (Kjærnsli og Olsen, 2012). Holdningene spiller også en viktig rolle i forhold til prestasjonene. Elever med god holdning oppnår bedre resultater på PISA undersøkelsen (Kjærnsli og Olsen, 2012). Ettersom elevene ikke har programmert noe tidligere, og dette er nytt i dagens skole, har de ikke noe medbrakte holdninger inn i programmeringsøkten. Det er interessant at det er en så stor forskjell, og det ligger en mulighet for å benytte seg av programmeringen for å jobbe med elevenes holdninger i matematikken.

5.4 Oppsummering og videre diskusjon

Gjennom denne analysen har jeg belyst flere elementer som viser at elevene jeg har studert arbeider likt på områder rundt problemløsningsstrategier. De jobber likevel ulikt i arbeidsmåter når de arbeider med problemløsningsoppgave i programmeringen og i matematikken. Det som kommer tydelig frem i mitt datamateriale er at elevene ikke ser ut til å inneha kompetanse rundt det å jobbe problemløsende. Elevene ser ikke ut til å inneha kunnskaper om ulike strategier og fremgangsmåter. Dette bekreftet også læreren deres ved at hun informerte at elevene ikke har lært om problemløsningsstrategier.

Selv om jeg viser til flere situasjoner hvor elevene viser manglende evne på å jobbe problemløsende, spesielt med tanke på strategiene, vil jeg likevel si at elevene innehar en begynnende kompetanse i problemløsning. Både i matematikkøkten og i programmeringsøkten klarer elevene å jobbe seg frem til et svar, eller et forsøk på et svar. Det hadde vært interessant og undersøkt på nytt hvordan elevene hadde jobbet etter å ha lært både *om* problemløsning, i tillegg til *gjennom* problemløsning (Van De Walle m.fl., 2010). Det er lærerens ansvar å lære elevene *om* problemløsning, om arbeidsmetoder og ulike problemløsningsstrategier. Som jeg har trukket frem i teorien, i kapittel 3.4, spiller læreren også en viktig rolle i det å bevisstgjøre elevene underveis med spørsmål. Når elevene har lært seg hvordan å samtale rundt problemløsning og strategiene vil de bli mer bevisste og kan beherske egne strategier og løsninger bedre (Chapin m.fl., 2014). De får gjennom en slik samtale om tenkning rundt egne strategier og tankegang altså utviklet sin metakognitive ferdighet.

Gjennom programmeringen er en nødt til å forholde seg til feil i koder, forfi en må feilsøke etter eventuelle feil. I programmeringen er koden fremgangsmåten og algoritmen (Hazzan m.fl., 2014). I matematikken må elevene også forholde seg til feil, og bør vurdere og reflektere rundt valg av strategier. I en matematikktime vil ofte den typiske reaksjonen til et barn på et feil svar være å skjule og glemme det. Papert (1980:61) beskriver i hans arbeid at elevene ikke blir kritisert for feil i LOGO miljøet, og som programmerer blir man oppfordret til å studere feilen og lære av dem fremfor å glemme den. Dette kan gi en gyldig mulighet til å endre en holdning rundt feil som noe flaut en vil forsøke å skjule. Slik jeg ser i datamaterialet virket det mindre skummelt for elevene å dele og snakke om feil i programmeringen, fremfor i matematikken. Som lærer får man en mulighet til å fokusere på at en kan lære av feil. Det hadde vært interessant og sett om elevene etter å ha blitt kjent og vant med programmet ville oppført seg slik som i matematikken, eller om de ville opprettholdt fokuset på å dele og samarbeide med hverandre og ikke bli forlegen over feile svar.

Det jeg også ser i datamaterialet mitt er at elevene til en viss grad klarer å beherske programmet og programmeringsspråket Scratch. Programmet i seg selv virker bra for elevene, og de forstår hvordan det fungerer og hva de må gjøre. Likevel virker deres leting og innsetting av kodebrikker noe tilfeldig. Elevene sier selv at de ønsker mer kunnskap om hva de ulike kodebrikkene gjør, og litt mer generelt hva som er mulig å gjøre i programmet. Det kan tenkes at grunnen til at de ikke klarer å legge frem en god plan er grunnet for lite kjennskap til programmet og dens muligheter. Polya (1957) understreker viktigheten med at elevenes plan bygger på forståelse. Bortsett fra noe mangelfull kunnskap om de ulike kodebrikkene så jeg ikke noen andre problemer elevene støtte på gjennom arbeidet.

5.4.1 Presentasjon av data i en matrise

Som en del av analysearbeidet lagde jeg også en matrise over egnetheten til strategiene elevene benyttet seg av, se figur 7 på side 51. Matrisen er delt inn etter øktene, og etter egnethet av strategi. I Scratch er elevene delt etter de gruppene de jobbet i, og i matematikken er de delt. Fred var som nevnt syk under matematikkøkten og får derfor bare en strek under matematikkøkten. 1 i matematikken representerer oppgave en, mens 2 representerer oppgave to.

Det som kommer tydelig frem i denne matrisen er at elevene hadde en bedre evne til å velge en mer egnet strategi under matematikken fremfor i programmeringen. Selv om ingen av gruppene klarte å løse oppgaven i programmeringen slik den ba om, har jeg valgt å anse Bente og Steinar sin strategi med mange «gå» brikker som en uegnet men med løsning, selv om de ikke oppdaget dette selv. I matematikkøkten jobbet elevene problemløsende, men det kan hende matematikkoppgavene var for enkle for dem. Tabellen må ses med et kritisk blikk, men likevel er dette noe jeg har observert og oppfattet gjennom dette arbeidet. Det er tydelig at elevene trenger mer innføring i programmeringen, og ikke minst i problemløsningsstrategier. At elevene samarbeidet to og to kan også ha påvirket deres valg av strategi. Gjennom kapittel 5.3 viser jeg at elevene samarbeider mer, og har en mer åpen delingskultur gjennom programmeringsøkten, i forhold til i matematikkøkten. Det kan tenkes at ettersom elevene diskuterte og delte gode ideer mellom seg ble også deres læringsstrategier og valg av problemløsningsstrategi forbedret. Elevene hjalp hverandre med å motivere til å finne en løsning på problemet. Gruppearbeid er et grunnprinsipp i problembasert læring, og mine funn kan være en indikasjon på at problembasert læring er en god arbeidsmodell.

	Strategi i Scratch		Strategi i matematikk			
Grad av egnethet	Bente og Steinar	Fred og Hilde	Bente	Fred	Steinar	Hilde
Benytter en god strategi som gir en løsning på oppgaven			1	/	1	1,2
Benytter en god strategi men finner ikke noe løsning på oppgaven		x		/		
Benytter en uegnet strategi men finner løsning på oppgaven	x		2	/		
Benytter en uegnet strategi og finner ikke noe løsning på oppgaven	x	x		/	2	

Figur 7 Matrise av egnethet av strategi hos elevene i programmerings- og matematikkøkten. Forklaring av symboler: x står for strategi, 1 står for matematikkoppgave 1, 2 står for matematikkoppgave 2 og / står for fravær.

5.4.2 Refleksjoner

I etterkant av forskningen ser jeg at det er noe jeg kunne ha gjort annerledes. Jeg kunne ha gitt elevene en bedre og grundigere innføring av programmeringen, og blant annet lært dem å

feilsøke. Jeg valgte ikke å gjøre det fordi jeg ønsket også å undersøke om det var mulig å ta i bruk i skolen uten stor innføring. Jeg kunne også gitt elevene en gjennomgang av de fire stegene i en problemløsningsprosess slik at de var bedre rustet til å møte problemløsende oppgaver. Dette er ting jeg ser i ettertid av prosjektet, men som var vanskelig å forutse før jeg startet ettersom jeg ikke visste hva utfallet skulle bli. Jeg kunne også gjort et bedre grunnarbeid for å undersøke hvor elevene var faglig slik at jeg var mer sikkert på at matematikkoppgavene var problemløsende for dem.

Jeg kan ikke på bakgrunn av dette arbeidet generalisere eller trekke noen slutninger. Likevel tilfører jeg forskning på et felt som har blitt gjort lite forskning på i norsk skole. Mine funn samsvarer med annen forskning rundt problemløsning i programmering, samtidig som jeg setter et nytt fokus på dette med samarbeidsvilje hos elevene.

6.0 Konklusjon og videre arbeid

Jeg har gjennom dette arbeidet undersøkt på hvilken måte elevene jobber problemløsende når de jobber med en problemløsende oppgave i programmeringsspråket og programmet Scratch. Min undersøkelse har vært innenfor rammen av matematikkfaget, men handler grunnleggende om at elevene skal utvikle læringsstrategier i kunnskapssamfunnet og det og «lære å lære». Problemstillingen min er:

«Hvordan kan programmering påvirke læringsstrategiene til elevene?»

i tillegg til følgende forskningsspørsmål:

- *Hvilke problemløsningsstrategier benytter elevene seg av?*
- *Hvordan jobber elevene problemløsende når de programmerer?*
- *Kan programmet Scratch benyttes som en problemløsende oppgave?*

Det jeg har belyst gjennom dette prosjektet er at elevene jeg har undersøkt ikke kjenner til problemløsningsstrategier og en problemløsende jobbing. Gjennom analysen har jeg også presentert at elevene ikke reflekterer rundt arbeidsprosessen og sin læringsprosess. Likevel gjenkjenner jeg problemløsende arbeid, og elevene fikk benyttet seg av ulike læringsstrategier. Innenfor rammene dette prosjektet er gjennomført i, ser jeg at programmering kan påvirke læringsstrategiene til elevene. Det er likevel med noen mangler, og derfor må dette undersøkes videre. For de elevene jeg har arbeidet med fungerte programmet Scratch som et problemløsende verktøy, men elevene trenger å lære seg flere problemløsningsstrategier. Elevene var mer villige til å samtale og dele gjennom programmeringsøkten i forhold til matematikkøkten. Hvis programmeringen gir en mulighet for å hjelpe elevene til å samtale rundt strategier og tankemåter er dette en positiv indikasjon som jeg mener er en grunn til at dette bør undersøkes videre. Det bør også undersøkes om dette gjelder kun en programmeringsaktivitet eller om det gjelder alle mer «uformelle» oppgaver knyttet til matematikken. Jeg kan ikke generalisere og si at dette gjelder for flere enn de elevene jeg har undersøkt.

Det jeg har presentert gjennom kapittel 5.0, er blant annet at elevenes planer var veldig fokusert på målet og svaret i seg selv, og ikke om veien til målet. De reflekterte ikke over hva de hadde forsøkt, og prøvde dermed ut samme kode flere ganger, blant annet med flere «gå» brikker. Det at elevene ikke tok seg tid til å vurdere sin egen læringsprosess, hverken i matematikken eller i programmeringen, kan være en indikasjon på at norsk skole ikke har

lyktes med vurdering for læring som ble satt i fokus når *Kunnskapsløftet* 2006 ble innført og begrepet vurdering for læring ble tatt i bruk (Dale, 2010). Vurdering for læring er «*en planlagt prosess der informasjon om elevenes kompetanse brukes av både læreren og elevene slik at 1) læreren kan tilpasse undervisningen, og 2) elevene kan justere sine egne læringsstrategier*» (Slemmen, 2011:53). Elevmedvirkning og grunnlag for tilpasset opplæring er to viktige prinsipper i vurdering for læring. Det kan tenkes at skolen jeg har gjort mine undersøkelser innenfor driver med formativ vurdering, men slik jeg opplevde det er ikke elevene vant med elevmedvirkning i vurdering av sin læringsprosess som grunnlag for å justere egne læringsstrategier. Elevene må gjennom en slik vurderingsprosess vurdere seg selv, og reflektere og begrunne sine valg. Når elevene er opptatt av målet og korrekt svar fremfor veien og arbeidsprosessen er dette et argument for at elevene ikke har jobbet systematisk med problemløsende arbeid. Hvis elevene har vært vant med en summativ form for vurdering har de blitt vant med en tankegang rundt at svaret er det viktige, og ikke prosessen frem til svaret.

Det er den formative vurderingen for læring som jeg ser er i tråd med problembasert læring. I den summative vurderingen med en karakter eller poengsum blir ikke elevenes ansvar og initiativ for læring vurdert eller premiert (Bjørke, 2000:135). Gjennom en summativ vurdering blir heller ikke elevenes kreative-, samarbeids- og problemløsningsevne vurdert ettersom det er vanskelig å «tallfeste» en slik ferdighet. Ettersom elevene må være delaktige i egen læringsprosess, og dette er viktig i problembasert læring, må dette også lyse gjennom i vurderingen for læringen. Målet med vurdering for læring er at elevene skal få en underveisvurdering som bidrar til å fremme læring (Utdanningsforbundet, u.å.). Her blir elevmedvirkning lagt vekt på, ved at elevene selv skal være med på å vurdere sin egen læringsprosess. Hvis lærerne ikke oppfordrer til dette, vil det være urealistisk å kreve av elevene at de skal kunne reflektere og vurdere sin egen læringsprosess i en problemløsningsprosess. Slik vurderingsforskriften viser er den formative vurderingen å betrakte som læringsfremmede, mens den summative vurderingen er for å sjekke at elevene har oppnådd kompetansen ved avslutningen av faget (Lovdata, 2009).

«*For å skape en læringskontekst der elevene kan bidra til å skape egne utfordringer, kreves en lærerrolle og en opplæring som fremmer deltakelse, aktivitet og samspill.*» (Meld. St. 30, 2003-2004:16). For å innføre en mer problemløsende jobbing kreves det mye av læreren, og som lærer har man et stort ansvar for å lære elevene opp i denne arbeidsmåten, både med tanke på strategier og metoder. Lambros (2002) peker på fokus på flere muligheter for løsning

fremfor rett svar som den mest signifikante fordel med problemløsende arbeid. En problemløsende arbeidsmetode kan dermed være en god måte for å få elevene mer deltagende og mer vant til den formative vurderingen ved at de må vurdere seg selv og sitt arbeid. Schoenfeld (2013:20) understreker også at ved å jobbe med problemløsning kan man oppdage typiske utviklingsbaner hos elevene, som kan være til hjelp til å utvikle læreplaner som bygger på utvikling.

I den generelle delen av læreplanen blir det presentert sju forskjellige mennesketyper som elevene i skolen skal utvikle gjennom grunnskolen (Utdanningsdirektoratet, 2011). Tre av disse mener jeg underbygger bruken av programmering som problembasert læring i skolen. *Det skapende mennesket* legger det vekt på at elevene får trening i å tenke, trekke slutninger, argumentere og utvikle kreative evner. *Det arbeidende mennesket* går ut på at elevene blir klar for arbeidslivet de skal møte i fremtiden, og her legges det vekt på bruk av teknologi og egen læring. I *det samarbeidende mennesket* skal elevene utvikle de sosiale og samarbeidene evner gjennom samarbeid med andre. Elevene skal gjennom grunnskolen ikke bare bli påført mer kunnskap, men være med i en prosess hvor de lærer seg å lære, og får kompetanse til å møte fremtiden (Utdanningsdirektoratet, 2011). Hvis elevene også gjennom problemløsende arbeid kan lære seg dybdekunnskap fremfor overflatekunnskap som Ludvigsenutvalget anbefaler (NOU, 2014), er en problembasert læringsmodell noe å se på for fremtidens skole.

6.1 Videre arbeid

Angående videre forskning på dette temaet hadde det vært interessant hvis en så nøyer på hvilken læring som ligger i programmering i Scratch, også utenfor evnen til problemløsning. Det bør også da undersøkes nærmere på hvilke muligheter som ligger innenfor programmet Scratch.

Jeg har sett på problemløsning innenfor Scratch, på bakgrunn av en nysgjerrighet mye grunnet andre land sin innføring av programmering på timeplanen. Det hadde vært spennende å undersøkt hvorfor og på hvilken måte andre land har implementert programmering i læreplaner, og om dette bør gjøres i norsk skole.

Som en implikasjon av mitt arbeid så jeg også at elevene tydelig ikke var vant med å samtale omkring egen læringsprosess som gjør at jeg spør om skolen har lyktes med vurdering *for* læring. Elevenes evne til en slik refleksjon, og hvordan de blir lært opp til dette er noe som

bør undersøkes videre. Når problembasert læring som arbeidsmetode også inneholder mange av prinsippene fra vurdering for læring mener jeg dette er en arbeidsmetode som også bør undersøkes nærmere og prøves ut i skolen i større grad.

Litteraturliste

- Alseth, B. (1998) *Matematikk på småskoletrinnet*. Læringscenteret, Oslo
- Alseth, B. og Røsseland, M. 2014. *Undersøkelleslandskap i matematikk*. In: Frislid, M. E. og Traavik, H. (eds.). *Lese, skrive, regne - Pedagogikk og fagdidaktikk i begynneropplæringen*. 2. utgave ed. Oslo: Universitetsforlaget, pp. 109-131.
- Befring, E. (2007). *Skolen for barnas best- oppvekst og læring i eit pedagogisk perspektiv*. Det Norske Samlaget, Fagernes
- Bjorklund, D. (1990) *Children's Strategies, Contemporary Views of Cognitive Development*. Lawrence Erlbaum Associates, Inc. Publishers, New Jersey
- Bjørke, G. (2000) *Problembasert læring, Ei innføring for profesjonsutdanningane*. Universitetsforlaget, Stamsund.
- Bjørndal, C. R. P. (2009). *Det vurderende øyet*. (1. utgave). Gyldendal akademisk, Oslo
- Brekke, M., & Tiller, T. (2013). *Læreren som forsker*. Universitetsforlaget AS, Oslo
- Brown, Q., Mongan, W., Kusic, D., Garbarine, E., Fromm, E. og Fontecchio A. (U.å). *Computer Aided Instruction as a Vehivcle for Problem Solving: Scratch Programming Environment in the Middle Years Classroom*. <url>
http://www.pages.drexel.edu/~dmk25/ASEE_08.pdf Lest 02.03.15
- Bø, I. og Helle, I. (2010) *Pedagogisk ordbok, praktisk oppslagsverk i pedagogikk, psykologi og sosiologi*. 2. utgave. Universitetsforlaget, Oslo
- Chapin, S. H., O'Connor, C., & Anderson, N. C. (2009). *Classroom discussions: using math talk to help students learn, grades K-6*. Sausalito, Calif.: Math Solutions.
- Christoffersen, Line og Johannessen, Asbjørn. 2012. *Forskningsmetode for lærerutdanningene*. Abstrakt forlag, Oslo
- Dale, E. L. (2010). *Kunnskapsløftet-på vei mot felles kvalitetsansvar?*. Universitetsforlaget AS, Oslo
- Fessakis, G., Gouli, E., og Mavroudi, E. (2012). *Problem solving by 5-6 years old kindergarten children in a computer programming environment: A case study*. *Computer & Education* 63 (2013) s. 87-97.
- Hattie, J. og Yates, G. (2014) *Synlig læring – hvordan vi lærer*. Cappelen Damm, Oslo.

- Hazzan, O., Lapidot, T. og Ragonis, N. (2014) *Guide to Teaching Computer Science, An Activity-Based Approach*. Second Edition. Springer-Verlag, London
- Illeris, K. (2012) *Læring*. Gyldendal Norsk Forlag, Oslo
- Imsen, G. (2012) *Elevenes verden , Innføring i pedagogisk psykologi*. 4. utgave, 5. opplag. Universitetsforlaget, Oslo
- Jensen, R. og Aas, M. (2011) *Å utforske praksis Grunnskolen*. Cappelen Damm Akademisk, Oslo
- Kafai, Y. B. og Burke, Q. (2014) *Connected Code Why Children Need to Learn Programming*. MIT Press, Massachusetts
- Kalelioglu, F., og Gülbahar, Y. (2014) *The Effects of Teaching Programming via Scratch on Problem Solving Skills: A discussion from Learners Perspective*. Informatics in Education, 2014, Vol. 13, No. 1, s. 13-50. Vilnius University
- Karlsdottir, R. og Hybertsen, I. (2013) *Læring – Utvikling – Læringsmiljø, En innføring i pedagogisk psykologi*. Akademika forlag, Trondheim
- Kjærnsli, m. og Olsen, R. (2012) *Fortsatt en vei å gå Norske elevers kompetanse i matematikk, naturfag og lesing i PISA 2012* <url> <http://www.uv.uio.no/ils/forskning/prosjekt-sider/pisa/publikasjoner/publikasjoner/fortsatt-en-vei-a-ga.pdf> Lest 01.05.15
- Koschmann, T. (1996) *Paradigm Shifts and Instructional Technology*. Book Chapters, Paper 4. <url> http://opensiuc.lib.siu.edu/meded_books/4 Lest 04.05.15
- Krumsvik, R. J. (2014) *Forskningsdesign og kvalitativ metode – ei innføring*. Fagbokforlaget, Bergen
- Krumsvik, R. J. og Säljö, R. (2013) *Praktisk-pedagogisk utdanning, En antologi*. Fagbokforlaget, Bergen
- Kvale, S., Brinkmann, S. 2009. *Det kvalitative forskningsintervju*. Gyldendal akademisk, Oslo
- Lambros, A. (2002) *Problem-Based Learning in k-8 Classrooms A Teacher's Guide to Implementation*. Crown Press, Inc., Thousand Oaks California
- Lovdata (2009) *Forskrift til vurdering i skolen*. <url> https://lovdata.no/dokument/SF/forskrift/2006-06-23-724/KAPITTEL_4-1#%C2%A73-2 Lest 28.04.15

- Lyngsnes, K., & Rismark, M. (2009). *Didaktisk arbeid*. (2. utgave). Gyldendal Norsk Forlag AS, Oslo
- Løwing, M., og Kilborn, W. (2002). *Baskunnskaper i matematik för skola, hem och samhälle*. Studentlitteratur AB, Lund.
- Lær Kidsa Koding (u.å.) *Om LKK* <url> <http://www.kidsakoder.no/om-lkk/> lest 24.03.15
- Matematikksenteret (2014) *Teoretisk bakgrunnsdokument for arbeid med regning på ungdomstrinnet*. <url> http://www.udir.no/Upload/Ungdomstrinnet/Rammeverk/Ungdomstrinnet_Bakgrunnsdokument_regning_vedlegg_2.pdf Lest 05.05.15
- Matematikk.org (u.å) *Tekstnøtt: Husnummer*. Hentet fra: <https://www.matematikk.org/trinn5-7/tekstnott.html?tid=105107> Lest 04.01.15
- Meld. St. 30 (2003-2004) (2004) *Kultur for læring*. Utdannings- og forskningsdepartementet
- MIT (Massachusetts Institute of Technology) Media Lab, Lifelong Kindergarten Group (u.å.) *Scratch* <url> <https://scratch.mit.edu/>
- NOU 2014:7 (2014) *Elevenes læring i fremtidens skole*. Departementenes sikkerhets- og serviceorganisasjon, Informasjonsforvaltning, Oslo
- Papert, S. (1980) *Mindstorms. Children, Computers and Powerful Ideas*. Basic Books, Inc. Publishers, New York
- Pettersen, R. C. (2005) *PBL problembasert læring for studenten*. 2. utgave. Universitetsforlaget, Oslo.
- Polya, G. (1957) *How to solve it, A new aspect of mathematical method*. Second edition. Ishie Press international, Bronx NY
- Postholm, M. B. (2005) *Kvalitativ metode: en innføring med fokus på fenomenologi, etnografi og kasusstudier*. Universitetsforlaget, Oslo
- Postholm, M. B., & Moen, T. (2009). *Forsknings- og utviklingsarbeid i skolen*. Universitetsforlaget AS, Oslo
- Rossen, Eirik. Store norske leksikon, *Programmering-IT*. <url> <https://snl.no/programmering%2FIT> Lest 01.05.15
- Røsseland, Mona (2005): *Hva er matematisk kompetanse?* Tangenten, blad 1 og 2.

- Saeli, M., Perrenet, J., Jochems, W. M. G. og Zwaneveld, B. (2011). *Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective*. Informatics in Education, 2011, Vol. 10, No. 1, s. 73-88
- Schoenfeld, A. H. (2013) *Reflections on Problem Solving Theory and Practice*. The Mathematics Enthusiast, Vol. 10, nos. 1 & 2, s.9-34
- Slemmen, T. (2011) *Vurdering for læring i klasserommet*. 2. Utgave. Gyldendal akademisk, Oslo.
- Sweller, J. (1988) *Cognitiv Load During Problem Solving: Effects on Learning*. Cognitive Science 12, s. 257-285
- Taylor, M., Harlow, A. og Forret, M. (2010) *Using a Computer Programming Environment and an Interactive Whiteboard to Investigate Some Mathematical Thinking*. Procedia Social and Behavioral Sciences 8 (2010) s. 561-570.
- Utdanningsdirektoratet (2011). *Generell del av læreplanen*. <url> <http://www.udir.no/Lareplaner/Kunnskapsloftet/Generell-del-av-lareplanen/> Lest 28.04.15.
- Utdanningsdirektoratet (2012¹). *Prinsipp for opplæringa*. <url> <http://www.udir.no/Lareplaner/Kunnskapsloftet/Prinsipp-for-opplaringa/Motivasjon-for-laring-og-laringsstrategiar/> Lest 04.05.15
- Utdanningsdirektoratet (2012²). *Rammeverk for grunnleggende ferdigheter*. <url> http://www.udir.no/Upload/larerplaner/lareplangrupper/RAMMEVERK_grf_2012.pdf?epslanguage=no Lest 04.05.15
- Utdanningsdirektoratet (2013¹) *Føremål i læreplan i matematikk fellesfag* <url> <http://www.udir.no/kl06/MAT1-04/Hele/Formaal/> Lest 24.04.15
- Utdanningsdirektoratet (2013²) *Læreplan i matematikk fellesfag* <url> <http://www.udir.no/kl06/MAT1-04/> 2013² Lest 25.04.15
- Utdanningsdirektoratet (2013³) *Grunnleggende ferdigheter matematikk fellesfag* <url> http://www.udir.no/kl06/MAT1-04/Hele/Grunnleggende_ferdigheter/ Lest 02.02.15
- Utdanningsdirektoratet (u.å). *Vurdering for læring*. <url> <http://www.udir.no/Vurdering-for-laring/Hva-er-Vurdering-for-laring/Hva-er-vurdering-for-laring/> Lest 29.04.15

Van De Walle, John A., Karp, Karen S. Bay-Williams, Jennifer M. (2010) *Elementary & middle school mathematics*. Pearson, USA

Wadel, Cato (2006). *Forskning i egne erfaringer*. SEEK as, Flekkefjord

Wikipedia (2015) *Scratch (programming language)* <url>
http://en.wikipedia.org/wiki/Scratch_%28programming_language%29#cite_note-7

Vedlegg 1 Forklaring om programmering

SCRATCH



Scratch er et programmeringsspråk som gjør det enkelt å lage dine egne interaktive eventyr, spill og animasjoner - og så dele det med andre på internett. Dette dokumentet er en liten introduksjon til hva dataprogrammering egentlig er og litt forklaring til noen begreper innenfor Scratch som du må kunne når du skal programmere. Disse begrepene finnes ikke bare i Scratch, men faktisk i alle andre programmeringsspråk også.

Du benytter deg sannsynligvis av dataprogrammer hver dag, hverken du vet det eller ikke. Noen finner du på datamaskinen din, som for eksempel Word eller Minecraft og noen ser du ikke i det hele tatt. Eksempler på skjulte dataprogrammer er programmer som styrer trafikklys, autopilot til fly, og mye, mye mer. Kort sagt så kan et dataprogram gjøre akkurat det du vil gjøre, for bak hvert eneste dataprogram som finnes så er det et menneske som har skrevet det.

Når et menneske skriver et program til en datamaskin, så kan hun jo ikke skrive på norsk. Heller ikke på engelsk. Datamaskiner snakker jo ikke norsk! De språkene datamaskiner kan heter **programmeringsspråk**, og det er disse språkene dataprogrammer er skrevet i. Scratch er et eksempel på et programmeringsspråk. Andre kjente programmeringsspråk er Java, Python og C.

Men hva er egentlig et dataprogram? Et dataprogram er en slags oppskrift som datamaskinen skal følge. Men siden datamaskinen ikke kan tenke så må oppskriften være så nøyaktig at det ikke *kan* gå galt. Jobben til en programmerer er å skrive oppskrifter som datamaskiner må følge til punkt å prikke. For selv om datamaskiner ikke kan tenke selv, så er de utrolig flinke til å gjøre akkurat det vi mennesker har bedt dem om å gjøre.

Ord	Forklaring
Skript og Skriptområdet	En skript er det som er selve koden i programmet, og som får noe til å skje. Dette skriver du i skriptområdet. All programmeringen ligger her i dette feltet.
Løkke	En løkke brukes for å gjøre noen skal skje flere ganger. I Scratch så finner vi alle løkkene under Styring-kategorien. Hvis vi vil at noe skal gjentas et bestemt antall ganger, så bruker vi en gjenta (10) ganger-blokk. En annen viktig type løkke heter for alltid. I stedet for å gjenta noe et bestemt antall ganger, så gjentar den noe <i>i evigheten</i> . Vi kan altså bruke det som heter en løkke for å be datamaskinen gjøre den samme oppgaven så mange ganger vi måtte ønske.
Variabel	I de aller fleste programmer så har man bruk for å lagre informasjon, for så å ha den lett tilgjengelig for senere bruk. Da bruker man en variabel for å holde på den informasjonen. I Scratch så finner du alt som har med variabler å gjøre under Data. For å lage en ny variabel klikker du på Lag en Variabel. I spill kan variable være nyttige for å holde styr på poeng, tid, tur, hastighet, vinner og mye mer.
Figurer/Ny figurer/ Drakter/	Du kan ha så mange figurer i prosjektet ditt som du vil. Du kan enkelt hente inn flere fra biblioteket til Scratch, eller laste opp fra din datamaskin. Du kan også tegne figurer. Inn under fanen drakter på skriptområdet kan du endre drakter til en figur. Det betyr at du kan endre litt av utseende på figuren og få det til å se ut som den for eksempel går.
Rotasjonsmåte	For hver figur bestemmer du rotasjonsmåten ved å trykke på «i» oppe til venstre på figuren. Dette betyr om figuren ser til høyre og venstre, eller om

	den bare ser en vei og dermed havner opp ned når den skal snu ved kanten.
Sende beskjed	Inn under kategorien «hendelser» kan du sende en beskjed mellom figurene, slik at de på en måte snakker med hverandre. Da kan en figur si noe til den andre slik at noe skjer.
Ryggsekk	Her kan du legge en skript(kode) som du har laget og lagre den. Du kan hente dem opp igjen til andre figurer, og også andre prosjekter.

Vedlegg 2 Observasjonspunkter

Observasjon for både programmeringen og matematikken

- Hvilke strategier de bruker når de skal løse oppgavene
- Om elevene forstår programmet/ problemoppgavene
- Om elevene foreslår forskjellige måter for å skrive programmet/å løse problemoppgavene
- Om elevene lett kan skrive programmet/lett forstår problemoppgavene
- Om elevene klarer å lage en plan for løsningen til programmet/ problemoppgavene
- Om elevene klarer å observere utkommet av programmet/ problemoppgavene
- Om elevene klarer å sjekke om koden er rett/om regningen til problemoppgavene er rett, og hva som eventuelt er feil
- Om elevene lett klarer å løse oppgavene gikk i Scratch (de uten fasit).
- Om elevene klarer å løse problemene som oppstår når de skriver koden/regner problemoppgavene
- Hvilke type spørsmål elevene stiller meg som lærer eller til medelever– er det i forhold til et problem
- Hvordan elevene samarbeider
- Hvordan elevene deler strategier/ideer
- Hva syntes elevene om programmeringen?(smil, latter, engasjement)

Vedlegg 3 Felix og Herbert- oppgave til steg 1

1.1

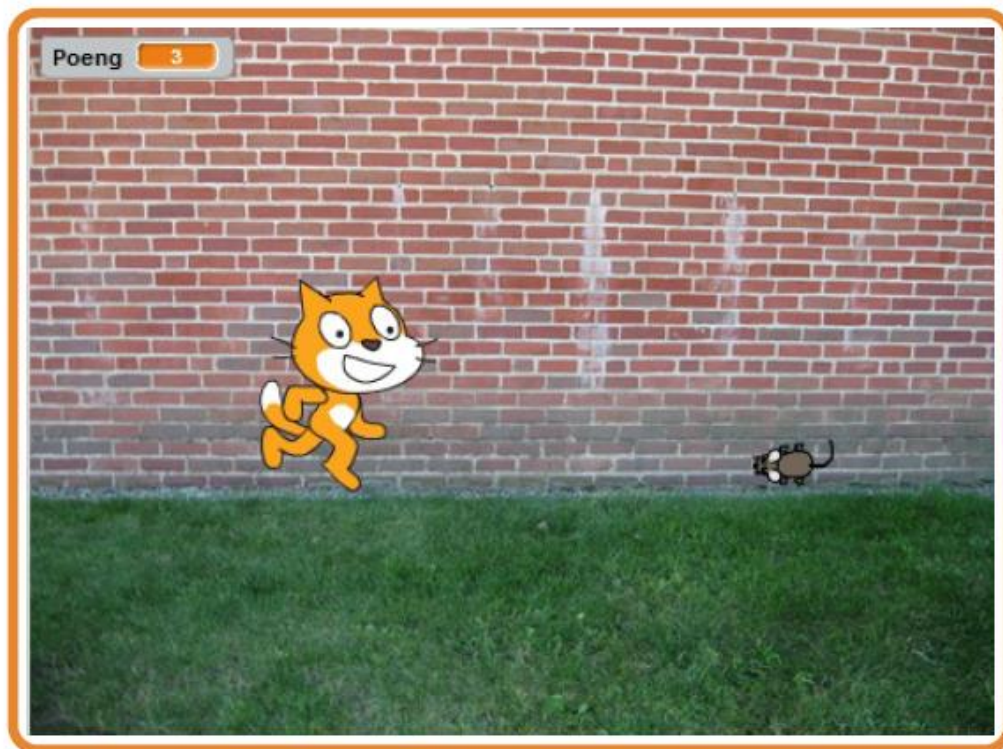
Felix og Herbert



Dette kurset er utviklet av [Code Club](http://www.codeclub.org.uk/) (http://www.codeclub.org.uk/) i England, og oversatt til norsk av [Kodeklubben](http://www.kodeklubben.no/) (http://www.kodeklubben.no/).

Introduksjon

Vi skal lage et spill hvor katten Felix skal fange musa Herbert. Du styrer Herbert med musepekeren og skal prøve å unngå å bli tatt av Felix. Jo lenger du unngår ham jo flere poeng får du, men blir du tatt, går poengsummen din ned.



Steg 1: Felix følger musepekeren


Vi ønsker at katten Felix skal følge etter musepekeren.





Start et nytt prosjekt.




Trykk på **i** i hjørnet av  og bytt navn på figuren til Felix .



Sørg for at Felix kun ser til høyre og venstre ved å sette rotasjonsmåte til \leftrightarrow .



Klikk på scenen ved siden av Felix i vinduet for figurer. Velg fanen **Bakgrunner** og trykk på  for å importere en ferdig bakgrunn. Velg den bakgrunnen du vil.



Velg **Skript**-fanen og lag dette skriptet:



Test prosjektet

Klikk på det grønne flagget.

- Følger Felix musepekeren?
- Ser det ut som han går når han beveger seg?
- Beveger han seg med riktig hastighet?



Lagre prosjektet

Scratch lagrer alle prosjektene dine automatisk med jevne mellomrom. Det kan likevel være lurt å passe på at prosjektet faktisk er lagret innimellom.




- I filmenyen, velg **Lagre nå** .

Steg 2: Felix jager Herbert

Nå ønsker vi at Felix skal jage musa Herbert i stedet for musepekeren.



Sjekkliste

- Lag en ny figur ved å trykke på  og velg figuren Dyr/Mouse1 .
- Bytt navn på figuren til Herbert og sørg for at også Herbert kun kan se til høyre og venstre,  .
- Gjør Herbert mindre enn Felix ved å trykke på  (øverst mot midten av vinduet). Prøv seks klikk.
- Gi Herbert dette skriptet:



Test prosjektet

Klikk på det grønne flagget.

- Flytter Herbert seg med musepekeren?
- Jager Felix Herbert?

Steg 3: Felix sier når han har fanget Herbert

Vi vil at Felix skal vite når han har fanget Herbert og fortelle det til oss.

Sjekkliste

- Endre skriptet til Felix til dette:



Test prosjektet

Klikk på det grønne flagget.

- Sier Felix fra når han har fanget Herbert?

Steg 4: Herbert blir et spøkelse når han fanges

I stedet for at Felix sier noe, vil vi at Herbert blir forvandlet til et spøkelse når han fanges.



Sjekkliste

- Endre skriptet til Felix slik at det sender en melding og lager en lyd når han fanger Herbert

```

når flagg klikkes
for alltid
  pek mot musepeker
  gå 10 steg
  neste drakt
  trommeslag 3 som varer 0.25 takter
  hvis berører Herbert?
    send melding Fanget!
    trommeslag 1 som varer 0.25 takter
    si Tok deg! i 1 sekunder
    vent 1 sekunder

```

- Velg Herbert og gå til Drakter-fanen.
- Hent en ny drakt ved å trykke på  og velg Fantasi/ghost2-a
- Gjør drakten mindre ved å velge  og trykke seks ganger på spøkelsesdrakten.
- Endre navnene på Herberts drakter slik at musedrakten heter levende og spøkelsesdrakten heter død .
- Lag et nytt skript for Herbert for å gjøre ham om til et spøkelse. Ikke slett det gamle skripte

```

når jeg mottar Fanget!
  bytt drakt til død
  vent 0.5 sekunder
  bytt drakt til levende

```

Test prosjektet

Klikk på det grønne flagget.

- Forvandles Herbert til et spøkelse når han fanges?
- Spiller Felix de riktige lydene til riktig tid?
- Står Felix stille lenge nok til at Herbert kommer seg unna?

Steg 5: Telle poeng

La oss legge til en poengsum slik at vi kan se hvor flink man er til å holde Herbert i live. Vi begynner med poengsummen null og øker den med en for hvert sekund. Hvis Felix fanger Herbert, minker vi poengsummen med ti.

Sjekkliste

- På Skript-fanen under kategorien **Data**, lag en ny variabel. La den gjelde for alle figurer og kall den Poeng.



Ny variabel

Variabelnavn:

For alle figurer For denne figuren

Nett variabel (lagret på nett)

- Lag disse to skriptene på scenen:



Test prosjektet

Klikk på det grønne flagget.

- Øker poengsummen med en hvert sekund?
- Går poengsummen ned med ti når Herbert blir fanget?
- Hva skjer når Herbert fanges før du har ti poeng?
- Går poengsummen tilbake til null når du starter spillet på nytt?

Lagre prosjektet

Du er ferdig. Godt gjort. Nå kan du spille spillet!

Husk at du kan dele spillet med familie og venner ved å trykke [Legg ut](#) på menylinjen.

Dette kurset er utviklet av [Code Club](http://www.codeclub.org.uk/) i England, og oversatt til norsk av [Kodeklubben](http://www.kodeklubben.no/).

Vedlegg 4 Spørsmål til elever

Spørsmål til gruppeintervjuet:

- Hva syntes dere om det vi har jobbet med nå?
- Hadde du vanskeligheter med å løse oppgaven/problemet i Scratch?
- Hvordan gikk dere frem når dere løste oppgaven?
- Fikk dere noen problemer underveis?
- Kunne du løse problemet på forskjellige måter?
- Hva er det beste med Scratch? / hva er det verste?
- Syntes du Scratch var vanskelig eller enkelt å bruke?
- Liker du programmering?
- Vil du lære mer om programmering? Hva vil du da lære?
- Kan du si noe om hva du har lært?

Spørsmål stilt underveis

- Hva gjorde du som hjalp deg å forstå problemet?
- Hvordan bestemte du deg for hva du skulle gjøre?
- Tenkte du på svaret ditt etter du fikk det?
- Hvordan fant du ut at svaret ditt var korrekt?
- Prøvde du noe annet som ikke fungerte? Hvordan fant du ut at det ikke kom til å fungere?
- Kan noe du gjorde i denne oppgaven hjelpe deg til å løse andre problemløsningsoppgaver?

Vedlegg 5 Godkjenning fra NSD

Norsk samfunnsvitenskapelig datatjeneste AS

NORWEGIAN SOCIAL SCIENCE DATA SERVICES



Harald Hårfagres gate 29
N-5007 Bergen
Norway
Tel: +47-55 58 21 17
Fax: +47-55 58 96 50
nsd@nsd.uib.no
www.nsd.uib.no
Org.nr. 985 321 884

Karin Rørnes

Institutt for lærerutdanning og pedagogikk UiT Norges arktiske universitet

9006 TROMSØ

Vår dato: 04.11.2014

Vår ref: 40374 / 3 / SSA

Deres dato:

Deres ref:

TILBAKEMELDING PÅ MELDING OM BEHANDLING AV PERSONOPPLYSNINGER

Vi viser til melding om behandling av personopplysninger, mottatt 20.10.2014. Meldingen gjelder prosjektet:

40374 *Koding som digital grublis*

Behandlingsansvarlig *UiT Norges arktiske universitet, ved institusjonens øverste leder*

Daglig ansvarlig *Karin Rørnes*

Student *Susanne Iversen*

Personvernombudet har vurdert prosjektet og finner at behandlingen av personopplysninger er meldepliktig i henhold til personopplysningsloven § 31. Behandlingen tilfredsstiller kravene i personopplysningsloven.

Personvernombudets vurdering forutsetter at prosjektet gjennomføres i tråd med opplysningene gitt i meldeskjemaet, korrespondanse med ombudet, ombudets kommentarer samt personopplysningsloven og helseregisterloven med forskrifter. Behandlingen av personopplysninger kan settes i gang.

Dokumentet er elektronisk produsert og godkjent ved NSDs rutiner for elektronisk godkjenning.

Avdelingskontorer / District Offices:

OSLO: NSD, Universitetet i Oslo, Postboks 1055 Blindern, 0316 Oslo. Tel: +47-22 85 52 11. nsd@uio.no

TRONDHEIM: NSD, Norges teknisk-naturvitenskapelige universitet, 7491 Trondheim. Tel: +47-73 59 19 07. kyrre.svarva@svt.ntnu.no

TROMSØ: NSD, SVF, Universitetet i Tromsø, 9037 Tromsø. Tel: +47-77 64 43 36. nsdmaa@sv.uit.no

Det gjøres oppmerksom på at det skal gis ny melding dersom behandlingen endres i forhold til de opplysninger som ligger til grunn for personvernombudets vurdering. Endringsmeldinger gis via et eget skjema, <http://www.nsd.uib.no/personvern/meldeplikt/skjema.html>. Det skal også gis melding etter tre år dersom prosjektet fortsatt pågår. Meldinger skal skje skriftlig til ombudet.

Personvernombudet har lagt ut opplysninger om prosjektet i en offentlig database, <http://pvo.nsd.no/prosjekt>.

Personvernombudet vil ved prosjektets avslutning, 01.12.2015, rette en henvendelse angående status for behandlingen av personopplysninger.

Vennlig hilsen

Katrine Utaaker Segadal

Sondre S. Arnesen

Kontaktperson: Sondre S. Arnesen tlf: 55 58 33 48

Vedlegg: Prosjektvurdering

Kopi: Susanne Iversen susanneiversenuit@hotmail.com

PERSONVERNOMBUDET FOR FORSKNING



Utvalget informeres skriftlig om prosjektet og samtykker til deltakelse. Informasjonsskrivet er godt utformet.

Personvernombudet legger til grunn at forsker etterfølger UiT Norges arktiske universitet sine interne rutiner for datasikkerhet. Dersom personopplysninger skal lagres på mobile enheter, bør opplysningene krypteres tilstrekkelig.

Forventet prosjektslutt er 01.12.2015. Ifølge prosjektmeldingen skal innsamlede opplysninger da anonymiseres. Anonymisering innebærer å bearbeide datamaterialet slik at ingen enkeltpersoner kan gjenkjennes. Det gjøres ved å:

- slette direkte personopplysninger (som navn/koblingsnøkkel)
- slette/omskrive indirekte personopplysninger (identifiserende sammenstilling av bakgrunnsopplysninger somf.eks. bosted/arbeidssted, alder og kjønn)
- slette lydopptak

Vedlegg 6 Informasjonsskriv med svarslipp

Forespørsel om deltakelse i forskningsprosjektet

”Koding som digital grublis”

Bakgrunn og formål

Prosjektet er et mastergradsprosjekt i master i lærerutdanning 1.-7. ved institutt for lærerutdanning og pedagogikk, UiT Norges arktiske universitet. Formålet er å undersøke hvordan programmering kan legge til rette for problemløsning i matematikken. Jeg skal undersøke hvordan man kan legge til rette for at elever gjennom digitale ferdigheter kan få erfaring med problemløsning. Videre vil jeg undersøke om elever utnytter strategiene fra programmering for å løse matematiske problem. Foreløpig problemstilling er: «Kan programmering fungere som en digital grublis?»

Grunnen til at jeg ønsker å ha deg/ditt barn med i prosjektet er fordi jeg ønsker å undersøke elever som ikke har mye erfaring med programmering fra før av.

Hva innebærer deltakelse i studien?

Deltakelse i studiet innebærer at jeg skal introdusere programmering (gjennom et program som heter Scratch) til elevene i full klasse. Dette vil ta ca. 1-2 skoletimer, og skal foregå i skoletiden. Elevene skal jobbe to og to sammen med oppgavene. Jeg skal observere elevene mens de jobber, gjennom video-, lydopptak og notater. Grunnen til at jeg bruker videoopptak er for å kunne samle inn mest mulig data for å kunne analysere i ettertid.

Etter denne sekvensen skal jeg velge ut noen elever som skal løse oppgaver i programmet Scratch, og en matematikkoppgave, i ettertid av introduksjonen. Jeg vil i denne sekvensen også kort intervju elevene. Spørsmålene vil blant annet omhandle barnets interesser/holdninger for matematikk og programmering. Jeg vil også her benytte meg av observasjon, video- og lydopptak. Du kan velge hva ditt barn kan delta på nederst i skrevet, og du/dere som foresatte kan få se intervjuguiden hvis ønskelig.

Hva skjer med informasjonen om ditt barn?

Alle personopplysninger vil bli behandlet konfidensielt. Det er kun jeg som student som vil ha tilgang til datamaterialet jeg samler inn. Om det skal diskuteres med veiledere, vil navn anonymiseres først. Alt av datamaterialene vil oppbevares i innlåst skal på instituttet, og lyd-, videoopptak og bilder vil kun lagres på ekstern harddisk.

Deltakerne i prosjektet vil ikke kunne gjenkjennes i publikasjon, ettersom alle skal bli anonymisert.

Prosjektet skal etter planen avsluttes 1. juni 2015, og datamaterialet vil bli anonymisert etter det. Det anonymiserte datamaterialet kan bli oppbevart videre for videre bruk, eventuelt for forskningsartikkel. Etter 1. desember 2015 skal datamaterialet slettes.

Frivillig deltakelse

Det er frivillig å delta i studien, og du kan når som helst trekke ditt samtykke uten å oppgi noen grunn. Dersom du trekker deg, vil ingen opplysninger om ditt barn bli brukt i studiet.

Dersom du har spørsmål til studien, ta kontakt med student Susanne Iversen, 48223918. Du kan også ta kontakt med veileder Karin Rørnes, 776 60464, eller Geir Olaf Pettersen, 776 60359.

Studien er meldt til og godkjent av Personvernombudet for forskning, Norsk samfunnsvitenskapelig datatjeneste AS.

Samtykke til deltakelse i studien

Jeg har mottatt informasjon om studiet, og har informert mitt barn om det. Vi har bestemt i lag at han/hun er villig/ikke villig til å delta:

(Signert av foresatte, dato)

Navn på barn som deltar:

Kryss av for hva du/dere godkjenner og samtykker deltakelse på:

- Innlæring i programmering/Scratch, med videoopptak**
- Deltakelsen på oppgaver i ettertid med intervju og videoopptak**
- Jeg samtykker i at korte videosekvenser kan bli brukt i ettertid som en del av undervisningen på ILP eller i presentasjonen av masteroppgaven.**

