



Forsuaret

Bacheloroppgave

OPG3301

Predefinert informasjon

Startdato:	05-12-2022 09:00 CET	Termin:	2022 HØST
Sluttdato:	19-12-2022 08:00 CET	Vurderingsform:	Norsk 6-trinns skala (A-F)
Eksamensform:	Oppgave		
Flowkode:	1627 OPG3301 1 O 2022 HØST SKSK		
Intern sensor:	Alexander Sauter		
Intern sensor:	Terje Fykse		

Deltaker

Naun:	Brage Danneck Stugu
Kandidatnr.:	
FHS-id:	bstugu@mil.no

Gruppe

Gruppenaun:	One-person group
Gruppennummer:	11
Andre medlemmer i gruppen:	Deltakeren har innlevert i en enkeltmannsgruppe



Sjøkrigsskolen

Bacheloroppgave

Kan jeg programmere et verktøy som kan erstatte tradisjonell undervisning?

Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?

av

Brage Danneck Stugu

Levert som en del av kravet til graden:

BACHELOR I MILITÆRE STUDIER MED FORDYPNING I LEDELSE OG MARINE-
INGENIØR VÅPEN, ELEKTRONIKK OG DATA

Antall ord: ca. 10 000

Innlevert: Desember 2022

Godkjent for offentlig publisering

Publiseringsavtale

En avtale om elektronisk publisering av bachelor/prosjektoppgave

Kadetten har opphavsrett til oppgaven, inkludert rettighetene til å publisere den.

Alle oppgaver som oppfyller kravene til publisering vil bli registrert og publisert i Bibsys Brage når kadetten har godkjent publisering.

Opgaver som er graderte eller begrenset av en inngått avtale vil ikke bli publisert.

Jeg gir herved Sjøkrigsskolen rett til å gjøre denne oppgaven tilgjengelig elektronisk, gratis og uten kostnader	<input checked="" type="checkbox"/> Ja	<input type="checkbox"/> Nei
Finnes det en avtale om forsinket eller kun intern publisering? (Utfyllende opplysninger må fylles ut)	<input type="checkbox"/> Ja	<input checked="" type="checkbox"/> Nei
Hvis ja: kan oppgaven publiseres elektronisk når embargoperioden utløper?	<input type="checkbox"/> Ja	<input type="checkbox"/> Nei

Plagiaterklæring

Jeg erklærer herved at oppgaven er mitt eget arbeid og med bruk av riktig kildehenvisning. Jeg har ikke nyttet annen hjelp enn det som er beskrevet i oppgaven.

Jeg er klar over at brudd på dette vil føre til avvisning av oppgaven.

Dato: 18 – 12- 2022

Brage Danneck Stugu



Forord

Denne bacheloroppgaven er utført og skrevet av Brage Danneck Stugu siste halvdel av året 2022. Oppgaven har sin bakgrunn i en generell interesse for programmering og en vedvarende irritasjon over middelmådige forelesere.

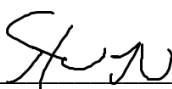
Gjennom et langvarig studieliv fra 2015 til 2022 har jeg opplevd utallige middelmådige forelesere, samt utallige middelmådige forelesninger av ellers gode forelesere. Etter å ha gjennomført et e-læringskurs i ING3391 ble jeg inspirert til å programmere en plattform undervisere kan benytte seg av til å gjennomføre sin undervisning på en bedre måte.

Takk til Terje Fykse, lektor på Sjøkrigsskolen for veiledning og deling av tidligere erfaringer.

Takk til klassen på Sjøkrigsskolen som sa seg villige til å teste ut verktøyet, samt gi viktige tilbakemeldinger.

Bergen, Sjøkrigsskolen, 18-12-2022

(Signatur)



Sammendrag

I en verden som har utallige elektroniske løsninger, hvorfor blir den samme forelesningen holdt flere ganger over flere år, når den bare kan spilles inn og vises på nytt? Eller bedre, legges inn som en del av et e-læringsverktøy som er spesialisert for at eleven skal tilegne seg kunnskap.

Bakgrunnen for oppgaven er påstanden:

Hvis en foreleser skreddersyr et elektronisk opplegg for å erstatte forelesningene sine, så vil dette opplegget være bedre enn forelesningene.

Videre er det utarbeidet en problemstilling som skal besvares.

Kan jeg programmere et verktøy undervisere kan benytte som erstatning for tradisjonell undervisning? Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?

Oppgaven omhandler utviklingen av et program som har til hensikt å kunne fungere som en erstatning for tradisjonell undervisning. Dette programmet blir benyttet av en klasse ved Sjøkrigsskolen. Klassen vil gjennom en spørreundersøkelse gi tilbakemeldinger på problemstillingen. Avslutningsvis sammenlignes resultater opp mot relevant litteratur.

Utviklingen av programmet gikk som planlagt, det ble utviklet et verktøy som fungerte, om det fungerte bra er derimot diskutabelt.

Det var høy korrelasjonen mellom tilbakemeldingene fra klassen ved Sjøkrigsskolen og relevant litteratur. Den tilsynelatende største fordelen med elektronisk undervisning er tilgjengeligheten og friheten til å lære hvor og når man vil. Den tilsynelatende største utfordringen med elektronisk med undervisning er å fremstille en generisk interaktivitet med underviseren.

Elektroniske løsninger kan enda ikke erstatte den tradisjonelle undervisningen, da det enda ikke er laget et elektronisk verktøy som godt nok etterligner fordelene med den tradisjonelle undervisningen.

Innholdsfortegnelse

Forord	ii
Sammendrag	iii
Innholdsfortegnelse	iv
Figurer	1
Forkortelser / Begrep	3
1 Introduksjon	4
1.1 Bakgrunn	4
1.2 Problemstilling	4
1.3 Mål	5
1.4 Avgrensninger	5
1.5 Valg av kilder	5
1.6 Oppgavens relevans for Sjøforsvaret	6
1.7 Struktur	6
2 Teori	7
2.1 En kort litteraturstudie om elektronisk undervisning	7
2.1.1 Elektronisk undervisning	7
2.1.2 Synkron og asynkron undervisning	8
2.1.3 Fordeler med elektronisk undervisning	8
2.1.4 Utfordringer med elektronisk undervisning	9
2.2 Objektorientert programmering	9
3 Fremgangsmetode	10
3.1 Programutvikling	10
3.1.1 Hvordan starte?	10
3.1.2 Programkrav	12
3.1.3 Arbeidsstrategi	14
3.2 Benytte programmet	14
3.3 Tilbakemeldinger	15
3.4 Tillatelse til tester og samtykkeerklæring	15
4 Programmet	16
4.1 Oppbygging	16
4.1.1 Lagring	16
4.1.2 Administratormodus og brukermodus	17
4.2 Klasser	17

4.2.1	Slides.....	18
4.2.2	Bilde.....	18
4.2.3	Video.....	19
4.2.4	Tekst.....	20
4.2.5	Quiz.....	21
4.2.6	Langsvar.....	23
4.3	Analyseverktøy	24
4.3.1	Video.....	24
4.3.2	Quiz.....	25
4.4	Utfordringer	25
4.4.1	Video.....	25
4.4.2	Kommunikasjon mellom klienter.....	26
5	Resultater og analyse	28
5.1	Gjennomføring.....	28
5.1.1	Videoanalyse.....	28
5.1.2	Quizresultater.....	30
5.1.3	Langsvarelementet	30
5.1.4	Oppfatninger og observasjoner	30
5.2	Spørreundersøkelse	30
6	Drøfting.....	34
6.1	Avgrensninger.....	34
6.2	Valg av kilder.....	34
6.3	Arbeidsstrategi	35
6.4	Programmet.....	35
6.5	Resultater	38
6.5.1	Videoanalyse.....	38
6.5.2	Quiz.....	38
6.5.3	Langsvarspørsmål	40
6.5.4	Spørreundersøkelse	40
6.6	Oppgaven som helhet.....	42
6.7	Viderebygging.....	43
6.8	Kognitiv bias	44
7	Avslutning.....	45
8	Referanser.....	46
	Vedlegg	47

Figurer

<i>Figur 1</i>	<i>Googlesøk som viser en liste over de «beste» Python kodekompilatorene.</i>	<i>10</i>
<i>Figur 2.</i>	<i>Googlesøk som viser en liste over de «beste» GUI-bibliotekene i Python.</i>	<i>11</i>
<i>Figur 3</i>	<i>Skisse av hvordan jeg så for meg at programmet kunne se ut. Øverste rad er hvordan programmet ville fremstå for en eleven, nederste er hvordan det kunne sett ut for en underviser.</i>	<i>13</i>
<i>Figur 4</i>	<i>Mappestrukturen i programmet, data og question er utelatt, de er like de andre undermappene. I dette programmet er det lagt inn to bilder, en tekst og en video.</i>	<i>17</i>
<i>Figur 5</i>	<i>Skjerm bilde av den første sliden, omtalt som hovedmeny, og slide nr. 1 begge slidene er tomme.</i>	<i>18</i>
<i>Figur 6</i>	<i>Skjerm bilde av hvordan legge til et bilde i programmet.</i>	<i>18</i>
<i>Figur 7</i>	<i>Til venstre: skjerm bilde av innholdet i bilde-mappen. Til høyre: innholdet i init-filen, og beskrivelse av innholdet.</i>	<i>19</i>
<i>Figur 8</i>	<i>Til venstre: en video slik det ser ut i brukermodus. Til høyre: En video som er lagt til i administratormodus, her er det flere funksjoner en kan benytte seg av.</i>	<i>19</i>
<i>Figur 9</i>	<i>Venstre: Oppsettet i videomappen, en video består av tre filer. Høyre: Denne videoen har ett spørsmål, det dukker opp på ti sekunder, dersom brukeren svarer feil sendes den til slide to, hvis den svarer riktig hopper den frem til det 45 sekundet i videoen.</i>	<i>20</i>
<i>Figur 10</i>	<i>Fra venstre til høyre. 1. Bilde av hvordan tekstelementet ser ut i programmet. 2. Tekstmappen. 3. Teksten lagret i en egen tekstfil.</i>	<i>21</i>
<i>Figur 11</i>	<i>Til venstre: Et spørsmål som vist i brukermodus. Høyre: Et spørsmål som nettopp er lagt til i administratormodus.</i>	<i>21</i>
<i>Figur 12</i>	<i>Til venstre: Et spørsmål lagt til i administratormodus, og riktig alternativ er markert som grønn. Til høyre: Det samme spørsmålet men i brukermodus, her har brukeren valgt riktig alternativ (grått), men ikke registrert svaret sitt enda.</i>	<i>22</i>
<i>Figur 13</i>	<i>Måten et spørsmål er lagret på i programmet. Det er mulig med opptil 8 alternativer.</i>	<i>22</i>
<i>Figur 14</i>	<i>Spørsmål som er lagt til, åpnet i administratormodus.</i>	<i>23</i>
<i>Figur 15</i>	<i>Til høyre: Langsvarelementet som sett i brukermodus. Til venstre: Langsvarelementet i administratormodus, her har det kommet inn et svar fra «navn».</i>	<i>23</i>
<i>Figur 16</i>	<i>Videoanalyse i programmet, her kan man se at Bruker1 har sett starten av videoen to ganger, og slutten kun en gang, Bruker2 har sett starten av videoen en gang, men slutten tre ganger, Bruker3 så videoen frem til syv sekunder før han skrudde av. Totalen er hva brukerne har sett til sammen.</i>	<i>24</i>
<i>Figur 17</i>	<i>Hva som kommer opp dersom en trykker på DATAknappen, her ser vi at brukerne burde forbedre matteegenskapene sine.</i>	<i>25</i>
<i>Figur 18</i>	<i>Utdrag av innboksen, dette er skjult for alle brukerne av programmet.</i>	<i>27</i>
<i>Figur 19</i>	<i>Viser seermønsteret til de åtte brukerne av programmet. Totalen er den øverste linjen. De fleste så hele videoen ca. 1 gang, det er da mange linjer oppå hverandre, det noteres at ingen hoppet over videoen, da ville det vært en strek på 0. Alle har sett hele videoen.</i>	<i>29</i>

<i>Figur 20 Seermønsteret til brukeren kanulebolle.</i>	<i>29</i>
<i>Figur 21.[1,2,3,4,5,6,7] Spørsmålene som ikke var fritekst som ble stilt i spørreundersøkelsen, 1-6 er spørsmål der det skal besvares på en skala, 7 er et åpent spørsmål om hvordan hver elev lærer mest, med mulighet for å legge til alternativ selv.</i>	<i>33</i>
<i>Figur 22 (Fykse, 2003). Ideskisse for forklaringsmodul. Finnes i vedlegg G</i>	<i>39</i>

Forkortelser / Begrep

Bibliotek	Ferdigskrevet kode som utfører en gitt hensikt
ED	United States Department of Education
FAQ	Frequently Asked Questions (ofte stilte spørsmål)
FPS	Bilder per sekund (Frames Per Second)
GUI	Graphical User Interface (Grafisk brukergrensesnitt)
OOP	Objektorientert programmering
Spagettikode	Ustrukturert, uoversiktlig kildekode
Tradisjonell undervisning	Klasseromsundervisning eller fysisk forelesning.

1 Introduksjon

Denne oppgaven vil undersøke om det er mulig for meg å programmere et læringsverktøy som kan benyttes fremfor tradisjonell undervisning. Oppgaven vil diskutere i hvilken grad det programmerte verktøyet fungerer. Deretter diskuteres det rundt temaet elektronisk undervisning. Diskusjonene skjer på grunnlag av en spørreundersøkelse utført på Sjøkrigsskolen, samt relevant litteratur. Gjennom oppgaven avdekkes hvilke egenskaper som er sentrale dersom elektronisk undervisning skal erstatte den tradisjonelle undervisningen.

1.1 Bakgrunn

Etter en vedvarende irritasjon over middelmådige forelesninger, ønsket jeg å prøve å utvikle et bedre alternativ til den ofte kjedelige tradisjonelle undervisningen. Det er klart at en student på bachelornivå trolig ikke vil lage et produkt som revolusjonerer undervisningen, men det skal være mulig å lage et program som kan erstatte den tradisjonelle undervisning til en viss grad.

Jeg er ikke den eneste som synes at elektronisk undervisning har et ubrukt potensiale, ifølge (Tušelj, Majcen, Mervar, Stepankina, & Čater, 2016) blir hele verden kontinuerlig digitalisert, men undervisningen henger etter. U.S Department of Education (videre omtalt som ED) nevner tre grunner til at elektronisk undervisning har blitt populært. Elektronisk undervisning øker tilgjengeligheten av læring, det er økonomisk gunstig og er mulig for en lærer å ha flere elever enn ved tradisjonell undervisning.

Hvorfor undervises den samme forelesningen år etter år, når den kunne blitt spilt inn og vist på ny.

Følgende påstand er et sentralt irritasjonsmoment som til dels er bakgrunnen for oppgaven:

Hvis en foreleser arbeider med å skreddersy et elektronisk opplegg som skal erstatte forelesningene sine, vil dette produktet være bedre enn forelesningene.

1.2 Problemstilling

Ut fra bakgrunnen og en ingeniørfaglig nødvendighet er det utarbeidet en problemstilling for oppgaven:

Kan jeg programmere et verktøy undervisere kan benytte som erstatning for tradisjonell undervisning? Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?

Valget av problemstillingen kommer av en interesse for programmering, samt en vedvarende irritasjon over middelmådige forelesninger.

For å løse problemstillingen vil det først utvikles et program, en målgruppe vil så bruke programmet, målgruppen vil til slutt besvare en spørreundersøkelse. Det er under arbeidet lagt mest arbeid ned i første del av problemstillingen, altså programutviklingen.

1.3 Mål

I denne oppgaven er det flere mål, de kan kortes ned til følgende liste: Mål nr. 1 er førsteprioritet, resten er ikke i prioritert rekkefølge.

1. Ha det kjekt mens det arbeides med oppgaven
2. Lage et produkt som kan benyttes iht. problemstillingen
3. Benytte produktet
4. Få tilbakemeldinger på produktet
5. Få tilbakemeldinger rundt problemstillingen
6. Tilegne seg kunnskap om temaet elektronisk undervisning
7. Bli bedre på å programmere

1.4 Avgrensninger

Første del av problemstillingen er *kan jeg programmere et verktøy?* Dette innebærer at jeg faktisk programmerer et verktøy selv, altså vil oppgaven avgrenses av hva jeg evner å programmere. Oppgaven vil samtidig tilstrebe å ikke benytte produkter som allerede eksisterer, oppgaven vil heller ikke sammenlignes med allerede eksisterende produkter.

Arbeidsfordelingen i oppgaven er i hovedsak programmering da den skal være relevant opp mot en ingeniørfaglig studieretning. Dette medfører at teori rundt informasjonsinnhenting, samt teori vedrørende utførelsens av spørreundersøkelser i større grad ikke er undersøkt.

1.5 Valg av kilder

Problemstillingen tar for seg elektronisk undervisning, valget av kilder vil reflektere problemstillingen da mesteparten av kunnskapen jeg tilegner meg er gjennom internett. Samtlige av kildene og referansene funnet i oppgaven er gjennom sporadiske søk i Google, det vil i oppgaven bli gitt tillitt til de svarene man får gjennom Google. I 6.2 diskuteres dette nærmere. Det

siteres ikke til kilder brukt under utviklingen av programmet, dette fordi det ville vært nødvendig å sitere til flere hundre nettsteder som har lært meg hvordan jeg skal løse diverse utfordringer i programmering.

1.6 Oppgavens relevans for Sjøforsvaret

Oppgaven undersøker et sentralt tema ved alle institusjoner som ønsker å utvikle seg, nemlig undervisning. Oppgaven er derfor å regne som høyst relevant for Sjøforsvaret. Samtidig er oppgaven på ingen måter mer relevant for Sjøforsvaret enn andre institusjoner.

1.7 Struktur

Oppgaven struktureres etter rekkefølgen arbeidet ble gjort. Introduksjonen har tatt for seg bakgrunn, mål, problem og avgrensninger for oppgaven. Videre tar oppgaven for seg et teoretisk aspektet tilknyttet oppgaven, etterfulgt av fremgangsmetoden i oppgaven. Fremgangsmetoden kan deles i tre deler. Første del går ut på selve programmeringen av verktøyet. Andre del tar for seg selve testingen av programmet. I tredje del samles det inn tilbakemeldinger på selve programmet, samt tilbakemeldinger på konseptet *elektronisk vs. tradisjonell undervisning*.

Avslutningsvis vil oppgaven drøfte de tre delene opp mot hverandre, og innhentet litteratur.

2 Teori

Oppgaven omhandler konseptet *elektronisk undervisning*, det er derfor ikke noe spesifikk litteratur som må redegjøres for. Det er istedenfor gjort en kort litteraturstudie om elektronisk undervisning. Det er vektlagt å finne fordeler og utfordringer med elektronisk undervisning.

Det er redegjort for et sentralt begrepet innenfor objektorientert programmering (OOP).

2.1 En kort litteraturstudie om elektronisk undervisning

Det er i Google søkt etter «paper on e-courses», der det er valgt ut noen kilder som besvarer deler av de samme spørsmålene som denne oppgaven og skal besvare. Litteraturstudiets oppbygging redegjør først for hva elektronisk undervisning er, det tar så for seg forskjellen mellom synkron og asynkron undervisning, for så å beskrive noen fordeler og utfordringer med elektronisk undervisning.

2.1.1 Elektronisk undervisning

Tre definisjoner av e-læring, oversatt fra engelsk.

«

1. Enhver læring som involverer internett eller intranett.
2. Alt som er levert, muliggjort eller formidlet gjennom elektronisk teknologi for å fremme læring.
3. E-en i e-læring bør ikke står for elektronisk, men en forkortelse for utvikling (evolving), forbedret (enhanced), overalt (everywhere), når som helst (every time) og alle(everybody).

» (Zheng, Bender, & Lyon, 2021)

Selv om de fleste har en egen oppfatning av hva e-læring er, og fort trekker konklusjonen om at det er all elektronisk undervisning, ser man at ut i fra hvordan man definerer e-læring kan man ha forskjellige oppfatninger av hva det faktisk er. I denne oppgaven brukes begrepet elektronisk undervisning, denne definisjonen vil i denne oppgaven innebære alle de tre definisjonene sitert over.

2.1.2 Synkron og asynkron undervisning

Synkron og asynkron undervisning er et sentralt begrep innenfor undervisning. Kort fortalt så er synkron undervisning når en har interaksjon med en lærer samtidig som undervisningen foregår, synkron læring kan man anse som å se noe *live*, altså å f.eks. være fysisk tilstede i en forelesning. Asynkron vil på det andre siden være å se opptak av den samme forelesningen, uten mulighet for interaksjon med underviseren.

2.1.3 Fordeler med elektronisk undervisning

En metaanalyse utført av ED på data i tidsrommet 1996-2008 viser til at fag med online læring, i snitt førte til bedre læring (U.S Department of Education, 2009)

ED nevner tre grunner til at elektronisk undervisning har blitt populært. Elektronisk undervisning er av ED omtalt som «Online learning», det anses ikke nødvendigvis å være det samme som elektronisk undervisning, men fordelene gjelder begge.

«Online learning has become popular because of its potential for providing more flexible access to content and instruction at any time, from any place. Frequently, the focus entails (a) increasing the availability of learning experiences for learners who cannot or choose not to attend traditional face-to-face offerings, (b) assembling and disseminating instructional content more cost-efficiently, or (c) enabling instructors to handle more students while maintaining learning outcome quality that is equivalent to that of comparable face-to-face instruction.» (U.S Department of Education, 2009)

Elektronisk undervisning er mulig å distribuere til flere, det kan også brukes flere ganger, noe som gjør det mer økonomisk gunstig enn tradisjonell undervisning.

Wlodkowski i (El- Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014) hevder at lærende lærer mer gjennom databaserte instruksjoner, fremfor de tradisjonelle klasseromsmodellene. En av grunnene til dette kan være en høyere grad av interaktivitet.

(Zheng, Bender, & Lyon, 2021) Videreformidler at forskning tyder på at vellagd online læring kan lede til forbedret motivasjon, tilfredshet og læring.

2.1.4 utfordringer med elektronisk undervisning

En av utfordringene med e-læring er at suksessen av e-læringsmetoder kun kan måles iht. lærerens evne til å formidle. E-læringsmetoder skades ofte av underviserens opplæring i metoden. (El- Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014)

Tamm skrev en artikkel som inneholdt en liste over ti ulemper med elektronisk undervisning, punktene mest relevant for oppgaven er listet under, punktene er oversatt fra engelsk.

«

- Tilbakemeldinger til studenter er begrenset
- Selvmotivasjon og kontroll over eget tidsfor driv.

» (Tamm, 2022)

2.2 Objektorientert programmering

Objektorientert programmering er basert på opprettelsen av objekter, et objekt er en instans av en klasse. (Wikipedia, 2022)

Dette gir ikke akkurat en god beskrivelse av hva OOP er, gjennom en rekke eksempler under forklares det forhåpentligvis bedre.

I et programmeringsspråk skiller man på forskjellige typer data, de mest grunnleggende typene data er ofte omtalt som *string(str)* og *integer(int)*, altså en tekststreng og et heltall. *Str* og *int* er regnet som klasser. Klassen *str* inneholder tekst, *int* inneholder et heltall.

Dersom man ønsker å opprette en egen klasse f.eks. en klasse kalt menneske. Hva ville vært naturlig for denne klassen å inneholde? En menneskeklasse ville kanskje inneholdt flere klasser, f.eks. en *string* som brukes til å lagre navnet på mennesket og kanskje en *integer* som brukes til å lagre alderen på mennesket.

En klasse kan ha tilhørende metoder, disse metodene kan f.eks. være addering, for en *integer* vil det selvsagt være lett å addere to tall og gi svaret. Men å addere to mennesker ville jo ikke vært mulig. En metode for en menneskeklasse ville kanskje vært bursdag, dersom bursdag inntrer så vil alderen som er lagret i en *int* for mennesket økes med en.

I oppgaven er det lagret bilder, det er derfor opprettet en klasse som heter bilde, bildeklassen inneholder da selve bildefilen, men også informasjon om plassering og oppløsning, da er plasseringen og oppløsningen lagret som integeres i bildeklassen.

3 Fremgangsmetode

Fremgangsmetoden er tredelt, den første og mest arbeidsomme delen består av programutvikling. Andre del består av bruken av læringsverktøyet, den siste delen går ut på tilbakemeldinger på programmet, samt problemstillingen.

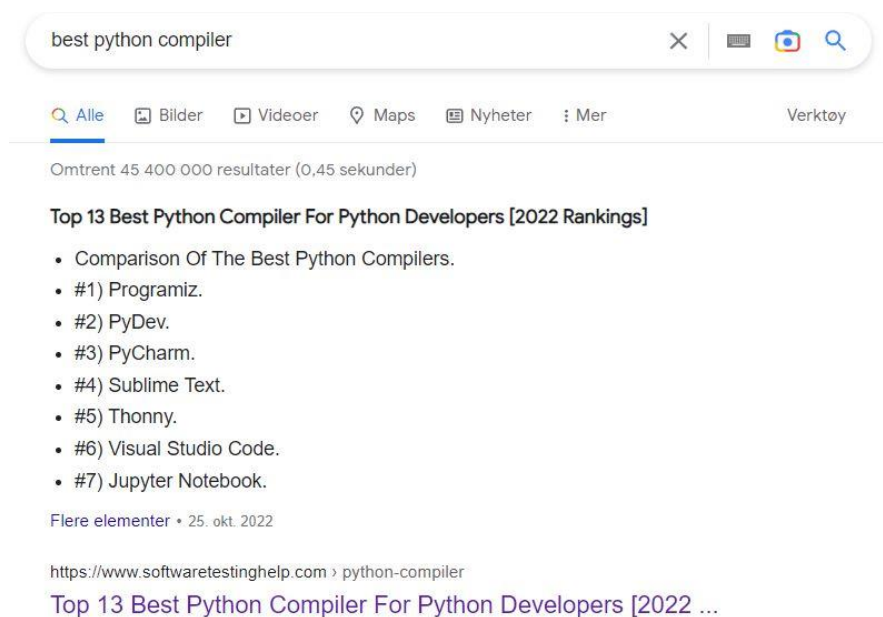
3.1 Programutvikling

Programutviklingen foregikk i all hovedsak i eget hode, for å oppnå mål nr.1 som går ut på å ha det kjekt mens det arbeides med oppgaven er programutviklingen gjennomført på en måte som lar meg arbeide slik jeg selv ønsker. Dette kan ha gått både positivt og negativt ut over måloppnåelsen av resterende mål.

3.1.1 Hvordan starte?

Det første valget som må gjøres er hvilket språk man skal programmere i. De eneste programmene som ble vurdert var Python og C++, da dette er de eneste jeg har erfaring med. Valget falt fort på Python da jeg har programmert visuelle program i dette før, mens jeg ikke har gjort noe lignende i C++.

Min tidligere erfaring i Python var kun på mindre prosjekt, og det har ikke blitt brukt en egen kodekompiletor. Jeg har tidligere kun brukt den medfølgende kompiletoren. Et enkelt google-søk og nedlastning løser dette.

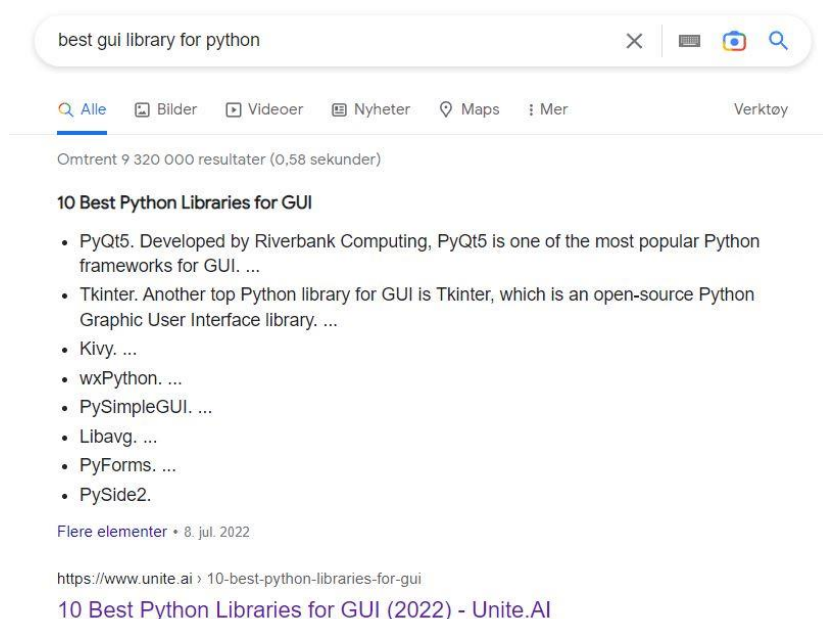


Figur 1 Googlesøk som viser en liste over de «beste» Python kodekompiletorene.

Jeg har sett flere YouTube-videoer om programmering, der har mange brukt PyCharm, valget falt derfor på PyCharm.

Et av målene med oppgaven var å utvikle seg selv som programmerer, for å i høyest mulig grad oppnå dette målet var det viktig å ikke kun sette sammen forskjellige bibliotek, og da i praksis kun benytte seg av ferdigskrevet kode. For å få best måloppnåelse må det brukes en god blanding av bibliotek og egenskrevne kode.

Den neste valget som måtte tas var hvilket bibliotek som skulle brukes som utgangspunkt i oppgaven, her var det viktig at det var lett å lære. Et enkelt googlesøk «best gui library for python», ga svaret.



Figur 2. Googlesøk som viser en liste over de «beste» GUI-bibliotekene i Python.

Etter en vurdering av hvilket av bibliotekene på listen som var best, etterfulgt av flere google-søk falt valget på Tkinter. Tkinter ble valgt fordi det omtales som lett å bruke, det er Python's *de facto standard* og er inkludert i Python. (McFarland, 2022) (Wikipedia, 2022)

I tillegg er Tkinter et biblioteket jeg har hørt om før, noe som ga det et fortrinn over dem andre. Tkinter er biblioteket lagt til grunn for programutviklingen, men det er også flere andre biblioteker brukt i oppgaven. Python, Tkinter og samtlige av bibliotekene brukt kan man ikke uten videre *stjele*, en må først vite at forfatteren har gitt tillatelse til å benytte dem. Python, Tkinter og samtlige av bibliotekene brukt er lisensiert under MIT License og/eller ZERO-CLAUSE

BSD LICENSE, eventuelt lisenser som formidler det samme. Disse lisensene er gitt i vedlegg C.

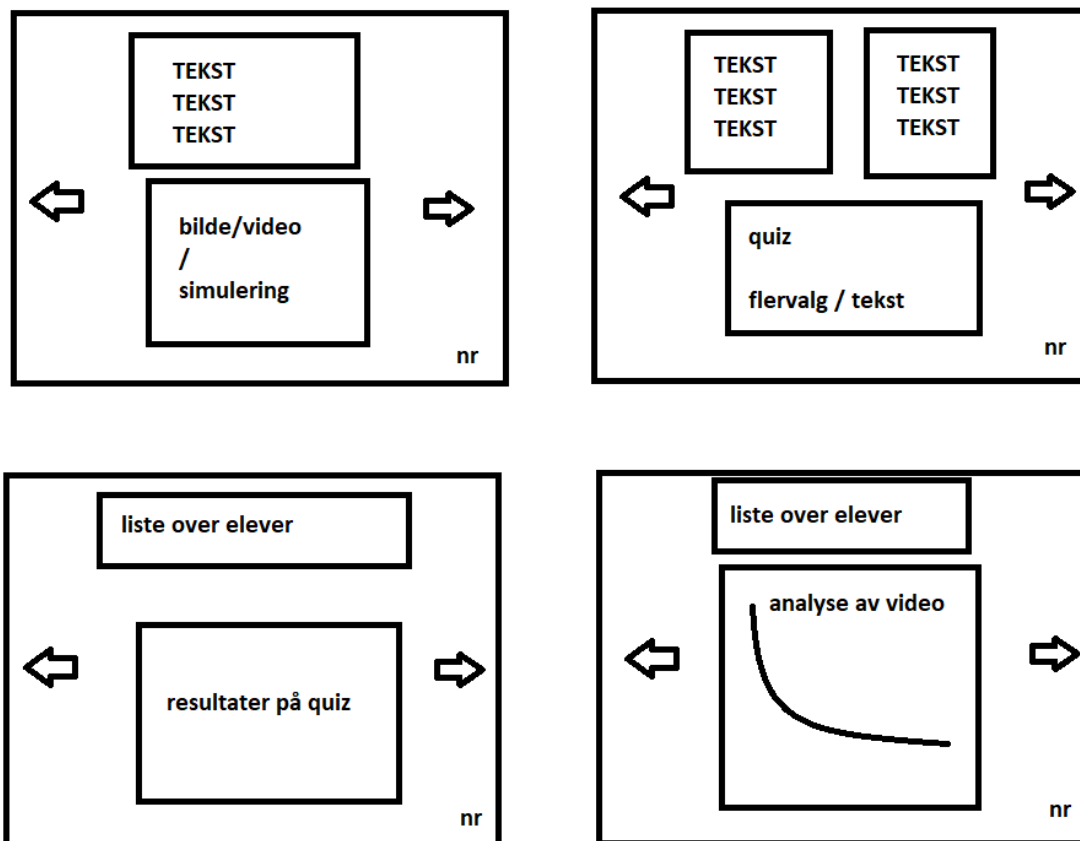
3.1.2 Programkrav

Før det settes i gang med å programmere, må det kartlegges en tydelig baktanke om programmet, Det hele startet med en visualisering av hvordan programmet skal se ut og fungere, det ble videre satt flere må-krav, ønskelig-funksjoner og *far-fetched* funksjoner.

3.1.2.1 Idéen

En enkel måte å beskrive ideen bak programmet vil være å sammenligne det med Microsoft PowerPoint. Å programmere et program som fungerer slik som PowerPoint, men med større vektning rundt videoer, quizer og analyse av disse er en god start.

Det var viktig at programmet ikke kun utførte de samme oppgavene til et allerede eksisterende program. At programmet kunne tilføre noe annerledes ble det lagt stor vekt på, i PowerPoint er det mulig å legge inn makroer, men dette er ofte utenfor kunnskapen til de fleste brukere av PowerPoint. I YouTube finnes det statistikk som følger med på hvordan videoen blir sett på. Om seerne hopper over eller ser bestemte deler på nytt er det mulig som opplaster å følge med på. Det å kunne samle slike funksjoner under ett brukervennlig program er ideen bak oppgaven. Et slikt program vil fungere som asynkron undervisning. En veldig god versjon av en slikt program vil tilstrebe å etterligne synkron undervisning i så stor grad som mulig.



Figur 3 Skisse av hvordan jeg så for meg at programmet kunne se ut. Øverste rad er hvordan programmet ville fremstå for en elev, nederste er hvordan det kunne sett ut for en underviser.

3.1.2.2 Minstekrav

Minstekravene til programmet er det som minimum må være på plass for at oppgaven kan gjennomføres på en tilstrekkelig måte, disse kravene listes under.

- Skifte mellom scener (slides)
- Legge inn og lagre tekst til hver slide
- Legge inn og lagre bilder til hver slide
- Legge inn og lagre videoer til hver slide
- Legge inn og lagre quiz til hver slide

3.1.2.3 Ønskelig-funksjoner

- Kommunikasjon mellom underviser og elev
- Videoanalyse
- Quizresultater

- Kjøres gjennom én fil

3.1.2.4 *Far-fetched* funksjoner

- Nettleserkompatibilitet
- Mobilkompatibilitet
- Interaktive simuleringer

3.1.3 Arbeidsstrategi

3.1.3.1 Tidsfrister eller konkrete mål?

Programutvikling kan anses å være et evigvarende prosjekt, det er på mange måter umulig å bli ferdig, man er kun ferdig når man sier man er ferdig, eller tidsfristen har gått ut. Arbeidsstrategien brukt er at det er satt en tidsfrist, så jobbes det frem mot fristen. Det var aldri et endelig mål som måtte nås, det var kun en endelig tidsfrist, tidsfristen var satt opp mot når programmet skulle benyttes, omtalt i 3.2.

3.1.3.2 Nytt? eller opprydning av gammelt?

Gjennom hele oppgaven må det tas vurderingen om det er best å fortsette utvikling av programmet med nye funksjoner, eller om det er bedre å vende blikket tilbake for å rydde opp i eldre kode. Valget falt i hovedsak på utvikling av nye funksjoner. Det er fordeler og ulemper uansett hvilket valg man tar, dette drøftes senere i oppgaven.

3.2 Benytte programmet

Programmet skal benyttes på en klasse ved Sjøkrigsskolen. Første steg vil være å orientere underviseren i klassen om problemstillingen, samt redegjøre for programmets kapabiliteter, og begrensninger. Underviseren i klassen gjør klart et opplegg, med diverse bilder, videoer, tekster og quizer. Disse elementene legges så i samråd med underviseren inn i programmet, programmet er så klart til å distribueres til klassen.

Klassen bes om å laste ned programmet, for så å gå gjennom programmet. Programmet skal fungere som en erstatning for den vanlige undervisningen.

3.3 Tilbakemeldinger

Den samme skoleklassen som benyttet programmet vil bli bedt om å fylle ut en spørreundersøkelse. Spørreundersøkelsen er todelt, første del vil omhandle det utviklede programmet, mens den andre delen vil stille spørsmål rundt problemstillingen *kan tradisjonell undervisning erstattes av elektroniske løsninger?*

3.4 Tillatelse til tester og samtykkeerklæring

For å kunne forske på personell i Forsvaret er det nødvendig å søke om tillatelse til å gjennomføre spørreundersøkelse/intervju/tester i Forsvaret. Det må da sendes inn en utfyllende prosjektbeskrivelse, et utkast til hvilke spørsmål som skal stilles, og det må utarbeides en samtykkeerklæring. En tillatelse må utstedes av Forsvarets forskningsnemd. Tillatelsen fra forskningsnemden er gitt i vedlegg E, samtykkeerklæringen er gitt i vedlegg F.

4 Programmet

I dette kapitlet forklares det hvordan programmet er bygget opp, hvilke sentrale metoder som er brukt, samt noen utfordringer og løsninger. Dette kapitlet regnes i seg selv som et resultat av oppgaven, resultatene fra bruken av programmet omtales i kapittel 5.

Programmet er kodet i Python og består av seks .py-filer, samt diverse biblioteker. For å kjøre Pythonfiler må man normalt ha Python og de tilhørende bibliotekene lastet ned på egen data-maskin. Dette er selvsagt ikke ønskelig, for å løse dette er *Auto PY to EXE* brukt. I *Auto PY to EXE* laster man opp den .py-filen som kjører programmet, så får man tilbake en .exe fil. Denne .exe-filen inneholder alt det nødvendige for å kjøre programmet. .exe-filen distribueres lett gjennom en nedlastningslenke. Denne lenken er tilgjengelig i vedlegg I. Når .exe-filen er lastet ned kan man bruke programmet.

Før programmet lastes ned er det anbefalt å lese brukermanualen, vedlegg D. Denne gir en fullstendig instruksjon av alle elementene i programmet. Det er også utarbeidet en brukermanual i selve programmet, dette er da en mappestruktur som kan åpnes i programmet, denne finnes i vedlegg H, denne fungerer som en brukermanual for programmet, men er ikke like utfyllende som brukermanualen i vedlegg D.

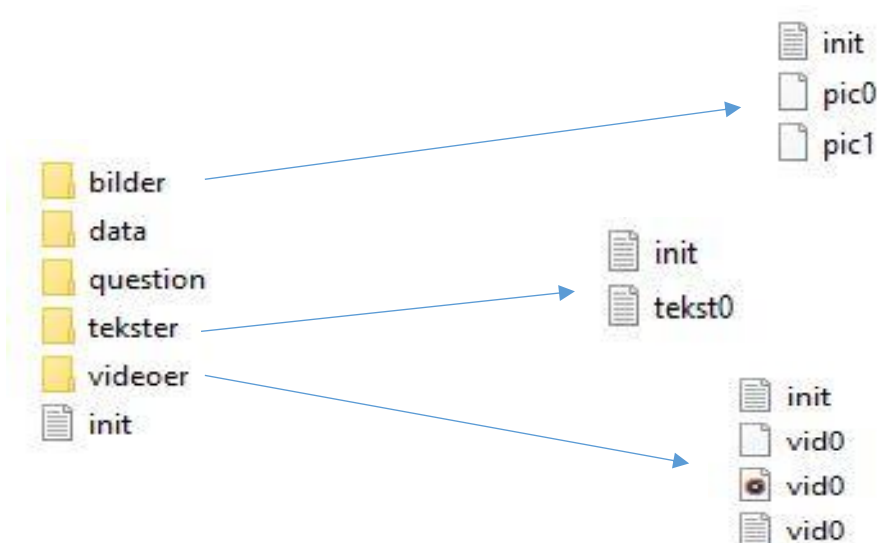
4.1 Oppbygging

Programmet er programmert i Python, og det er tatt utgangspunkt i objektorientert programmering. Tkinter er utgangspunktet for *GUI'en*, i Tkinter er det mange klasser å velge mellom, men ingen som er spesialisert til bruken i denne oppgaven. Programmet inneholder flere egne klasser som er bygget opp av grunnleggende funksjoner i Python, samt opp til flere av Tkinter sine klasser.

4.1.1 Lagring

Informasjon lagret i programmet lagres i en mappestruktur, mappestrukturen inneholder flere mapper med flere .txt-filer som leses av Python. Python gjør handlinger ut i fra innholdet i

mappestrukturen.



Figur 4 Mappestrukturen i programmet, data og question er utelatt, de er like de andre undermappene. I dette programmet er det lagt inn to bilder, en tekst og en video.

Det er en tekstfil kalt init i mappestrukturen, i hver undermappe er det og en tilhørende tekstfil som også kalt init. Init er kort for *initialization*, filene inneholder instruksjoner som skal gjennomføres hver gang programmet blir kjørt, det er disse instruksene som kontinuerlig endres når programmet benyttes.

4.1.2 Administratormodus og brukermodus

Programmet skiller mellom administratormodus og brukermodus. I brukermodus er det ikke mulig å legge til, endre og/eller flytte på elementer, altså er det i brukermodus ikke mulig å endre på mesteparten av mappestrukturen. Det er selvsagt mulig å besvare spørsmål, spole, og sette på pause i videoer i brukermodus. For å få tilgang til administratormodus må man kun trykke på en knapp, dette for enkelthetsens skyld, dette er noe man enkelt kunne lagt inn et passord for å gjøre, eventuelt kunne man distribuert forskjellige versjoner til lærer og elev.

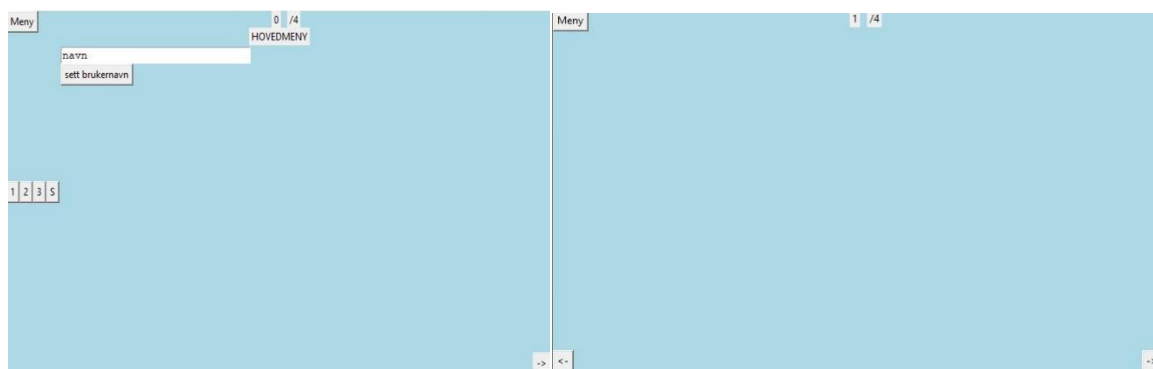
4.2 Klasser

Hovedklassene som er laget i programmet er slides, bilde, video, tekst og quiz. Slides er den overordnede scenen hvor de andre klassene blir plassert. Slides kan regnes som et nivå *over* de

andre klassene. Utenom slides, så har alle klassene til felles at de har en posisjon, og størrelse i seg. Hvis man f.eks. legger inn et bilde på en slide, så er det hensiktsmessig at man kan flytte og endre størrelse på det. Gjennom mappestrukturen husker programmet hvor det er plassert til neste gang programmet åpnes.

4.2.1 Slides

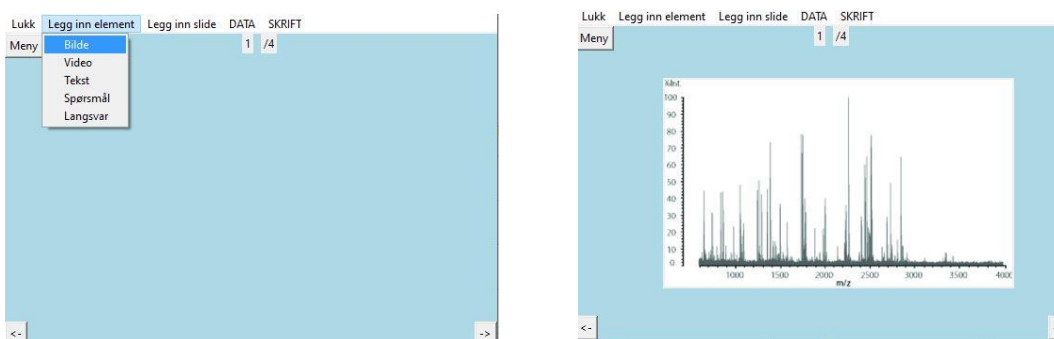
Slides kan omtales som en scene, eller et lysbilde. Det er mulig å legge til ubegrenset antall slides, hovedoppgaven til slides er å organisere programmet på en god måte, og være en indikasjon på fremgang.



Figur 5 Skjerm bilde av den første sliden, omtalt som hovedmeny, og slide nr. 1 begge slidene er tomme.

4.2.2 Bilde

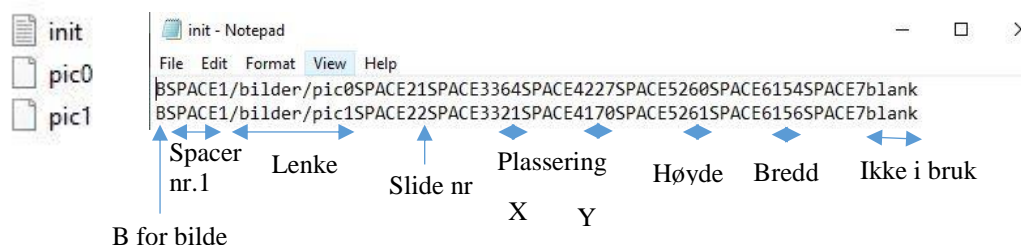
For å legge til et bilde må administratormodus aktiveres, det dukker opp en meny, trykker man så på bilde vil man bli bedt om å velge fil, etter man har valgt fil vil bilde bli lagt til i programmet.



Figur 6 Skjerm bilde av hvordan legge til et bilde i programmet.

Det som skjer i bakgrunnen er at det bilde som ble valgt kopieres og legges inn i bildemappen fra *Figur 4*. Samtidig fylles det inn i tekstfilen *init*, i bildemappen som vist i *Figur 7*.

Mappen bilder består av alle bildene som er lagret, samt en tekstfil som inneholder informasjonen om plasseringen og størrelsen til bilde.

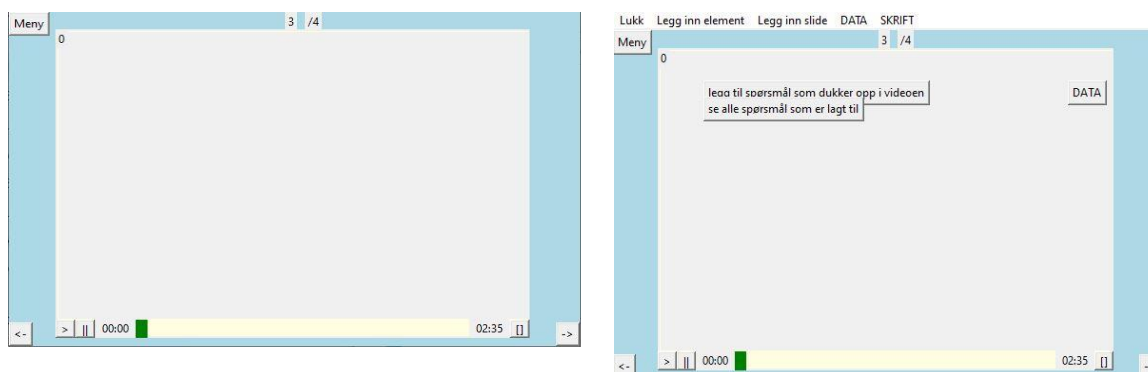


Figur 7 Til venstre: skjermbilde av innholdet i bilde-mappen. Til høyre: innholdet i *init*-filen, og beskrivelse av innholdet, det er hittil lagt to bilde til i denne mappestrukturen.

Når et bilde er lagt til i programmet kan man i administratormodus endre plassering og størrelse på bildet, dette lagres slik vist i *Figur 7*. I brukermodus kan man dobbeltklikke på bildet for å zoome inn.

4.2.3 Video

Video legges til på samme måte som bilder, slik vist i *Figur 6*. Det lagres også på samme måte som vist til høyre i *Figur 7*.

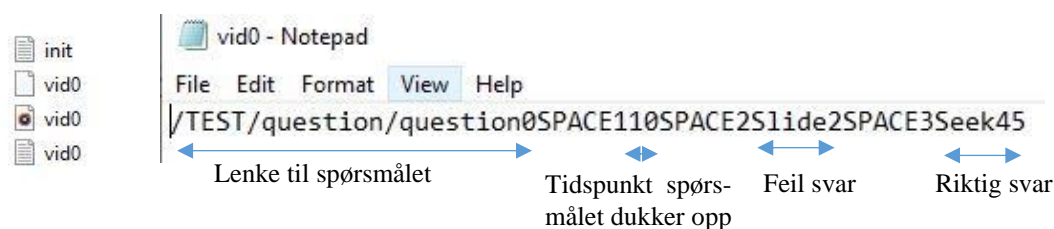


Figur 8 Til venstre: en video slik det ser ut i brukermodus. Til høyre: En video som er lagt til i administratormodus, her er det flere funksjoner en kan benytte seg av.

Videoelementet er ikke kun en video, det består også av diverse knapper og funksjoner, i brukermodus er det de grunnleggende spill av (>), pause (||) og fullskjerm ([]). I administratormodus er det flere knapper, knappene er brukt til å legge til spørsmål i videoen, og DATA knappen

er brukt for å få en analyse av hvordan videoen har blitt sett på. Spørsmålene som legges til i videoen er basert på den klassen som er beskrevet i quiz 4.2.5. Men det er noen tilleggsfunksjoner spesifikt for spørsmål i en video. Spørsmålene som er lagt inn i en video vil ha muligheten til å dirigere brukeren til en gitt tid i videoen, eller til en gitt slide i programmet. Hvis brukeren svarer feil på spørsmålet, så kan den automatisk bli flyttet til det tidspunktet i videoen som omhandler spørsmålet, eventuelt tilbake til den sliden som omhandlet det aktuelle temaet. Det samme gjelder selvsagt hvis brukeren svarer riktig, men da er det trolig aktuelt å hoppe fremover, ikke bakover.

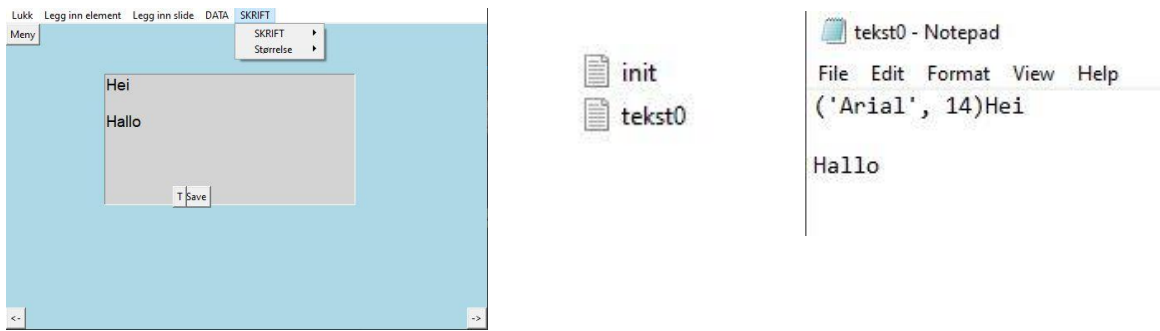
For å kunne lagre informasjonen i videoen blir det opprettet en egen tekstfil for hver video. Tkinter hadde ikke en egen klasse for å spille av videoer med lyd, derfor måtte videoer splittes i selve videoen uten lyd, og en egen fil som kun er lyd. For å spille av selve videoen er det brukt et eget bibliotek kalt *TkinterVideo*. For å spille av lyden er biblioteket *Pygame* brukt, disse er satt sammen i videoklassen, og spiller av samtidig. Brukeren vil aldri vite at det egentlig er to filer som spilles av samtidig.



Figur 9 Venstre: Oppsettet i videomappen, en video består av tre filer. Høyre: Denne videoen har ett spørsmål, det dukker opp på ti sekunder, dersom brukeren svarer feil sendes den til slide to, hvis den svarer riktig hopper den frem til det 45 sekundet i videoen.

4.2.4 Tekst

Tekst legges til på samme måte som bilder og videoer, som vist i *Figur 6* og *Figur 7*. Det er mulig å endre skrifttype og skriftstørrelse som vist i figuren under.

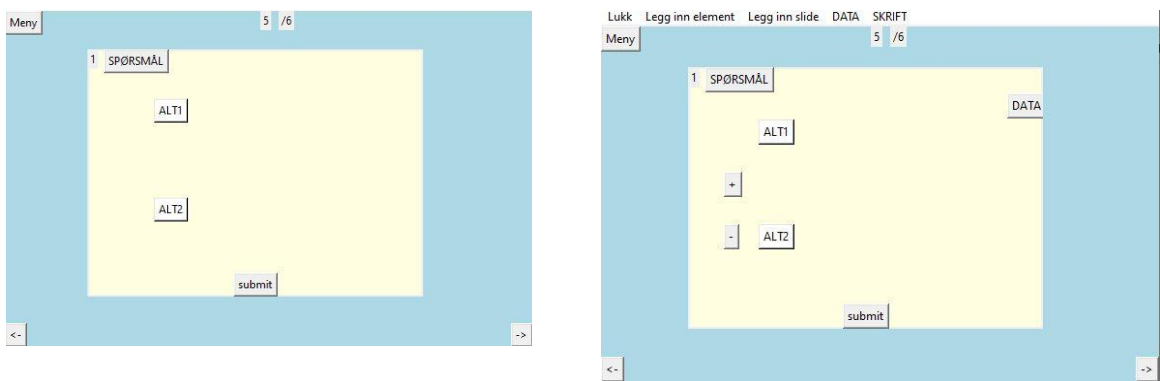


Figur 10 Fra venstre til høyre. 1. Bilde av hvordan tekstelementet ser ut i programmet. 2. Tekstmappen. 3. Teksten lagret i en egen tekstfil.

Når teksten er lagt inn i programmet er det mulig å endre skrifttype ved å bruke SKRIFTmenyen i administratormodus, for å bekrefte valget av skrift må man trykke på T, for å lagre selve teksten trykker man på *Save*.

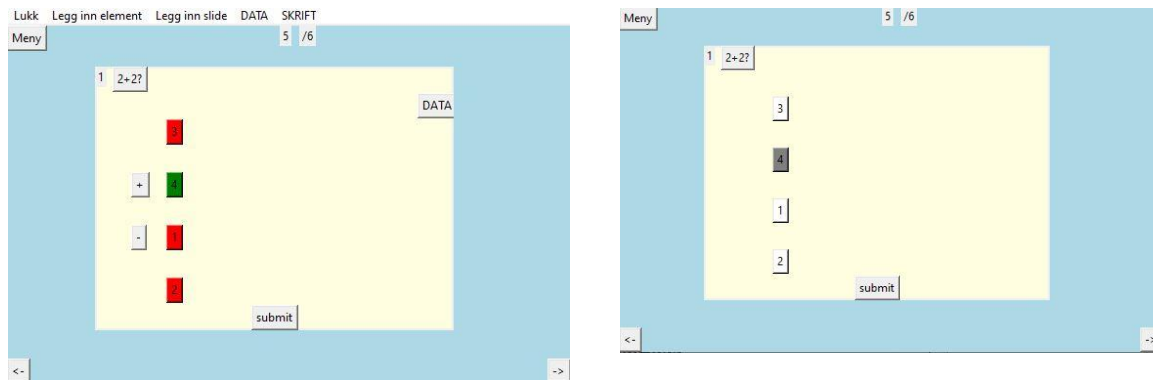
4.2.5 Quiz

Quiz legges til på samme måte som ovennevnte elementer.



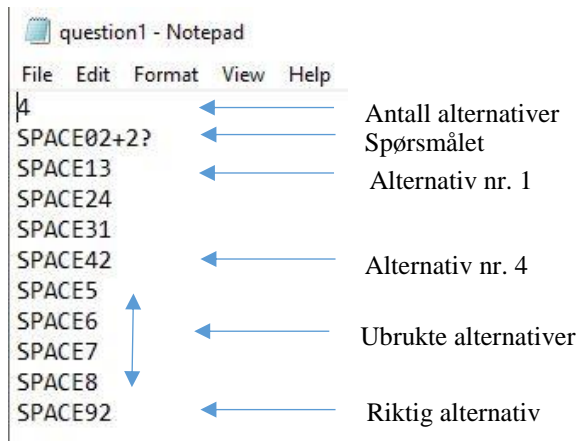
Figur 11 Til venstre: Et spørsmål som vist i brukermodus. Høyre: Et spørsmål som nettopp er lagt til i administratormodus.

Når programmet er i administratormodus er det flere knapper en kan trykke på, det er (+) og (-) for å legge til eller fjerne alternativer, også er det (DATA) for å se resultatene på spørsmålet.



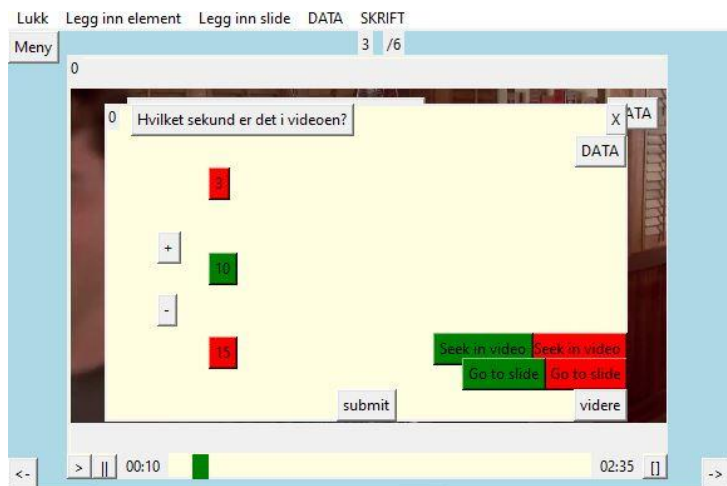
Figur 12 Til venstre: Et spørsmål lagt til i administratormodus, og riktig alternativ er markert som grønn. Til høyre: Det samme spørsmålet men i brukermodus, her har brukeren valgt riktig alternativ (grått), men ikke registrert svaret sitt enda.

Spørsmålene lagres i en egen tekstfil, den ser slik ut:



Figur 13 Måten et spørsmål er lagret på i programmet. Det er mulig med opptil 8 alternativer.

I videoelementet er det som nevnt mulig å legge til spørsmål som dukker opp i videoen, og som ut i fra svaret på spørsmålet tar brukeren til et sted i programmet, i administratormodus vil spørsmålet se ut som vist i Figur 14.

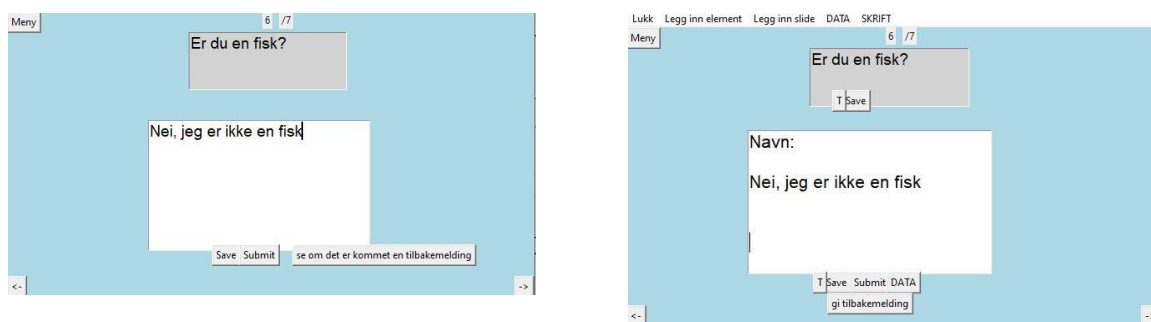


Figur 14 Spørsmål som er lagt til, åpnet i administratormodus.

Her gjenkjenner vi knappene som dirigerer brukeren ut i fra riktig eller feil svar nede i høyre hjørne. *Seek in video* vil sende brukeren til en gitt tid i videoen, mens *Go to slide* vil sende brukeren til en gitt slide. De grønne knappene er for riktig svar, og rød for feil.

4.2.6 Langsvar

Langsvarelementet er akkurat det samme som tekstelementet, men med noen ekstra funksjoner. Langsvarelementet er i motsetning til de andre elementene mulig å endre på i brukermodus. Langsvarelementet har en knapp som lar brukeren laste opp svaret sitt, samt en knapp som lar brukeren se om det er kommet en tilbakemelding på svaret.



Figur 15 Til venstre: Langsvarelementet som sett i brukermodus. Til høyre: Langsvarelementet i administratormodus, her har det kommet inn et svar fra «navn».

I administratormodus vil man med DATAknappen få opp alle svarene som er gitt av brukere, ved å trykke på gi tilbakemelding kommer det opp en ny tekstboks der man kan sende tilbakemelding til den respektive brukeren.

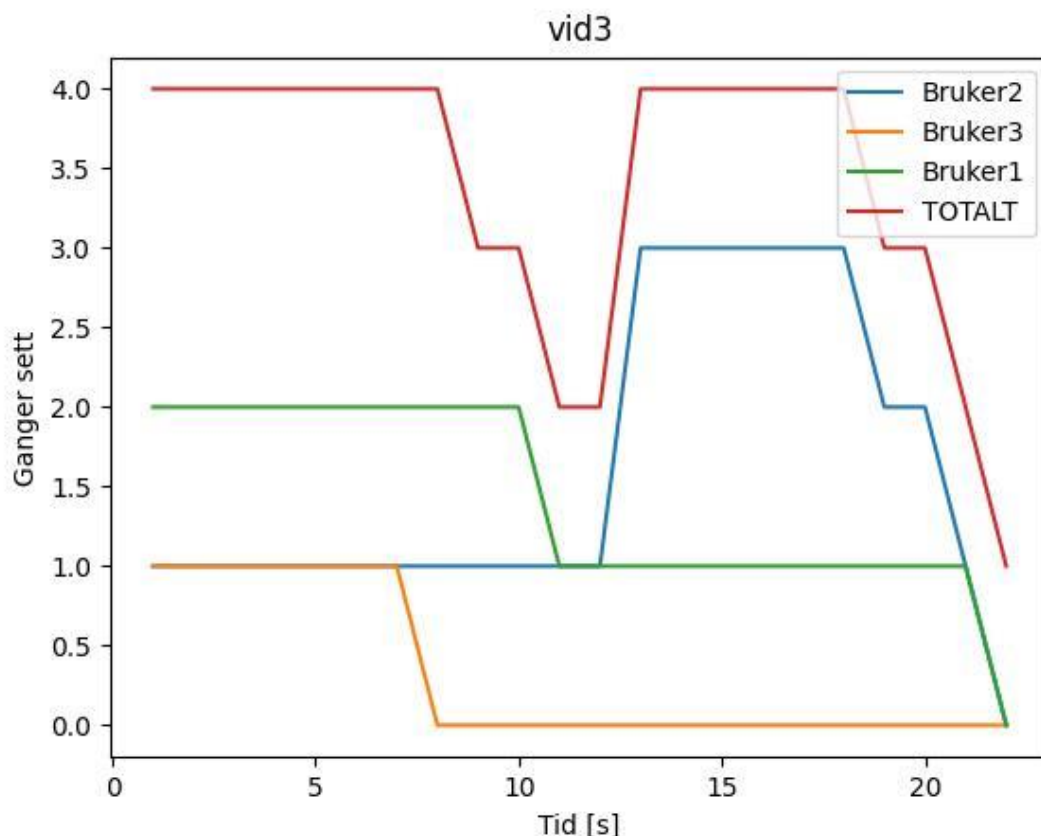
4.3 Analyseverktøy

Programmet vil registrere hvordan de forskjellige brukerne bruker programmet. Det vil registreres hvordan brukerne ser på videoer, hvilke svar de avgir på spørsmålene, og registrere svarene på langvaroppgavene.

For å laste opp resultatene brukes det en mailadresse som er laget spesifikt for programmet, hver gang programmet sender inn data, sendes det en mail. Innholdet i mailen er slik at det er mulig å hente det ut og skille det fra hverandre på en hensiktsmessig måte.

4.3.1 Video

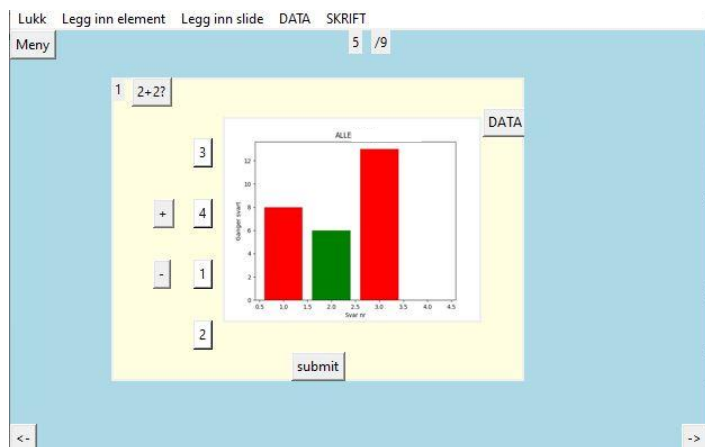
Mens brukeren ser på en video registreres dette, når brukeren setter på pause, eller videoen er ferdig vil det lastes opp, og være tilgjengelig for underviseren. Dette er et verktøy underviseren kan bruke for å videreutvikle undervisningen sin.



Figur 16 Videoanalyse i programmet, her kan man se at Bruker1 har sett starten av videoen to ganger, og slutten kun en gang, Bruker2 har sett starten av videoen en gang, men slutten tre ganger, Bruker3 så videoen frem til syv sekunder før han skrudde av. Totalen er hva brukerne har sett til sammen.

4.3.2 Quiz

Hvert spørsmål har i administratormodus en DATAknapp, denne brukes til å få opp resultatene på spørsmålet, her er det mulig å få opp resultatene for alle samlet, eller enkeltvis for hver bruker.



Figur 17 Hva som kommer opp dersom en trykker på DATAknappen, her ser vi at brukerne burde forbedre matteegenskapene sine.

4.4 utfordringer

Gjennom kodingen oppsto det flere større og mindre utfordringer, noen ble løst gjennom Googlesøk, andre gjennom mye tenking, mens andre ble kanskje ikke løst, men unngått. I dette avsnittet ønsker jeg å få frem noen av de største utfordringene jeg støtte på, og hvordan jeg valgte å løse dem.

I en slik oppgave kan man f.eks. skrive inn «jeg la inn et videoelement i programmet mitt» og ikke si noe mer. Men for å frembringe vanskelighetene, utfordringene og de eventuelle løsningene, er det riktig å sette av plass til dette. Denne delen inneholder også noen drøftingspunkter.

4.4.1 Video

Videoelementet var det klart vanskeligste elementet å få til å fungere. Det finnes ikke en egen videospiller i Tkinter, så jeg måtte lage en egen, dette viste seg å være problematisk, biblioteket *tkintervideo* ble benyttet. Dette biblioteket la kun inn video uten lyd, det hadde heller ingen knapper som lot deg spole eller sette på pause, disse måtte derfor kodes selv. For å få lyden til å fungere måtte det når videoen ble lagt inn, være en funksjon som hentet og lagret lyden i en

egen fil. Dette er ikke nødvendigvis å anse som en løsning på problemet, men en unngåelse av problemet. Det første som skjer når en bruker laster opp en video i programmet, er at den trekker ut lyden fra filen og lagrer det i en egen mp3-fil. Når videoen spilles av så vil både videofilen (mp4) og mp3-filen spille samtidig.

Videre oppsto det problemer med at videoen enten spilte for fort eller sakte, dette skjønnte jeg hadde med oppløsningen av videoen å gjøre fordi dersom videoen var i en liten boks, spilte den for fort, var den i en stor boks spilte den for sakte. En video er egentlig bare veldig mange bilder som kommer fort etter hverandre, dette tyder på at avspillingshastigheten avhenger av pc-ens yteevne. Det fantes funksjoner i biblioteket som angivelig skulle rette på avspillingshastigheten, men disse funksjonene fungerte ikke, eventuelt fikk jeg ikke dem til å fungere. For å løse dette måtte jeg endre på selve biblioteket. Jeg la inn en timer som sørget for a videoen ikke ville spille for fort, men det løste ikke at den spilte for sakte. For å løse dette la jeg inn noen linjer i biblioteket som hoppet over bilder dersom videoen var bak tidsskjema. Eks. hvis en video er 30 FPS, og programmet detekterer at videoen er ett sekund bak, så må programmet hoppe over 30 bilder. Alt i alt førte dette til at videoelementet er noe rotete, og løst med halvveisløsninger. Videoelementet avhenger av både videopløsning og pc-ens yteevne. Dette kan være problematisk dersom det lastes opp en video med høy oppløsning, eller brukeren har en dårlig pc. Dette var heldigvis ikke problematisk i gjennomføringen da videoene ikke hadde høy oppløsning, og brukerne hadde tilstrekkelige pc-er.

4.4.2 Kommunikasjon mellom klienter

For å kommunisere og lagre informasjon på nettet er det nødvendig å ha et felles kommunikasjonspunkt, altså en server. Når jeg jobbet med å finne ut av dette var det ikke noe problem å sette opp en lokal server gjennom Python, men det er jo ingen garanti for at underviser og elev er på samme sted. I tillegg ville det vært nødvendig å ha en påskrudd pc som drev severen. Det finnes nok mange løsninger på dette, men jeg gikk heller for en løsning som jeg visste var mulig, nemlig email.

For email er det som regel flere sikkerhetsfunksjoner som gjør at det ikke er mulig å logge på, sende og motta mail gjennom *usikre* lokasjoner. Å logge på en mailkonto gjennom et Pythonskript er å regne som en *usikker* lokasjon. Gmail har en funksjon kalt tredjepartstilgang, her er det mulig å opprette en tredjepartsinnlogging som Python kan logge inn på, det er slik kommunikasjonen er løst. Når det lastes opp svar, resultater og videoanalyse fra elevene, så sendes det en mail med informasjon. Når underviseren skal sjekke disse resultatene, så vil programmet

åpne innboksen, og lese av dataene. Dette er som i videoelementet ikke nødvendigvis en god løsning på problemet, men det var en oppnåelig løsning som fungerer på liten skala. Her er det også vektlagt arbeidsstrategien, jeg kunne satt meg inn i hvordan opprette servere for å lagre informasjon, men dersom det hadde blitt prioritert ville dette tatt ekstra tid, og da gått ut over kvaliteten til en annen del av programmet.

<input type="checkbox"/>	☆ meg	vid2,12345 - starther 4114124134144154164174184194204214222423424425426427428429430431432433434435436437438439440,
<input type="checkbox"/>	☆ meg	question10,12345 - starther 1,
<input type="checkbox"/>	☆ meg	vid2,12345 - starther 127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,
<input type="checkbox"/>	☆ meg	vid2,MrTest - starther 4114124134144154164174184194204214222423424425426427428429430431432433434435436437438439440,
<input type="checkbox"/>	☆ meg	question10,MrTest - starther 1,
<input type="checkbox"/>	☆ meg	vid2,MrTest - starther 355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392
<input type="checkbox"/>	☆ meg	vid2,MrTest - starther 346,347,348,349,350,351,352,353,354,
<input type="checkbox"/>	☆ meg	vid2,MrTest - starther 244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281

Figur 18 Utdrag av innboksen, dette er skjult for alle brukerne av programmet.

Figur 18 viser innkommende mail, disse er for underviseren ubrukelig, men programmet oversetter det til leselige format som vist i Figur 16 og Figur 17.

5 Resultater og analyse

Kapittel 5 vil presentere resultatene fra gjennomføringen, samt spørreundersøkelsen.

En kan anse selve programmet som et resultat, men da det er en så sentral del av oppgaven, og samtidig «kun» er et forarbeid før problemstillingen besvares, det ble derfor omtalt i et eget kapittel.

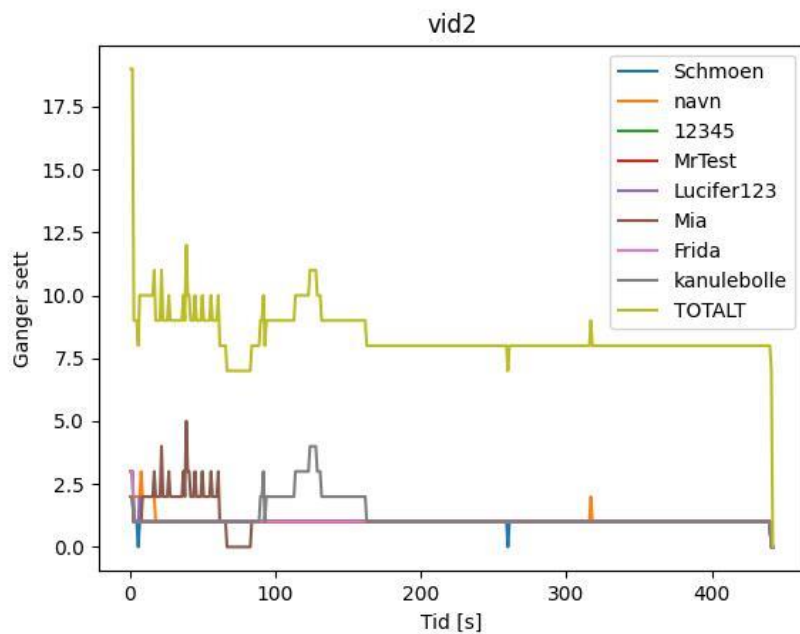
Deler av resultatene som presenteres under vil være ikke-målbare. Det å benytte programmet, er å anse som et resultat, egne oppfatninger av hvordan det var å bruke programmet vil være en del av resultatene. I kapittel 6 drøftes det rundt dette.

5.1 Gjennomføring

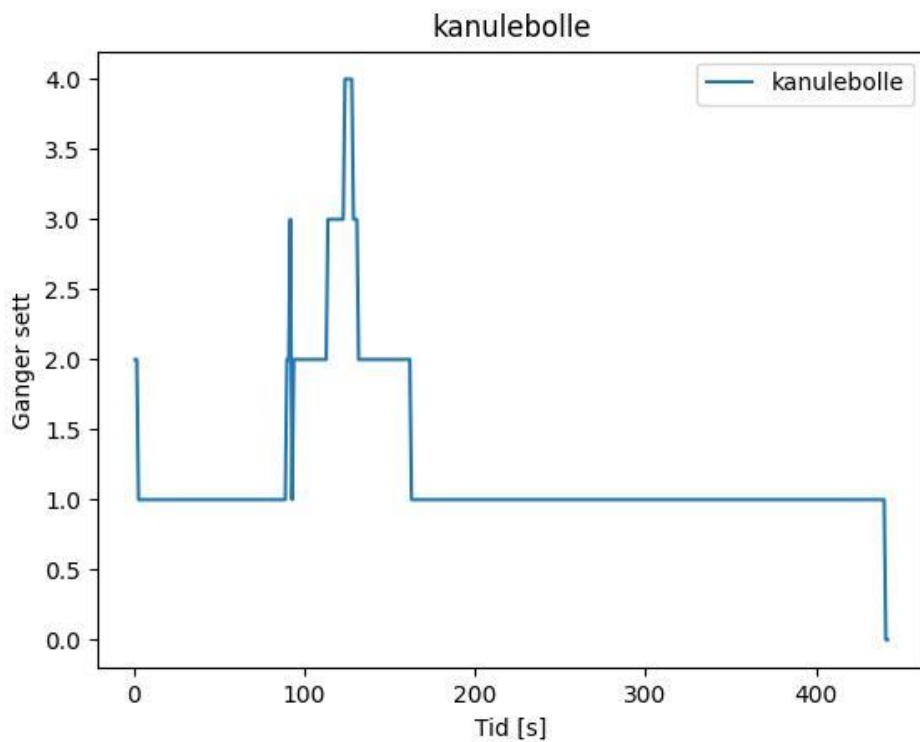
Første del av gjennomføring gikk ut på å lage en undervisningstime i programmet, dette er da å utarbeide en mappestruktur som kan åpnes av programmet. Veileder av oppgaven er lærer ved Sjøkrigsskolen, en av klassene til veileder ble spurt om de ønsket å benytte programmet og svare på en spørreundersøkelse. Da læreren også var veileder var han godt informert om problemstillingen, og kjente godt til intensjonen samt baktanken til problemstillingen. Læreren ble så bedt om å gjøre klar et opplegg som kunne legges inn i programmet. Det ble gjort klart tre videoer og noen tilhørende spørsmål til hver video. I samråd med læreren ble dette opplegget lagt inn i programmet. Det gikk greit å legge inn videoene og spørsmålene. Funksjonene som ble brukt var video, spørsmål, tekst og langsvaer. Spørsmålene var alle knyttet opp mot en video, men det var ikke lagt inn noen handlinger ut i fra om brukeren svarte rett eller galt, dette ble derfor ikke testet i klassen.

5.1.1 Videoanalyse

I gjennomføringen ble det brukt tre videoer, det er ikke nødvendig å ta med resultatene fra alle tre, så det er kun valgt å ta med analysen av én av videoene. Videoanalysen fungerte slik den er programmert til, men en kan se at den trenger finpussing dersom den skulle blitt brukt på en større gruppe. Det presiseres at disse figurene under er produsert gjennom programmet og kan også fremstilles i programmet. Bildefilene er lagt direkte inn i dette dokumentet for å få bedre oppløsning.



Figur 19 Viser seermønsteret til de åtte brukerne av programmet. Totalen er den øverste linjen. De fleste så hele videoen ca. 1 gang, det er da mange linjer oppå hverandre, det noteres at ingen hoppet over videoen, da ville det vært en strek på 0. Alle har sett hele videoen.



Figur 20 Seermønsteret til brukeren kanulebolle.

5.1.2 Quizresultater

Det er valgt å ikke ta med quizresultater, da resultatene ikke er relevant for oppgaven. Hvordan resultatene fremstilles er vist i *Figur 17*.

5.1.3 Langsvarelementet

Det er valgt å ikke ta med resultater fra langsvarelementet, da svarene avgitt ikke er relevant for oppgaven. Hvordan langsvarene fremstilles er vist i *Figur 15*.

5.1.4 Oppfatninger og observasjoner

Egen oppfatningen av programmet er at videoene spilte av greit, og at spørsmålene var greie å besvare. Det ble oppfattet som irriterende når programmet brukte lang tid på noen funksjoner. Den funksjonen som i noen tilfeller gjorde at programmet bufret over lengre tid var *se om det er kommet tilbakemeldinger-knappen* under langsvarelementet. Dette var allerede et kjent problem, men ble et enda større problem enn først antatt. Dette skyldtes trege pc-er, og data fra åtte brukere samtidig, tidligere hadde programmet kun blitt testet av en bruker av gangen, da hadde det ikke blitt oppfattet som et stort problem.

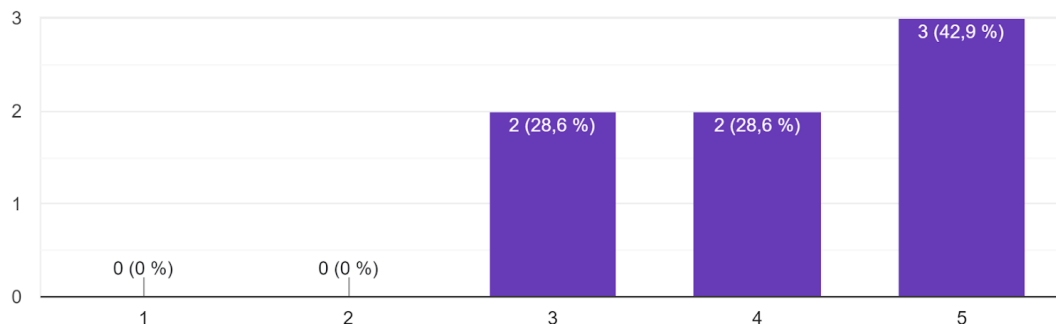
Jeg støtte på et stort problem rett før undervisningsøkten skulle gjennomføres, jeg var uheldig og slettet den ene videoen, da ble i utgangspunktet alle spørsmålene tilhørende videoen også slettet. Heldigvis gjorde jeg feilen selv, og visste hvordan jeg kunne gå inn i mappestrukturen og fikse det. Dersom en lærer eller noen andre hadde gjort tilsvarende ville nok arbeidet vært borte, og måttet bli gjort på nytt.

5.2 Spørreundersøkelse

Det ble lagt vekt på at spørreundersøkelsen ikke skulle ta lengre tid enn ti minutter, det er derfor mulig å presentere alle flervalgsspørsmålene i oppgaven. Klassen det ble testet på var kun åtte elever, der syv av dem besvarte undersøkelsen.

Fungerte læringsverktøyet? (5 er ja)

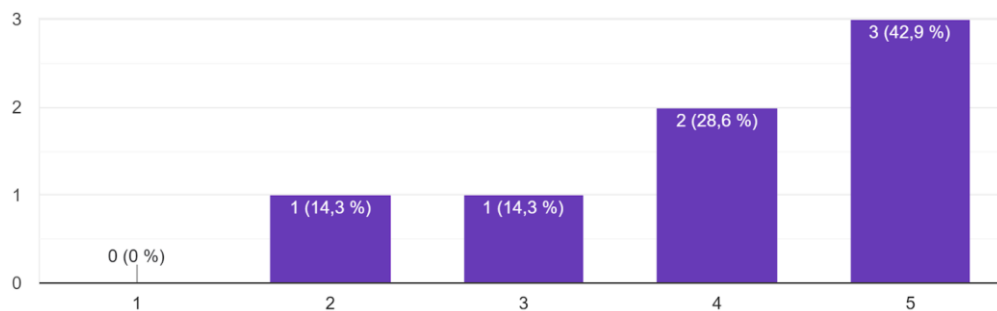
7 svar



1

Fungerte læringsverktøyet som erstatning for "vanlig" undervisning? (5 er ja)

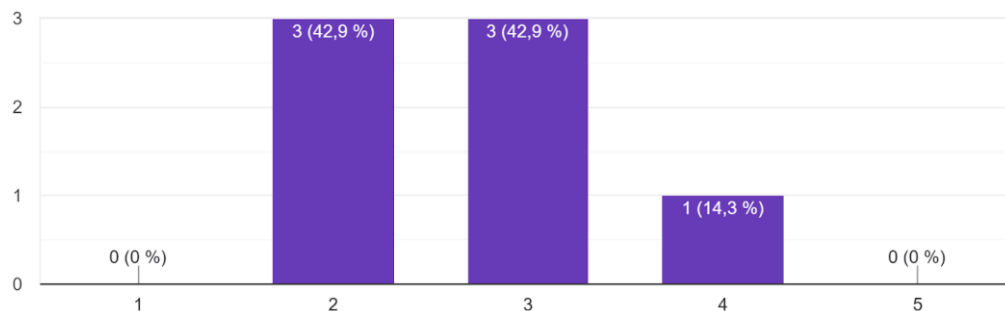
7 svar



2

Kunne du tenke deg å bruke det fremfor vanlig undervisning fremover? (5 er ja)

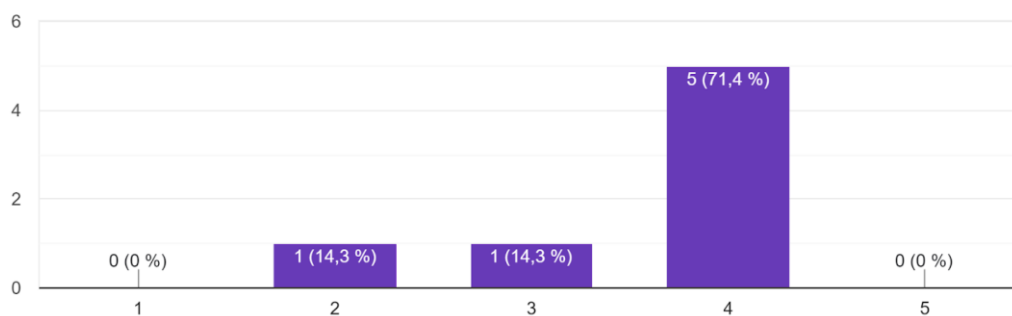
7 svar



3

Hvis en profesjonell utvikler hadde utviklet et lignende program, ville du foretrukket det over vanlig undervisning? (5 er ja)

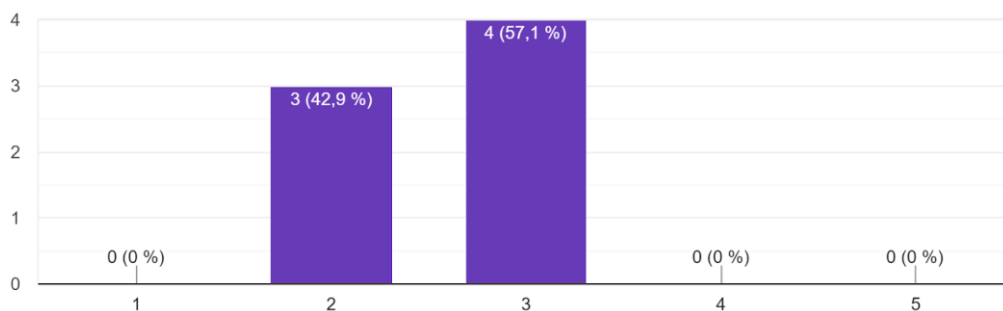
7 svar



4

Tror du at et e-læringskurs er bedre enn en vanlig undervisningstime? bruk hele skalaen, (5 er JA) (1 er NEI)

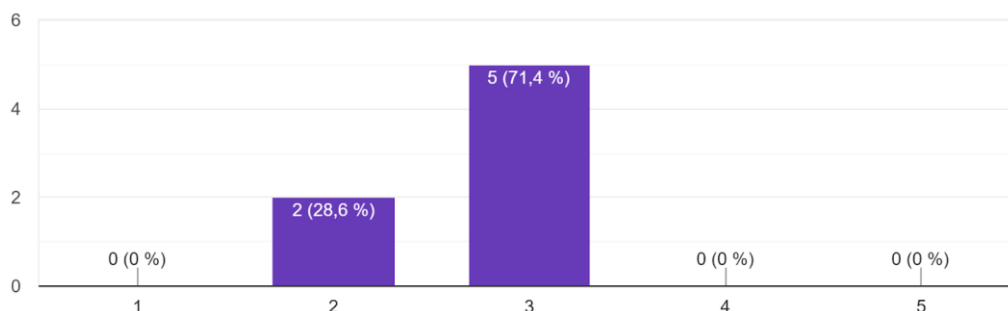
7 svar



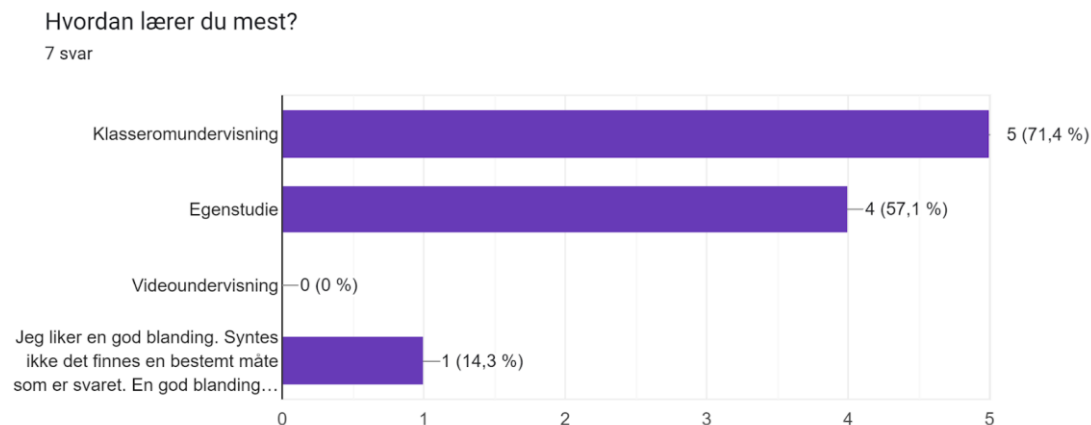
5

I hvilken grad føler du at elektroniske verktøy brukes i undervisning? (1 brukes for lite, 5 brukes for mye)

7 svar



6



7

Figur 21.[1,2,3,4,5,6,7] Spørsmålene som ikke var fritekst som ble stilt i spørreundersøkelsen, 1-6 er spørsmål der det skal besvares på en skala, 7 er et åpent spørsmål om hvordan hver elev lærer mest, med mulighet for å legge til alternativ selv.

Spørreundersøkelsen inneholdt også to fritekst spørsmål. Det ene etterspurte tilbakemeldinger om programmet, mens det andre spørsmålet ba elevene om å sammenligne elektronisk og tradisjonell undervisning. Svarene på disse spørsmålene er gitt i vedlegg A.

Det var to punkter som gikk igjen i svarene som ble gitt, det første var at elektronisk undervisning ga en større grad av frihet, denne friheten brukes på både godt og vondt. Noen liker at det er mulig å gjøre ting i sitt eget tempo, andre likte at det var mulig å jobbe foran skjema. På den negative siden var det lettere å bli distraheret av andre ting, mens noen skrev at de ble sløvere av å ha all undervisning på skjerm.

Det andre punktet var at ved elektronisk undervisning er det ikke mulig å stille spørsmål dersom en lurer på noe, terskelen for å stille spørsmål er også nevnt som lavere i en fysisk undervisning. Av de syv som besvarte undersøkelsen hadde hele seks elever nevnt interaksjon med lærer som en av de positive sidene ved tradisjonell undervisning.

6 Drøfting

I denne delen vil sentrale deler av oppgaven drøftes. Først vil noen valg i oppgaven diskuteres, deretter vil resultatene fra spørreundersøkelsen diskuteres og sammenlignes med relevant litteratur. Videre vil det rettes et kritisk blikk mot oppgaven som helhet. Det fremstilles så noen anbefalinger til et fremtidig arbeid innenfor samme tema. Avslutningsvis drøftes det rundt min kognitive bias mot oppgaven.

6.1 Avgrensninger

Det er to sentrale avgrensninger i oppgaven, den første omhandler egne evner til å programmere, dette kan og tolkes som en begrensning. Sett i lyst av problemstillingen er det å anse som en avgrensning, da jeg selv har valgt å programmere store deler av programmet, fremfor å hente ferdigskrevet kode fra diverse kilder. Dersom problemstillingen ikke hadde inneholdt *kan jeg?* er jeg overbevist om at det kunne vært mulig å sette sammen et bedre program ved å bruke allerede utviklet kode. I så måte kunne man også satt seg inn i makroer i PowerPoint, og linket til YouTubevideoer og fått et resultat som i helhet ville lignet på programmet. Denne avgrensningen gjorde at kvaliteten på produktet trolig var lavere enn det kunne ha vært, men mål syv som omhandlet å bli bedre på å programmere ble i større grad oppfylt grunnet avgrensningen.

Den andre avgrensningen i oppgaven er kunnskapen om informasjonsinnhenting og spørreundersøkelser. Det er laget en spørreundersøkelse som gir svar på sentrale spørsmål i problemstillingen, men det er ingen faglig strategi bak designet av undersøkelsen. En faglig bakgrunn i dette feltet ville med all sannsynlighet gitt bedre resultater, samt mulighet til å besvare problemstillingen med høyere presisjon. På en annen side ble det viet mer tid til selve programutviklingen, noe som ga et bedre ingeniørfaglig produkt.

6.2 Valg av kilder

Det er valgt å bruke google-søk som hovedkilde i oppgaven, dette for å leve opp mot problemstillingen som omhandler elektronisk undervisning.

Det skal ikke legges skjul på at forfatteren ikke har lest en eneste bok om Python-programmering foruten et fag ved Universitet i Bergen i 2016. All ny kunnskap tilegnet om programmering er gjort gjennom Google og YouTubevideoer. Dette anses ikke som en negativ påvirkning på

oppgaven. Å finne kunnskap gjennom Google er lett og raskt, noe som er kritisk når en programmerer. For min del ble mesteparten av tiden benyttet til læring, tiden brukt på faktisk koding er vesentlig mindre enn tiden brukt på Googlesøking. Det er selvsagt en risiko for å finne feilinformasjon, men gjennom kritiske øyne anses risikoen for dette som liten.

Når det gjelder funn av kilder, er samme fremgangsmåte brukt, selv med vanlige Google-søk virker det som at det var fullt mulig å finne masse troverdig informasjon. F.eks. er utdanningsdepartementet i USA sin artikkel funnet gjennom et vanlig Google-søk, med mindre noen har forfalsket denne artikkelen er det å anse som en god og troverdig kilde til informasjon.

6.3 Arbeidsstrategi

Arbeidsstrategien gikk ut på utvikle programmet helt frem til en tidsfrist, fremfor å nå konkrete mål. Fordelen med å bruke dette som strategi er at man aldri vil bli ferdig med utviklingen. Selv om tilstrekkelig resultat er oppnådd, fortsetter arbeidet. På en annen side kan dette være en fallgrube, der man slapper av og har tankegangen om at arbeidet gjort er irrelevant, sluttresultatet kommer uansett til å være det produktet en sitter igjen med ved tidsfristen. I tilfelle av denne oppgaven ble begge sider av strategien tydelige. Det ble jobbet godt med både *må-kravene* og *bør-funksjonene*, når disse var på plass var det en stor oppgave å starte på *far-fetched funksjonene*. Jeg er av den oppfatningen at den valgte arbeidsstrategien til tider førte til høy effektivitet, mens andre ganger førte til ingen effektivitet. Gjennomsnittlig oppleves effektiviteten som akseptabel.

En del av arbeidsstrategien var en kontinuerlig vurdering av valget om å lage nytt, eller forbedre gammelt. For det meste falt valget på å lage nytt, dette medførte at jeg fikk inn samtlige av *ønskelig-funksjonene*, men på bekostning av brukervennlighet, utseende og kodeskikk. Den største problematikken med dette er at det vil være vanskelig for noen å fortsette på det arbeidet som er gjort, slik programmet er i dag. Dersom koden skulle blitt distribuert ville den vært relativt vanskelig for andre å sette seg inn i. Det er selvsagt fulgt standard kodeskikk, koden er ikke bevisst blitt gjort rotete, men når det som regel har blitt prioritert utviklingen av nye elementer fremfor å rydde gammel kode, så resulterer det i en viss grad av spagettikode.

6.4 Programmet

Den beste metoden for å vurdere et produkt er å teste det, det er det også gjort, men i liten skala kun på én lærer, og én klasse. Ideelt burde programmet blitt distribuert til så mange undervisere

og klasser som mulig, dette er ikke prioritert grunnet knapphet på tid. Drøftingen av programmet er derfor i stor grad basert på egne oppfatninger og observasjoner.

En god måte å vurdere programmet på vil være å sammenligne det med kravene og funksjonene som ble satt før oppgaven.

Samtlige av må-kravene og bør-funksjonene ble oppnådd, ingen av *far fetched-funksjonene* ble oppnådd.

Må-kravene var forholdsvis lette å oppnå, kravene som gikk ut på å legge inn elementer gikk greit, her var den største utfordringen å på en god måte la brukeren av programmet endre størrelse og plassering av elementene. Denne redigeringen fungerte, men kunne til tider være irriterende dersom det var viktig å plassere elementene nøyaktig, det var ikke i gjennomføring noe nødvendighet for nøyaktig plassering, men det ville nok vært et problem hvis programmet skulle blitt brukt på større skala. Video-elementet var som nevnt i 4.4.1 det klart vanskeligste elementet å få til å fungere. Da det ikke fantes en egen videospiller i Tkinter måtte jeg lage min egen, noe som viste seg å ikke være lett. Løsningene benyttet i videoelementet skader programmet med tanke på videreutvikling, men har på oppgavens skala liten betydning. Selv om det ble benyttet en løsning som kanskje unngikk de faktiske problemene, så vet jeg ikke hvor jeg i det hele tatt skulle startet for å finne en god løsning på problemene med videoelementet. I alt er jeg veldig fornøyd med at fungerer som det skal, det er dessuten ikke merkbart for brukeren.

Bør-funksjonene gikk i all hovedsak ut på å finne en metode å laste opp og ned data på. Her gikk jeg for en interessant løsning, som var å bruke en mailkonto som erstatning for en server. Som i videoelementet er denne løsningen ikke ideell, men heller en unngåelse av problemet, i dette tilfellet er jeg også fornøyd med at det fungerer og for så vidt er usynlig for brukerne av programmet.

Kommunikasjon mellom elev og underviser er løst gjennom langsvarelementet, dette langsvarelementet kunne til fordel blitt markert som noe annet. Det er lagt inn til hensikt å være et spørsmål der elevene svarer med tekst, men det kan uten problem også brukes til generell kommunikasjon, og ikke minst fungere som en arena der elevene kan stille spørsmål til underviseren. Det er i spørreundersøkelsen og (Tamm, 2022) kartlagt at den viktigste fordelen med tradisjonell undervisning er interaksjon med lærer.

Videoanalysen fungerer som den skal, ut fra *Figur 19* kan man se at linjen med den totale seertiden er veldig høyt over den enkelt bruker sin tid, her burde det heller vært brukt gjennomsnitt fremfor total. Det var ikke et stort problem da det var få brukere, men på en større skala ville dette gjort grafen uleselig for underviseren.

Quizresultatene fremstilles som vist i *Figur 17*, dette oppfatter jeg som løst på en grei måte. Merk at det er også mulig å få opp resultater for enkeltindivid dersom det er nødvendig. En funksjon som kanskje ville vært grei å ha her ville vært muligheten til å samle inn resultat fra flere spørsmål samtidig, hvis det f.eks. skal gjennomføres en test der en må ha syv av ti riktige så kunne programmet fremstilt hvem som besto og hvem som ikke besto. Det er i (U.S Department of Education, 2009) nevnt at quiz ikke nødvendigvis er en god metode for å lære, det å kunne ha individuelle resultater og tester i programmet er en god egenskap, men for å la programmet fungere som et testverktøy og heller et læringsverktøy kunne det til fordel fjernet mulighet for å se individuelle resultater.

Å ha programmet kjørbart gjennom kun én fil var ikke essensielt for utviklingen av selve programmet, men var høyst viktig for distribuering av programmet. Dersom en bruker måtte laste ned Python, biblioteker og samtlige filer ville programmet ikke kunne blitt omtalt som *distribuerbart*.

Far-fetched funksjonene ble ikke prioritert, arbeidsstrategien tilsa at dersom noen av disse funksjonene ble prioritert ville det gå hardt ut over andre, viktigere funksjoner. Det ble gjort et forsøk på nettleserkompatibilitet, men etter utallige googlesøk konkluderte jeg med at det ikke var verdt det. For å få til nettleserkompatibilitet måtte jeg først opprettet en server, noe jeg allerede hadde valgt å ikke gjøre. Videre måtte jeg så finne en måte å laste programmet opp på serveren. Nettleserkompatibilitet ville gjort programmet langt mer *distribuerbart*, og ville ført til store muligheter for massetesting. Fremfor å teste programmet på kun én klasse, kunne det blitt testet i langt større skala, noe som ville ført til en mer inhomogen testmasse som videre ville gitt spørreundersøkelsen mer hold.

Mobilkompatibilitet hadde samme fordelene som nettleserkompatibilitet. Mobilkompatibilitet ville inneholdt apputvikling og ble ansett som vanskeligere enn nettleserkompatibilitet, det ble derfor tidlig klart at det ikke kom til å bli en prioritet.

Interaktive simuleringer i oppgaven ville vært mulig å gjøre, og heller ikke så vanskelig å implementere. Utfordringen med simuleringene er at de må være programmert av læreren, og programmert spesifikt i Python, samt bruke Tkinter som grunnlag for GUI. Dersom en lærer har

kompetansen vil det være fullt mulig å implementere noe slikt i programmet. Interaktivitet i programmet vil aktivisere elevene og føre til læring. (El- Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014) og (U.S Department of Education, 2009).

6.5 Resultater

Programmessig fungerte videoanalysen, quizen og langsvarspørsmålet, det var visse problemer med innlastningen, men det var kun tidsmessig. Disse tre elementene anses å være vellykket, og utført.

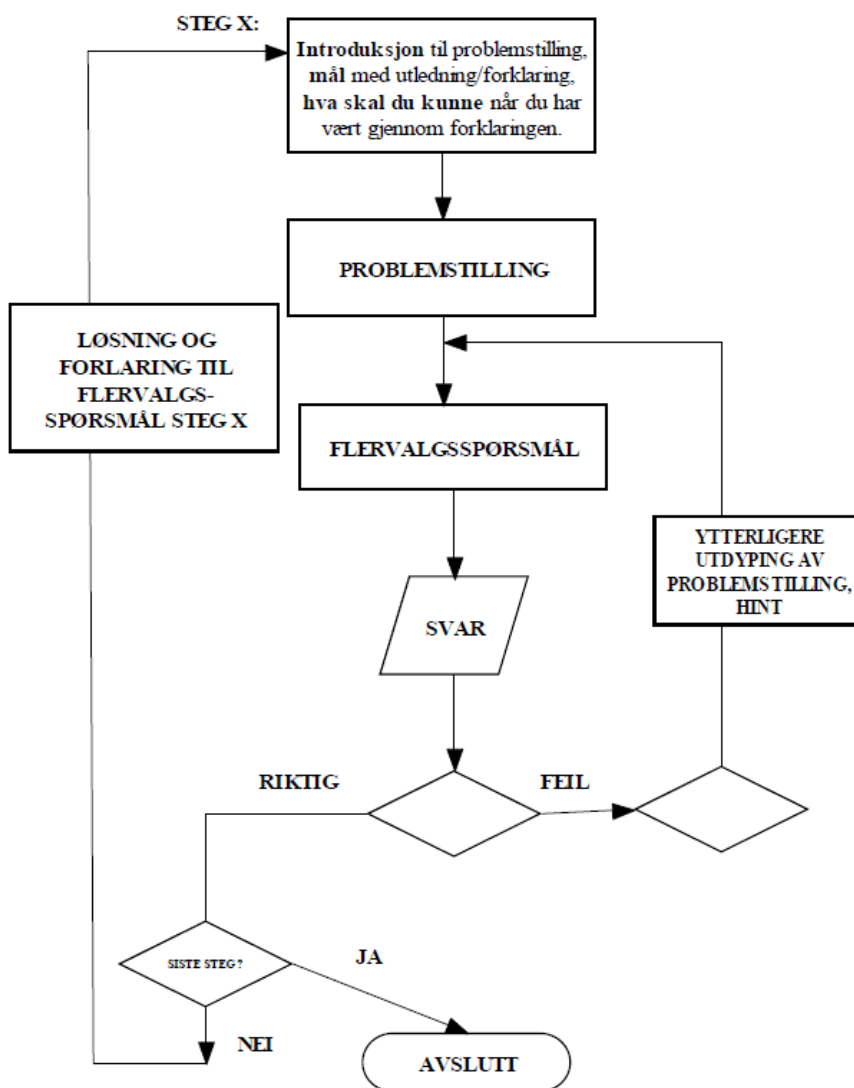
6.5.1 Videoanalyse

Videoanalysen anser jeg som en av de viktigste funksjonene til programmet, det å kunne registrere hvor elevene setter på pause, hvilke deler de ser om igjen, og hvilke deler de eventuelt hopper over kan være god tilbakemelding for en underviser. Da underviserens eventuelle mangel på opplæring i metodikk er en utfordring (El- Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014) påstår jeg at en slik videoanalyse er et hjelpemiddel underviseren kan bruke til å forbedre sin undervisningsmetodikk. Ved å klippe bort, eller forklare deler av en video der elevene spoler eller hopper over, eller å spille inn en del på nytt der det ser ut til at elevene har måttet se videoen flere ganger så kan underviseren gradvis forbedre sine forelesninger og sin metodikk.

6.5.2 Quiz

Det kan drøftes om quiz i seg selv er en effektiv metode for læring. (El- Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014) hevder at interaktivitet er en av fordelene med e-læring. På en annen side konkluderte ED med følgende påstand: «The providing of simple multiple-choice quizzes did not appear to enhance online learning» Videre påpeker også ED at individuell instruksjon, altså at programmet reagerer på brukerens spørsmål, svar og fremgang viste seg å ha en positiv virkning på læring. Programmet mitt gir underviseren mulighet til å legge inn en handling ut i fra om det svares riktig eller galt på spørsmål, men dette ble ikke benyttet. Dersom konklusjonene til ED er riktige kan en se for seg at en eventuell videreutvikling av programmet ikke bør fokusere på selve quizen, men heller veien videre ut i fra resultatet på quizen.

Et arbeid som i stor grad omhandler diskusjonen over er allerede gjort ved Sjøkrigsskolen. I 2003 arbeidet Terje Fykse med en funksjon i itslearning kalt *forklaringsmodul*. Denne baserte seg i stor grad på å sende elever fremover eller bakover i programmet ut i fra om de svarte riktig eller galt på et spørsmål. Skissen under var ideskissen for *forklaringsmodul*.



Figur 22 (Fykse, 2003). Ideskisse for forklaringsmodul. Finnes i vedlegg G

Det er altså likheter mellom konklusjonene jeg har kommet frem til, og tidligere arbeider på området. Det er ved Sjøkrigsskolen tidligere forsket og utviklet program som til en viss grad er lik mitt eget. Grunnet avgrensningen som tilsa at jeg ønsket å lage et eget produkt ble ikke tidligere produkter som dette undersøkt før slutten av prosjektet. Det kunne vært en fordel å legge skisser som dette til grunn for oppgaven, det er samtidig betryggende at mitt og Fykse sitt program kom frem til lignende læringsstrategier. *Forklaringsmodul* var i en periode en funksjon

tilgjengelig i itslearning, men *forklaringsmodul* ble på ett tidspunkt avviklet. Dette er for så vidt ikke betryggende for eget program. *Foklaringsmodul* ble laget i 2003, da teknologien var helt annerledes enn i dag, kanskje et lignende program ville blitt bedre tatt i mot i dag.

6.5.3 Langsvarspørsmål

Langsvarspørsmålet fungerte, det ble samlet inn en del svar, og gitt tilbakemeldinger som dukket opp hos elevene. Det var ikke særlig problematisk for elevene å laste opp sine svar i langsvarelementet, det som derimot var problematisk var når de skulle se om de hadde fått en tilbakemelding på svaret sitt. Når de så etter denne tilbakemeldingen måtte programmet laste ned data fra email, noe som var tidkrevende, programmet frøs i lengre perioder grunnet dette. Dette er da en direkte ulempe av å velge å ikke sette opp en server. Det er allerede kartlagt at interaksjon med underviser er sentralt, langsvarspørsmålet kunne da til fordel vært brukt som en toveis kommunikasjonsportal, fremfor utelukkende langsvarspørsmål.

6.5.4 Spørreundersøkelse

Resultatene av spørreundersøkelsen er det eneste i oppgaven som er et selvprodusert resultat som ikke er direkte påvirket av mine egne oppfatninger. Som leser kan man ut fra spørreundersøkelsen trekke egne konklusjoner.

Videre skal betydningen av spørsmålene i spørreundersøkelsen diskuteres.

Figur 21.1 og Figur 21.2 fremstiller et svar på i hvilken grad første del av problemstillingen er oppnådd. *Kan jeg programmere et verktøy undervisere kan benytte som erstatning for tradisjonell undervisning?* I figurene er det en antydning mot ja, noe som antyder at første del av problemstillingen er løst. Fra *Figur 21.3* kommer det derimot frem at jeg ikke nødvendigvis har programmert et program som ønskes brukt.

Det var ikke overraskende at programmet ikke var en ønsket undervisningsform, men for å ikke kun evaluere egenutviklet program, ble det stilt spørsmål om selve ideen, eller konseptet bak programmet. Av *Figur 21.4* kommer det frem at dersom programmet hadde blitt utviklet av en profesjonell utvikler, så kunne det vært foretrukket over vanlig undervisning. Svaret er betydelig nærmere ja, enn ved eget program. Dette tyder på at et slikt program kan være hensiktsmessig å bruke, dersom det er gjort ordentlig. Dette støtter opp om påstanden til (Zheng, Bender, & Lyon, 2021) om at vellagd online undervisning øker motivasjon, tilfredshet og læring.

Figur 21.4 besvarer til dels andre del av problemstillingen *Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?* Det er vanskelig å trekke konklusjoner ut i fra kun egen spørreundersøkelse, den tradisjonell undervisningen ser ut til å ikke være veldig langt unna å kunne bli erstattet av elektroniske løsninger. (Zheng, Bender, & Lyon, 2021) som studerte resultater ved en tannhelseutdanning konkluderte med at online-kurs gjennom koronapandemien oppnådde resultater tilsvarende eller bedre enn de fysiske kursene før pandemien.

Figur 21.5, 6 og 7 besvarer til dels også andre del av problemstillingen, men ut i fra etterpåklokskap kommer det frem at spørsmålene ikke var særlig relevante for problemstillingen. Spørsmålene gir en antydning mot at tradisjonell undervisning er bedre, men problemstillingen tar heller for seg om elektroniske løsninger *kan* ikke om det *er bedre*.

Spørsmålene som ble stilt i fritekst ga gode innblikk i hva én klasse tenker om elektronisk sammenlignet med tradisjonell undervisning. Det var klart at det å kunne direkte interagere med en lærer var svært viktig for elevene. I programmet er det mulighet for interaksjon med en lærer gjennom langsvarelementet, men dette fungerte tregt og ble ikke brukt i stor grad i undervisningsøkten. De negative sidene ved e-læring fra spørreundersøkelsen samsvarte med (Tamm, 2022) Det å kunne ha en direkte interaktivitet med læreren er noe ikke bare klassen ved Sjøkrigsskolen verdsetter høyt. Å evne å holde fokus og balansere eget tidsfordriv er å finne igjen i klassen ved Sjøkrigsskolen og (Tamm, 2022).

Den siste delen av spørreundersøkelsen inneholdt fritekst, fordelen med å kunne jobbe hvor, når og i hvilket tempo man selv ønsket stemmer overens med ED sin påstand om at e-læring har blitt populært grunnet friheten til å velge hvor og når undervisningen finner sted. (U.S Department of Education, 2009)

En tilbakemelding i spørreundersøkelsen oppsummerte problemet godt.

«Det kan bli veldig bra med elektroniske forelesninger og notater liggende tilgjengelig, men dette krever en god struktur på innholdet fra læreren sin side. Kan fort bli destruktivt hvis det blir rotete» (vedlegg A)

Denne tilbakemeldingen treffer godt mot bakgrunnen av oppgaven, at en foreleser eller lærer kan bruke god tid på å lage et godt elektronisk opplegg, som kan brukes og forbedres i flere år, fremfor å holde den samme middelmådige forelesningen hvert år. Denne tilbakemeldingen kommer riktignok kun fra én elev, det utelukker dog ikke at andre elever er enige i påstanden. Uansett traff tilbakemeldingen meget godt i forhold til oppfatningen jeg selv har om elektronisk undervisning, det støtter og opp om påstanden til (El- Seoud, Taj-Eddin,

Seddiek, El-Khouly, & Nosseir, 2014) om at undervisningen avhenger av lærerens opplæring i metoden.

Det skal ikke utelukkes at jeg er et offer for kognitiv bias, når jeg vektlegger denne tilbakemeldingen. Det skal heller ikke utelukkes at det finnes flere, og større utfordringer med elektronisk undervisning, men det er akkurat dette jeg selv også oppfatter som *løsningen* for å tre inn i en verden der elektronisk undervisning er normalen, og den tradisjonelle undervisningen er unntaket.

6.6 Oppgaven som helhet

For å diskutere oppgaven som helhet er det rimelig å sammenligne sluttsituasjonen med målene satt i starten av oppgaven. Målene gjengis under.

1. Ha det kjekt mens det arbeides med oppgaven
2. Lage et produkt som kan benyttes iht. problemstillingen
3. Benytte produktet
4. Få tilbakemeldinger på produktet
5. Få tilbakemeldinger rundt problemstillingen
6. Tilegne seg kunnskap om temaet elektronisk undervisning
7. Bli bedre på å programmere

Det første målet er en subjektiv vurdering, det er i mine øyne nådd, i hvert fall frem til skriveingen av selve oppgaven. For meg er det viktigste for et prosjekt en skal jobbe med i flere måneder å ha det kjekt. Derfor valgte jeg en oppgave som i stor grad gikk ut på programmering, noe jeg trives med å gjøre. Å skrive selve oppgaven kommer man ikke utenom, målet vil jeg anse som nådd.

Den andre målet er til dels nådd, som diskutert under punkt 6.4 kan produktet brukes som erstatning, men ikke nødvendigvis som en god erstatning.

Mål nr. tre, fire og fem er utført.

Mål nr. seks er gjennomført. Gjennom i hovedsak (U.S Department of Education, 2009) og (El-Seoud, Taj-Eddin, Seddiek, El-Khouly, & Nosseir, 2014) har jeg tilegnet meg mye kunnskap rundt temaet, og evnet å sette mine egne resultater opp mot tidligere forskning. Mål nr. seks kunne til fordel for oppgaven vært mer omfattende. Programutviklingen var regnet som hoveddelen av oppgaven, da synker kvaliteten på litteraturstudien.

Mål nr. syv er i stor grad nådd, jeg har utviklet meg på flere områder innenfor programmering, spesielt i Python. Tidligere hadde jeg kun programmert med klasser i C++, dette har jeg nå fått god kjennskap til i Python og. Det elementet jeg konkret kan fortelle at jeg har lært er *decorators*. Uten å gå inn på hva det er, så er det et veldig konkret element jeg er glad for at jeg har lært, hadde jeg kunnet mer om dette når jeg startet oppgaven ville koden vært vesentlig annerledes, trolig mer effektiv og oversiktlig. Når jeg lærte om *decorators* måtte jeg ta en vurdering på om jeg skulle fortsette med det jeg hadde, eller gå bakover og implementere den nye kunnskapen, valget falt på å gå videre, da en implementering av *decorators* i praksis ville vært å kode hele programmet på ny. Dette er et eksempel på at begrensede kunnskaper om programmering var et tydelig hinder for oppgaven.

Mot slutten av oppgaven opplevde jeg at jeg kunne implementere *alt*, det stod kun på tid. Denne følelsen er noe jeg har hatt mens jeg har jobbet med andre prosjekter, og er en indikasjon på at man mestrer det man holder på med. Dette styrker opp om at mål nr. syv i stor grad er oppnådd.

6.7 Viderebygging

Dersom en ønsker å ta steget videre med et slikt program ville jeg anbefalt å sette et stort fokus på å etterligne en interaksjon med underviser. Dersom langsvarelementet hadde blitt lagt til på hver slide, og blitt merket med «still dine spørsmål her», kunne dette vært ett steg mot mer høyere interaktivitet. Samtidig tror jeg en metode for å komme nærmere underviseren vil være å lage en FAQ-liste. Denne vil inneholde de spørsmålene læreren har blitt stilt før, og være tilgjengelig for alle. Å prioritere interaktiviteten er det det viktigste elementet. Som nevnt i 6.5.2 er det ikke selve quizen som er viktig, men hva som skjer etter quizen.

Asynkron undervisning vil aldri bli like interaktivt som synkron undervisning, men å prøve å etterligne interaktiviteten fra den synkrone undervisningen bør være hovedprioriteten i et læringsverktøy som har til hensikt å erstatte tradisjonell undervisning

Det anbefales ikke å bygge videre på programmet brukt i denne oppgaven, jeg har lært veldig mye nytt mens jeg har utviklet programmet og ville selv starten fra bunnen av hvis jeg skulle lage et tilsvarende verktøy en gang til. Det er selvsagt mulig å bruke deler av programmet, eventuelt bruke det som inspirasjon, men som helhet er det ikke anbefalt å videreutvikles.

6.8 Kognitiv bias

«Kognitive bias fører til en tilbøyelighet til å tolke vekk informasjon som er i strid med ens eksisterende virkelighetsoppfattelse i stedet for å endre mening.» (Wikipedia, 2019).

Dersom en leser forord samt bakgrunnen for oppgaven er det ikke å skjule at jeg selv er en forkjemper for elektronisk undervisning. Det skal ikke utelukkes at jeg gjennom arbeidet med denne oppgaven ubevisst har favorisert opplysninger og meninger som favoriserer elektronisk undervisning. Denne oppgaven har i all hovedsak tatt for seg fordeler og ulemper med elektronisk undervisning, den har ikke isolert sett på fordelene ved tradisjonell undervisning. Ut i fra kun denne oppgaven vil det være vanskelig å trekke faglige konklusjoner om elektroniske løsninger kan erstatte tradisjonell undervisning, det må sees på i et bredere perspektiv, spørreundersøkelsen ble kun gjennomført av syv elever. En spørreundersøkelse, med tilhørende statistiske resultater i større omfang vil bedre kunne besvare del to av problemstillingen min.

7 Avslutning

I denne oppgaven har jeg forsøkt å besvare problemstillingen *Kan jeg programmere et verktøy undervisere kan benytte som erstatning for tradisjonell undervisning? Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?*

Det har blitt utviklet et program som kan erstatte den tradisjonelle undervisningen, dette programmet har fremdeles mangler og bør ikke erstatte den tradisjonelle undervisningen ... enda.

Den tradisjonelle undervisningen kan til en viss grad bli erstattet av elektroniske løsninger, egne resultater og tidligere forskning tyder på at den største utfordringen ved elektronisk undervisning er få til interaktivitet og interaktivitet med underviseren.

Konklusjonene kommet frem til i denne oppgaven stemmer god overens med en annen undersøkelse som konkluderer med følgende:

«the majority of students do not want the full integration of Information Technology into the study process, meaning that the traditional learning methods combined with IT are preferred.» (Tušelj, Majcen, Mervar, Stepankina, & Čater, 2016)

Det er enda ikke tillit til opplegg bestående av kun elektroniske løsninger, en blanding av tradisjonell og elektronisk undervisning virker å være den ønskede undervisningsformen.

Dersom det skal utvikles et produkt innenfor e-læring bør det ha hovedvekt på å implementere interaktivitet, samt interaktivitet med underviser.

Kan jeg programmere et verktøy undervisere kan benytte som erstatning for tradisjonell undervisning?

Ja, til en viss grad.

Kan den tradisjonelle undervisningen erstattes av elektroniske løsninger?

Nei, ikke i dag, men kanskje i fremtiden.

8 Referanser

- El- Seoud, M., Taj-Eddin, Seddiek, El-Khouly, & Nosseir. (2014). E-Learning and Students' Motivation: A Reasearch Study on the Effect of E-Learning on Higher Education.
- Fykse, T. (2003). *Utvikling av elæring ved Sjøkrigsskolen og spesifikasjon av forklaringsmodul.*
- McFarland, A. (2022, Juli 8). *unite.ai*. Hentet fra <https://www.unite.ai/10-best-python-libraries-for-gui/>
- Tamm, S. (2022, Januar 7). *e-student.org*. Hentet fra <https://e-student.org/disadvantages-of-e-learning/#8-e-learning-is-limited-to-certain-disciplines>
- Tušelj, M., Majcen, L., Mervar, L., Stepankina, T., & Čater, B. (2016). E-Learning in Higher Education: Focus Groups and Survey Among Students in Central Europe. *i-manager's Journal of Educational Technology vol 13 no.2.*
- U.S Department of Education. (2009). *Evaluation of Evidence-Based Practices in Online Learning.*
- Wikipedia. (2019, Januar 5). *Wikipedia.org*. Hentet fra https://no.wikipedia.org/wiki/Kognitiv_bias
- Wikipedia. (2022, Desember 10). *wikipedia.org*. Hentet fra https://en.wikipedia.org/wiki/Object-oriented_programming#Objects_and_classes
- Wikipedia. (2022, November 5). *Wikipedia.org*. Hentet fra <https://en.wikipedia.org/wiki/Tkinter>
- Zheng, M., Bender, D., & Lyon, C. (2021). Online learning during COVID-19 produced equivalent or better student course performance as compared with pre-pandemic: empirical evidence from a school-wide comparative study. *BMC Medical Education.*

Vedlegg

VEDLEGG A: Spørreundersøkelse

VEDLEGG B: Kode

VEDLEGG C: Lisens (MIT LICENSE og ZERO-CLAUSE BSD LICENSE)

VEDLEGG D: Manual læringsverktøy

VEDLEGG E: Tillatelse til å innhente opplysninger i og om Forsvaret til forskningsmål

VEDLEGG F: Samtykkeerklæring

VEDLEGG G: Fykse, Forklaringsmodul

VEDLEGG H: Nedlastningslenke til manualen som kan brukes i programmet:

<https://1drv.ms/u/s!Ag9psFhLwIy34mzeGFQS1jI-aAuN?e=HFbRKc>

(Denne kan være utgått hvis du er fra fremtiden)

VEDLEGG I: Nedlastningslenke til programmet:

<https://1drv.ms/u/s!Ag9psFhLwIy34m1VN1ISCOOELHyZ?e=kfYVEf>

(Denne kan være utgått hvis du er fra fremtiden)