



INSTITUTT FOR DATATEKNOLOGI OG INFORMATIKK

IDATT1004 - TEAMBASERT SAMHANDLING

GRUPPE 21

LEGO prosjektoppgave

Authors:

Johannes Aamot-Skeidsvoll
William Laukvik Holtsdal
Vetle Nilsen
Kristian Ask Selmer
Harald Theodor Helland Velde

1 Forord

Hensikten med prosjektet er å utforske ulike aspekter av team-arbeid med en konkret oppgave. Oppgaven vi har tatt for oss er å lage et system som kan detektere farger på LEGO-brikker, og deretter føre brikkene inn i sine respektive båser og muligheten til å hente de lagrede brikkene ut igjen.

Oppgaven har lært oss basiskunnskap om prosjektarbeid som arbeidsform, herunder etablering, organisering, gjennomføring og rapportering. Vi har lært ulike prosesser for kreativitet og nyttenking i prosjektoppgaven. Sammen har vi klart å identifisere, planlegge og gjennomføre et lite utviklingsprosjekt i samarbeid. I løpet av oppgaven har vi lært hvordan vi benytter digitale samhandlingsplattformer, faglige verktøy, teknikker og terminologi for prosjektsamarbeidet.¹

Ettersom vi brukte smidige metoder startet vi med en idemyldringsfase og lagde en kort beskrivelse av minimumsforventningene til produktet (MVP) før vi startet å lage sorteringssystemet. Deretter lagde vi flere ulike prototyper av systemet i LEGO, men på grunn av stor mangel av nødvendige deler, er størsteparten av det ferdige produktet blitt 3D-printet med overskuddsmateriell. Kodingen skjedde parallelt med byggingen siden vi måtte teste de ulike prototypene og vi har dokumentert prosessen kontinuerlig underveis.

Vi valgte sorteringssystemet etter å vurdert mange ulike idéer, alt fra chat-bot med morsekode til kanoner til håndskriftsforfalsker. Likevel ble konklusjonen til at sorteringsystemet er en av få oppgaver som oppfylte FNs bærekraftsmål² og vi antok videre at denne oppgaven var av riktig kompleksitet i forhold til tid.

Vi ønsker til slutt å rette en stor takk til studieassistentene våre for deres uvurderlige hjelp og oppfølging. Vi vil især gi en stor takk til studieassistenten som har fulgt oss gjennom hele prosjektet, Shiza Ahmad, og hennes standhaftighet, samt villighet for å gi oss særdeles god veiledning i møte av våres besvær underveis i vårt prosjektarbeid. Hennes engasjement og tålmodighet har vært avgjørende for at vi har kunnet overkomme utfordringene vi har møtt, og vi er svært takknemlige for hennes innsats og støtte.

Harald Theodor Helland Velde

Kristian Ask Selmer



Vetle Nilsen

William Laukvik Holtsdalén



Johannes Aamot-Skeidsvoll



Figur 1: Signaturer

¹NTNU (2024, 6. november). Emne - Teambasert samhandling. NTNU. <https://www.ntnu.no/studier/emner/IDATT1004#learning-goal-toggler>

²Forente Nasjoner (2024, 2. februar). FNs bærekraftsmål. FN. <https://fn.no/om-fn/fns-baerekraftsmaal>

2 Sammendrag

Denne rapporten beskriver utviklingen av et automatisert sorteringsystem for LEGO-klosser, designet for å kategorisere klossene basert på farge. Systemet består av en sensor som skanner et lite område der klossene plasseres, og en robotstyrt kran som beveger seg langs to horisontale akser og én vertikal akse for å plassere klossene i dedikerte lagringsbåser. Prosjektet fokuserte både på teknisk utvikling og teamarbeid. Arbeidet ble utført ved bruk av smidige metoder og digitale verktøy som GitLab og Overleaf for versjonskontroll og rapportskriving. Til tross for utfordringer med utstyr og tekniske begrensninger, ble prosjektets mål oppfylt, og systemet fungerte som planlagt. Videre analyserer rapporten bærekraftsaspekter ved systemet, inkludert hvordan det kan bidra til økt effektivitet og sikkerhet i industrielle lagermiljøer, i tråd med FNs bærekraftsmål.

3 Abstract

This report presents the development of an automated sorting system for LEGO bricks, designed to categorize the bricks by color. The system consists of a sensor that scans a small area where the bricks are placed and a robot-controlled crane operating on two horizontal axes and one vertical axis to position the bricks into designated storage compartments. The project emphasized both technical development and teamwork, and agile methods and digital tools, such as GitLab for version control and Overleaf for report writing, were utilized throughout the project. Despite challenges with equipment and technical limitations, the project goals were achieved, and the system performed as expected. Additionally, the report discusses the system's sustainability aspects, including how it may contribute to increased efficiency and safety in industrial storage environments, aligning with the United Nations Sustainable Development Goals.

Innhold

1 Forord	i
2 Sammendrag	ii
3 Abstract	iii
Figurer	v
Tabeller	vi
4 Introduksjon	1
5 Presentasjon av Teamets medlemmer	1
5.1 Johannes Aamot-Skeidsvoll	1
5.2 William Laukvik Holtsdal	2
5.3 Vetle Nilsen	2
5.4 Kristian Ask Selmer	3
5.5 Harald Theodor Helland Velde	3
6 Problembeskrivelse	4
7 Resultater	5
7.1 Sammendrag av prosessen	5
7.2 Problemer underveis	5
7.3 Administrative resultater	6
7.4 Konstruksjonsresultater	6
7.5 Forklaring av kildekode	7
7.5.1 Main	8
7.5.2 Robot	8
7.5.3 Meny	8
7.5.4 Robot kontroller	8
7.6 Arbeidsfordeling	8
7.7 Metode	9
8 Diskusjon	10
9 Konklusjon	10
10 Bærekraft og samfunnspåvirkning	11

10.1	Bærekraftighet og samfunnsnytte	11
10.2	Etikk	13
	Referanser	14
	Appendix	15
A	Prosjektplan	15
B	Arbeidskontrakt	22
C	Kjernekvadranten	27
D	Møteinkalling veiledningsmøte 1	32
E	Møteinkalling veiledningsmøte 2	34
F	Møtereferat veiledningsmøte 1	36
G	Møtereferat veiledningsmøte 2	39
H	Timelister	42
I	Tekniske illustrasjoner	55
J	Kode listing	58
K	Presentasjon av prosjektet	64

Figurer

1	Signaturer	i
2	Teamets medlemmer: f.v. Kristian, Johannes, Harald, William og Vetle	1
3	AutoStore, et eksempel på automatisert lagersystem	5
4	Full modell, med markerte 3D-printede deler og beskrivelse.	6
5	Tverrsnitt gjennom X-aksen	7
6	Tverrsnitt gjennom Z-aksen	7
7	Gantt diagram	9
8	FNs bærekraftmål til 2030 består av 17 mål og 169 delmål.	12
9	Hele sorteringsroboten	55
10	Tverrsnitt av sensorfeste	55
11	Tverrsnitt gjennom x-aksen	56

12	Tverrsnitt gjennom y-aksen	56
13	Tverrsnitt gjennom z-aksen	57

Tabeller

1	Ukentlige aktiviteter, grovt beskrevet	11
---	--	----

4 Introduksjon

I denne rapporten beskriver gruppen utviklingen av et sorteringssystem som er designet for å identifisere og sortere LEGO-klosser basert på farge. Prosjektet er gjennomført som en del av emnet Teambasert Samhandling, der målet har vært å anvende og utvikle ferdigheter innen både teknisk design, utvikling og teamarbeid. Oppgaven har gitt innsikt i hvordan automatiserte systemer kan resultere i effektivitet og nøyaktighet i sorteringsprosesser. Videre har gruppen utforsket hvordan samarbeid, prosjektstyring og tekniske verktøy, som versjonskontroll og smidige arbeidsmetoder, kan optimalisere teamprosesser. Denne rapporten vil gjennomgå systemets oppbygning, beskrive utfordringene teamet møtte underveis, og drøfte erfaringer som kan anvendes i fremtidige prosjekter.

5 Presentasjon av Teamets medlemmer



Figur 2: Teamets medlemmer: f.v. Kristian, Johannes, Harald, William og Vetle

5.1 Johannes Aamot-Skeidsvoll

Johannes er en 18 år gammel gutt fra Bergen som liker kaffe, matlaging, programmering og musikk. Han har tidligere jobbet i hotellbransjen og tar nå en bachelor i dataingeniør på NTNU. Gjennom sine studier utvikler han ferdigheter i programmeringsspråk som Python og Java og viser en sterk interesse for utvikling av applikasjoner med robust arkitektur og modularitet.

Han startet å eksperimentere med programmering da han var 13 år gammel, og tidligere har han vært med på Talentsenter i Realfagsprogrammet der han har jobbet med mange ulike prosjekter innen for eksempel rakettfysikk, dronebygging og utvikling, astronomi, gener og mer. Han har også vært med å bygge og programmere en forskningsbøye som ble sendt til nordpolen for å samle inn data til forskere verden rundt. Johannes har også i to år vært sjef i lyd og lys for Bergen Katedralskoles revy, der han har forhandlet kontrakter, settet opp lyd- og lysutstyr, gjort all lydmiksing og ledet gruppen.

Johannes er engasjert i ulike prosjekter innen programvareutvikling, hvor han blant annet har erfaring med å designe grensesnitt i Kotlin og håndtere databaseintegrasjon via Azure SQL. I tillegg arbeider han med utvikling av verktøy som kan brukes i hotellbransjen, basert på erfaringene fra tidligere jobb, samt å være med i en startup som skal automatisere produksjon av mikrogrønt.

I tillegg har han laget en årlig dataanalyserapport av Balestrand Hotell som finner trender, nyttig innsikt og finner punkter på forbedring.

5.2 William Laukvik Holtsdalen

William Laukvik Holtsdalen er en 20 år gammel student, som kommer fra Horten i Vestfold. Han bor for tiden i Trondheim hvor han tar en bachelorgrad på NTNU for å bli dataingeniør, med fokus på systemutvikling. William har en stor interesse for teknologi, programmering og hvordan datamaskiner fungerer. Ytterligere er han svært interessert i alt som har motor og er en ivrig tilhenger av motorsport, sim-racing og luftfart.

William har erfaring med elektrofag og automatisering fra videregående opplæring. Han fikk innsikt i industrielle prosesser og teamarbeid samtidig som han opparbeidet seg erfaring med automatiserte kontrollsystemer gjennom både teoretisk og praktisk arbeid. Denne erfaring ga ham innsikt i hvordan automatisering kan effektivisere og forbedre industrielle produksjonsprosesser og viktigheten av grundige og pålitelige sikkerhetsprosedyrer.

Videre har han arbeidet med flere personlige prosjekter, som har gitt ham grunnleggende erfaring med programmeringsspråk og utviklingsverktøy som Python, C, Flutter, Django og SQL. Han lærte Python og C gjennom å ta CS50, et introkurs til *Computer Science* fra Harvard University som han fullførte på nett. Flutter utforsket han på egen hånd ettersom han ønsket å lære seg app-utvikling da han syntes idéen om å lage apper som tilfredsstilte hans egne behov virket attraktiv. Videre lærte han grunnleggende bruk av SQL i kombinasjon med Django for å kunne lage web-applikasjoner med solid database-backend.

William har også omfattende erfaring med 3D-printing og elektronikk. Fra våren 2022 til sommeren 2023 drev han et enkeltpersonforetak som spesialiserte seg på produksjon og salg av sim-racing ratt til kunder over hele verden. Dette ga ham verdifull erfaring med markedsføring, salg og regnskap, samt tekniske ferdigheter som design og produksjon av kretskort.

Målet til William er å jobbe som utvikler, da han tiltrekkes av kombinasjonen av kreativitet, struktur og problemløsning. Drømmen hans er å starte et eget firma eller arbeide med softwareutvikling for et Formel 1 team, som for eksempel McLaren F1. Dette reflekterer hans lidenskap for teknologi og motorsport, og ønsket om å kombinere begge interessefeltene i en spennende karriere. For å få en smakebit av dette har han et ønske om å være med i Revolve NTNU i tredje og fjerde semester av studiet.

5.3 Vetle Nilsen

Vetle Nilsen er en 23 år gammel student fra Lillestrøm. Vetle er en førsteårsstudent Dataingeniør på NTNU Trondheim. Interessen for data og programmering kom på videregående da Vetle valgte valgfagene Programmering og modellering X, Informasjonsteknologi 1 og Teknologi og forskninglære 1-2. Disse programfagene ga Vetle inspirasjon til å forfølge en karriere innenfor programmering og ingeniørfag. Vetle har tidligere studert på NTNU Ålesund hvor han studerte ett semester på Automatisering og intelligente systemer (AIS). AIS bekreftet for Vetle at det var en karriere som softwareutvikler som skulle forfølges. Dette førte til at Vetle byttet studie fra AIS til Dataingeniør. Vetle har opp gjennom skole og studie perioden bygget opp en grunnleggende forståelse for programmering og forskjellige programmeringsspråk.

Vetle sin arbeidserfaring har kommet fra diverse sommerjobber fra da han var 16 år gammel. Han har jobbet som logistikkmedarbeider, butikkmedarbeider og flyarbeider på Container Consulting, Plantasjen og SAS. Som butikkmedarbeider har Vetle fått erfaring med kundeservice og selger. Som lagermedarbeider har Vetle fått erfaring innenfor lagerstyring, lagerkontroll, varemottak, vareplukking, pakking og fraktforberedelser. Som flyarbeider har Vetle fått erfaring innenfor teamarbeid og sikkerhet.

Vetle har jobbet med diverse prosjekter hvor han har tatt i bruk mikrokontrollere som arduino.

Prosjektene har vært forskjellige ting fra linjefølger roboter til svevestøv-sensor.

Hobbyene til Vetle er buldring, surfing og dataspill. Vetle har buldret som hobby i åtte år og surfet rundt om i verden året 2019-2020.

5.4 Kristian Ask Selmer

Kristian Ask Selmer er en 19 år gammel student fra Bærum, som nå er førsteårsstudent i ingeniørfag - data på NTNU i Trondheim. Interessen for teknologi begynte tidlig, påvirket av en oppvekst hvor faren, en utdannet dataingeniør, jobbet i IT-bransjen. Fra ung alder ble Kristian eksponert for både tekniske begreper og jobblivet som utvikler, hvilket ga ham en nyskjerrighet for softwareutvikling. Denne nyskjerrigheten ble til konkret interesse allerede på ungdomsskolen, hvor han i 8. klasse valgte programmering som valgfag. Siden den gang har han bygget opp kompetanse innen flere programmeringsspråk og rammeverk, og utviklet en bred forståelse for hvordan teknologi kan brukes til å løse praktiske utfordringer.

Kristians arbeidserfaring kommer fra en fireårig periode hos startupen Ewave Solutions, et selskap som spesialiserte seg på salg av strømsparingssystemer til private kunder. Gjennom sin tid i Ewave Solutions fikk Kristian en variert arbeidsdag og praktisk erfaring med hvordan teknologi kan implementeres. Han startet med grunnleggende oppgaver som pakking og kvalitetskontroll av produkter, men fikk raskt nye utfordringer og mer ansvar. Etter hvert som hans teknologiske ferdigheter vokste, ble han involvert i interne prosjekter hvor han tok del i utviklingen av systemer for automatisering. Dette prosjektet ga Kristian verdifull erfaring med automatisering og viste hvordan programmering kan optimalisere forretningsprosesser og redusere kostnader.

I tillegg til tekniske ferdigheter lærte Kristian mye om samarbeid og kommunikasjon i et tverrfaglig miljø. Han fikk erfaring med å tilpasse seg kolleger med ulik faglig bakgrunn, noe som gjorde ham fleksibel og løsningsorientert. Gjennom ulike teamprosjekter og veiledning av nye teammedlemmer fikk han innsikt i hvordan god kommunikasjon kan bidra til et produktivt arbeidsmiljø.

Med en sterk interesse for softwareutvikling og en genuin nyskjerrighet for teknologi, ser Kristian fram til å utvide sine ferdigheter ytterligere både gjennom sine studier og egne prosjekter.

5.5 Harald Theodor Helland Velde

Harald Th. H. Velde er per dags dato, en førsteårsstudent på dataingeniør-linjen til NTNU.

Tidligere har han vært student på Greåker VGS sin realfagslinje (SCIENCE), der han var på den eneste gruppen som fikk jobbe med firmaet NxTech som gir løsninger til gründere. Med NxTech fikk gruppen hans jobbet med å lage en enhet som skulle på teater og registrere bevegelsene til skuespillerne som gir diverse effekter som å endre lyd og lys.

Etter sin tid på videregående var Harald logistikksoldat for Polar Bear VI, innsatsstyrken i Oslofjord Heimevernsdistrikt hvor han fikk kjenne på en rekke mentale og fysiske utfordringer i sitt ellers komfortable liv.

Fra Harald var liten har han hatt en interesse for realfag. Både faren og moren hans er utdannet ingeniører (hhv. bygg- og branningeniør) som har sammen med Harald hatt det mye gøy med matematisk orienterte temaer, og de har lært han noen artige teknikker for spesielle matematiske uttrykk som å løse kvadratuttrykk på en spesiell form. En av favoritt-bøkene hans er "What if?" av Randall Munroe³ som regner med fysikk på absurde problemstillinger som hva skjer når et glass blir bokstavelig halvtomt (dersom den nedre delen fordufter blir glasset knust ganske langt mot bunn og overdelen av glassen vil fly et stykke oppover).

Da Harald lærte seg å lage veldig grunnleggende nettsider, ble han raskt solgt over til teknologiens fantastiske verden, der i senere tid HTML/CSS/JS har gått i glemmeboken til fordel for Python

³Munroe, R. (2015) What if?: Serious scientific answers to absurd hypothetical questions. London: John Murray.

og Java. Langtidsmålet hans er å ha en stor kunnskapsbase på tvers av mange kodespråk, også en del litt mer unike som Fortran, Julia og Perl.

Hobbyene til Harald består mye av speed-cubing (løse Rubik's kube så raskt som mulig) og gaming, men også programmering og å spille håndball. Siden han fikk en forkjærighet for håndball, ble han også en nivå 2 dommer (t.o.m. J/G-15), hvilket som også ble en liten inntektskilde på videregående ved siden av å jobbe på ARK bokhandel som butikkassistent.

6 Problembeskrivelse

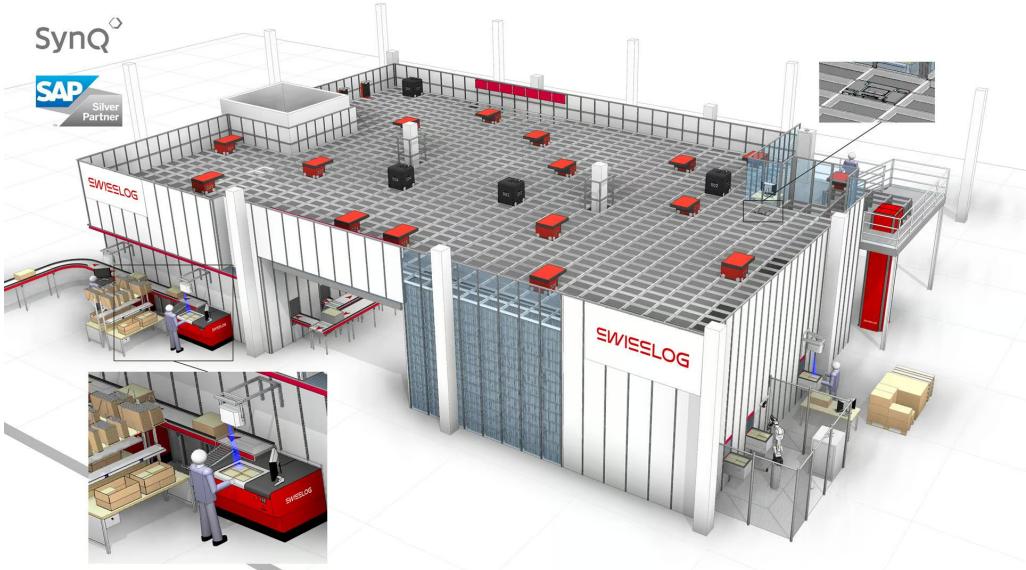
Innledningsvis i prosjektet satte gruppens medlemmer seg sammen i plenum for en grundig diskusjon om prosjektets formål og målsetninger. Gruppen konkluderte med at den høyeste prioriteten under prosjektarbeidet bør være å sikre at alle deltar aktivt i læringsprosessen. Dette inkluderer å forstå hvordan man effektivt arbeider i team, hvordan man kommuniserer tydelig og konstruktivt, og hvordan man kan dra nytte av hverandres styrker og ferdigheter for å oppnå best mulig resultat. Det ble enighet i gruppen om at denne tilnærmingen ikke bare vil bidra til et vellykket prosjekt, men også til å utvikle verdifulle samarbeidsevner og et godt grunnlag for fremtidige situasjoner med teamarbeid. I tillegg ble det lagt vekt på viktigheten av å skape et positivt og støttende arbeidsmiljø hvor alle føler seg hørt og verdsatt.

Prosjektet som gruppen valgte var å designe og bygge en robot som sorterer de fargede LEGO-klossene automatisk. Systemet består av en 3D-printet ramme med et lasteområde, der en 2 dimensjonal x-y portalkran (lik som laserkuttere, CNC-maskiner, 3D-printere og liknende) henter klossen og plasserer den i en av seks tilgjengelige båser basert på farge. Portalkranen har et lastebur, montert på en motorisert z-akse, som omfavner LEGO-brikke og muliggjør for flytting av dem langs x- og y-aksene. Antall båser er valgt med hensyn til de tilgjengelige klossene og den nødvendige sorteringen, der hver bås representerer en unik farge. Fargene som LEGO-klossene sorteres etter er rød, blå, grønn, gul, lysegrønn og oransje. Klosser med andre farger enn dette vil ikke bli godkjent av systemet. Dette prosjektet ble valgt fordi det simulerer prinsippene for automatisert sortering, som også finnes i lagersystemer på større skala, slik som SwissLogs AutoStore-løsning (SwissLog 2024). Ved å bruke automatisering kan systemet utvides for å håndtere større objekter og bidra til økt effektivitet og organisering i et lagermiljø.

Den opprinnelige planen for prosjektet besto av å sende de fargede LEGO-klossene inn i sorteringssystemet ved bruk av et motorisert belte. Dette skulle fungere som et transportbånd som førte LEGO-klossene fra utsiden av lageret og inn til lasteområdet. Herfra kunne roboten ta tak i hver enkelt brikke og plassere dem på sine tilhørende plasseringer i lageret. Gruppen bestemte seg etter grundig evaluering for å skrote denne idéen ettersom medlemmene innså at de fysiske ressursene ikke var tilstrekkelige, med tanke på motorer og kabler med tilstrekkelig lengde. Det var også usikkerhet rundt hvordan den tekniske implementasjonen av dette skulle fungere med tanke på avlasting av LEGO kloss fra transportbånd til område for fargegenkjenning, og som resultat av dette bestemte gruppen seg for å heller benytte et statisk lasteområde hvor brikkene plasseres rett foran fargesensoren, i riktig posisjon for å bli hentet av kranen umiddelbart.

Portalkranens x- og y-akse var opprinnelig ment til å bevege seg med tannhjul på tannstenger av LEGO, men gruppen innså omsider at den alvorlige mangelen på tannstang-brikker førte til at det ikke ville være mulig å dekke den nødvendige distansen på alle tre aksene. Resultatet av dette var at gruppen tok beslutningen om å ty til 3D-printing for å lage flere LEGO-brikker. Denne idéen ble raskt erstattet av planen om å 3D-printe en komplett struktur med tenner til tannhjul integrert i rammeverket, for å spare tid og plass, samt for å forbedre strukturens stivhet.

Den opprinnelige idéen for robotens z-akse var å bruke en motor med fiskesnøre på en spole til å heise lasteburet opp og ned, samt stabilisatorstag for å sikre bevegelse utelukkende langs z-aksen. Planen for denne mekanismen ble så revidert til å benytte en motor med tannhjul og tannstang integrert i lasteburet til å heve og senke det, for å sikre bedre presisjon og pålitelighet, samt for å spare plass på en ellers kompakt robot.



Figur 3: AutoStore, et eksempel på automatisert lagersystem

7 Resultater

7.1 Sammendrag av prosessen

Prosessen begynte med en idémyldringsfase, hvor teamet utforsket flere mulige løsninger for et sorteringssystem for LEGO-klosser. Etter en grundig vurdering ble idén om å bygge en robot med evne til å sortere klosser basert på farge valgt som prosjektets hovedmål. Etter valg av konsept laget teamet en plan for arbeidet som inkluderte både design og funksjoner til roboten, samt tanker rundt konstruksjon og programmering av roboten. Det ble også laget en røff plan for arbeidsfordelingen hvor oppgavene ble fordelt slik at hvert medlem fikk ansvar for ulike deler av prosjektet. Samtidig ble det lagt opp til faste møter med studentassisterenter for å overvåke fremdriften og sikre god koordinering av arbeidet. Siden gruppen besto av flere medlemmer enn det var unike arbeidsoppgaver, ble visse oppgaver delt for å fremme samarbeid og kvalitetssikring underveis i prosjektet. Dette førte til en naturlig overlapping i arbeidsoppgaver, som førte at gruppen måtte kommunisere effektivt, for å sikre god koordinering av arbeidsoppgaver og fremdrift i prosjektet.

7.2 Problemer underveis

Underveis i prosjektet støtte gruppen på flere utfordringer:

- Utdatert micro-python versjon på EV3 kontrolleren.
- Begrenset tilgang til nødvendige komponenter som LEGO-klosser, motorer og lange kabler.
- Vekt og stabilitet av maskineriet som krevde endringer i designet for å sikre pålitelig drift.
- Maskinvare problemer med EV3 kontrolleren.
- Vanskelig å lage enhetstester til et fysisk produkt

En teknisk begrensning var at micro-python versjonen som EV3 kontrolleren kjører er basert på python 3.4, og dermed var det ikke mulig å bruke nyttige metoder som enums, dataklasser og f-strenger i robotens programkode. Det var heller ikke mulig å legge til feilhåndtering som en vanligvis ville gjort med å bruke `raise` kodeordet.

En annen utfordring som oppsto relativt tidlig i dette prosjektet var at det var mangel på ressurser i form av LEGO-klosser. Prosjektet var stort og krevde mange LEGO-klosser for å kunne bygge opp rammen. Det var ikke nok klosser tilgjengelig til å kunne utføre prosjektet. Heldigvis hadde William Laukvik Holtsdalén både tilgang til 3D-printer og kunnskapen til å produsere rammeverket til prosjektet. Med et 3D-printet rammeverk reduserte dette vekten og økte stivheten til strukturen, som medførte til bedre stabilitet og vektfordeling totalt sett.

Tekniske problemer som har oppstått underveis har vært problemer med port B på EV3 kontrolleren. Under testing av tidlig programkode oppdaget gruppen at porten ikke fungerte. Når en motor var tilkoblet port B klarte ikke EV3 kontrolleren å kommunisere med motoren, noe som førte til I/O feil i programmet. Videre var det tekniske problemer med fargesensorene, i den forstand at verdiene de forskjellige fargesensorene viste var forskjellige fra hverandre. En konsekvens av dette var unødvendig tidsbruk til å sammenlikne sensorer for å identifisere den med de mest pålitelige avlesningene.

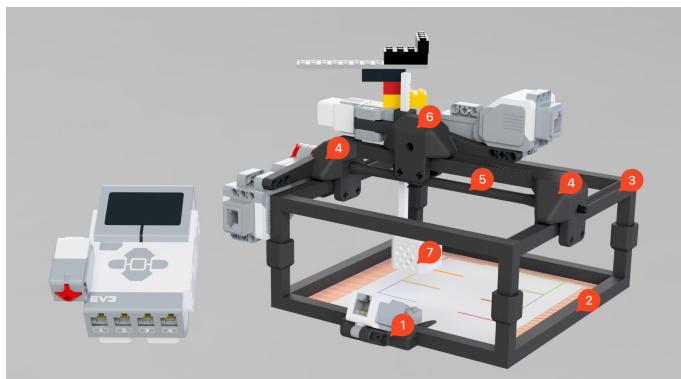
Videre har det vært vanskelig å teste programmet, selv om det er skrevet enhetstester for mange av funksjonene i koden. Utfordringen ligger i det å jobbe med et fysisk system, og det er krevende å lage tester som dekker både programlogikk og hvordan systemet oppfører seg i praksis. Utfordringen har vært å balansere automatiserte tester med behovet for praktisk validering, noe som har ført til mer tidkrevende iterasjoner og manuelle tester.

7.3 Administrative resultater

Gjennom prosjektetarbeidets forløp møtte teamet jevnlig med eksterne veileder for en total av fem møter, henholdsvis to veiledningsmøter med studieassistent, to *stand-up* møter med emneansvarlig og ett *stand-up* møte med studieassistent. Disse møtene sikret god koordinering av arbeid, samt at de bidro til å ha åpne og felles diskusjoner om gruppens fremgang i prosjektet.

Gruppen har til sammen brukt omtrent 80 timer på konstruksjon, hvorav omtrent 30 timer ble brukt på 3D-modellering, 20 timer på design og resten på prototyping av roboten. Videre brukte gruppen 75 timer på dokumentasjon, som inkluderer innledende prosjektplan, møteinnkallinger og referater, samt prosjektrapporten. 45 timer ble brukt på programmering av roboten, 25 timer på administrasjon og 17 timer på diverse andre aktiviter, som teamutvikling og liknende.

7.4 Konstruksjonsresultater

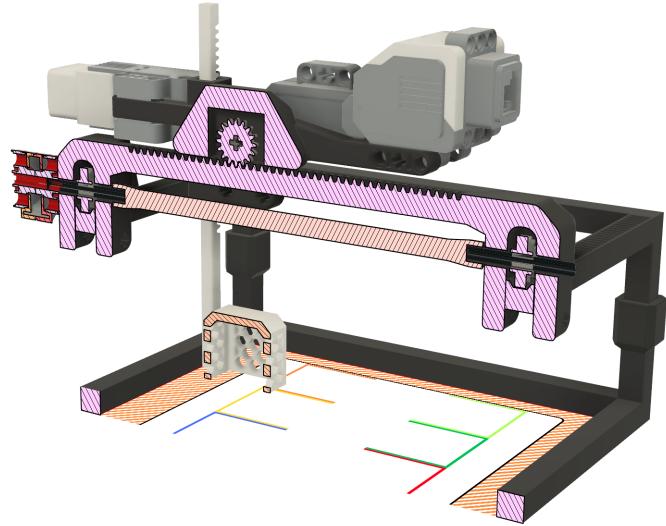


1. Fargesensor holder
2. Bunnside av rammen
3. Topp side av rammen
4. Y-akse motorholder, bjelke og tannhjul-hus
5. Y-akse drivaksel
6. X-akse motorholder, Z-akse motorholder og tannhjul-hus
7. Lastebur

Figur 4: Full modell, med markerte 3D-printede deler og beskrivelse.

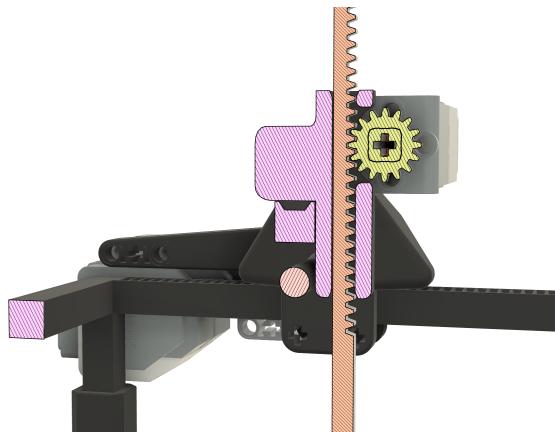
Figur 4 viser en 3D-modell av prosjektet gruppen har laget. Modellen består av syv forskjellige 3D-printede deler som bygger opp rammen til prosjektet.

Figur 5 viser en utsnitt fra X-aksen. Her kan man se innsiden av tannhjul-husene fra innsiden, både forfra og bakfra. Man får også sett hvordan drivakselen mellom tannhjul-husene på Y-aksener ble bygd opp. Denne akselen ble inkludert for å øke stabiliteten under kjøring. Tannhjul-husene ble



Figur 5: Tverrsnitt gjennom X-aksen

konstruert slik man ser på figur 5 av 2 grunner. Grunn nummer 1 er for en mer stabil kjøring. Med en vid base reduserer det risting uønskede bevegelser under akselerasjoner fra motorene. Grunn nummer 2 er for sikkerhet. Huset dekker over hele tannhjulet. Dette er implementert slik at det ikke skal være mulig å skade seg ved å få fingre klemt og for at ikke noe skal kunne komme inn i tannhjule og på føre skade på tannhjulet.



Figur 6: Tverrsnitt gjennom Z-aksen

Figur 6 viser et tverrsnitt gjennom Z-aksen. Her får man sett hvordan heisefunksjonen for lasteburet fungerer.

7.5 Forklaring av kildekode

Prosjektets kildekode er strukturert rundt flere ulike moduler som samarbeider for å kontrollere robotens funksjonalitet på en effektiv og oversiktlig måte. Dette gjør det enklere å feilsøke, videreutvikle og vedlikeholde koden. Hovedfunksjonene inkluderer automatisk kalibrering, automatisk fargesortering, automatisk uthenting fra lager via en brukerstyrt meny og manuell styring av roboten. Nedenfor er en oversikt over de ulike delene av kildekoden og deres ansvar.

7.5.1 Main

Denne filen, se kodelisting 1, fungerer som inngangspunktet for programmet. Den har som ansvar å initialisere en instans av klassen `RobotController`, se kodelisting 2, som styrer hovedflyten i programmet. Hovedfunksjonen `main()`, se kodelisting 1, kalles når programmet starter, og sørger for å starte roboten på korrekt måte.

7.5.2 Robot

Robot-klassen, se kodelisting 4, representerer robotens maskinvare og logikk. Blant annet håndterer den interaksjoner med robotens tre motorer og to sensorer, samt all bevegelseslogikk på EV3 kontrolleren. Noen av nøkkelfunksjonene denne klassen innehar er:

- **Manuell kontroll:** Muliggjør at brukeren kan styre robotens bevegelser langs X-, Y- og Z-aksene via knappetrykk på EV3 kontrollenhetens D-pad.
- **Automatisk lagring:** Definerer bevegelses-sekvens for lagring av LEGO-brikker til ulike posisjoner i lagersystemet basert på farger. Fargene har forhåndsdefinerte posisjoner i lagersystemet, og flyttes dit automatisk av roboten. Funksjonen skriver jevnlig ut hvilken farge som er oppdaget av fargesensoren.
- **Automatisk henting:** Definerer bevegelses-sekvens for automatisk henting av LEGO-brikker med spesifikke farger, basert på inndata fra en grafisk brukermeny.
- **Kalibrering:** Ettersom all posisjonsdata må ha et nullpunkt som referanse, er kalibrering en svært essensiell funksjon. Det fungerer ved å kjøre motorene i negativ retning helt til de møter motstand tilsvarende 40% av maksimalt dreiemoment. Dette utløser kode for å nullstille motorenes posisjonsdata og dermed sette et nullpunkt, også kalt origo.
- **Fargedekksjon:** Gjenkjenner farger ved hjelp av *RGB*-verdier fra fargesensoren på roboten. Dette gjøres matematisk ved å sjekke om avlesningene for rød, grønn og blå er innenfor en satt toleranse ut ifra fargenes referanse-verdier.

7.5.3 Meny

`MenuSystem`-klassen, se kodelisting 3, håndterer navigasjon i en grafisk brukermeny som lar brukeren hente ut LEGO-klosser fra lageret. Systemet bruker EV3 kontrollerens D-pad-knapper til å bla gjennom lagrede farger og vise tilhørende menyvalg på skjermen. Når en farge er valgt instruerer menyen roboten til å hente klossen fra riktig posisjon i lageret.

7.5.4 Robot kontroller

Denne filen, se kodelisting 2, inneholder `RobotController` klassen, som fungerer som en hovedkontroller for robotens drift. Den har som ansvar å koordinere robotens oppførsel og navigasjon mellom de tre ulike driftsmodusene: Manuell styring, automatisk lagring og automatisk uthenting.

Kontrolleren initialiserer robotens motorer, sensorer og menyer. Den overvåker også brukerinput for å bestemme hvilken modus som er aktiv, samtidig som den sørger for å oppdatere skjermen på EV3 kontrolleren med relevant informasjon for modusen.

7.6 Arbeidsfordeling

Arbeidsoppgaver er blitt jevnt fordelt der som regel to personer har samarbeidet på én ting, da gruppen er på 5 personer og det ikke er like mange oppgaver til enhver tid å takle. De kortsliktige oppgavene ble spesifisert muntlig i gruppen.

I starten av prosjektet var ikke arbeidsfordelingen spesifisert, men generelle ansvarsområder ble derimot fordelt i starten av semesteret. Kristian var møtereferent, Harald var ressursansvarlig, William var dokumentansvarlig, Johannes var møteansvarlig og Vetle var kvalitetskontrollør. Som møtereferent hadde Kristian i oppgave å skrive møtereferat for hvert møte vi hadde som gruppe. Som ressursansvarlig hadde Harald ansvaret for nøkkelen som tilhørte skapet til gruppe 21 med alt av ressurser gruppen trengte til å utføre prosjektet. Harald hadde også en oversikt over ressurser gruppen hadde og ressurser gruppen trengte å anskaffe seg i løpet av prosjektet. Som dokumentansvarlig hadde William ansvaret for å dokumentere prosessen av prosjektet, samt sørge for at møteinnkallinger og annen prosessdokumentasjon ble produsert og distribuert til fristene. Johannes sitt ansvar som møteansvarlig, var å sette opp møter for gruppen og lede disse. Vetles ansvar som kvalitetskontrollør var å gå over utkast til programkode, prototyper osv. for å sjekke at alt fulgte standarden gruppen var enige om. I uke 44 innførte gruppen en tydeligere og bedre spesifisert arbeidsfordeling. Tabell 1 viser resultatet av arbeidsfordelingen, samt en overordnet oversikt over hva hvert enkelt gruppemedlem har gjort på ukesbasis.

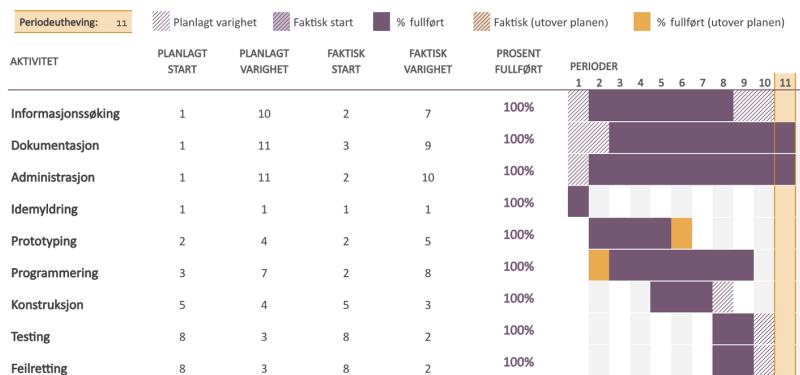
7.7 Metode

Teamet arbeidet etter smidige prinsipper. Dette innebar at teamet etablerte viktige systemer for iterativ testing og evaluering av prosjektets fremdrift. Arbeidsoppgaver ble fortløpende prioritert basert på deres innvirkning på produktet, sluttmålet og prosjektets tidsramme, slik at man kunne håndtere eventuelle endringer og uventede problemer på en effektiv og organisert måte.

Gruppen har også laget en teknisk designvalgbank i GitLab, der betydelige beslutninger for designvalg for prosjektet ble loggført. Dette har gjort det lettere å se tilbake på når vi valgte å gjøre hva, og hvorfor.

GitLab er et digitalt verktøy som ble anvendt flittig under prosjektarbeidet, for versjonskontroll og prosjektstyring, med støtte fra kommunikasjon på Discord. Gruppen har tatt i bruk branches i GitLab for å utvikle og implementere forskjellige ideer for koden. Automatisk DevOps var satt opp i GitLab for å automatisk kjøre tester og linting (kodeformattering) for at kvaliteten på koden skal være så konsekvent som mulig. Dette reduserer antall trivielle og oversettelsesfeil som ofte oppstår i programmering.

Planen ble i stor grad fulgt gjennom prosjektet, som det vises på grafikken under.



Figur 7: Gantt diagram

8 Diskusjon

Det digitale samarbeidet har gått meget bra, selv med utfordringen av å lære å ta i bruk versjonskontrollsystemet Git, GitLab og L^AT_EX-editoren Overleaf, som er det rapporten er skrevet i. Dette går også hånd i hånd med samarbeidet under timene der gruppen har sittet sammen. Oppmøtet fra hvert enkelt gruppemedlem har vært svært bra, til tross for sykdom, noe som har bidratt til at kommunikasjonen undervis i prosjektet har vært utmerket. Denne gode kommunikasjonen og høye oppmøte har vært en suksessfaktor i å oppnå målene gruppen satt for innledningsvis, da det har lagt til rette for et sterkt samarbeid der gruppen har kunnet drøfte ideer og derav løse problemer raskt og effektivt.

Valget om å heller 3D-printe rammeverket til roboten, enn å lage det av LEGO-brikker, viste seg å være en svært viktig avgjørelse for prosjektets progresjon. Konstruksjonen endte opp med å være svært stabil og fungerte godt til prosjektets formål. Denne erfaringen lærte gruppen å være ressurssterke og å tenke utenfor boksen.

Ettersom robotens kildekode ble utviklet med sikkerhet og presisjon i tankene, fungerte funksjonaliteten og sikkerhetsfunksjonene utmerket.

Rollefordeling innad i gruppen gikk smidig for seg i starten av prosjektet, hvor Johannes fikk rollen møteansvarlig, Kristian ble møtereferent, Harald ble ressursansvarlig, William ble dokumentansvarlig og Vetle var kvalitetskontrollør. Rollene har fungert fint, siden det ikke har oppstått noe konflikter eller misnøye over roller. Avgjørelser som har blitt tatt under prosjektet har blitt gjort i fellesskap med i form av avstemning.

Gruppen fikk satt opp et system når det gjelder kodebidrag ved bruk av flere forskjellige Branches. Det var gjort et poeng å skrive gode commit-meldinger på foreslatté endringer for å holde en logg som er lett å se igjennom. Endringer er foreslatt med issues og større mål er skrevet som milestones. Merge Request og issues er videre markert med labels for å kategorisere endringer. Det kunne imidlertid blitt gjort flere grundige Code Review. Likevel er dette et såpass lite og nytt prosjekt, at det ligger lite nytte i det.

Den automatiske testingen via GitLab sin automatisk DevOps fungerte bra, og fanget opp både syntaks- og formateringsfeil underveis i utviklingen. Dette var verdifullt og bidro til å forhindre slurvefeil underveis i arbeidet. I slutten av prosjektet oppsto det noen tekniske problemer der lagringsplassen for disse automatiske testene ble brukt opp, som gjorde det vanskelig å merge koden fra utviklings-branchen i Git inn i hoved-branchen der den 'produksjonsklare' koden skal være.

Felles rapportskriving i Overleaf med L^AT_EX har vært problemfritt. Noen av teamets medlemmer hadde tidligere erfaring med L^AT_EX, og andre hadde ikke det som har ført til at medlemmer har opplært hverandre, som videre har styrket teamets moral. Dette har i tillegg hjulpet med å forberede medlemmene til større samarbeidsprosjekter og oppgaver som skal skrives med L^AT_EX. Rapporten var i tillegg versjonskontrollert med Git og koblet til et GitHub-repo. Dersom noe skulle gå galt, fungerer det som et sikkerhetsnett og det gjør rapporten mer tilgjengelig.

For videre utvikling av produktet er det flere ting en kan se på og forbedre. For eksempel kan et kamera være festet i bur som kan merke om en kloss er feilplassert eller har blitt dultet vekk fra sin riktige posisjon, for å så rette opp i feilen. Videre kan man i stedet for å dra klossene langs bakken, heller plukke de opp i luften med en plattform- eller vakuumløsning. En annen mulig forbedring kan være å bruke QR-koder til å identifisere last istedenfor fargekoder. Det vil gjøre det mulig å sortere et større sortiment med varer og vil være mindre problematisk enn fargesensorer som er ofte sensitiv til lys og refleksjon.

9 Konklusjon

Gjennom utviklingen av dette prosjektet tilegnet gruppen seg verdifull innsikt i hvordan automatiserte systemer kan anvendes for å både effektivisere og forbedre diverse sorteringsprosesser.

Tabell 1: Ukentlige aktiviteter, grovt beskrevet

	Johannes	Kristian	Vetle	William	Harald
Uke 41	Idémylding og Informasjons-søking	Informasjons-søking	Informasjons-søking	Informasjons-søking	Informasjons-søking
Uke 42	Programmering	Informasjons-søking	Prototyping	Prototyping og dokumentasjon	Prototyping og rapportskriving
Uke 43	Rapportskriving og programmering	Rapportskriving og prototyping	Prototyping	Prototyping	Rapportskriving og prototyping
Uke 44	Rapportskriving og programmering	Prototyping og rapportskriving	Prototyping og testing	Prototyping og testing	Prototyping, rapportskriving
Uke 45	Rapportskriving og programmering	Testing	Rapportskriving	Prototyping, programmering og testing	Prototyping, og testing
Uke 46	Rapportskriving	Rapportskriving	Informasjons-søking, rapportskriving og presentasjon	Programmering og testing	Rapportskriving og dokumentasjon
Uke 47	Rapport- og presentasjons-skriving	Presentasjons-forberedelse	Presentasjon og rapportskrivign	Presentasjon og rapportskriving	Presentasjon og rapportskriving

Prosjektet vurderes som vellykket basert på resultatene tilknyttet tydelig rollefordeling og effektivt samarbeid, noe som bidro til effektiv problemløsning og fleksibilitet til å tilpasse løsninger underveis i prosjektet. I denne vurderingen vektes det også at gruppen oppnådde et velfungerende og godt testet produkt. Det konkluderes med at prosjektets opprinnelige mål, både med fokus på prosjekteoretiske aspekter og praktisk gjennomføring, ble oppfylt, og at teamets strukturerte samarbeid har vært avgjørende for prosjektets suksess.

Dette prosjektet kan anses som en modell for hvordan lignende løsninger kan implementeres i stor skala, eksempelvis i et stort lagerhus. I slike omgivelser vil det automatiserte sorteringsystemet øke effektiviteten ved å håndtere et større volum og redusere behovet for manuell arbeidskraft. Ved å forvandle prosjektet om til storskala, kan man også utforske muligheter for sanntids dataovervåkning og integrasjon med eksisterende programvare for å optimalisere driften.

For de som kunne tenke seg å lage noe liknende vil vi sterkt anbefale å 3D-printe løsningen over å bygge det i LEGO, siden å bygge med tilgjengelige LEGO-brikker resulterer ofte i overkant store konstruksjoner.

Hvis det er ønskelig å bygge videre på prosjektet, anbefales det å jobbe med oppskalering av systemet og se på muligheten til å sortere i høyden. Fargesensoren kan erstattes med en QR-kodeavleser, slik at man ikke lenger er avhengig av umøyaktig avlesning av lasten.

10 Bærekraft og samfunnspåvirking

10.1 Bærekraftighet og samfunnsnytte

I denne drøftingen av bærekraftighet og samfunnsnytte kan man se for seg at sorteringsystemet er implementert i storskala, som for eksempel en del av en lagerløsning for flytting av varer, kontra et LEGO-prosjekt der systemet flytter på LEGO-brikker. Ved å tenke på sorteringsystemet som en storskala løsning, kan man bedre analysere de potensielle effektene på bærekraft, arbeidsmiljø og likestilling. Denne tilnærmingen gjør det mulig å trekke linjer mellom prototypen og praktiske anvendelser i lagerhus, og dens innvirkning på bærekraftighet og samfunnsnytte.



Figur 8: FNs bærekraftmål til 2030 består av 17 mål og 169 delmål.

For det første vil vårt sorteringsystem redusere menneskefeil og øke den økonomiske veksten, noe som går i tråd med FNs åttende bærekraftsmål (FN 2024a). Menneskelige feil som feilplasserte og feilregistrerte varer, samt forsinkelser i prosessen vil senke. Man får en jevnere arbeidsflyt som øker effektiviteten, og derav øker man den økonomiske veksten. Automatiseringen gjør det også mulig å optimalisere lagringsplass, fordi varene blir sortert og plassert på en mer konsekvent og nøyaktig måte. Dette kan føre til at man får bedre utnyttelse av eksisterende lagerplass, og at man kan takle et høyere volum av varer, uten å måtte betale for et større lager.

Sorteringssystemet bidrar betydelig til å redusere behovet for trucker og andre tunge maskiner som vanligvis brukes til å transportere varer på lageret. Ved å implementere automatiserte løsninger kan sorteringsprosessen utføres både mer effektivt og nøyaktig. Dette reduserer behovet for bruk av trucker og andre tunge transportkjøretøy, som ofte er drevet av diesel eller fossilt brensel, og som ellers bidrar til økte CO₂-utslipper. Dette skaper en mer bærekraftig driftsmodell på lageret, der sorteringsystemet oppfyller både FNs åttende mål om anstendig arbeid og økonomisk vekst, og det niende målet om innovasjon og infrastruktur (FN 2024c).

I tillegg kan den reduserte avhengigheten av manuelle kjøretøy forbedre sikkerheten på lageret, siden automatiserte systemer er mer tilregnelige og pålitelige, og er programmert til å operere med høy nøyaktighet til enhver tid. Det minsker også nødvendigheten for å ansette eller opplære folk som ikke har truckførerbevis.

Sorteringssystemet oppfyller også det tredje bærekraftsmålet til FN om god helse og livskvalitet. Automatisering av lagersortering vil i stor grad redusere behovet for manuell håndtering av tunge varer, hvilket betyr at ansatte vil i mindre grad være eksponert for arbeidsoppgaver som er fysisk krevende (FN 2024b). Dette har en positiv effekt på helse og sikkerhet på arbeidsplassen. Ved å fjerne behovet for tunge løft av lagerarbeidere, reduseres risikoen for akutte skader, samt småskader som over tid kan føre til større skader.

Denne forberedingen i arbeidsforhold hos de ansatte som jobber på lageret kan også føre til redusert sykefravær og færre langtidsfravær relatert til arbeidsskader. Arbeidsmiljøet vil bli tryggere, og det vil sannsynligvis resultere i lavere kostnader når det gjelder yrkesskadeforsikringer. Færre arbeidsulykker kan også øke trivselen og moralen hos de ansatte på lageret, noe som kan føre til en økt effektivitet av arbeidet.

Ytterligere vil automatiseringen av lagersortering bidra til økt likestilling på arbeidsplassen, noe som går i tråd med det femte målet til FN (FN 2024d), likestilling mellom kjønnene. Ved å redusere behovet for tunge løft, blir arbeidsmiljøet mer tilgjengelig for flere, inkludert kvinner. Dette kan føre til at arbeidsgiveren kan fokusere mer på ferdigheter som samarbeid, fremfor fysisk stryke, noe som kan åpne for et større mangfold.

10.2 Etikk

Automatiseringen av lagersortering byr på utfordringer og fordeler når man titter på de etiske betydningene. Systemet vil erstatte manuelle jobber med automatiserte systemer, hvilket betyr at det vil redusere sysselsettingen av mennesker med tradisjonelle lagerroller. Skifter man lageret til noe mer teknologisk avansert, krever det en høyere kynighet for å jobbe på lageret, som kan også være til ulempe for enkelte ansatte. Dette vil skape usikkerhet for de som er avhengige av slike jobber som dette.

Samtidig vil automatiseringen skape en rekke nye arbeidsmuligheter, især innen ingeniørfag og systemvedlikehold. Det vil også bli et økt behov for selvger, ettersom automatiseringen vil effektivisere gjennomstrømningen av varer, hvilket øker behovet for klienter. Skoler og andre læringssentere må ta ansvar for å følge slike teknologiske endringer ikke bare på lager, men også i andre dimensjoner av samfunnet generelt, ved å tilby opplæring og omskolering for slike teknologiske endringer.

Når det gjelder bærekraftsmålene, vil automatiseringen bidra til en betydelig grønnere drift. Det blir et mindre utsipp av drivhusgasser, det blir tryggere for ansatte, det åpner muligheter for et større mangfold i arbeidsplassen og det øker effektiviteten. Det er mange av de positive fordelene som går i tråd med bærekraftsmålene til FN.

Referanser

- FN (2024a). *Anstendig arbeid og økonomisk vekst — fn.no*. URL: <https://fn.no/om-fn/fns-baerekraftsmaal/anstendig-arbeid-og-økonomisk-vekst> (sjekket 13. nov. 2024).
- (2024b). *God helse og livskvalitet — fn.no*. URL: <https://fn.no/om-fn/fns-baerekraftsmaal/god-helse-og-livskvalitet> (sjekket 13. nov. 2024).
- (2024c). *Industri, innovasjon og infrastruktur — fn.no*. URL: <https://fn.no/om-fn/fns-baerekraftsmaal/industri-innovasjon-og-infrastruktur> (sjekket 13. nov. 2024).
- (2024d). *Likestilling mellom kjønnene — fn.no*. URL: <https://fn.no/om-fn/fns-baerekraftsmaal/likestilling-mellom-kjønnene> (sjekket 13. nov. 2024).
- SwissLog (2024). *AutoStore: Space saving storage and order picking system for small parts*. URL: <https://www.swisslog.com/en-gb/products-systems-solutions/asrs-automated-storage-retrieval-systems/autostore-integrator> (sjekket 13. nov. 2024).

Appendix

Ordbok

Branch En separat utviklingslinje innenfor et versjonskontrollsyste m som lar flere utviklere jobbe med forskjellige funksjoner uten konflikt. 10

Code Review En prosess der utviklere vurderer hverandres kode for å sikre kvalitet og fange opp feil før koden legges inn i hovedprosjektet. 10

DevOps DevOps er en tilnærming til programvareutvikling og drift som fremmer samarbeid mellom programvareutviklere og driftsteam. Målet er å muliggjøre raskere og mer pålitelig utvikling og testing. . 9, 10

GitLab En plattform for versjonskontroll og samarbeid på kodeprosjekter, ofte brukt i profesjonelle utviklingsmiljøer. 10

Merge Request En forespørsel om å slå sammen endringer fra en branch til en annen i et prosjekt, ofte etter en vellykket code review. 10

A Prosjektplan

2024

Prosjektplan Gruppe 21

LEGO PROSJEKT I EMNET IDATT1004

HARALD, KRISTIAN, VETLE, JOHANNES OG WILLIAM

Innholdsfortegnelse

Problembeskrivelse.....	2
SMART-mål.....	2
<i>Spesifikk (Effektmål og Resultatmål).....</i>	<i>2</i>
<i>Målbart (Resultatmål)</i>	<i>2</i>
<i>Ansvarlig / Oppnåelig (Prosessmål)</i>	<i>2</i>
<i>Realistisk (Prosessmål og Effektmål)</i>	<i>3</i>
<i>Tidsbestemt (Resultatmål og Prosesstmål).....</i>	<i>3</i>
Samfunnsmål.....	3
Fremdriftsplan	3
Risikovurdering	4
<i>Risikoregister.....</i>	<i>4</i>
<i>Risiko- og mulighetsmatrise</i>	<i>4</i>
Kvalitetssikring av programkode.....	5

Problembeskrivelse

Formålet med prosjektet er å utvikle en mini-sorteringsrobot som skal sortere fargeide legoklosser. Roboten skal bestå av en 3D-printet ramme med et lasteområde, hvor en kran (montert på en todimensjonal x-y portalkran med bevegelig z-akse) plukker opp klossen og plasserer den i en av de seks tilhørende båsene. Båsene benyttes til lagring og sortering av klosser basert på ulike farger. Vi har valgt seks båser som et hensiktsmessig antall, gitt begrensningen på antall byggeklosser tilgjengelig. Systemet vil dermed bidra til en effektiv og automatisert sorteringsprosess som kan utvides betraktelig ved behov.

SMART-mål

Spesifikk (Effektmål og Resultatmål)

Roboten skal kunne transportere klosser til og fra lasteområdet og plassere dem i én av seks båser, noe som skal bidra til en automatisk og effektiv sorteringsprosess. Dette avlaster manuell håndtering, slik at arbeiderne kan vise tiden sin til andre oppgaver.

Målbart (Resultatmål)

Systemet skal kunne sortere minst 95 % av klossene korrekt i løpet av flere tester, og kunne sortere seks klosser i løpet av maksimalt tre minutter. Dette skal dokumenteres gjennom testing av roboten.

Ansvarlig / Oppnåelig (Prosessmål)

Hver oppgave i prosjektet gjennomføres av teammedlemmene i fellesskap, hvilket sikrer gjennomføring av oppgavene i større grad, og at alle medlemmene får prøve å løse ulike utfordringer som oppstår. Alle teammedlemmene er enige om målet satt. Målet anses som litt utfordrende, men samtidig realistisk å gjennomføre oppgaven med de tilgjengelige ressursene. Gruppen skal fullføre alle definerte milepåler i framdriftsplanen, inkludert bygging, programmering, testing og dokumentasjon.

Realistisk (Prosessmål og Effektmål)

Et av målene med prosjektet er å øke teamets kompetanse innenfor robotteknologi, programmering og teamarbeid gjennom praktisk erfaring. Med de tilgjengelige ressursene (lego-klosser, komponenter og tid) anses dette som realistisk.

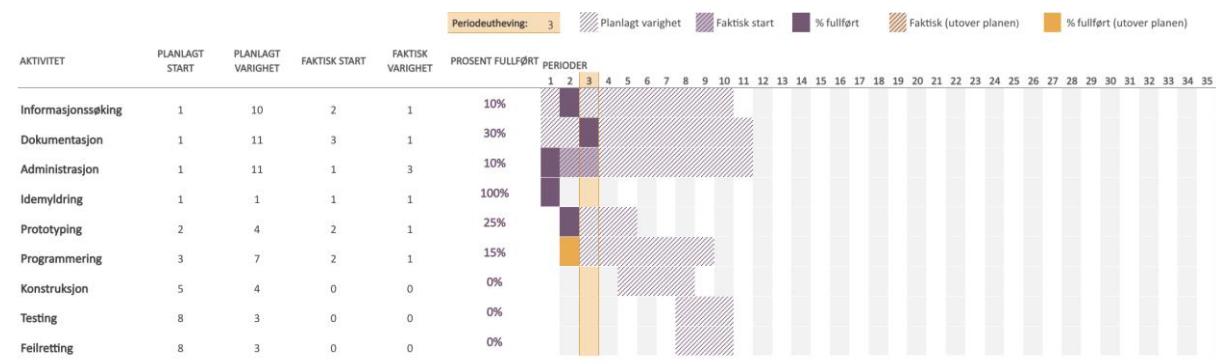
Tidsbestemt (Resultatmål og Prosessmål)

Roboten skal være ferdig utviklet og testet i henhold til den oppsatte tidsplanen. Rapport og nødvendig dokumentasjon for arbeidet skal også være fullført innen oppsatt tid.

Samfunnsmål

Formålet med prosjektet er å utvikle en prototype som kan illustrere hvordan automatiserte sorteringssystemer kan fungere i den virkelige verden. Slike systemer brukes i ulike industrielle sammenhenger for å effektivisere produksjonslinjer og redusere manuelt arbeid. Gjennom prosjektet ønsker vi å demonstrere hvordan en enkel sorteringsrobot kan løse reelle utfordringer knyttet til effektivitet og presisjon i sorteringsprosesser.

Fremdriftspläne



Risikovurdering

Risikoregister

- A. **Tekniske utfordringer:** Problemer med motorer eller sensorfeil kan forstyrre robotens funksjonalitet.
 - **Tiltak:** Kjøre regelmessige tester.
- B. **Tidsforsinkelser:** Uventede forsinkelser kan oppstå ved bygging eller programmering.
 - **Tiltak:** Ha en buffer i tidsplanen for å håndtere uforutsette problemer.
- C. **Komponentmangel:** Begrensninger i tilgjengelige legoklosser kan redusere sorteringskapasiteten.
 - **Tiltak:** Juster sorteringsstrategien for å matche tilgjengelige ressurser, og planlegg bruken av komponenter nøye.
- D. **Kommunikasjonsproblemer:** Manglende kommunikasjon i teamet kan føre til misforståelser og feil.
 - **Tiltak:** Innføre regelmessige statusmøter og sørge for at alle teammedlemmer er oppdatert på fremdriften.
- E. **Kvalitetsutfordringer:** Koden eller konstruksjonen kan ha feil som påvirker systemets ytelse.
 - **Tiltak:** Ha rutiner for kontinuerlig testing og kvalitetskontroll, inkludert kodegjennomgang og funksjonstesting etter hver milepål.
- F. **Mangel på erfaring:** Manglende erfaring blant teammedlemmene kan føre til ineffektiv problemløsning.
 - **Tiltak:** Gjennomføre opplæring i relevante verktøy og teknologier ved behov, samt jobbe sammen for å dra nytte av hverandres kompetanse.

Risiko- og mulighetsmatrise

		Risiko			
		Liten	Middels	Kritisk	Katastrofal
Sannsynlighet					
Svært stor				C	
Stor					
Middels			E	A	
Liten			B		
Minimal	D F				

Kvalitetssikring av programkode

- **Testing:** Det skal være grundig og *kontinuerlig* testing, både unit og e2e. Det må testes hvor bra man kan teste e2e på programvare. Det skal bli satt opp automatisk testing.
- **Kodeinspeksjon:** Teammedlemmer vil gjennomgå hverandres kode for å avdekke feil og mulige forbedringer. Kodeinspeksjoner vil bli utført regelmessig for å sikre at koden holder høy kvalitet og at beste praksiser følges.
- **Versjonskontroll:** Bruke versjonskontrollsystemer for å spore endringer i koden og muliggjøre enkel tilbakeføring ved behov. Dette sikrer også at alle teammedlemmer kan jobbe med oppdaterte versjoner av koden.
- **Kodedokumentasjon:** Sørge for at koden er godt dokumentert, slik at man raskt og enkelt kan få oversikt ved å lese gjennom. Dokumentasjon vil også hjelpe med vedlikehold og videreutvikling av systemet. Koden skal være så selvdokumenterende som mulig. Man skal ikke skrive en kommentar om noe er selvsagt, ettersom kommentarer fort blir utdatert.

B Arbeidskontrakt

Arbeidskontrakt for gruppe 21

Medlemmer:

- Harald Theodor Helland Velde
- Johannes Aamot-Skeidsvoll
- Kristian Ask Selmer
- Vetle Nilsen
- William Laukvik Holtsdal

Innledning

Denne arbeidskontrakten definerer samarbeidsbetingelsene og forpliktelsene mellom medlemmene i gruppe 21. Kontraktenes formål er å sikre et effektivt og strukturert samarbeid, tydeliggjøre medlemmenes forventninger, samt danne prosedyrer for hvordan utfordringer og konflikter skal håndteres underveis i arbeidet.

Alle medlemmer av gruppen har gjennomgått denne kontrakten og aksepterer vilkårene. Dette dokumentet skal bidra til å sikre at prosjektet leveres i henhold til avtalt kvalitet og tidsfrist.

Mål

Effektmål

- Gruppen skal opprettholde god intern kommunikasjon og samarbeidsånd gjennom hele prosjektperioden.
- Gruppen skal sikre at hvert medlem har lært noe nytt og utvikler sine ferdigheter gjennom prosjektet.

Resultatmål

- Oppgaver/øvinger skal fullføres/leveres til avtalt tid, uten forsinkelser.
- Alle medlemmer er fornøyde med og stolte over fullført arbeid.
- Alle medlemmer står sterkere sammen ved prosjektets slutt.

Oversikt over roller

Utstyr/ressurs ansvarlig: Harald Th. H. Velde

Dokumentansvarlig: William Laukvik Holtsdalen

Møteansvarlig: Johannes Aamot-Skeidsvoll

Kvalitetskontrollør: Vetle Nilsen

Møtereferent: Kristian Ask Selmer

Prosedyrer

A. Møteinnkalling

Møteinnkallinger kommer til å bli distribuert på Discord (via “events”), samt i delt Outlook kalender, men kopier kan forekomme også på e-post og/eller andre kanaler. Innkallingen skal i utgangspunktet inneholde møtets agenda og eventuelle medlemmer som ikke er del av gruppen. Liste over deltakere som ikke skal delta bør være skrevet i innkallingen. Innkallingen skal så videre inneholde tid for møtet og sted for oppmøte. Innkallingen skal sendes senest minst 2 dager før møtets startdato. Unntak gjelder kun hastemøter, der kan heller ikke oppmøte kreves.

B. Varsling ved fravær eller andre hendelser

Dersom man kommer for sent i henhold til avtalt tid, eller er i ustand til å møte opp, så skal man varsle på discord i fravær-kanalen.

C. Dokumenthåndtering

Dokumenter og filer skal lastes opp på discord i riktig kanal. Det finnes tre ulike kategorier; en for møtereferat, en for dokumentarkiv og en for andre filer, som for eksempel applikasjoner og PDF-filer ment for å dele og ikke å arkivere.

D. Innleveringer av gruppearbeider

Faglig innhold skal kontrolleres av gruppens kvalitetskontrollør før ferdigstilling av arbeid. Innleveringer og gruppearbeid skal fullføres og/eller leveres til avtalt tidspunkt.

E. Beslutninger

Alle beslutninger skal fattes i felleskap der minst tre medlemmer er til stede.

Demokratisk flertall ved håndsopprekning gjelder ved beslutninger.

Interaksjon

A. Oppmøte og forberedelse

Medlemmer bør utføre nødvendige forberedelser før møter og arbeidstimer, og alle medlemmer skal møte opp til avtalt tid. Ved uforutsette hendelser som forsinket buss eller lignende, skal gruppa varsles så tidlig som mulig, slik at de kan justere planene sine deretter.

B. Tilstedeværelse og engasjement

Alle medlemmer skal være fokuserte og engasjere under møter og samarbeidsoppgaver. Alle idéer skal bli lyttet på, og tas i betraktning av de andre gruppemedlemmene.

C. Hvordan støtte hverandre

Man skal si ifra når man trenger hjelp, og medlemmene skal assistere hverandre ved behov. Har man overskudd og muligheten til å hjelpe, assisterer man uoppfordret.

D. Uenighet, avtalebrudd og utkastelse

Ved uenigheter som ikke kan løses umiddelbart, skal et møte settes opp med dette som hovedfokus. For å oppnå enighet kan det benyttes demokratisk flertall ved håndsopprekning, eller samtaler med studieassistent og/eller faglærer.

Ved avtalebrudd skal eventuelle sanksjoner diskuteres i et møte hvor faglærere eller studieassistenter skal involveres om nødvendig. Det er viktig at gruppa er klar over at sanksjoner ikke er ment som en straff, men som et middel for å opprettholde ansvarlighet og samarbeidsånd. Kriteriet for utkastelse fra gruppa er mer enn fire dager udokumentert fravær. Om et slikt tilfelle skulle oppstå i løpet av arbeidsforholdet, skal det settes opp et møte med faglærer for å diskutere situasjonen, og alle involverte parter skal gis muligheten til å presentere sin side av saken før en endelig beslutning blir vedtatt.

Signaturer

Harald Theodor Helland Velde

Harald Th. H. Velde

Kristian Ask Selmer

Kristian Ask Selmer

Vetle Nilsen

VN

William Laukvik Holtsdalén

William L. Holtsdalén

Johannes Aamot-Skeidsvoll

J. Aamot

C Kjernekvadranten

Kjernekvadranten

Harald:

Kjernekvalitet / talent: <ul style="list-style-type: none">• Arbeidsvillig• Tålmodig• Nøyaktig	FALLGRUVER: <ul style="list-style-type: none">• Utbrent• Unødvendig tidsbruk• For detaljorientert
ALLERGI: <ul style="list-style-type: none">• Uinteresserte personer som ikke gir en oppgave en sjanse• Dårlig løste oppgaver	TRENINGSLEIR / UTFORDRINGER: <ul style="list-style-type: none">• Begrense arbeidsmengde• Sette tidskrav og leve uansett tilstand

William:

Kjernekvalitet / talent <ul style="list-style-type: none">• Rolig• Optimistisk• Nøye	Fallgruve <ul style="list-style-type: none">• Kan være for rolig, og utsette oppgaver jeg er trygg på• Tenker at alt går bra, finner en løsning. Kan være negativt fremover med økt vanskelighetsgrad i fagene.• Kan være for nøye og bruke for mye tid på forbedre allerede fungerende løsninger.
Allergi <ul style="list-style-type: none">• Mot ansvarsfraskrivelse• Mot unødvendig stress	Treningsleir / utfordring <ul style="list-style-type: none">• Jobbe med å ikke utsette oppgaver til siste frist• Bli mindre nøye når det ikke har noe å si for resultatet

Kristian

Kjernekvalitet / talent:	Fallgruve:
<ul style="list-style-type: none">• Pålitelig• Arbeidsvillig• Analytisk	<ul style="list-style-type: none">• Pedantisk• Utbrent• Perfeksjonisme og overtenking
Allergi:	Treningsleir / utfordring:
<ul style="list-style-type: none">• Upålitelig• Lite arbeidsvillig• Overfladisk	<ul style="list-style-type: none">• Fleksibilitet• Fleksibilitet• Handle raskere

Vetle

Kjernekvalitet / talent:	Fallgruve:
<ul style="list-style-type: none">• Løsningsorientert• Realistisk• Logisk	<ul style="list-style-type: none">• Tendens til å overkjøre• Negativ• Tunell-syn
Allergi:	Treningsleir / utfordring:
<ul style="list-style-type: none">• Hvis teamkamerater ikke bryr seg, og ikke løfter sin last• Ubesluttsomhet	<ul style="list-style-type: none">• Kreativitet• Multitasking• Positivitet

Johannes

Kjernekvalitet / talent: <ul style="list-style-type: none">• Rask lærer• Effektiv• Optimistisk	Fallgruve: <ul style="list-style-type: none">• Overstyrer hvis engasjert• Kan unødvendig stresse
Allergi: <ul style="list-style-type: none">• Ingen innsats/passiv	Treningsleir / utfordring: <ul style="list-style-type: none">• Langvarig motivasjon• Prokrastinere• Passiv hvis lite motivasjon

Egenskaper på teamet:

Teamets kvaliteter

- Arbeidsvillig
- Tålmodig
- Nøyaktig
- Rolig
- Optimistisk
- Nøye
- Pålitelig
- Analytisk
- Løsningsorientert
- Realistisk
- Logisk
- Rask lærer
- Effektiv
- Optimistisk

Allergi

- Dårlig løste oppgaver
- Ansvarsfraskrivelse
- Unødvendig stress
- Upålitelig
- Lite arbeidsvillig
- Overfladisk
- Teamkamerater som ikke bryr seg, og ikke løfter sin last
- Ubesluttsomhet

Fallgruver

- Utbrent
- Unødvendig tidsbruk
- For detaljorientert
- Prokrastinere
- Tenker at alt går bra
- Pedantisk
- Perfeksjonisme og overtenking
- Kan overkjøre
- Negativ
- Tunell-syn
- Kan unødvendig stresse

Treningsleir

- Begrense arbeidsmengde
- Fleksibilitet
- Handle raskere
- Kreativitet
- Multitasking
- Positivitet
- Langvarig motivasjon
- Prokrastinere
- Passiv hvis lite motivasjon

D Møteinkalling veiledningsmøte 1

Innkalling til veiledningsmøte 4.9.24 med Team 21

Møteinnkallingen går til:

Harald, Kristian, Vetle, Johannes, William og Shiza

Tid og sted:

Onsdag 4.9.2024 klokken 16:00 – 16:25

Gløshaugen IT-bygget,

Sydfløy 2. etasje

Møterom 119c

(Lenke til MazeMap): <https://bit.ly/2DxwYpo>

Møteleder: Johannes

Referent: Kristian

Agenda for møtet:

	Sak
1	<u>Team-etablering</u> Hvilke aktiviteter er gjennomført for å etablere teamet? Er teamet godt etablert og fungerer det bra?
2	Status på Gitlab-bruk i teamet
3	<u>Kjernekvadranten</u> - erfaringer og kort presentasjon av resultat
4	<u>Arbeidskontrakt</u> - Gjennomgang/godkjenning av innholdet i kontrakten. Er den tilstrekkelig for deres team for å håndtere fremtidige situasjoner?

Det blir ingen pauser underveis i møtet.

Send melding i «fravær» kanalen på discord dersom du ikke har mulighet til å møte opp.

Velkommen til et hyggelig veiledningsmøte!

E Møteinkalling veiledningsmøte 2

Innkalling til veiledningsmøte 23.10.24 med Team 21

Møteinnkallingen går til:

Harald, Kristian, Vetle, Johannes, William og Shiza

Tid og sted:

Onsdag 23.10.2024 klokken 15:00 – 15:22

Gløshaugen Elektrobygget,

3. etasje

Grupperom E304

(Lenke til MazeMap): <https://link.mazemap.com/FMh8BBWw>

Møteleder: Johannes

Referent: Kristian

Agenda for møtet:

	Sak
1	<u>Team-utvikling</u> Hvilke aktiviteter er gjennomført for å videreutvikle teamet? Fungerer teamet bra?
2	<u>Gjennomgang av prosjektplanen</u> Problembeskrivelse, SMART-mål, Framdriftsplan, risikovurdering og rutiner for kvalitetssikring av kode
3	<u>Status for prosjektet</u> Fremdrift (planleggingsfase, bygging og koding) Behov for ekstra Lego
4	<u>Revidert arbeidskontrakt</u> Gjennomgang/godkjenning av revidert innhold i kontrakten.

Det blir ingen pauser underveis i møtet.

Send melding i «fravær» kanalen på discord dersom du ikke har mulighet til å møte opp.

Velkommen til et hyggelig veiledningsmøte!

F Møtereferat veiledningsmøte 1

Møtereferat - gruppe 2I

Dato: 4.9.2024

Tid: 16:00 – 16:30

Sted: IT-bygget, sydfløy Etasje 2, rom 201

Møteleder: Johannes

Referent: Kristian

Deltakere:

- Kristian Ask Selmer
- Harald Theodor Helland Velde
- Vetle Nilsen
- Johannes Aamot-Skeidsvoll
- William Laukvik Holtsdal
- Shiza

Agendaen

1. Team-establering:

Hvilke aktiviteter er gjennomført for å etablere teamet? Er teamet godt etablert og fungerer det bra?

Det var stor enighet i gruppa om at teamet fungerer godt per i dag, selv om dette er vanskelig å si tidlig i samarbeidet. Dette skyldes at det ikke har vært nødvendig at hele gruppa har måttet jobbe sammen for å gjennomføre arbeid, der øvinger hittil bare har krevd én eller to personer på gruppa for å gjennomføre det.

2. Status på Gitlab-bruk i teamet

Alle på gruppa har satt opp Git på hver sin maskin, og har muligheten til å gjøre endringer på prosjektet og øvinger. Oppsettet av selve prosjektet på Gitlab er også gjennomført iht. instrukser gitt av professoren.

3. Kjernekvadranten

En kjernekvadrant som hadde alle kjernekvalitetene til hvert enkelt gruppemedlem ble gått gjennom med studieassistenten.

4. Arbeidskontrakt

Gjennomgang/godkjenning av innholdet i kontrakten. Er den tilstrekkelig for deres team for å håndtere fremtidige situasjoner?

Førsteutkastet av arbeidskontrakten ble sett igjennom sammen med studieassistenten, og var stort sett grei. Den hadde eksempelvis gode rammer for effektmål og resultatmål, men den bar på enkelte mangler når det kom til noen prosedyrer og interaksjoner.

Studieassistenten hadde et ønske om at prosedyrer iverksatt av for eksempel stort fravær av en person, eller uenigheter innad i gruppa, var mer konkretiserte og entydige, slik at de nødvendige stegene videre var helt klare. En klar prosedyre for en stor uenighet i gruppa kan for eksempel innebære en håndsopprekning.

Møtet ble godkjent av studieassistenten.

G Møtereferat veiledningsmøte 2

Møtereferat - gruppe 21 - Avsluttende prosjekt

Dato: 23.10.2024

Tid: 15:00 – 15:20

Sted: IT-bygget, sydfløy, Etasje 2

Møteleder: Johannes

Referent: Kristian

Deltakere:

- Kristian Ask Selmer
- Harald Theodor Helland Velde
- Vetle Nilsen
- Johannes Aamot-Skeidsvoll
- Shiza

William Laukvik Holtsdal hadde ikke muligheten for å delta på møtet grunnet sykdom.

Agendaen for møtet

1. Team-utvikling

Hvilke aktiviteter er gjennomført for å videreutvikle teamet? Fungerer teamet bra?

Gruppa har gjennomført flere sosiale aktiviteter, som for eksempel å gå ned på caféen for å kjøpe kanelboller sammen. Samarbeidet har også strukket seg utover selve faget, der gruppa har jobbet sammen i både matematikk for ingeniørfag I og programmering, numerikk og sikkerhet.

2. Gjennomgang av prosjektplanen

Problembeskrivelse, SMART-mål, Framdriftsplan, risikovurdering og rutiner for kvalitetssikring av kode

Gruppa gikk igjennom prosjektplanen med studieassistenten. Framdriftsplanen så ganske grei ut, men gruppa ble enige om at den kanskje ikke vil være helt realistisk, og at det kan bli nødvendig å justere planen underveis.

I risiko- og sannsynlighetsmatrisa fikk gruppa avklart med studieassistenten at det holder å inkludere kun sannsynlighet og risiko i den, og at delen med sannsynlighet og mulighet kan sløyfes.

Gruppa diskuterte også muligheten for å forme matrisa om til en graf, noe som kan tilby mer nøyaktighet.

3. Status for prosjektet

Fremdrift (planleggingsfase, bygging og koding)

Gruppa har så vidt begynte på koden, men har sjekket at alle sensorene fungerer som de skal. Foreløpig jobber gruppa med å bygge roboten og finne deler.

Behov for ekstra lego?

Gruppa kan spørre studieassisterter om å få tilgang til et skap som har flere brikker.

4. Revidert arbeidskontrakt

Gjennomgang/godkjenning av revidert innhold i kontrakten

Gikk gjennom endringene, og fikk det godkjent av studieassistenten.

Vurdering av møtet:

Gruppa fikk godkjent møtet av studieassistenten.

H Timelister

Oppsummering av timelister i prosjekt nr:

Ukenr	Harald	Johannes	Kristian
Uke 41	3.5	3.5	3.0
Uke 42	7.5	5.0	5.0
Uke 43	10.0	10.0	10.0
Uke 44	8.5	9.5	9.5
Uke 45	7.0	10.5	6.0
Uke 46	10.0	6.0	10.5
Uke 47	6.0	6.0	6.0
Sum antall timer pr person/totalt	52.5	50.5	50.0

Oppsummering av timer fordelt på aktivitet

Aktivitet	Harald	Johannes	Kristian
Egenopplæring	0	0	0
Informasjonssøking	2	0.5	8
Administrasjon av eget arbeid	0	0	5
Prototyping	15	8	9
Implementasjon - kildekode	0	12.5	0
Testing av egne program	2	3.5	3
Feilretting av program	1.5	0.5	0
Utarbeidelse av prosjektrapport	24.5	17	23.5
Presentasjon med forberedelse	1	1	1
Teammøter	0	0	0.5
Teammøter med veileder	1	1	0
Skole	1	1	0
Annet	4.5	5.5	0
Sum antall timer totalt	52.5	50.5	50

Oppsummering av timer fordelt på kategori

Kategori	Harald	Johannes	Kristian
Dokumentasjon	26.5	17	24

Administrasjon	1	4	10
Programmering	5.5	18	3
Konstruksjon	16	8	10
Annet	3.5	3.5	3
Sum antall timer totalt	52.5	50.5	50

Vetle	William	Sum timer pr uke
3.5	3.5	17.0
5.0	6.5	29.0
10.0	13.0	53.0
10.0	12.0	49.5
7.0	10.0	40.5
10.0	8.0	44.5
8.0	1.0	27.0
53.5	54.0	260.5

Vetle	William	Sum totalt pr aktivitet
0	0	0
6.5	3.5	20.5
0	0	5
21	28.5	81.5
0	8.5	21
4	2.5	15
0	0	2
17	8.5	90.5
4.5	2	9.5
0	0	0.5
0.5	0.5	3
0	0	2
0	0	10
53.5	54	260.5

Vetle	William	Sum totalt pr kategori
21.5	9.5	98.5

10.5	2.5	28
7	10	43.5
11	28.5	73.5
3.5	3.5	17
53.5	54	260.5

Aktiviteter
Egenopplæring
Informasjonssøking
Administrasjon av eget arbeid
Prototyping
Implementasjon - kildekode
Testing av egne program
Feilretting av program
Utarbeidelse av prosjektrapport
Presentasjon med forberedelse
Teammøter
Teammøter med veileder
Sykdom
Skole
Annet

Kategorier
Dokumentasjon
Administrasjon
Programmering
Konstruksjon
Annet
Testing
Oppstart

Timelister med statusrapporter

Ukerapport

Detraktor
Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

Vi har fått en innledende presentasjon om gruppearbeidet.

Dette er den innledende fasen, og på onsdagen var det idemyldring.

Gruppen utførte en brainstrom-aktivitet, hvor medlemmene sendte sine ufiltrerte ideer i en felles Discord kanal. Deretter leste gruppen gjennom forslagene og forkastet de aller mest unrealistiske.

Timelister med statusrapporter

Ukerapport

Skrappart
Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

Vi begynte med å bygge forslag til lasteburet

Lagde forslag / prototyper til bevegelse-system. Forslag inkluderte hjul, skinnesystem, "slider" med fiskesnøre. Gruppen diskuterte også mulighet for å gjøre om prosjektet til at selve roboten heller kan kjøre rundt og flytte på lego-klosser på den måten, men dette forble en idé.

Gruppen startet å arbeide med prosjektrapporten

Timelister med statusrapporter

Ukerapport

Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

- Omskrivning til Latex og utvide dokumentet til alle punktene
- Div, nettspk og opplæring i Latex
- Forslag til skinne-mekanismen.
- Proto-design av skinne/gantry system CAD
- Endelig oversikt over lego-ressurser, inkl. ekstrader fra skap
- 3D Printet noen test-delar.

Timelisten		Johannes
Aktivitet	Kategori	Antall timer
Informasjonssøking	Programmering	0.5
Utarbeidelse av prosjektrapport	Dokumentasjon	3.0
Implementasjon - kildekode	Programmering	1.0
Testing av egne program	Programmering	1.0
Teammøter med veileder	Administrasjon	0.5
Prototyping	Konstruksjon	3.0
Utarbeidelse av prosjektrapport	Dokumentasjon	0.5
Feilretting av program	Programmering	0.5
Uke 43		10.0

Timeliste		William
Aktivitet	Kategori	Antall timer
Utarbeidelse av prosjektrapport	Dokumentasjon	2.0
Sykdom	Annet	
Prototyping	Konstruksjon	4.0
Prototyping	Konstruksjon	7.0
Uke 43		13.0

Timelister med statusrapporter

Ukerrapport

Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

- Montert delene til første 3D-print prototype
- Arbeidet videre med rapport
- Jobbet med endringer på CAD modell, samt idemryldring ang. Z-akse mekanisme.
- Lagt til automatisk testing og formattering i GitLab
- Laget og testet kode for manuell styring langs x, y og z akse med knapper på EV3 Brick.

Timelister med statusrapporter

Ukerapport

Skjerpet
Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

- Montert sensor og nedre del av rammen
- Kalibrert sensor / mappet rbg-verdier til farger i koden.
- Laget ark med "design" av lagersystemet, med farjede båser osv.
- Laget kode for automatisk sortering/lagring
- Laget kode for automatisk henting av kloss med bestemt farge
- Laget meny-side for automatisk henting av kloss, samt funksjonallitet for å velge farge med knappene på ev3.
- Ordnet UI på EV3

Timelister med statusrapporter

Ukerapport

Skjerpet
Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

Vi har nå blitt ferdige med produktet og fullfører rapporten og lager en presentasjon til fremvisningen.

Har også spilt inn video(er) til presentasjonen, og redigert disse sammen til en profesjonell fremvisning av robotens funksjonalitet.

Timeliste		William
Aktivitet	Kategori	Antall timer
Implementasjon - kildekode	Programmering	1.0
Utarbeidelse av prosjektrapport	Dokumentasjon	2.0
Testing av egne program	Administrasjon	1.0
Teammøter med veileder	Administrasjon	0.5
Presentasjon med forberedelse	Dokumentasjon	1.0
Utarbeidelse av prosjektrapport	Dokumentasjon	2.5
Uke 46		8.0

Timelister med statusrapporter

Ukerapport

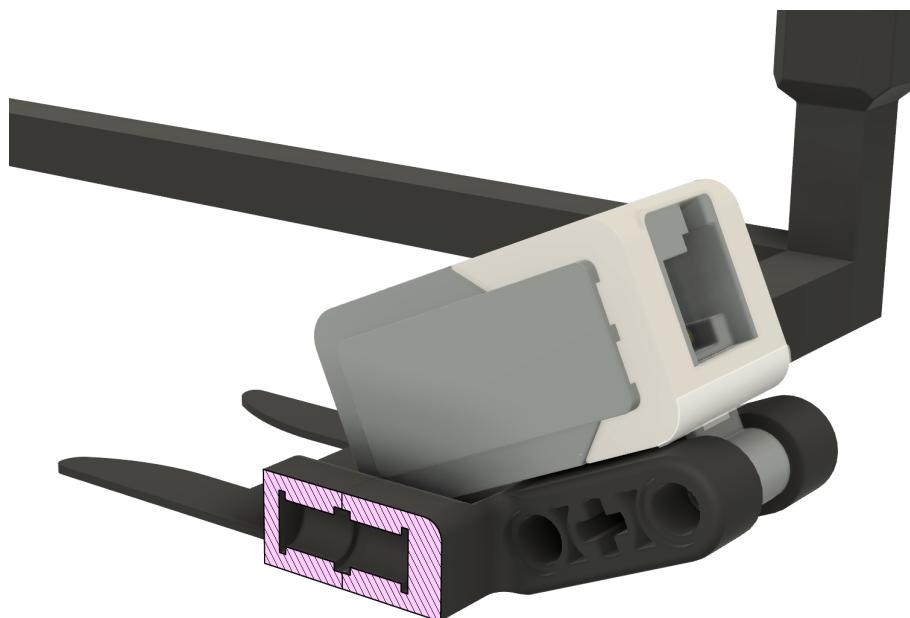
Dette skal være en kort beskrivelse av hva hver enkelt i gruppa (med navn) har arbeidet med i siste periode - og også en overordnet vurdering.

Forberedelse til presentasjon, og presentasjon av prosjekt Rapportskriving

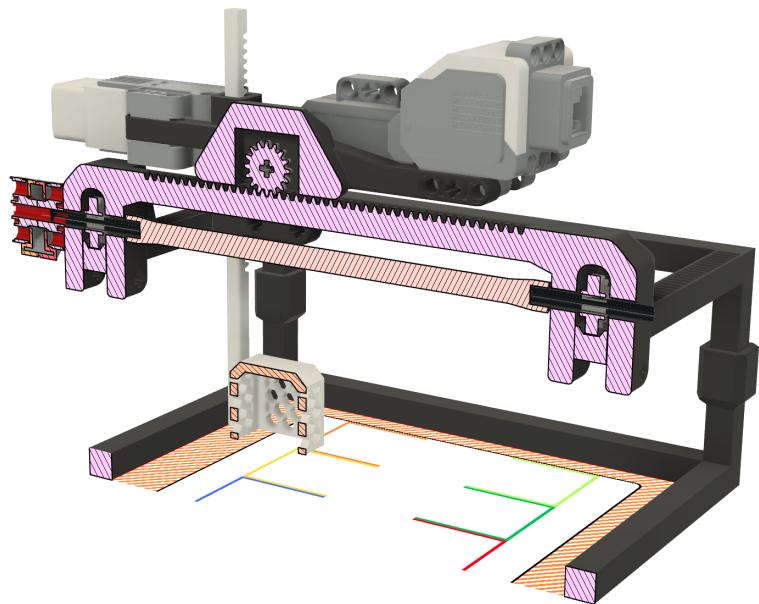
I Tekniske illustrasjoner



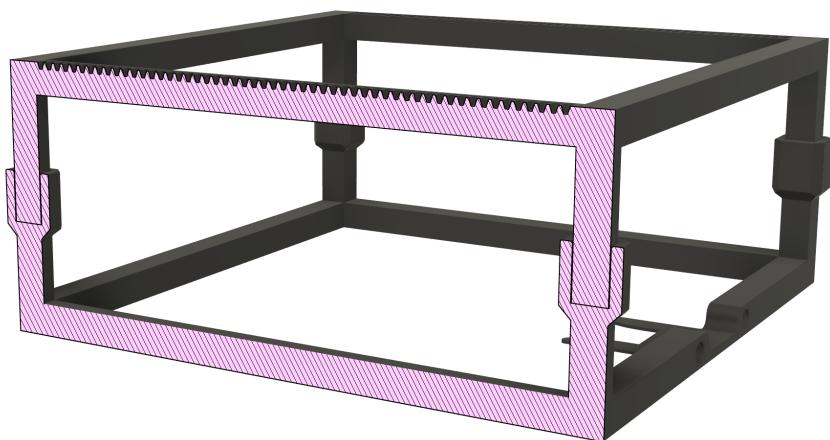
Figur 9: Hele sorteringsroboten



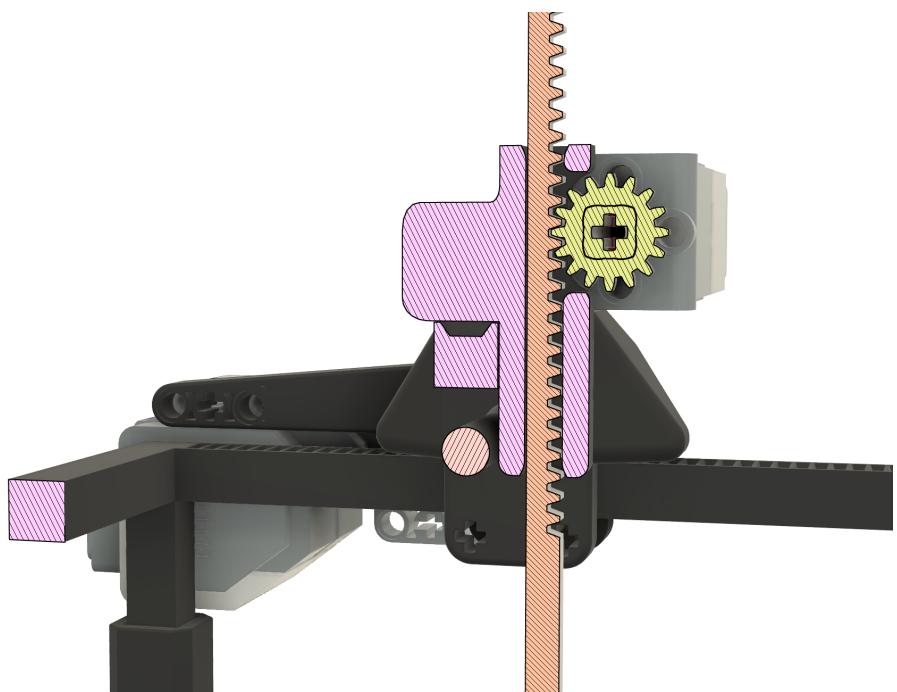
Figur 10: Tverrsnitt av sensorfeste



Figur 11: Tverrsnitt gjennom x-aksen



Figur 12: Tverrsnitt gjennom y-aksen



Figur 13: Tverrsnitt gjennom z-aksen

J Kode listing

```
1 #!/usr/bin/env pybricks-micropython
2 from robot_controller import RobotController
3
4
5 def main():
6     controller = RobotController()
7     controller.run()
8
9
10 if __name__ == "__main__":
11     main()
```

Listing 1: Inngangspunkt

```
1 from pybricks.ev3devices import Motor, ColorSensor, TouchSensor
2 from pybricks.hubs import EV3Brick
3 from pybricks.parameters import Port, Direction, Color, Button
4 from pybricks.tools import StopWatch, wait
5
6 from menu import MenuSystem
7 from robot import Robot
8
9
10 class RobotController:
11     def __init__(self):
12         # Initialize EV3 brick, motors, sensors, and timer
13         self.ev3 = EV3Brick()
14         self.timer = StopWatch()
15
16         self.motor_x = Motor(Port.D, positive_direction=Direction.
17             COUNTERCLOCKWISE)
18         self.motor_y = Motor(Port.C, positive_direction=Direction.
19             COUNTERCLOCKWISE)
20         self.motor_z = Motor(Port.B, positive_direction=Direction.
21             COUNTERCLOCKWISE)
22
23         self.color_sensor = ColorSensor(Port.S4)
24         self.touch_sensor = TouchSensor(Port.S1)
25
26         # Initialize Robot and MenuSystem
27         self.robot = Robot(
28             ev3=self.ev3,
29             max_x=785,
30             max_y=1000,
31             max_z=400,
32             motor_x=self.motor_x,
33             motor_y=self.motor_y,
34             motor_z=self.motor_z,
35             color_sensor=self.color_sensor,
36             horizontal_speed=550,
37             vertical_speed=550
38         )
39
40         self.menu_system = MenuSystem(self.ev3, self.robot)
41
42         # Initialize mode variables
43         self.mode = "manual"
44         self.z_axis_mode = False
45         self.center_button_pressed = False
46         self.mode_switch_button_is_pressed = False
```

```

45     # Calibrate the robot
46     self.robot.calibrate()
47
48     def run(self):
49         while True:
50             pressed_buttons = self.ev3.buttons.pressed()
51             mode_switch_button = self.touch_sensor.pressed()
52
53             # Mode switching logic
54             if Button.LEFT in pressed_buttons and Button.RIGHT in
55                 pressed_buttons and self.mode != "calibrate":
56                 self.ev3.light.on(Color.RED)
57                 self.ev3.screen.clear()
58                 self.ev3.screen.print("Calibration active.")
59                 self.ev3.screen.print("Please wait.")
60                 self.robot.calibrate()
61                 self.mode = "manual"
62                 self.ev3.light.on(Color.GREEN)
63                 self.ev3.screen.clear()
64                 self.ev3.screen.print("Mode: Manual control")
65                 continue
66
66             if mode_switch_button and not self.
67                 mode_switch_button_is_pressed:
68                 self.mode_switch_button_is_pressed = True
69
69             if self.mode == "manual":
70                 self.mode = "automatic_storage"
71                 self.ev3.light.on(Color.ORANGE)
72                 self.ev3.screen.clear()
73                 wait(100)
74             elif self.mode == "automatic_storage":
75                 self.mode = "automatic_retrieval"
76                 self.ev3.light.on(Color.YELLOW)
77                 self.ev3.screen.clear()
78             elif self.mode == "automatic_retrieval":
79                 self.menu_system.menu_rendered = False
80                 self.mode = "manual"
81                 self.ev3.light.on(Color.GREEN)
82                 self.ev3.screen.clear()
83                 self.ev3.screen.print("Mode: Manual control")
84
85             elif not mode_switch_button:
86                 self.mode_switch_button_is_pressed = False
87
88             # Handle modes
89             if self.mode == "manual":
90                 # Update display every 1000 ms
91                 if self.timer.time() > 1000:
92                     self.ev3.screen.clear()
93                     self.ev3.screen.print("\nZ-axis mode: " + str(self.
94                         z_axis_mode))
94                     self.ev3.screen.print("X-axis: " + str(self.robot.
95                         motor_x.angle()))
95                     self.ev3.screen.print("Y-axis: " + str(self.robot.
96                         motor_y.angle()))
96                     self.ev3.screen.print("Z-axis: " + str(self.robot.
97                         motor_z.angle()))
97                     self.ev3.screen.print("Colors: " + str(self.robot.
98                         color_sensor.rgb())))
98                     self.timer.reset()
99
100            # Toggle z-axis mode with center button

```

```

101         if Button.CENTER in pressed_buttons and not self.
102             center_button_pressed:
103                 self.z_axis_mode = not self.z_axis_mode
104                 self.center_button_pressed = True
105             elif Button.CENTER not in pressed_buttons:
106                 self.center_button_pressed = False
107
108             self.robot.manual_control(self.z_axis_mode)
109
110         elif self.mode == "automatic_storage":
111             self.robot.automatic_storage(self.timer)
112
113         elif self.mode == "automatic_retrieval":
114             self.menu_system.navigate_menu()

```

Listing 2: Robot kontroller

```

1 from pybricks.parameters import Button
2
3
4 class MenuSystem:
5     def __init__(self, ev3, robot):
6         self.ev3 = ev3
7         self.robot = robot
8         self.menu_page = 0
9         self.menu_rendered = False
10        self.up_button_pressed = False
11        self.down_button_pressed = False
12        self.center_button_pressed = False
13        self.colors = ["red", "blue", "green", "yellow", "lightGreen", "
14            orange"]
15
16    def navigate_menu(self):
17        pressed_buttons = self.ev3.buttons.pressed()
18
19        if Button.UP in pressed_buttons and Button.DOWN not in
20            pressed_buttons and not self.up_button_pressed:
21            self.up_button_pressed = True
22            self.ev3.screen.clear()
23            self.menu_rendered = False
24            self.menu_page = (self.menu_page - 1) % 6
25        if Button.UP not in pressed_buttons:
26            self.up_button_pressed = False
27
28        if Button.DOWN in pressed_buttons and Button.UP not in
29            pressed_buttons and not self.down_button_pressed:
30            self.down_button_pressed = True
31            self.ev3.screen.clear()
32            self.menu_rendered = False
33            self.menu_page = (self.menu_page + 1) % 6
34        if Button.DOWN not in pressed_buttons:
35            self.down_button_pressed = False
36
37        if Button.CENTER in pressed_buttons and not self.
38            center_button_pressed:
39            self.center_button_pressed = True
40            self.robot.retrieve(self.colors[self.menu_page])
41        if Button.CENTER not in pressed_buttons:
42            self.center_button_pressed = False
43
44        if not self.menu_rendered:
45            images = [
46                "images/red.png",

```

```

43         "images/blue.png",
44         "images/green.png",
45         "images/yellow.png",
46         "images/lightgreen.png",
47         "images/orange.png"
48     ]
49     self.ev3.screen.load_image(images[self.menu_page])
50     self.menu_rendered = True

```

Listing 3: Meny system

```

1 from pybricks.parameters import Stop, Button
2 from pybricks.tools import wait
3
4
5 class Robot:
6     def __init__(self, ev3, max_x, max_y, max_z, motor_x, motor_y, motor_z,
7                  color_sensor, horizontal_speed, vertical_speed):
8         self.ev3 = ev3
9         self.max_x = max_x
10        self.max_y = max_y
11        self.max_z = max_z
12        self.motor_x = motor_x
13        self.motor_y = motor_y
14        self.motor_z = motor_z
15        self.color_sensor = color_sensor
16        self.horizontal_speed = horizontal_speed
17        self.vertical_speed = vertical_speed
18
19        self.MOTOR_SPEED = 200
20        self.z_axis_top = 0
21        self.standby_position = (0, 0)
22        self.pick_up_position = (350, 54)
23        self.drop_off_position = (350, 0)
24        self.colors_storage = {"red": 0, "blue": 0, "green": 0, "yellow": 0,
25                               "lightGreen": 0, "orange": 0}
26        self.color_positions = {
27            "red": [(360, 450), (600, 450)],
28            "blue": [(360, 450), (75, 450)],
29            "green": [(360, 680), (600, 680)],
30            "yellow": [(360, 680), (75, 680)],
31            "lightGreen": [(360, 910), (600, 910)],
32            "orange": [(360, 910), (75, 910)]
33        }
34
35    def manual_control(self, z_axis_mode):
36        """Handle manual button-based control of the robot."""
37        pressed_buttons = self.ev3.buttons.pressed()
38
39        if z_axis_mode:
40            self._handle_vertical_movement(pressed_buttons)
41        else:
42            self._handle_forward_movement(pressed_buttons)
43
44        self._handle_horizontal_movement(pressed_buttons)
45
46    def _handle_movement(self, motor, min_angle, max_angle, pos_button,
47                        neg_button, pressed_buttons):
48        """Handle movement for a specified motor in a given axis, with min
           and max angle limits."""
49        current_angle = motor.angle()
50        if pos_button in pressed_buttons and neg_button not in
51            pressed_buttons and current_angle < max_angle:

```

```

49         motor.run(self.MOTOR_SPEED)
50     elif neg_button in pressed_buttons and pos_button not in
51         pressed_buttons and current_angle > min_angle:
52         motor.run(-self.MOTOR_SPEED)
53     else:
54         motor.stop()
55
56     def _handle_vertical_movement(self, pressed_buttons):
57         """Handle up/down movement in z-axis mode."""
58         self._handle_movement(self.motor_z, self.z_axis_top, self.max_z,
59             Button.DOWN, Button.UP, pressed_buttons)
60
61     def _handle_forward_movement(self, pressed_buttons):
62         """Handle forward/backward movement."""
63         self._handle_movement(self.motor_y, 0, self.max_y, Button.UP,
64             Button.DOWN, pressed_buttons)
65
66     def _handle_horizontal_movement(self, pressed_buttons):
67         """Handle left/right movement."""
68         self._handle_movement(self.motor_x, 0, self.max_x, Button.RIGHT,
69             Button.LEFT, pressed_buttons)
70
71     def automatic_storage(self, timer):
72         """Handle automatic storage of the LEGO bricks."""
73         color = self.detect_color()
74
75         if color is not None and self.colors_storage.get(color, 0) == 1:
76             if timer.time() > 500:
77                 self.ev3.screen.clear()
78                 self.ev3.screen.print("\n")
79                 self.ev3.screen.print("Color is already".center(21))
80                 self.ev3.screen.print("in storage".center(25))
81                 timer.reset()
82             return
83
84         # Display the detected color every 500 ms
85         if color is not None:
86             wait(250)
87             color = self.detect_color()
88             self.display_color(color)
89             timer.reset()
90
91         elif timer.time() > 500:
92             self.ev3.screen.clear()
93             self.ev3.screen.print("\n")
94             self.ev3.screen.print("No color".center(25))
95             self.ev3.screen.print("detected".center(25))
96             timer.reset()
97
98         # Exit if color is not recognized
99         if color not in self.color_positions:
100             return
101
102             self.colors_storage[color] += 1
103             # Perform the storage sequence
104             self.lift_to_top()
105             self.move_to_pickup_position()
106             self.lower_to_grab()
107             self.move_to_storage(color)
108             self.release_brick(color)
109             self.move_to_dropoff_position()
110             self.move_to_standby_position()
111             self.ev3.screen.clear()

```

```

108     def display_color(self, color):
109         """Display the detected color on the EV3 screen."""
110         self.ev3.screen.clear()
111         self.ev3.screen.print("\n")
112         self.ev3.screen.print("Detected color:".center(21))
113         self.ev3.screen.print(str(color).center(25))
114
115     def lift_to_top(self):
116         """Move the z-axis to the top position."""
117         self.motor_z.run_target(self.vertical_speed, self.z_axis_top, then=Stop.HOLD, wait=True)
118
119     def lower_to_grab(self):
120         """Move the z-axis to the bottom to grab the LEGO brick."""
121         self.motor_z.run_target(self.vertical_speed, self.max_z, then=Stop.HOLD, wait=True)
122
123     def move_to_storage(self, color):
124         """Move to the hallway and target position for the specified color.
125             """
126         positions = self.color_positions[color]
127         self.move_to_position(positions[0])
128         self.move_to_position(positions[1])
129
130     def release_brick(self, color):
131         """Release the LEGO brick by lifting the z-axis, then move back to
132             hallway."""
133         self.lift_to_top()
134         hallway_position = self.color_positions[color][0][0]
135         self.motor_x.run_target(self.horizontal_speed, hallway_position,
136             then=Stop.HOLD, wait=True)
137
138     def move_to_standby_position(self):
139         """Move to the standby position."""
140         self.move_to_position(self.standby_position, wait_x=False)
141
142     def move_to_pickup_position(self):
143         """Move to the pickup position."""
144         self.move_to_position(self.pick_up_position, wait_y=False)
145
146     def move_to_dropoff_position(self):
147         """Move to the drop-off position."""
148         self.move_to_position(self.drop_off_position)
149
150     def move_to_position(self, position, wait_x=True, wait_y=True):
151         """Move the robot to a specific position."""
152         self.motor_x.run_target(self.horizontal_speed, position[0], then=Stop.HOLD, wait=wait_x)
153         self.motor_y.run_target(self.horizontal_speed, position[1], then=Stop.HOLD, wait=wait_y)
154
155     def retrieve(self, color):
156         """Retrieve a brick of the specified color from storage."""
157         if self.colors_storage.get(color, 0) == 0:
158             return
159
160         self.colors_storage[color] -= 1
161
162         self.move_to_position(self.color_positions[color][0])
163         self.move_to_position(self.color_positions[color][1])
164         self.lower_to_grab()
165         self.move_to_position(self.color_positions[color][0])
166         self.move_to_dropoff_position()

```

```

164     self.lift_to_top()
165     self.move_to_standby_position()
166
167     def detect_color(self):
168         """Detect the color of the object in front of the color sensor."""
169         threshold = 3
170         colors = {
171             "red": (8, 0, 0),
172             "blue": (0, 2, 24),
173             "green": (-1, 5, 0),
174             "yellow": (12, 8, 1),
175             "lightGreen": (4, 7, 1),
176             "orange": (13, 1, 0)
177         }
178         value = self.color_sensor.rgb()
179
180         for color, rgb in colors.items():
181             if all(abs(value[i] - rgb[i]) < threshold for i in range(3)):
182                 return color
183
184     def calibrate(self):
185         """Calibrate the motors to the zero position."""
186         stall_torque = 40 # Percentage of the maximum torque
187         self.motor_z.run_until_stalled(-self.vertical_speed, Stop.HOLD,
188                                         stall_torque)
188         self.motor_x.run_until_stalled(-self.horizontal_speed, Stop.HOLD,
189                                         stall_torque)
189         self.motor_y.run_until_stalled(-self.horizontal_speed, Stop.HOLD,
190                                         stall_torque)
190         self.motor_z.reset_angle(0)
191         self.motor_x.reset_angle(0)
192         self.motor_y.reset_angle(0)
193         self.motor_z.run_target(self.vertical_speed, 20, then=Stop.HOLD,
194                                 wait=False)
194         self.motor_x.run_target(self.vertical_speed, 50, then=Stop.HOLD,
195                                 wait=False)
195         self.motor_y.run_target(self.vertical_speed, 55, then=Stop.HOLD,
196                                 wait=True)
196         self.motor_z.reset_angle(0)
197         self.motor_x.reset_angle(0)
198         self.motor_y.reset_angle(0)

```

Listing 4: Robot klassen

K Presentasjon av prosjektet

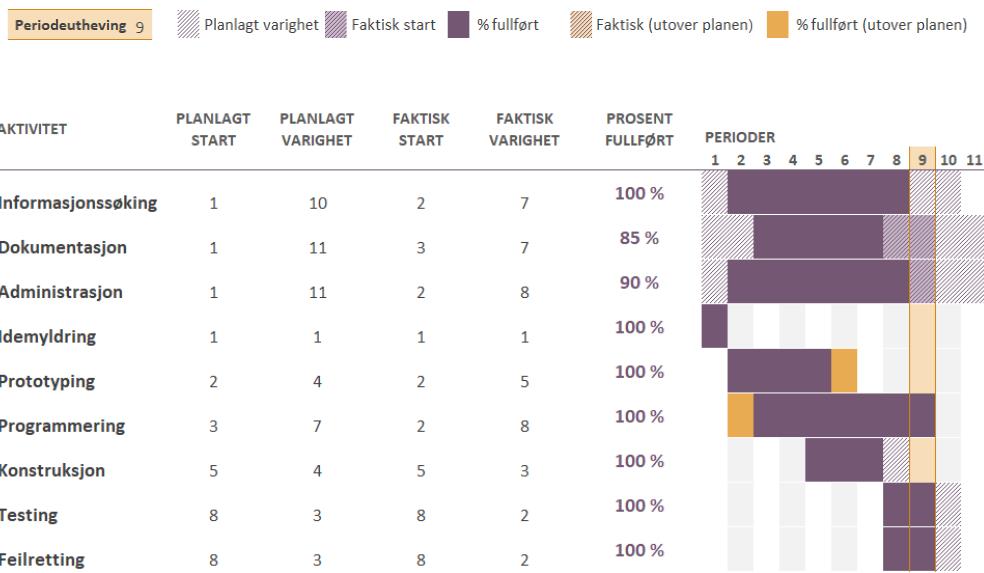
Gruppe 21

IDATT1004 – Lego prosjekt

Johannes, William, Vetle, Kristian og Harald

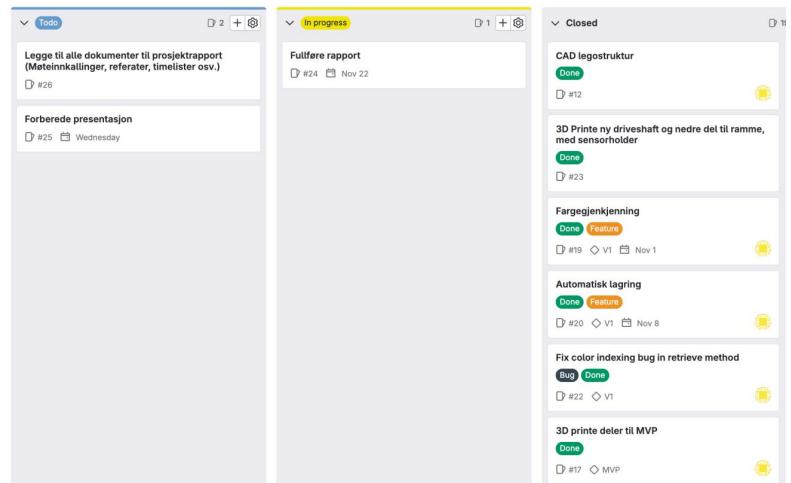
Om prosjektet

1. Idemyldringsfase
 - *Mind*-storming*
(*Brain)
 - Sorteringsrobot
2. Planlegging av fremdrift
3. Arbeidsfordeling
4. LEGO og 3D-printing
5. Programmering og testing
6. Dokumentasjon



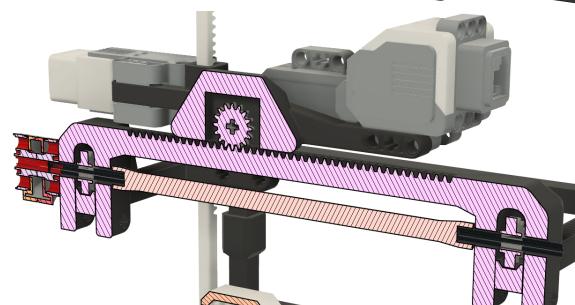
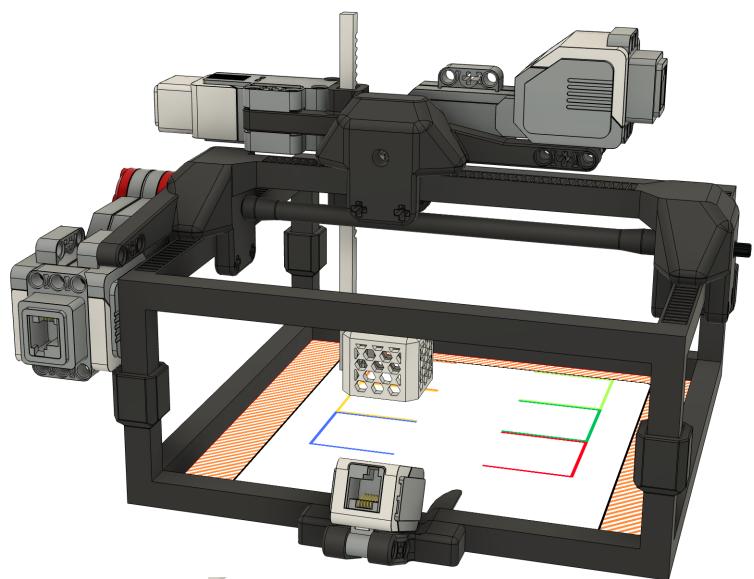
Prosjektstyring

- GitLab
 - Issueboard
 - Milestones
 - Issues
- Overleaf til rapportskriving
 - GitHub for versjonskontroll av rapport
- Discord



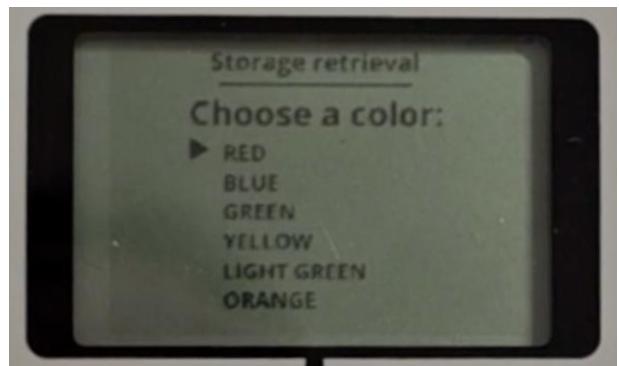
Konstruksjon

- 3D-printet struktur
 - Tannhjul med spor innfelt i ramme
 - Legobasert
- Fargesensor ved lasteområde
- Drivaksel på y-akse for presis bevegelse



Funksjonalitet

- Automatisk kalibrering av alle akser ved oppstart.
 - Sikrer nøyaktig posisjonsdata
- Bytte av modus med trykksensor
- Automatisk sortering (av farger)
- Automatisk henting av brikker (menystyrt)
- Manuell styring med EV3 knappene



Lenke til video av prosjektet:

<https://drive.google.com/file/d/1KxpAHFSPMF7ZC5oGf99bhn1ZaHlIgpXz/view>

Automatisk vs. Manuell styring

Automatisk

- Presisjon
- Effektivitet
- Følger sikkerhetsprosedyrer
- Høy oppstartskostnad

Manuell

- Fargegjenkjenning
- Interaktivitet
- Opplæring
- Avhengig av ansatte (dyrt)

Utfordringer underveis i prosjektet

- Pålitelighet av fargegjenkjenning
- Ressursmangel
 - Byggeklosser
 - Lange kabler
 - Motorer
- Belastningsstyrke ramme
- Tekniske problemer med EV3 Brick