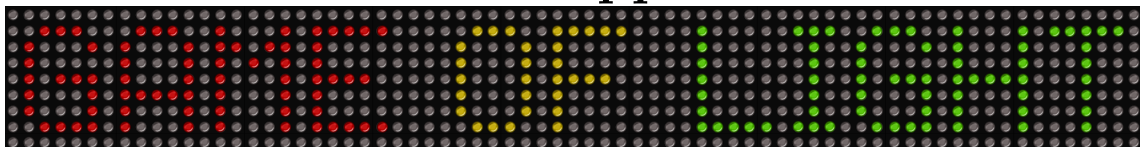


INF1510 - Bruksorientert design

22. april 2013

Sluttrapport



Skrevet av:
André Kramer Orten
Eivind Wikheim
Stian Selbek
Martin Holøs
Abdimalik

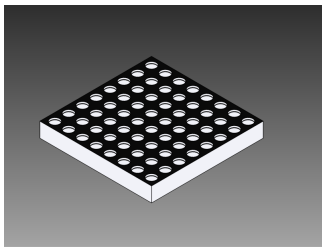
Innhold

1	Tekniske spesifikasjoner	4
2	Presentasjon av gruppa	5
2.1	Deltakere og arbeidsfordeling	5
2.2	Verktøy og software	5
2.3	Arbeidsprosessen	5
3	Valg av tema	5
3.1	prosjektets mål	6
3.1.1	Mål	6
3.1.2	Undersøke behov	6
3.1.3	Design	6
3.2	Ferdig produkt	6
3.3	Fremdriftsplan	6
4	Tilbakeblikk	8
4.1	Oppnådde mål	8
4.1.1	Skjerm	8
4.1.2	Simulator	8
5	API	9
5.1	Hjelp til oppstart	9
5.1.1	Kode eksempler	9
5.2	Metoder	9
5.3	Protokoll mot mot simulator	10
5.4	Buffer	10
6	Veien videre	10
7	Oppsummering - evaluering av eget arbeid	10
8	Referanser	10

1 Tekniske spesifikasjoner

- 1 x Arduino mega 2560 microcontroller
- 1 x Arduino uno
- 4 x SNES controllers
- 1 x Atmega328P
- 8 modules of 8 pices of 8 x 8 bicolor LED dot matrix (delivered from sure electronics)
- Y x 2-channel multiplexer
- Y x 4-channel multiplexer
- 1 x DC-DC converter 12V/24V to 5V 15A
- 1 x ADP-70EB power supply 20V 3.5A
- 4 x Liquid Crystal displays
- Plexiglass

Figur 1 En 8x8 ledmatrise-modul.



Figur 1:

2 Presentasjon av gruppa

2.1 Deltakere og arbeidsfordeling

Arbeidsfordelingene sier bare hvem som har hovedansvaret for de ulike områdene. Alle på gruppen bidro de forskjellige stedene.

- **Stian Selbek**
Hoved ansvar for kobling av elektronikk. Også ansvar for protokoll mellom arduino skjerm, og implementasjon av det.
- **Eivind Wikheim**
Ansvarlig for protokollen mellom arduino og simulator, samt ansvarlig for materialer og innkjøp.
- **Martin Holøs**
Ansvarlig for interaksjon mellom bruker og brett, og implementasjonen av protokollen mellom kontrollene skjerm.
- **Abdimaalik**
Hovedansvar for brukerundersøkelser og design.
- **André Kramer Orten**
Ansvarlig for utvikling av software, med tanke på programmer og spill til skjermen, i tillegg til regissering av produktfilm.

2.2 Verktøy og software

Gjennom arbeidsprosessen har vi benyttet flere ulike verktøy. Disse verktøyene er listet opp under, med en liten beskrivelse av hvor i prosjektet de ble brukt.

- Github
For fildeling og versjonskontroll.
- AVR studios
For programmering av ATmega chippen.
- SONY Vegas
Filmeredigerings verktøy
- Arduino og Processing
Vi har bygd biblioteket vårt til arduino og simulator til processing som begge er open source.
- L^AT_EX
Benyttet til å skrive rapporten
- Project Planning Pro
Prosjekt planlegger for å strukturere arbeidsprosessen.

2.3 Arbeidsprosessen

Vi kom over en artikkel(REFERER til artikkel / fotnote!!). .

Vi startet prosessen med å gå rundt å snakke med forskjellige personer på ifi for å kartlegge behov. Fra denne prosessen kom vi fram til at vi vil satse på kategorien, formidling av IT, da dette er noe vi alle synes er spennende.

3 Valg av tema

(Game of light) er en ledmatriseskjerm. Den er satt sammen av 8 moduler som igjen er satt sammen av 8*8 to fargede leds. Dette gir totalt en oppløsning på $64 \times 64 \times 2 = 8192$.

3.1 prosjektets mål

Game of Light er laget i sammenheng med faget INF1510 (bruksorientert design) og fokuserer på formidling av IT. Prosjektets deltakere står listet nedenfor.

3.1.1 Mål

Målet med prosjektet er å skape interesse for elektronikk og programmering. Vi vil skape en utviklerplattform hvor man forbinder programmering og elektronikk. Plattformen skal være intuitiv og enkel å benytte. Det vil si at man ved å se i API'et over hvilke metoder som er tilgjengelig, skal man ved lite eller ingen programmeringsbakgrunn kunne skrive forskjellige ting til skjermen. Dette håper vi vil forbedre hver enkelts forståelse av programmering, og at de forhåpentligvis vil utvikle seg ved bruke av vår plattform.

Vår målgruppe vil i hovedsak være de med lite eller ingen tilknytning til elektronikk og programmering, men vi vil i noen av undersøkelsene benytte studenter med kjennskap til dette for å kartlegge kvaliteten av produktet.

MÅ MED: god og presis definisjon av hva som skal oppnås, og hvilke kriterier gruppa evaluerer det ferdige resultatet opp mot. Alle i prosjektgruppa har fått lære seg om alle elementene i prosjektet, men de har også utnyttet forskjellene i kompetanse.

3.1.2 Undersøke behov

MÅ MED: Gode undersøkelser av bruks-kontekst, gruppas metodevalg er begrunnet godt, flere metoder er benyttet. Gruppa har observert og intervjuet flere brukere enn nødvendig, og etablert samarbeid med en brukergruppe Sluttrapporten beskriver både metoder, data og analyse godt.

3.1.3 Design

MÅ MED: flere ideer har vært med i prosessen lenge, dokumentert i skisser. Design-resultatet kan forankres i undersøkelser av bruk + i Arduinos muligheter. Løsningen utnytter Arduino og kommuniserer visjonen gjennom materialer, funksjon og form. Gruppa har vært åpne og jobbet med minst to ulike ideer (til design-forslag). De har jobbet iterativt i dialog med brukere, og fått ny innsikt gjennom dette. Solid testing og evaluering (f.eks. DECIDE) med brukere. Gruppa har klart å forkaste egne ideer etter brukerevaluering.

D	Determine the goals
E	Explore the questions
C	Choose the evaluation methods
I	Identify the practical issues
D	Decide how to deal with the ethical issues
E	Evaluate, analyze, interpret and present the data

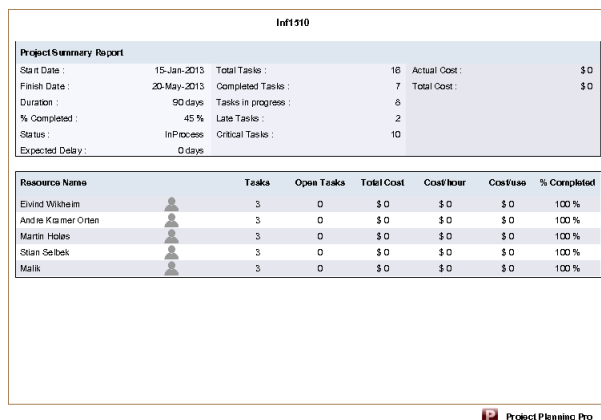
3.2 Ferdig produkt

MÅ MED: Nyskapende prosjekt der gruppa har tenkt ut av boksen. Gruppa har samarbeidet godt og alle har lært om alle sidene i prosjektet. Alle innleveringene er godt gjennomarbeidet og kommuniserer godt gruppas arbeid.

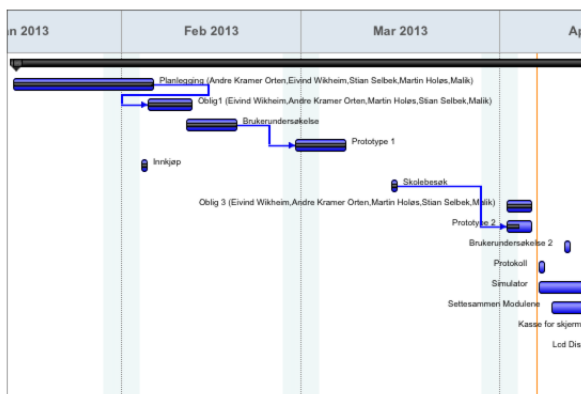
3.3 Fremdriftsplan

Figur 2

Figur 3 beskriver gruppens fremdriftsplan gjennom semesteret.



Figur 2:



Figur 3:

4 Tilbakeblikk

4.1 Oppnådde mål

4.1.1 Skjerm

4.1.2 Simulator

Etter de første brukerundersøkelsene kom det frem at det fantes et behov om at flere kunne tenke seg å lage spill og lignende til skjermen. Fra disse resultatene satte vi oss et mål om å lage en simulator og et bibliotek hvor man ikke trengte å ha stor kjennskap til programmering for å lage enkle ting til skjerm. Dette vil gjøre prosjektet åpent for bort i mot hvem som helst. Simulatoren bygger oppå processing IDE og arduino IDE som begge er open source. Biblioteket kan lastes ned på www.gameoflight.com og er open source (SIER SEG SELV SIDEN DET BYGGER OPPÅ ARDUINO OG PROCESSING). Vi har også laget et API over alle metoder tilgjengelig i biblioteket (REFERER TIL SENERE KAPITTEL).

5 API

Class GameOfLight

void	gotoXY (uint8_t index, const uint8_t line) Go's to position (index, line).
void	print (const char ch) Prints a given char from current position (x, y).
void	print (const char ch, const uint8_t colour) Prints a given char with a given colour (Red, green, orange) from current position (x, y).
void	print (char *string) Prints a given string to screen from current position (x, y).
void	print (const char *string, const uint8_t colour) Prints a given string to screen with a given colour (Red, green, orange) from current position (x, y).
void	setColour (const uint8_t colour) Sets the printing colour.
uint8_t	getPixel (const uint8_t x, const uint8_t y) Returns the color of the pixel in position (x, y).
void	setPixel (uint8_t x, uint8_t y, uint8_t val) Sets the pixel color of pixel (x, y).
void	clear () Clears the buffer by setting all elements to zero.
void	clear (int count) Clears count space ahead of the position (x, y). Stops once count has been cleared or screen overflows back to position (0, 0).
void	write (const uint8_t data) Write data to the current position of the buffer.

Class GameOfLightSim

void	update () Sends the buffer to the simulator.
void	update (const uint8_t line) Sends a single line to the simulator.
void	clearDisplay () Clears screen and buffer.
void	screen_cmd (uint8_t type, uint8_t value)
void	screen_data (uint8_t data) Write data directly to the display at current display position.
void	screen_goto (uint8_t index, uint8_t line) Go's to the position (index, line)

5.1 Hjelp til oppstart

5.1.1 Kode eksempler

5.2 Metoder

Her skal metoder etc for kode til skjermen.

5.3 Protokoll mot mot simulator

Sender 3-byte av gangen. Siste byte er \n som indikerer slutten av overføring

Type	Byte1								Byte2								Description
bits	b7	b6	b5	b4	b3	b2	b1	b0	b7	b6	b5	b4	b3	b2	b1	b0	bit
Burst	x	x	x	1	0	0	1	1	x	x	x	x	x	x	x	x	Burst data
Data	x	x	x	1	0	0	1	0	Data-byte								Send data
CMD	x	x	x	1	0	0	0	1	1	color	x-value						Cmd set color and X
	x	x	x	x	0	0	0	1	0	1	y-value						CMD set Y
	x	x	x	x	0	0	0	1	0	0	1	x	x	x	x	x	CMD clear display
	x	x	x	x	0	0	0	1	0	0	0	1	brightness				CMD global dim
	x	x	x	x	0	0	0	1	0	0	0	0	1	x	x	b1	Blink on/off

5.4 Buffer

Bufferet som inneholder hvilke LED som er satt høye. I bufferet er verdier med kolonne <64 Grønn og kolonne >63 rød. Når begge er høye vil dette gi oransj.

Byte	GREEN						RED				
0	0	1	62	63	64	65	126	127	
1	
2	
3	
4	
5	
6	
7	

6 Veien videre

7 Oppsummering - evaluering av eget arbeid

8 Referanser

Sonans utdanning.

v/ Tim dorfman.

Div artikler og lenker

<http://www.kidsakoder.no/2013/03/31/formidling-av-it-til-de-unge-ta-kidsa-til-neste-level/>

<http://techcrunch.com/2013/04/20/cognitive-overhead/>