

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgave i TDT4180 Menneske-maskin-interaksjon

Faglig kontakt under eksamen: Hallvard Trætteberg

Tlf.: 91897263

Eksamensdato: 31. mai

Eksamenstid (fra-til): 9.00-13.00

Hjelpemiddelkode/Tillatte hjelpemidler: D (Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt enkel kalkulator tillatt.)

Annen informasjon:

Oppgaven er utarbeidet av faglærer Hallvard Trætteberg og kvalitetssikret av Dag Svanæs.

Målform/språk: Bokmål

Antall sider: 3

Antall sider vedlegg: 1

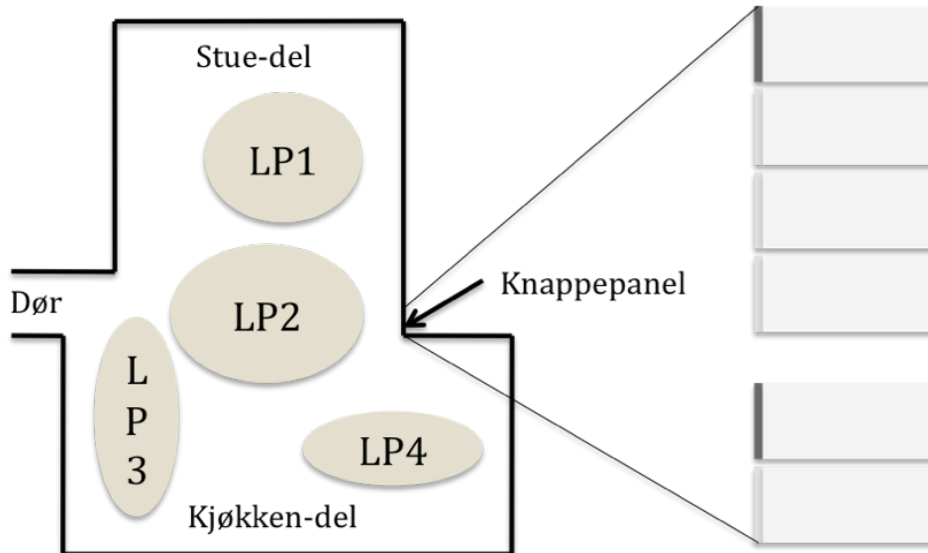
Kontrollert av:

Dato

Sign

Del 1 – Grensesnittdesign (30%)

Du ankommer en moderne leilighet som du har leid, og ønsker å slå på lysene i det kombinerte stue-kjøkken-rommet. Du finner et panel med knapper, men det er ingen tekst eller annen merking som forteller noe om funksjonen til panelet. Rommet er utformet som vist i figuren under, med knappepanelet forstørret til høyre. De fire lyspunktene er tegnet som ovaler tilsvarende området de lyser opp.



Stue-kjøkken-rommet, med lyspunktene LP1-LP4.

De seks knappene er gruppert i to grupper, fire i den øverste og to i den nederste. Knappene lar seg trykke litt inn, men spretter helt ut når en slipper. Hver knapp har en smal stripe til venstre som lyser opp i blått (vist som mørkegrått i figuren) når du trykker på knappen. Denne stripa forblir lysende når du slipper knappen, inntil en annen knapp i samme gruppe trykkes. I illustrasjonen over til høyre er det altså de to øverste knappene i hver gruppe som er sist trykket på.

Etter litt prøving og feiling finner du følgende sammenheng mellom knappene og de fire lyspunktene i kjøkken-stue-rommet: Hver av de fire knappene i den øverste gruppa tilsvarer ett lyspunkt. Når du trykker lenger enn ett sekund på en av disse knappene, så slår tilsvarende lyspunkt seg på/av. De to nederste knappene dimmer opp og ned så lenge knappen holdes inne. Lyspunktet som dimmes er det som tilsvarer knappen i øverste gruppe som lyser blått. Det nest nederste lyspunktet (LP3) er imidlertid et lysrør som ikke kan dimmes, så i det tilfellet har de to nederste knappene ingen effekt.

a) Kritiser (både positivt og negativt) designet av knappepanelet, med referanse til generelle prinsipper for god design.

b) Knappepanelet på veggen skal *erstatte* av en høyoppløselig og trykkfølsom fargeskjerm festet på samme sted, med størrelse og format som en smart-telefon, f.eks. iPhone. Skisser og forklar virkemåten til *to ulike* design for et slikt touch-panel, hvor hvert design begrunnes basert på kritikken (både positiv og negativ) i a). Vurder også de to designene opp mot hverandre.

Del 2 – Designprosess (30%)

a) Hva er forskjellen (formål og innhold) på *formativ* og *summativ* evaluering, og når brukes hver av dem i designprosessen?

Du jobber i et IT-firma og har laget to forslag til design for styring av lys i en leilighet basert på smart-telefon-formatet (tilsvarende oppgave 1b). Nå lurer utleiefirmaet på hvorvidt disse vil fungere som en app som kan kjøre på gjestenes egne smart-telefoner. Tanken er at lysanlegget skal styres via det trådløse nettverket i leiligheten og at det skal være mulig å styre det med både det eksisterende panelet og app-en kjørende på én eller flere telefoner.

b) Beskriv hvordan du vil bruke brukersentrerte metoder for å studere brukssammenhengen (context of use) for en slik lysstyrings-app.

c) Anta at det er besluttet at en slik lysstyrings-app kan ha noe for seg. Du får i oppgave å gjennomføre én iterasjon av en brukersentrert prosess, for å videreutvikle begge design-forslagene, for så å velge hvilken som skal tas videre. Forklar (og begrunn) hvordan du vil gjennomføre iterasjonen og vurderingen av forslagene.

Del 3 – Konstruksjon (40%)

De to design-forslagene fra 1b) vurderes som så likeverdige at det besluttes å lage en app (i Java og Swing) som *kombinerer* dem, ved at en kan bytte mellom dem når som helst uten å restarte app-en.

Anta at lysanlegget kan styres vha. (en klasse som implementerer) følgende Java-grensesnitt:

```
// controls a set of lights indicated by the String identifiers (lightId) "LP1"-“LP4”
public interface LightSystem {
    // returns the number of light levels/states the light with the given id supports
    public int getNumberOfLightLevels(String lightId);
    // returns the current level/state of the light with the given id (0 means off)
    public int getLightLevel(String lightId);
    // sets the current level/state of the light with the given id (0 means off)
    public void setLightLevel(String lightId, int level);
    // adds a listener that will be notified when one of the lights changes state
    public void addLightListener(LightListener lightListener);
    // removes a previously registered listener
    public void removeLightListener(LightListener lightListener);
}
```

a) Hva slags rolle vil en klasse som implementerer LightSystem-grensesnittet ha i en konstruksjon basert på MVC? Spesifiser ferdig LightListener-grensesnittet og gjør rede for evt. antagelser om virkemåte for både LightSystem og LightListener.

b) Forklar med diagrammer, tekst og kode hvordan du vil konstruere (GUI-delen av) app-en (som kombinerer begge design-forslagene fra 1b) i Java Swing, basert på MVC-arkitekturen, og som bruker LightSystem og LightListener. Se vedlegget (bakerst) for en oversikt over en del relevante klasser/grensesnitt. Dersom du trenger andre klasser, men ikke husker nødvendige detaljene, så forklar hvilke antagelser du gjør.

c) Forklar med tekst og sekvensdiagram hva som skjer når 1) app-en startes opp og 2) en slår på et lyspunkt og dimmer det ned.

d) App-en er ment å kunne styre de samme lyspunktene fra flere smart-telefoner samtidig. Forklar hvordan dette håndteres av din konstruksjon (du kan se bort fra evt. tidsforsinkelser i nettet osv.).

Department of computer and information sciences

Examination paper for TDT4180 Human-Computer Interaction

Academic contact during examination: Hallvard Trætteberg

Phone: 91897263

Examination date: 31. May

Examination time (from-to): 9.00-13.00

Permitted examination support material: D (No printed or hand-written material is allowed. Simple calculator is allowed.)

Other information:

The exam is written by teacher Hallvard Trætteberg, with quality assurance by Dag Svanæs.

Language: Bokmål

Number of pages: 3

Number of pages enclosed: 1

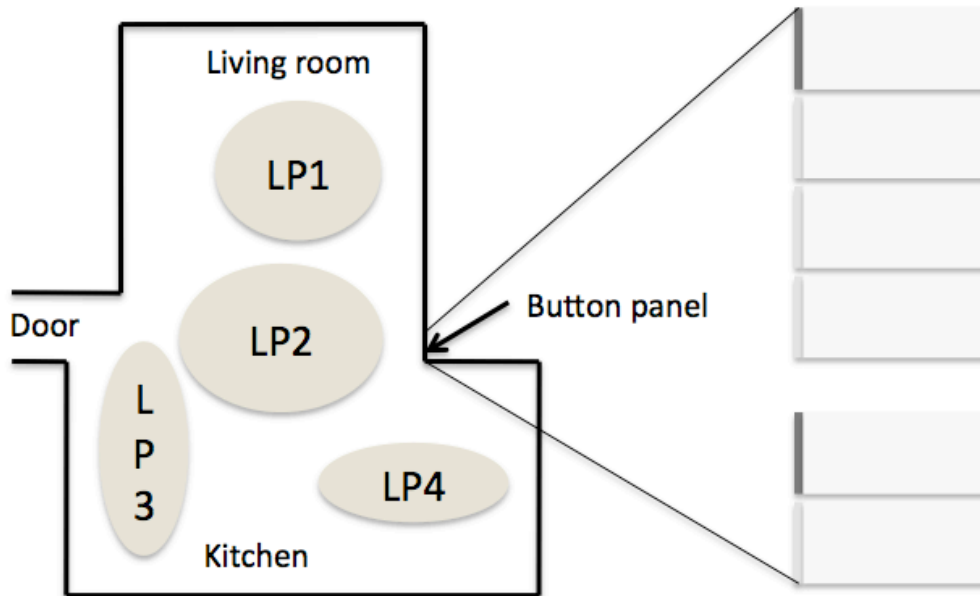
Checked by:

Date

Signature

Part 1 – User interface design (30%)

You arrive at a modern apartment that you have rented, and wish to turn the lights on in the combined living room and kitchen. You find a button panel, but there are no labels or other notes that explains how the panel operates. The room is designed as shown in the figure below, with an enlarged button panel shown on the right. The four lights are drawn as ovals corresponding to the areas they cover.



Living room and kitchen, with the lights LP1-LP4.

The six buttons are laid out in two groups, four in the upper and two in the lower. The buttons may be pressed, but do not stay in when released. Each button has a small strip on the left that shines blue (shown as gray in the figure) when you press the button. The strip stays on when you release the button until another button in the same group is pressed. In the figure above, the two top-most buttons in each group are the ones that were pressed most recently.

After some trial and error you find the following correspondence between the buttons and the four lights in the living room and kitchen. Each of the four buttons in the upper group corresponds to one of the lights LP1-LP4. When one of these buttons are pressed a second or longer, the corresponding light is turned on/off. The two buttons in the lower group dim a light up or down while pressed. The light that is dimmed is determined by the light in the upper button group that shines. However, the third light from above (LP3) is fluorescent and cannot be dimmed, so in that case the two bottom buttons have no effect.

- Critique (both positive and negative) the button panel design, with reference to design principles.
- The button panel on the wall is to be *replaced* by a high resolution colour touch screen attached, with size and format similar to a smart phone, e.g. an iPhone. Sketch and explain the behaviour of *two different* designs for such a touch panel, where you argue based on the critique (both positive and negative) in a). Compare and judge the two designs.

Part 2 – Design process (30%)

- What is the difference (purpose and content) between *formative* and *summative* evaluations, and when are they used in the design process?

You work in an IT company that has made two design proposals for controlling the lights in an apartment based on the smart phone format (as in part 1b). The rental company now considers how well these designs will work as apps running on the guests' own smart phones. The idea is to control the light system using the apartment's wireless network and that the lights may be controlled by means of both the existing button panel and the app running on one or more phones.

b) Describe how you would employ user-centred methods to study the *context of use* for such a light controller app.

c) Assume it is decided to develop the light controller app. You are assigned the task of performing one iteration of a user-centred process to develop both design proposals, before one is chosen for further development. Explain (and argue) how you will perform the iteration and evaluation of the proposals.

Part 3 – Construction (40%)

The two design proposals from 1b) are evaluated as equally good, so it is decided to build an app (using Java and Swing) that *combines* the designs, by allowing the user to switch between them at any time without restarting the app.

Assume the light system can be controlled using (a class that implements) the following Java interface:

```
// controls a set of lights indicated by the String identifiers (lightId) "LP1"-“LP4”
public interface LightSystem {
    // returns the number of light levels/states the light with the given id supports
    public int getNumberOfLightLevels(String lightId);
    // returns the current level/state of the light with the given id (0 means off)
    public int getLightLevel(String lightId);
    // sets the current level/state of the light with the given id (0 means off)
    public void setLightLevel(String lightId, int level);
    // adds a listener that will be notified when one of the lights changes state
    public void addLightListener(LightListener lightListener);
    // removes a previously registered listener
    public void removeLightListener(LightListener lightListener);
}
```

a) What role will a class that implements the `LightSystem` interface have in a construction based on MVC? Complete the `LightListener` interface definition and explain any assumptions made about the behaviour of both `LightSystem` and `LightListener`.

b) Explain with diagrams, text and code how you will construct the (UI part of the) app (that combines the two design proposals from 1b) in Java Swing, based on the MVC architecture, and that uses `LightSystem` and `LightListener`. See the appendix for an overview of some relevant classes/interfaces. If you need other classes, but don't remember the necessary details, explain the assumptions you make.

c) Explain with text and sequence diagrams what happens when 1) the app is started and 2) a light is turned on and dimmed.

d) The app is meant to control the same lights using several smart phones simultaneously. Explain how this is handled in your construction (you can ignore time delays in the network etc.).

Institutt for datateknikk og informasjonsvitenskap

Eksamensoppgåve i TDT4180 Menneske-maskin-interaksjon

Fagleg kontakt under eksamen: Hallvard Trætteberg

Tlf.: 91897263

Eksamensdato: 31. mai

Eksamenstid (frå-til): 9.00-13.00

Hjelpemiddelkode/Tillatte hjelpemiddel: D (Ingen trykte eller håndskrevne hjelpemidler tillatt. Bestemt enkel kalkulator tillatt.)

Annan informasjon:

Opgåva er utarbeidd av faglærer Hallvard Trætteberg og kvalitetssikra av Dag Svanæs.

Målform/språk: Nynorsk

Antall sider: 3

Antall sider vedlegg: 1

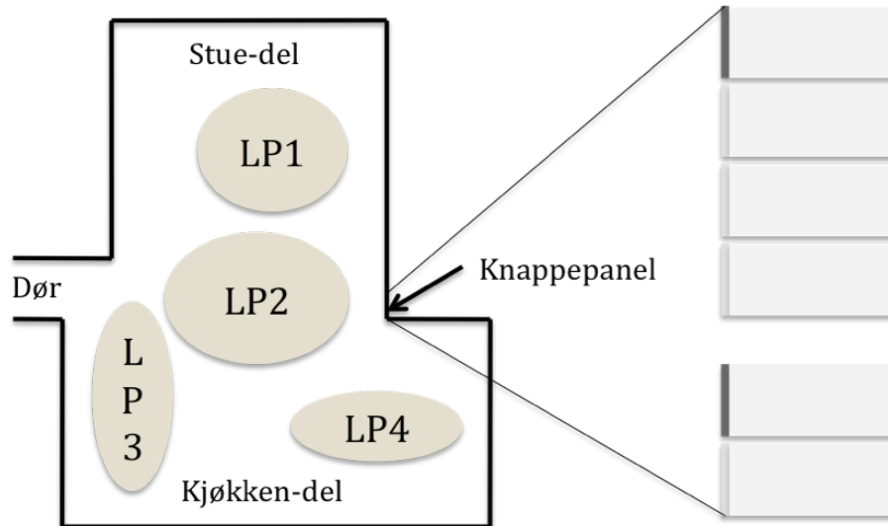
Kontrollert av:

Dato

Sign

Del 1 – Grensesnittdesign (30%)

Du kjem til ei moderne leilighet som du har leid, og ønskjer å slå på lysa i det kombinerte stue-kjøkken-rommet. Du finn eit panel med knappar, men det er ikkje tekst eller anna merking som fortel noko om funksjonen til panelet. Rommet er forma som vist i figuren under, med knappepanelet forstørret til høgre. Dei fire lyspunkta er teikna som ovalar som svarar til området dei lysar opp.



Stue-kjøkken-rommet, med lyspunkta LP1-LP4.

De seks knappane er gruppert i to grupper, fire i den øvste og to i den nedste. Knappane let seg trykke litt inn, men sprett heilt ut når ein slipp. Kvar knapp har ei smal stripe til venstre som lysar opp i blått (vist som mørkegrått i figuren) når du trykker på knappen. Denne stripa blir lysande når du slipp knappen, inntil ein annan knapp i same gruppe trykkast. I illustrasjonen over til høgre er det altså dei to øvste knappane i kvar gruppe som er sist trykt på.

Etter litt prøving og feiling finn du følgjande samanheng mellom knappane og dei fire lyspunkta i kjøkken-stue-rommet: Kvar av dei fire knappane i den øvste gruppa svarar til eitt lyspunkt. Når du trykker lenger enn eitt sekund på ein av disse knappane, så slår lyspunktet den svarar til seg på/av. Dei to nedste knappane dimmer opp og ned så lenge knappen heldast inne. Lyspunktet som dimmast er det som svarar til knappen i øvste gruppe som lyser blått. Det nest nedste lyspunktet (LP3) er eit lysrør som ikkje kan dimmast, så i det dømet har de to nedste knappane inga effekt.

a) Kritiser (både positivt og negativt) utforminga av knappepanelet, med referanse til generelle prinsipp for god design.

b) Knappenepanelet på veggen skal *erstattast* av ein høgoppløseleg og trykkjenslevar fargeskjerm festa på same stad, med storleik og format som ein smart-telefon, t.d. iPhone. Skisser og forklar virkemåten til *to ulike* design for eit slikt touch-panel, kor kvart design grunnjevnes basert på kritikken (både positiv og negativ) i a). Vurder også dei to utformingane opp mot kvarandre.

Del 2 – Designprosess (30%)

a) Kva er skilnaden (formål og innhald) på *formativ* og *summativ* evaluering, og når nyttast kvar av dei i designprosessen?

Du jobbar i eit IT-firma og har laga to forslag til design for styring av lys i leiligheten basert på smart-telefon-formatet (svarande til oppgåve 1b). Nå lurar hyttefirmaet på om disse vil fungere som ein app som kan kjøre på gjestenes egne smart-telefonar. Tanken er at lysanlegget skal styrast via det trådlause nettverket i leiligheten og at det skal være mogleg å styre det med både det eksisterande panelet og app-en kjørande på ein eller fleire telefoner.

b) Beskriv korleis du vil nytta brukersentrerte metodar for å studere brukskonteksten (context of use) for ein slik lysstyrings-app.

c) Anta at det er slutta at en slik lysstyrings-app kan ha noko føre seg. Du får i oppgåve å gjennomføre ein iterasjon av ein brukersentrert prosess, for å utvikle vidare båe forslaga til utforming, for så å vele kva for ei som skal tas vidare. Forklar (og grunngje) korleis du vil gjennomføre iterasjonen og vurderinga av forslaga.

Del 3 – Konstruksjon (40%)

De to design-forslaga frå 1b) vurderast som så likeverdige at det sluttast å lage ein app (i Java og Swing) som *kombinerer* dei, ved at en kan bytte mellom dei når som helst utan å starte app-en opp att.

Anta at lysanlegget kan styrast vha. (ein klasse som implementerar) fylgjande Java-grensesnitt:

```
// controls a set of lights indicated by the String identifiers (lightId) "LP1"-“LP4”
public interface LightSystem {
    // returns the number of light levels/states the light with the given id supports
    public int getNumberOfLightLevels(String lightId);
    // returns the current level/state of the light with the given id (0 means off)
    public int getLightLevel(String lightId);
    // sets the current level/state of the light with the given id (0 means off)
    public void setLightLevel(String lightId, int level);
    // adds a listener that will be notified when one of the lights changes state
    public void addLightListener(LightListener lightListener);
    // removes a previously registered listener
    public void removeLightListener(LightListener lightListener);
}
```

a) Kva for rolle vil ein klasse som implementerar LightSystem-grensesnittet ha i ein konstruksjon basert på MVC? Spesifiser ferdig LightListener-grensesnittet og gjer reidde for evt. antakingar om virkemåte for både LightSystem og LightListener.

b) Forklar med diagrammer, tekst og kode hvordan du vil konstruere (GUI-delen av) app-en (som kombinerer båe design-forslaga fra 1b) i Java Swing, basert på MVC-arkitekturen, og som nyttar LightSystem og LightListener. Se vedlegget for ein oversikt over ein del relevante klasser/grensesnitt. Dersom du treng andre klasser, men ikkje hugsar naudsynte detaljane, så forklar kva for antakingar du gjer.

c) Forklar med tekst og sekvensdiagram kva som skjer når 1) app-en startast opp og 2) når ein slår på eit lyspunkt og dimmar det ned.

d) App-en er ment å kunne styre dei same lyspunktta frå fleire smart-telefonar samstundes. Forklar korleis dette handterast av din konstruksjon (du kan sjå bort frå evt. tidsforsinkingar i nettet osv.).

Appendix

JLabel

[JLabel](#) () - Creates a JLabel instance with no image and with an empty string for the title.

[setText](#) (String text) - Defines the single line of text this component will display.

[setIcon](#) (Icon icon) - Defines the icon this component will display

JButton

[setText](#) (String text) - Defines the single line of text this component will display.

[setIcon](#) (Icon icon) - Defines the icon this component will display

[setEnabled](#) (boolean b) - Enables (or disables) the button.

[addActionListener](#) (ActionListener l) - Adds an ActionListener to the button.

ActionListener

void [actionPerformed](#) (ActionEvent e) - Invoked when an action occurs.

JSlider

[JSlider](#) (int min, int max, int value) - Creates a horizontal slider using the specified min, max and value.

int [getValue](#) () - Returns the slider's current value.

void [setValue](#) (int n) - Sets the slider's current value to n.

void [addChangeListener](#) (ChangeListener l) - Adds a ChangeListener to the slider.

ChangeListener

void [stateChanged](#) (ChangeEvent e) - Invoked when the target of the listener has changed its state.

JSpinner

[JSpinner](#) (SpinnerModel model) - Constructs a spinner for the given model.

[getValue](#) () - Returns the current value of the model, typically this value is displayed by the editor.

[setValue](#) (Object value) - Changes current value of the model, typically this value is displayed by the editor.

void [addChangeListener](#) (ChangeListener l) - Adds a ChangeListener to the spinner.

SpinnerNumberModel (implements SpinnerModel)

[SpinnerNumberModel](#) (int value, int minimum, int maximum, int stepSize)

Constructs a SpinnerNumberModel with the specified value, minimum/maximum bounds, and stepSize.

JComboBox

[JComboBox](#) (Object[] items) - Creates a JComboBox that contains the elements in the specified array.

int [getSelectedIndex](#) () - Returns the index of the current selected item.

Object [getSelectedItem](#) () - Returns the current selected item.

[addActionListener](#) (ActionListener l) - Adds an ActionListener.