TUUUT TUUUUT

Thomas Ax Gulbrandsrød

Kasper Chan

TUUUT TUUUUT

Læringsapplikasjon



Innhold

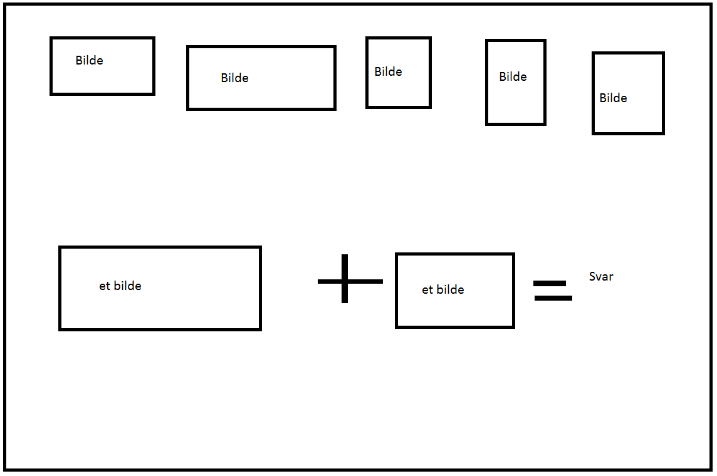
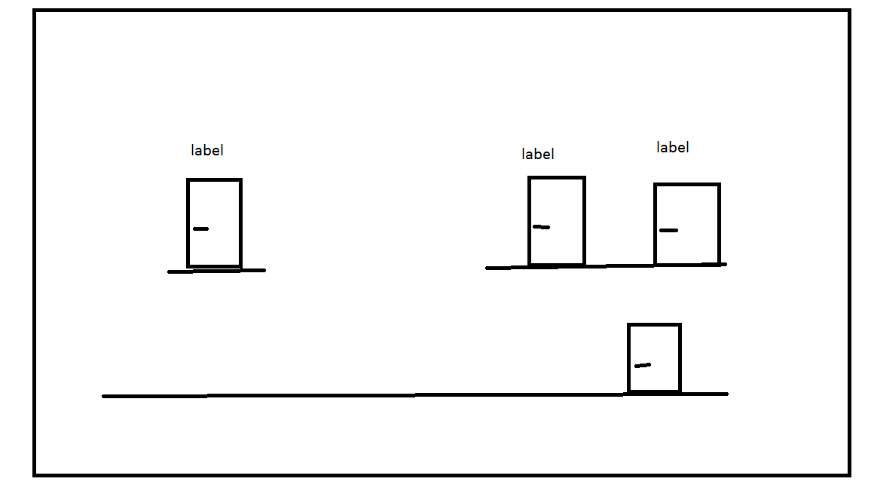
[Kravspesifikasjon 2](#_Toc476056426)

# Kravspesifikasjon

Vi ønsker å lage en interaktiv applikasjon som minner om et spill, hvor en karakter (picturebox) kan styres med tastaturet, og gjennom dette spillet kan forskjellige oppgaver aktiveres, for eksempel dører som man åpner. Disse oppgavene skal lære brukeren å koble bilder til forskjellige dataspill.

Applikasjonen krever mus- og tastatur input. Vi skal bruke dra- og slippfunksjoner, samt lyd og visuell feedback til det brukeren gjør i applikasjonen. I tillegg skal brukeren slå en ‘boss’ til slutt, hvor brukeren må trykke rett knapp til riktig tid for å vinne. Vanskelighetsgraden baseres på oppnåelsen i de tidligere oppgavene, slik at det blir enklere dersom du hadde alt riktig.

GUI er et område hvor karakteren kan bevege seg, og det er dører som kan åpnes, i stil med Super Mario. Når man «går inn» i dørene kommer man til oppgaver som skal løses. Her skal bilder dras til riktig plass for å komme videre. Poengscore (label) og fremgang i spillet (progressbar) står alltid på skjermen.

Applikasjonen skal ha en størrelse på 1280x720 piksler slik at den passer inn på alle skjermer. Bildene skal være 180x100 piksler. Dørene er like store, men rotert 90. Karakteren er ¾ stor som dørene.

# System og program design

Alle metoder skal kunne være så modulære som mulig. Det vil si at det skal brukes argumenter når man kaller metoden.

I hovedmenyen må det være en dør som skal hete pctbxDoorn (n gjelder for alle dører starter fra 1 og oppover) med korresponderende label som passer: lblDoorn. Dette gjøres for alle dørene i menyen. Det skal være en begynn dør, en exit dør og en settingsdør.

Når man er i spill: da skal det finnes 5 bilder å velge mellom på skjermen. De skal hete bilde1, bilde2 og videre. Området man legger svarene sine i skal hete answer1 og answer2. Svaret skal hete result.

Metoder:

1. Movement(), Interaction(). Bevegelse og interaksjoner i meny. Denne metoden skal utføre bevegelsene til karakteren du styrer mens du er i menyen. Trenger flere elementer:
   1. Sjekke om du har beveget deg
   2. Sjekke om du får lov til å flytte deg
      1. Om man er utenfor menyen
      2. Å sjekke om man kommer til å krasje med noe
2. Question(). Finne fram et spørsmål og bilder som hører til spørsmålet
   1. Henter det ut fra klassen som inneholder bildene og spørsmålene
3. Drag() og Drop(). Drag and drop metoder for å sjekke om objektet er blir sluppet ned på riktig plass og om ikke sette den tilbake til der den kom fra
4. Disperse().Metode for å legge ut alle bildene og sette opp spørsmålet. Kanskje pushe alle bildene inn i en list og legge alle elementene ut på spill-området
5. Initialization(panel). En initialiseringsmetode som plasserer ut alle elementene til hovedmenyen.

Klasser:

1. Bricks. En klasse som det skal lages en instanse av når spille starter. Den skal brukes til å holde på alle spørsmålene og bildene som skal brukes til hvert enkelt spørsmål
2. Information. Spill informasjonsklasse. Denne klassen skal passe på all spillinformasjonen. Alle verdier og hvordan de skal brukes og kan hentes ut og bli brukt andre steder i spillet. Informasjon er poeng, hvilket nivå du er på, hvor langt inn i en rundt du er.

Metode og klassefordeling:

Thomas: metode 2 og 3. Klasse 1.

Kasper: metode 1 og 4. Klasse 2.

# Implementasjon

La til mange ekstrametoder.

Thomas:

Form1\_MouseDown

For å flytte på objekter

Form1\_MouseMove

For å flytte på objekter

Form1\_MouseUp

For å flytte på objekter

Kasper:

AddMenuItems

Legge til elementer i menuPanel

AddGameItems

Legge til elementer i questPanel

Endring av oppgavefordeling: Thomas lagde begge klassene og Kasper fokuserte på GUI og Movement().