

Videregående Programmering (02324)

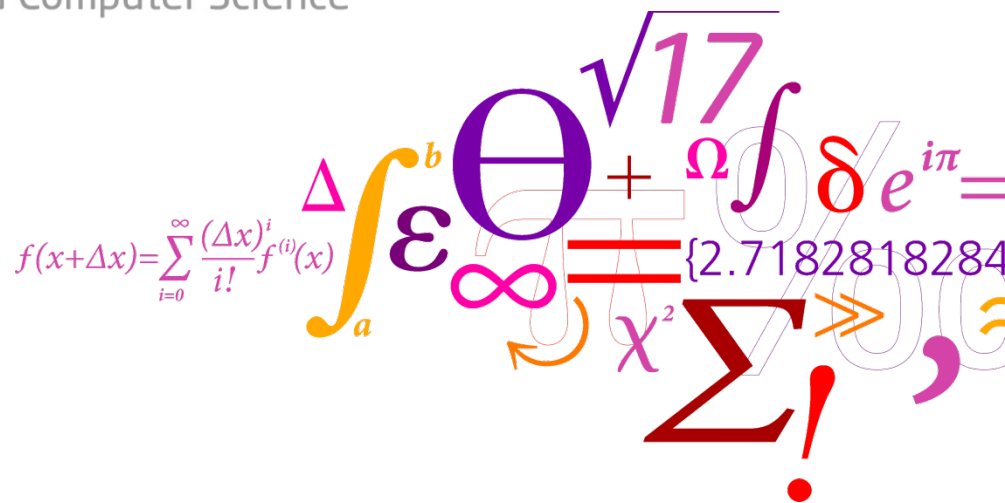
Projekt i software-udvikling (02362)

Forår 2021

Ekkart Kindler

DTU Compute

Department of Applied Mathematics and Computer Science



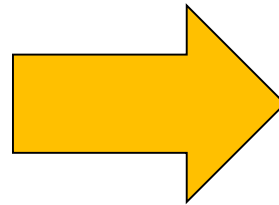
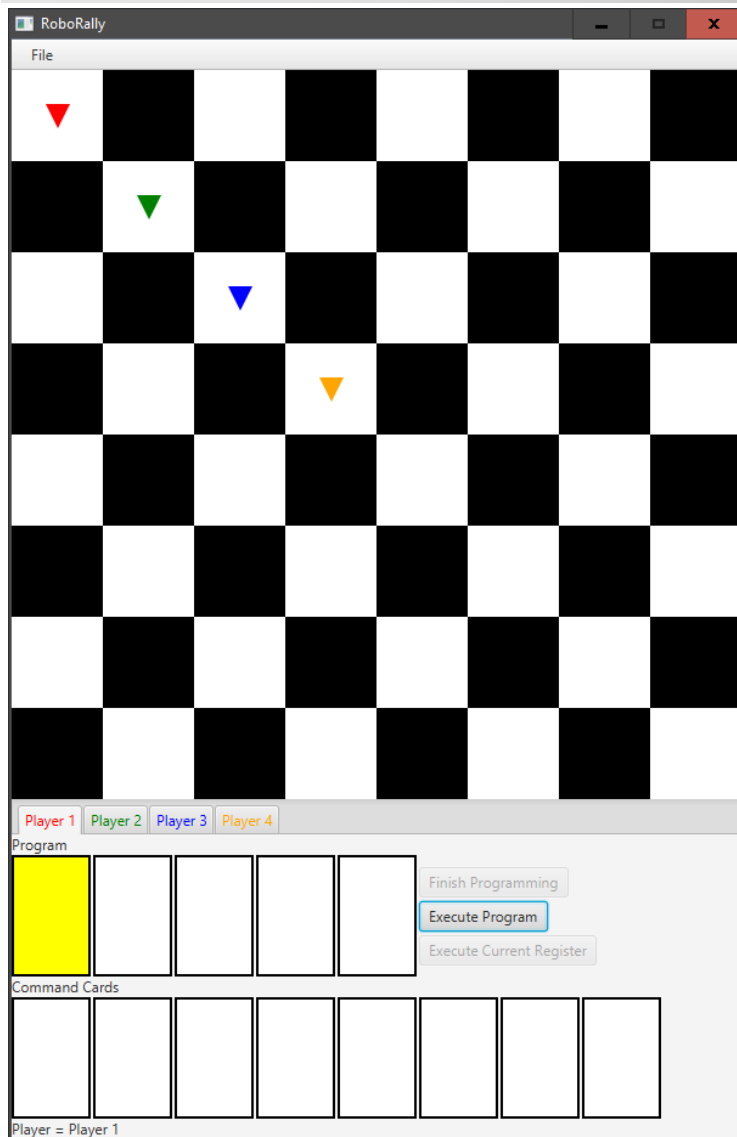
Projekt

DTU Compute

Department of Applied Mathematics and Computer Science

$$f(x+\Delta x)=\sum_{i=0}^{\infty}\frac{(\Delta x)^i}{i!}f^{(i)}(x)$$
$$\int_a^b \varepsilon \Theta^{\sqrt{17}} + \Omega \int \delta e^{i\pi} = \{2.7182818284\}$$
$$\chi^2 \Sigma ! > ,$$

- Er et spil, hvor hver spiller programmerer sin robot med kommandokort. Hvorefter, robotterne helt automatisk bevæger sig efter programmet i den fabrik (spilleplade) de bevæger sig i.
- Kommandokort siger noget om hvordan robotten flytter sig og drejer rundt få felterne i fabrikken.
- Felterne i fabrikken og andre robotter har også indflydelse på ens robot (flytter den, skyder på den) og der kan også være væg, så at robotten ikke kan flytte sig som programmeret.
- Formålet er at ens robot når alle "checkpoints" i den rigtige rækkefølge som den første.



I skal implementere **RoboRally** så at

- en **betydelig mængde af spillets regler** er realiseret,
- man skal kunne **gemme** (flere) **spil** i en database, så at man kan fortsætte disse spil senere,
- alle **spillets informationer vises grafisk** på en hensigtsmæssige måde,
- evt. kan gengive et spils forløb og
- evt. kan vælge (og selv definere) nogle spilleplader
- ...

Spilletets regler finder man på:

- <https://avalonhill.wizards.com/games/robo-rally>
- http://media.wizards.com/2017/rules/roborally_rules.pdf

At implementere RoboRally med alle dens regler og detaljer ville være meget ambitiøst! Derfor består kunsten i at udvælge features og reglerne fornuftig. Det taler vi om under kursets forløb.

- Fokus skal ikke være på den mest brugervenlige og grafisk tiltalende brugergrænseflade (GUI). Men GUIen skal vise spillets tilstand og køre stabilt.
- Man kan udvide "brættet" fra den første (V1) eller anden (V2) opgave trinvis.
- Til gengæld skal softwaren have en klar struktur:
 - adskille model (spillets tilstand), view (GUI) og kontrol
 - god design af API (model og kontrol)
 - klart og enkelt interface mellem database og selve software
 - være nemt at udvide (især når I ikke når at implementere alle regler; og til 02324 overføres til en webapplikation)

I skal finde og argumentere for jeres prioritering af implementerede regler (eller regelområder):

- forskellige programmeringskort
(som inkluderer interaktive kort → se V3)
- forskellige fabrikkelter som interagerer med robotten
- evt. mulighed for at opgradere og lade robotten
- ...

Jeres prioritering skal garantere at det giver mening at spille spillet og spillet kan slutte med en vinder

- Det ville give mest mening, at spille RoboRally online!

Men det skal I ikke implementere i 02362!

Det er et formål i 02324 (web-programmeringsdel)

- Sofistikerede GUI (drag and drop, etc. Ud over det I fik); men særlige felter, skal markeres på spillepladen

Vi diskuterer det nærmere sammen med domæneanalysen og aflevering af projektdefinition.
Og vi taler om hvordan man kan realisere det i kursets forløb.

- Endelige aflevering af rapport og software:

Dato til aflevering bliver aftalt senere. Men det bliver sandsynligvis i kalenderuge 20.

- Mundtlig eksamen:

Projektpræsentation som gruppe (ca. 15min)

Individuelle eksamen efterfølgende (ca. 10min)

Dato til præsentationer/mundtlige eksaminer er planlagt midt i maj (det bliver aftalt senere).

- Endelige aflevering af rapport og software:

Dato til aflevering bliver aftalt senere. Men det bliver sandsynligvis i kalenderuge 24 (3-ugers-periode juni).

- Mundtlig eksamen:

Projektpræsentation som gruppe (ca. 15min)

Individuelle eksamen (ca. 10min)

Dato til præsentationer/mundtlige eksaminer er planlagt i uge 25 i 3-ugers-periode juni.

Det taler vi nærmere om
i kursusuge 4
(kalenderuge 8).

■ Introduktion

- Kort overblik over projektet og hovedpunkter, som opgaven indebærer
- Overblik over selve rapporten og hvordan den skal læses

■ Problembeskrivelse

- Kontekst og baggrund
- Overblik over hovedfunktioner (i grupper) med prioritering og argumentation for denne (kan fx bruge MoSCoW-kategorier: Must, Should, Could, Won't)
- Husk også at nævne, hvilke data, der skal gemmes permanent (persistence)
- Hvad kan I bygge videre på (fx GUI)

■ Kravspecifikation (Analyse)

Beskrivelse af krav fra brugerens perspektiv

- Beskriv begreber, funktioner og use-cases (som brugeren opfatter dem)
- Beskriv de relevante informationer som domænemodel:
 - Klassediagrammer
 - Aktivitetsdiagrammer
 - evt. tilstandsdiagrammer
- Beskriv ikke-funktionelle krav: fx brugbarhed (usability), vedligeholdbarhed (maintainability), ...
- Evt. UC-diagram med beskrivelse af de vigtigste use-cases

Alle diagrammer skal
forklares i teksten!

I må gerne genbruge jeres eget materiale fra
de andre afleveringsopgaver igen til jeres
egen rapport! Vi arbejder inkrementel; og det
gælder software og rapporten!

- Database- og softwaredesign
- Overblik over de vigtigste komponenter, som inkluderer, også tilkobling af databasen: arkitektur
 - Software design: hovedkomponenter af software og deres sammenspil
 - Klassediagrammer (indeholder især de vigtigste model-, kontroller- og view-klasser)
 - Sekvensdiagrammer som viser samspil mellem vigtige klasser/komponenter
 - Database design → se DB kursus

Husk at alle diagrammer skal forklares i teksten!

- Implementering
 - Hvordan er designet bliver implementeret (men kun nogle interessante udtræk, ikke hele koden)
 - Dette omfatter software og database
 - Herunder kan I også diskutere evt. mangler af jeres implementering
- Udviklingsproces og test
 - Udviklingsproces og brugte værktøjer
 - Diskussion af JUnit-test (automatisk)
 - Diskussion af acceptance-test (manuelle test, fx use-cases)

Det skal være muligt
at installere og bruge
softwaren uden
anden information!

■ Håndbog

- Hvordan installerer man selve software
- Hvordan starter man softwaren
- Hvordan bruger man software (den eksisterende GUI er I ikke nødt til at diskutere med alle detaljer); evt. med nogle screenshots

■ Konklusion

- Opsamling af hele resultatet
- Nogle afsluttende bemærkninger

- Referencer
 - Litteratur og
 - Websider I har brugt
- Appendiks (evt.)
 - Nogle relevante diagrammer som ikke kunne være med i hoveddelen
 - Alle use-cases, måske med aktivitetsdiagrammer til de vigtigste regler/spille-faser
 - Glossar/taksonomi (som I ville have i forvejen gennem jeres domæneanalyse)