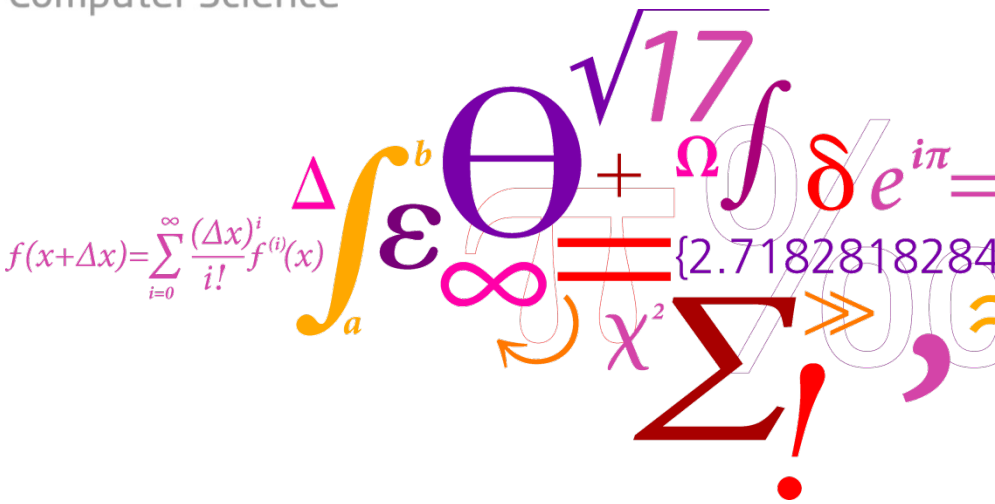
Projekt i software-udvikling (02362)
Forår 2021

Ekkart Kindler

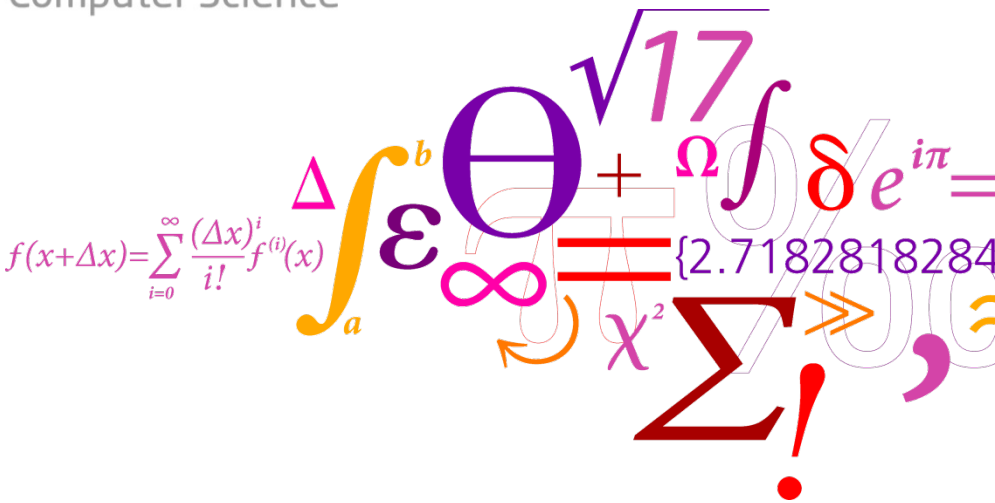
DTU Compute
Department of Applied Mathematics and Computer Science



Projekt

Slides fra uge 1 (men her kun til 02362)

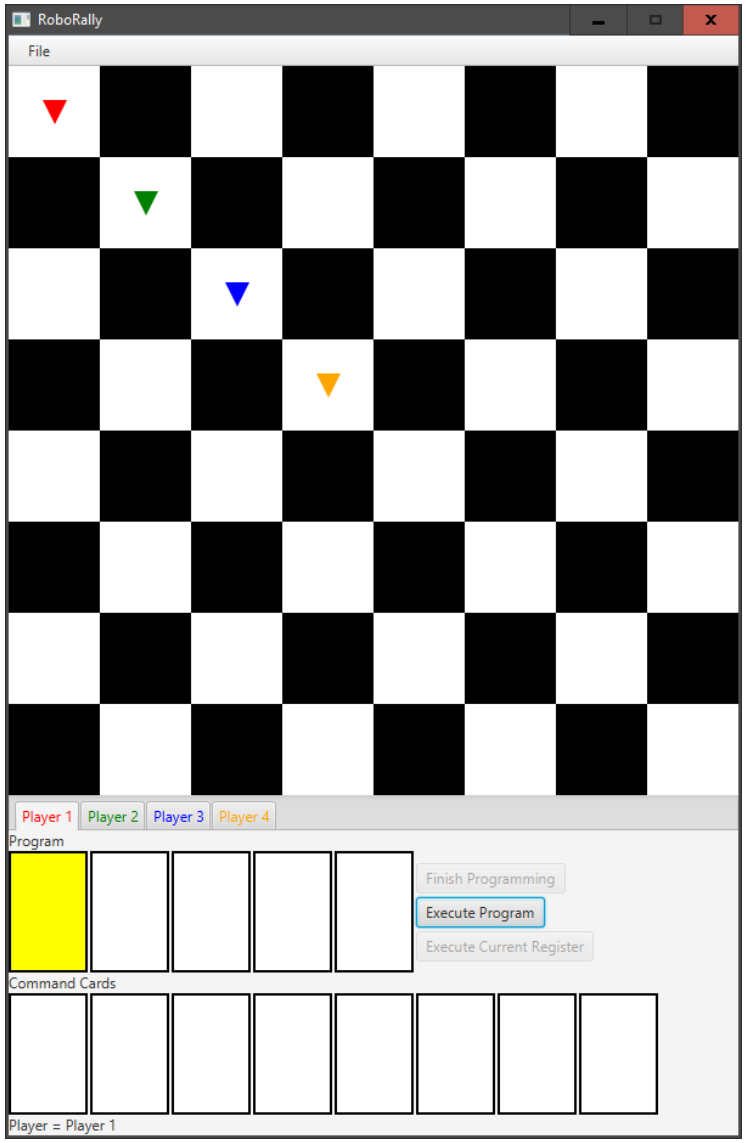
DTU Compute
Department of Applied Mathematics and Computer Science

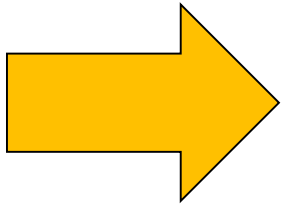


RoboRally

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- 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.


RoboRally
File
Player 1
Player 2
Player 3
Player 4
Program
Finish Programming
Execute Program
Execute Current Register
Command Cards
Player = Player 1



?

Opgave

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

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
- …

RoboRally

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

Spillets 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 reger og detaljer ville være meget ambitiøst! Derfor bestå kunsten i at udvælge features og reglerne fornuftig. Det taler vi om under kursets forløb.

Realisering

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- 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)

Støttede regler

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

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

- forskellige programmeringskort (som inkluderer interaktive kort → se V3)
- forskellige fabrikfelter som interagerer med robotten
- evtl. 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

Out of scope

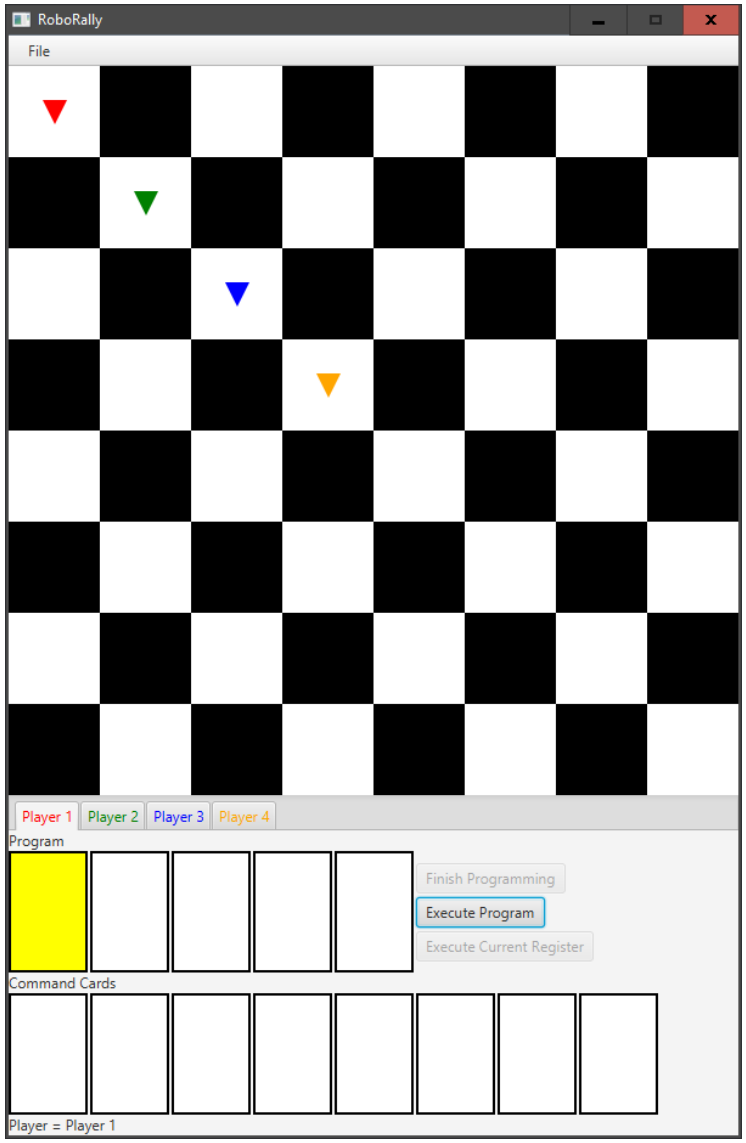
DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

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

Men det skal I ikke implementere i 02362!

- 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.



?

Hvilke features skal der være med ud over det som I fik udleveret i starten?

Formelle krav

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- De næste slides præsenterer minimale krav vedr. Features af jeres software
- Men I skal også huske tekniske og formelle krav til aflevering (der kommer en tjekliste på kursets websider):
  - softwarens arkitektur (MVC, DAL)
  - author tags
  - Javadoc kommentarer
  - test til vigtig spillelogik
  - …

Vigtigt til eksamen: **alle gruppens medlemmer** har sat sig ind i softwaren og kan forklare dens struktur og nogle detaljer (især til MVC, DAL, …)

Minimumskrav vedr. features

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

- Gemme spil i databasen og lade spil igen fra databasen
  - spillerens håndkort og program skal også gemmes
  - det skal også gemmes hvilken spilleplade spillet var på
- Spilleplader er defineret gennem JSON-filer (mindst to)
- Interaktive spillekort
- Vægge på spillepladen
- Mindst to feltaktioner
- Checkpoints

Selvfølgelig skal også featurens logik implementeres og features kan skulle ses på spillepladen.

Features til lidt højere niveau

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

For at komme lidt højere op skal der være flere features (fx.):

- eksplicitte startfelter til robotter
- flere feltaktioner (af forskellig art)
- damage-kort
- feltaktioner bliver udført på det rigtige tidspunkt i spillets forløb (se reglerne)
- lidt finere grafik
- spil kan afsluttes fornuftigt

Alle informationer om spillets aktuelle situation skal kunne ses på spillepladen!

Men prioriter ikke mængden af features over softwarens kvalitet (især stabilitet)

Avancerede features

DTU Compute
Department of Applied Mathematics and Computer Science
Ekkart Kindler

For at komme endnu længere (fx.):

- antennen (som afgør rækkefølgen robotternes registre og feltaktionerne bliver udført)
- engergy cubes, pits, reboot
- upgrade cards

Bemærk at nogle upgrade cards har meget avancerede logik. I er ikke nødt til at have alt det mest avancerede med.

- Hver spiller har deres eget kortdæk
- pænere grafik
-

Også her: prioriter ikke mængden af features over softwarens arkitektur og kvalitet (især stabilitet)

- … eller andre ”coole” features