

02362: Projekt i software-udvikling (f23)

Ekkart Kindler (ekki@dtu.dk), 6. februar 2023

Denne opgave skal fremme forståelsen af projektet (især begreberne fra RoboRally-spillet) og hvordan programmeringen af en robot med såkaldte kommandokort kan lade sig gøre.

Opgaven består i to dele:

1. Den første del er en **afleveringsopgave (A1: 21. februar)**, som er at færdiggøre taksonomien af RoboRally-spillet fra opgaven V1 og videreudvikle den til en detaljeret (!) domænemodel, som indeholder alle begreber af RoboRally samt alle vigtige faser og aktioner af spillet.
2. Den anden del er en **visningsopgave (V2: 14. februar)**; den skal fremme forståelse af projektets opbygning (MVC-princippet) og sammenhæng mellem viewet og controlleren. Vedr. softwaren skal denne opgave fortsætte med den software I fik udleveret med opgave V1.

Vi starter med at beskrive opgaven V2; derefter beskrives opgaven A1.

Opgave V2 (til 14. februar): Programmering: Controller & View

Opgaven er nedbrudt i enkle trin:

1. Til opgaven skal I bruge koden som I fik udleveret med opgave V1 (DTU Learn, uge 1). Nogle af de dele af koden som er relevant til opgave V2, er markeret med kommentarer `XXX: V2` eller `TODO Assignment V2`.
2. Når et nyt spil startes i denne kode, bliver spillets programmeringsfase aktiveret, hvor hver spiller får tildelt nogle tilfældige kommandokort (på den nederste linje af kortfelterne). Og spillerne kan trække nogle af dem til de fem kortfelter på den øverste linje, som repræsenterer denne spillers robots program (register).

Når alle spiller har programmeret deres robot, skal man kunne trykke på "Finish Programming".

Men der vil ikke ske noget. Grunden til det er at knapperne "Finish Programming", "Execute Program" og "Execute Current Register" i `PlayerView` stadigvæk er tilknyttet til controllerens metode `notImplemented()`. Men de burde været tilknyttet til de tilsvarende metoder i `GameController`.

Som det første, skal I så sørge for at alle knapper i GUI'en (view) bliver tilknyttet til de respektive metoder i `GameController`.

Så burde det være sådan at knapperne "Execute Program" og "Execute Current Register" bliver aktiveret, så snart man trykker på "Finish Programming". Når man så trykker på "Execute Program" bliver alle programmer af alle robotter eksekveret. Når man trykker "Execute Current Register" bliver kun den aktuelle robots aktuelle kort eksekveret; på

denne måde kan man klikke sig trinvis igennem hele programmet. Og det anbefales at bruge denne trinvis eksekvering i starten.

Når man trykker "Finish Programming" og "Execute Current Register" nu, så bliver der vist hvilket kort bliver eksekveret; men selve kommandoer har ingen effekt. Grunden til det er at de metoder, som realiserer de forskellige kommandoer i GameController, ikke er implementeret endnu (de er markeret `TODO Assignment V2`). Dem skal I implementere i næste skridt.

3. Implementer de fire kommandoer som programmet støtter indtil videre: *Move Forward*, *Fast Forward*, *Turn Left* and *Turn Right*. Dertil skal I implementere de fire metoder i GameController som svarer til disse kommandoer (som sagt er de markeret med `TODO Assignment V2`). Bemærk at model-klasserne (især klasse Board) har en masse metoder som kan hjælpe med det, så at der ikke er en hel masse kode I skal skrive selv.

Når I er færdige med at implementere de der fire metoder, afprøv om jeres software nu virker og robotterne udfører deres programmer korrekte.

4. Implementere en test til hver af de fire kommandoer som programmet støtter indtil videre. Til det kan I bruge den eksisterende klasse GameControllerTest og implementere en ny test-metode til hver kommando (der findes allerede en test til `moveForward` — dog i Java-kommentarer. Den kan I bruge).

Eksekver alle test. Når alle disse test kører igennem, så er opgaven V1 og V2 klaret. Denne løsning skal vises til underviseren i uge 3 (14. februar).

I kan finde mere information om Testing med IntelliJ på JetBrains web sider:

<https://www.jetbrains.com/help/idea/create-tests.html>

Opgave A1 (til 21. februar): Projektforståelse: Taxonomie & domænemodel

Opgaven nedbrudt i enkle trin:

1. Til den 21. februar (husk at det er altid kl. 12, hvis der ikke bliver sagt noget andet) skal I aflever en komplet taksonomi og en detaljeret domænemodel for spillet RoboRally, som fanger alle relevante begreber og aktiviteter. Det er relevant for at forstå hvad (WHAT) jeres software skal gøre.

Informationen om RoboRally samt spillets regelsæt findes som PDF-fil:

http://media.wizards.com/2017/rules/roborally_rules.pdf. Som udgangspunkt burde I allerede have en liste med de vigtigste begreber fra opgave V1; og en del bliver også diskuteret under timen.

2. Læs teksten systematisk (igen) og marker alle relevante ord. Klassificer dem som ting som bliver brugt til at repræsentere spillets aktuelle tilstand og dem som beskriver spillets aktioner. Typisk er den første slags begreb navneord (engelsk: noun) og den anden slags begreb udsagnsord (engelsk: verb).

Den første slags begreber med deres egenskaber og relationer kan man overføre til en domænemodel (repræsenteret som klassesdiagram). Den anden slag begreber udgør typisk softwarens use cases (som senere hen kan beskrives som aktivitetsdiagrammer). Det bliver også diskuteret i timen.

3. Diskuter jeres begreber indbyrdes i grupperne, så at de dækker alle RoboRallys væsentlige begreber. Derefter skal I lave et domænemodel (gerne diskuter og tegn indbyrdes på en tavle eller på et stykke papir). Som inspiration kan I måske også kigge ind i softwaren, som I fik udleveret til opgave V1; klasserne i model-pakken burde jo også være relevante til RoboRally spillet (men det er jo kun begyndelsen).

Aflever jeres taksonomi og jeres domænemodel som PDF-fil via DTU Learn som A1.