

Obligatorisk oppgave 1: Prosjekt RoboRally - oppstart

Dette semesterets programmeringsprosjekt er å lage en digital versjon av brettspillet RoboRally. I RoboRally plasseres en serie flagg ut på spillebrettet, og så er det spillernes oppgave å navigere brettet for å komme til alle flaggene raskest mulig. Selve navigeringen gjøres ved å legge programkortene hver spiller får utdelt i ønsket sekvens, "kompilere" sekvensen og la robotene rundreis utføre instruksjonssettet.

Regelboken finner du [her](#).

Spillet skal skrives i Java, med spillmotoren [LibGdx](#). For testing skal JUnit brukes. Dere vil få en tutorial om LibGdx neste uke som vil hjelpe dere med å løse deloppgave 4.

Dere står fritt til å velge mellom Maven og Gradle som build system. Det er vedlagt et skjelettprosjekt med LibGdx, JUnit og Maven som kan brukes. Dette er helt frivillig, og dere kan også sette opp prosjektet selv.

Deloppgave 1: Organiser teamet

Gruppen trenger et gruppenavn, som også blir navnet på github-repositoriet deres. Kartlegg hvilken kompetanse de ulike medlemmene av teamet har, og ta med en kort oppsummering i innleveringen.

Dere skal så fordele roller dere bestemmer dere for (kan feks være teamlead, kundekontakt osv). Skriv en kort begrunnelse for hvilke roller dere bestemmer dere for og hvorfor. Sett opp et project board. En mulighet er å bruke GitHub project board eller trello. Dette må settes opp, og det er viktig at alle i gruppen vet hvordan de bruker verktøyene dere velger.

Deloppgave 2: Velg og tilpass en prosess for laget

Dere må finne ut om dere vil følge en bestemt prosjektmetodikk (XP, Scrum, Kanban, parprogrammering, testing osv), evt hvilke elementer fra ulike prosjektmetodikker dere vil ha med.

Diskuter i teamet hvilke metoder som hjelper teamet med å utvikle fungerende og veldokumentert programvare under prosjektet. Diskuter også hvilke tilpassninger som trengs for å fungere godt i et slikt studentprosjekt. Involver gjerne TA i diskusjonen om mulige problemer.

Vurder viktige aspekter ved prosessen, for eksempel hvordan organisere møter, definisjon og tildeling av oppgaver, oppfølging av arbeid, hvilke programvareutviklingsaktiviteter som trengs (og når), hvilke prosjekteringsaktiviteter som trengs (og når). Skriv en kort beskrivelse av prosessen i prosess- og prosjektplanen.

Diskuter i teamet hvordan dere skal organisere dere under prosjektet. Noen viktige elementer:

- Møter og hyppighet av dem
- Kommunikasjon mellom møter
- Arbeidsfordeling
- Oppfølging av arbeid
- Deling og oppbevaring av felles dokumenter, diagram og kodebase

Skriv en kort beskrivelse av hvordan teamet ditt planlegger å organisere prosjektet den første tiden.

Deloppgave 3: Få oversikt over forventet produkt

Dere skal lage en digital versjon av RoboRally. Denne versjonen skal inkludere de fleste av RoboRallys funksjoner som prosesskort, mekanismer på spillebrettet og mulighet for å spille flere sammen.

Dere skal lage en spesifikasjon som inneholder:

- En kort beskrivelse av det overordnede målet for applikasjonen
- En liste over brukerhistorier til systemet basert på MVP-kravene.
- For hver brukerhistorie, skal dere ha akseptanskriterier og arbeidsoppgaver, samt beskrivelse av hvilke krav brukerhistorien oppfyller (dette lager dere kun for historier dere er ferdige med, holder på med, eller skal til å begynne med)
- En prioritert liste over hvilke brukerhistorier dere vil ha med i første iterasjon (altså frem til levering av denne oppgaven, se [deloppgave 4](#) for forslag).
- **(Frivillig)** Krav til MVP er gitt i neste deloppgave. Dersom dere ønsker å utvide denne listen med ytterligere funksjonalitet, skal det også med som en del av denne spesifikasjonen.

Deloppgave 4: Kode

En del av leveransen for denne oppgaven skal være kode. Fokuset i starten bør være på å levere et minimum viable product (MVP) så raskt som mulig (vi regner med at flesteparten klarer å nå gjennom MVP til andre oblig). Krav til MVP:

1. Vise et spillebrett
2. Vise brikke på spillebrett
3. Flytte brikke (vha taster e.l. for testing)
4. Robot besøker flagg
5. Robot vinner ved å besøke flagg
6. Spille fra flere maskiner (vise brikker for alle spillere, flytte brikker for alle spillere)
7. Dele ut kort
8. Velge 5 kort
9. Bevege robot ut fra valgte kort

Vi forventer at dere skriver brukerhistorier (husk å få med hvilke krav brukerhistoriene dekker i beskrivelsen) og kode for punkt 1 til 5 ved denne innleveringen. Dere fortsetter med de neste kravene etterhvert som dere blir ferdig, det lønner seg å begynne på multiplayer så snart som mulig etter 1-5 er oppfylt. **Libgdx and Tiled Tutorial** dekker det dere trenger for å klare kodebiten nå i starten. I tillegg til koden skal dere kort beskrive hvilke arbeidsoppgaver som må løses for å realisere brukerhistoriene. I vurdering vil vi vektlegge hvor fullstendige brukerhistoriene er fremfor hvor mange brukerhistorier som er påbegynt.

Hovedvekten av vurderingen på sluttproduktet ligger på spillmekanikken og måten dere jobber sammen, ikke det visuelle. Det er ok å bruke [Open Game Art](#) for grafikk, og hvis dere finner ikoner med originalgrafikken har dere tillatelse til å bruke dette. Hvis gruppen har tid mot slutten er det opp til dere om dere ønsker å bruke dette, eller om dere vil produsere noe selv.

Koden skal være plattformuavhengig, altså skal den fungere uavhengig av operativsystem.

Oppsummering

Utfør et kort prosjekt-retrospektiv og diskuter hva som gikk bra, hva som ikke fungerte helt som forventet, hva (om noe) som ikke virket i det hele tatt, og eventuelle nye aktiviteter eller verktøy som teamet vil prøve ut i løpet av neste obligatoriske oppgave. Diskuter hvorfor ting fungerte eller ikke fungerte. Skriv opp en kort oppsummering av diskusjonen, og last opp til team repo-et.

På slutten av denne oppgaven kan dere gjøre en liten vurdering av hvor bra dere traff på oppgaven. Dette kan dere bruke til å justere hvor mange oppgaver dere tenker å få inn i neste iterasjon, som da leveres med obligatorisk oppgave 2.

Vurderingskriterier og vekting

Innleveringsfrist: **12. februar 2021, klokken 16:00.**

For å få oppgaven godkjent må gruppen:

- Opprettet en repo-struktur som beskrevet i deloppgave 1.
- Skrive fornuftige commit meldinger når endringer lastes opp i gruppens repo.
- Lastet opp de etterspurte resultatene i repo.
- Lastet opp leveransen til github i markdown-format
- Leveransen skal lastes opp i egen mappe i repo-et som heter "Deliverables". Hver obligatoriske oppgave blir da "ObligatoriskOppgaveX.md" i denne mappen.
- Tag en commit av kildekoden som er leveransen til hver innlevering. Dette gjør det enkelt for de som skal vurdere å gjøre det på riktig tidspunkt.

Måten dere jobber sammen og løfter alle medlemmene i teamet vil vurderes sammen med kodekvalitet, oversiktighet og dokumentasjon. Dere vil få tilgang til retteskjema samtidig som oppgaven slippes, slik at dere vet hvordan dere blir vurdert.

For å få oppgaven godkjent må hvert teammedlem:

- Møte og delta aktivt i minst 75% av ukentlige møter med deres TA.
- Utføre de tildelte oppgavene under og mellom disse ukentlige møtene, som avtalt i teamet.
- Rapportere tilbake til teamet om fremdrift og/eller problemer som oppstår, som avtalt i teamet.

Vektlegging: Denne leveransen teller 12,5% av slutt karakteren.