

Semesteroppgave for Programutvikling, V18

Denne oppgaven gir deg mulighet til å skaffe erfaring innen programutvikling utover enkle ukes- og obligatoriske oppgaver. Dette vil sannsynligvis bli den vanskeligste tekniske utfordringen du noen gang har møtt. Semesteroppgaven er designet for å både pushe dine tekniske egenskaper og utfordre dine ferdigheter i å arbeide i gruppe.

Hovedmålene i denne semesteroppgaven er å lære hvordan å arbeide i team, hvordan å utarbeide et større program, og lære om planlegging og gjennomføring av et større programmeringsprosjekt i gruppe. Det forventes at du arbeider ca. 110 timer med semesteroppgaven totalt gjennom semesteret, fordelt over 14 uker fra uke 6 til uke 20 (ekskludert påskeuken).

Denne oppgaven representerer eksamen i Programutvikling og karakteren du får i emnet er utelukkende basert på ditt bidrag i denne semesteroppgaven. Karakter blir satt etter følgende generelle mål:

- At du har arbeidet effektivt med medlemmene i din gruppe (17 %)
- At dere har implementert et program i samsvar med oppgavebeskrivelsen (33 %)
- At du har produsert et signifikant teknisk bidrag til gruppebesvarelsen (50 %)

Hver gruppe har en unik komposisjon av personligheter, erfaringer, og motivasjonsnivå. Din gruppe må planlegge bidraget til hvert medlem av gruppen, der alle medlemmene har blitt tildelt et konkret ansvarsområdet med genuine tekniske utfordringer. Hvert medlem skal kunne påvirke planleggingen av deres egne bidrag og dette burde være basert på erfaring, interesse, og motivasjon. Dere vil i løpet av prosjektet møte ulike administrative utfordringer, noen som kan håndteres og andre som ikke kan håndteres. Hvis dere møter utfordringer eller konflikter dere ikke løser sammen, må du eller dere i gruppen kontakte deres kontaktperson. I verste fall, kan eneste naturlige løsning være å gjøre endringer i gruppen, som å dele gruppen i to.

Evaluerings av semesteroppgaven vil kvalitativt gjennomføres ved å evaluere graden av din oppnåelse av de tre målene beskrevet over. En tredjedel av evalueringen omhandler deres totale gruppebesvarelse og alle medlemmene i gruppen blir evaluert likt for denne delen. Den resterende delen av evalueringen baserer seg på dine individuelle bidrag til prosjektet. Hvis du er i løpet av semesteret bekymret over ditt bidrag, må du kontakte din kontaktperson snarest. Enkelte studenter kan ofte føle seg utelukket fra gruppearbeidet og dette må håndteres så tidlig som mulig.

Obligatoriske arbeidskrav

For å kunne levere semesteroppgaven, må du ha bestått de tre obligatoriske arbeidskravene i emnet. Hvis du allerede har bestått disse, er det ikke nødvendig å få godkjent arbeidskravene på nytt (vær obs på 2-års regelen).

Hvis du ikke får den individuelle prøven godkjent, får du dermed ikke levere inn semesteroppgaven. Det er av denne grunn ikke hensiktsmessig å arbeide med oppgaven i gruppe med andre studenter (som har bestått prøven). Du kan gjerne gjennomføre semesteroppgaven på egenhånd, men din besvarelse kan ikke bli evaluert.

De to obligatoriske oppgavene skal utarbeides sammen i gruppe og blir godkjent på individuelt basis. Den første oppgaven vil fokusere på deres planlegging og spesielt de planlagte individuelle bidragene. Den andre oppgaven omhandler en rapport for utførelse og vil fokusere på de utførte individuelle bidragene.

Oppgavebeskrivelsene for de obligatoriske oppgavene gir fullstendig informasjon angående krav for innlevering og godkjenning, samt tidsfrist for innlevering. Disse finner du på Canvas under Oppgaver.

Oppgavebeskrivelse

Besvarelse av semesteroppgaven skal bestå av følgende elementer:

- Programkode. Programmet må implementeres med programmeringsspråket Java, med bruk av JavaFX/FXML biblioteket for det grafiske brukergrensesnittet. Alternative språk og biblioteker er tillatt som utvidelser, som for eksempel Android SDK for Android applikasjon. Hvis du er usikker på om du kan bruke en spesiell teknologi eller et bestemt språk eller bibliotek, så ta kontakt med emneansvarlig for avklaring.
- JavaDoc dokumentasjon for alle klasser.
- Et kjørbart Javaprogram. Det anbefales å pakke programmet i en kjørbart jar-fil. Alle moderne IDE'er tilbyr denne funksjonaliteten.
- Grupperapport.
- Individuell rapport. Denne rapporten skal leveres i en egen Canvas oppgave og skal ikke inkluderes i gruppebesvarelsen.

Gruppeinnleveringen skal leveres i Canvas som en pakket ZIP fil. Besvarelsen skal ikke overstige 100 MB.

Programmet som skal implementeres består av en kjernedel, videregående elementer, og andre utvidelser. Kjernedelen må være korrekt implementert for å bestå minimumskravene til emnet. Gjennomførelse av de videregående elementene vil demonstrere håndtering av store deler av pensum. Andre utvidelser kan også gjennomføres, noe som spesielt vil demonstrere selvstendighet.

Kjernedel

Kjernedelen skal bestå av to elementer:

- Et spill der brukeren kan bevege en avatar i en 2-dimensjonal verden. Avataren må i det minste være gjenkjennelig av spilleren, som for eksempel med et enkelt grafisk objekt som en sirkel eller et rektangel. Brukeren skal kunne bevege denne avataren ved input, som tastetrykk eller bevegelse av mus. I tillegg, skal spillet automatisk animere fiendtlige objekter. Hvis et fiendtlig objekt "treffer" avataren til spilleren, skal spillet avsluttes med en melding til brukeren. Et treff kan for eksempel bety at de 2-dimensjonale koordinatene for avataren og fienden er identiske. Spilleren skal ha muligheten til å sette spillet på pause og starte på nytt.
- Filbehandling med lagring av spillets tilstander og opprettelse av spill fra fil. Spilleren skal kunne lagre spillet til fil slik at man kan fortsette spillet der man slapp. Hvis filen brukeren etterspør ikke eksisterer eller hvis filen er korrupt, skal det gis beskjed til brukeren om dette.

Videregående elementer

De videregående elementene er relatert til to aspekter: logikk og ytelse.

Logikk. Spillets kjerne omhandler et generelt spill der brukeren beveger en avatar og der bevegelsen har som mål i å unngå fiendtlige objekter. Dere kan videreutvikle dette til et mer fullstendig spill, med forskjellige nivåer, ulike typer fiender, bruk av lyd og musikk, utforming av forskjellige verdener (maps), og aspekter som liv (health points) og bonus objekter (power ups). Inspirasjon fra arkadespill som Pac-Man og Space Invaders kan brukes og tolkes ikke som plagiat (med mindre du bruker konkret kode fra andre kilder).

Ytelse. Forbedring av ytelse kan oppnås i Java med tråder. Det må komme tydelig frem i besvarelsen hvordan bruk av tråder forbedrer ytelse og i hvor stor grad ytelsen har blitt forbedret.

Andre utvidelser

Andre utvidelser til programmet kan også gjennomføres. Slike utvidelser kan hjelpe til med å demonstrere meget eller svært god vurderingsevne og selvstendighet. Her står du fritt til å komme på ulike utvidelser, men kravet er at utvidelsen skal forbedre brukeropplevelsen til spilleren. Noen idéer:

- En editor for opprettelse av spillverdener eller nivåer til spillet
- En 3D versjon av spillet, der spilleren beveger 3D objekter
- En Android applikasjon der spillet er konvertert til en mobilapplikasjon
- Integrere spillet med Raspberry Pi og en arkade kit

Grupperapport

Gruppen må sammen utarbeide en grupperapport for semesteroppgaven. Denne rapporten skal belyse de delene av oppgaven dere er spesielt fornøyd med. I tillegg, skal dere diskutere elementer som dere mener kunne ha blitt gjennomført bedre og/eller aspekter i besvarelsen dere ikke er helt fornøyd med. Utdyp hva dere har lært i henhold til prosjektarbeid og hva dere vil gjøre annerledes for deres neste store gruppeprosjekt. Til slutt, dokumenter arbeidet som har blitt utført for hvert medlem, med beskrivelse for medlemmets tekniske bidrag.

Individuell rapport

Den individuelle rapporten skal inneholde følgende elementer:

1. Generell refleksjon av din erfaring med semesteroppgaven.
2. En oppsummering av dine individuelle bidrag til gruppearbeidet.
3. Evaluering av bidragene gjort av hver av de andre medlemmene i gruppen.
4. Et utvalg av kildekode med forklaring som belyser dine viktigste tekniske bidrag.

Evaluering av tekniske bidrag

Evaluering av programkoden følger pensum i Programutvikling, som bygger på emnet Programmering. Det refereres dermed til semesterplanen og materialet fra forelesningene for hvilke tekniske temaer som tas hensyn til i evalueringen. En anbefaling er å bruke dokumentasjonen aktivt til å begrunne ulike valg dere gjør i utarbeidelsen av koden. Javadokumentasjon skal først og fremst beskrive klasser, men vi tillater i dette emnet at dere også diskuterer problemstillinger her.

Når det gjelder karaktersetting vil dette, som alltid, følge de generelle kvalitative beskrivelsene for de ulike karakterene fra universitets- og høyskolerådet:

http://www.uhr.no/utdanning/karakterpanel_1

Code of conduct

Kopiering av kode fra Internett eller fra ferdigstilte løsninger, samt kopiering av kode mellom grupper, tilsvarer juks. Dette vil ikke bare føre til strykarakter, men vil også føre til utvisning fra alle norske universiteter og høyskoler. Vi bruker de beste verktøyene på markedet for å identifisere juks.

Gruppearbeid kan være krevende, spesielt for et lengre prosjekt som denne semesteroppgaven. Vi tolererer imidlertid ikke uakseptabel oppførsel. Dette inkluderer mobbing eller trakassering basert på kjønn, etnisitet, hudfarge, uførhet, seksualitet, religion eller tro, eller alder.

Mobbing er en form for psykologisk trakassering. Mobbing kan være trusler som undergraver selvtillit, kompetanse, og integriteten til offeret. Psykologisk trakassering kan også inkludere kontinuerlig

ufortjent kritikk, nedsettende bemerkninger, høylytt kjefting, banning og støtende ordbruk, og påtrengende oppførsel. Om du er selv et offer for trakassering eller observerer trakassering, må du rapportere dette til emneansvarlig.