

PG4400 – C++ programmering

Innleveringsmåte: "WiseFlow".

Deltakere

Innleveringen skal leveres *individuell*.

Det er lov å snakke sammen og hjelpe hverandre, men ikke kopiere kode direkte.

Kode

- Forståelse for C++. Ta med elementer du har lært om, der det er naturlig.
- Bruk mest mulig moderne C++, med moderne funksjonalitet fra standardbiblioteket.
- Fokuser på god software design.
- Ryddig kode. Gode navn på variabler og funksjoner, naming conventions og konsekvent språkbruk. Det er ikke nødvendig å bruke samme kodestandard som foreleser, så lenge du forteller hvor standarden kommer fra.
- Den innleverte løsningen skal kompilere og kjøre, med en beskrivelse av hva sensor må gjøre for å kunne kjøre løsningen.
- Velg «Build -> clean» før du leverer, så blir filene mye mindre.
- Lever en kjørbar fil ved siden av.

Dokumentasjon

- Her gjør du rede for prosessen, og valgene du har gjort underveis.
- Få fram hvorfor din løsning er bra, vis med ord eller figurer at du har tenkt, planlagt og resonert du frem til gode valg underveis.
- Dersom du får problemer med implementasjonen og ikke føler du er ferdige når du leverer, påpek dette og beskriv hva du mener løsningen mangler.
- Gi en begrunnet redegjørelse for klasser/funksjoner du lager
- Forklar hvordan og i hvilken grad du har samarbeidet med andre
- Legg ved en kort video for å vise hvordan spillet fungerer i tilfelle sensor ikke får til å kjøre koden.

Muligheter og begrensninger

- Det er lov å google etter tips.
- Det er ikke lov å kopiere andres kode/løsninger direkte.
- Det er greit å benytte kode delt ut i timene/i forbindelse med øvingene.
- Hvis ferdig kode brukes, gir en begrunnet redegjørelse for hva du bruker.

Dokumentasjonen leveres sammen med prosjektoppgaven (i samme .zip-fil – se neste side).

Prosjektoppgaven

Hver gruppe skal programmere en softwareløsning i C++. Semesteroppgaven består i planlegging, programmering og bygging av softwareløsningen. Alle prosjektfiler (kildekode, grafikk, m.m.) skal leveres sammen i en .zip-fil. Teknisk bør oppgaven baseres på 2d-grafikk i SDL2. Oppgaven er basert på den klassiske sjangeren scrolling shooter. Et slikt spill kan sees her:

Hovedkrav til løsningen:

Minimumskrav til løsningen (med dette kan man maks få karakteren B):

1. Romskipet til spilleren kan bevege seg langs nedre kant av skjermen, styrt av knapper, mus, joystick eller liknende.
2. Fra toppen av skjermen kommer det inn fiender som beveger seg mot nedre kant av skjermen.
3. Fiender følger et enkelt forhåndsprogrammert bevegelsesmønster uavhengig av spilleren.
4. Romskipet kan skyte prosjektiler. Når disse treffer fiendene blir de borte.
5. Fiendene skyter prosjektiler.
6. Hvis spilleren treffes av et prosjektil eller en fiende, avsluttes spillet.
7. Den innleverte løsningen må være kjørbart, med en beskrivelse for hva sensor må gjøre for å kunne kjøre løsningen.

Mulige utvidelser:

- Romskip og fiender skal ha enkle animasjoner
- Punkt 6 over byttes ut med: Hvis spilleren treffes av et prosjektil eller en fiende, mister den ett liv. Den har 3 liv.
- Når alle liv er brukt opp kommer en «game over screen» og spilleren får tilbud om å starte på nytt.
- Det er flere typer fiender i spillet, disse har forskjellige bevegelsesmønstre og prosjektiler.
- Alternativ til punkt 3: Noen fiender skal være litt smarte, som at de faktisk sikter på spilleren eller liknende.
- Det skal finnes en fil som definerer hvor og når fiender kommer inn i spillet.
- Spillet skal telle og vise poeng basert på hvor mange fiender spilleren har truffet.
- Kollisjonssjekk skal være parallellisert.
- To spillere skal kunne spille samtidig. Det finne to romskip som kjemper mot fiendene på samme skjerm, og de kan kjøre gjennom hverandre. Dette kan gjøres lokalt eller over nett.
- Fornuftig bruk av lyd

Mulig alternativ:

- I stedet for SDL2 kan oppgaven lages for et dot-matrix display og noen knapper koblet til en microcontroller. Da må det leveres grundig videodokumentasjon av spillet som kjører og hvilke featurers det har. Punkt 7 gjelder fortsatt, men det er ikke sikkert sensor har tilgang til riktig hardware.

Ellers står du fritt til å utvide oppgaven med funksjonalitet så lenge det er begrunnet i dokumentasjonen.