

Mappe AIS1003 høst 2023 (utlevering 3. oktober)

Prosjektoppgave:

Kravspesifikasjon:

Dere skal skrive programvare som benytter seg av biblioteket [threepp](#). Dette er et bibliotek som gjør det mulig å tegne og interagere med 3D objekter på skjermen, samt hente input fra bruker gjennom mus og tastatur. Det er også mulig å lage enkle brukergrensesnitt ved hjelp av ekstra biblioteker. Et knippe oppgaver å velge mellom finner dere lenger ned i dokumentet. Et overordnet mål er at løsningen deres skal demonstrere bruk av objekt-orienterte konsepter. Koden skal skrives i C++ og bruke CMake.

Kildehenvisninger

Dersom dere i merkbar grad benytter dere av kode som dere selv ikke har skrevet, skal dette dokumenteres i kodebasen sammen med henvisning til kilden, og en begrunnelse for hvorfor og hvordan dere mener benyttet kode bidrar til deres løsning.

Gruppe/samarbeid

Det er lagt opp til at man kan jobbe enten alene eller sammen i grupper på to. Det er viktig at man raskt finner ut om man er på en gruppe der begge er aktive og samarbeidsvillige. Det er bedre å jobbe alene enn på en gruppe som ikke fungerer skikkelig. Alle skal melde seg inn i en gruppe på Blackboard, enten alene eller i par.

Versjonskontroll/Git

Git **skal** benyttes, og det forventes at prosjekthjelpemidler som issues og pull requests benyttes i prosjektet.

Enhetstester

Prosjektet skal inneholde enhetstester (unit-test) som kan teste ulike deler av koden deres.

Continious Integration (CI)

Automatisk bygging samt testing ved bruk av [github workflows](#) vil bidra til en bedre løsning som gjør det mulig å sjekke kryssplatforms-kompabilitet. Ta gjerne utgangspunkt i threepp sitt eget CI oppsett, men husk å kreditere kilden som med all annen kode dere finner/gjenbruker.

Bruk av tredjeparts bibliotek

Dere står fritt til å bruke tredjeparts bibliotek, men disse må være konfigurert på en slik måte at det er mulig for andre å trivielt bygge koden. Dette gjelder også bruk av rammeverk for enhetstesting. Catch2 brukt i øvinger og demonstrert i klassen oppfordres til å brukes. Dvs. benytt vcpkg (manifest mode) og/eller CMake fetchcontent slik dere har fått demonstrert.

Vurderingskriterier

Besvarelser vil bli vurdert basert på følgende kriterier:

- Om kravspesifikasjonen har blitt fulgt.
- Om god forståelse for grunnleggende prinsipper i programmering som variabler, datatyper, metoder, betingelser, løkker osv. kan demonstreres.
- Fornuftig design og bruk av klasser og funksjoner.
- Fornuftig bruk av datastrukturer.
- Om prinsippene for god design (abstraction, encapsulation, coupling, cohesion, responsibility driven design osv.) har blitt fulgt.
- Om det er brukt gode, selvforklarende navn på klasser, metoder og variabler/felt/parametere.
- Riktig og ellers fornuftig bruk av referanser, pekere og «value types».
- Bruk av konsekvent og ryddig kode design (formatering, konvensjoner etc.)
- Om koden er selvforklarende og ellers godt dokumentert.
- Om gode og dekkende enhetstester er implementert.
- Tilstrekkelig feilhåndtering («defensive programming», håndtering av ugyldig input fra bruker, verifisering av parametere osv.).
- Evne til å reflektere over løsningen.
- Jevn progresjon fra prosjektstart.
- Kreativitet og selvstendighet.

Merk: Det forventes en innlevering som gjenspeiler tiden dere har til rådighet (7 uker + eksamensperiode).

Besvare og levere

Hver gruppe **skal** levere en versjon av prosjektet sitt i Blackboard **innen torsdag 2. november**. Dere vil få en *overordnet* tilbakemelding på dette arbeidet.

Den endelige mappen leveres individuelt i Inspera. Denne skal bestå av:

- Prosjektfilene dere leverte på Blackboard tidligere (se ovenfor), sammen med tilbakemeldingen dere fikk (feedback.md).
- Endelige prosjektfiler (Kode, CMake konfigurasjon, README, evt. ressurser, etc.)
 - Git repoet (.git folder) skal være med i innleveringen.
 - Det forventes at prosjektet har en godt beskrivende README fil (markdown syntax) som beskriver prosjektet. **Legg innsats i denne.**
 - Hva er dette?
 - Hva kan det gjøre?
 - Hvordan skal det brukes?
 - **Få med kandidatnummer(e) i README.**
 - Maskinspesifikke konfigurasjons- og byggfiler (cmake-build-*, .idea, etc.) skal ikke være med.

Dette skal leveres i en enkelt zippet mappe (.zip). Hver del skal ligge i en egen folder:

```
/Midlertidig
  <Midlertidig prosjektløsning og tilbakemelding i denne mappen>
/Endelig
  <Endelig prosektløsning i denne mappen>
Kandidater.txt
```

I tillegg til dette vil det gis en individuell skriveoppgave gjennom Inspira, som vil være åpen for innlevering i perioden 11. des 09:00 – 12. des 12:00.

Endelig karakter settes basert på en helhetlig vurdering av innholdet i mappen.

Oppgaver

1. Tetris.
2. Astroids.
3. Fysikksimulering.
 - Enkle kollisjoner mellom primitiver / hydrodynamikk.
4. Oppsett og kontroll av serielle kinematiske lenke(r). F.eks. slange / kran / robot.
 - Kalkulasjon av posisjon gitt vinkler (forward kinematics).
 - Kalkulasjon av vinkler gitt posisjon (inverse kinematics).
 - F.eks ved bruk Cyclic Coordinate Descent.
5. Innlastning av robotmodeller basert på [URDF](#).
 - Går godt sammen med (3).
6. Visualisering og simulering av [boids](#).
 - Med eller uten fysikk.
 - Kan lett utvides/forbedres.
7. Serialisering / deserialisering av scene.
 - Ved bruk av XML, JSON eller lignende velkjente formater.

Har du egne forslag til oppgaver, så forhør deg med faglærer. Ikke start på en oppgave som ikke er oppført uten et OK.

Merk: Om man ser at løsningen man tenker seg blir for lite omfattende (brukt lite tid / har ikke fått demonstrert kunnskapene sine nok), kan man fint implementere flere løsninger i ett og samme prosjekt. Spør om du er usikker. Kanskje greier man også å gjenbruke kode mellom løsninger?