

Individuel tag-hjem eksamen

Objektorienteret Programmering og Design

Jon Sparring

Deadline 19. januar 2014

1 Simulation af solsystemet

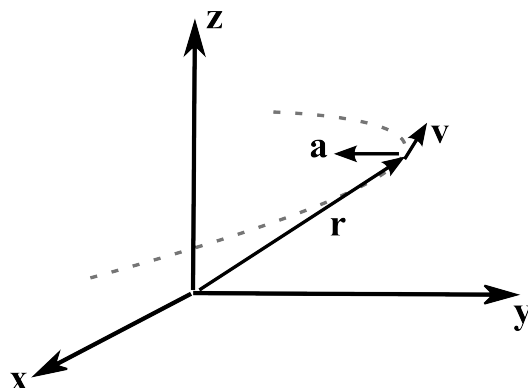
I denne opgave skal du implementere en simulator der kan simulere planeternes og Solens bevægelse i vores solsystem. Simulationen består af beregning af Solen og planeternes position og hastighed som funktion af tid. I simulationen inkluderer vi, udover de 8 planeter, også dværgplaneten Pluto.

1.1 Modellen

Planeternes bevægelse er bestemt af gravitationskraften. Dertil kommer at den enkelte planet følger en baneurve rundt om Solen som har en vis afstand til Solen og de andre planeter, samt har en hastighed som afhænger af hvor på banekurven planeten befinder sig (se Figur 1).

Vi vil antage følgende simplificeringer:

1. I simulatoren antager vi at planeter er punktformede masser (dvs. de har ingen udbredelse i rummet) og at de ikke roterer om sig selv.
2. Vi ser desuden bort fra påvirkning fra måner, kometer og andre former for planeter.
3. Solens masse er langt større end alle andre himmellegemer i solsystemet og derfor vil Solens gravitationstiltrækning på planeterne være betragtelig større end påvirkningen mellem planterne.
4. Vi kan også antage at planeternes gravitationstiltrækning på Solen er forsvindende lille og derfor sætte gravitationskraften på Solen til nul, dvs. at Solens acceleration er nul, og dermed ændrer Solens hastighedsvektor sig ikke med tiden.



Figur 1: En planets tilstand til tiden t angives ved en positionsvektor \mathbf{r} , en hastighedsvektor \mathbf{v} og en accelerationsvektor \mathbf{a} . Når tiden går, følger planeten en baneurve (illustreret med stiplede kurve) og tilstandsvektorene ændres.

Tabel 1: Relevante naturkonstanter og enheder, samt estimater af produktet mellem gravitationskonstanten og de forskellige planeters masser. Tal fra JPL Ephemeris [1]

Gravitationskonstant, G	$6.67384 \times 10^{-11} \text{ m}^3/(\text{s}^2 \cdot \text{kg})$
Astronomisk enhed, AU	$1 \text{ AU} = 149597870700 \text{ m}$
GM_{Solen}	$2.959122082322128 \times 10^{-4} \text{ AU}^3/\text{dag}^2$

Dette er selvfølgelig grove tilnærmelser til det virkelige solsystem, men modellen kan fungere til forudsigelse af planeterne baner over kortere tidsrum.

Under disse antagelser, kan vi for planet i beregne dens acceleration omkring Solen ved,

$$\mathbf{a} = -\frac{GM_{\text{Solen}}}{r^3} \mathbf{r}_{i-\text{Solen}} \quad (1)$$

hvor G er gravitationskonstanten (se Tabel 1), og hvor $\mathbf{r}_{i-\text{Solen}} = \mathbf{r}_i - \mathbf{r}_{\text{Solen}}$ angiver vektoren fra Solen til planet i og $r = \|\mathbf{r}_{i-\text{Solen}}\|$. Oftest estimeres produktet GM_i for de enkelte planeter i stedet for at estimere massen M_i og gravitationskonstanten G separat. Se Tabel 1 for et estimat af dette produkt for Solen.

1.2 Løsningsmetode

I simulatoren ønsker vi at beregne de enkelte planeters position $\mathbf{r}(t) = (x(t), y(t), z(t))$ som funktion af tiden t og som resultat af påvirkningen af gravitationsaccelerationen. Tiden t diskretiseres i tidsskridt Δt , således at tiden fremskrives ved hjælp af følgende rekursionsformel

$$t_{n+1} = t_n + \Delta t, \quad (2)$$

og en approximation af vektorerne $\mathbf{r}(t_{n+1})$, og $\mathbf{v}(t_{n+1})$, fremskrives ved¹

$$\mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + \mathbf{v}(t_n)\Delta t, \quad (3)$$

$$\mathbf{v}(t_{n+1}) = \mathbf{v}(t_n) + \mathbf{a}(t_n)\Delta t. \quad (4)$$

Accelerationen $\mathbf{a}(t_n)$ kan beregnes ud fra $\mathbf{r}(t_n)$ for alle planeter ved at anvende (1). Endeligt forudsættes initielle værdier for $\mathbf{r}(t_0)$ og $\mathbf{v}(t_0)$ for hvert himmellegeme kendte.

For at opnå høj præcision i simulatoren er det vigtigt at de initielle værdier angives med høj præcision. Dertil kommer at det er vigtigt at vælge så små tidskridt Δt som muligt, under hensyntagen til at beregningstiden stiger når Δt gøres mindre (flere beregningsskridt er nødvendige for at simulere eksempelvis 1 år frem i tiden).

2 Data

For at vurdere hvor præcis din simulation er, skal du sammenligne simulatorens resultat med data fra NASA JPL Ephemeris [1]. JPL Ephemeris er en kombination af observationer af planeterne bevægelse og simulering med høj præcision. NASA anvender blandt andet dette datasæt til planlægning af rummissioner.

Sammen med opgaveteksten finder du en tekstfil for hvert himmellegeme med informationer taget fra JPL Ephemeris datasættet. Datasættet består af position (X, Y, Z) og hastighed (V_x, V_y, V_z) for Solen, de otte planeter og dværgplaneten Pluto ved midnat den første dag i hver måned i perioden 1/1 2013 til 1/1 2014. Formatet for filerne er kommaseparerede dataelementer med en linje pr. tidspunkt. De første to dataelementer er datoangivelser i hhv. Epoch Julian Date (EJD) og koordinattidspunkt (CT) (standard dato-tid format), derpå kommer position (X, Y, Z) som de næste tre dataelementer og til slut hastighed (V_x, V_y, V_z) som de sidste tre dataelementer:

$$\text{EJD, CT, } X, Y, Z, V_x, V_y, V_z$$

¹Disse ligninger er Euler's metode til samtidigt at løse differentialligningerne $\frac{d\mathbf{r}(t)}{dt} = \mathbf{v}(t)$ og $\frac{d\mathbf{v}(t)}{dt} = \mathbf{a}(t)$.

EJD er angivet som et reelt tal som tæller antal dage siden en referencedato og kan med fordel anvendes som intern repræsentation af dato-tid i din implementation af simulatoren.

For præcision i beregningerne anvendes ofte astronomiske enheder (AU) i stedet for meter (m). Den astronomiske enhed er defineret ved gennemsnitsafstanden mellem jorden og solen og er fastsat til $1 \text{ AU} = 149597870700 \text{ m}$. Denne relation kan anvendes til konvertering af tal angivet i meter til astronomisk enhed eller omvendt. I JPL Ephemeris er position (X, Y, Z) angivet i enheden AU og hastighed (V_x, V_y, V_z) er angivet i enheden AU/dag.

JPL Ephemeris datasættet benytter det solsystem-barycentriske koordinatsystem, hvor origo er angivet ved solsystemets massecenter og akser angivet udfra jordens orientering på et referencetidspunkt. Du skal anvende dette koordinatsystem i din simulator (dette gøres uden videre ved at anvende JPL Ephemeris data til at angive initiale positioner og hastigheder for planeterne).

Yderligere information om planeterne og Solen kan eksempelvis findes på NASA Planetary Fact Sheets [2], men er ikke nødvendig for at løse denne opgave.

3 Opgaven

I denne opgave skal du designe en solsystem simulator og skrive et program i Java, der implementerer dit design. Minimumskrav til din løsning er at

1. den kan simulere solsystemets bevægelse over tid;
2. den kan indlæse JPL Ephemeris data filer for planeterne og solen;
3. den kan illustrere dine simulationsresultater, og sammenligne dem med JPL Ephemeris data;
4. brugeren interaktivt kan indtaste, hvor lang tid simulatoren skal simulere;
5. den følger Model-View-Control (MVC) mønsteret.

Opgavens besvarelse skal bestå af

1. Det udviklede program som kildekode dokumenteret efter Javadoc standarden. Når programmet køres skal følgende grafiske fremstillinger produceres i separate plotvinduer (og ikke mere end):
 - (a) en animation af planeterne's bevægelse som beregnet af din simulator i tidsrummet 1/1/2013-1/1/2014. Det skal blot være en 2D animation som viser x og y komponenterne af $\mathbf{r}(t)$;
 - (b) en tabel som for hver planet og for hvert tidspunkt i NASA JPL Ephemeris's tilhørende datafil viser den euklidiske afstand mellem positionen angivet i NASAs datafil og den beregnet af din simulator. Initiale værdier skal være de tidligste angivet i NASAs datafil.
2. Et repræsentativt billede fra animationen samt en udskrift af tabellen.
3. Et UML diagram, som illustrerer dit design.
4. Skemaer, der beskriver din afprøvning samt udskrift fra kørsel af din unittest.
5. 1-2 siders tekst i pdf format, som kort beskriver
 - (a) overvejelser og valg gjort ved udvikling af programmet;
 - (b) en kort fortolkning af tabellernes indhold;
 - (c) en kort beskrivelse af evt. fejl og mangler ved løsningen.

Ved vurdering af din besvarelse vil der, udover ovenstående, blive lagt vægt på dine valg af datatyper, datastrukturer og dit programs struktur.

Husk at tjekke at både kildekode og pdf med rapport er inkluderet i din aflevering!

4 Bedømmelseskriterier

Der lægges vægt på at der afleveres et kørende program, dvs. at et simpelt og lille men veludviklet, kørende program foretrækkes frem for et ambitiøst eller stort program, der ikke rigtig virker. Det er vigtigt, at individuelle klasser er veludviklet, specificeret, implementeret og afprøvet, selv hvis programmet som helhed ikke kan køre. Det anbefales stærkt at designe, implementere og afprøve et system med meget enkel funktionalitet, inden eventuelle udvidelser føjes til. Besvarelsen bedømmes ud fra deltagerens evne til at omsætte kursets målsætninger: at udarbejde et objektorienteret design og så programmere og afprøve det vha. af de på kurset gennemgåede faciliteter i Java. Se kursets målsætning på kursets hjemmeside i KUs kursuskatalog² herunder punktet ”Målbeskrivelse”.

En egentlig problemanalyse indgår ikke i bedømmelsen, hverken fra et fagligt perspektiv om eksempelvis animationsteori, fysik eller numeriske metoder, eftersom disse emner ikke er genstand for kurset.

5 Formalia

Denne opgave skal løses individuelt. Der tillades ikke genaflevering. Under helt særlige omstændigheder (sygdom, dødsfald, m.v.) kan der gives en udsættelse efter dispensation fra studienævnet. En dispensationsansøgning skal være studienævnet i hænde inden afleveringsfristens udløb. Yderligere oplysninger kan findes i eksamensbestemmelserne for Det Naturvidenskabelige Fakultet samt studieordningen for Datalogiuddannelsen.

Eksamensopgaven udgøres af nærværende dokument samt eventuelle supplerende beskeder fra kursets undervisere, det måtte være nødvendigt at give på kursets hjemmeside under ”Diskussionsforum”. Det er deltagerens ansvar løbende at holde sig orienteret på hjemmesiden.

Bemærk følgende politik for kommunikation under eksamensugen:

1. Eksaminanderne må stille spørgsmål til hinanden, som er uafhængig af opgaven, f.eks. om de obligatoriske opgaver eller vedr. stof i en af lærebøgerne, Java API, vejledningerne eller andre almene ressourcer.
 - Tænk på denne eksamen som en ”open book” eksamen: det er tilladt at bruge medeksaminander, lige som man bruger en bog eller anden alment tilgængelig ressource (som altså ikke ved noget specifikt om denne eksamensopgave).
 - Det er altså tilladt at bruge hinanden som kilde for f.eks. hvordan Observer/Observable fungerer, hvor man kan finde noget læsbart om Swing JButtons, hvad der adskiller et HashSet fra et TreeSet m.m.
 - Det er ikke tilladt at spørge hinanden eller tredjepart, hvordan en anden har tænkt sig at løse eksamensopgaven.
2. Hver linie kode og tekst, der afleveres skal være produceret af eksaminanden og af eksaminanden alene.
3. Alle kan stille spørgsmål om opgaven på kursets hjemmeside under ”Diskussionsforum”. Generelle svar (uden specifik kode eller specifikke designs for opgaven) på disse, herunder af andre studerende, er tilladt, men skal i denne forbindelse gives på diskussionsforummet for at tilgå alle eksamensdeltagere og således ligestille alle indbyrdes.
4. Alle eksaminander er underlagt KUs eksamensregler i hele opgaveperioden. Det betyder, at snyd, forsøg på snyd og hjælp til at begå snyd (f.eks. ved at hjælpe nogen konkret med løsningen) resulterer i bortvisning fra eksamen samt anmeldelse til studielederen mhp. eventuel forelæggelse hos dekanen mhp. yderligere skridt (f.eks. bortvisning fra alle eksaminer i indværende eksamensperiode). Det gælder for så vidt også ikke-OOPD/studerende, som hjælper en OOPD-eksaminand.

²<http://kurser.ku.dk/course/ndaa04012u>

5. Desuden er alle eksaminander også underlagt citatloven og KUs ordensregler i øvrigt, herunder om videnskabelig etisk adfærd. Det betyder, at kilder til viden, der viderebringes, som er væsentlige og ikke har almenviden inden for faget, være det i form af personer, bøger, artikler, eller elektroniske ressourcer, skal angives, og citater skal markeres som sådan. Bemærk, at det at kopiere en stump kode således kræver citation.

Litteratur

- [1] NASA Jet Propulsion Laboratory, California Institute of Technology, USA. JPL Ephemeris Horizons System <http://ssd.jpl.nasa.gov/horizons.cgi>.
- [2] NASA, USA. NASA Planetary Fact Sheets <http://nssdc.gsfc.nasa.gov/planetary/planetfact.html>.