**Analyse**

Vores program er en visuel simulering, der viser hvordan gravitation påvirker en planets bane omkring en stjerne, og det er bygget op omkring Newtons gravitationslov. Ideen var at tage de vigtigste elementer fra den fysiske teori og gøre dem synlige og forståelige ved hjælp af kode og grafik. Selvom vi har været nødt til at nedskalere alle vores værdier for at få det hele til at passe på skærmen, er princippet bag formlen stadig det samme og det er det, der er det vigtige for at vise sammenhængen mellem masse, kraft og afstand.

I programmet bruger vi den isolerede version af Newtons gravitationslov, så vi kan beregne afstanden [Equation]r mellem to legemer ud fra deres masser og tyngdekraften mellem dem.   
Formlen vi bruger er:

Picture

Den er direkte implementeret i funktionen beregnEllipse(), som bliver kaldt hver gang programmet tegner en ny frame. Variablerne M og m er henholdsvis massen af solen og planeten, og F er tyngdekraften mellem dem. Det hele sættes sammen med G, som er tyngdekonstanten. I virkeligheden er G = 6.67430·10⁻¹¹, men i vores kode bruger vi bare 6.67430igen på grund af skalering, så beregningerne ikke bliver for små eller for store til skærmstørrelsen.

Vi har valgt at bruge fire forskellige planeter i programmet: Merkur, Venus, Jorden og Mars. Hver planet har sin egen masse og en tilhørende tyngdekraftsværdi, som vi har defineret i starten af koden. For eksempel har vi givet Merkur en masse på 3.3011e2 og en tyngdekraft på 1.31e3, hvilket giver et realistisk forhold, når man tager skaleringen i betragtning. Programmet starter altid med Merkur som standard, men man kan skifte mellem planeterne ved at trykke på tasterne 1 til 4. Når man skifter planet, opdateres værdierne for m og F, og beregnEllipse() beregner en ny radius, som bruges til at tegne en ny bane.

Selve visualiseringen sker i draw()-funktionen, som kører igen og igen i Processing. Først tegner vi baggrunden sort og placerer solen som en gul cirkel i midten af vinduet. Derefter tegner vi en ellipse, som repræsenterer planetens bane. Bredde og højde på ellipsen bestemmes af radius højden er sat til 60 % af bredden, så vi får et realistisk elliptisk kredsløb. Det gør vi, fordi rigtige planeter sjældent har helt cirkulære baner, og det giver mere mening at vise en lidt aflang form.

En vigtig detalje er, at vi også viser formlen F = GMm/r^2 og dens isolerede version r = sqrt(GMm/F) direkte på skærmen som tekst. Det gør vi for at tydeliggøre, at det programmet gør, faktisk er baseret på rigtig teori det er ikke bare grafik, men en beregning, der viser noget fysisk korrekt. Det var vigtigt for os, at der er en tydelig forbindelse mellem teori og det, man ser på skærmen.

Når man skifter mellem planeterne, kan man tydeligt se, hvordan det påvirker kredsløbet. For eksempel har Mars lavere masse og kraft end Venus, og derfor bliver radius større, og ellipsen trækker sig længere ud. Det sker automatisk og i realtid, så man med det samme ser, hvordan forskellige masse og kraftforhold ændrer resultatet.

Selvom vi ikke har implementeret planetens bevægelse endnu den flytter sig ikke rundt i kredsløbet, men står stille så viser programmet alligevel det vigtigste: hvordan masser og kræfter hænger sammen og bestemmer afstanden mellem to himmellegemer. Hvis vi havde tilføjet hastighed og brugt numerisk integration (for eksempel Euler-metoden), kunne vi have vist selve bevægelsen, men det ville kræve et mere komplekst program. Vi valgte i stedet at fokusere på en ren visuel forklaring af kredsløbet og selve kraftberegningen bag.

Alt i alt viser vores analyse, at programmet fungerer som en slags **forenklet model**, hvor teorien er bevaret, men tallene og beregningen er tilpasset, så det hele giver mening grafisk. Og selvom vi bruger skalerede værdier, ændrer det ikke på, at de underliggende sammenhænge er rigtige. Vi bruger den samme matematik, og vi viser tydeligt, hvordan ændringer i masse og tyngdekraft påvirker radius. Det var hele pointen med simuleringen og det synes vi faktisk er lykkedes ret godt.

**Diskussion**

Vi synes, at vores program fungerer godt til det, vi ville opnå, men vi kan også godt se, hvor der er ting, der kunne være bedre. Det vigtigste for os var, at man kunne se, hvordan tyngdekraften påvirker en planets kredsløb og det gør programmet. Man trykker bare på en tast, og så opdateres ellipsen med det samme. Det er nemt at bruge og giver et hurtigt overblik over, hvordan masse og kraft hænger sammen med radius. Vi føler, at det giver mening, også selvom det er en meget simpel udgave af virkeligheden.

Det visuelle var noget af det, vi brugte meget tid på at få til at se pænt og overskueligt ud. Vi valgte sort baggrund, gul sol og en hvid ellipse, fordi det er tydeligt og nemt at aflæse. Vi har også placeret formlen øverst i vinduet, så man hele tiden kan se, hvordan programmet regner. Det gør det mere tydeligt, at det faktisk er fysik, der ligger bag ikke bare noget tilfældigt grafik.

Når det så er sagt, så er der også ting, vi kunne have lavet bedre eller bygget videre på. For det første viser vi ikke planetens bevægelse vi tegner bare selve kredsløbet som en ellipse. Det betyder, at man ikke kan følge med i, hvordan planeten bevæger sig rundt eller hvor hurtigt den går. Det ville vi kunne lave, hvis vi brugte noget som Euler-metoden, hvor man regner bevægelsen ud frame for frame. Men det er noget, der kræver lidt mere kode og lidt mere viden om integration og tid. Det kunne være et fedt næste skridt, hvis vi skulle bygge videre.

En anden ting er, at vi bruger faste værdier for hver planet. Det fungerer fint, men det kunne være fedt, hvis man selv kunne skrive værdier ind eller ændre dem løbende med en slider eller noget lignende. Så kunne man eksperimentere lidt mere og få en bedre forståelse for, hvordan tingene hænger sammen. Lige nu er det meget sat op på forhånd man kan skifte mellem fire planeter, men ikke ændre så meget selv. Det kunne være fedt at udvide.

Vi har også snakket om, at vores værdier er skaleret ned, så det hele kan vises på skærmen. Det gør selvfølgelig, at programmet ikke er 1:1 realistisk, men det var vi nødt til. Hvis vi brugte de rigtige værdier for f.eks. solens masse eller G, så ville radius blive alt for stor til at kunne vises. Men selvom tallene er justeret, så virker princippet stadig og det er det, vi synes, er det vigtigste. Vi bruger stadig den rigtige formel, bare med tilpassede tal, så man kan se effekten af ændringerne.

Alt i alt synes vi, at vi har lavet et program, der forklarer tingene på en nem og visuel måde. Det er ikke en realistisk simulation i forhold til virkelige målinger og bevægelser, men det er en god model til at vise, hvordan tyngdekraft virker, og hvordan masse og afstand påvirker en planets bane. Vi har gjort det simpelt, men vi har stadig holdt fast i teorien bag. Hvis vi skulle videreudvikle det, ville vi klart starte med at tilføje bevægelse og mulighed for mere brugerinput men som det er nu, synes vi, det gør det, vi havde brug for.