Fysik og Programmering:

Af Josva M.

Indholdsfortegnelse

[Projektbeskrivelse 3](#__RefHeading___Toc382_1637631377)

[Problemformulering 4](#__RefHeading___Toc384_1637631377)

[Krav Spec. 4](#__RefHeading___Toc386_1637631377)

[Rutediagram 5](#__RefHeading___Toc388_1637631377)

[Pseudo kode 6](#__RefHeading___Toc390_1637631377)

[Klassediagram 6](#__RefHeading___Toc392_1637631377)

[Prototype (skitse/mockup) 8](#__RefHeading___Toc394_1637631377)

[Litteraturliste 9](#__RefHeading___Toc396_1637631377)

# Projektbeskrivelse

Hoved idé:

Design og implementer et simuleringssystem i form af et Processing-program, der simulerer bevægelsen af en genstand under påvirkning af tyngdekraften og acceleration. Programmet skal tage højde for parametre som partiklens masse, initialhastighed og påvirkningskraftens egenskaber. Kriterier for bedømmelse: Korrekt fysisk modellering af bevægelsen, realistisk simulering af partiklens adfærd, og anvendelse af relevante programmeringskoncepter.

Programmet kører på telefonen og bruger telefonens sensorer for at bestemme tyngdekraften der påvirker genstanden. Telefonen vil bruge tyngdekraften til at påvirker partiklerne i simulationen. Man kan vælge mellem forskellige planeter som eksempel, månen og Mars hvor man bruger deres tyngdekraft i simulationen, der vil påvirke partiklerne i en anden måde som med Jordens tyngdekraft.

Appen kan have sliders der kan skifte på partiklernes egenskaber, som viskositet, vægt og størrelse osv.

Andre Ideer:

* Morse kode program
* Vejrudsigt app
* Budget omregningsprogram.
* Astronomi app...

# Problemformulering

Hvordan kan et interaktivt simuleringssystem designes og implementeres i Processing, der realistisk modellerer bevægelsen af en partikel under påvirkning af tyngdekraft og acceleration?

* Hvordan kan telefonens sensorer anvendes til at bestemme og simulere den aktuelle tyngdekraftpåvirkning?
* Hvordan kan simulationen tilpasses forskellige planeters tyngdekraft (f.eks. Månen, Mars og Jorden)?
* Hvilke parametre og brugerinteraktioner (f.eks. sliders til masse, viskositet, størrelse) kan implementeres for at skabe en fleksibel og realistisk simulation?
* Hvordan sikres det, at den fysiske modellering af bevægelsen er korrekt og realistisk?

# Indledning

Vi startede med at brainstorme forskellige ideer vi kunne lave indenfor programmering.

# Krav Spec.

Hårde krav:

* Programmet skal bruge telefonens sensorer (accelerometer, gyroskop) til at påvirke partiklerne i simuleringen.
* Programmet skal have nogle sliders der tillader brugeren at skifte på partiklernes egenskaber.
* Simulationen skal ske i realtid på telefonen.

Bløde krav

* Der kan være justeringer til hvor mange partikler der er i simulationen, til de tilfælde hvor telefonen ikke har nok ydeevne til at simulere
* Programmets brugeroverflade skal være nem at bruge. Knapperne der skifter mellem planeterne og sliders skal være brugervenlige.

# Rutediagram

Ligger ind på Draw.io

Nyeste screenshot

A screenshot of a computer

AI-generated content may be incorrect.

Denne rutediagram viser hvordan programmet virker når den starter. Efter program start, så kører den setup() og hvor den kalder og starter GUI og starter ParticleSystem. draw() sørger for at ParticleSystem er konstant opdateret. Data fra SensorManager og AccelerometerListener bliver sent til ParticleSystem som bliver brugt af simulationen.

# Pseudo kode

WIP

# Klassediagram

Her er nogle klasser som kan sættes ind i et klassediagram. Klassediagrammer er

|  |
| --- |
| Klasse: UI (brugeroverflade) |
| Attributter   * Sliders * Buttons |
| Metoder:   * justerEgenskaber() |

Klasse for UI/brugeroverfladen, hvor der er nogle sliders- og knapper til at skifte partiklernes egenskaber.

|  |
| --- |
| Klasse: Partikel/Partikler |
| Attributter:   * vægt * størrelse * viskositet |
| Metoder:   * simulerBevægelse() |

Klassen for en partikel eller partiklerne i simuleringen. Den har tre attributter med vægt, størrelse og viskositet af partiklerne, som afgør hvordan partiklerne bevæger sig på.

|  |
| --- |
| Klasse: Simuleringssystem |
| Attributter:   * form * kriterier * bedømmelse * anvendelse |
| Metoder:   * modellering() * simulering() |

Her er klassen for programmets simuleringssystem.

|  |
| --- |
| Klasse: UI (brugeroverfladen) |
| Attributter:   * Sliders * Buttons |
| Metoder:   * justerEgenskaber() |

Klassen for brugeroverfladen.

# Prototype (skitse/mockup)



Her er en skitse af hvordan appen skal se ud. Til højre i den sorte del, vil simuleringen foregå. Og til venstre er der nogle sliders og knapper, der ændrer på hvordan simulationen virker. Der er en ”tyngdekraft” slider, men der kunne også være en ”viskositet” eller ”vindmodstand” der også kunne skifte hvordan simulationen fungerer. Så er der knapperne. De her kunne virke som en slags ”preset” der sætter tyngdekraften og de andre sliders så de passer til den planet man trykker på.

# Litteraturliste

[www.Processing.org](http://www.Processing.org/)

<https://www.color-hex.com/color-palette/8919>

[https://www.chatgpt.com](https://www.chatgpt.com/)