

Rapport

20 februar 2024

Narongchai Cherdchoo

1 INTRODUKSJON

Boids simulasjon er et kunstig liv program med tre enkle regler som styrer deres naturlig begavelser. De prinsippene som er bruk i denne simulasjonen er boids sin reaksjon for hvordan de skal oppføre seg dersom de møter sine utfordringer. De prinsippene gjør det vanskelig for oss å forutsi på boids sin neste retning og dermed er dette et kunstig liv program.

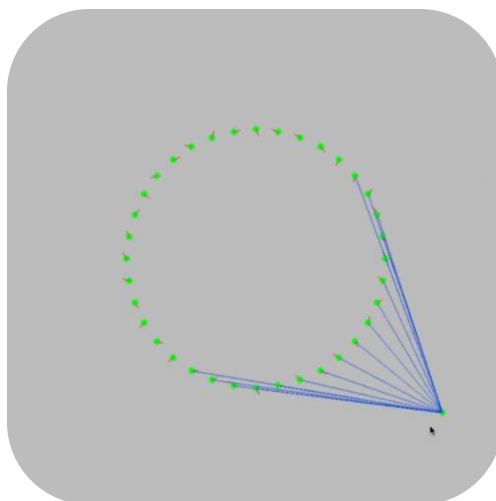
2 TEKNIKK & BAKGRUNN

Teknikken som brukes i Boids-simulasjonen er basert på tre disse prinsippene, dette vil få boids til å bevege seg fritt :

1. Seperasjon handler om å gå unna boids som er i nærheten.
2. Felles retning for hvert individuelt boid må tilpasse sin retning i forhold til nabo.
3. Gå mot senteret av massen innebære å endre sin retning mot det gjennomsnittlige posisjon av nabo.

Boids prinsippene er ment for å samle dem i en flokk uten at de blir påvirke av mindre flokk. Vi skal bruke vektor sin egenskapene for å definere vinkel og avstand for deres synsområde, slik at boids kan ta sin egen beslutning basert på avstand og vinkel til andre.

Synsvinkelen er boids sin evne til å bevege seg i en bestemt retning uten å ta hensyn på andre boids som er bak. Deretter vil posisjonen oppdateres gradvis i forhold til retningen, slik at de bevege seg naturlig med stabil hastighet.



3 DESIGN

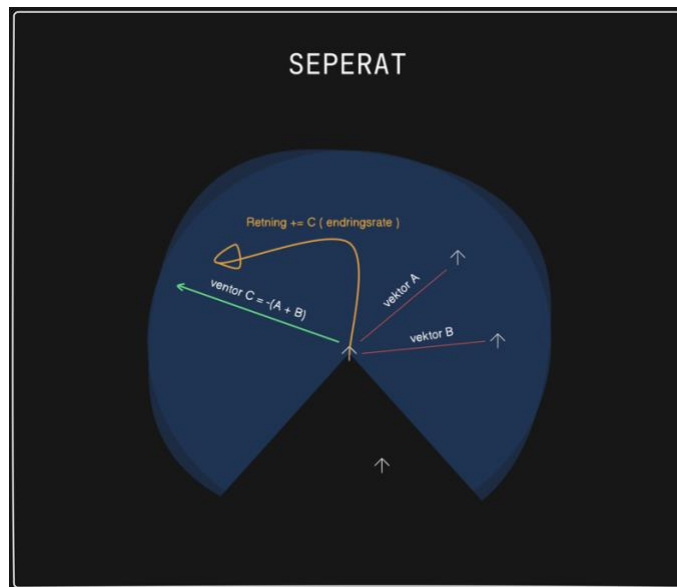
3.1 POSISJON & RETNING

Posisjon til boids oppdateres i forhold til retningen, ideen er at hvert enkelte boid skal ha to vektorer som presenterer sin posisjon og retningen. Posisjonen skal endres mot en bestemt retning. Denne logikken er naturlig og effektivt for posisjon endring etter en beslutning er tatt.

3.2 VINKEL & AVSTAND

Vinkel mellom individuelt retningen og nabo sin posisjon er for å sjekke om nabo boids finnes seg innen sin syns områdes eller ikke, og vi bruke avstand som deres radius slik at den kan reagere og ta et valgt dersom boids er i deres synsområde.

3.3 SEPARAT

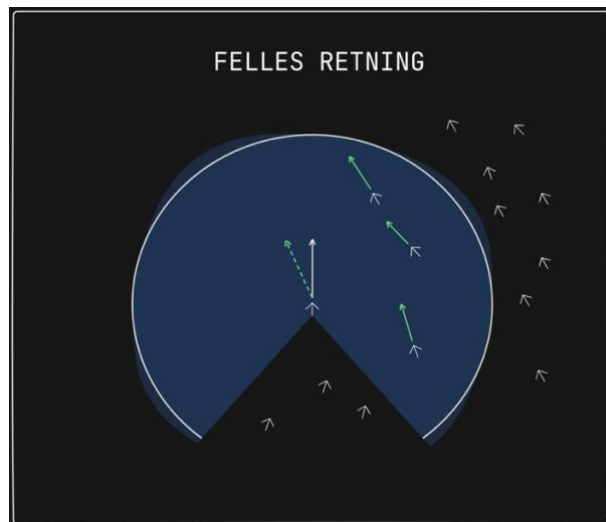


Tanken i dette er å få den individuelle boids til å bevege seg i motsatte retningen i forhold til andre boids som er i nærheten.

Vi kan definere andre boids ved å bruke vinkel og avstand for å sjekk om de er innenfor separate området eller ikke, slik at vi kan finne den inverse gjennomsnittlige retningen med å summere deres vektorer. Deretter normalisere vi retningen slik at hastigheten til boids blir stabil når posisjon skal oppdateres.

3.4 FELLES RETNING

I istedenfor å endre den individuelle sin posisjon som skjer ved separasjonen så endrer vi boids sin retning, slik at de gå på det samme retningen som deres nabo. Dette er en måte å skille mellom sin flokk og andre flokk.



3.5 SENTRUM AV MASSEN



Ideen her er å få boids til å bevege seg der det er mest boids. Dette kan gjøres ved å summere alle nabo sin posisjon, deretter beregne og bruke gjennomsnittlig posisjon som sentrum av massen. Måten det er beregnet på, begynne med å sjekke om avstand til naboene er innenfor et bestemt intervall, så deres vinkel.

Hvis naboene befinner seg innenfor definerte området, så blir deres posisjon lagt til i en samlest for posisjons vektor, deretter endrer vi boid retningen til sentrumet.

3.6 STIMULASJON

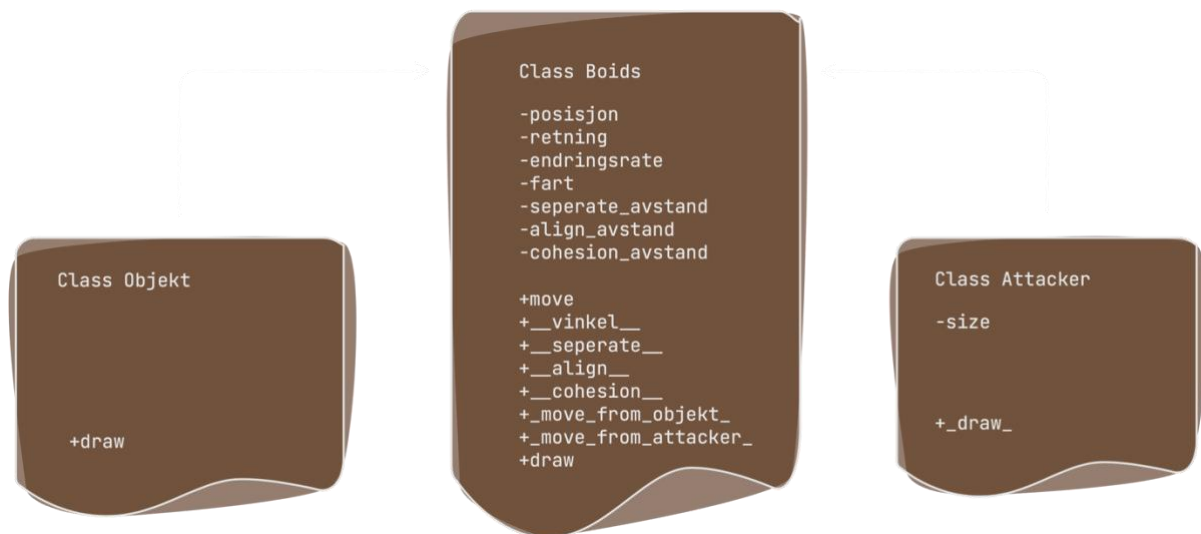
Stimulasjon vil kjøre av pygame med klokke funksjon som tillater stimulasjonen med å gå gjennom de nødvendige sjekk status per ramme, som posisjon endring per ramme, retningen endring per ramme.

4 IMPLEMENTERING

PYGAME

Stimulasjonen kjøres i pygame på 120 rammer per sekund, uten sprite biblioteket. Gruppering for ulike objekt er definert som en liste, dermed kan vi bruke en løkke for å hente alle objekter inn til stimulasjonen.

KLASSE OVERSIKT



Boids er hoved klassen med attributter som endrer klassens metoder, som synsvinkel, avstand og fart. Subklasene Objekt og Attakker arves fra Boids, dermed vil de ha muligheten til å bruke metodene som er definert i Boids klassen, derfor vil det være effektivt å tilpasse subklassen personlighet med sitt attributt defensjon som påvirke arv metodene.

BOIDS KLASSEN

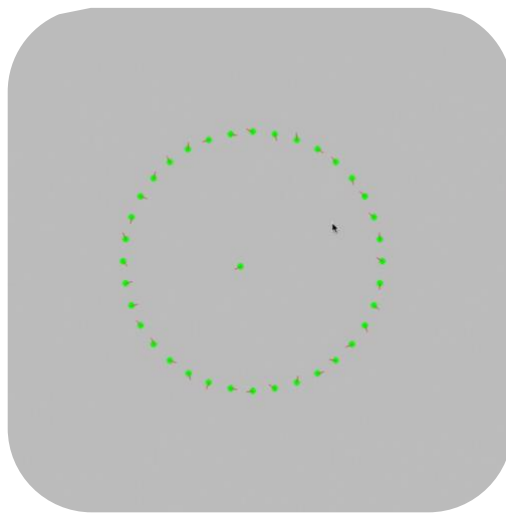
Attributt posisjon og retning er vektor med x og y, vi bruke disse vektor verdiene for å definere hvor i simulasjons området boids skal starte og retningsvektoren definer hastighet, avstand og vinkel til naboene. Vi kan bestemme hastighet eller posisjons endring for boids ved bruk av endringsrate verdiene, som gjør at de bevege seg gravis til et mål.

Avstand og vinkel attributter definer deres synsområde i forhold til reglene, slik at boids kan reagere når avstand og vinkel er innenfor det intervallet som er definert deres prinsippene. Dette er en evne vi har gitt til boids slik at de kan observere og reflektere det som skjer i deres synsområde.

BEVEGELSE METODEN

Denne metoden bruke retnings vektor for å oppdatere deres posisjon per ramme, slik at vi får en vennlig bevegelse endring, ved bruk av denne formelen : **$pos = pos + retning$** .

Funksjon for at boids skal ikke gå ut av skjermen, er lagt inn slik at retningsvektoren endrer seg når deres posisjon er nærme stimulasjonskaten. Målet med dette er for å bruke retningsvektoren som deres mål i forhold til deres prinsipper slik at posisjonen endres ramme etter ramme.



VINKEL BEREKNING^[1]

Vinkel er et verktøy for å beregne vinkelen mellom sin retningsvektor og andre boids posisjonen. Her brukes boids sin retningsvektor som et null punkt, dette betyr at boids kan ha en synsvinkel på 180° ved å se 90° til både høyre og venstre.

$$\frac{\vec{u} * \vec{v}}{|\vec{u}| * |\vec{v}|} = \cos(\theta)$$

Ved bruk av denne formelen kan vi bruke det individuelle retningen som \vec{u} og andre sin posisjon som \vec{v} , slik at produktet av lengden til vektorene deler plikkproduktet, så konvergerer radian til grader slik : **$\cos(\cos^{-1}(\theta))$** .

Da til denne metoden returner grader i vinkel med å bruke deres retning og posisjon som input. Videre kan vi bruke denne informasjon til å definere boids synsvinkel for å sjekke om andre boids er i deres synsvinkler for de definerte prinsippene.

SEPARAT

I denne metoden så bruke vi posisjon og retning vektor for å avgjøre om andre boids er i deres separate område. Deretter vil endringen på retning endres i forhold til endringsrate.

Dersom det er flere enn en boids så beregne vi den gjennomsnittlig vektor fra boid sin retning til andre boids posisjonen, da vil vi få en ny retning for å separere fra andre.

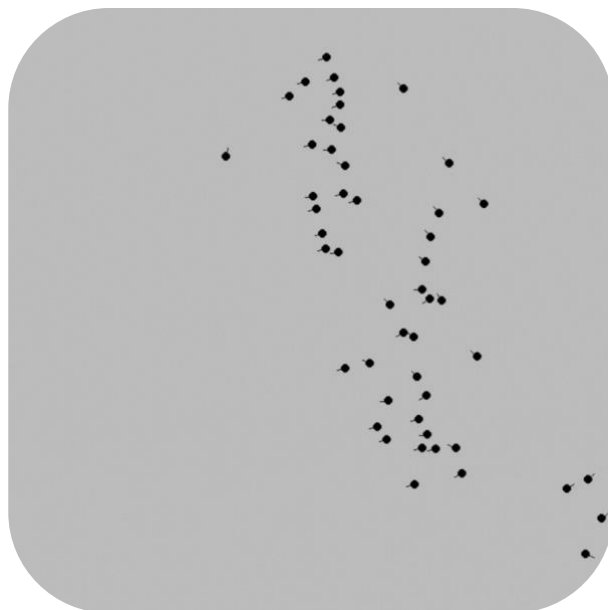
Når vi endrer retningen så vil posisjonen oppdateres etter den, og vi kan normalisere retningen etter endring er gjort, slik at hastigheten er stabilt.

FELLES RETNING

I denne metode så bruke vi det samme teknikken som er i separate, og forskjellen er at når andre boids finnes seg i sin området så skal boiden tilpasset sin retning til den gjennomsnittlige retningen.

Dette gjøres vi ved å bruke den gjennomsnittlig retning for andre boids innenfor den definerte området, deretter oppdatere vi retningen med en vist andel, slik at den ikke endrer seg raskt, men gravis.

Felles retningen vil gjør at boids bevege på det samme retning og hastighet, dermed kan det se ut som de er i en flokk, og hvis noen er for nært så blir den separate metoden aktivert slik at de går det motsatte retning og ikke kollidere med hverandre.



BEVEG MOT SENTER AV MASSEN

I denne metoden så ønsker vi at boid skal bevege seg der det er mest boid, dette kan vi gjøre etter å ha beregnet den gjennomsnittlige posisjon på alle boids som er i sitt sirkelen.

Forda vil vi ha et punkt som kan omgjøres til en vektor for å endre retningen som vil deretter påvirke posisjonen.

SUBKLASSE

For spenning sin skyld, har vi implementert Objekt og Attakker som en subklasse til Boids, der de bruke Boids klassen metoder, med sitt egen attributts for å få unike personalitet. Dette gjør at implementeringen er enklere for andre objekter som påvirker sin foreldreklasse.

Foreksempel vi kan legge til en ny metode i foreldreklassen slik at denne klassen utfør metoden som er relatert til andre objekt.

5 EKSPERIMENTER & RESULTATER

Vi har utforsket boids sin oppførsel og deres data i stimulasjonen, med ulike avstand og vinkel for å definere området for prinsippene. Dersom boids 360° vinkel, vil det føre til ingen leder for andre boids, dermed har vi begrense vinkelen. For avstand vil også påvirke deres syn og mellom rom, derfor har jeg valgt at boids sin avstand til andre skal være den minste.

I en situasjon, men flere boids som finnes i alle tre prinsippene området, så vil dette algoritme ikke prioriteres for hvilket regla skal reagere først eller sist. Den vil da oppdater retningen fra alle tre prinsippene som kan tolke som en sum. Det vil gjøre at boids sin posisjon vil endres litt kraftigere, dermed har jeg definert den nye retning for separate skal være invers vektor slik at det balansere med andre endringene.

Det som er nyttig å legge merke til for bruk av pygame, er at vil kjører i while-løkken, dermed for alle andre funksjonene vil også følger dermed er det ikke behov for bruk av while-løkken for å endre noe gradvis.

6 KONKLUSJON

I denne rapport har vi gått gjennom bakgrunn for boids, og hvordan vi kombinere matte formular med stimulasjonen. Så viser vi vårt plan for hvordan vi ønske å få alt til å fungere, deretter under implementasjon vi har snakket oss inn for de utfordringene og løsningene.

Totalt sett så har jeg blitt utfordret siden starten av implementering for denne stimulasjonen, og det er fra boids bevegelsen til normalisering av vektor.

Det som sliter meg mest, er å sette opp en struktur for implementering og finne en stil og logikk for å tilpasse stimulasjonens regler. Når man har funnet en struktur som fungerer så blir alt lettere.

8 REFERANSER

[1] For vektor og vinkel formular.

https://www.matematikk.org/artikkel.html?tid=192383&within_tid=154821

[2] <https://www.youtube.com/watch?v=6dJlhv3hfQ0&t=104s>

0:54 - 1:06 : For synsområdet ideen

1:25 - 2:30 : For prinsippene ideen og hvordan resultatet vil se ut, slik at jeg kan tilnærme den.

[3] Hentet matte formula for Felles retning.

https://vanhunteradams.com/Pico/Animal_Movement/Boids-algorithm.html#Alignment

[4] Hentet matte formula for senter av mass.

https://vanhunteradams.com/Pico/Animal_Movement/Boids-algorithm.html#Cohesion