 2VSIM101-1 22H Mappe-Oppgave

**Kandidatnr**: 811

Innhold

[Introduksjon 1](#_Toc115938591)

[Metode 1](#_Toc115938592)

[Resultat 3](#_Toc115938593)

[Diskusjon 5](#_Toc115938594)

[Git link: 5](#_Toc115938595)

# Introduksjon

Denne oppgaven er brancha fra Oblig 3, som var samarbeida med Adam Aske. Det gjelder denne Mappe-oppgaven også, ettersom dette var tillat.

I dette prosjektet skal man simulere fysikken innad et geografisk område, dette vil både visualisere selve fysikken som fungerer på ulike legemer og konkretisere fysikken, slik at man vet hvilke formler og lover som fungerer på disse legemene.

# Metode

Samling av høydedata til vertexer.

Først ble det samlet høydedata fra hoydedata.no, det ble valgt et vilkårlig jorde i Ringerike kommune, Viken Fylke. Denne høydedataen er målt ved hjelp av LIDAR, som kort forklart er måling av høyde og lengde ved hjelp av laser. Denne informasjonen blir så omgjort til en punktsky som er et 3D-Dimonsjalt-sett med koordinater og punkter. Denne dataen kommer i form av en LAS fil som er industristandarden innenfor landmåling og lasermåling.

Et bilde som inneholder tekst

Automatisk generert beskrivelseVidere ble LAS-fila gjort om til leselig data ved i form av en txt-fil. Dette ble gjort med et ferdiglaget program som konverter denne dataen.

Disse tallene måtte nå bli gjort om til koordinater, og det er nå den faktiske kodingen begynner.

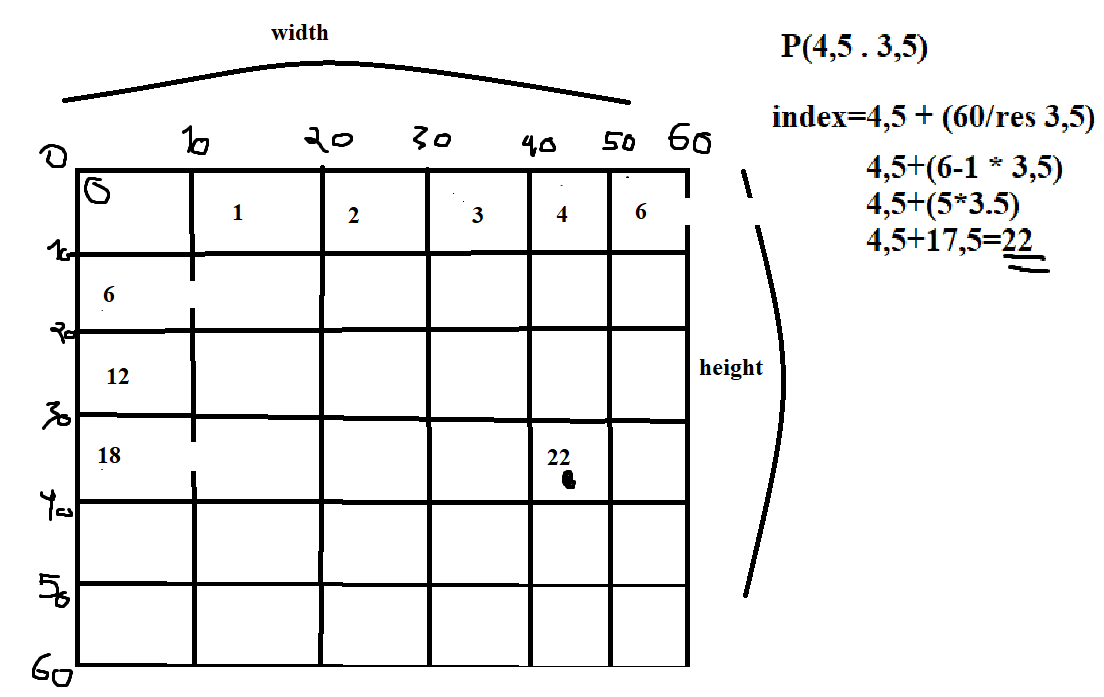
Denne txt fila ble puttet inn i QT Creator, der C++ ble brukt. Måten dette ble gjort på er at programmet går gjennom hver linje og finner antall mellomrom per linje for å separere tallene i hver sin blokk. Disse tallene ble så gjort om fra strings til floats og doubles etter størrelsen på tallene.

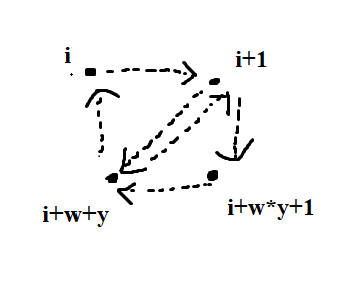
Denne fila hadde 1763401 punkter.

Nå var problemet at disse punktene var veldig langt unna hverandre på hver akse. Derfor ble de kortet ned ved å dele på 100000, 1000 og 100. for å få alle ned til 10er plasser.

Etter dette blir det lagd Convex Hull, dette blir gjort med en struct som har fire koordinater., hver av disse Quadsene, legger til en Vertex i mVertices som har sentrum av sin egen Quad som posisjon.

For å plassere riktig punkt i riktig quad blir rekkefølgen i mQuads og koordinatene til punktet brukt. X koordinaten til quadden finner bredde indexen. . For å finne høyde-indexen, bruker man bredden av convex hullet ganget med y-aksen i quadden, dereter plusser man på bredde indexen.





glDrawElements krever at man indekserer vertexene, som er 2 triangler per Quad,

dette blir indeksert slik:

Regndråper

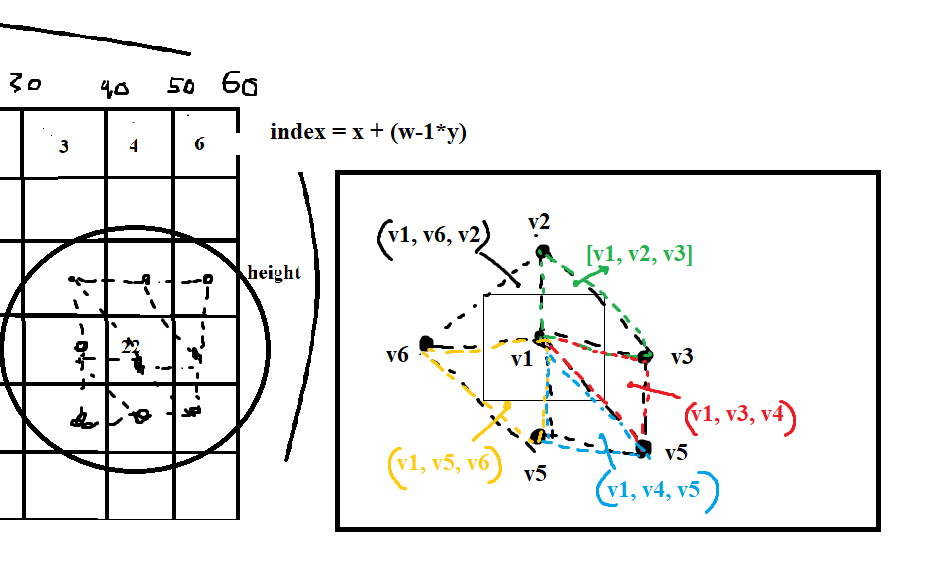
Regndråper er bare kuler fra hentet fra rollingBall klassen. det blir lagd en vector som henter RollingBall klassen fra Rollingball.h som peker mot renderwindow.h. Fra der får regnet et satt maks mengde regndråper slik at det ikke skal spawne en altfor stor mengde som påvirker ytelsen.

Videre for å aktivere regnet, skjer dette i event->key funksjonen som i dette tilfellet er (event->key() == Qt::Key\_I).

Når denne funksjonen blir aktivert starters det er for loop som er begrenset av anttal regdråper som har lov til å spawne. Inni denne loopen blir det lagd en 3DVector som lager tilfeldige spawn-points til regnet. Etter regnet spawnet blir fysikk lagt til ballene.

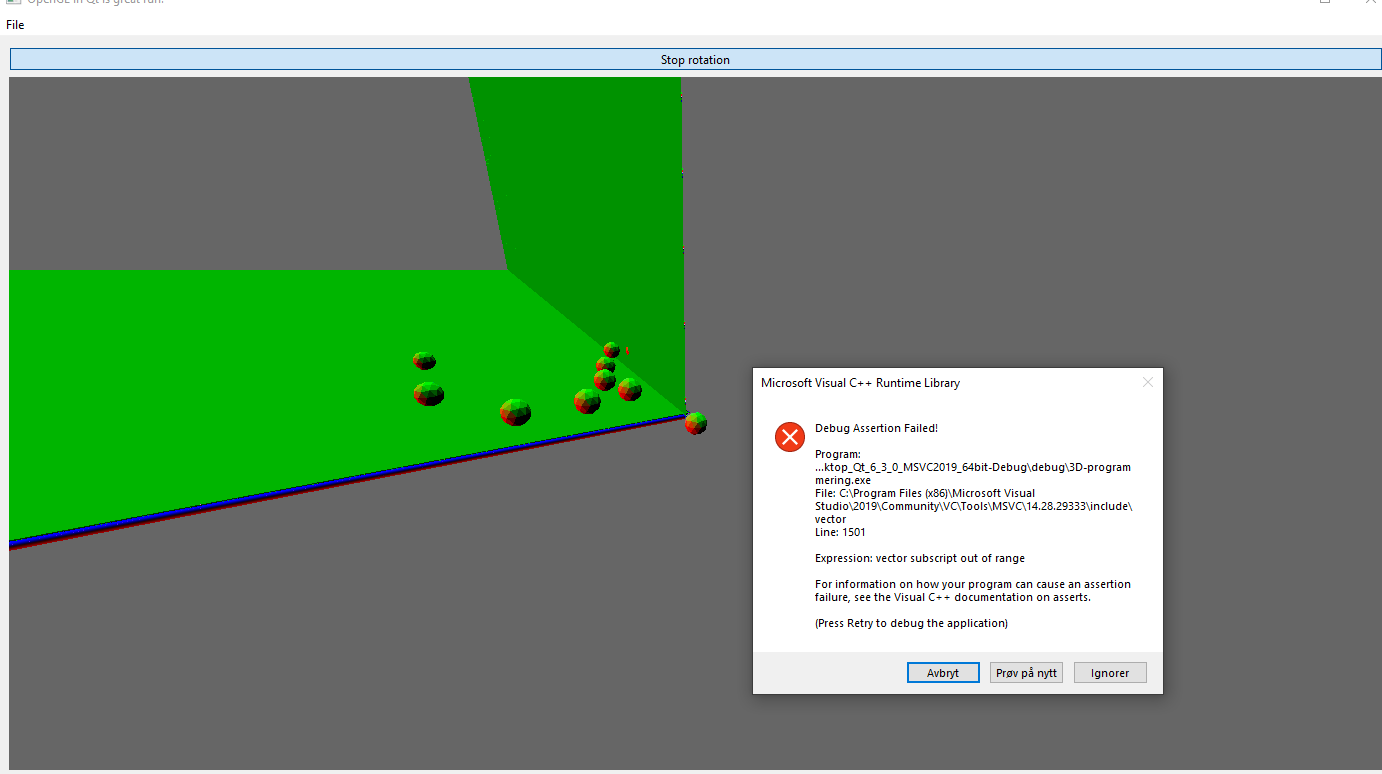
Barysentriske koordinater

Barysentriske koordinater går gjennom alle trekantene for å sjekke høyden til Regnet, den gjør dette gjennom getHeight() funskjen, og sjekker disse trekantene for vert punkt og iterer gjennom dem, på samme måte på den gjør i Quad bredde x høyde funksjon. Ballen trenger denne funksjon for å få fysikk på overflaten, slik at systemet vet hvilken høyde og trekant ballen befinner seg i.



# Resultat

Flaten viser en helt flat flate som er lagd fra en punktsky, prøvde med andre punktskyer, men de samme problemene artet seg, dvs, rare overflater som legger seg i én retning.

Flaten 


Figur 1: flaten fra punktsky

Regnet fungerer helt fint i fritt fall men programmet returnerer en vector out of range i det ballene treffer underlaget og y verdien deres blir mindre enn 0 , dette kan tyde på at det er noen feil i de barysentriske koordinatene, som returner en feil verdi.

B-splines

Den mattematiske delen av b splines fungerer ettersom qDebug returnerer riktig kontrollpunkt og kurver. Men har samme problemet som barysentriske der aktivering av de visuelle punktene og kurvene returner en vector out of range.

# Diskusjon

I denne oppgaven har jeg prokrastinert betraktelig mye mindre enn det jeg har gjort før, selv om mye av programmet breaka før innlevering, dette er også noe man burde vært klar over kan skje når man har koda 2 snart 3 år.

Matte-delen var ikke fullt så ille ettersom jeg har erfaring fra fysikk 1 og 2 fra vgs.

Samarbeidet med Adam har vært fint som alltid der vi føler vi har lært mye, men må personlig bli mye bedre på troubleshooting og debugging.

# Git link:

https://github.com/Meijkiqman/VSIM\_Oblig\_3