Visualisering og Simulering Mappeoppgave

Kandidatnummer: 869

Innhold

[Introduksjon 1](#_Toc115825212)

[Metode 1](#_Toc115825213)

[Geodata til SurfaceMesh 1](#_Toc115825214)

[Regn 1](#_Toc115825215)

[Kvadratisk BSpline 1](#_Toc115825216)

[Resultater 1](#_Toc115825217)

[SurfaceMesh 1](#_Toc115825218)

[Diskusjon 2](#_Toc115825219)

# 

# Introduksjon

For å gjøre denne oppgaven har jeg brukt Oblig 3 som startpunkt.

# Metode

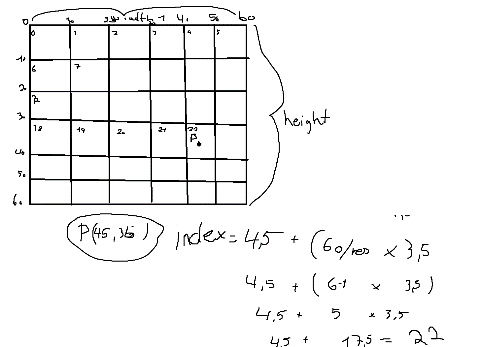
## Geodata til SurfaceMesh

Jeg starter SurfaceMesh::SurfaceMesh med å lese filen «fullData.txt» som inneholder punktene, hver linje består av x, y og z koordinatene. Løkken kjører så lenge std::getline returnerer sann. Getline putter linjen inn i line variablet. Hver karakter i stringen blir gått gjennom, den henter tallet hver gang den møter på mellomrom eller siste karakter. Std::stof blir brukt for å gjøre det fra en string til en float, og tallene blir puttet inn i vectoren points.

For å justere de ville tallene så går jeg gjennom hvert eneste punkt og justerer.

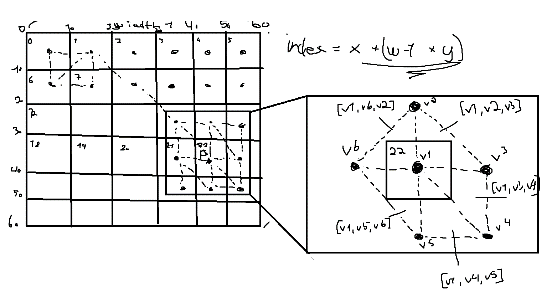
For å lage et convex hull så har jeg en struct, Quad, som har fire hjørner og en høyde verdi. Quad’ene blir laget i en dobbel løkke så jeg får riktig høyde og bredde.

Så går jeg igjennom alle punktene og finner hvem Quad de tilhører. For å riktig Quad bruker jeg funksjonen index = x + (widt -1\* y), hvor width er bredden på meshet. Quad’en får så z-koordinatet til punktet. Quad::AddHeight legger til verdien og øker antall punkter i seg. Quad::GetHeight returnerer den totale høyde delt på antall punkter.

Tegning for index = x + width-1\*y

Jeg lager så en vertex per Quad og de får sentrum og høyden til quaden som posisjonen sin.

For å finne hvem Quad hvert punkt tilhører bruker jeg en funksjon «index =x + (width-1 \* y)». Dette fungerer når jeg har 1 som i resolusjon, altså 1 quad er 1x1 meter. Denne funksjonen fungerer ikke for andre resolusjoner.



Denne tegningen er visualisering for SurfaceMesh::GetHeight og SurfaceMesh::GetBarycentric. Ettersom alle vertexene er plassert i sentrum av hver quad, så finner jeg hvem quad punktet tilhører og tar bruker det som indexen for vertexen i vectoren. Så finner jeg hvem trekanter punktet kan være på utifra quaden det er i og tester de riktige trianglene.

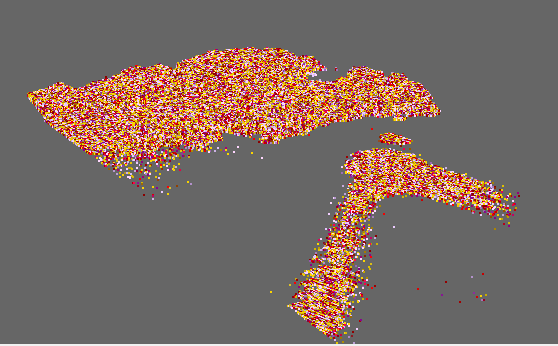
## Regn

## Kvadratisk BSpline

# Resultater

## SurfaceMesh

Dette er bilde av vertexer som er laget av «points» i SurfaceMesh::SurfaceMesh og tegnet som punkter. Fargene kommer av en cos(sin(tan(i))), sin(cos(tan(i))) og tan(sin(cos(i))) funksjon. Disse er plassert riktig i henhold til område jeg hentet høydedataen fra. Der det ikke er vertexer er det vann.



Et bilde som inneholder laser, vektorgrafikk

Automatisk generert beskrivelse

Dette bilde viser annahver triangel blir tegnet og alle vertexene har 0 som y koordinat(render annahver for å vise at den tegner riktige trekanter).

# Diskusjon

Hva som kunne vært bedre, hva var vanskelig, etc