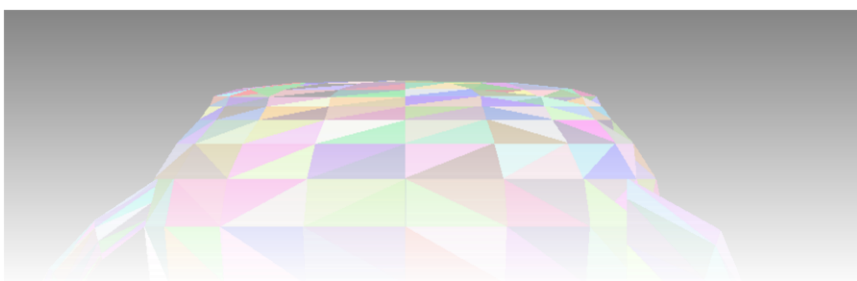




Innlevering Oblig 3



By

Alexander Einshøj
INF-1100
Å. A. Kvalnes

Introduksjon

I denne oppgaven har jeg implementert kode for å tegne fylte trekanter i et vindu på skjermen. For å få til dette har jeg anvendt utdelt kodemateriell og egenskrevet kode i programmeringsspråket C. Oppgaven er skrevet i Notepad++ på Windows. Jeg har laget en rapport for å vise hvordan jeg har utført oppgaven.

ScaleTriangle

```
Triangle->sx1 = (int) * (triangle->x1(int) * triangle->scale(float));
```

I denne funksjonen er koordinatene **integers**, og **scale** er float. Vi vil at svaret skal være integer, og jeg har derfor skrevet **(int)** på høyresiden hele veien, slik at **(int) (triangle->x1 * triangle->scale) = int**. Slik jeg har forstått det vil resultatet i denne situasjonen bli **int** uansett.

TranslateTriangle

```
Triangle->sx1 = triangle->sx1 + triangle->tx;
```

Denne koden er ganske rett frem, da den ved calling av funksjonen bare reassigner `triangle->sx1-3` og `triangle->sy1-3` ved å legge til translation-koordinatene.

CalculateTriangleBoundingBox

```
triangle->bx = triangle->sx1;
if(triangle->bx > triangle->sx2){
    triangle->bx = triangle->sx2;
}
if(triangle->bx > triangle->sx3){
    triangle->bx = triangle->sx3;
}
```

Dette er metoden jeg har brukt for å finne den minste verdien av **(bx,by)**, **w(BoundingBox)** og av **h(BoundingBox)**. Jeg setter her et av koordinatene til å ha laveste verdi, og går gjennom de andre koordinatene for å sjekke om de har lavere verdi, ved å bruke masse if-tester.

FillTriangle

I denne funksjonen var den vanskelige delen for meg å kun fylle trekantene, ikke utenfor.

Den første for-loopen går gjennom alle y-verdiene fra øverst til venstre i boundingbox og til den treffer bunnen av boundingboxen nederst til venstre.

Den andre for-loopen går gjennom bredden på boundingboxen på samme måte som den andre går gjennom høyden. Det er i den andre for-loopen det viktige skjer. Den finner fargen på pixelene på skjermen, og hvis de matcher

TRIANGLE_PENCOLOR, settes den x(og dermed også y)-verdien til **startx** og **endx**. Neste gang den finner **TRIANGLE_PENCOLOR** vil det være andre gang, altså er det slutten av trekanten, og fill-funksjonen skal slutte her.

```

        int x, y;

    for(y = triangle->by; y <= triangle->by + triangle->bh; y++){
        int startx = 0;
        int endx = 0;

        for(x = triangle->bx; x <= triangle->bx + triangle->bw; x++){
            if(GetPixel(screen, x, y) == TRIANGLE_PENCOLOR && startx == 0){
                startx = x;
                endx = x;
            }
            else if(GetPixel(screen, x, y) == TRIANGLE_PENCOLOR) {
                endx = x;
            }
        }
        DrawLine(screen, startx, y, endx, y, triangle->fillcolor);
    }
}

```