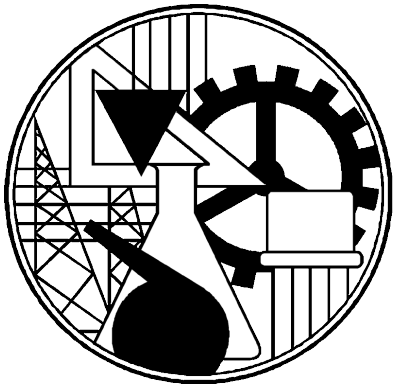


ISEL

Instituto Superior Engenharia de Lisboa

1ª Serie

Engenheiro: Carlos Guedes

Engenharia Informática e de Computadores

Semestre de Inverno 2011/2012

Ana Correia - 31831

Diogo Cardoso - 32466

João Silvestre - 32766

01/11/11

Índice

Introdução

O objectivo deste trabalho é que o grupo aplique os conhecimentos obtidos nas aulas sobre algoritmos básicos de preenchimento de polígonos convexos e côncavos. Também como objectivo que os elementos do grupo adquiram prática na utilização de transformações geométricas e a sua composição, assim como na definição de grafos de cena

Parte Teórica

1º Exercício

Alínea 1.1

Na implementação do algoritmo de *Bresenham* fornecida, já resolvia o problema de o declive ser superior a 0 e inferior a 1 (0 <= m <= 1). O objectivo deste exercício era completar o algoritmo de forma que consiga resolver os outros 7 casos possíveis. Para resolver os casos em que o declive é superior a 1 é foi trocar a operação de iteração, pois o crescimento é vertical, iterando-se então sobre X em vez de Y, sendo necessário igualmente, ajustar o cálculo do erro para que este seja feito através da distância à coordenada que não se encontra a ser iterada (X quando Y é iterado, Y quando X é iterado). Quando o declive é negativo é necessário trocar as fórmulas para que o crescimento seja negativo ao invés de positivo.

Alínea 1.2

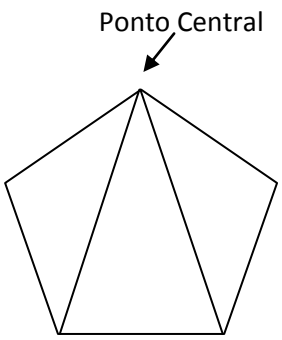
Para corrigir esta função foi apenas necessário alterar o método *writeCirclePixel* para que este escreve-se os pixéis contando com o deslocamento em relação ao ponto central do círculo, ou seja, antes este método escrevia os pixéis sempre em relação à origem, ao alterar a coordenada onde ele escreve somando a coordenada do ponto central este passa a escrever em relação ao ponto central.

Alínea 1.3

Neste exercício é usado um algoritmo que usa coordenadas baricêntricas, usando estas sabe-se que um pouco se encontra dentro do triângulo se as coordenadas forem todas positivas. As coordenadas foram calculadas recorrendo a uma conversão de um ponto em coordenadas cartesianas para coordenadas baricêntricas, estas tem a vantagem de estarem independentes da escala do triângulo. Chegando às fórmulas finais é apenas necessário implementa-las e proceder ao cálculo das mesmas sempre que se pretenda saber as coordenadas baricêntricas de um determinado ponto. As coordenadas são depois usadas para determinar a cor a ser usada no *pixel* para ter o efeito de cores desejado.

Alínea 1.4

Para a implementação do *fillPolygon* fez-se uma triangulação do polígono convexo e usada a implementação da alínea anterior para pintar os triângulos. Esta divisão baseia-se no conhecimento que um qualquer segmento de recta entre dois vértices de um polígono convexo estará sempre completamente dentro do mesmo. Para se definir um triângulo é necessário 3 pontos, logo escolhendo um ponto central e ignorando os dois vértices "adjacentes" a este sabe-se que se terá n – 2 triângulos (sendo n o número de vértices). Procedendo então a divisão em n – 2 triângulos iremos obter algo semelhante ao exemplo seguinte:



Mandando então pintar estes triângulos iremos obter um polígono convexo completamente pintado.

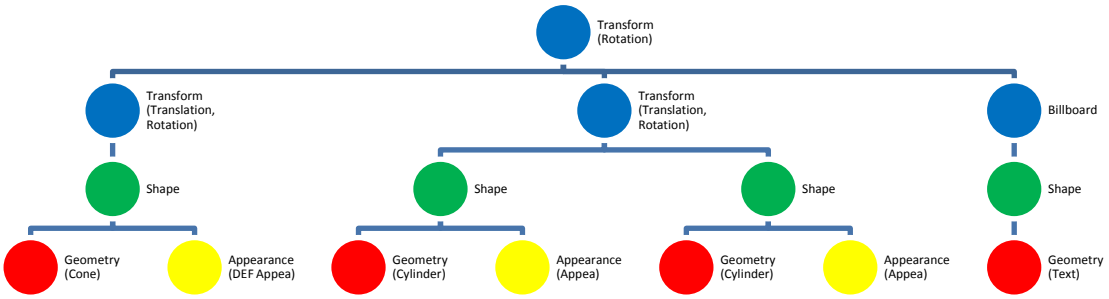
- **Alínea 1.5**

Para a implementação do método *fillConcavePolygon* foi também feita uma triangulação do polígono, esta foi feita recorrendo ao algoritmo "*Ear Cutting*". Este algoritmo baseia-se na definição de uma *ear*, que é um vértice em que os dois vértices adjacentes formam um segmento de recta (ou diagonal) que se encontra por completo dentro do polígono. Caso um determinado vértice seja uma *ear* é criado um triângulo usando esses 3 vértices e o vértice que é considerado *ear* é removido do polígono formando um "novo" polígono. Este processo é repetido até que o polígono esteja reduzido a 3 vértices, sendo que estes últimos 3 formam também um triângulo.

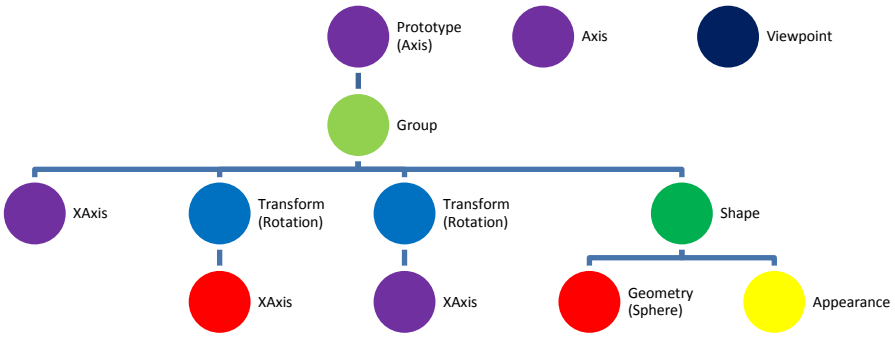
Para verificar se um vértice é uma *ear* verifica-se se existe algum dos outros vértices dentro do triângulo, no entanto, isto não é suficiente pois existem casos em que se vai formar um triângulo completamente fora do polígono, para isso é feito o produto "cruzado" e caso o resultado deste seja positivo significa que temos um ângulo interno superior a 180º que diz que o triângulo encontra-se fora do polígono.

➢ 2º Exercício

○ Alínea 2.1 XAxis Prototype – Gráfico de Cena



○ Alínea 2.1 Axis Prototype – Gráfico de Cena



Conclusão

Esta série de exercícios fez com que o grupo conseguisse adquirir conhecimentos básicos de como são feitas algumas das coisas que as bibliotecas gráficas fazem. Serviu também para adquirir conhecimentos básicos em VRML que irá ser usado no próximo trabalho, servindo assim como rampa de lançamento para o mesmo.