

Sistemas de coordenadas

Introdução

Em computação gráfica grande quantidade de dados são manipulados e esses dados são altamente quantificados em vários sistemas de coordenadas. Os dispositivos gráficos possuem um conjunto de pontos ou pixels endereçáveis e os padrões gráficos são produzidos ascendendo (ou pintando) e apagando esses pontos.

Coordenadas do dispositivo

Os pixels ou pontos podem ser endereçados por dois inteiros, o número horizontal do pixel (dcx) e o número vertical do pixel (dcy).

Assim, temos:

$$0 \leq dcx \leq ndhm1 = ndh - 1$$

$$0 \leq dcy \leq ndvm1 = ndv - 1$$

O ponto com endereço (0, 0) é a origem do sistema de coordenadas do dispositivo, podendo estar no canto superior esquerdo ou no canto inferior esquerdo.

As coordenadas dcx e dcy são números inteiros, e o número de pontos endereçáveis varia de dispositivo para dispositivo e em um sistema gráfico, como anteriormente visto, existem vários dispositivos de entrada/saída.

Coordenadas de dispositivo normalizado (NDC)

Com o propósito de realizar uma abordagem padronizada para os diferentes dispositivos gráficos, foram propostas as coordenadas de dispositivo normalizado. O sistema de coordenadas utiliza os valores $ndcx$ e $ndcy$ para endereçar posições no dispositivo. Esses valores, a diferença das coordenadas do dispositivo (dcx , dcy), assumem valores reais e são usualmente definidos no intervalo de 0 a 1. Assim:

$$0 \leq ndcx \leq 1$$

$$0 \leq ndcy \leq 1$$

A vantagem do NDC é que os padrões gráficos podem ser discutidos usando um sistema de coordenadas padrão independente de qualquer dispositivo em particular.

Por outro lado, será necessário transformar os dados gráficos descritos em coordenadas de dispositivo normalizado para valores de um dispositivo particular. O mapeamento de coordenadas no sistema NDC (reais) para coordenadas de dispositivo é linear e pode ser implementado em uma rotina (`ndc_to_out`). Esta relação é dada por:

$$dcx = \text{round} (ndcx * ndhm1)$$

$$dcy = \text{round} (ndcy * ndvm1)$$

Devem ser implementadas rotinas semelhantes para os dispositivos de entrada.

Sistemas de coordenadas físicas

Um outro sistema de coordenadas útil em computação gráfica é o sistema de coordenadas físicas. Neste sistema as coordenadas *pcx* e *pcy* representam distância física nos eixos x e y. As unidades usadas para medir essas distâncias podem ser polegadas ou milímetros. A transformação de coordenadas físicas para coordenadas do dispositivo é dada por:

$$dcx = \text{round} (ndhm1 * pcx / \text{width})$$

$$dcy = \text{round} (ndvm1 * pcy / \text{height})$$

Coordenadas do usuário ou do mundo

Este sistema é um sistema cartesiano de coordenadas (*x*, *y*) de qualquer tamanho selecionado pelo usuário para trabalhar em sua aplicação. Os parâmetros que definem os valores possíveis para *x* e *y* definem uma área retangular no espaço cartesiano:

$$x_{\min} \leq x \leq x_{\max}$$

$$y_{\min} \leq y \leq y_{\max}$$

O usuário deve informar ao programa gráfico que amplitudes de x e y devem ser mapeadas no dispositivo de saída gráfica. Essas amplitudes definem uma janela (window) no espaço bidimensional abstrato.

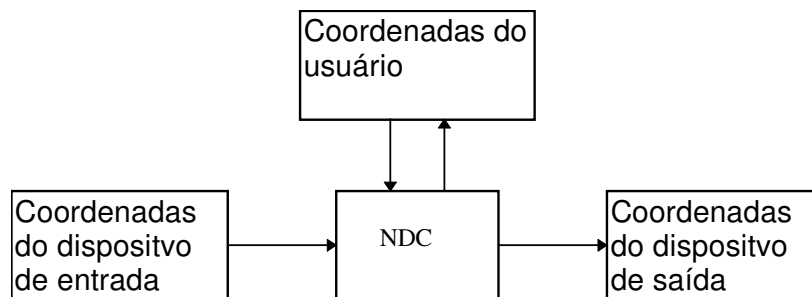
Transformação de coordenadas do usuário para NDC

Esta transformação é chamada de transformação de normalização e ela é realizada pelas seguintes equações:

$$ndx = (x - xmin) / (xmax - xmin)$$

$$ndy = (y - ymin) / (ymax - ymin)$$

Rotinas que transformam dados em coordenadas NDC, do usuário ou físicas para coordenadas do dispositivo de saída devem ser fornecidas com as correspondentes rotinas para conversão dos dados de entrada.



Correção da relação de eixos

Na maioria dos dispositivos de saída o retângulo de exibição não é quadrado ou seja:

$$\text{width (largura)} \neq \text{height (altura)}$$

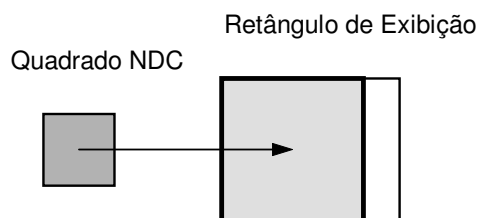
Assim, um quadrado quando digitalizado será distorcido para um retângulo em coordenadas NDC e distorcido novamente pela transformação de saída. Se a saída é direcionada a um outro dispositivo, as proporções serão novamente alteradas.

Normalmente, em computação gráfica deseja-se que as figuras desenhadas em coordenadas do usuário sejam exibidas nas mesmas proporções no dispositivo de saída.

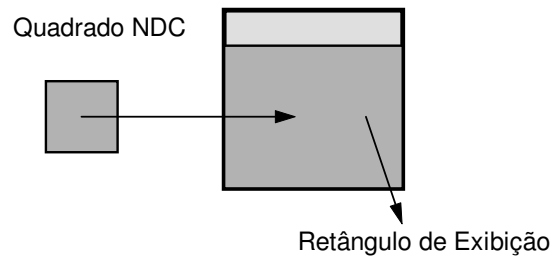
O caso mais simples é quando o usuário define as figuras em uma área quadrada em seu sistema de coordenadas que é mapeada exatamente no espaço quadrado do NDC. Quando o usuário define um gráfico (por exemplo, um quadrado), então, a rotina de conversão para coordenadas de dispositivo normalizado, o converterá em um quadrado em coordenadas NDC. Porém, se o dispositivo de saída não tem um aspecto físico igual a unidade, o mapeamento para as coordenadas deste dispositivo apresentará distorções, desenhando um retângulo como saída.

Três métodos básico para preservar as proporções durante a saída são vistos a seguir.

No primeiro, o quadrado do NDC é mapeado no maior quadrado físico dentro do retângulo de exibição do dispositivo de saída. Isso preserva as proporções e nenhum desenho é perdido (por ser mapeado fora do retângulo de exibição) . Porém, não é realizado a utilização total do retângulo de exibição.



No segundo método, o quadrado de coordenadas NDC é mapeado no menor quadrado que contém a área de saída. Assim, toda a área de exibição do dispositivo é utilizada mas, partes das figuras gráficas do quadrado NDC podem ser perdidas uma vez que são mapeadas fora do retângulo de exibição.



O terceiro método tem uma abordagem um pouco diferente: estende os limites das coordenadas do usuário de forma mínima para corresponder ao aspecto físico do dispositivo de saída. Assim cada vez que um novo dispositivo de saída é selecionado os limites da janela definida em coordenadas do usuário devem ser reajustados. Após muitas dessas mudanças a janela pode aumentar muito, ou seja, o desenho aparece muito pequeno na saída. Além disso, se após selecionar um segundo dispositivo de saída retornar ao primeiro dispositivo, a saída não apresentará o mesmo tamanho. Esse efeito pode ser evitado copiando-se os limites da janela do usuário para uma outra área de dados globais quando a janela é definida, e reinicializar a janela para esses valores antes de estender novamente os limites.

Os três métodos acima descrito podem ser utilizados se o aspecto físico do dispositivo de entrada são diferentes da unidade (aspecto físico do NDC é 1).

O Sistema de Coordenadas do Mundo e as Coordenadas de Dispositivo Normalizado

As coordenadas de dispositivo normalizado (NDC) proporcionam uma abordagem uniforme para o desenvolvimento de “drivers” para os dispositivos, porém, podem ser pouco naturais e confundir o programador de um aplicativo.

Um problema é que os objetos são descritos utilizando valores entre 0 e 1 (NDC) mas, o programador pode desejar desenhar um objeto que tem 12 metros de largura por 5 metros de altura. Mudanças de escala são necessárias para converter as coordenadas naturais do objeto nas dimensões forçadas das coordenadas de dispositivo normalizado.

O objeto parece ter um conjunto de dimensões e o desenho outro, o que deixa clara a diferença entre modelagem e visualização.

Modelagem é o processo de criação e manipulação de um modelo de um objeto ou sistema. Em computação gráfica é, geralmente, utilizado um modelo geométrico, que é uma descrição do objeto que proporciona uma representação numérica da sua forma, seu tamanho e várias outras propriedades.

Visualização é o processo de desenho de uma vista do modelo no visualizador. A descrição geométrica do objeto proporcionada pelo modelo é convertida em um conjunto de primitivas gráficas que são mostradas em um local desejado da superfície de visualização. O mesmo modelo pode ser utilizado para muitas vistas diferentes e a escolha da vista vai determinar o tamanho e a posição das primitivas.

A modelagem é mais convenientemente realizada no sistema de coordenadas do usuário ou coordenadas do mundo.

Três questões surgem quando se deseja produzir uma visualização de um modelo: quanto do modelo deveria ser desenhado, onde deveria aparecer no dispositivo de saída e como as coordenadas do mundo serão convertidas nas coordenadas de dispositivo normalizado.

O programador necessita portanto flexibilidade e controle sobre:

- que parte do objeto desenhado
- onde é desenhado
- de que tamanho é desenhado

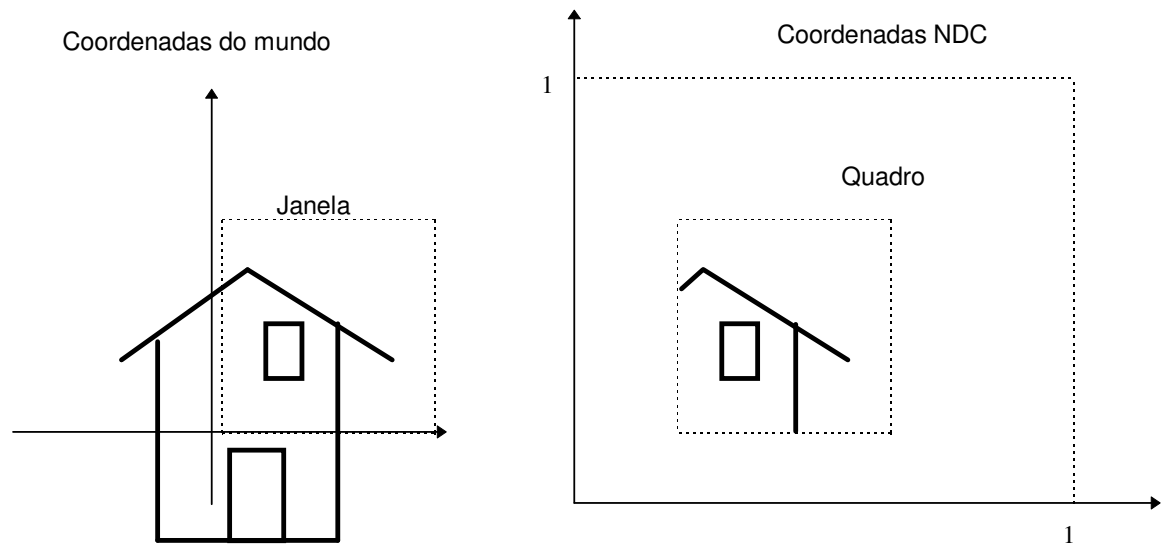
Janelas e Quadros (*Windows e Viewports*)

Uma vez estabelecida a parte do programa que desenvolve o modelo, o programador deve especificar uma vista do mesmo. Isso é realizado através de dois retângulos, uma janela dada em coordenadas do mundo (usuário) e um quadro dado em coordenadas de dispositivo normalizado.

A janela define a fração do modelo que deve ser desenhada. As partes que estão fora são recortadas e não aparecem. Algoritmos de recorte

examinam cada linha ou arco do modelo (primitivas) e determinam qual parte está dentro da janela.

O quadro (viewport) define em que lugar, dado em coordenadas de dispositivo normalizado, as partes visíveis devem ser desenhadas.



Os elementos geométricos dentro da janela são mapeados proporcionalmente no quadro. A janela e o quadro, não necessariamente devem ter a mesma relação de aspecto (altura/largura) e, se diferentes, ocorrerão distorções na figura desenhada. Geralmente, essa relação de aspecto da janela e do quadro são iguais.

