

Computación Gráfica

Práctica

Ejercicio

Gabriela Rojas Castro¹
UNIVERSIDAD NACIONAL DE SAN AGUSTIN
Escuela Profesional de Ciencia de la Computación
Computación Gráfica

1

1. 1.6 Ejercicios

En el caso de los sistemas de coordenadas, como se muestra en la Figura 1.3, los procedimientos *inp a ndc*, *ndc a user*, *user to ndc* y *ndc a dc*, que transforman datos entre los varios sistemas de coordenadas. Repita el ejercicio asumiendo que el intervalo de variación del sistema NDC va de:

1. -1 a +1 (coordenadas normalizadas centradas)
2. 0 a 100

```
1 | - inp_to_ndc
2 |
3 |     ndcx = dcx / ndhml;
4 |     ndcy = dcy / ndvml;
5 |
6 | - ndc_to_user
7 |
8 |     x = ( ndcx * (xmax - xmin) ) + xmin;
9 |     y = ( ndcy * (ymax - ymin) ) + ymin;
10 |
11 |     reemplazando...
12 |
13 |     x = ( ndcx * (1 - (-1)) ) + (-1);
14 |     y = ( ndcy * (1 - (-1)) ) + (-1);
15 |
16 | - user_to_ndc
17 |
18 |     ndcx = (x - xmin) / (xmax - xmin);
19 |     ndcy = (y - ymin) / (ymax - ymin);
20 |
21 |     reemplazando...
22 |
23 |     ndcx = (x - (-1)) / (1 - (-1));
24 |     ndcy = (y - (-1)) / (1 - (-1));
25 |
26 | - ndc_to_dc
27 |
28 |     dcx = round(ndcx * ndhml);
29 |     dcy = round(ndcy * ndvml);
30 |
31 | (ii) 0 a 100
```

```

32 |
33 | - inp_to_ndc
34 |
35 |         ndcx = dcx / ndhml;
36 |         ndcy = dcy / ndvml;
37 |
38 | - ndc_to_user
39 |
40 |         x = ( ndcx * (xmax - xmin) ) + xmin;
41 |         y = ( ndcy * (ymax - ymin) ) + ymin;
42 |
43 |         reemplazando...
44 |
45 |         x = ( ndcx * (100 - (0)) ) + (0);
46 |         y = ( ndcy * (100 - (0)) ) + (0);
47 |
48 | - user_to_ndc
49 |
50 |         ndcx = (x - xmin) / (xmax - xmin);
51 |         ndcy = (y - ymin) / (ymax - ymin);
52 |
53 |         reemplazando...
54 |
55 |         ndcx = (x - (0)) / (100 - (0));
56 |         ndcy = (y - (0)) / (100 - (0));
57 |
58 | - ndc_to_dc
59 |
60 |         dcx = round(ndcx * ndhml);
61 |         dcy = round(ndcy * ndvml);

```

2. Ejercicios parte 2

Ejercicio 2.4.2

Modifique el programa del ejercicio anterior para solicitar también el radio r del círculo que circunscribe el polígono. Haga que su programa estime el número n de vértices necesarios para que el polígono parezca una "buena aproximación" para un círculo. Utilizando los parámetros ndh , ndv , $width$, $height$ del dispositivo, obtenga una relación teórica entre n y r para la aproximación de círculos a través de polígonos.

SOLUCION:

- ndh - el número de posiciones direccionables horizontalmente.
- ndv - el número de posiciones direccionables verticalmente.
- $width$ - el ancho del rectángulo de visualización en mm.
- $height$ - la altura del rectángulo de visualización en mm.

La relación teórica para la aproximación de círculos mediante polígonos es:

$\text{angulo} = 2 \cdot \pi / n$ donde n es el número de vértices

y el radio por el cos y sin del ángulo

$\text{radio} \cdot \cos(\text{angulo})$

$\text{radio} \cdot \sin(\text{angulo})$

Ejercicio 2.4.3

Algunos dispositivos vectoriales ofrecen un conjunto de tres primitivas gráficas:

- pen: levanta la pluma del papel, o apaga el haz de electrones;
- pen down: coloca la pluma sobre el papel, o enciende el haz;
- locate (dcx, dcy): coloca la CP en un punto del rectángulo de visualización.

SOLUCIÓN

```

1      pen {
2          moveto (dcx, dcy)
3          drawto (dcx, dcy)
4      }
5
6      pen_down {
7          drawto (dcx, dcy)
8          drawto (dcx, dcy)
9      }
10
11     locate (dcx, dcy) {
12         moveto (dcx, dcy);
13     }

```

Ejercicio 2.4.4

Calcule las razones de aspecto (gráfica y física), y las resoluciones de área horizontal y vertical de una pantalla de TV a color estándar, donde:

- width = 42cm;
- height = 31cm;
- ndh = 546;
- ndv = 434.

SOLUCIÓN

```

1
2 1. resolucion horizontal: horiz_res = ndh/width = 546/420
3
4 2. tamaño punto horizontal: horiz_dot_size = width/ndh = 420/546
5
6 3. resolucion vertical: vert_res = ndv/height = 434/310
7
8 4. tamaño punto vertical: vert_dot_size = height/ndv = 310/434
9
10 5. total puntos direccionables: total_nr_dots = ndh.ndv = 546*434
11
12 6. resolucion de rea : rea_res = total_nr_dots/(width.height)
   = (546*434)/(420*310)
13
14 7. razon de aspecto grafico: aspect_ratio = vert_dot_size/horiz
   _dot_tamaño = (310/434)/(420/546)
15
16 8. razon de aspecto físico: physical_aspect_ratio = height/width
   = 310/420

```