Organización del Computador
Práctico de Taller 2: Algoritmos Gráficos en ARC

B. Gonzalez Kriegel - N. Wolovick

28 de mayo de 2001

1 ARCSim en el Laboratorio

En las máquinas del laboratorio esta instalado ARCSim, el cual se invoca mediante el *script* `ARCSim.sh`.

2 Problemas

2.1 Scan Converting Rectangles

rect1.asm Escriba una rutina ARC de *convención de paso de parámetros por registros* que a partir de los puntos (x_1, y_1), (x_2, y_2) y un color c, dibuje en el *framebuffer* los puntos $\{(x,y) \cdot x_1 \leq x < x_2 \wedge y_1 \leq y < y_2\}$ del color indicado. Suponga que se da la precondición $\{0 \leq x_1 \leq x_2 < MaxX \wedge 0 \leq y_1 \leq y_2 < MaxY\}$.
Escribir un programa utilizando esta rutina que dibuje un rectángulo entre $(MaxX/4, MaxY/4)$ y $(MaxX/2, MaxY/2)$ de color Gris50%.

rect2.asm Escriba una rutina que funciona de la misma manera que la anterior sólo que la precondición se relaja a $\{0 \leq x_1, x_2 < MaxX \wedge 0 \leq y_1, y_2 < MaxY\}$.
Muestre que su rutina no requiere orden entre los x's e y's escribiendo un programa que dibuje su bandera preferida.

rect3.asm Idem al punto anterior, solo que la precondición se relaja al máximo quedando $\{true\}$. Utilice recortado interno (*scissoring clipping*) o externo (recortado a fuerza bruta).
Muestre que su rutina funciona correctamente dibujando dos rectángulos de distinto color, uno de los cuales no cabe en el eje x y el otro en el eje y.

2.2 Scan Converting Lines

line1.asm A partir de [4, sec.3] obtenga una rutina de *convención de paso de parámetros por pila* que dibuja una línea de un color c entre los puntos $(0,0)$ y (a,b). Suponga la precondición $\{0 \leq b \leq a < MaxX\}$
Escriba un programa haciendo uso de su rutina que trace todas las líneas

$$\{((0,0), (MaxX - 1, 2^i)) \cdot 0 \leq i < \lceil log_2 MaxX \rceil\}$$

line2.asm Escriba una rutina como la anterior sólo que ahora recibe como parámetros dos puntos y un color $((x_1, y_1), (x_2, y_2), c)$ y dibuja una línea entre esos puntos. Suponga la precondición

$$\{0 \leq x_1 \leq x_2 < MaxX \wedge 0 \leq y_1 \leq y_2 < MaxY \wedge m = \frac{y_2 - y_1}{x_2 - x_1} \wedge 0 \leq m \leq 1\}$$

Hint: la rutina es como la anterior salvo una traslación.
Pruébela dibujando el par de líneas

$$((0,0), (MaxX/2, MaxY/4), Amarillo)$$
$$((MaxX/2, MaxY/4), (MaxX - 1, 3MaxY/4), Magenta)$$

2.3 Scan Converting Circles

circ1.asm A partir de [4, sec.4] implemente una rutina de *paso de parámetros por la pila* que dibuja un octavo de círculo de centro $(0,0)$, radio r y color c. Suponga la precondición $\{0 \leq r < min(MaxX, MaxY)\}$
Escriba un programa que dibuje todos los octavos de círculo de radio

$$r \in \{2^i \cdot 0 \leq i < \lceil log_2 min(MaxX, MaxY)\rceil\}$$

circ2.asm Escriba una rutina que a partir de un centro (x,y) y un radio r, dibuje un círculo completo. Suponga la precondición

$$\{0 \leq r \wedge 0 \leq x - r \wedge 0 \leq y - r \wedge x + r < MaxX \wedge y + r < MaxY\}$$

que dice que todos los puntos del círculos están dentro del *framebuffer*. *Hint*: utilice traslaciones y simetrías de octantes.
Pruebe la rutina dibujando los círculos

$$((MaxX/2, MaxY/2), min(MaxX, MaxY)/2, Rojo)$$
$$((3MaxX/8, 3MaxY/8), min(MaxX, MaxY)/8, Blanco)$$
$$((5MaxX/8, 3MaxY/8), min(MaxX, MaxY)/8, Blanco)$$

3 Entrega

Se deberá **entregar** 1 **de los** 3 **ejercicios** propuestos por grupo. Los grupos son los mismos que para el práctico de taller anterior.

Se deberá enviar por correo electrónico un archivo `.asm` por cada subitem del ejercicio elegido con el nombre indicado entre paréntesis cuadrados. Cada uno de estos archivos deberá ensamblar correctamente y efectuar la prueba que se pide para en cada caso. Los archivos deberán estar debidamente comentados a fin de que la corrección no se transforme en una tarea de *hacking* y que los autores sean capaces de comprender su código luego de que transcurran 15 minutos de haberlo finalizado. Los comentarios en ARCAssembler comienzan con **!** y se extienden hasta el final de la línea.

No se pide "estilo" alguno, pero se recomienda ser estructurado y prolijo a fin de minimizar los *bugs* que suelen inundar los códigos escritos en ensamblador. En [2] se muestran ejemplos con diferentes estilos de codificación.

No existen límites respecto a la cantidad de veces que el trabajo puede ser entregado hasta que haya sido aprobado. Cuando el trabajo está aprobado, se generará una calificación que comprenderá el **30% de la nota del segundo parcial**. El plazo de entrega es hasta el minuto antes de comenzar el segundo parcial. La dirección de correo es `odc@hal.famaf.unc.edu.ar` y el asunto del mail deberá ser exactamente "`Taller 2, Grupo n`". El cuerpo deberá contener el nombre completo de los integrantes junto al número de *LTP*. Los archivos deben ser adjuntados con el correo electrónico.

Referencias

[1] J. D. Foley and A. Van Dam, *Computer Graphics, principles and practice*, Second Edition, Addison Wesley, 1990.

[2] M. Kurban, N. Wolovick *Framebuffer para ARCSim*, CeIMAF, Mayo 2001.

[3] M. Murdocca and V. P. Heuring, *Principles of Computer Architecture*, Prentice-Hall, 2000.

[4] N. Wirth, *Drawing Lines, Circles and Ellipses in a Raster*, in *Beauty is our Business: a birthday salutation to Edsger W. Dijkstra*, Springer Verlag, 1990.

[5] N. Wolovick *Descripción de ARCSim y ARCAssembler*, CeIMAF, Mayo 2001.