

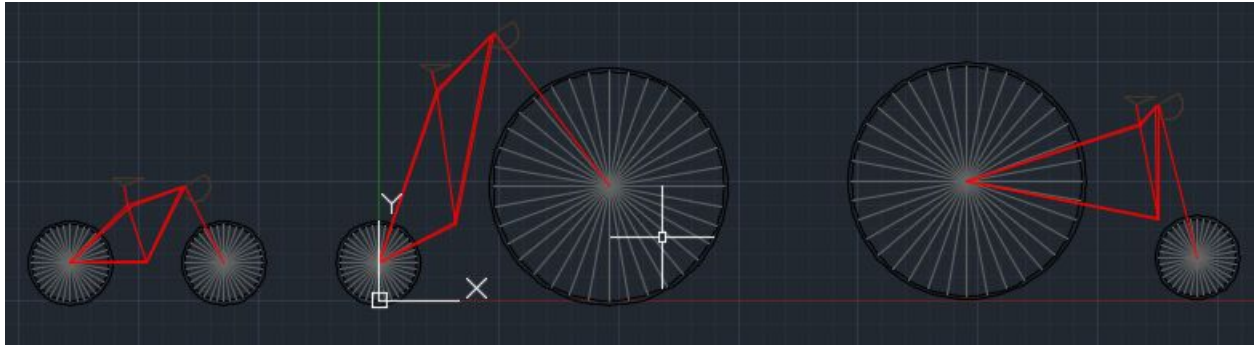


Tarea 1 - Dibujo libre

Computación gráfica
Miguel Angel Baquero

Dibujo libre de una bicicleta automáticamente en autocad

En la tarea se usan los conceptos vistos en clase para la creación de una script que dibuje bicicletas de manera automática, este programa solo le pide al usuario el tamaño de las ruedas y el color del marco y posteriormente realiza algunos cálculos simples para que la bicicleta esté bien dimensionada con respecto a las ruedas y obtenemos una de las tres bicicletas de las siguientes.



Objetivos

Los objetivos de esta tarea son:

- Aplicar los conocimientos obtenidos en la clase del lenguaje autolisp para realizar un dibujo libre.
- Crear una figura compleja mediante el uso de figuras básicas como círculos, líneas y arcos.
- Utilizar diferentes colores, para añadir más complejidad a la tarea.
- Usar un razonamiento matemático si se desea (Opcional)

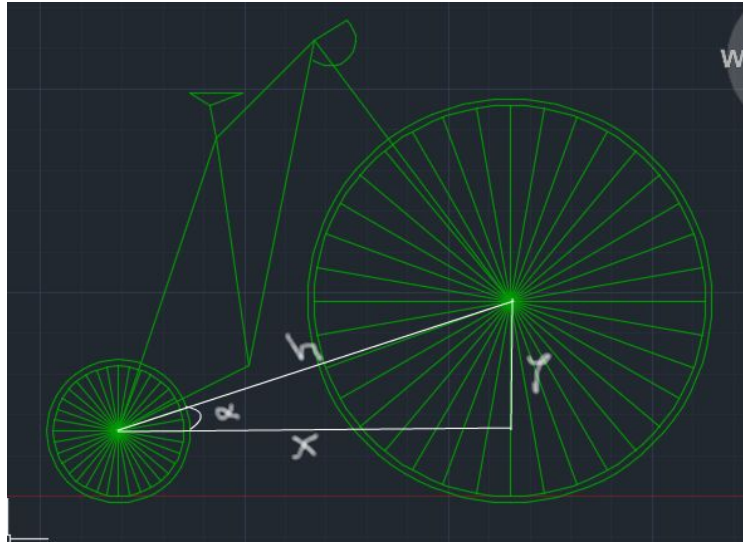
Matemática

Para mantener la consistencia en la bicicletas se necesitó hacer uso de algunos conceptos matemáticos, estos cálculos serán explicados a continuación:

Cálculo de punto bajo de marco bicicleta

Este punto es el punto bajo del marco de la bicicleta junto al centro de la primera rueda y no resulta ser un problema si la bicicleta tiene las dos ruedas del mismo tamaño, pero en caso

contrario nos toca calcular una inclinación a la segunda rueda como se puede ver en la siguiente imagen.



Este cálculo es bastante simple y se puede hallar las componentes de este ángulo usando razones trigonométricas.

$$y = \sin(\alpha) * h$$

$$x = \cos(\alpha) * h$$

Con esto obtenemos el valor de y para las bicicletas desiguales y poder colocar el punto del marco con una inclinación, este principio también se usó para dibujar las líneas de la rueda cada 10°.

Programa

A continuación se pondrán pantallazos del código del programa con sus respectivos comentarios, por facilidad de visualización se uso Sublime text, un editor de texto que resalta las palabras definidas del lenguaje.

```
1 ; funcion para pasar de grados a radianes
2 ; grado: grado a convertir
3 (defun aGrados(grado)
4   (* pi (/ grado 180.0))
5 )
6
7 ; funcion para calcular las componentes de una linea mediante su angulo y tamaño
8 ; angle : angulo de la linea
9 ; h : tamaño de la linea
10 (defun cords (angle h)
11   (list (* (cos angle) h) (* (sin angle) h))
12 )
13
14 ; Funcion para dibujar las ruedas de las bicicleta
15 ; C: Centro de la rueda
16 ; R: Radio de la rueda
17 (defun rueda (C R / ang delta p2)
18   (command "_layer" "_S" "Ruedas" "")
19   ; Se dibuja el arco interno de la rueda y arco externo que tiene 10 unidades mas de radio
20   (command "_circle" C R)
21   (command "_circle" C (+ R 10))
22
23   ; Mediante un ciclo se dibujan las barras de la rueda cada uno con una separacion de 10°
24   ; Para calcular las coordenadas del segundo punto de la linea se usa cords (angulo R)
25
26   (setq ang 0.0)
27   (command "_layer" "_S" "Ruedas_Detalles" "")
28   (while (< ang 360.0)
29     (setq delta (cords (aGrados ang) R))
30     (setq p2 (list (+ (nth 0 delta) (nth 0 C)) (+ (nth 1 delta) (nth 1 C))))
31     ; (command "_layer" "_C" 250 "0" "")
32     (command "_line" C p2 "")
33     (setq ang (+ ang 10.0))
34   )
35 )
```

```

81 ; Funcion la cual dibuja la bicicleta en su totalidad
82 (defun bicy (/ Rueda1 Rueda2 Crueda1 Crueda2 distMarco delta p2MarcoBajo p1MarcoAlto p2MarcoAlto)
83   ; Se piden los radios de las ruedas
84   (setq Rueda1 (getreal "Radio de la rueda tracera: "))
85   (setq Rueda2 (getreal "Radio de la rueda delantera: "))
86   (setq color (getint "Numero de color del marco: "))
87
88   ; Se crean las capas para colocarles el color correspondiente a cada elemento
89   (command "._layer" "_N" "Ruedas" "_C" 250 "Ruedas" "")
90   (command "._layer" "_N" "Ruedas_Detalles" "_C" 252 "Ruedas_Detalles" "")
91   (command "._layer" "_N" "Otros" "_C" 37 "Otros" "")
92   (command "._layer" "_N" "Marco" "_C" color "Marco" "")
93
94
95   ; Se calcula la ubicacion de las ruedas y la distancia entre ellas dependiendo de sus radios
96   (setq Crueda1 (list 0 100))
97   (setq Crueda2 (list
98     (+ Rueda1 (if (= Rueda1 Rueda2) (* 2 Rueda2) 200) Rueda2)
99     (+ Rueda2 (- Rueda1) 100))
100 )
101
102   ; Dibujamos las ruedas
103   (rueda Crueda1 Rueda1)
104   (rueda Crueda2 Rueda2)
105
106   (command "._layer" "_S" "Marco" "")
107   ; Calculamos el marco mediante la distancia media entre las ruedas
108   (setq distMarco (* (distance Crueda1 Crueda2) 0.5))
109   (setq delta (cords (angle Crueda1 Crueda2) distMarco))
110
111   ; Calculamos los puntos del marco teniendo en cuenta el radio y la distancia anterior
112   (setq p2MarcoBajo (list
113     (+ Rueda1 (if (> Rueda1 Rueda2) 200 Rueda1) (nth 0 Crueda1))
114     (+ (nth 1 delta) (nth 1 Crueda1)))
115 )
116   (setq p1MarcoAlto (list (- (nth 0 p2MarcoBajo) 50) (+ 100 (* 1.5 Rueda2))))
117   (setq p2MarcoAlto (list (- (nth 0 Crueda2) Rueda2) (+ 100 (* 2 Rueda2))))
118   ;(command "._layer" "_C" "red" "0" "")
119   ; Dibujamos el marco
120   (marco Crueda1 p2MarcoBajo p2MarcoAlto p1MarcoAlto Crueda2)
121
122   (command "._layer" "_S" "Otros" "")
123   ; Dibujamos la silla
124   (silla p1MarcoAlto)
125
126   ; Dibujamos el manubrio
127   (manubrio p2MarcoAlto)
128 )

```



```

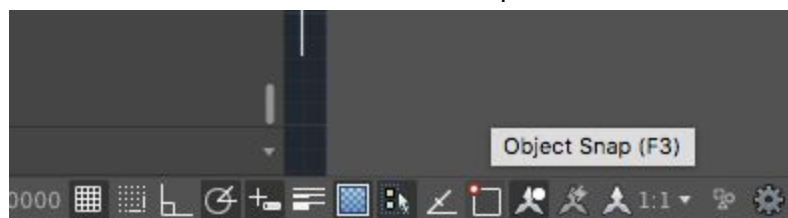
37 ; Funcion para dibujar la silla
38 ; pini : punto en el cual se conecta a la bicicleta
39 (defun silla (pini / psillabajo psillaIz psillaDr)
40   (setq psillabajo (list (- (nth 0 pini) 10) (+ (nth 1 pini) 50)))
41   (setq psillaIz (list (- (nth 0 psillabajo) 30) (+ (nth 1 psillabajo) 20)))
42   (setq psillaDr (list (+ (nth 0 psillabajo) 50) (+ (nth 1 psillabajo) 20)))
43
44   (command "._layer" "_S" "Marco" "")
45   (command "._line" pini psillabajo "")
46   (command "._layer" "_S" "Otros" "")
47   (command "._line" psillabajo psillaDr psillaIz psillabajo "")
48 )
49
50 ; Funcion para dibujar el manubrio
51 ; pini: punto en el cual se conecta con la bicicleta
52 (defun manubrio (pini / parco1 parco2 parco3)
53   (setq parco1 (list (+ (nth 0 pini) 50) (+ (nth 1 pini) 30)))
54   (setq parco2 (list (+ (nth 0 pini) 50) (- (nth 1 pini) 30)))
55   (setq parco3 (list (- (nth 0 pini) 1) (- (nth 1 pini) 30)))
56   (command "._line" pini parco1 "")
57   (command "._layer" "_S" "Otros" "")
58   (command "._arc" parco1 parco2 parco3)
59 )
60 ; Funcion para dibujar el marco de la bicicleta
61 ; CR : centro de la rueda tracera
62 ; p2MB : segundo punto del marco por abajo
63 ; p2MA : segundo punto del marco por arriba
64 ; p1MA : primer punto del marco por arriba
65 ; CR2 : centro de la segunda rueda
66 (defun marco (CR p2MB P2MA P1MA CR2 / CRD p2MBD P2MAD P1MAD)
67
68   ; Variables adicionales para que el marco no sea una sola linea sino
69   ; con un poco de grosor
70   (setq p2MBD (list (- (nth 0 p2MB) 3) (+ 5 (nth 1 p2MB))))
71   (setq P2MAD (list (- (nth 0 P2MA) 8) (- (nth 1 P2MA) 5)))
72   (setq P1MAD (list (+ (nth 0 P1MA) 5) (- (nth 1 P1MA) 5)))
73
74   (command "._line" CR p2MB P2MA P1MA CR "")
75   (command "._line" p2MA CR2 "")
76   (command "._line" p1MA p2MB "")
77
78   (command "._line" CR p2MBD P2MAD P1MAD CR "")
79 )

```

Carga y ejecución

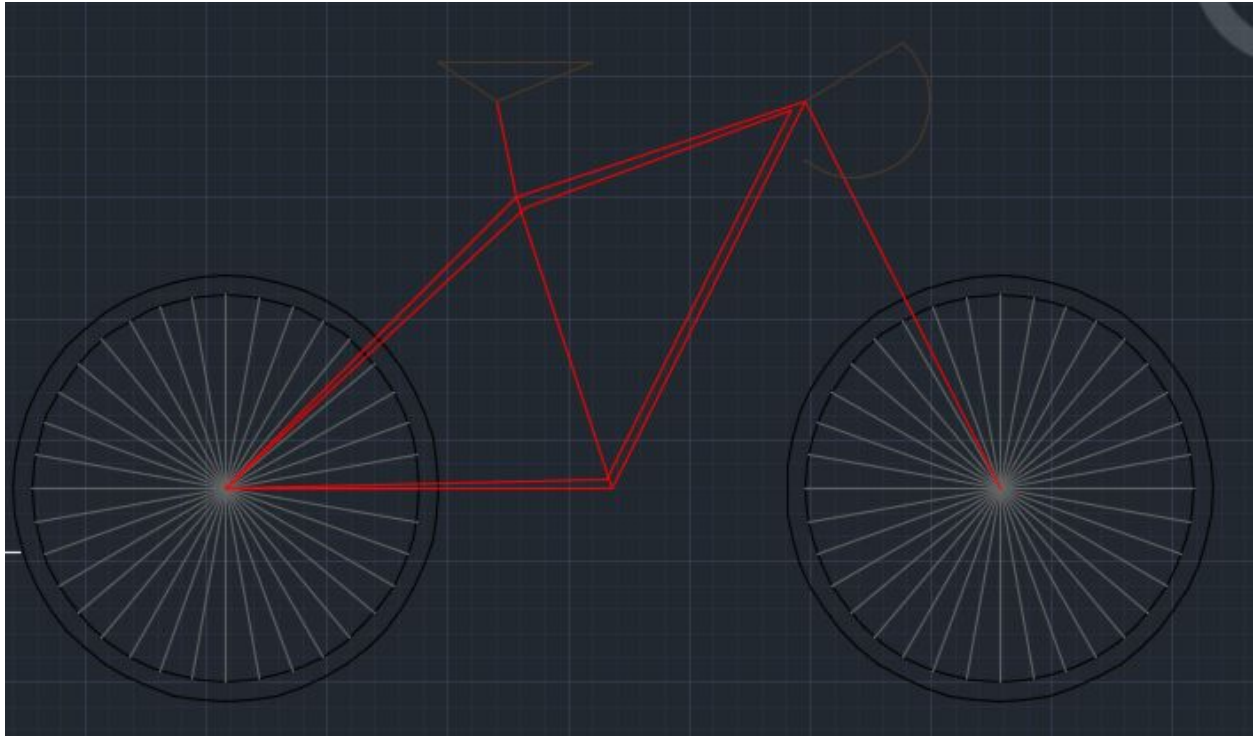
Para que el programa funcione hay que tener en cuenta lo siguiente:

- Tener desactivado la opción de Auto snap, esto se debe a que hay un problema con las barillas de las ruedas si esta opción está activada.



- cargar el script mediante el comando APPLOAD

Una vez cargado el programa se usa la función (**bicy**), al ejecutar la función nos pedirá el radio de las ruedas y un color para el marco, si los datos son correctos la bicicleta deseada aparecerá en el canvas.



La bicicleta de la imagen es el caso en que las ruedas son iguales.

Conclusiones

Al hacer esta tarea puede concluir:

- Mediante autolisp podemos crear scripts que nos ahorran trabajo manual y repetitivo de AutoCAD.
- AutoCAD puede construir figuras complejas con pocos datos.
- Al usar **command** nuestro programa funcione en los diferentes idiomas de autocad permitiendo una facilidad de trabajar en cualquier lado.
- Se pueden crear partes a partir de figuras simples como círculos y líneas.