UNIVERSIDAD NACIONAL DE COLOMBIA

SEDE BOGOTÁ
FACULTAD DE INGENIERÍA

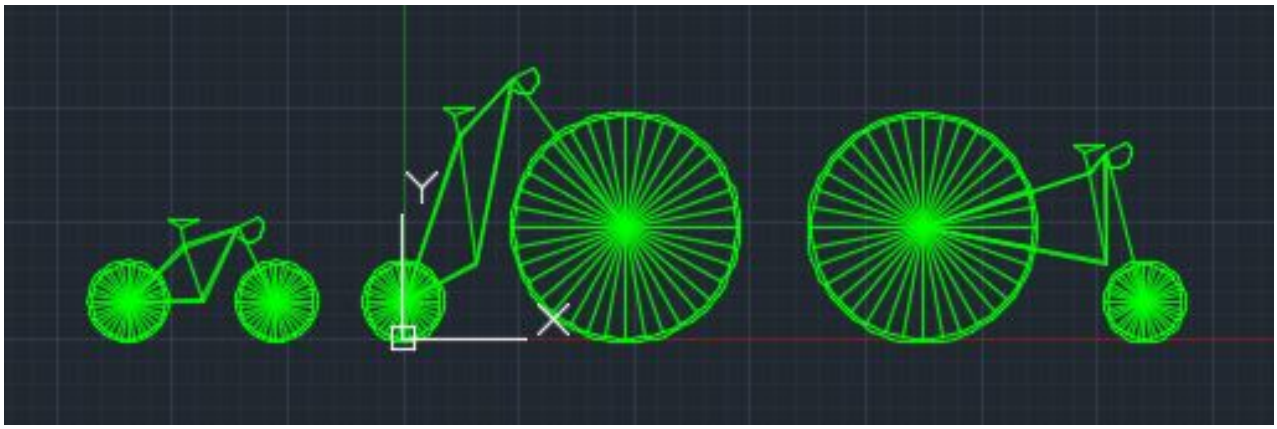
Tarea 1 - Dibujo libre

Computación gráfica
Miguel Angel Baquero

2017 - I

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:

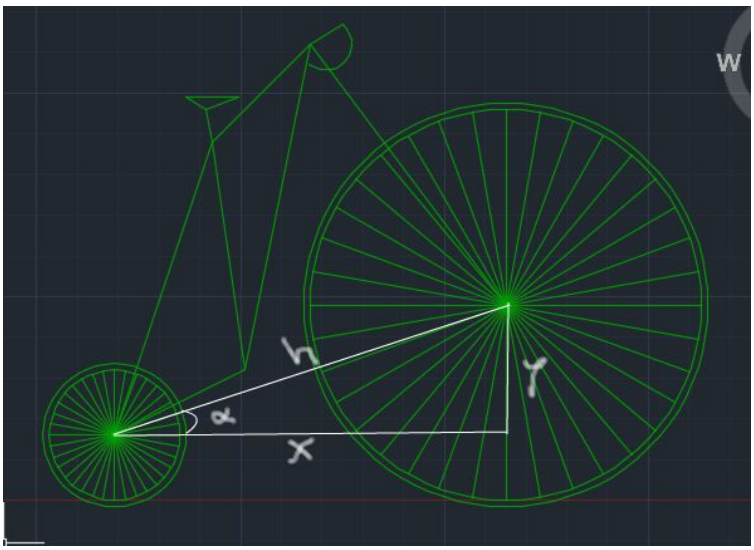
- Aplicar los conocimientos obtenidos en la clase del lenguaje autolisp para realizar un dibujo libre.
- Crear una figura compleja mediante el uso de figuras básicas como círculos, líneas y arcos.
- Utilizar diferentes colores, para añadir más complejidad a la tarea.
- Usar un razonamiento matemático 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 el 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 10º.

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 comvertir
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     ; Se dibuja el arco interno de la rueda y arco externo que tiene 10 unidade mas de radio
19     (command "._circle" C R)
20     (command "._circle" C (+ R 10))
21
22     ; Mediante un siclo se dibujan las baras de la rueda cada uno con una separacion de 10º
23     ; Para calcular las cordenadas del segunod punto de la linea se usa cords (angulo R)
24
25     (setq ang 0.0)
26     (while (< ang 360.0)
27         (setq delta (cords (aGrados ang) R))
28         (setq p2 (list (+ (nth 0 delta) (nth 0 C)) (+ (nth 1 delta) (nth 1 C))))
29         ;(command "._layer" "_C" 250 "0" "")
30         (command "._line" C p2 "")
31         (setq ang (+ ang 10.0))
32     )
33 )
```

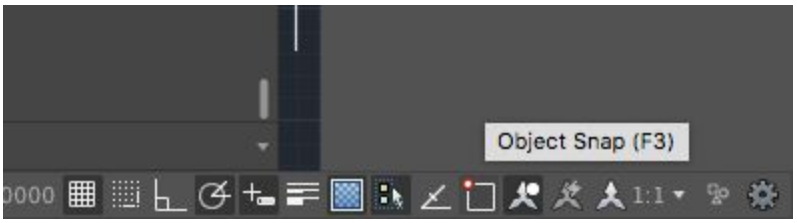
```
; Funcion para dibujar la silla
; pini : punto en el cual se conecta a la bicicleta
(defun silla (pini / psillabajo psillaIz psillaDr)
    (setq psillabajo (list (- (nth 0 pini) 10) (+ (nth 1 pini) 50)))
    (setq psillaIz (list (- (nth 0 psillabajo) 30) (+ (nth 1 psillabajo) 20)))
    (setq psillaDr (list (+ (nth 0 psillabajo) 50) (+ (nth 1 psillabajo) 20)))
    (command "._line" pini psillabajo psillaDr psillaIz psillabajo "")
)

; Funcion para dibujar el manubrio
; pini: punto en el cual se conecta con la bicicleta
(defun manubrio (pini / parco1 parco2 parco3)
    (setq parco1 (list (+ (nth 0 pini) 50) (+ (nth 1 pini) 30)))
    (setq parco2 (list (+ (nth 0 pini) 50) (- (nth 1 pini) 30)))
    (setq parco3 (list (- (nth 0 pini) 1) (- (nth 1 pini) 30)))
    (command "._line" pini parco1 "")
    (command "._arc" parco1 parco2 parco3)
)
; Funcion para dibujar el marco de la bicicleta
; CR : centro de la rueda tracera
; p2MB : segundo punto del marco por abajo
; p2MA : segundo punto del marco por arriba
; p1MA : primer punto del marco por arriba
; CR2 : centro de la segunda rueda
(defun marco (CR p2MB P2MA P1MA CR2 / CRD p2MBD P2MAD P1MAD)

    ; Variables adicionales para que el marco no sea una sola linea sino
    ; con un poco de grosor
    (setq p2MBD (list (- (nth 0 p2MB) 3) (+ 5 (nth 1 p2MB))))
    (setq P2MAD (list (- (nth 0 P2MA) 8) (- (nth 1 P2MA) 5)))
    (setq P1MAD (list (+ (nth 0 P1MA) 5) (- (nth 1 P1MA) 5)))

    (command "._line" CR p2MB P2MA P1MA CR "")
    (command "._line" p2MA CR2 "")
    (command "._line" p1MA p2MB "")

    (command "._line" CR p2MBD P2MAD P1MAD CR "")
)
```

```
; Funcion la cual dibuja la bicicleta en su totalidad
(defun bicy ( / Rrueda1 Rrueda2 Crueda1 Crueda2 distMarco delta p2MarcoBajo p1MarcoAlto p2MarcoAlto)
        ; Se piden los radios de las ruedas
    (setq Rrueda1 (getreal "Radio de la rueda tracera: "))
    (setq Rrueda2 (getreal "Radio de la rueda delantera: "))

    ; Se calcula la ubicacion de las ruedas y la distancia entre ellas dependiendo de sus radios
    (setq Crueda1 (list 0 100))
    (setq Crueda2 (list
        (+ Rrueda1 (if (= Rrueda1 Rrueda2) (* 2 Rrueda2) 200) Rrueda2)
        (+ Rrueda2 (- Rrueda1) 100))
    )

    ; Dibujamos las ruedas
    (rueda Crueda1 Rrueda1)
    (rueda Crueda2 Rrueda2)

    ; Calculamos el marco mediante la distancia media entre las ruedas
    (setq distMarco (* (distance Crueda1 Crueda2) 0.5))
    (setq delta (cords (angle Crueda1 Crueda2) distMarco))

    ; Calculamos los puntos del marco teniendo en cuenta el radio y la distancia anterior
    (setq p2MarcoBajo (list
        (+ Rrueda1 (if (> Rrueda1 Rrueda2) 200 Rrueda1) (nth 0 Crueda1))
        (+ (nth 1 delta) (nth 1 Crueda1)))
    )
    (setq p1MarcoAlto (list (- (nth 0 p2MarcoBajo) 50) (+ 100 (* 1.5 Rrueda2))))
    (setq p2MarcoAlto (list (- (nth 0 Crueda2) Rrueda2) (+ 100 (* 2 Rrueda2))))

    ; Dibujamos el marco
    (command "._line" Crueda1 p2MarcoBajo p2MarcoAlto p1MarcoAlto Crueda1 "")
    (command "._line" p2MarcoAlto Crueda2 "")
    (command "._line" p1MarcoAlto p2MarcoBajo "")

    ; Dibujamos la silla
    (silla p1MarcoAlto)

    ; Dibujamos el manubrio
    (manubrio p2MarcoAlto)
)
```

Carga y ejecución

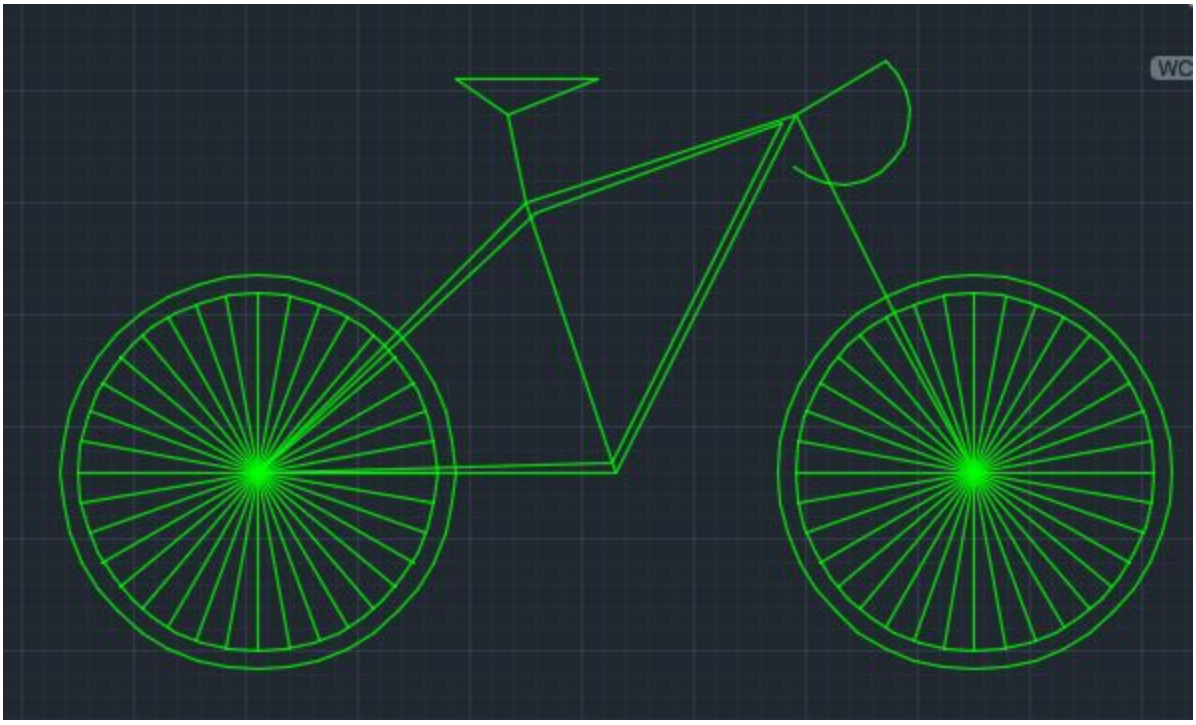
Para que el programa funcione hay que tener en cuenta lo siguiente:

- 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.