

¿Piedra, papel o tijera?

Héctor Augusto Daza Roa

March 11, 2018

1 Descripción

Este dibujo junto con su rutina de lisp, es una aplicación del famoso juego "¿Piedra, Papel o tijera?", El usuario elige el número de partidas que quiere jugar, y luego se enfrenta a la computadora (ambos representados por caras de robots). El dibujo registra las victorias de cada uno, y las caras de los robots se alegran o enfurecen según sea el caso.

2 Funciones

2.1 random

Genera un número aleatorio entre 0, 1 y 2. Para lograrlo toma los segundos de la hora del sistema en el momento en el que es ejecutada y le aplica la operación módulo 3, es decir, toma el residuo de la división en tres, cuyos únicos resultados posibles son 0, 1 y 2.

Parámetros: Ninguno.

Variables internas:

- Aleatorio: Resultado que retorna la función
- Tiempo: Hora y fecha del sistema, más adelante se convierte solo en la hora (sin fecha), luego en minutos y segundos, y finalmente solo en segundos.
- Entero: Resultado de la conversión a entero del parámetro tiempo

2.2 redibujarJugada

Edita las figuras actuales que representan la piedra, el papel o las tijeras (o signo de interrogación en su defecto), cambiando la propiedad 2 de las entidades; y así representa las jugadas actuales (de la máquina y de la persona).

Parámetros:

1. Antigua: Figura actual
2. Jugada: Número que representa la jugada ya sea de la máquina o de la persona ("0" para piedra, "1" para papel y "2" para tijeras)

Variables internas:

1. Propiedades: Lista de propiedades de la figura actual.
2. Origen: Punto base de la figura actual (propiedad "10")
3. Ninguno: Variable que activa el condicional que dibuja el signo de interrogación (cuyo nombre de bloque es piedra), esto en caso de que el usuario no marque ni piedra, ni papel ni tijera.

2.3 redibujarCara

Edita las caras actuales de los robots que representan a los jugadores (la máquina o el usuario), cambiando la propiedad 2 de las entidades; y así representa la victoria, derrota o empate de cada jugador (cara neutral para empate (amarilla), cara feliz para victoria (verde) y cara brava para derrota (rojo)).

Parámetros:

1. Antigua: Cara actual
2. Cara: Número que representa la victoria, empate o derrota del jugador, ya sea de la máquina o de la persona ("0" para empate, "1" para victoria y "2" para derrota)

Variables internas:

1. Propiedades: Lista de propiedades de la cara actual.
2. Origen: Punto base de la cara actual (propiedad "10")

2.4 cambiarPalabra

Retorna una cadena de caracteres (una palabra), según sea el caso: "Empate", "Victoria" o "Derrota".

Parámetros:

1. Número: Valor que representa cada palabra.

Variables internas:

1. Palabra: Resultado que retorna la función

2.5 redibujarTexto

Edita el texto del objeto que recibe como parámetro. Este objeto puede ser uno de los textos bajo las caras de los robots o uno de los marcadores de victorias.

Parámetros:

1. Texto: Objeto a ser editado.
2. Palabra: cadena de caracteres con la que se edita al objeto.

Variables internas:

1. Propiedades: Lista de propiedades del objeto.

2.6 actualizar

Ejecuta las demás funciones para actualizar todo el dibujo. Actualiza las caras, las figuras y los textos (pero no los marcadores).

Parámetros:

1. Número1: Representa la victoria, empate o derrota de la máquina.
2. Número2: Representa la victoria, empate o derrota del usuario.

Variables internas: Ninguna.

3 Funcionamiento

Lo primero en la ejecución del programa es el establecimiento del número de victorias de ambos jugadores en cero (el usuario y la máquina).

Luego se le solicita al usuario el número de partidas que desea jugar, y se inicia un ciclo que abarca a casi todo el resto del programa.

Dentro del ciclo, lo primero que se realiza es la creación de varias variables que guardan los nombres de entidad (entity name) de todos los objetos en el dibujo.

Más adelante se generan las jugadas de la máquina (función random) y la persona (se le solicita al usuario), para luego determinar el ganador a través de la resta. El resultado de esta diferencia pasa por una serie de condicionales que ejecutan la función "actualizar" (con sus parámetros correspondientes en cada caso) actualizan el número de victorias y los textos de los marcadores; esta es la última ejecución dentro del ciclo. Este proceso se repite iterativamente según sea el número de partidas.

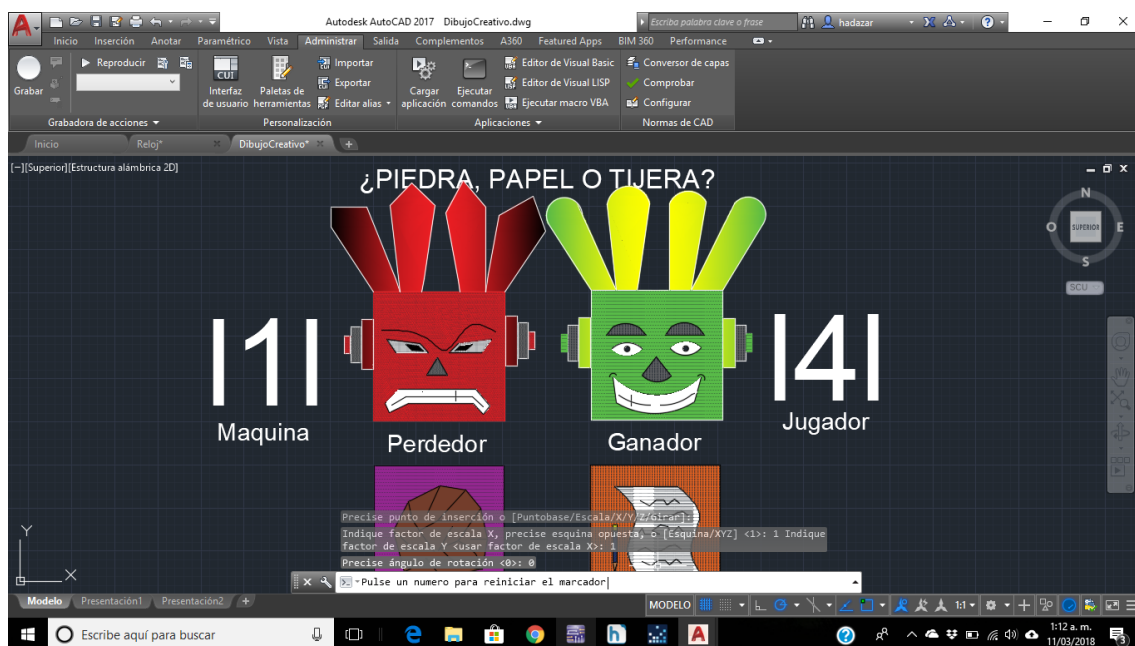
Ya fuera del ciclo se le solicita al usuario un valor entero, sin ningún significado dentro del programa. La razón de esta sentencia, es darle una pausa a la rutina de lisp para que alcance a actualizar el dibujo, y pueda ser observado el resultado final, antes de que se actualicen los marcadores con las últimas dos sentencias.

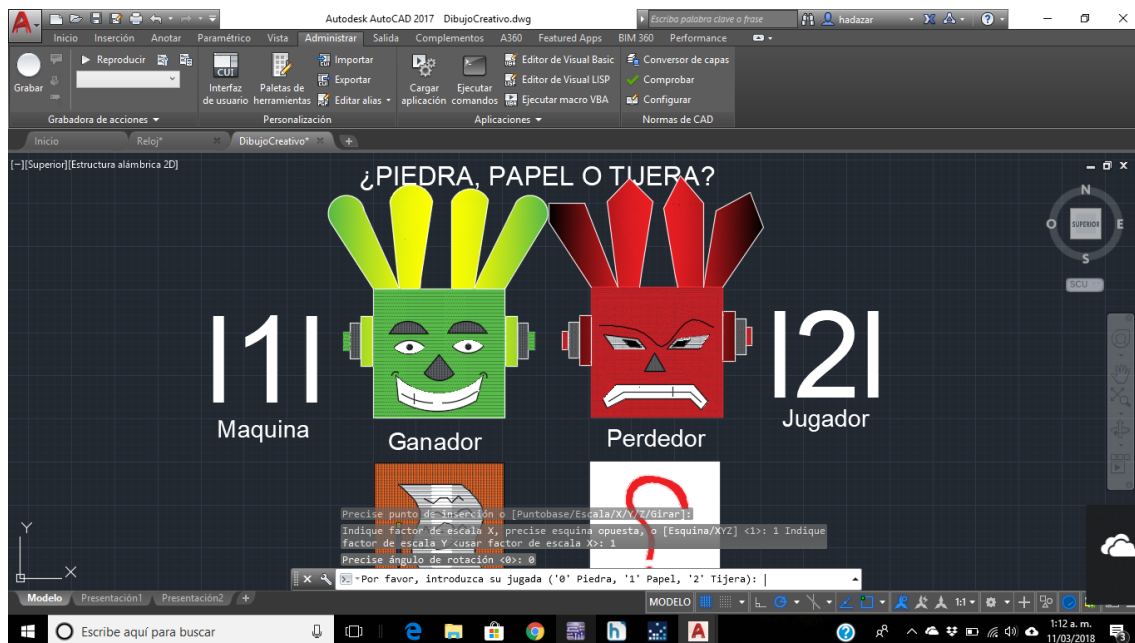
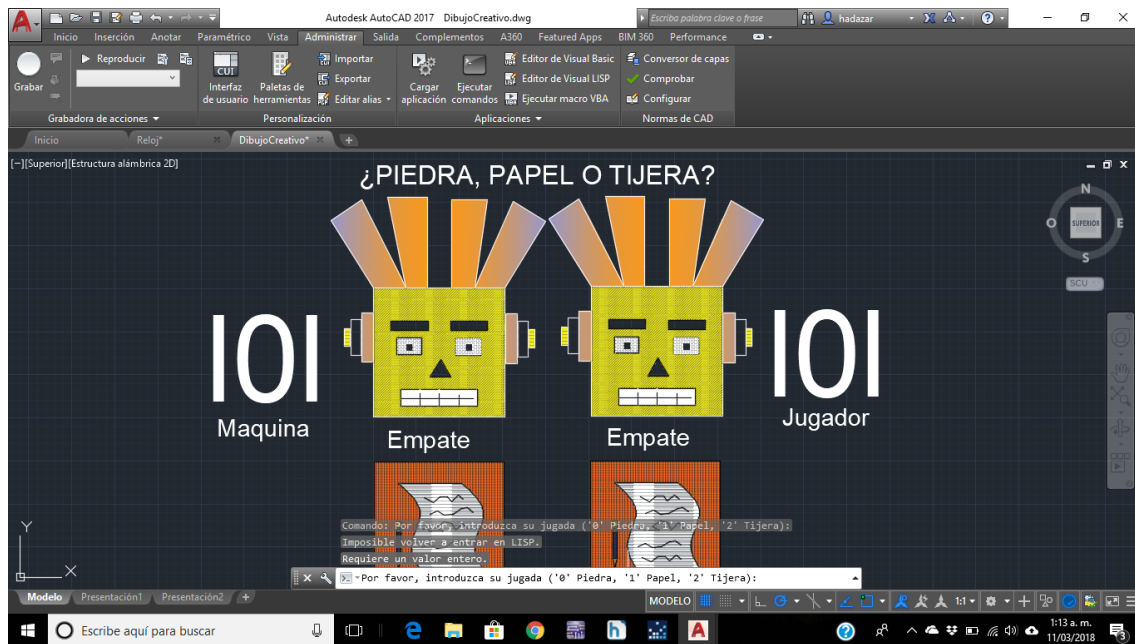
4 ¿Cómo cargar el programa?

Abrir el dibujo "DibujoCreativo.dwg" (hecho en autocad 2017, pero reditado en autocad 2022 para ajustar algunos detalles).

Una vez abierto, escribir en la línea de comandos "APPLOAD" y cargar la rutina de lisp "Logica.lsp". Al cargar la rutina, solicitará el número de partidas (escribir un número entero), luego solicitará una jugada: "0" para piedra, "1" para papel y "2" para tijeras. Al insertar la jugada se actualizará el dibujo. Este proceso se repetirá tantas veces como el número de partidas que se ingreso.

5 Pantallazos





References

- [1] *Autodesk help*. Available: <http://jpr dintprev.autodesk.com/adn/servlet/autoindex>
- [2] *Funciones Primitivas de Lisp*. Available: <http://www.davidam.com/software/primitivas.html>
- [3] *Programación de Aplicaciones Gráficas*. Available: <http://www.togores.net/v1/curso/lisp>