

Analizaré el código línea por línea siguiendo tus reglas estrictas:

Línea 1: `def mostrar_historia():`

- Define una función llamada `mostrar_historia`. Esta función no acepta ningún parámetro. Su propósito es imprimir información sobre la historia de la animación por computadora.

Línea 2: `print("\n--- Historia de la Animación por Computadora ---")`

- Dentro de la función `mostrar_historia`: Imprime una cadena de texto que sirve como título para esta sección. `\n` al principio añade una línea en blanco antes del título para una mejor presentación.

Línea 3: `print("* Década de 1960: Inicios con Ivan Sutherland y el programa Sketchpad.")`

- Dentro de la función `mostrar_historia`: Imprime una línea de texto que describe el inicio de la animación por computadora en la década de 1960, mencionando a Ivan Sutherland y Sketchpad.

Línea 4: `print("* 1982: 'Tron' de Disney usa secuencias generadas por computadora.")`

- Dentro de la función `mostrar_historia`: Imprime una línea de texto que destaca la película 'Tron' de 1982 como un hito temprano en el uso de gráficos generados por computadora (CGI).

Línea 5: `print("* 1995: Pixar lanza 'Toy Story', el primer largometraje 100% animado digitalmente.\n")`

- Dentro de la función `mostrar_historia`: Imprime una línea de texto que menciona el lanzamiento de 'Toy Story' en 1995 como el primer largometraje completamente animado digitalmente. `\n` al final añade una línea en blanco después del contenido para una mejor separación.

Línea 6: (En blanco)

- Línea en blanco: Separa la definición de la función `mostrar_historia` de la siguiente función `mostrar_evolucion`, mejorando la legibilidad del código.

Línea 7: `def mostrar_evolucion():`

- Define una función llamada `mostrar_evolucion`. Esta función no acepta ningún parámetro. Su propósito es imprimir información sobre la evolución tecnológica en el campo de la animación por computadora.

Línea 8: `print("\n--- Evolución Tecnológica ---")`

- Dentro de la función `mostrar_evolucion`: Imprime una cadena de texto que sirve como título para esta sección, con una línea en blanco antes.

Línea 9: `print("* 1980-1990: Nace el CGI, mejora el modelado 3D y el renderizado.")`

- Dentro de la función `mostrar_evolucion`: Imprime una línea de texto que describe el período de los 80 y 90 como el nacimiento del CGI y las mejoras en modelado y renderizado 3D.

Línea 10: `print("* 2000-2010: Realismo con captura de movimiento, sombras y simulaciones físicas.")`

- Dentro de la función `mostrar_evolucion`: Imprime una línea de texto que destaca las mejoras de realismo en la década de 2000, como la captura de movimiento y simulaciones físicas.

Línea 11: `print("* 2010 en adelante: Se integran IA, realidad aumentada y motores como Unity o Unreal Engine.\n")`

- Dentro de la función `mostrar_evolucion`: Imprime una línea de texto que describe las tendencias a partir de 2010, incluyendo IA, realidad aumentada y motores de juego. \n al final añade una línea en blanco después del contenido.

Línea 12: (En blanco)

- Línea en blanco: Separa la definición de la función `mostrar_evolucion` de la siguiente función `mostrar_aplicaciones`.

Línea 13: `def mostrar_aplicaciones():`

- Define una función llamada `mostrar_aplicaciones`. Esta función no acepta ningún parámetro. Su propósito es imprimir información sobre las diversas aplicaciones de la animación por computadora.

Línea 14: `print("\n--- Aplicaciones de la Animación por Computadora ---")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una cadena de texto que sirve como título para esta sección, con una línea en blanco antes.

Línea 15: `print("* Cine y TV: Personajes virtuales, efectos visuales.")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una línea de texto que detalla las aplicaciones en cine y TV, como personajes virtuales y efectos visuales.

Línea 16: `print("* Videojuegos: Modelado, cinemáticas, interacción.")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una línea de texto que describe las aplicaciones en videojuegos, incluyendo modelado, cinemáticas e interacción.

Línea 17: `print("* Medicina: Simulación quirúrgica, visualización de órganos.")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una línea de texto que menciona las aplicaciones en medicina, como simulaciones quirúrgicas.

Línea 18: `print("* Arquitectura: Recorridos virtuales, modelos 3D.")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una línea de texto que describe las aplicaciones en arquitectura, como recorridos virtuales.

Línea 19: `print("* Educación: Recursos animados y e-learning.\n")`

- Dentro de la función `mostrar_aplicaciones`: Imprime una línea de texto que destaca las aplicaciones en educación, como recursos animados y e-learning. `\n` al final añade una línea en blanco después del contenido.

Línea 20: (En blanco)

- Línea en blanco: Separa la definición de la función `mostrar_aplicaciones` de la función `menu`.

Línea 21: `def menu():`

- Define una función llamada `menu`. Esta función no acepta ningún parámetro. Su propósito es mostrar un menú interactivo al usuario y gestionar sus elecciones.

Línea 22: `while True:`

- Dentro de la función `menu`: Inicia un bucle `while` infinito (`True` siempre es verdadero). Esto significa que el menú se mostrará repetidamente hasta que se encuentre una condición para romper el bucle (la opción '4' para salir).

Línea 23: `print("=== Animación por Computadora ===")`

- Dentro del bucle `while`: Imprime una línea de texto que sirve como encabezado principal del menú.

Línea 24: `print("1. Historia")`

- Dentro del bucle `while`: Imprime la primera opción del menú: "1. Historia".

Línea 25: `print("2. Evolución")`

- Dentro del bucle `while`: Imprime la segunda opción del menú: "2. Evolución".

Línea 26: `print("3. Aplicaciones")`

- Dentro del bucle `while`: Imprime la tercera opción del menú: "3. Aplicaciones".

Línea 27: `print("4. Salir")`

- Dentro del bucle `while`: Imprime la cuarta y última opción del menú: "4. Salir".

Línea 28: `opcion = input("Selecciona una opción: ")`

- Dentro del bucle `while`: Solicita al usuario que ingrese una opción. La entrada del usuario se lee como una cadena de texto y se almacena en la variable `opcion`.

Línea 29: (En blanco)

- Dentro del bucle `while`: Línea en blanco para separar la entrada del usuario de la lógica condicional.

Línea 30: `if opcion == '1':`

- Dentro del bucle `while`: Comienza una estructura condicional `if-elif-else`. Esta línea verifica si el valor de `opcion` es exactamente la cadena '1'.

Línea 31: `mostrar_historia()`

- Dentro del bloque `if`: Si `opcion` es '1', se llama a la función `mostrar_historia()`, que imprimirá la información histórica.

Línea 32: `elif opcion == '2':`

- Continúa la estructura condicional. Si la condición anterior (`opcion == '1'`) fue falsa, esta línea verifica si `opcion` es exactamente la cadena '2'.

Línea 33: `mostrar_evolucion()`

- Dentro del bloque `elif`: Si `opcion` es '2', se llama a la función `mostrar_evolucion()`, que imprimirá la información sobre la evolución tecnológica.

Línea 34: `elif opcion == '3':`

- Continúa la estructura condicional. Si las condiciones anteriores fueron falsas, esta línea verifica si `opcion` es exactamente la cadena '3'.

Línea 35: `mostrar_aplicaciones()`

- Dentro del bloque `elif`: Si `opcion` es '3', se llama a la función `mostrar_aplicaciones()`, que imprimirá la información sobre las aplicaciones.

Línea 36: `elif opcion == '4':`

- Continúa la estructura condicional. Si las condiciones anteriores fueron falsas, esta línea verifica si `opcion` es exactamente la cadena '4'.

Línea 37: `print("Gracias por explorar la animación por computadora.")`

- Dentro del bloque `elif`: Si `opcion` es '4', imprime un mensaje de despedida al usuario.

Línea 38: `break`

- Dentro del bloque `elif`: Si `opcion` es '4', la sentencia `break` detiene inmediatamente la ejecución del bucle `while True`, finalizando la función `menu`.

Línea 39: `else:`

- Dentro de la estructura condicional: Este bloque `else` se ejecuta si ninguna de las condiciones `if` o `elif` anteriores fue verdadera (es decir, el usuario ingresó una opción diferente de '1', '2', '3' o '4').

Línea 40: `print("Opción no válida. Intenta de nuevo.\n")`

- Dentro del bloque `else`: Imprime un mensaje de error indicando que la opción ingresada no es válida. `\n` al final añade una línea en blanco para separar el mensaje del siguiente menú.

Línea 41: (En blanco)

- Línea en blanco: Separa la definición de la función `menu` de la estructura condicional `if __name__ == "__main__":`.

Línea 42: `if __name__ == "__main__":`

- Esta es una construcción estándar en Python. El código dentro de este bloque solo se ejecutará cuando el script se ejecute directamente (no cuando se importe como un módulo en otro script).

Línea 43: `menu()`

- Dentro del bloque `if __name__ == "__main__":` Llama a la función `menu()`. Esta es la línea que inicia la ejecución del programa al mostrar el menú al usuario cuando el script se ejecuta.

Resumen del Código

Este script de Python es un programa interactivo basado en consola que proporciona información sobre la animación por computadora, dividida en secciones de historia, evolución tecnológica y aplicaciones.

El código está estructurado en varias funciones:

- `mostrar_historia()`: Imprime hitos clave en la historia de la animación por computadora.
- `mostrar_evolucion()`: Detalla el desarrollo tecnológico en el campo a lo largo de las décadas.
- `mostrar_aplicaciones()`: Enumera y describe los diversos usos de la animación por computadora en diferentes industrias.
- `menu()`: Es la función principal del programa que presenta un menú al usuario. Permite al usuario elegir entre ver la historia, la evolución, las aplicaciones o salir del programa. Utiliza un bucle para mostrar el menú repetidamente hasta que el usuario decida salir.

Finalmente, el bloque `if __name__ == "__main__":` asegura que la función `menu()` se ejecute automáticamente cuando el script se inicia, presentando el menú interactivo al usuario.

```
jalomo's graficacion-practicas/ on master = ?12 RAM:1
> py practica41/assets/main.py
=== Animación por Computadora ===
1. Historia
2. Evolución
3. Aplicaciones
4. Salir
Selecciona una opción: 1

--- Historia de la Animación por Computadora ---
* Década de 1960: Inicios con Ivan Sutherland y el programa Sketchpad.
* 1982: 'Tron' de Disney usa secuencias generadas por computadora.
* 1995: Pixar lanza 'Toy Story', el primer largometraje 100% animado digitalmente.

=== Animación por Computadora ===
1. Historia
2. Evolución
3. Aplicaciones
4. Salir
Selecciona una opción: 2

--- Evolución Tecnológica ---
* 1980-1990: Nace el CGI, mejora el modelado 3D y el renderizado.
* 2000-2010: Realismo con captura de movimiento, sombras y simulaciones físicas.
* 2010 en adelante: Se integran IA, realidad aumentada y motores como Unity o Unreal Engine.

=== Animación por Computadora ===
1. Historia
2. Evolución
3. Aplicaciones
4. Salir
Selecciona una opción: 3

--- Aplicaciones de la Animación por Computadora ---
* Cine y TV: Personajes virtuales, efectos visuales.
* Videojuegos: Modelado, cinemáticas, interacción.
* Medicina: Simulación quirúrgica, visualización de órganos.
* Arquitectura: Recorridos virtuales, modelos 3D.
* Educación: Recursos animados y e-learning.

=== Animación por Computadora ===
1. Historia
2. Evolución
3. Aplicaciones
4. Salir
Selecciona una opción: 4
Gracias por explorar la animación por computadora.
```