

The logo of the University of Mariano Galvez is a circular seal. It features a central figure of a person in a crouched position, possibly a Native American, holding a staff. The text "UNIVERSIDAD MARIANO GALVEZ" is written in a circle around the top, and "GUATEMALA" is at the bottom. The year "1966" is inscribed above the figure. Below the figure, the motto "CONOCEREIS LA VERDAD Y LA VERDAD OS HARA LIBRES" is written in a smaller circle.

MANUAL DE USUARIO

PROGRAMACIÓN III

21-3898 - EDWIN ADONY MONTEJO MARTINEZ

01/06/2024

Manual de Usuario para la Aplicación de "Totito" con Interfaz Gráfica

Introducción

Este manual de usuario está diseñado para ayudarte a utilizar la aplicación de "Totito" (Tic-Tac-Toe) con una interfaz gráfica desarrollada en Python utilizando la biblioteca `tkinter`. La aplicación también incorpora un árbol de decisión (ARBOL) para la lógica del juego, que ayuda a la computadora a tomar decisiones inteligentes durante el juego.

Requisitos Previos

- Python 3.x instalado en tu sistema.
- Biblioteca `tkinter` para la interfaz gráfica.
- Biblioteca `random` para generar movimientos aleatorios.
- Biblioteca `re` para manejo de expresiones regulares (si es necesario).
- Archivo `Arbol.py` que contiene la implementación de la clase ARBOL.

Instalación y Configuración

1. **Instalación de `tkinter`:**
 - En Windows: `pip install tk`
 - En Linux: `sudo apt-get install python3-tk`
 - En MacOS: `brew install python-tk`
2. **Estructura del Proyecto:**
 - `main.py`: Contiene el código principal del juego.
 - `Arbol.py`: Contiene la implementación de la clase ARBOL.
3. **Código de `Arbol.py`:**
 - Asegúrate de tener el archivo `Arbol.py` en el mismo directorio que `main.py`.
 - `Arbol.py` debe contener la implementación completa de la clase ARBOL proporcionada en la sección anterior.

Uso de la Aplicación

Iniciar la Aplicación

Ejecutar el Programa:

- Abre una terminal o línea de comandos.
- Navega hasta el directorio donde se encuentra `main.py`.
- Ejecuta el siguiente comando: `python main.py`.

Interfaz del Juego:

- Al ejecutar el programa, se abrirá una ventana titulada "Totito".
- La ventana contiene un tablero de 3x3 y un botón de "Reiniciar".

Jugar el Juego

Iniciar una Partida:

- Haz clic en cualquier botón del tablero para colocar una "X" en la celda seleccionada. Este es el turno del jugador.

Turno de la Computadora:

- Después de cada movimiento del jugador, la computadora hará su movimiento automáticamente colocando una "O".

Detectar el Ganador:

- Si el jugador o la computadora consiguen alinear tres de sus símbolos en una fila, columna o diagonal, aparecerá un mensaje emergente anunciando el ganador.
- Si todas las celdas están llenas y no hay ganador, el juego termina en empate, y se mostrará un mensaje emergente.

Reiniciar el Juego:

- Puedes reiniciar el juego en cualquier momento haciendo clic en el botón "Reiniciar". Esto limpiará el tablero y reiniciará el turno del jugador.

Funcionalidades del Código

Clase TotitoBoard:

Atributos:

- `root`: Ventana principal de tkinter.
- `current_player`: Jugador actual ("X" o "O").
- `board`: Lista que representa el estado actual del tablero.
- `buttons`: Lista de botones del tablero.
- `arbol`: Instancia de la clase ARBOL.

Métodos:

- `__init__(self, root, arbol)`: Inicializa la ventana y el tablero.
- `create_board(self)`: Crea los botones del tablero y el botón de reinicio.
- `on_button_click(self, index)`: Maneja el evento de clic en los botones del tablero.
- `computer_move(self)`: Genera y maneja el movimiento de la computadora.
- `check_winner(self)`: Verifica si hay un ganador.
- `reset_board(self)`: Reinicia el tablero y los estados del juego.

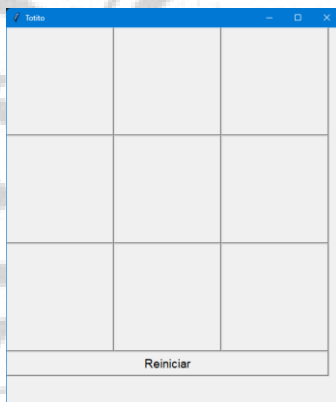
Interacción con ARBOL:

- El árbol de decisión (`arbol`) se utiliza para decidir el mejor movimiento para la computadora.
- `arbol.avanzar(index)`: Avanza al nodo correspondiente al movimiento actual.
- `arbol.insert(index, index)`: Inserta un nuevo nodo en el árbol.
- `arbol.buscaOpc()`: Busca la mejor opción de movimiento para la computadora.
- `arbol.asignarValores(ganador)`: Asigna valores a los nodos del árbol basado en el resultado del juego.

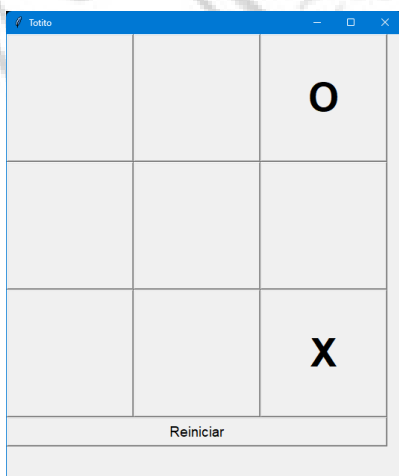
Ejemplo de Uso

Aquí hay un ejemplo básico de cómo iniciar y jugar al "Totito":

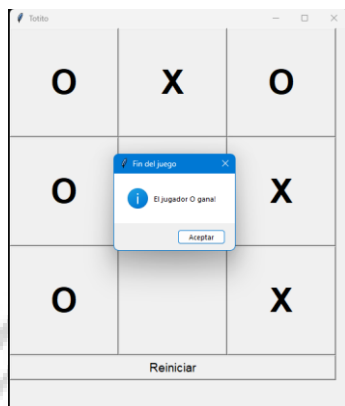
1. Ejecuta el programa.
2. Aparecerá una ventana con un tablero de 3x3.



3. Haz clic en cualquier celda para colocar una "X".
4. La computadora colocará automáticamente una "O".



5. Continúa jugando hasta que alguien gane o sea un empate.
6. Reinicia el juego si lo deseas haciendo clic en "Reiniciar".



Uso de la Aplicación

Iniciar la Aplicación

1. Interfaz del Menú Principal:

- Al ejecutar el programa, verás el siguiente menú:

```
Copiar código
1. Jugar
2. Opciones Juego
0. Salir
```

2. Seleccionar una Opción:

- Ingresa el número correspondiente a la opción que deseas seleccionar y presiona Enter.

Opciones del Juego en la Consola

1. Seleccionar "Opciones Juego" (Opción 2):

- Al seleccionar "Opciones Juego", verás el siguiente menú:

```
Copiar código
1. Juego Consola
2. Generar Pruebas
3. Generar Json
4. Consultar Json
5. Opciones Arbol
0. Salir
```

2. Juego en Consola (Opción 1):

- Inicia una partida de "Totito" en modo consola.

3. Generar Pruebas (Opción 2):

- Genera múltiples partidas de prueba para entrenar el árbol de decisión (ARBOL).

4. Generar Json (Opción 3):

- Guarda la estructura actual del árbol de decisión (ARBOL) en un archivo JSON (arbol.json).

5. Consultar Json (Opción 4):

- Carga una estructura de árbol de decisión desde un archivo JSON (arbol.json).

6. Opciones Arbol (Opción 5):

- Configura y administra el árbol de decisión (ARBOL). Las opciones incluyen:

```
Copiar código
1. Volver a la raiz
2. ubicaciones
3. Avanzar
4. Retroceder
5. Eliminar
6. Grafica
7. Asignar Valores
0. Salir
```

Funcionalidades del Código

1. Clase `TotitoBoard`:

○ Atributos:

- `root`: Ventana principal de `tkinter`.
- `current_player`: Jugador actual ("X" o "O").
- `board`: Lista que representa el estado actual del tablero.
- `buttons`: Lista de botones del tablero.
- `arbol`: Instancia de la clase `ARBOL`.

○ Métodos:

- `__init__(self, root, arbol)`: Inicializa la ventana y el tablero.
- `create_board(self)`: Crea los botones del tablero y el botón de reinicio.
- `on_button_click(self, index)`: Maneja el evento de clic en los botones del tablero.
- `computer_move(self)`: Genera y maneja el movimiento de la computadora.
- `check_winner(self)`: Verifica si hay un ganador.
- `reset_board(self)`: Reinicia el tablero y los estados del juego.

2. Interacción con `ARBOL`:

- El árbol de decisión (`arbol`) se utiliza para decidir el mejor movimiento para la computadora.
- `arbol.avanzar(index)`: Avanza al nodo correspondiente al movimiento actual.
- `arbol.insert(index, index)`: Inserta un nuevo nodo en el árbol.
- `arbol.buscaOpc()`: Busca la mejor opción de movimiento para la computadora.
- `arbol.asignarValores(ganador)`: Asigna valores a los nodos del árbol basado en el resultado del juego.

Conclusión

Este manual de usuario proporciona una guía completa sobre cómo utilizar la aplicación de "Totito" tanto en modo consola como con interfaz gráfica en Python. Siguiendo estos pasos, puedes jugar al juego, generar partidas de prueba, administrar el árbol de decisión y entender cómo interactúa la lógica del juego con el árbol de decisión para tomar decisiones inteligentes durante el juego.