

Manual de Uso para el Usuario: Juego "Adivina Quién"

Introducción

Bienvenido al juego "Adivina Quién". Este juego te permite agregar personajes con diferentes atributos a una base de datos. A través de una interfaz gráfica sencilla, podrás introducir características como estatura, color de piel, habilidad, tipo de especie, y una característica distintiva. Este manual te guiará paso a paso para utilizar la aplicación de manera efectiva.

Requisitos Previos

Antes de comenzar, asegúrate de cumplir con los siguientes requisitos:

1. **Sistema Operativo:** Windows, macOS o Linux.
2. **Python:** Debes tener Python instalado en tu computadora.
3. **Base de Datos MySQL:** Asegúrate de tener una base de datos MySQL configurada y en funcionamiento.

Configuración Inicial

1. **Descargar el Código:** Descarga el archivo AdivinaQuien.py a tu computadora.
2. **Configurar la Base de Datos:** Crea una base de datos en MySQL y una tabla llamada personajes con la siguiente estructura:

```
sql
Copiar código
CREATE TABLE personajes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(255),
  atributos JSON
);
```

3. **Configurar Credenciales:** Abre el archivo AdivinaQuien.py y modifica las credenciales de conexión a la base de datos en la sección `connection_params`:

```
python
Copiar código
connection_params = {
    'host': 'localhost',
    'user': 'root',
    'password': 'tu_contraseña',
    'database': 'nombre_de_tu_base_de_datos',
    'port': '3306'
}
```

Ejecución del Juego

1. **Iniciar el Juego:** Abre una terminal o línea de comandos y navega hasta el directorio donde descargaste AdivinaQuien.py. Ejecuta el siguiente comando para iniciar el juego:

```
sh
Copiar código
python AdivinaQuien.py
```

2. **Interfaz Gráfica:** Una vez que el juego se inicia, aparecerá una ventana con el título "Adivina Quién".

Paso a Paso: Ingresar un Nuevo Personaje

1. **Estatura:**
 - La primera ventana te pedirá que ingreses la estatura del personaje.
 - Escribe la estatura (Baja, Mediana, Alta) en el campo de texto.
 - Haz clic en el botón "Siguiente" para continuar.
2. **Color de Piel:**
 - La siguiente ventana te pedirá que ingreses el color de piel del personaje.
 - Escribe el color de piel en el campo de texto.
 - Haz clic en el botón "Siguiente" para continuar.
3. **Habilidad:**
 - La siguiente ventana te pedirá que ingreses la habilidad del personaje.
 - Escribe la habilidad en el campo de texto.
 - Haz clic en el botón "Siguiente" para continuar.
4. **Tipo de Especie:**
 - La siguiente ventana te pedirá que ingreses el tipo de especie del personaje.
 - Escribe el tipo de especie en el campo de texto.
 - Haz clic en el botón "Siguiente" para continuar.

5. Característica Distintiva:

- La última ventana te pedirá que ingreses una característica distintiva del personaje.
- Escribe la característica en el campo de texto.
- Haz clic en el botón "Agregar Personaje" para guardar el personaje en la base de datos.

Confirmación

- Después de hacer clic en "Agregar Personaje", recibirás un mensaje de confirmación indicando que el personaje se ha agregado exitosamente.
- La ventana del juego se cerrará automáticamente.

Verificación

- Para verificar que el personaje se ha agregado correctamente, puedes consultar la tabla personajes en tu base de datos MySQL ejecutando la siguiente consulta:

```
sql
Copiar código
SELECT * FROM personajes;
```

Problemas Comunes

Error de Conexión a la Base de Datos

Si ves un mensaje de error relacionado con la conexión a la base de datos, verifica que:

- La base de datos MySQL está en funcionamiento.
- Las credenciales de conexión (host, user, password, database, port) son correctas.

Datos Incorrectos o Vacíos

Si ingresas datos incorrectos o dejas campos vacíos, es posible que el juego no funcione como se espera. Asegúrate de proporcionar información válida en cada campo.

Explicación Detallada del Funcionamiento del Código "Adivina Quién"

El código implementa un juego de "Adivina Quién" utilizando una base de datos MySQL para almacenar la información de los personajes. A continuación se detalla el funcionamiento del código y se proporcionan instrucciones de uso.

Estructura del Código

Importación de Librerías:

```
import mysql.connector  
  
import json  
  
import random  
  
from tkinter import Tk, Label, Entry, Button, messagebox
```

Se importan las librerías necesarias para conectarse a la base de datos MySQL (mysql.connector), manejar datos en formato JSON (json), generar números aleatorios (random) y crear la interfaz gráfica (tkinter).

Clase JuegoAdivinaQuien:

Esta clase maneja la lógica principal del juego, incluyendo la conexión a la base de datos, la selección de personajes y la interfaz gráfica.

class JuegoAdivinaQuien:

```
    def __init__(self, connection_params):  
        self.conn = mysql.connector.connect(**connection_params)  
        self.cursor = self.conn.cursor()  
  
        self atributos = ["Estatura", "Color de Piel", "Habilidad", "Tipo de Especie", "Característica Distintiva"]  
  
        self.personaje_seleccionado = None  
  
        self.interfaz()
```

`__init__`: Inicializa la conexión a la base de datos y establece los atributos de los personajes. Llama al método `interfaz` para crear la interfaz gráfica.

```
def interfaz(self):  
    self.root = Tk()  
    self.root.title("Adivina Quién")  
    self.preguntar_estatura()  
    self.root.mainloop()
```

interfaz: Configura la ventana principal del juego utilizando tkinter y llama a la primera ventana de entrada de datos.

Métodos de Pregunta:

Se definen métodos específicos para preguntar cada atributo del personaje y capturar la entrada del usuario.

```
def preguntar_estatura(self):  
    self.limpiar_ventana()  
    Label(self.root, text="Estatura (Baja, Mediana, Alta):").pack()  
    self.entry_estatura = Entry(self.root)  
    self.entry_estatura.pack()  
    Button(self.root, text="Siguiente", command=self.preguntar_color_piel).pack()
```

```
def preguntar_color_piel(self):  
    self.limpiar_ventana()  
    self.estatura = self.entry_estatura.get()  
    Label(self.root, text="Color de Piel:").pack()  
    self.entry_color_piel = Entry(self.root)  
    self.entry_color_piel.pack()  
    Button(self.root, text="Siguiente", command=self.preguntar_habilidad).pack()
```

```
def preguntar_habilidad(self):  
    self.limpiar_ventana()  
    self.color_piel = self.entry_color_piel.get()
```

```
Label(self.root, text="Habilidad:").pack()

self.entry_habilidad = Entry(self.root)

self.entry_habilidad.pack()

Button(self.root, text="Siguiente", command=self.preguntar_tipo_especie).pack()
```

```
def preguntar_tipo_especie(self):

    self.limpiar_ventana()

    self.habilidad = self.entry_habilidad.get()

    Label(self.root, text="Tipo de Especie:").pack()

    self.entry_tipo_especie = Entry(self.root)

    self.entry_tipo_especie.pack()

    Button(self.root, text="Siguiente", command=self.preguntar_caracteristica).pack()
```

```
def preguntar_caracteristica(self):

    self.limpiar_ventana()

    self.tipo_especie = self.entry_tipo_especie.get()

    Label(self.root, text="Característica Distintiva:").pack()

    self.entry_caracteristica = Entry(self.root)

    self.entry_caracteristica.pack()

    Button(self.root, text="Agregar Personaje", command=self.agregar_personaje).pack()
```

Método agregar_personaje:

Este método captura todos los datos ingresados y los inserta en la base de datos.

```
def agregar_personaje(self):

    self.caracteristica = self.entry_caracteristica.get()

    self.cursor.execute("SELECT MAX(id) FROM personajes")

    max_id = self.cursor.fetchone()[0] or 0

    nuevo_id = max_id + 1

    nombre = f"Personaje {nuevo_id}"
```

```
atributos = {  
    "Estatura": self.estatura,  
    "Color de Piel": self.color_piel,  
    "Habilidad": self.habilidad,  
    "Tipo de Especie": self.tipo_especie,  
    "Característica Distintiva": self.caracteristica  
}  
  
self.cursor.execute(  
    "INSERT INTO personajes (id, nombre, atributos) VALUES (%s, %s, %s)",  
    (nuevo_id, nombre, json.dumps(atributos))  
)  
  
self.conn.commit()  
  
messagebox.showinfo("Éxito", "Personaje agregado exitosamente")  
  
self.root.destroy()
```

Método limpiar_ventana:

Limpia la ventana actual antes de mostrar la siguiente pregunta.

```
def limpiar_ventana(self):  
    for widget in self.root.winfo_children():  
        widget.destroy()
```

Instrucciones de Uso

Configuración de la Base de Datos:

Asegúrate de tener una base de datos MySQL configurada con una tabla llamada personajes. La tabla debe tener la siguiente estructura:

|

```
CREATE TABLE personajes (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(255),  
    atributos JSON  
);
```

Configuración de Credenciales:

Define las credenciales de conexión a la base de datos en un diccionario `connection_params` en el archivo Python:

```
connection_params = {  
    'host': 'localhost',  
    'user': 'root',  
    'password': 'tu_contraseña',  
    'database': 'nombre_de_tu_base_de_datos',  
    'port': '3306'  
}
```

Ejecución del Código:

Ejecuta el archivo Python desde tu entorno de desarrollo o línea de comandos:

```
python AdivinaQuien.py
```

Interacción con la Interfaz Gráfica:

Aparecerá una ventana solicitando la estatura del personaje. Ingresas el valor correspondiente y presiona "Siguiente".

Repite el proceso para cada atributo solicitado: color de piel, habilidad, tipo de especie, y característica distintiva.

Después de ingresar todos los atributos, presiona "Agregar Personaje" para guardar el nuevo personaje en la base de datos.

Verificación de la Base de Datos:

Puedes verificar que el personaje se ha agregado correctamente consultando la tabla personajes en tu base de datos MySQL.