ZUHAUSE50

A Tech Related Word Guessing Game

Python (Hybrid) Final Project

By Mustafa Lutaaya

Inspiration Behind The Game.

- ZuHause50's Ideas Was Inspired By A Word Game That I, My Young Sister & Elder Brother Used To Play While Growing Up Called "Buyumba" Which Means "Home".

- "Zu Hause" The German Word Simply Means Home While The Number Represents 50 Tech Words.

- During Game Play, Once The Tech Word Is Guessed Correctly, A Display Of The Word's Deutsch Translation & Definition Is Made Available To the Player. Therefore "ZuHause50" Serves As A Fusion Of Tech Words Between Two Languages.

LEARNING OPPORTUNITY

While Building ZuHause50 I Gained Hands-On Experiences With;

Virtual Environments

Object Oriented Programming

File Operations

GitHub

Database Creation, Connection & Handling

GUI Integration

Code Testing

Python Programming Best Practices

PRACTICAL VALUE

Create A Game That Would Help People Transitioning Into Tech In Deutschland Know How Their Favorite Devices, Software & Methods Are Called In German

Project Objectives & Features

- Designed & Developed A Python-Based Word Guessing Game
- Integrated MongoDB Database To Store User Profiles & Enable Level Progression.
- Allowed Users To Manage Profiles Through Creating & Deleting.
- Implemented File Operations & Error Handling Incase The File Isn't Found.
- Used A Responsive GUI" to Enhance Player Experience By Displaying Player Scores , Levels, Hints, Important Notifications, Word Meanings & Messages.

Process

1. Planning:

- Upon Conclusion Of The Decision To Recreate Our Childhood "Buyumba" Game TechWise, It Took Me About 8 Days To Develop Word Guessing Game Code With A Few Errors Here & There Using OOP & A Simple Python List Of About 5 Words(Non-Tech Related) That Ran Through The Command Prompt.
- After That I Decided To Give It More Life By Integrating A GUI.

2. Research:

- I Used DataCamp's GUI Tkinter Tutorial & W3Schools MongoDb Tutorial To Learn Fundamentals Of Tkinter & Databases Respectively.

3. Implementation

- I Then Developed The Game's First Window While Integrating It With The OOP.
- Later Opened A MongoDB Account That I Connected To The Game To Initiate Profiles.
- I Then Created A CSV File From Where I Would Be Getting The Words From.
- Then Made My First Commit To Github Before I Went On To Further Develop The Game.

4. Testing

- Played All Words Personally To Ensure The Game Ends & Also Tested DB Connection

Challenges & Solutions

Challenge 1: Integrating OOP Into The Tkinter GUI

Solution: Since It Was Something New To Me, Apart From DataCamp's GUI Tkinter Tutorial Where I Learned The Basics Of Tkinter, I Had Too Go Through More Tutorials Of Tkinter That Helped Me Understand How It Could Be Done & Initiate More Involvement

Challenge 2: Word Repetition Which Provoked An Unending Game

Solution: Since I Had Created Certain Words With Spaces And Then Programmed The Space To Be Viewed As An Underscore During Game Play. Saving The Played Words Became A Problem Since They Didn't Look Alike Anymore Hence Repetition & An Unending Game.

Challenge 3: Flow Of The Code:

Solution: I Rearranged The Code Again In A Logical Way For The Programme To Run.

Notable Resources

- **W3Schools MongoDB Tutorial** :https://www.w3schools.com/mongodb/index.php
- **Datacamp's Gui Tkinter Tutorial :** https://www.datacamp.com/tutorial/gui-tkinter-python
- Linie 1 A1 - 1 Deutsch Books & ChatGPT For German Words & Translation

ZuHause50 Future State

Future Improvements:

- Add Sound Effects For Wrong & Correct Moves.
- Add Animations For Game End & Correct Word Guess.
- Make The Game Both Sided For Both Language Users To Play.
- Display High Scorers & Add More Words.
- Adding More Buttons For Simplicity Like Restart, Back and so forth

Uses:

- Game Can Be Used For Educational Purposes & Fun Too.

DEMONSTRATION

THANK YOU!