Who's that Pokemon **05.01.2025 Inf. Pk. Ersatzklausur**

Michelle Wallmann, Leonie Riedel

Als Projekt hatten Michelle und ich das Ziel, "Who's that Pokémon", aus dem Pokémon Anime (hier ein Video sollte man es nicht kennen: https://www.youtube.com/watch?v=er_amDd6U4U), als kleines Videospiel nachzustellen. Unsere Motivation dafür war, dass wir bereits als Kinder den Pokémon Anime geguckt haben und damit, sowie mit Who's that Pokémon aufgewachsen sind, als uns dann auch noch aufgefallen ist das eine nachstellung in Python vom Code her ziemlich ähnlich wie Flaggenraten (einem der vorgeschlagenen Projektthemen), nur mit einem größeren Dataset wäre, waren wir sofort von der Projektidee überzeugt.

Das Spielkonzept ist relativ simpel gehalten, zuerst wählt der Endnutzer zwischen den **2 verschiedenen Spielmodi**, Normal oder Silhouetten, daraufhin gelangt der Nutzer in das eigentliche Spiel. Je nach Spielmodus wird dem Nutzer entweder ein **Bild von einem Pokémon** in Farbe oder komplett schwarz angezeigt. Der Nutzer hat dann **10 Sekunden** um den Namen des Pokémon in das Eingabefeld einzugeben und "Check" zu drücken. Ist der Name richtig, erhält der Nutzer einen Punkt und das nächste Pokémon wird angezeigt. Der Nutzer darf dabei sowohl den englischen als auch den deutschen Namen eingeben. Ist der Name falsch, oder die Zeit läuft aus, werden die Punkte des Nutzers auf 0 zurückgesetzt und das nächste Pokémon wird angezeigt. Die höchste Punktzahl wird in einem Highscore gespeichert. Das Spiel endet, wenn alle Pokémon einmal angezeigt wurden. Ziel des Spiels ist es also, so viele Punkte wie möglich (max. 1025 weil es 1025 Pokémon gibt) zu erhalten.

Um die **Benutzerfreundlichkeit** zu erhöhen, gibt es nur die minimal benötigte Anzahl an Eingabemöglichkeiten für Nutzer und alle Button sind beschriftet. Des Weiteren zeigt der Startbildschirm **Spielbeschreibungen**, um den Nutzern noch einmal das Spiel zu erklären.

Es wurde **tkinter für die grafische Oberfläche** benutzt. Um die Einbindung von .webp Bildern möglich zu machen, wurde pillow (importiert als PIL) benutzt, dies war nötig, um die Größe der 2050 Bilder drastisch zu reduzieren.
Letztlich wurde pyinstaller benutzt, um **ausführbare Dateien** vom Code zu generieren.

Die ganze Logik des Spiels ist in einer "class" und die einzelnen Funktionen werden alle mit "def" definiert, diese **objektorientierte Programmierung** erlaubt einfache Übersicht und Modifizierung des Codes. Z.B. sind in der Definition der Initialisierung des Spiels alle Variablen. Damit kann man sehr einfach Global verändern, wieviel Zeit z.B. Der Timer haben soll. Des Weiteren ruft die Funktion alle anderen Funktionen auf, die benötigt werden, um das Spiel zu starten.

```
class whosThatPokemon:
# initialise game function
    def __init__(self, root):
            self.root = root
            self.root.title("Who's the Pokémon?") #displayed in window
title
            self.root.geometry("700x500") #might experiement with
different sizes if time = True

            # all the important variables
            self.current_pokemon_name = "" # this is the english name that
will be checked
            self.current_pokemon_name_german = "" # same thing but german
(note from leonie: ew)
            self.current_score = 0 # tracks consecutive right guesses
            self.high_score = 0 # highest streak max should be smth like
1025/26
            self.time_left = 10 #timer 1 unit is 1 second
            self.remaining_pokemon = pokemon_data.copy() # use a temp list
to make sure we only display each pokemon once
            self.timer_id = None  # timer ID to cancel the previous one to
prevent a bug where timers would get faster
            self.mode = "normal"  # default mode is colour images

            # folder for colour images
            self.image_folder = "pokemon_images_full"

            # startup the selection screen
            self.setup_start_screen()
```

Eine weitere wichtige Funktion ist “load_random_image”, diese lädt ein zufälliges Pokémon, entfernt dieses aus einer temporären Liste, damit es nicht nochmal ausgewählt werden kann und lässt es dann anzeigen. Diese Funktion stellt den Kern des Spiels dar. Sie ruft außerdem die Funktion auf, das Spiel zu beenden, sollte es keine Pokémon mehr geben. (Hier verkürzt)

```
    def load_random_image(self):

        if not self.remaining_pokemon:
            self.end_game()
            return
        # pick a random Pokémon
        random_pokemon = random.choice(self.remaining_pokemon)
        self.remaining_pokemon.remove(random_pokemon)
        self.current_pokemon_name = random_pokemon["name"]
        self.current_pokemon_name_german = random_pokemon["name_german"]
```

```
        # load and display the image
        image = Image.open(image_path)
        image = image.resize((200, 200))  # resize so its all one size
        photo = ImageTk.PhotoImage(image)
        # update the image label
        self.image_label.config(image=photo)
        self.image_label.image = photo
        # reset the timer
        self.time_left = 10
        self.update_timer()
```

Eines der Probleme, auf die wir bei der Programmierung gestoßen sind, ist, dass mit der originalen Implementierung von Timern (Einfach eine Variable die runterzählt) der Timer immer schneller wurde, wenn man zu schnell geantwortet hatte. Das Problem haben wir gelöst indem wir jedem Timer eine Unique ID zuordnen, wenn wir ihn initialisieren, sodass der Timer für jedes Pokemon separat ist.

```
    def update_timer(self):
        """Updates the timer and moves to the next Pokémon if time runs
out."""
        if self.time_left > 0:
            self.time_left -= 1
            self.update_status()
            self.timer_id = self.root.after(1000, self.update_timer)  #
store the timer ID
def load_new_pokemon(self):
        """loads a new Pokémon."""
        if self.timer_id:
            self.root.after_cancel(self.timer_id)  # Cancel the previous
timer
        self.load_random_image()
```

Das Projekt hat auf jeden Fall das ursprüngliche Ziel erreicht. Wir haben nun Who's that Pokémon voll funktionsfähig als Python Anwendung. Wir haben den Umgang mit Dictionaries (Key:Value Pairs) in Python gelernt und wie man effektiv mit den Pack-Manager von tkinter arbeitet.