

# SPORTS TRIVIA

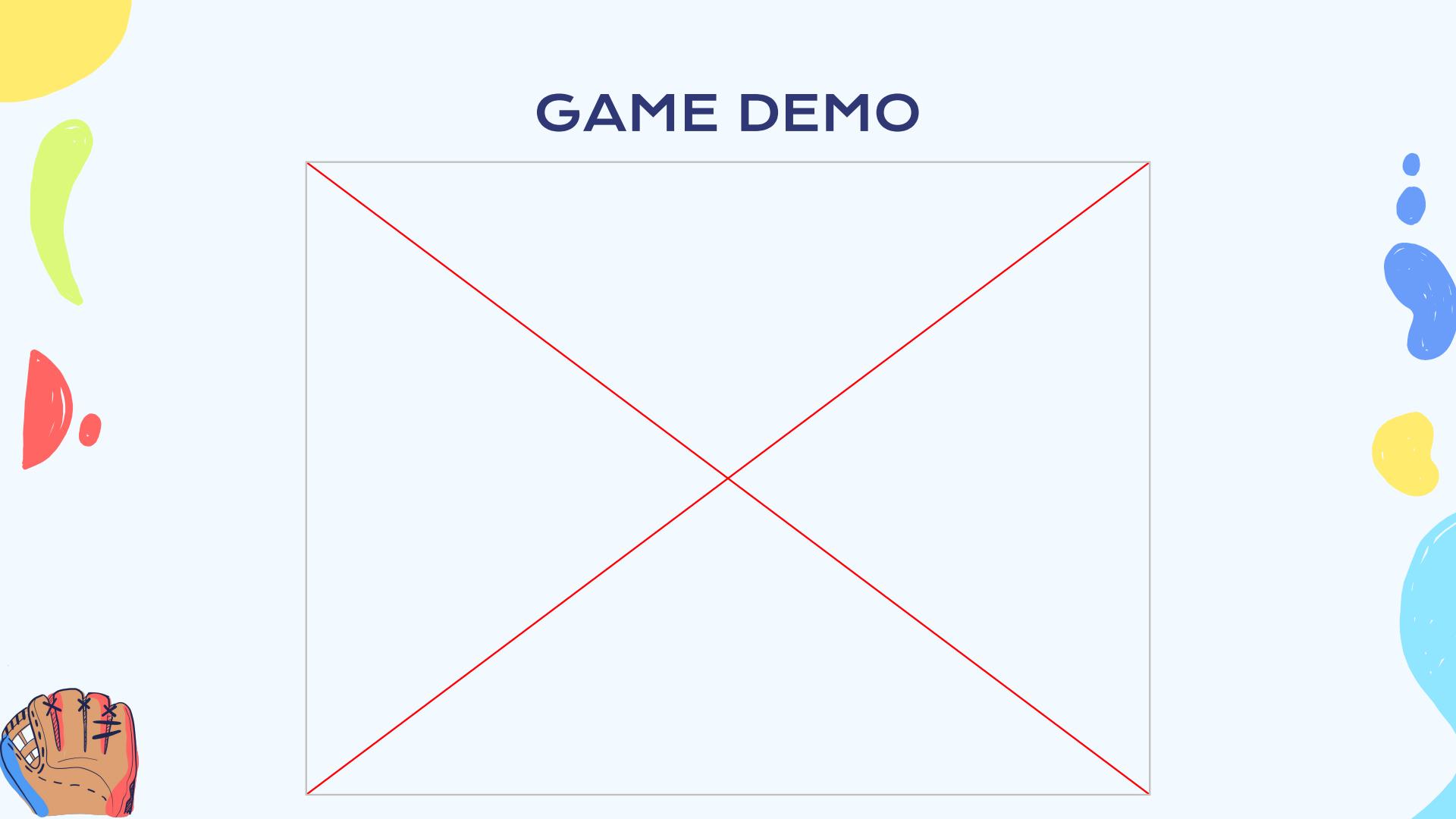
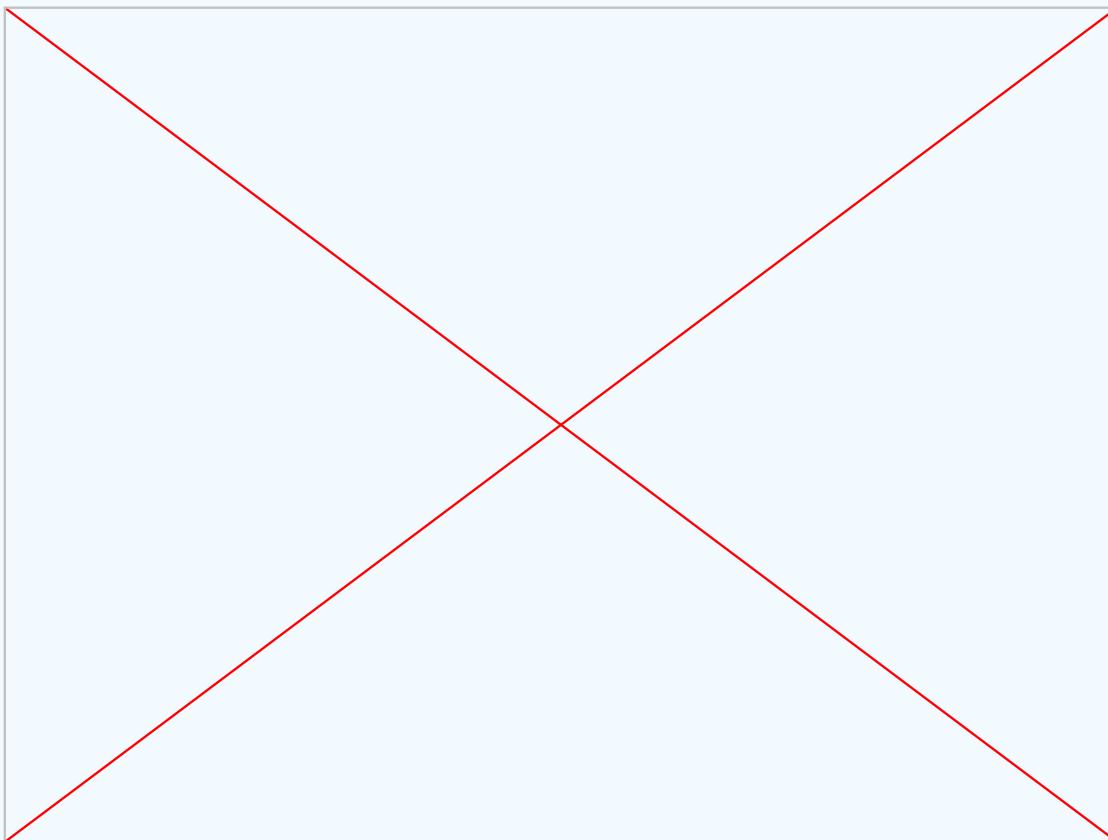
By: Abigail, Becky, Lizzie



# INSPIRATION

- Our inspiration mainly stems from the fact that we are all athletes, more specifically soccer and volleyball players.
- We wanted to recreate something that we also would enjoy playing ourselves.
- When we were brainstorming we wanted to make sure to include our favorite sports.
- The layout of our game is based off of any traditional trivia game you see everyday.

# GAME DEMO



# DEMO RECAP

- MAIN FUNCTIONS

- Start Game
- Choose Category
- Answer questions one by one
- Submit answer button
- Lets user know if right or wrong (green/red)
- Shows right answer or says correct
- At end shows results
- Option to return back to categories

# WORK DISTRIBUTION & COLLAB

- We collaborated on the types of questions and categories to include.
- Then we divided the responsibilities into:
  - Making dictionary
  - Starting tkinter/welcome screen
  - Questions and answer part of game
- We made sure to communicate any changes outside of class through email and met up when needed.

# Game Code - Start

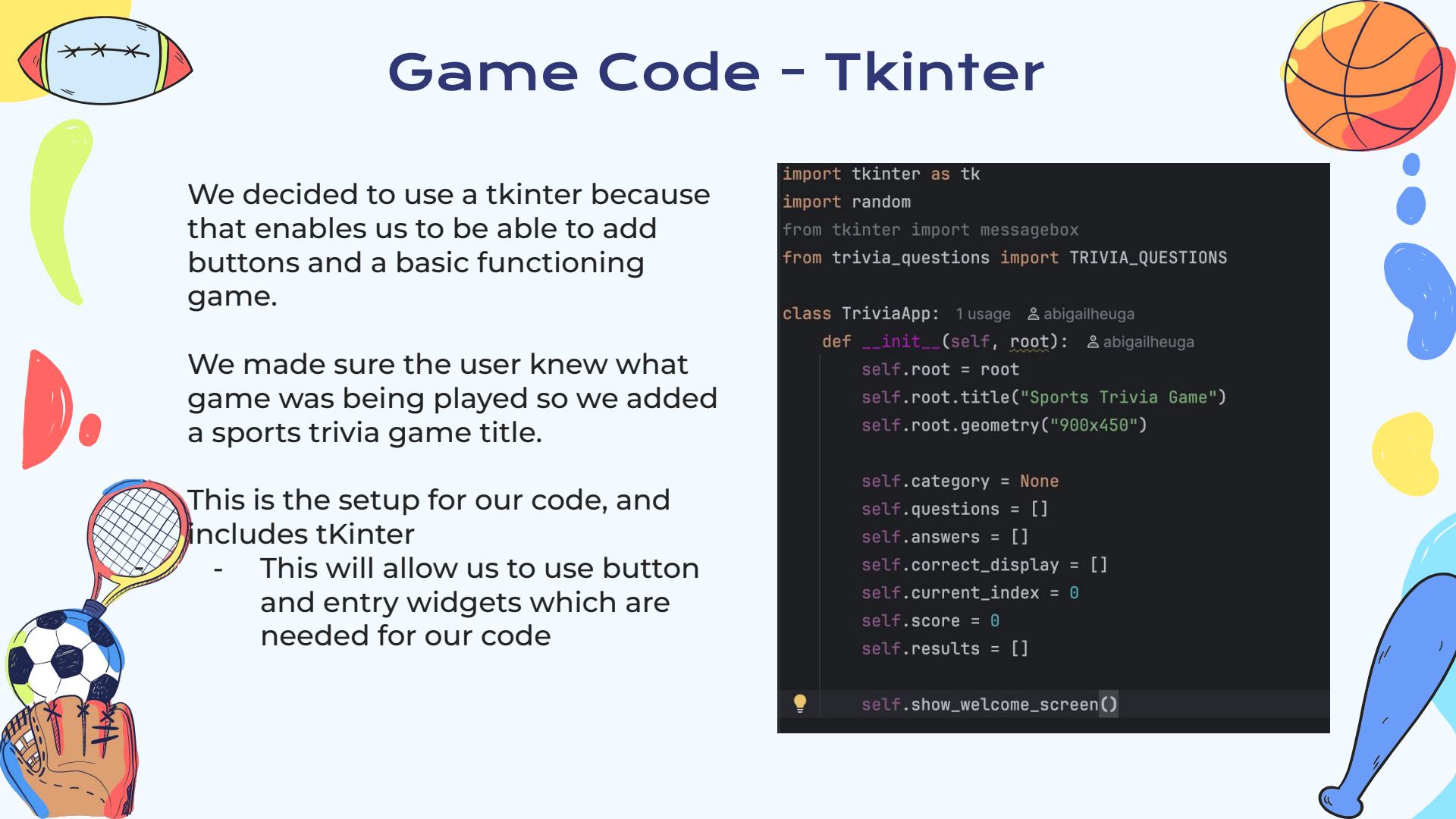
```
TRIVIA_QUESTIONS = {
    "Football": {
        1: {"question": "Which NFL team became the first team to win six Superbowls?", "answer": {"display": "Pittsburgh Steelers", "accepted": ["Pittsburgh Steelers", "Steelers"]}}, 2: {"question": "Who holds the NFL record for most career passing yards?", "answer": {"display": "Tom Brady", "accepted": ["Tom Brady", "Brady"]}}, 3: {"question": "What year did the first Superbowl take place?", "answer": {"display": "1967", "accepted": ["1967"]}}, 4: {"question": "Which player is known for the famous 'immaculate reception'?", "answer": {"display": "Franco Harris", "accepted": ["Franco Harris", "Harris"]}}, 5: {"question": "Which NFL franchise has the most regular season wins in franchise history?", "answer": {"display": "Chicago Bears", "accepted": ["Chicago Bears", "Bears"]}}},
```

Created a dictionary that includes all questions and answers for our selected categories with multiple accepted answers.

On the screen is an example of the football trivia dictionary.

We put this in it's own file so that we can easily import it into our functions without needing to copy it over.

# Game Code - Tkinter



We decided to use a tkinter because that enables us to be able to add buttons and a basic functioning game.

We made sure the user knew what game was being played so we added a sports trivia game title.

This is the setup for our code, and includes tKinter

- This will allow us to use button and entry widgets which are needed for our code

```
import tkinter as tk
import random
from tkinter import messagebox
from trivia_questions import TRIVIA_QUESTIONS

class TriviaApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Sports Trivia Game")
        self.root.geometry("900x450")

        self.category = None
        self.questions = []
        self.answers = []
        self.correct_display = []
        self.current_index = 0
        self.score = 0
        self.results = []

    def show_welcome_screen()
```

# Game Code - Main Code

```
def start_category(self, category): 1 usage  ↵ abigailheuga
    self.category = category

    self.questions = []
    self.answers = []
    self.correct_display = []
    self.results = []
    self.current_index = 0
    self.score = 0

    qa_pairs = []
    for i in range (1, 6):
        q = TRIVIA_QUESTIONS[category][i]["question"]
        ans = TRIVIA_QUESTIONS[category][i]["answer"]

        if isinstance(ans, dict):
            display_answer = ans["display"]
            normalized = [a.lower() for a in ans["accepted"]]

        elif isinstance(ans, list):
            display_answer = ans[0]
            normalized = [a.lower() for a in ans]
        else:
            display_answer = ans
            normalized = ans.lower()

        qa_pairs.append((q, normalized, display_answer))

    random.shuffle(qa_pairs)

    self.questions = [q for (q, norm, disp) in qa_pairs]
    self.answers = [norm for (q, norm, disp) in qa_pairs]
    self.correct_display = [disp for (q, norm, disp) in qa_pairs]
```

The majority of our code consisted of the question and answer portion where we were able to apply to every category.

We made sure that one question was asked at a time.

The answer is able to identify upper and lower cases, along with answers that like 'bears' that is the same as 'chicago bears'.

The questions are also randomized so they are not the same order every time you play.

# Game Code - Check Answers

We used if/else statements to check whether the user's answer is correct

- Works for both single correct answers and multiple acceptable answers

It updates the interface based on correctness

- Answer box turns green(correct)/red(incorrect), updates score, and displays the correct answer if incorrect response

```
def check_answer(self): 1 usage  ↵ abigailheuga
    user = self.entry.get().strip()
    user_answer = user.lower()

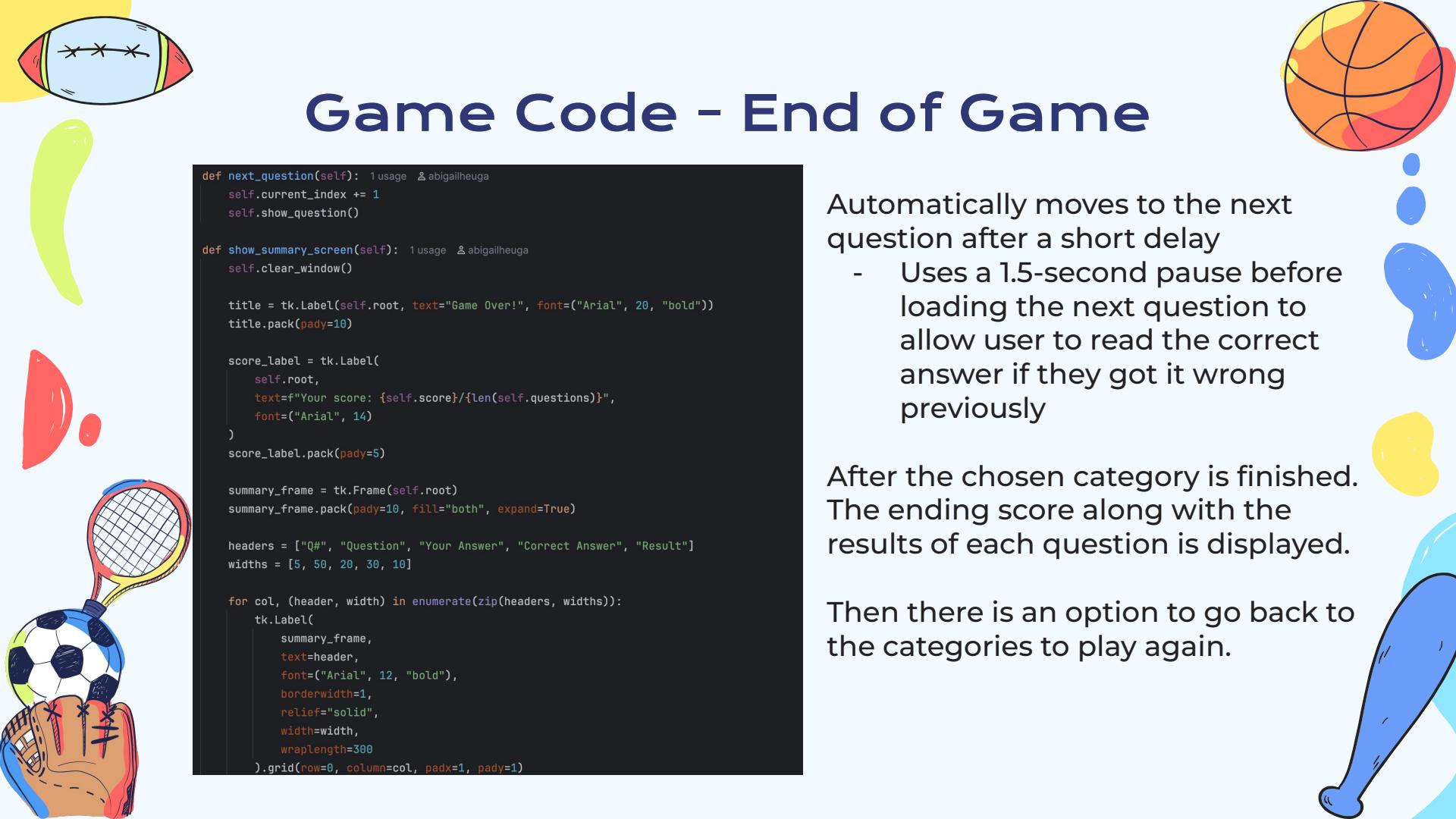
    correct = self.answers[self.current_index]
    display_correct = self.correct_display[self.current_index]

    if isinstance(correct, list):
        is_correct = user_answer in correct
    else:
        is_correct = user_answer == correct

    if is_correct:
        self.entry.config(bg="green")
        self.correct_label.config(text="")
        self.score += 1
    else:
        self.entry.config(bg="red")
        self.correct_label.config(text=f"Correct Answer: {display_correct}")

    self.results.append({
        "question": self.questions[self.current_index],
        "user_answer": user if user else "(blank)",
        "correct_answer": display_correct,
        "is_correct": is_correct
    })

    self.root.after(1500, self.next_question)
```



# Game Code - End of Game

```
def next_question(self): 1 usage & abigailheuga
    self.current_index += 1
    self.show_question()

def show_summary_screen(self): 1 usage & abigailheuga
    self.clear_window()

    title = tk.Label(self.root, text="Game Over!", font=("Arial", 20, "bold"))
    title.pack(pady=10)

    score_label = tk.Label(
        self.root,
        text=f"Your score: {self.score}/{len(self.questions)}",
        font=("Arial", 14)
    )
    score_label.pack(pady=5)

    summary_frame = tk.Frame(self.root)
    summary_frame.pack(pady=10, fill="both", expand=True)

    headers = ["Q#", "Question", "Your Answer", "Correct Answer", "Result"]
    widths = [5, 50, 20, 30, 10]

    for col, (header, width) in enumerate(zip(headers, widths)):
        tk.Label(
            summary_frame,
            text=header,
            font=("Arial", 12, "bold"),
            borderwidth=1,
            relief="solid",
            width=width,
            wraplength=300
        ).grid(row=0, column=col, padx=1, pady=1)
```

Automatically moves to the next question after a short delay

- Uses a 1.5-second pause before loading the next question to allow user to read the correct answer if they got it wrong previously

After the chosen category is finished. The ending score along with the results of each question is displayed.

Then there is an option to go back to the categories to play again.

# THANK YOU

