NBA Play-by-Play Animation Tracker - Python Project

Project: NBA Play-by-Play Animation Tracker

Objective:

Create a Python program that fetches play-by-play data for an NBA game and animates key moments (like scoring, assists, rebounds, fouls) as a simple visual timeline. The animation will display the flow of the game, showing player actions, scores, and possession changes.

Step 1: Fetching Play-by-Play Data

Use an NBA stats API to fetch play-by-play data for a particular game. Libraries needed: requests for fetching API data and json for processing the data.

Example code for fetching play-by-play data:

```
import requests


def get_play_by_play(game_id):
    url = f"https://www.balldontlie.io/api/v1/games/{game_id}/play_by_play"
    response = requests.get(url)
    plays = response.json()
    return plays['data'] if 'data' in plays else None
```

Step 2: Visualizing the Court

Use matplotlib to draw a basic basketball court and visualize the events like shots, rebounds, assists, etc.

Example code for drawing the court:

```
import matplotlib.pyplot as plt


def draw_court():
    fig = plt.figure(figsize=(6, 7))
    ax = fig.add_subplot(1, 1, 1)
    ax.plot([0, 50], [0, 0], color='black')  # Baseline
    ax.plot([0, 50], [94, 94], color='black')  # Opposite baseline
    # Key area
    ax.plot([17, 33], [19, 19], color='black')
    ax.plot([17, 33], [33, 33], color='black')
    ax.plot([25, 25], [19, 33], color='black')
    return fig, ax
```

Step 3: Animating the Play-by-Play Data

Use matplotlib's animation capabilities to animate key moments like scoring, assists, and rebounds over time.

Example code for animating events:

```
import matplotlib.animation as animation
```

```
def animate_play_by_play(play_data):
    fig, ax = draw_court()

    def update_frame(i):
        ax.clear()
        draw_court()
        current_play = play_data[i]
        event_type = current_play['event_type']
        if event_type == 'made_shot':
            player = current_play['player']['last_name']
                ax.text(20, 50, f"Shot Made by {player}!", color='green', fontsize=14,
ha='center')
        elif event_type == 'rebound':
            player = current_play['player']['last_name']
                  ax.text(20, 50, f"Rebound by {player}", color='blue', fontsize=14,
ha='center')
        elif event_type == 'assist':
            player = current_play['player']['last_name']
                 ax.text(20, 50, f"Assist by {player}", color='orange', fontsize=14,
ha='center')

           ani = animation.FuncAnimation(fig, update_frame, frames=len(play_data),
repeat=False)
    plt.show()
```

Step 4: Bonus - Add a Scoreboard

Display the score in the corner of the screen that updates with each play. Example code for updating the scoreboard:

```
def animate_play_by_play_with_score(play_data):
    fig, ax = draw_court()
    score = {'home': 0, 'away': 0}

    def update_frame(i):
        ax.clear()
        draw_court()
        current_play = play_data[i]
        if current_play['event_type'] == 'made_shot':
            team = current_play['team']['abbreviation']
            if team == 'home':
                score['home'] += current_play['points']
            else:
                score['away'] += current_play['points']
            ax.text(20, 50, f"{team} scores!", color='green', fontsize=14, ha='center')
              ax.text(2,  90,  f"Score:  Home  {score['home']}  -  Away  {score['away']}",
fontsize=12)

          ani   =   animation.FuncAnimation(fig,   update_frame,   frames=len(play_data),
repeat=False)
    plt.show()
```

Step 5: Running the Program

Allow the user to select a game by inputting a game ID or choosing from recent games, fetch the data, and run the animation.

Example code:

```
def main():
    game_id = input("Enter the game ID: ")
    play_data = get_play_by_play(game_id)
    if play_data:
        animate_play_by_play_with_score(play_data)
    else:
        print("Could not fetch data for the given game ID.")


if __name__ == "__main__":
    main()
```