CISC-505-Object Oriented Programming

Team Members:

- **Abdul Rafeh**
- **Samith Kamarthi**
- **Kerolos Gobrial**
- **Komi Amevigbe**

🔧 STRUCTURE OVERVIEW

There are **4 major parts** in this code:

1. **Data Classes** – Person and Scenario
2. **Input & Generation Functions** – to gather person details (manual or random)
3. **Decision Logic** – to score passengers vs. pedestrians
4. **Visualization** – to display the scenario using matplotlib

---

1. ✅ CLASSES

class Person

This represents any individual (passenger or pedestrian) in the scenario.

Attributes:

- role: "passenger" or "pedestrian"
- category: 'child', 'adult', 'elderly', ...
- gender: 'male' or 'female' (optional)
- is_pregnant: True/False (only for females and adults)
- is_criminal: True/False (for humans)

__repr__: Nicely formats person description when printed.

---

class Scenario

Stores a group of passengers and pedestrians along with a barricade indicator.

- passengers: list of Person
- pedestrians: list of Person
- has_barricade: True or False

---

2. 📥 INPUT AND GENERATION FUNCTIONS

input_person(role)

Prompts user to input details of one person manually:

- Asks for category, gender, pregnancy (if female adult), criminal status.

---

get_group(role, method)

Generates a list of people for either passengers or pedestrians.

- If method='random', calls random_person()
- If method='manual', uses input_person() in a loop

---

random_person(role)

Randomly creates a Person with appropriate attributes:

- Randomly selects a category
- Assig gender (if human)
- Randomizes pregnancy and criminal status (if applicable)

---

3. ⚖ DECISION-MAKING

decide(scenario)

Determines whether the car should **stay** (protect passengers) or **swerve** (protect pedestrians) based on ethical scoring.

score(group, is_pedestrian)

- Assigns scores based on ethical value:
  - child = 3, adult = 2, elderly = 1, doctor = 4, athlete = 3, animals = 1
  - +2 for pregnancy
  - -2 if criminal
  - -5 penalty for pedestrians if there's a barricade

The side with the **higher score is spared**:

- If **pedestrians score more** → "swerve" (save pedestrians)
- Else → "stay" (save passengers)

---

4. 📊 PLOTTING THE SCEARIO

add_person_marker(ax, x, y, person)

Plots an individual with a **marker and color** based on their category:

- * for doctor, ^ for child, D for elderly, > for dog, < for cat, etc.
- Criminals shown as 'X' in black

---

plot_scenario(passengers, pedestrians, has_barricade)

Displays a **top-down visual** of:

- Pedestrians at the top
- Car with passengers in the middle
- Red barricade if present
- Legend on the right

Uses matplotlib patches and markers to show people with different icons.

---

5. □ MAIN EXECUTION LOGIC

At the bottom, the script:

1. Prints welcome message

2. Asks if scenario should be generated randomly or manually
3. Gets both groups
4. Asks if there's barricade (if manual)
5. Builds the Scenario
6. Displays the scenario (print)
7. Decides what the car should do (decide)
8. Plots the scenario (plot_scenario)
9. Prints the final ethical decision

💡 **EXAMPLE OUTPUT**

```
⚖️ Welcome to the Moral Machine Simulator

Do you want to generate a random scenario? (yes/no): yes



🚗 Passengers:

 - adult female pregnant

 - dog

🚶 Pedestrians:

 - doctor male

 - child female

🚧 Barricade Present: Yes



Score — Passengers: 5, Pedestrians: 4

🚦 Ethical Decision: STAY (Car will stay)
```
