Project Title: Pac-Man Game using Python and Pygame

Description

This project is a simple Pac-Man-style game created in Python with the Pygame library, offering a basic yet interactive experience. The objective is to navigate Pac-Man, controlled by the player, around the screen to collect pellets while evading moving ghosts. If Pac-Man collides with a ghost, the game ends, and the player can choose to restart. Scoring is implemented based on the number of pellets collected, with a winning condition if all pellets are collected.

Technologies Used

- **Python**: A versatile, high-level programming language that is easy to learn and use, ideal for game development, especially for beginners.
- **Pygame**: A Python library specifically developed for creating 2D games. It includes modules for graphics, sound, and handling user input, providing essential functionalities for game development.

Code Functionality Overview

Key Components

1. **Constants and Settings**:
   - **Screen and Character Dimensions**: Set with constants like SCREEN_WIDTH, SCREEN_HEIGHT, PACMAN_RADIUS, and GRID_SIZE to define the playing field and character sizes.
   - **Colors**: Defined using RGB values for Pac-Man, ghosts, pellets, and the game screen.
2. **Classes**:
   - **Player (Pac-Man)**: The Player class represents the main character, allowing the player to move using the arrow keys and restricting movement within screen boundaries.
   - **Ghosts**: The Ghost class provides autonomous movement for ghost characters. Each ghost randomly changes direction upon colliding with screen edges.
   - **Pellets**: The Pellet class defines collectible items placed on the screen in a grid-like formation. Each pellet is removed upon collision with Pac-Man.
   - **Button**: Implements a button with a hover effect and a clickable area to end the game.
3. **Game Mechanics**:
   - **Pellet Collection**: Each pellet collected increases the player’s score.
   - **Ghost Collision**: If Pac-Man collides with a ghost, the game ends.
   - **Winning Condition**: Collecting all pellets triggers a win state.

4. **Sound Integration**:
    - Background music is loaded and played continuously during the game using the pygame.mixer module. The music stops when the game ends.
5. **Game Loop**:
    - The game_loop function handles the main game actions, including updating player and ghost positions, drawing elements, detecting collisions, and refreshing the display.
    - Restart and quit options are provided when the game ends.
6. **Main Function**:
    - Starts the background music and calls the game loop to initialize the game.

Access Instructions

1. **Requirements**:
    - **Python 3**: Ensure Python 3.x is installed on your system.
    - **Pygame Library**: Install Pygame using the command:

      {pip install pygame}
2. **Download**:
    - Download the project files, including pacman_game.py (the main game script) and any media files (such as the music file) to the same directory on your system.
3. **Running the Game**:
    - Open a terminal or command prompt in the project directory and run the following command:

      {python pacman_game.py}
    - The game window will open, and the background music will start automatically.
4. **Game Controls**:
    - Use arrow keys to move Pac-Man in all four directions.
    - Click the **Quit** button or press the close button on the window to exit the game.
    - **R Key**: Restarts the game upon a game-over or win.
5. **Objective**: Collect all pellets while avoiding ghosts to win. If Pac-Man collides with a ghost, the game is over, and the player’s final score is displayed.