SULEMAN ROMAIL 261936310

The code is a simple space-invading game implemented using the Pygame library in Python. Here is an overview of the code structure and relationships:

1. Importing Libraries: The necessary libraries, including Pygame, are imported.

2. Initializing Pygame: Pygame is initialized, and the game window is created with a specified width and height.

3. Loading Images: The images used in the game, including the player spaceship, enemy spaceship, lasers, and background, are loaded from the image files.

4. Laser Class: Defines the Laser class, which represents the lasers used to destroy enemies and asteroids in the game. It has methods for drawing the laser, moving it, checking if it's off the screen, and detecting collisions with other objects.

5. Ship Class: Defines the Ship class, which is the parent class for both the player and enemy spaceships. It contains common attributes and methods for the ships, such as position, health, images, lasers, shooting, and collision detection.

6. Player Class: Inherits from the Ship class and represents the player's spaceship. It adds functionality specific to the player, such as health bar rendering.

7. Enemy Class: Inherits from the Ship class and represents the enemy spaceship. It adds functionality specific to enemies, such as movement and shooting.

8. Asteroid Class: Inherits from the Ship class and represents the asteroids in the game. It adds functionality specific to asteroids, such as movement.

9. Helper Function: Defines a helper function `collide` that checks for collisions between two objects based on their masks.

10. Main Function: Defines the main game loop. It initializes variables, including fonts, enemies, asteroids, and various velocities. The game loop handles events, updates the game state, checks for collisions, and redraws the window.

11. Redraw Window Function: Renders the game's visuals by drawing the background, labels for lives and level, enemies, asteroids, player ship, health bar, and the "You Lost!" message if the player has lost the game.

12. Main Menu Function: Implements a simple main menu loop that waits for a mouse button press to start the game.

13. Running the Game: The `main\_menu` function is called to start the game.

Overall, the code follows an object-oriented approach, with separate classes for different entities in the game, such as lasers, ships, enemies, and asteroids. The classes inherit from each other to reuse and extend functionality. The main game loop handles the game logic and rendering.