Overview

This Snake game uses the PyGame library for graphics, user input, and game logic. The objective is to control a snake, eat food to grow longer, and avoid collisions with the boundaries or the snake's own body. The game includes features like a start menu, dynamic scoring, and an option to restart after losing.

Structure of the Program

1. **Constants and Initialization**
    - Define screen dimensions, colors, and block size.
    - Set up PyGame, fonts, and the game clock.
2. **Core Functions**
    - `start_menu()`: Display the welcome screen with options to start or quit.
    - `game_loop()`: The main game logic, including player input, movement, collisions, and rendering.
    - `display_score(score)`: Render the current score on the screen.
    - `draw_snake(block_size, snake_list)`: Draw the snake's body segments on the screen.
    - `message(msg, color, position, font)`: Render text messages on the screen.
    - `check_food_position(snake_list)`: Ensure new food is not generated on the snake.
    - `reset_game()`: Reinitialize the game state for a new game.
3. **Gameplay Flow**
    - Start the game from the `start_menu()`.
    - Enter `game_loop()` to play.
    - Handle game-over state and allow restarting or quitting.
4. **Endgame Features**
    - Display the final score and offer a restart option.

Pseudocode

```
1. Initialization
Import pygame, time, random, and sys
Initialize PyGame
Set up constants:
   WIDTH, HEIGHT, BLOCK_SIZE, FPS
   Colors (WHITE, BLACK, RED, GREEN, BLUE, GRAY)
Initialize game screen with WIDTH and HEIGHT
Initialize fonts for displaying text
Set up the game clock for controlling frame rate
```

```
2. Helper Functions
Define `display_score(score)`:
   Render score text using score_font
   Blit the score at the top-left corner of the screen

Define `draw_snake(block_size, snake_list)`:
   FOR each segment in snake_list:
      Draw a rectangle at the segment's coordinates

Define `message(msg, color, position, font)`:
   Render the given message text in the specified color and font
   Blit the message at the given position

Define `start_menu()`:
   WHILE True:
      Fill screen with BLACK
      Display welcome message and instructions
      Update the screen
      FOR each event in pygame.event.get():
         IF user clicks Quit:
            Exit program
         IF user presses "S":
            RETURN to start the game
         IF user presses "Q":
            Exit program

Define `check_food_position(snake_list)`:
   REPEAT:
      Generate a random position for food (aligned to BLOCK_SIZE grid)
   UNTIL food position is not in snake_list
   RETURN valid food position

3. Main Game Logic
Define `game_loop()`:
   Initialize snake position at center of the screen
   Initialize direction changes as (0, 0)
   Initialize snake list and length
   Generate initial food position using `check_food_position()`
   Set game_over and game_close flags to False

   WHILE game_over is False:
      WHILE game_close is True:
         Fill screen with WHITE
         Display game-over message and final score
```

```
            Update the screen
            FOR each event in pygame.event.get():
                IF user presses "Q":
                    Set game_over to True
                    RETURN
                IF user presses "C":
                    Call `game_loop()` to restart
        FOR each event in pygame.event.get():
            IF user clicks Quit:
                Set game_over to True
            IF user presses an arrow key:
                Update direction (prevent opposite direction changes)
        Update snake's head position based on direction
        IF snake collides with boundaries or itself:
            Set game_close to True
        Append new head position to snake_list
        IF snake length exceeds the current length:
            Remove the last element of the list
        IF snake's head matches food position:
            Generate new food using `check_food_position()`
            Increase snake length by 1
        Clear screen
        Draw food
        Draw snake using `draw_snake()`
        Display current score using `display_score()`
        Update the display
        Control frame rate using clock.tick(FPS)
    Exit PyGame
    Exit program

4. Main Function
Define `main()`:
    Call `start_menu()`
    Call `game_loop()`

IF __name__ == "__main__":
    Call `main()`
```

Key Changes in Pseudocode

1. **Food Validation**: Added a helper function `check_food_position()` to ensure food does not overlap with the snake's body.
2. **Restart Logic**: Avoided recursion in `game_loop()` by allowing state resetting.

3. **Input Handling**: Prevent opposite direction changes using a check during input capture.
4. **Menu Design**: Included a welcome message and navigation keys in `start_menu()`.