ERLAN AKBARALIEV (RTSUMV)

1. Task Description

Snake

We have a rattlesnake in a desert, and our snake is initially two units long (head and rattler). We have to collect with our snake the foods on the level, that appears randomly. Only one food piece is placed randomly at a time on the level (on a field, where there is no snake). The snake starts off from the center of the level in a random direction. The player can control the movement of the snake's head with keyboard buttons. If the snake eats a food piece, then its length grow by one unit. It makes the game harder that there are rocks in the desert. If the snake collides with a rock, then the game ends. We also lose the game, if the snake goes into itself, or into the boundary of the game level. In these situations show a popup messagebox, where the player can type his name and save it together with the amount of food eaten to the database. Create a menu item, which displays a highscore table of the players for the 10 best scores. Also, create a menu item which restarts the game.

2. Analysis

Main Class

Purpose: Entry point of the application.

GameFrame Class

Purpose: Manages the game window and UI.

Key Fields: JFrame frame, GamePanel gamePanel.

Key Methods:

 GameFrame(String dbURL, String dbUser, String dbPwd)

 createMenu()

 displayHighscores()

GamePanel Class

Purpose: Handles game logic and rendering.

Key Fields: int x[], y[], int applesEaten, appleX, appleY, x, y, char direction, boolean running, Timer timer, Random random, String dbURL, dbUser, dbPwd.

Key Methods:

GamePanel(String dbURL, String dbUser, String dbPwd): Initializes the panel.

startGame(): Starts the game.

move(), checkApple(), checkCollisions(): Core game logic.
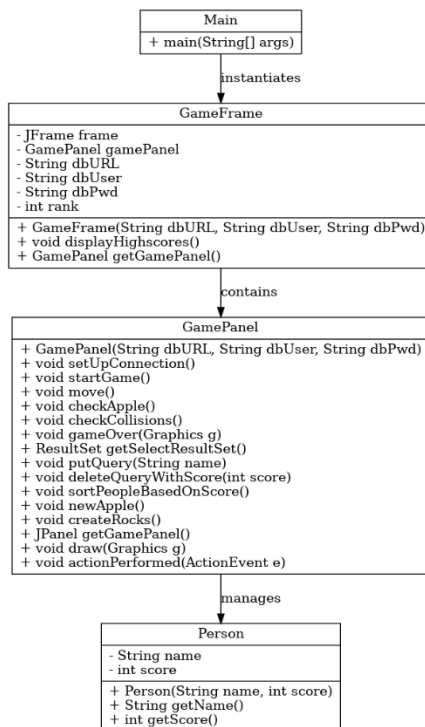
newApple(), createRocks(): Generate game elements.

draw(Graphics g), gameOver(Graphics g): Rendering and game over handling.
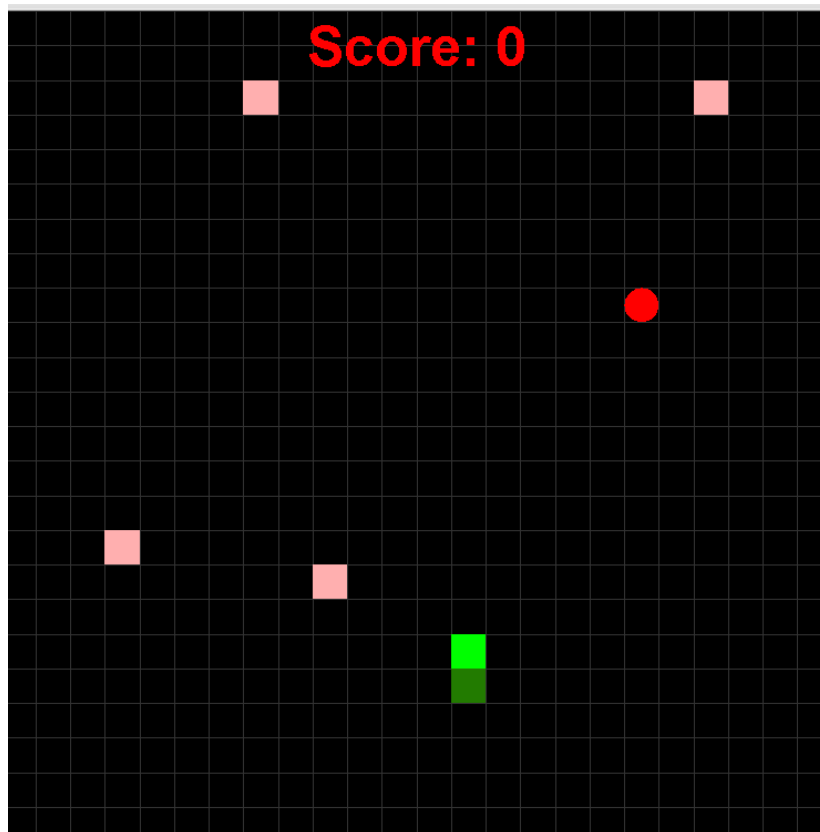
Person Class

Purpose: Represents a player with a name and score.

3. Class Diagram

4. Screenshot of the running program

5. Tests

1. **AS A player, I WANT TO check if the game starts correctly, GIVEN the game is initialized, WHEN the game starts, THEN the game should be running.**

2. **AS A player, I WANT TO ensure apples are generated within game bounds, GIVEN the game is running, WHEN a new apple is generated, THEN the apple should be within screen boundaries.**

3. **AS A player, I WANT TO verify collision with self ends the game, GIVEN the snake's head is at a body part's location, WHEN the collision is checked, THEN the game should stop running.**

4. **AS A player, I WANT TO check collision with a rock ends the game, GIVEN the snake's head is at a rock's location, WHEN the collision is checked, THEN the game should stop running.**

5. **AS A player, I WANT TO ensure the score increments after eating an apple, GIVEN the snake eats an apple, WHEN the apple is eaten, THEN the score should increment.**

6. **AS A player, I WANT TO ensure the snake's body increments after eating an apple, GIVEN the snake eats an apple, WHEN the apple is eaten, THEN the body parts should increment.**

7. **AS A developer, I WANT TO verify the database connection is set up correctly, GIVEN the database credentials, WHEN the connection is set up, THEN the connection should be established.**

8. **AS A developer, I WANT TO verify a `Person` object can be created, GIVEN a name and score, WHEN a `Person` object is instantiated, THEN it should have the correct name and score.**

9. **AS A player, I WANT TO ensure rocks are placed within game bounds, GIVEN the game is running, WHEN rocks are created, THEN the rocks should be within screen boundaries.**

10. **AS A player, I WANT TO ensure the score list is sorted correctly, GIVEN multiple players with different scores, WHEN the scores are sorted, THEN the highest score should be first.**

11. **AS A user, I WANT TO check if the menu is initialized correctly, GIVEN the game frame is set up, WHEN the game frame is created, THEN the menu bar should be initialized.**

12. **AS A player, I WANT TO check the snake moves correctly, GIVEN the snake is moving, WHEN the move method is called, THEN the snake should move in the correct direction.**

13. **AS A player, I WANT TO verify the snake moves right correctly, GIVEN the snake is directed to move right, WHEN the move method is called, THEN the snake should move to the right.**

14. **AS A player, I WANT TO verify the snake changes direction correctly, GIVEN the snake receives direction commands, WHEN the key is pressed, THEN the snake should change direction.**

15. **AS A player, I WANT TO ensure collisions with walls end the game, GIVEN the snake's head moves out of bounds, WHEN the collision is checked, THEN the game should stop running.**