# Description

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.

# **Descriptions for Test Methods**

1. **testHighScoreGetName()**: Tests if [getName()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") returns the correct name for a [HighScore](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") object.
2. **testHighScoreGetScore()**: Tests if [getScore()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") returns the correct score for a [HighScore](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") object.
3. **testHighScoreGetTime()**: Tests if [getTime()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") returns the correct time for a [HighScore](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") object.
4. **testSnakeInitialLength()**: Tests if the initial length of the snake is correct.
5. **testSnakeGrow()**: Tests if the snake grows correctly by increasing its length.
6. **testSnakeMove()**: Tests if the snake moves correctly by updating its head position.
7. **testSnakeChangeDirection()**: Tests if the snake changes direction correctly and moves accordingly.
8. **testSnakeCheckCollisionWithSelf()**: Tests if the snake correctly detects a collision with itself.
9. **testSnakeCheckCollisionWithBoundary()**: Tests if the snake correctly detects a collision with the boundary.
10. **testSnakeCheckCollisionWithFood()**: Tests if the snake correctly detects a collision with food.
11. **testSnakeCheckCollisionWithRocks()**: Tests if the snake correctly detects a collision with rocks.
12. **testGamePanelPauseGame()**: Tests if the game pauses correctly by stopping the timer.
13. **testGamePanelResumeGame()**: Tests if the game resumes correctly by starting the timer.
14. **testGamePanelRestartGame()**: Tests if the game restarts correctly with a new difficulty level.
15. **testRockGetX()**: Tests if [getX()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") returns the correct x-coordinate for a [Rock](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) object.
16. **testRockGetY()**: Tests if [getY()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") returns the correct y-coordinate for a [Rock](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) object.
17. **testRockDraw():** Tests if the [draw()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) method correctly draws the rock on the graphics context.

# Event Handlers and Connections

1. **Window Closing Event**
   * **Handler**: windowClosing(WindowEvent e)
   * **Connection**: Handles window closing in Assignment\_3 by calling [GamePanel.handleWindowClosing()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
2. **Key Press Event**
   * **Handler**: keyPressed(KeyEvent e)
   * **Connection**: Changes snake direction in [GamePanel](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") by calling [Snake.changeDirection()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
3. **Timer Event**
   * **Handler**: Timer action listener
   * **Connection**: Moves the snake, checks collisions, and repaints the game panel in [GamePanel](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
4. **Menu Selection Event**
   * **Handler**: MenuListener
   * **Connection**: Pauses and resumes the game in [GamePanel](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
5. **High Score Display Event**
   * **Handler**: Action listener for high score menu item
   * **Connection**: Displays high scores and pauses the game in [GamePanel](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
6. **Difficulty Selection Event**
   * **Handler**: selectDifficulty()
   * **Connection**: Prompts user to select difficulty in [GamePanel](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").
7. **Direction Change Event**
   * **Handler**: [changeDirection(int keyCode)](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")
   * **Connection**: Updates snake direction in [Snake](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
8. **Move Event**
   * **Handler**: [move()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)
   * **Connection**: Updates snake position in [Snake](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
9. **Collision Detection Event**
   * **Handlers**: [checkCollisionWithSelf()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [checkCollisionWithBoundary()](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [checkCollisionWithFood(Food food)](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), [checkCollisionWithRocks(ArrayList<Rock> rocks)](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")
   * **Connection**: Detects collisions in [Snake](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
10. **Drawing Event**
    * **Handler**: [draw(Graphics g)](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)
    * **Connection**: Draws rocks and food in [Rock](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [Food](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
11. **Relocation Event**
    * **Handler**: relocate()
    * **Connection**: Relocates food in [Food](vscode-file://vscode-app/c:/Users/DELL/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
12. **Score Insertion Event**
    * **Handler**: insertScore(String name, int score, int time)
    * **Connection**: Inserts score into the database in DatabaseHandler.
13. **High Score Retrieval Event**
    * **Handler**: getHighScores()
    * **Connection**: Retrieves high scores from the database in DatabaseHandler.

# Important Methods and Descriptions

1. **GamePanel.checkCollisions()**
   * **Description**: Checks for collisions with rocks, self, and boundaries. Stops the game and handles game over logic if a collision is detected.
2. **GamePanel.paintComponent(Graphics g)**
   * **Description**: Draws the snake, food, and rocks on the game panel. Also displays the elapsed time.
3. **Snake.move()**
   * **Description**: Moves the snake in the current direction. Updates the position of the snake's head and segments.
4. **Snake.changeDirection(int keyCode)**
   * **Description**: Changes the direction of the snake based on the key code. Ensures the snake does not move in the opposite direction.
5. **DatabaseHandler.insertScore(String name, int score, int time)**
   * **Description**: Inserts a new score into the database. Stores the player's name, score, and time.
6. **DatabaseHandler.getScores()**
   * **Description**: Retrieves the top 10 high scores from the database. Orders the scores by score and time.
7. **GamePanel.selectDifficulty()**
   * **Description**: Prompts the user to select a difficulty level. Returns the selected difficulty level.
8. **GamePanel.handleWindowClosing(JFrame frame)**
   * **Description**: Handles the window closing event. Prompts the user to confirm if they want to exit the game and handles the game state accordingly.