## **Practical Software Engineering I.**

### **Exercise for the 3rd assignment**

Student Name: Nadeen Hussein

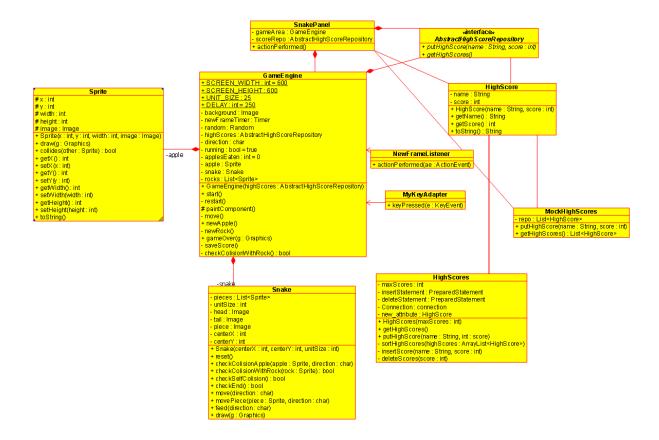
**Neptun Code: ZZKUIL** 

#### Task:

#### 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 grows 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 message box, 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 high score table of the players for the 10 best scores. Also, create a menu item which restarts the game.



#### **Methods:**

- 1. GameEngine: initializes the frame and the game
- 2. paintComponent: Draws the game gui with the screen size, Draws the rock, Draws the apple
- 3. refreshPanel: refreshes the panel when a player wins or no one wins
- keyPressed: Moves the snake around the screen according to the keyboard buttons
- 5. checkColisionWithRock: Checks if the snake hit a rock
- actionPerformed Makes sure that while the game is going on the snake didn't hit any rock
- 7. checkEnd: Checks if snake collides with the boarder of the screen
- 8. feed: If snake's head collided with an apple, adds one more unit to the snake's body and moves the head in front

# Testing plan

- 1. Snake hits a boarder
- 2. Snake eats an apple
- 3. Snake hits itself
- 4. Snake hits a rock
- 5. Snake moves in all directions
- 6. Snake high score is saved