

Assignment3 – Task 2 Snake

Tezuka Kei – HS6REY

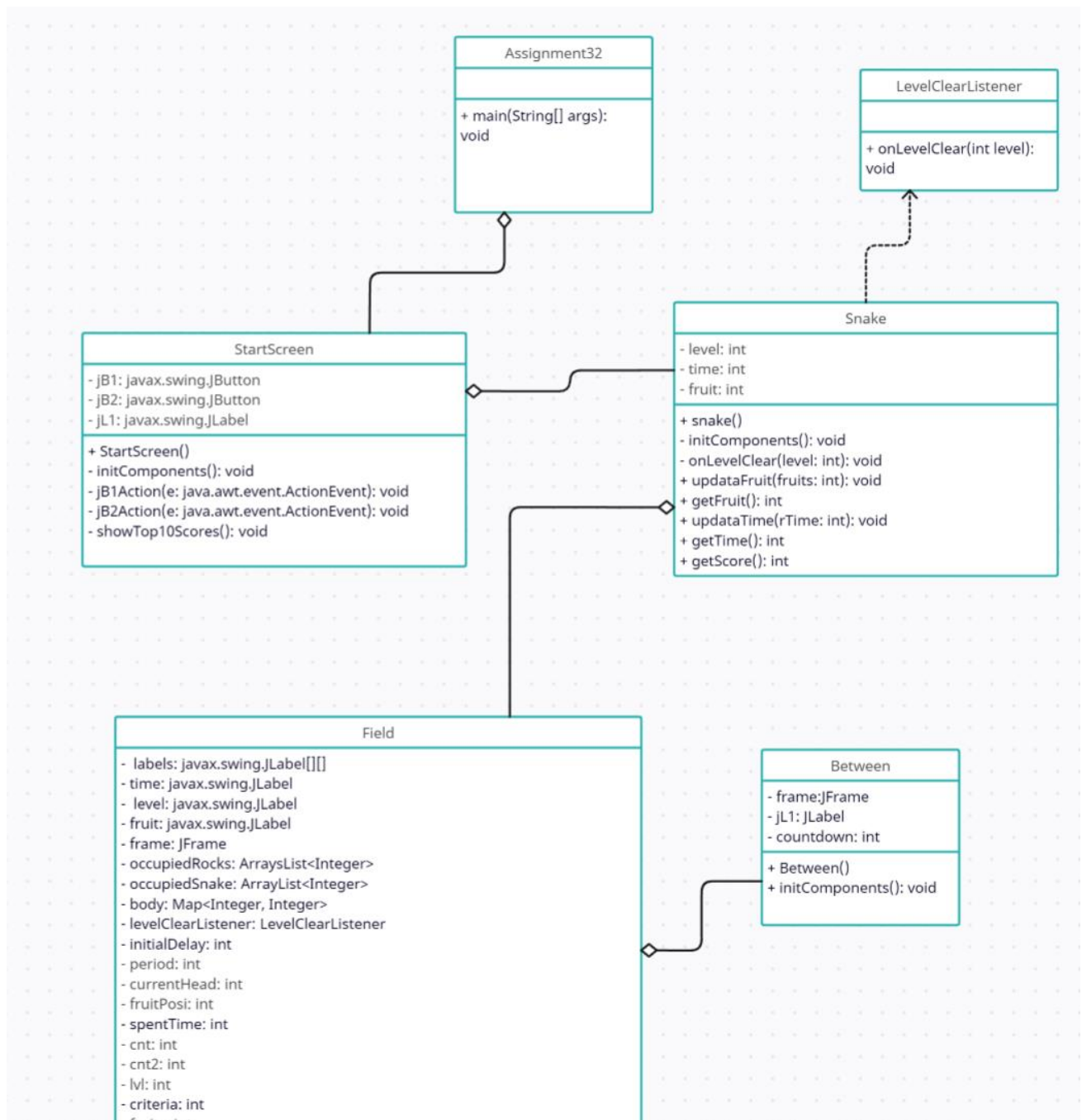
Task Description

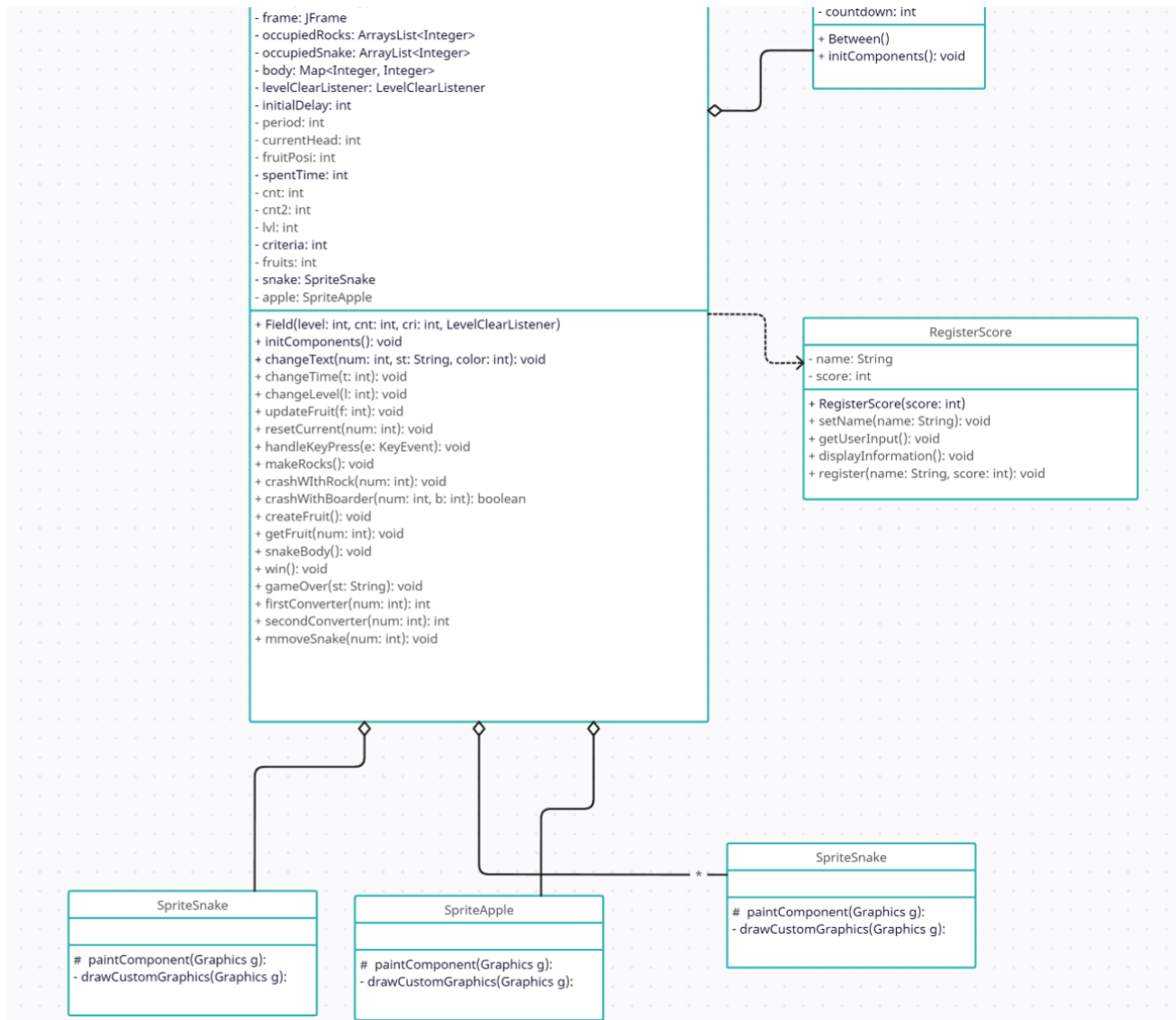
We have a rattlesnake in a desert, and our snake is initially two units long (head and rattler). We have to collect with our snake the foods on the level, that appears randomly. Only one food piece is placed randomly at a time on the level (on a field, where there is no snake). The snake starts off from 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.

Analysis

First of all, I have to create a startscreen and a user can choose to start the game or see the top 10 players. If a user choose to see the top 10 players, he have to push second button and the data will be shown from mysql database. If a use choose to start the game, the game will be started. In the code, snake class will be called, and level, how many time a user didn't use and score will be recorded in this class, and this class is necessary to call field class. In field class, mostly the game is created by JFrame and JLabel. Other Components are created by SpriteSnake, SpriteApple and SpriteRock class.

The structure of the program





A chapter for the implementation

It's going to be quite long description so I only explain main part of the program.

`changeText(int num, String st, int color):`

Changes the text of the JLabel at the specified coordinates (num) and sets the text color to the specified color (color).

`changeTime(int t):`

Updates the text of the JLabel representing the remaining game time to the specified time (t).

`changeLevel(int l):`

Updates the text of the JLabel representing the game level to the specified level (l).

updateFruit(int f):

Updates the text of the JLabel representing the number of fruits obtained by the player.

resetCurrent(int num):

Modifies the current position (currentHead) in the game by the specified value (num).

handleKeyPress(KeyEvent e):

Responds to keyboard key presses, moving the player's snake or updating the game state accordingly.

makeRocks():

Places rocks on the game field.

crashWithRock(int num):

Checks if the snake collides with a rock, triggering a game over if a collision is detected.

crashWithBoarder(int num, int b):

Checks if the snake collides with the boundaries of the game field, triggering a game over if a collision is detected.

createFruit():

Generates a new fruit and sets its position randomly on the game field.

getFruit(int num):

Checks if the snake reaches a fruit, collects the fruit, and updates the score and snake length accordingly.

snakeBody():

Updates the display of the snake's body, removing unnecessary parts.

win():

Checks if the victory condition is met, displays a victory message, and executes level-clear processing.

gameOver(String st):

Displays a game over message, shows the score, and exits the game upon game over.

firstConverter(int num):

Converts the given numerical value to horizontal coordinates.

secondConverter(int num):

Converts the given numerical value to vertical coordinates.

The connections between the events and their handlers

I used onLevelClear method for dividing levels, and I will explain about it briefly.

onLevelClear Method:

The onLevelClear method generates a new Field instance for the next level upon reaching a level clear.

Depending on the argument level, different parameters are used to create the Field instance.

For each condition, a new Field instance is created, starting a new game.

For levels 1 to 9, the number of fruits decreases from 5 to 20, and the time limit decreases from 60 to 20.

For level 10, the number of fruits is set to 3, and the time limit is set to 40.

The connection between the events and their handlers

Handles keyboard key-pressed events.

Retrieves the type of the pressed key from the KeyEvent object and performs corresponding actions.

Each key is assigned the following actions:

W Key: Move upward

A Key: Move leftward

S Key: Move downward

D Key: Move rightward

Checks conditions such as collision with borders (field boundaries) and rocks, and executes appropriate actions if the conditions are met.

If the movement is successful, the snake's position is updated, and actions like fruit acquisition, snake growth, and checking for victory conditions are performed.

The timer is also activated simultaneously, and game-over or transition to the next step occurs based on the elapsed time or specific conditions.