Third Assignment

# By: Purevnyamaa Demberelnyambuu NEPTUN CODE: BDAZOV

## Task 3 Description:

Yogi Bear wants to collect all the picnic baskets in the forest of the Yellowstone National

Park. This park contains mountains and trees, that are obstacles for Yogi. Besided the

obstacles, there are rangers, who make it harder for Yogi to collect the baskets. Rangers

can move only horizontally or vertically in the park. If a ranger gets too close (one unit

distance) to Yogi, then Yogi loses one life. (It is up to you to define the unit, but it should

be at least that wide, as the sprite of Yogi.) If Yogi still has at least one life from the

original three, then he spawns at the entrance of the park.

During the adventures of Yogi, the game counts the number of picnic baskets, that Yogi

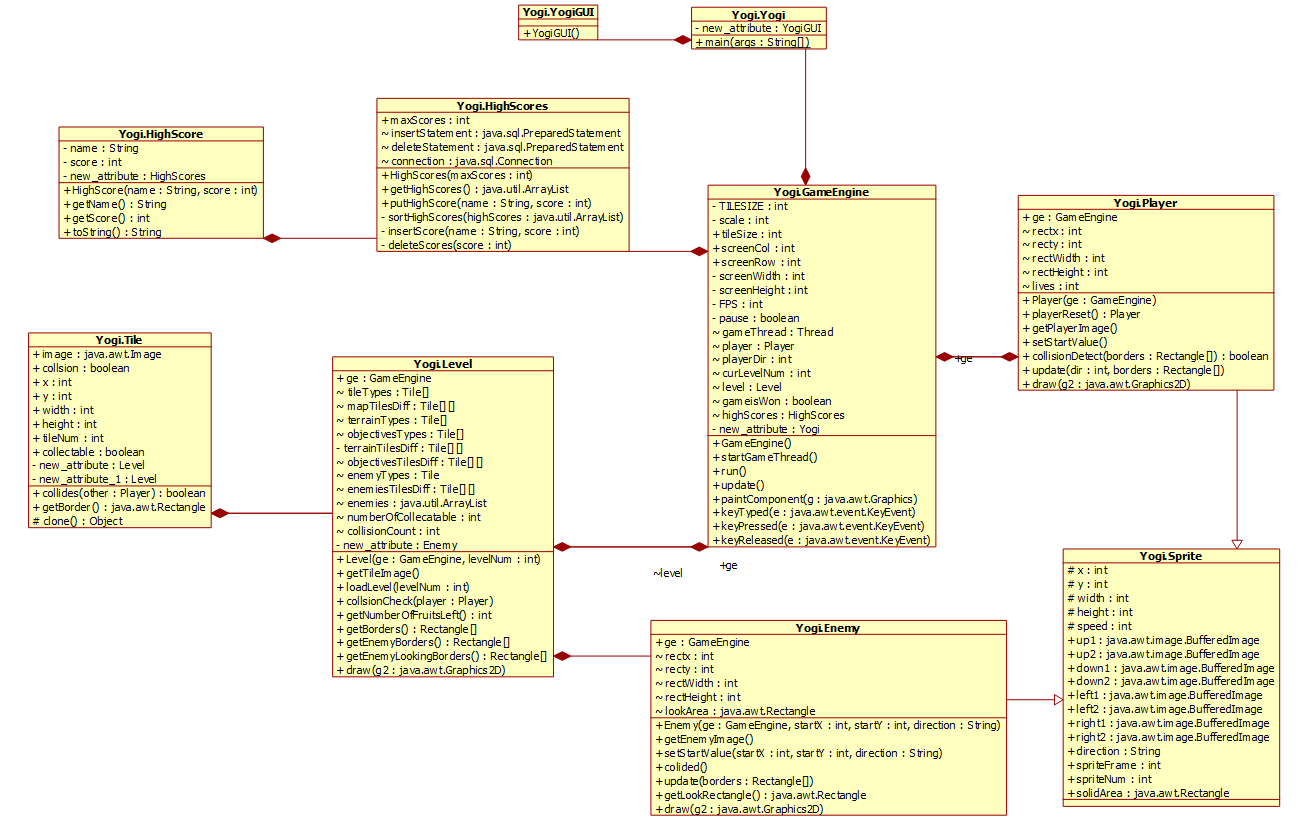
collected. If all the baskets are collected, then load a new game level, or generate one. If

Yogi loses all his lives, then show a popup messagebox, where the player can type his

name and save it 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.

## UML DIAGRAM:



## METHODS:

1. GameEngine() Constructor:

- Initializes the game engine, setting up screen dimensions and background.

- Creates a Player object, initializes the first level, and sets up the HighScores database.

2. startGameThread():

- Initiates a new thread for the game loop, allowing the game to run asynchronously.

3. run():

- The main game loop.

- Manages the timing of updates and repaints to achieve a consistent frame rate (FPS).

- Calls the update() method to handle game logic and collision detection.

- Repaints the game screen.

4. update():

- Checks if the player has run out of lives. If true, prompts the user to input their name for the high scores and resets the game.

- Checks if all fruits in the current level are collected. If true, displays a message, updates the level or finishes the game if all levels are completed.

- Checks for collisions between the player and the environment.

- Updates the player's position and resets if a collision with an enemy occurs.

5. paintComponent(Graphics g):

- Overrides the JPanel method to paint the game components.

- Draws the current level, the number of fruits left, and the player on the screen.

6. keyPressed(KeyEvent e) and keyReleased(KeyEvent e):

- Handles keyboard input for player movement and game pause.

- Adjusts the player's direction based on arrow key input.

- Pauses the game if the Escape key is pressed.

These functions collectively form the core of the Yogi game engine, managing game states, player input, and interactions with the environment. The code follows an object-oriented structure, separating concerns into different classes for clarity and maintainability.

### TESTING:

Checking if the player is bounded by the area of the board:

A screenshot of a video game

Description automatically generated

Checking if player is bounded by the area of the terrain:

A screenshot of a video game

Description automatically generated

Checking if the player gets caught by the enemies:

A screenshot of a video game

Description automatically generated

Checking if the character can collect fruits:

A screenshot of a video game

Description automatically generated

Checks if the player can advance to the next level:

A screenshot of a computer screen

Description automatically generated

A screenshot of a video game

Description automatically generated