**Programming Technology**

**Assignment-3**

**Documentation for Task-1**

**YOGI BEAR**

Written and implemented by

Zayar Htet (DRCVG2)

**Task 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.

**Solution**

**UML Diagram**

**Diagram, schematic

Description automatically generated**

**Classes and Methods**

**MainWindow**

This class extends JFrame and the parent class of the windows. This will initiate the window dimension and on click exit button feature. The MainWindow has 2 parts which are Game Statistics Label part and Game Board part. Both parts are represented with JPanel and stored as member fields.

The constructor will initialize the all the necessary field and define the KeyListener and Timer.

1. **prompt() : void** – PRIVATE

This will stop all the timer and prompt the message with JOptionPane and JDialog Text Editor.

1. **deadAction() : void**- PRIVATE

Will check if the player is still alive and if not, it will end the game and ask the player name.

1. **askForNameAndStore() : void**- PRIVATE

Will Prompt the JDialog with text editor.

1. **createGameLevelMenuItems(p: Point) : void** - PRIVATE

Create the menu item for the MainWindow.

1. **refreshGameStatLabel() : String** – PUBLIC ABTRACT

Will refresh the game statistics label.

**Board**

This class extends the JPanel and will create a game board. Parsing the model and transforming into the User Interface will be done in this class by overriding paintComponent() method of the JPanel.

**Game**

This class will read the file and store all the data in the list and will iterate the stored levels once the user has finished. All the features of the game are inside the GameLevel.java.

1. **readMaps() : void - PRIVATE**

will parse the text file into the proper model.

1. **Proper getters and setters**

**GameLevel**

This class performs the main functionality of the game. All the incoming data are parsed and stored properly in the matrix and array respectively. Rangers are stored in the array and player is stored as a member variable.

1. **moveRanger(Game gl) : void** – PUBLIC

This will move the ranger according to horizontally or vertically by 1 unit. If the roaming ranger collided with the player, then the lives of the player will be reduced, which is inside the Game gl.

1. **isCollided(Game gl) : boolean** – PRIVATE

Will iterate the stored ranger and check if the player (x,y) values is equal to the position of the each ranger.

1. **movePlayer(Direction d) : Cell** – PUBLIC

This will move the player to the proper direction if the direction is valid.

**CellItem** – Enum of value to represent the each cell.

**Direction** – Enum of value to represent the movement of the player and ranger.

**Events and Event Handler**

There are 3 event handler, which are **KeyPressed**, **Timer** and **ActionEvent** for menuitems. If the user clicked A,W,S,D or UP, DOWN, LEFT, RIGHT, then the position of the player in the model will be changed and the whole UI will be rendered again. Timer is for elapsed time and ranger movement. The ranger will be moved according to the model and the delay is 600 ms. The statistics label will be rendered during the delay for elapsed time is 1000 ms, ActionEvent will perform the on click event like showing scoreboard.

After every movement of player and rangers, the program will check if they are collided each other or not and perform necessary action like, reducing the lives. After that, the user interface will be rendered again.