Dheeraj Gururani
GZMI8P
Assignment 3

TASK – LABYRINTH GAME

Create the Labyrinth game, where objective of the player is to escape from this labyrinth. The player starts at the bottom left corner of the labyrinth. He has to get to the top right corner of the labyrinth as fast he can, avoiding a meeting with the evil dragon. The player can move only in four directions: left, right, up or down.

There are several escape paths in all labyrinths. The dragon starts off from a randomly chosen position, and moves randomly in the labyrinth so that it choose a direction and goes in that direction until it reaches a wall. Then it chooses randomly a different direction. If the dragon gets to a neighboring field of the player, then the player dies. Because it is dark in the labyrinth, the player can see only the neighboring fields at a distance of 3 units. Record the number of how many labyrinths did the player solve, and if he loses his life, then save this number together with his name into 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.

Take care that the player and the dragon cannot start off on walls.

Plan:

GameGUI:- for rendering window

GameEngine:- the backbone of the game

Darkness:- for rendering the darkness of the rest of the board

Database:- the class to configure the database to store LeaderBoard

Dragon:- storing the properties and states of the dragon

Player:- storing the properties and states of the Player

movableObject :- derived from gameobject but has further methods , parent class of player and dragon

ScoreObject :- class for every object of leaderboard

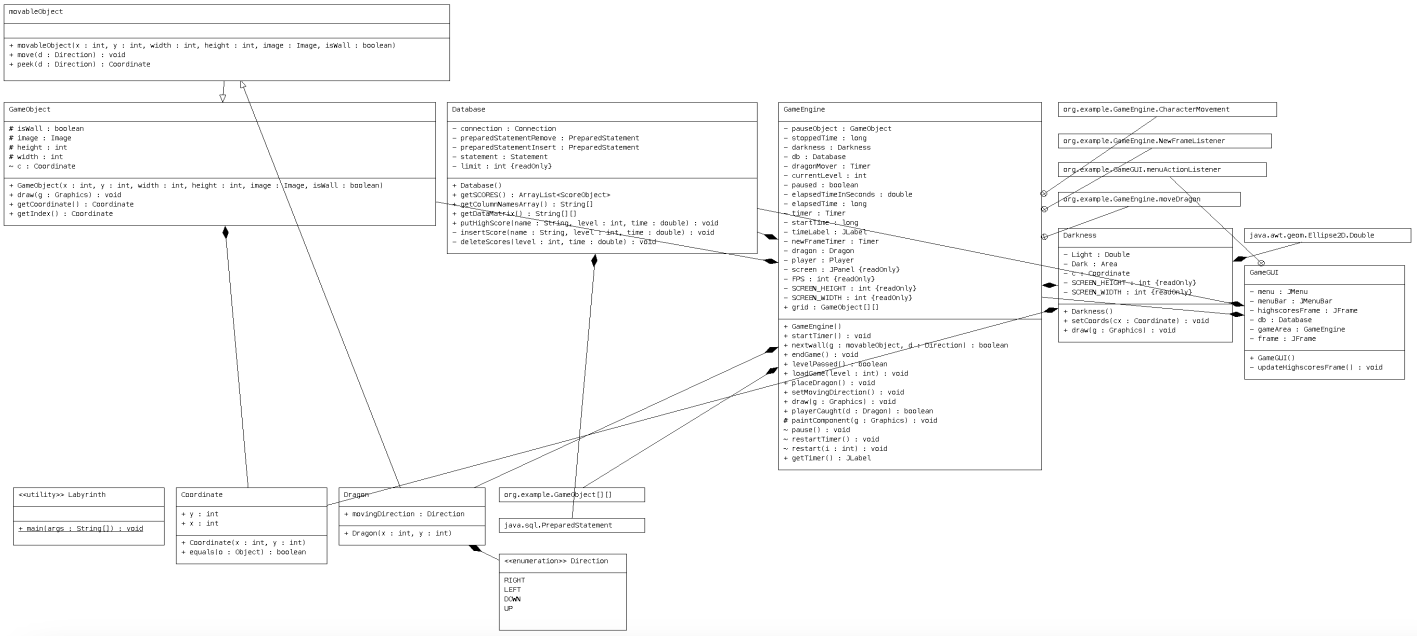
GameObject:- storing the state and property of the individual cells

Direction:- enum to handle direction

Coordinate :- class to group together x y coordinate of every object

Dheeraj Gururani
GZMI8P
Assignment 3

UML:



Dheeraj Gururani
GZMI8P
Assignment 3

Testing:

1. Checking if only 3 cells are illuminated and rest are dark
2. checking if the dragon moves when facing a wall
3. dragon moves away if it reaches the upper right cell
4. game over if player and dragon meet
5. next level starts if player passes current level
6. Player doesn't move over walls
7. Dragon never appears over the player.