**PROJECT**

***-SNAKE-***

**Short description**

A simple JAVA project that simulates the game “snake”. The user controls the movement of the snake for the purpose of eating the red “apple” to get bigger and get a higher score. The score of the user will be collected in a database and displayed on the start screen if it is in the TOP 10. The game can be played without connection to the database, but there will be no scores displayed from previous “players” and the users score will not be saved in the database. An error, such as the one below, will appear if that is the case.



**Simple state diagram**

ENTER USERNAME

PRESS SPACE

START PAGE

PLAYING THE GAME

GAME ENDED

PRESS SPACE

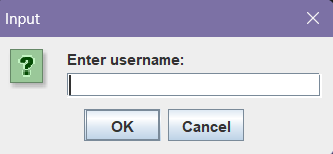
ARROW KEYS

COLLISION

\*If a collision happens the game ends.

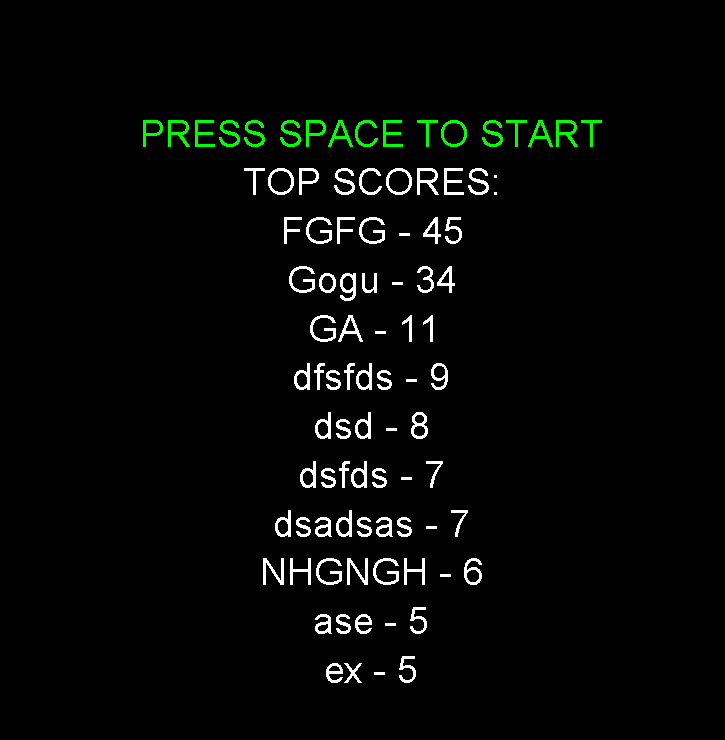
**Tutorial**

**1.Enter username.**

****

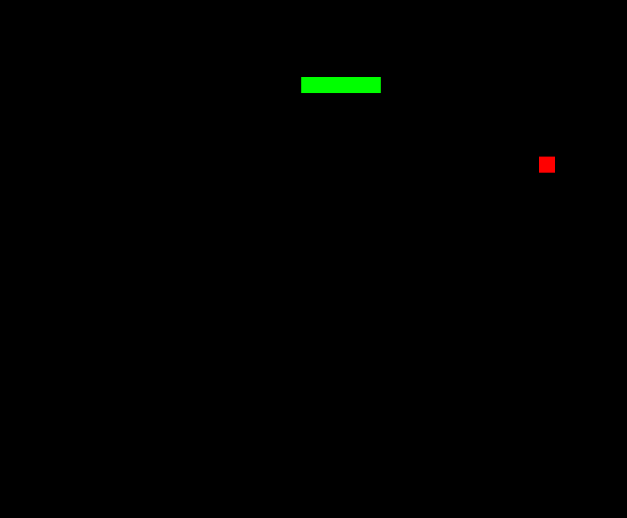
The user should write the name they wish to have in the database where their score will be stored.

**2.Starting the game.**

****

Here the user can see the TOP 10 scores stored in the database along with the button they should press to start the game.

**3.Playing the game.**

****

Using the arrow keys (↑ ↓ → ←) the user can choose the direction in which the snake moves.

HINT: Maybe he should go for the red “apple”?

**4.Finishing & trying again.**

****

After the snake hits a wall or itself the game ends, showing the current score and the high score in that session. Pressing space will take the user in the start menu so he can try again.

**JAVA CODE**

**1.MAIN.JAVA**

MAIN sets up the window and uses the GAME class to run the game.

\*DIMENSIUNE represents the size of the game window, it will be a square 600x600;

\* the window frame is named SNAKE;

\*the GAME class is used and the dimensions are added there also;

\*the next code represents the window setup;

\*game.startGame() initializes the game;

**2.GAME.JAVA**

\*the GAME class extends JPANEL for the graphical interface and implements the ACTIONLISTENER for handling game events;

\*GAMEPOINT is the representation of a point (x,y);

\*DIRECTION is an enum for each direction (UP DOWN LEFT RIGHT);

\*variables store dimensions (width,height,cell\_size), game states(game\_start,game\_over),food position (food),snake direction(dir,newDir) and snake body(snake);

\*there are also variables for the database url,name and password;

\*the constructor initializes some variables and creates a prompt for a username to be registered in the database;

\*STARTGAME resets the game state,sets focus and uses KEYPRESSED and starts the timer (for a slower snake change the timers delay to a bigger number);

\*KEYPRESSED handles inputs from the keyboard for movement,starting and restarting the game;

\*RESET resets the snake and generates new food;

\*GENERATEFOOD places a random food on the map;

\*PAINTCOMPONENT calls MESSPRINT, colors and displays the snake,food and text;

\*MESSPRINT used for formatting the text;

\*MOVEMENT moves the snake and checks for food consumption and collisions;

\*ISCOLLISION checks if the snake hits itself or the wall;

\*ACTIONPERFORMED allows movement if the game started and repaints the screen;

\*DATABASEPLAYER adds the player in the and handles database errrors;

\*UPDATESCORE updates the score and high score in the database;

\*SELECTSCORES takes the TOP 10 from the database;