**SIMON SAYS SCHOOL PROJECT**

The simon says project is an electronic game in which a circle or surface is divided into 4 equally sized areas and each area has its own individual color. These colors ring out in a random order, after which no more areas light up. You have to try to press the areas again in the correct order, if you have managed to memorize the whole sequence and enter it correctly, you have to remember an additional field each time. However, if you have clicked a field incorrectly or forgot to click, the game starts all over again and you only have to remember one field. The appeal of the game is to swing yourself as high as possible and improve your memory.

To implement this game we need several classes:

* MainFrame
* DrawPanel
* Game
* Field
* RedField
* BlueField
* GreenField
* YellowField

The 4 classes RedField, BlueField, GreenField, YellowField inherit from the class Field.

**MainFrame Class**

The MainFrame class is for everything to display graphically, add timers and bind the computer mouse, it also allows you to add buttons.

* Restart Button
* Mouse Listener

**DrawPanel Class**

The DrawPanel class allows everything to be painted during the game, so that the board is always updated, which is important when the order of the color is shown. It also colors the background and other details.

**Field Class**

The field class is the class from which the other field classes inherit, it contains the following variables:

* category
* color
* text
* x1
* x2
* y1
* y2

The class also has a few methods:

* Field(int x1, int x2, int y1, int y2)
* getValue()
* isArea(int x, int y)

The isArea(int x, int y) method is supposed to check if the mus clicked in that particular field.

**Other Field Classes**

Each field has a different range, with inheritance, each field gets the same methods, but has a different area, color, and text.

**Game Class**

The game class is responsible for the logic of the game, it generates the random order in which the colors are played, checks if the one field is correct and restarts the game if the player loses.

* alFields (ArrayList)
* score
* loses

the methods needed:

* restartGame()
* generateOrder()
* removeFieldFromList()
* checkIfCorrect(int index)
* gameOver()