Programming Technology

Name: Murad Huseynov

Neptun ID: RX15MW

Second Assignment: TASK 10.

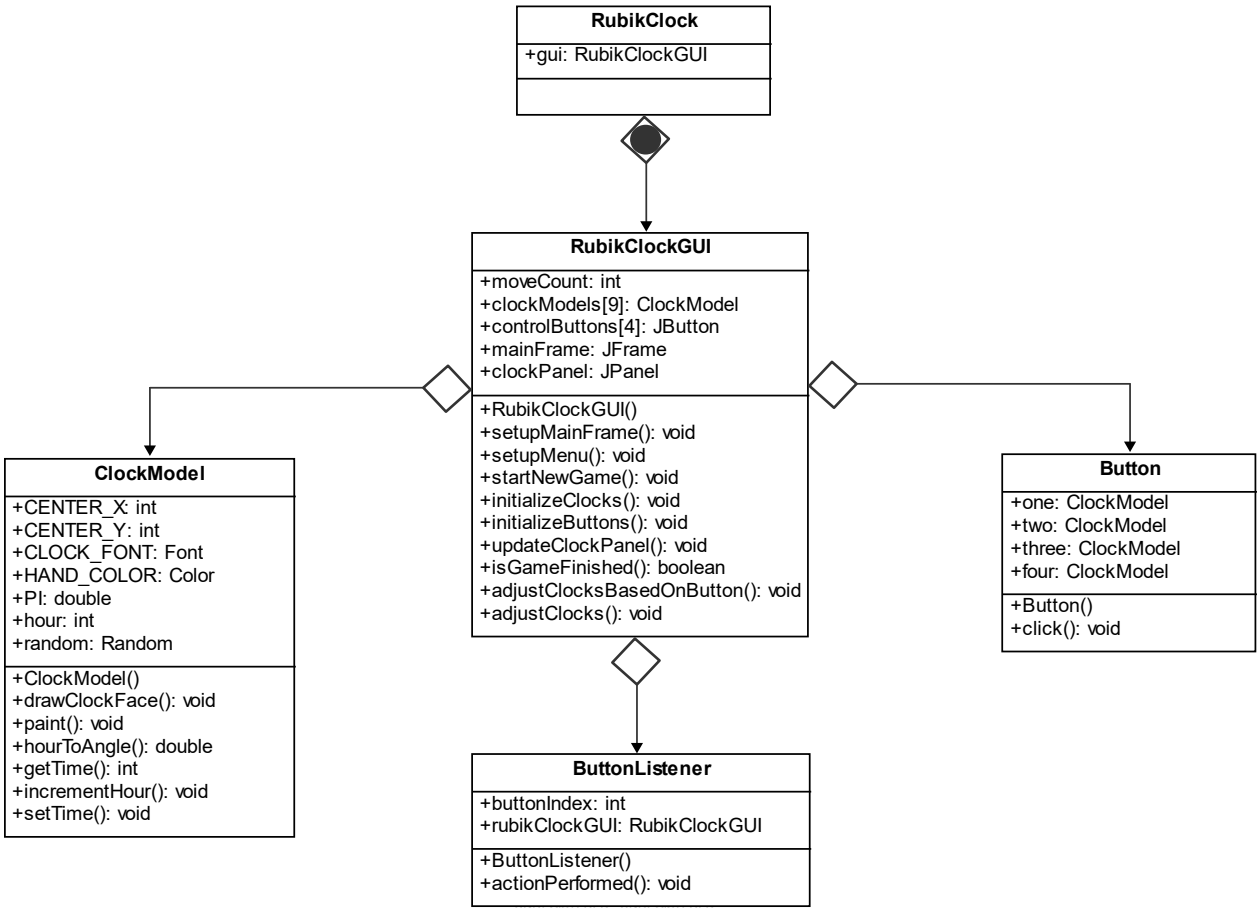
Exercise Description

Rubic clock

Create a game, which implements the Rubik clock. In this game there are 9 clocks. Each clock can show a time between 1 and 12 (hour only). Clocks are placed in a 3x3 grid, and initially they set randomly. Each four clocks on a corner has a button placed between them, so we have four buttons in total. Pressing a button increase the hour on the four adjacent clocks by one. The player wins, if all the clocks show 12.

Implement the game, and let the player restart it. The game should recognize if it is ended, and it has to show in a message box how much steps did it take to solve the game. After this, a new game should be started automatically.

Class Diagram



Method Description

1. Button.java
   a) Button(): contructor for the Button class.
   b) click(): calls the 4 clocks' incrementHours() method.

2. ClockModel.java
   a) ClockModel(): contructor for the ClockModel class, initializing a clock with random hour.
   b) drawClockFace(): draws graphical interface of the clock.
   c) paint(): add randomly generated hour on the graphical interface of the clock.
   d) hourToAngle(): converts hour to an angle.
   e) getTime(): return hour.
   f) incrementHour(): increments hour.
   g) setTime(): setter for the attribute.

3. ButtonListener.java
   a) ButtonListener(): contructor for the ButtonListener class.
   b) actionPerformed(): performs action when a button is pressed.

4. RubikClockGUI.java
   a) RubikClockGUI(): contructor for the class.
   b) setupMainFrame(): sets up a frame and does some actions when exitting the game.
   c) setupMenu(): sets up the menu bar and add the options to exit and create a new game.
   d) startNewGame(): initializes a game for start.
   e) initializeClocks(): initializes 9 clocks.
   f) initializeButtons(): initializes 4 buttons.
   g) updateClockPanel(): handles how the buttons and clocks are placed on the frame.
   h) isGameFinished(): sets the winning rule.
   i) adjustClocksBasedOnButton(): handles how buttons adjust specific clocks.
   j) adjustClocks(): handles what happens when the game is finished.

5. RubikClock.java
   a) main(): Main method which initializes an instance of the game and starts it.