Rubikan Requirements

by

Christerpher Hunter

Nova Southeaster University

July 26, 2021

Requirements

System Description

The purpose of this software project is to be easy to use software application that allows users to solve a Rubik's Cube on a computer. The application is named Rubikan. A username will be required in order to track statistics. This application will be accessed via a downloadable application. The downloadable application method will be designed and produced as an application that will then be published online for download. Rubikan will display a visually appealing GUI with color. The Rubik's Cube presented will be the classic Rubik's Cube patented in 1975 and made of 26 individual cubes. Although there are others, there are presently no plans to integrate shapes other than a cube further. This application will reinforce intuitive notions of 3-D shapes. Inherently, manipulating a rotating cube with rotating rows and columns will be akin to a toy as it is a game and will entertain the user. The complexity of the Rubik's Cube puzzle will be a brainteaser.

Rubikan will be targeting ages five and up. Testing new algorithms and storing them for later retrieval will be featured. The GUI will present the cube in color and accurately depict an actual cube. The difference between a Rubik's cube in hand and the one in the application will be naught. Algorithmic theories can be practiced in the software then applied in reality. Rubikan will be able to rotate and reset the cube in its entirety. The reset will also randomize the cube. There will also be a randomize button. There will be a time attack feature that will countdown from a preset time down to zero. Rubikan will also store every move made by the user and every time to complete or reset each attempt to solve the cube. A scoreboard of the previous attempt will be able to be toggled on and off. The initial cube will be completed upon opening the application.

The rows and columns will spin on an axle, similar to the actual Rubik's Cube. In order to manipulate the column and rows, each column or row will be selected one at a time. If a column is selected, an option to rotate north or south will be presented. If a row is selected, the option to rotate the row east or west will be presented. Three sides of the cube will be visible at all times. A compass will be displayed on the screen for the users not familiar with cardinal directions. Lastly, a current score will be on-screen during the attempt, along with the time thus far. If in time attack mode, no score will be displayed.

Scenario

Now we will go through a usage scenario. The players include user 1, Patricia, and user 2, Jane. Patricia is using a PC with the latest Microsoft operating system, and Jane is using a widespread distribution of Linux. Patricia sits down and turns on her quad-screen gaming pc. Jane is using a laptop. Patricia has opted for the downloadable application version of Rubikan. Jane has done the same. After visiting an open-source source control repository, they are presented following list:

```
Rubikan.0.4.7.exe
Rubikan.0.4.7.sh
Rubikan.0.4.7.src
```

The latest version is available at the time of their download. There were options to download the source code directly. The list of options being presented is the intent to submit a pull request and issue testing. An option to download a shell executable intended for use in a shell environment and an .exe executable for use in a Microsoft operating system. The difference in starting the game is indistinguishable from the application version of different operating systems. Once Rubikan starts, it opens a GUI and presents the Rubik's cube, solved, in three dimensions in the center of the application window. In the upper right-hand corner, the names of previous users will be displayed. If there is no previous user, a blinking cursor will be standing by to be filled in. There will be RESET, START, and TIME ATTACK buttons in the lower right-hand corner. The TIME ATTACK button will not be highlighted unless previous timed attempts are logged—a compass, of old-fashioned nature, in the upper left-hand corner and static. Once the START button is actuated, the cube will display buttons outside of each row and column. Only one row or one column will be clickable at a time. Once one row or one column is chosen, the appropriate option to go east, west, north, or south will be displayed. A timer will also begin once the attempt to solve begins, i.e., the START button. There will be no time limit. If the cube is solved, the username and time to completion will be logged automatically. If the reset is clicked before the puzzle is solved, no time is logged, and the cube is randomized. The quit button lives, in red letters, in the lower right near the other buttons. Once the cube is solved, all the moves are also committed to that specific attempt to solve. The known algorithms to solve the cube will also be available if the compass is clicked.

Use Case

- Present a GUI Rubik's Cube
- Username required
- Teach 3-D objects
- Entertain the user
- Brainteaser
- Test computer functionality
- To be a toy
- Rotate on screen
- Present an appealing GUI
- Display colors on the cube
    1. Display white on one side of the cube
    2. Display yellow opposite of white on the cube
    3. Display blue on one side of the cube
    4. Display green on the opposite of the blue side of the cube
    5. Display orange on one side of the cube
    6. Display red on the opposite side of the orange side of the cube
    7. Arrange in clockwise pattern: red, white, blue
- Reset the cube
    - Mix up the colors via Reset Button
- Recall past times to completion
- Time attack

- Compare times to other users of the application
- Turn each column or row independently using an internal pivot mechanism
- Display a finished cube
  - each face has a solid color
- Present controls to manipulate the cube
  - select a row and choose to move east or west
  - select a column and choose to move north or south
- Display a compass on the screen
  - Store solving algorithms
- Display a score based on time to complete
- Multi-monitor support
- Will work on any popular operating system

Classes

1. class Initialize
2. class Username
3. class Main
4. class Cube
5. class Reset
6. class Restart
7. class TimeAttack
8. class Comparison

9. class Selection
10. class Rotate
11. class Models
12. class Controls
13. class Compass
14. class Scores
15. class MultiMonitor
16. class OSCheck
17. class Completed
18. class Quit
19. class Timer
20. class DataBase

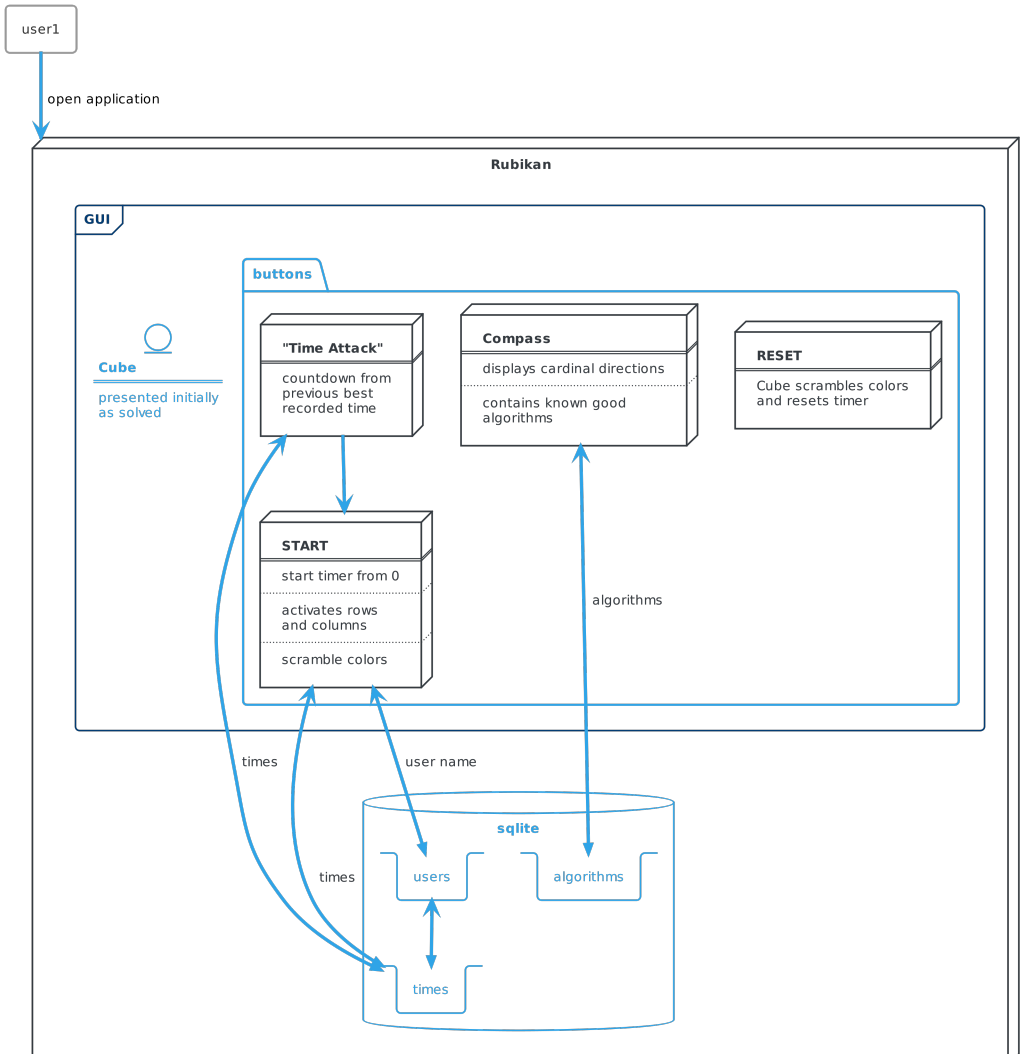
Exceptions

- Rotating the cube in the GUI will prove difficult
- Implementing rotating the cube on an axis will be time-consuming
- Presenting a 3-D item will have to be researched
- Is the application OS-specific?
- How will this application be deployed?
- Will there be a charge to use the application?
- Will this run on a phone?
- Will this save information for later retrieval?

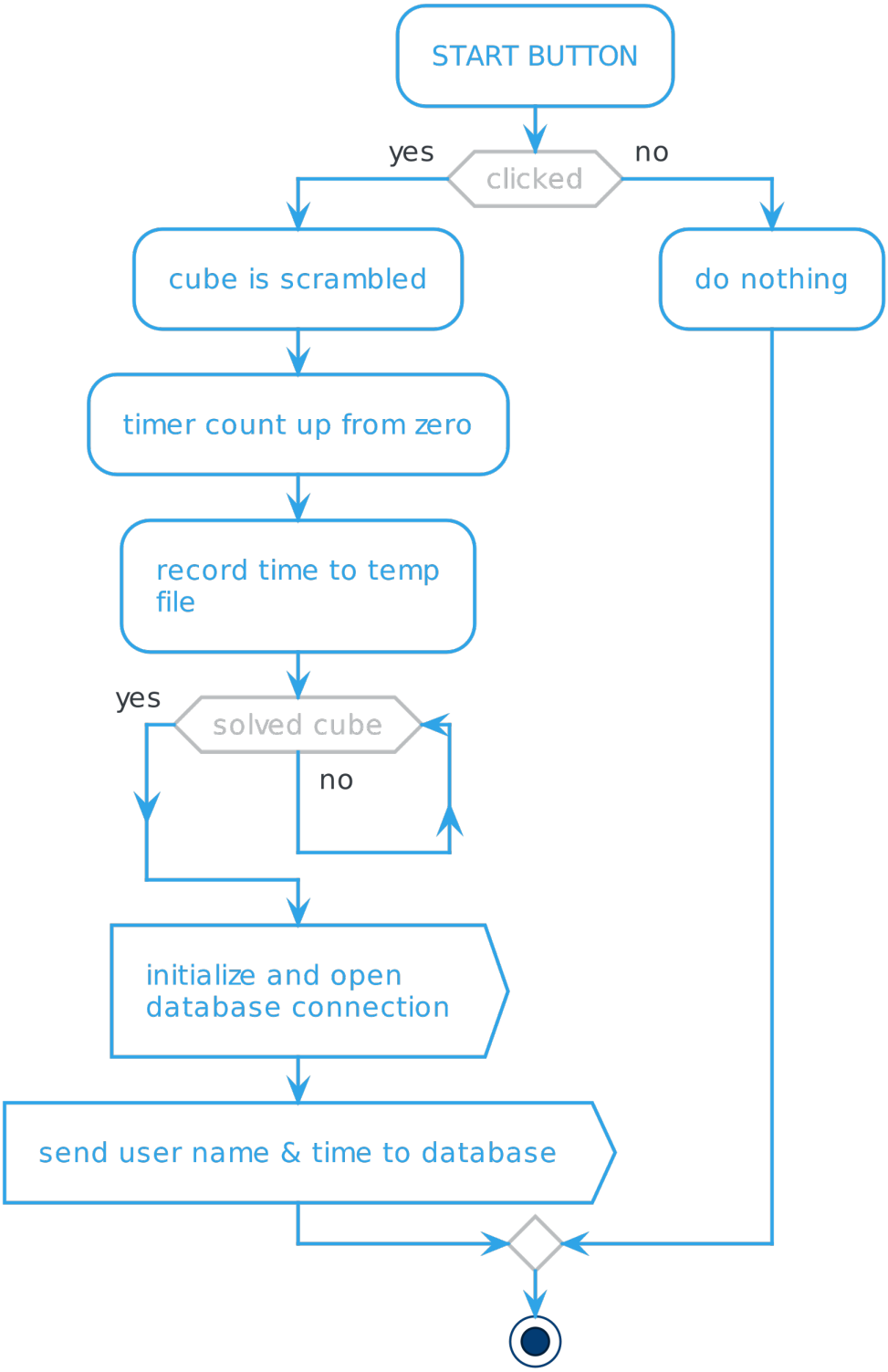
- Will the states of the application conflict?
- Time attack mode will be similar to the regular attempt mode. (Inheritance?)
- How will Rubikan entertain the user?
- What database will be employed?

Diagrams

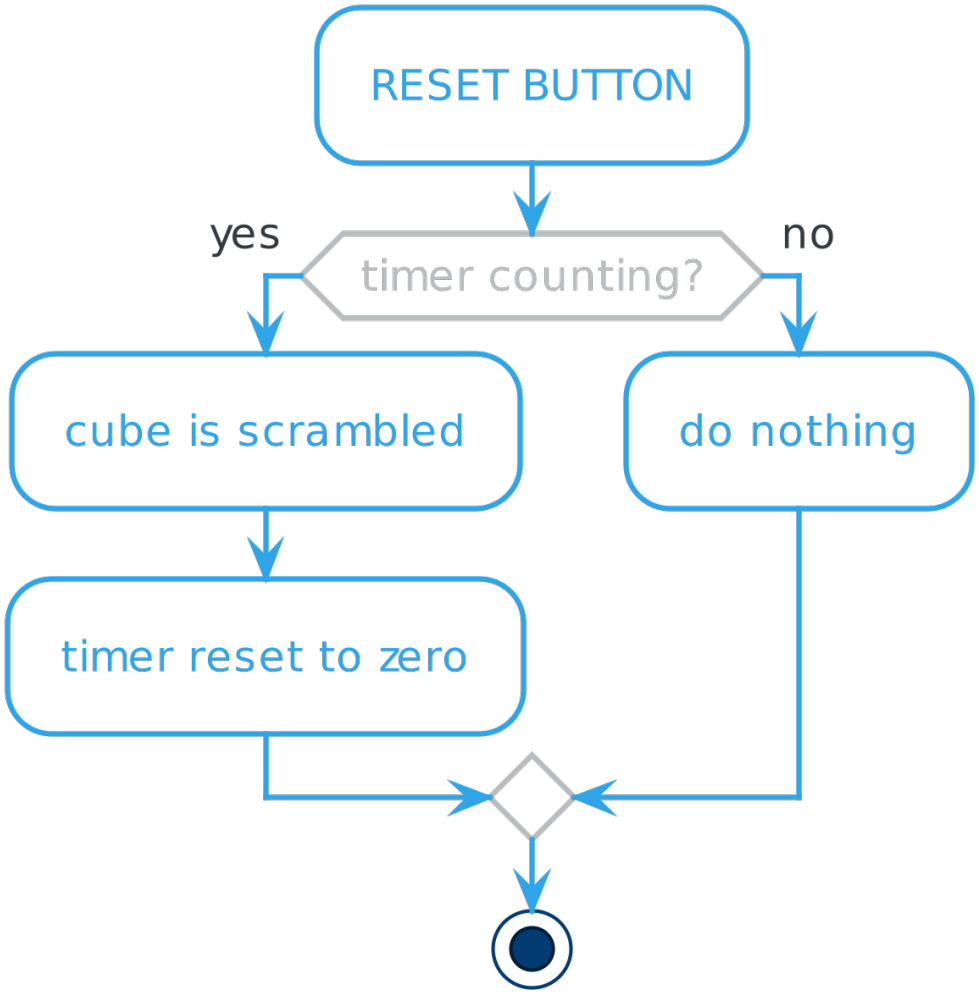
- GUI Diagram



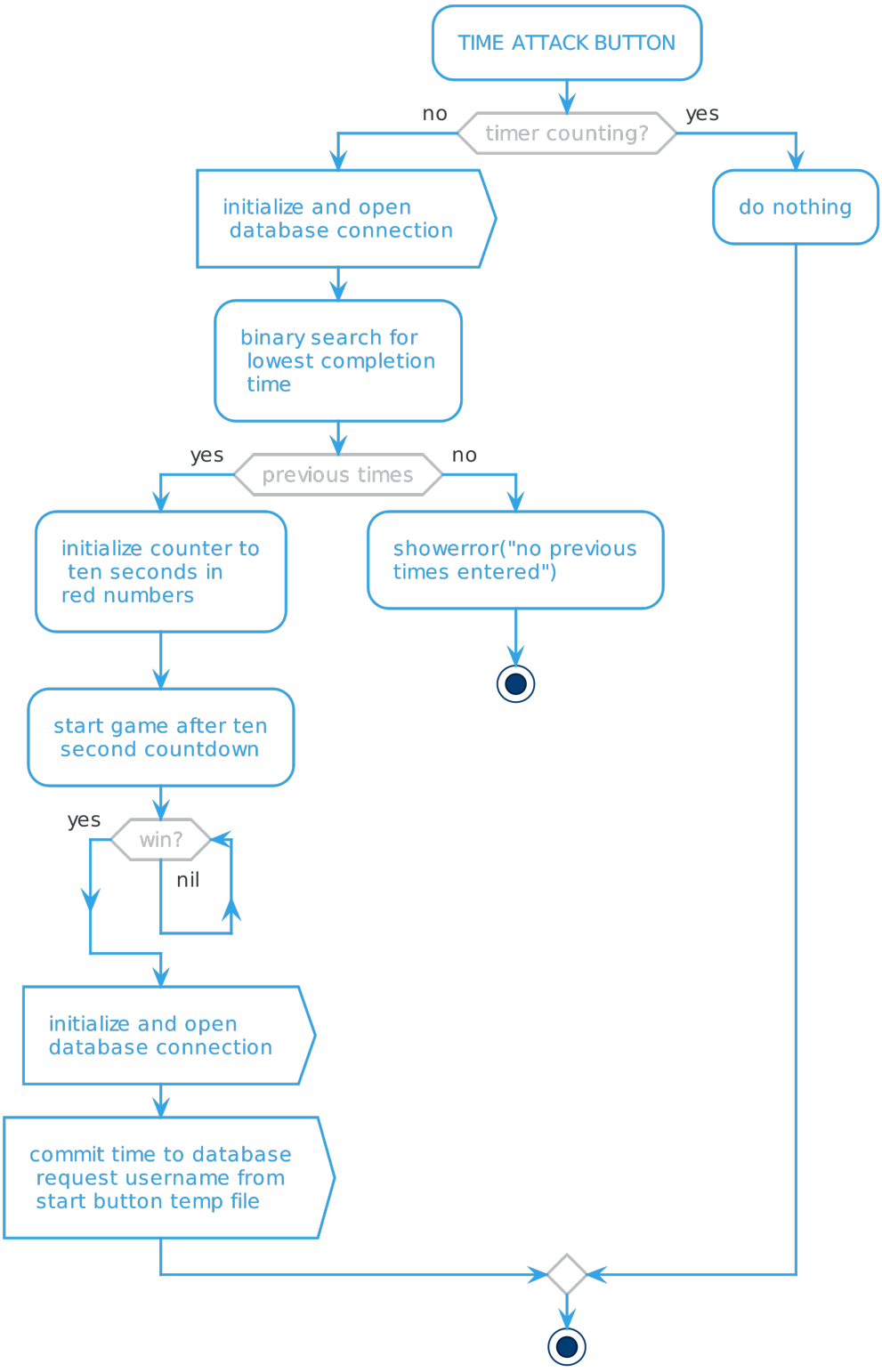
- Start Button Diagram



- Reset Button Diagram



- Time Attack Button



- Compass Button