

Lab8: Analyze/Plan/Design/Implement a Game with a GUI

Objectives

- Analyzing a problem
- Planning
- Designing a solution
- Applying MVC principles
- GUI
- Implementing the solution in a programming language

Problem Specification: Find or Die Game

Write a program for the game "Find or Die".

- This is a board game of size n by n where n is an odd number
- There will be n rows and n columns.
- There will be clickable buttons in each cell of the board.
- Each button can be clicked only once.
- First randomly put one great prize
- Randomly put mines half of the cells
- Randomly put a second chance for the rest of the cells.
- If the player selects the great prize button, game is over and the player wins.
- If the player selects one of the mine buttons, game is over and the player loses.
- If the player selects one of the second chance buttons, then the player can continue by rolling a dice(dice as singular is acceptable. According to the Oxford Dictionary (US and UK versions) the word "dice" is used for both singular and plural. It also mentions that the use of "die" is becoming increasingly uncommon.). If the dice has a value greater than 4, i.e., 5 or 6, then the user can continue the game until a mine or great price is selected or the dice value is less than or equal to 4.

Your program should

- Show the board with clickable buttons
- Get the user's choice,
- Execute the choice,
- and then continue until the user wins/loses.

In your design

- Focus on the game first without visual components.
- Analyze the problem and apply divide-and-conquer idea.
- Run your program without visual components and test for all cases.
- Then add the visual components, i.e., a frame, buttons, click events and responses.
- Clearly write your classes/constructors/setters/getters/toString methods
- Generate Javadoc
- Ask your friends to play your game and give them a chance to test your program.

SEE SAMPLE EXECUTIONS BELOW!

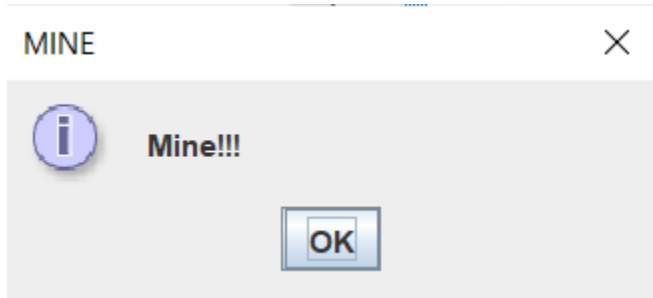
Sample execution:

Without GUI

Board has

0 - DICE 1 - DICE 2 - DICE
3 - MINE 4 - MINE 5 - AWARD
6 - MINE 7 - MINE 8 - DICE

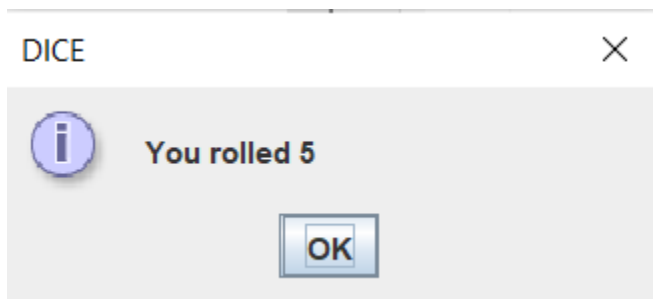
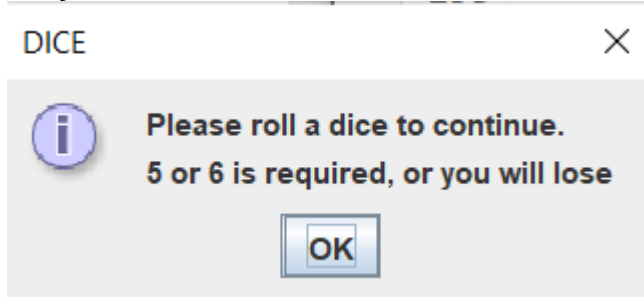
Player selected 3



Board has

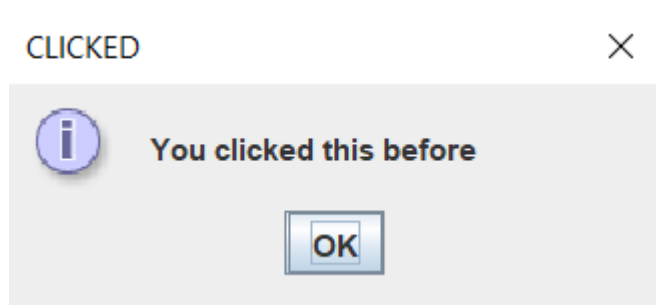
0 - MINE 1 - MINE 2 - DICE
3 - DICE 4 - MINE 5 - DICE
6 - AWARD 7 - DICE 8 - MINE

Player selected 5



Player can continue

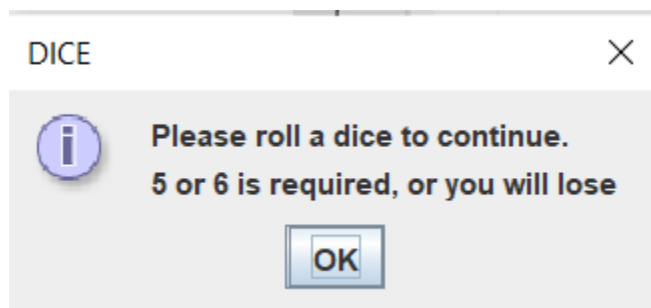
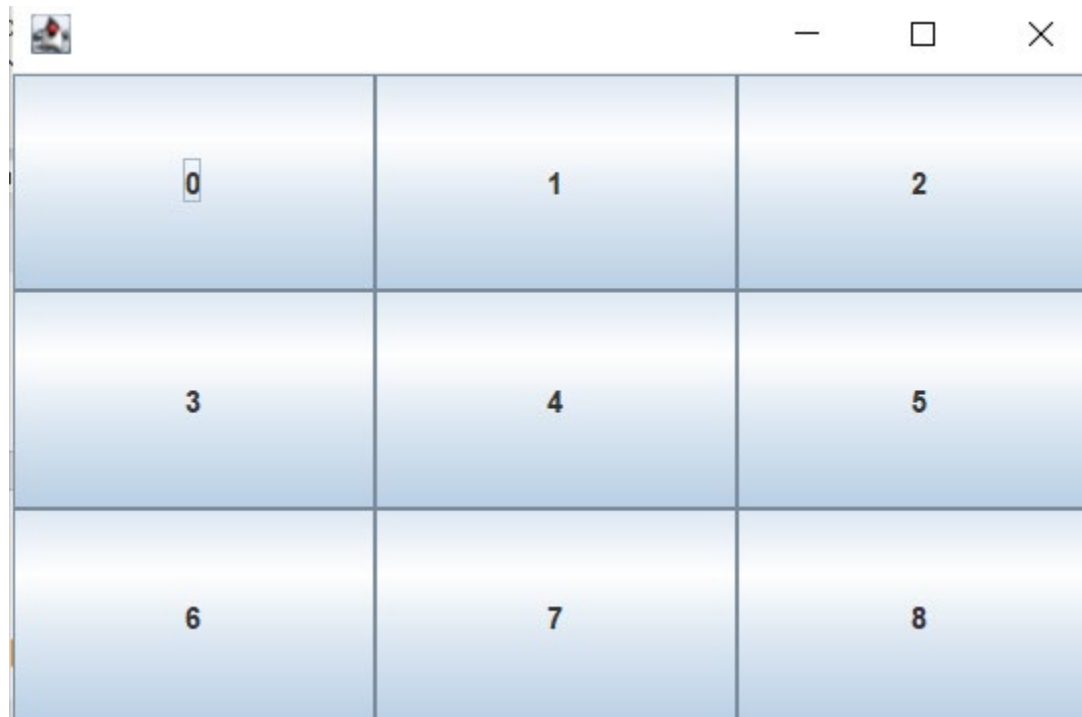
Player selected 5 again

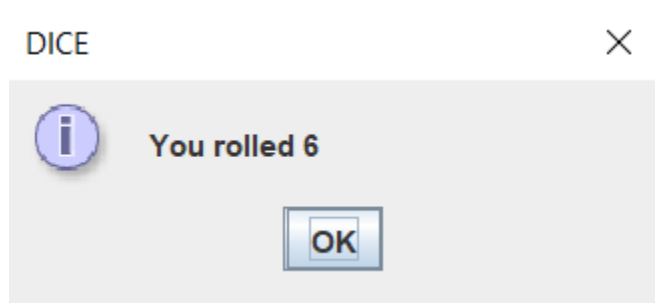


With GUI

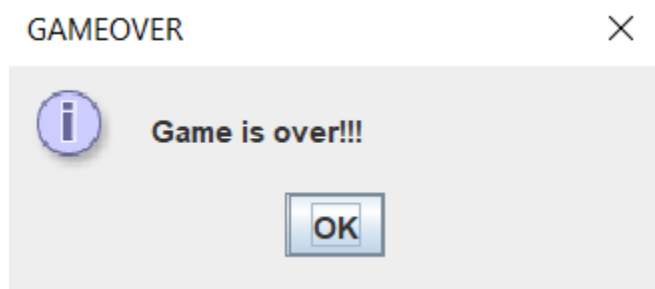
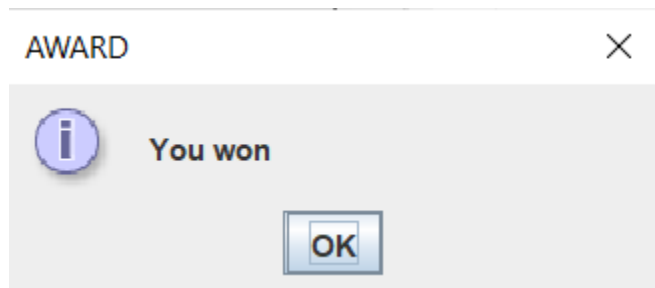
Board has

0 - DICE 1 - MINE 2 - DICE
3 - DICE 4 - MINE 5 - DICE
6 - MINE 7 - MINE 8 - AWARD
Player selected 3





Player selected 8



How to turn in this lab

Before turning in any program in this class, remember this mantra:

Just because it works does not mean it's good.

Your grade will also come from the following aspects of your code:

- Submission
- Accuracy/correctness
- Readability
- Neatness
- Presentation
- Style
- Testing
- Commenting

For all labs, turn in only an **electronic** version.
Please submit the followings after all labs:

- zip file of your project (the project folder, not just the .java file(s)): zip file name will be your **YourFirstNameLastNameLab8.zip**

- a single pdf file of all your codes (.java), screenshots of your output for each file: pdf file name will be **YourFirstNameLastNameLab8.pdf**

Submit

- **the zip and pdf files**
- after Monday Lab session until **SUNDAY 11 PM EST**
- from the Nexus Lab submission link that will be accessible in Week 2-10

Ask for help if you are having problems!