

Project Report.

Modification of 4 – Gewinnt Game in GUI JAVA

Nadeem Ahmad.

Matrikelnr. 119371.

02 /April/2019.

Abstract

Java graphics user interface is a useful tool to visualize the data in easy and efficient manner. In this project I have implemented and modified an existing game 4-gewinnt in a Java graphics user interface with the help of objects from JFrame library. At a beginning of a game it was aimed to provide a free choice to players for select a desired game field size by declaring number of columns and rows of a game field. In addition to the modification players are provided an opportunity to choose a number of stones to declaration a winner. To update all the decided features to game JFrame objects like text boxes and buttons are used. The given inputs in form of rows and columns are processed and are used to initialize a game field in form of JButtons grid. Each button of a grid is made to work in a way so that it could place a stone in lowest free position of a grid vertically according to alternate moves of players. It was expected to analyze the graphical changes in the button field on each press of a button for declaration of a winner. The functionality of buttons in a grid are assigned and executed in a way so all the principles of a game are fulfilled.

Motivation:

The main reason of deciding to work on this project was to learn the implementation of JAVA graphics user interface package known as JFrame for the implementation of a game.

Except developing visual games JAVA JFrame package is a very useful tool in JAVA to solve complex calculation problems in hand by developing an easy and user friendly interface. I am personally interested to implement the computer's ability to solve a complex and tedious calculation processes of several Engineering fields in an easy and user friendly manner.

Below are some of the references used to achieve the task:

<https://www.coding-board.de/threads/vier-gewinnt-java.37791/>

https://en.wikipedia.org/wiki/Connect_Four

<https://www.dreamincode.net/forums/topic/234693-connect-four-gui/>

<https://stackoverflow.com/questions/40062306/java-swing-how-to-build-connect-4-gui>

<https://www.upwork.com/hiring/for-clients/write-perfect-brief-mobile-app-development-project/>

Goal:

My project was to implement and modify an existing game called 4-Gewinnt in JAVA JFrame package. The main goal was to add some new features which provide players more options like selecting a desired game field size and winning stone criteria.

Background

Description

This game is played on a vertical board which has seven hollow columns and six rows. Each column has a hole in the upper part of the board, where pieces are introduced. There is a window for every square, so that pieces can be seen from both sides.

In short, it's a vertical board with 42 windows distributed in 6 rows and 7 columns.

Both players have a set of 21 thin pieces (like coins); each of them uses a different color. The board is empty at the start of the game.

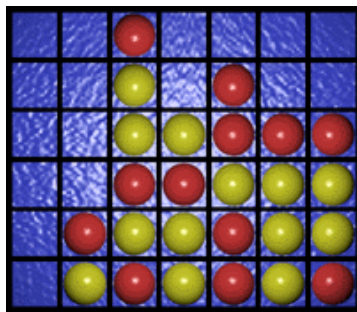


Fig 1:-Vertical Board with Red and Yellow stones.

Objective

The aim for both players is to make a straight line of four own pieces; the line can be vertical, horizontal or diagonal.

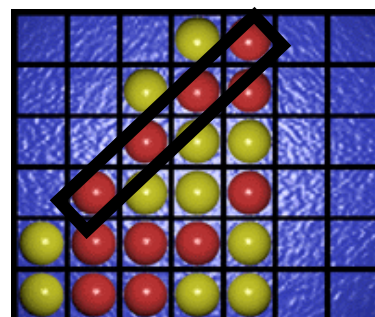
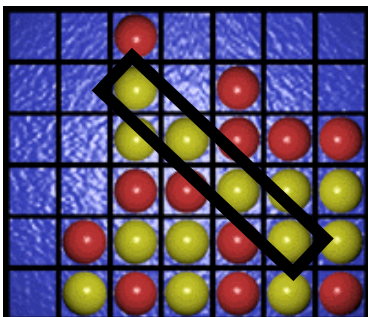


Fig 2:- Win for yellow in diagonal.

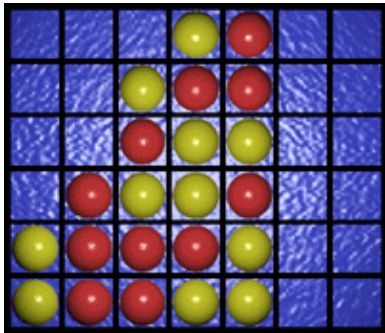


Fig 3:-Win for red in diagonal.

Fig:-draw.

How the game goes on ?

Before starting, players decide randomly which of them will be the beginner; moves are made alternatively, one by turn.

Moves entail in placing new pieces on the board; pieces slide downwards upper holes, falling down to the last row or piling up on the last piece introduced in the same column. So, in every turn the introduced piece may be placed at most on seven different squares.

The winner is the first player who gets a straight line made with four own pieces and no gaps between them.

Concept

I have followed the simple approach to solve a problem by analyzing what should be the outcome and what programming skills I have and what skills are required to solve the problem.

Firstly, listing out the input methods to be included and selecting the relevant JFrame objects to obtain a desired functionality. To achieve this target some text fields are introduced to obtain desired inputs as a string to decide game field size and number of stones to be matched for declaring a winner. In addition to these text fields an individual button is included to proceed further the choice for game field initialization.

Secondly, by focusing on how to process and utilize this given user inputs for game field initialization. Further given user inputs are taken as a length for 2D array of JButton. A specific method is executed within the class which iterates and creates a 2D array of JButton as a game field. Within this method some additional final integer datatypes are introduced to save the indices of an array uniquely for the precise locations in further stages of the development. By keeping all the appearance and utility aspects in mind almost all the desired JButton assignments are done during the iteration within the method. Like icon, action listener and other relevant visibilities.

Thirdly, now the button field is initialized and concern is to assign a functionality of each button to resemble the actual working of a real game. In real game gravity effect is one the principle of game like entry of a player should fill the lowest possible row of a particular column first. To achieve this a separate method is created within the class to make a two different graphics of oval shaped with two different color filling so that the moves of each player can be identified uniquely and are assigned to the lowest possible button of the column in field. This method is executed on pressing any button of the field and within the method indices of buttons is called for iterations to choose the lowest possible position of the column and assign a particular graphic according to the alternate turns of players.

Lastly, as the moves by each player are separated by their unique graphics on game field now the question arises who is winning the game? To get the answer of this question another method is introduced within the class to decide the winner of the game. To execute this method on game field we should ensure that state of the each button is set false at beginning during initialization and it is changed to true after a particular entry is made by each player. Also, during the assignment of graphic by each player it is also important to set unique Boolean representation like true for red and false for yellow entry. In this method simple iteration is executed to look for a consecutive match of the entry horizontally, vertically and diagonally to declare a winner of the game. This method is executed after graphic assignment to a button.

Implementation:

To achieve logical functionality of a game, simple implementation is taken into consideration. Normally begins with class creation by the name of GUI which is inherited from JFrame class. In this class, all the attributes and the JFrame objects are defined like JTextfield and JButton. The implementation of the whole logic is obtained by dividing tasks into several parts. In this project 4 simple consecutive methods are used to solve the each part of the task.

- 1) To initialize the graphic user interface objects a method namely `initUi()` is implemented. JAVA graphic user interface objects are initialized and are used to take the inputs from a player in the form of string to initialize a game field. The input String data is converted to a primitive data type integer which is further used as a length of 2D JButton array, 2D Boolean array and 2D integer array. In total 3 text boxes are used in which String data from text field 1 and 2 are used as row and column length of 2D arrays. The remaining string data from text field 3 is used as the length for consecutive match to declare a winner. In addition to the text fields a JButton namely calculate is implemented to execute the sequential method to initialize a game field of JButtons.
- 2) To achieve a game field according to a given data a method namely `initField()` is implemented. This method considers an input string from text boxes as a length to initialize the 2D arrays for the achievement of required tasks. In this method, a JButton 2D array is initialized by iterated to its length and indices are saved as final integers for further operations. It is considered to assign the state and color uniquely by initialization

of 2D arrays namely state and color within the loop. All other JButton utilities are like button appearance and button functionalities are stated within this method. The functionality of each button is achieved by declaration of further methods within the action listener commands of an object.

- 3) Furthermore, an important principle of the game is gravity of stones in a field on alternative moves of players. To occupying the lowest vertical position of column button in a field on pressing a method called addStone() is assigned to a task of button. This function is formally obtained by passing the saved indices of any button in a game field in method and performs iteration on column index to achieve lowest possible position for a stone. This method differentiates the assignment and alternative turns of players. To analyze and declare the winner on every press a method called getWinner() is declared sequentially within the addStone() method.
- 4) To achieve a result of a game method namely getWinner() is created. In this method state and color arrays are checked for the consecutive matching conditions horizontally, vertically and diagonally to declare the a winner of the game and restart the game. To check true conditions for buttons horizontally, vertically and diagonally an algorithm on array indices is implemented iteratively until the matching conditions are found.

The challenges faced during implementations were few like to achieve the column and row arrangement of the buttons in a way to get the lowest possible position of a column. To achieve gravity function of a game took me longer than other implementation because I was not much familiar with the JFrame objects to get attributes of a particular button in a field.

It was quite challenging to use objects of the JFrame library to get the graphical attributes of the JButton for analyzing and checking the true statements in order to declare a game winner.

Conclusion

This project was a challenging task and a good start to understand the scope of JAVA graphics user interface implementation. It is concluded many more modifications are possible for this project in future to make it more flexible and efficient. As I know being a beginner in JAVA and managed to solve almost all the tasks but I believe it could be done in more effective and efficient way. So in future, if i will get an opportunity to modify and extend this project, i would focus on learning the implementation of objects from the JFrame library to make code short, flexible and easy.

