Data Structures with JAVA laboratory

A mini project report
on

# PICTURE PUZZLE GAME

Submitted by

1PI14CS069 ,PRADEEP G HEGDE

1PI14CS013, AKSHAY N GOGERI

1PI14CS007, ABHISHEK V KATWA

Department of Computer Science & Engineering

**PES INSTITUTE OF TECHNOLOGY**

*(An Autonomous Institute under VTU Belgaum)*

100 Feet Road, BSK III Stage, Hosakerehalli, Bengaluru - 560 085

## INDEX

# Project Description

We are doing project on PICTURE PUZZLE GAME , which is player friendly and entertaining for them.It is basically a shuffled pieces of a picture that we are going to generate and then ask the player to solve it by moving the pieces of the puzzle to put them in order.

We help the player by giving him the rules of our game with self desprictive .gif images.We have two levels of difficulty one with 3*3pieces of  Picture puzzle(easier) and other with 4*4pieces  of picture puzzle.

Rules of our puzzle game is

*Only the neighbours of the piece(slider) that has no image can be clicked.

*On clicking the neighbours of the slider,images get swapped.

*Keep on moving the pieces of puzzle to put them in order.

We also provide the player solve button to know the actual Picture.

In this project we are providing many functions to store and manipulate the whole puzzle during player is solving it,

* Storing the actual Picture.

* for storing the changes that has happened to picture after every swap.

* for shuffling the actual picture ,so that for that a solution exist.

* for storing the clickable neighbouring pieces of  slider after every swap of pieces of puzzle.

*a function to check after each swap ‘is the puzzle solved?’.
*a function to solve the puzzle for player at any time.

## Data Structures Used

We have used **‘Doubly Linked List’** implementation for our project.We basically use two DL Lists for our project.One is to store actual picture in one DL list and other DL list to store the shuffled picture puzzle.Where first DL list is constant for that picture but where as second DL list is used to dynamically modify according to to the players movements each time.And these two DL list are used to check for ‘is puzzle solved?’after every movement the player makes.

Here each Dnode that we are linking has four addresses to ascess them, namely next,left of type Dnode itself str of type String, and token of type int. left and right are used to move through the list in both forward and backward direction.And String str is used to store the image of the pieces, int token is for storing the value/count of each node.

## TOOLS USED

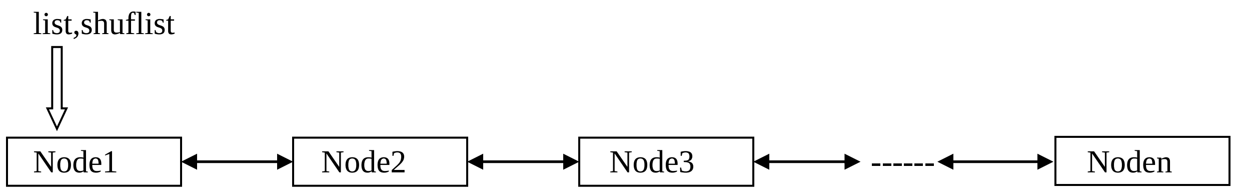
### JAVA SWINGS

"Swing" refers to the new library of GUI controls (buttons, sliders, checkboxes, etc.) that replaces the somewhat weak and inflexible AWT controls.

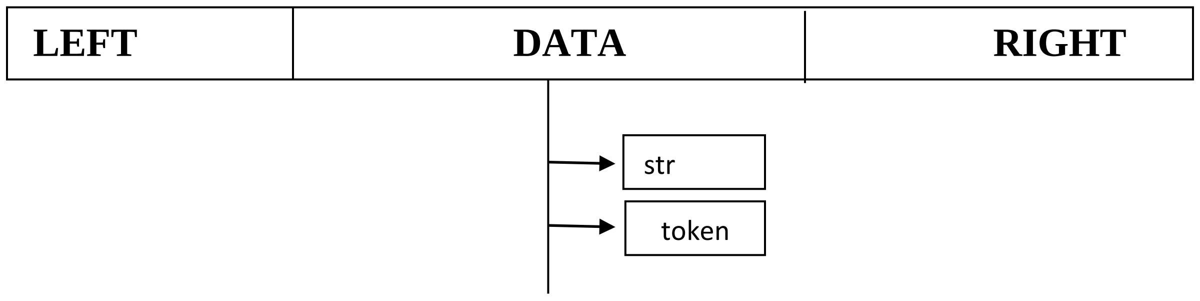
The Swing classes eliminate Java's biggest weakness: its relatively primitive user interface toolkit. Java Swing helps you to take full advantage of the Swing classes, providing detailed descriptions of every class and interface in the key Swing packages. It shows you how to use all of the new components, allowing you to build state-of-the-art user interfaces and giving you the context you need to understand what you're doing. It's more than documentation; Java Swing helps you develop code quickly and effectively.

## DESIGN ( DATA STRUCTURE OVERVIEW )

### Doubly Linked List DESIGN:



Node Structure:



We are using the node Node to form a doubly linked list and in each node we are storing the details of the piece of picture.

In each node we are having different fields as follows

- left- Address of the previous node.
- str-image of piece of puzzle.
- token-value/count of each node/image in list.
- right-Address of the next node.

## IMPLEMENTATION AND ALGORITHM

- **<u>Class Node</u>**

  In this class we have fields to store the details of employees. In each node we are having different fields as follows
  - left- Address of the previous node.
  - str-image of piece of puzzle.
  - token-value/count of each node/image in list.
  - right-Address of the next node.

- **<u>Class SimpleLIST and DifficultLIST</u>**

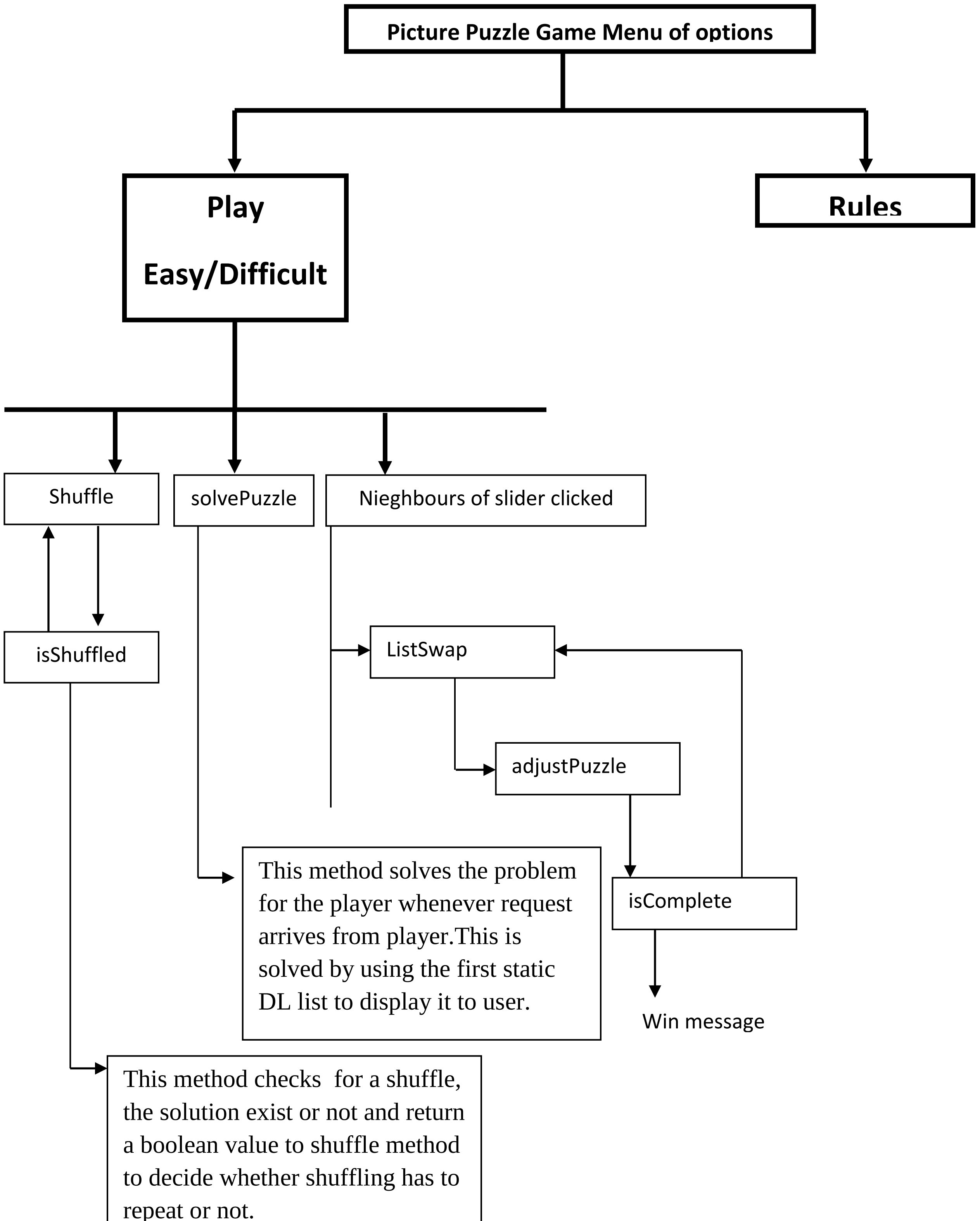
  In these classes we have implemented the following methods

  - **<u>createList()</u>**:this method creates the Doubly linked list of actual picture with list as the starting Node as well as assigns respective images and tokens.
  - **<u>createPuzzle</u>**: this method does the job of copy the contents of earlier list to another DL list called shuflist .
  - **<u>shuffle()</u>** : this method does the job of shuffling the images of the picture creating the puzzle.We have used modern day shuffling algorithm to shuffle list.
  - **<u>Actionperformed(ActionEvent)</u>**: this method handles the event occurring and redirects to respective methods to be called namely createpuzzle on clicking shuffle button,solvPuzzle on clicking solve button,ListSwap when neighbourings of slider are clicked.
  - **<u>ListSwap</u>** : this method perform the swapping after action is performed.
  - **<u>adjustpuz</u>** : this method is called after every swap to redraw the frame of puzzle picture pieces according to the swap has happened,assigning the respective images to pieces of Picture.
  - **<u>iscrctShuffled</u>**: this method checks for a shuffle the solution exist or not and return a boolean value to shuffle method to decide whether shuffling has to repeat or not.
  - **<u>Iscomplete:</u>**this method is called after every swap to check ‘whether the puzzle is solved?’ and return a boolean value to calling method to stop further swapping and display a win message.
  - **<u>solvePuzzle:</u>**this method solves the problem for the player whnever request arrives from player.this is solved by using the first static DL list to display it to user.

- **StartPage**

  In this class we just decorate our home page using JAVA SWING and provide menu of playing options to the player.

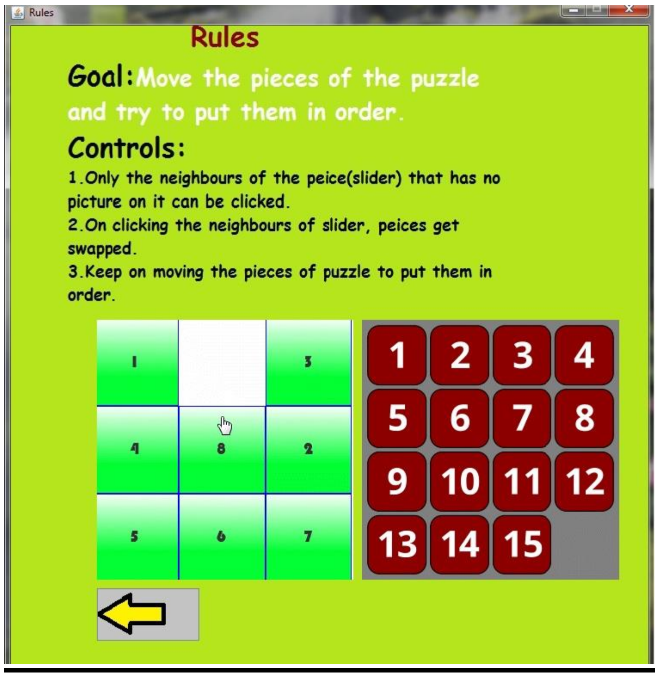
## FLOW CHART



This method solves the problem for the player whenever request arrives from player.This is solved by using the first static DL list to display it to user.

This method checks for a shuffle, the solution exist or not and return a boolean value to shuffle method to decide whether shuffling has to repeat or not.

# TEST CASES & SNAPSHOTS



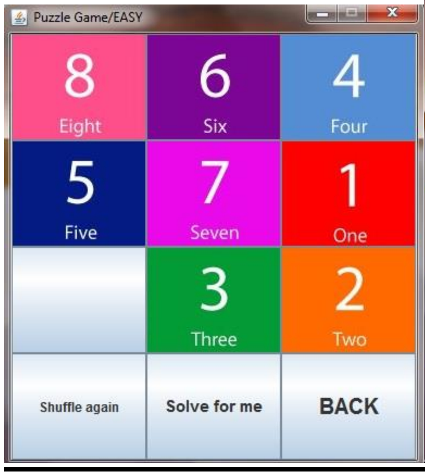
Rules

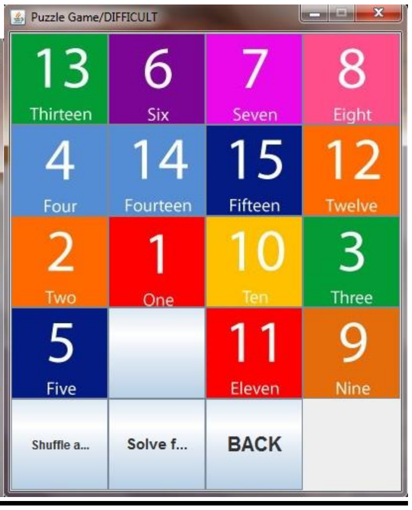
Rules

**Goal:** Move the pieces of the puzzle and try to put them in order.

**Controls:**

1.Only the neighbours of the peice(slider) that has no picture on it can be clicked.
2.On clicking the neighbours of slider, peices get swapped.
3.Keep on moving the pieces of puzzle to put them in order.





## CONCLUSION

This was our project on PICTURE PUZZLE GAME.We think this appliacation will give a good entertain to players. We have tried our hard to program efficiently without any problem.We take this opportunity to express our sense of indebtedness and gratitude to all those people who helped us in completing this project. This project has contributed a lot to my knowledge that has proved to be a value addition for me.

## SCOPE FOR IMPROVEMENT

**We have used only one picture to create and shuffle the picture slices,**

**But we would like to improve in such a way that there should be a option for user to select a user input picture to create the puzzle.**

**There is no option of time elapsed since start of the game. We should have included that**

**And also there should be way that we can store and display total number of puzzle solved in the game.**