

AlgorithmicPuzzles app

Technical Document

Contents

1	Introduction	2
2	Abstract	2
3	System Requirements Specification	2
4	Design of Application	3
5	Conclusion	4

1 Introduction

AlgorithmicPuzzles app is a android application which helps to learn the algorithms in an effective manner by solving the puzzles. Algorithms are a step-by-step procedure for calculations. Algorithms are used for calculation, data processing and automated reasoning. Algorithms are essential to the way computers process the data. Algorithms are basics to solve any problems.

Although algorithms do constitute the cornerstone of computer science and no sensible computer programming is possible without them, it is a common misconception to equate the two. Some algorithmic puzzles predate computers by more than a thousand years. It is true, however, that the proliferation of computers has made algorithmic problem solving important in many areas of modern life, from hard and soft science to art and entertainment. Solving algorithmic puzzles is the most productive and definitely most enjoyable way to develop and strengthen one's algorithmic thinking skills. In this approach, AlgorithmicPuzzles app helps them to learn the algorithms, improve their algorithmic thinking by analysing and solving the puzzles built on them.

2 Abstract

Many people often find it difficult to understand the algorithms which are very essential in problem solving approach. Hence, they find it difficult to solve the problems. Algorithms are indeed the basic step for any programmers and those who are working in the computer science field. Without the help of these algorithms they cannot proceed the further steps. Hence, it is very important to learn these algorithms logic correctly and precisely. Puzzles are an ideal vehicle for mastering algorithmic thinking skill for two reasons. First, Puzzles are fun and a person is normally willing to put more effort into solving them than in doing routine exercises. Second, algorithmic puzzles force a solver to think on a more abstract level. Even computer science students have a tendency to think about algorithmic problems in terms of a computer language they know instead of applying general design and analysis strategies. Puzzles can rectify this important deficiency. Algorithmic Puzzles app will help them in this approach by solving the puzzles which are built on the algorithms and thus helping them to understand the problem solving approach using those algorithms. Thus by solving those Puzzles will help them to overcome their deficiency and also help them to enrich their problem solving capability.

3 System Requirements Specification

AlgorithmicPuzzles app will work on the following system requirements specification

- Android-version:** version 3.0(HONEYCOMB,API-11) and above
- Memory:** 5-6 MB
- CPU:** A single core processor of clock speed 600Mhz with a RAM of 256MB or higher.
- Additional-Software:** This is a standalone application. This app doesnot require any support from other app.

4 Design of Application

Algorithmic Puzzles app helps to learn the algorithms very effectively with the help of solving the puzzles built on them. The application provides the following puzzles:

- **Eight Queen Puzzle:**

In this puzzle, the user needs to place 8 queens in a chessboard, so that no one attacks another. This puzzle consists of only one level which upon solving correctly, a message containing Time required by you will be displayed on the screen.

- **Sorting Puzzle:**

In this puzzle, the user needs to place the coloured bubbles present on UI which disappear one by one from the screen at first and then the coloured bubbles once again appear on the screen upon which the user needs to place those coloured bubbles to their respective numbered circles in the order of their disappearance. This puzzle consists of 10 levels. Solving a level correctly will lead to the next level by the help of next button.

- **Tower of Hanoi:**

In this puzzle, the user needs to place all the 3 rings present on the first tower to the third tower such that at any time smaller ring should be present on the larger ring. This puzzle consists of only one level which upon solving the puzzle correctly Game over dialog and message containing the number of moves required by you will be displayed on the screen.

In the first Interface of app, the link to these puzzles are given by the help of buttons. The algorithms of backtrack and Tower of hanoi are shown for the respective puzzles and for the Sorting puzzle it directly lead to the puzzle. The design of 3 Puzzles are given as below.

- **Eight Queen Puzzle**

As we know, it should consist of chessboard like board on which eight queens will be placed and this board which we want is resized to screen and it is designed by `ImageResizer` method in graphics package. And with the help of `onDraw` method canvas is drawn which represents the chessboard like board. There is also `init solver` method which initialises the solution once the solving of puzzle is started. A `array moveable` is used to store the solution.

In the `OnCreate` method of the `PuzzleActivity`, a `Puzzleview` instance is initialised. Also a `Puzzle` object is created and is injected into the `Puzzleview`. The `Puzzleview` is responsible for servicing the user interface, performs the job by calling the `Puzzle.draw` and `Puzzle.onTouchEvent` methods.

The `ImageResizer` method is a utility class that provides a method for dynamically resizing a bitmap. An instance of this class needs to be initialized by providing the dimensions of the original and resized bitmaps. This class is utilized in the `onSizeChanged` method of `Puzzleview`. The `onSizeChanged` method is called whenever the dimensions of the screen changes (e.g. when an orientation change happens or the screen is first displayed). At this method, the tile images are loaded from the application's resources. The loaded bitmaps are then resized to fit into the existing screen and kept in memory. The images are drawn on the canvas at the `draw` method of the `Eight Queen Puzzle`.

In this Interface, first a chessboard appears on the screen on which you have to place the 8 Queens such that without attacking each other and once you start to solve the timer gets running and the restart button appears on the screen. `ImageResizer` method is helpful in the resizing the image to screen.

- **Sorting Puzzle**

This Puzzle implements the Drag and Drop feature where there are coloured bubbles and circles associated with the interface in which you have to place the coloured bubbles into respective circles in order of their disappearance. This Puzzle consists of 10 levels. Solving the first level correctly a next button appears on the screen otherwise retry button appears on the screen. This is implemented by `OnTouchListener` and `OnDragListener` methods. These methods

helps us to drag the coloured bubbles towards the respective circles and place it.

In this Interface, there will be coloured bubbles appearing on the screen which disappears one by one when Activity gets started and after all the bubbles disappear these bubbles appear once again on the screen and user have to place the bubbles in the respective order in the circles present. If users place the bubbles correctly they will be moved to next level. Through the methods `OnTouchListener` and `OnDragListener` those ideas are implemented.

- **Tower of Hanoi**

This Puzzle consists of 3 towers and 3 rings and this is a classical 3 ring Tower of hanoi instance.

Designing the tower of hanoi puzzle, as we know it should consist of three towers and three rings and rings are the area where we should be able to touch. These designs are made possible by `Toh` class which extends the `SimpleBaseGameActivity`. The `SimpleBaseGameActivity` provides callbacks and contains the code to make `AndEngine` which is used as library to work with the Activity lifecycle. Each callback that it provides is used for specific purpose. As soon as you extend this class, we have to override three functions. Here is a brief description of those functions that are used in designing the puzzle:

- **onCreateEngineOptions:**

- This function is where you create an instance of the engine. Every activity that the puzzle uses will have its own instance of the engine that will run within the activity lifecycle.

- **onCreateResources:**

- This is the function where you'll load all the resources that the activity requires into the VRAM.

- **onCreateScene:**

- This function is called after the above two call backs are executed. This is where we will create the scene for puzzle and use all the textures that are previously loaded into the memory.

At first when the `SimpleBaseActivity` class is started all the three rings are stacked to first tower and you have to replace these three rings from first tower to third tower such that at any instance smaller ring should be present on the larger ring. The game scene is created by the `onCreateScene` where x and y co-ordinates are specified. And the three towers are added as sprites. And with the help of `Ring` class the rings are controlled and to them touch handlers are added so that rings are the areas where the user will be able to touch. The method `checkForCollisionsWithTowers` helps to check if the ring has collided with or rather, is touching a tower.

The game logic is that the three towers represent as three stacks and you can only remove the above ring. After the third stack size is equal to 3 that is after solving the puzzle then the Game over dialog and the number of moves needed by you to solve the puzzle appears on the screen. The `onCreateScene` helps to create the scene on the screen.

In the Tower of hanoi Puzzle as we know at every stage the ring present on the top can be removed, thus the towers represent as three stacks where initially the first stack will be filled with rings and other two stacks will be empty. After that you have to fill the third stack with the rings such that every stage the smaller ring should be present on the larger ring.

5 Conclusion

Thus this technical document tells about the AlgorithmicPuzzle app's system requirements specification and the designs of the puzzles such as chessboard design of eight queen puzzle, drag and drop feature of sorting puzzle and the design of tower of hanoi puzzle. Thus this document explains about the design implementations of the AlgorithmicPuzzles application.