CS-240 Project Report

CROSSWORD PUZZLE GENERATION SYSTEM

Abstract: In this project, we attempt to design a random crossword puzzle generator system. We have subdivided our problem of crossword puzzle generation into two sub-goals. One is the design of Graphical User Interface and other is algorithm implementation part.

For each sub-goal, we conduct a series of experiments to check the generated crossword puzzle efficiency.

Submitted by

16010108 Anupriya Sinha
16010103 Kshitiz Kumar

4th Semester :: January - May 2018 session
Bachelor of Technology in Computer Science and Engineering

Under the guidance of:
Dr. Navanath Saharia

Department of Computer Science and Engineering
Indian Institute of Information Technology Manipur
Mantripukhri, Imphal, India - 795002

Chapter 1

Introduction

Crossword puzzles is one of the most popular word game. The first crosswords appeared in England during the 19th century. The first known published crossword puzzle was created by a journalist named Arthur Wynne from Liverpool, and he is usually credited as the inventor of the popular word game. The Crossword puzzle checks the ability of people to see words puzzled together in given geometrical patterns and thrie IQ.

1.1 Problem definition

To make a CROSSWORD PUZZLE GENERATION SYSTEM that will retrieve certain number of commands from database and generate crossword puzzle with the retrieved words.

1.2 Introduction to problem statement

To design CROSSWORD PUZZLE GENERATION SYSTEM.

1.3 Objective of this study

To analyse crossword puzzle generator algorithms and implement algorithm ourself.

1.4 Background and recent study

Crossword Maker is a great, free game, that belongs to the category PC games with subcategory Pastimes (more specifically Crosswords) and has been published by Adders. The interface created is based on this game only. It has a matrix of 10*10 order that can generate random puzzles for players.

Chapter 2

System architecture

To implement our project we are using the following softwares :

1. Eclipse : To implement the project
2. Pgsql : To store Unix commands and its description in database

Both the above mentioned software are open source softwares so, the project implementation is free of cost.

For Graphical User Interface (GUI), swing components of java are used. GUI for game consists of one 10x10 matrix and two text area, one to display hint and other one to dispaly solution for the puzzle.

The "postgres" SQL is used to store words i.e unix commands and its description that are used to generate crossword puzzle. Words are used to generate crossword puzzle and descriptions are used to generate hint for the game.

2.1 Design

In the process of crossword puzzle generation, first of all the words are extracted from database and stored in string array; among the extracted words, eight words and its hint are are randomly selected in sequence (order in which they are stored in database) and stored in string arrays. These words are than feeded to the crossword puzzle generator algorithm which result in 10x10 matrix puzzle generation. The generated game is finally displayed in GUI 10x10 matrix and hints are displayed in text frame. After submittimg the solution by user the correct solution will be dispalyed.

PROJECT LAYOUT :

Here is the layout of CROSSWORD PUZZLE generator and schema of database used in this project :

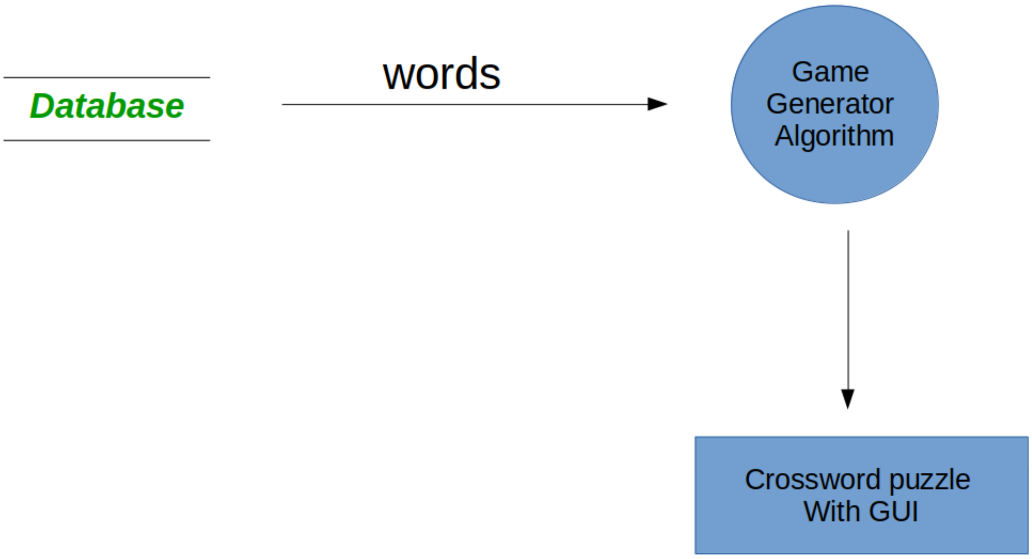
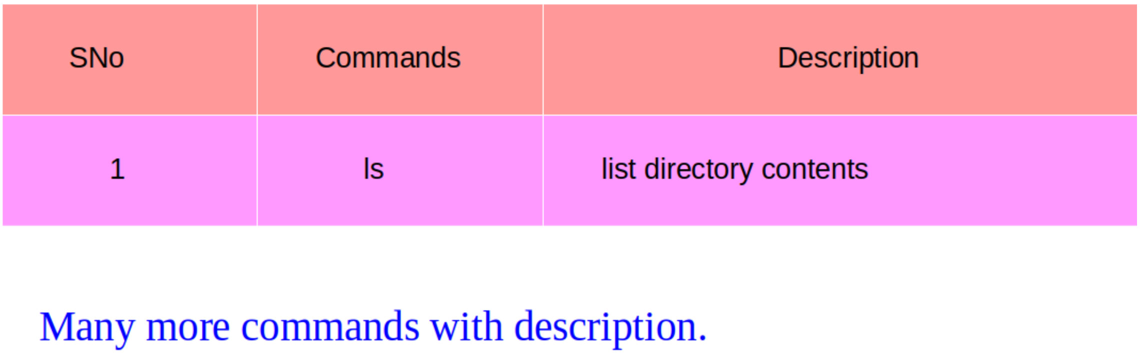


Figure 2.1: Project Layout



SNo	Commands	Description
1	ls	list directory contents

Many more commands with description.

Figure 2.2: DB schema

Chapter 3

Work Done

3.1 Introduction

This chapter discusses the implementation i.e the algorithm used to design this system and and the result. It also dicusses the analaysis of designed system and the screenhots of the implemented system.

3.2 Implementation

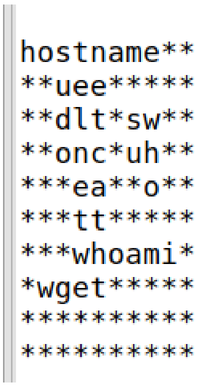
3.2.1 Algorithm

The implemented algorithm takes eight words as input and results in 10x10 matrix crossword puzzle and hints for game as output. Two variables to keep track of last horizontal and vertical indexes occupied by the character are maintained and an integer list (predefined datastructure) to store the indexes of column that are verically occupied by other words whose character matches with the characters placed in first row.

The steps involved in the algorithm are as follows :

```
1. Arrange the random words selected in descending order of words length.
2. Longest word is placed in the first row horizintally and a variable is maintained to keep track of the horizontal index next to the last character.
3.   Check if next word can fit in first row.
4.     If it can be placed in first row then it will be placed their.
5.     Else check if the first character of next word to be placed matches any character of word(words) already placed in first row.
6.       If it matches then word is placed vertically and a variable is maintained to keep track of vertical index next to last character.
7.       Else word is stored in String array say, ”x”
8. Half of the words in array ”x” are placed vertically from the last index occupied by characters in first row
9. Rest half are placed horizintally from the last index occupied by any character vertically starting from any random horizontal index
```

Here is the sample output of implemented algorithm :



```
hostname**
**uee*****
**dlt*sw**
**onc*uh**
***ea**o**
***tt*****
***whoami*
*wget*****
**********
**********
```

Figure 3.1: Puzzle Interface

3.3 Result

A crossword puzzle generation system has been designed that can generate many random crossword puzzles.

3.4 Analysis

LIMITATIONS :

On analysis, it has been found that the implemented crossword puzzle system has the following limitations :

1. If first chatacter of words doesn't matches any chatacter placed in first row then puzzle will not be an effecient.
2. Algorithm doesn't matches any characters except for the first one to generate puzzle.
3. Crossword puzzle can only be generated for words less than or equal to 10.
4. Number of words generated in puzzle depends on the length of word. If words with longer length(more than 5) are more, less number of words will be generated in puzzle.

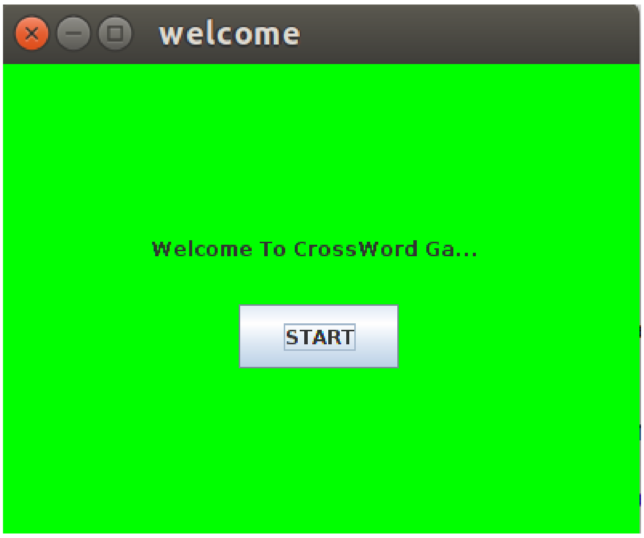
3.5 How to use

In order to use this crossword puzzle generation system; Eclipse, pgsql and pqsql connector are required.

3.6 Screenshots

3.6.1 Welcome Page

This is the welcome page to greet users and ask them to start the crossword puzzle.

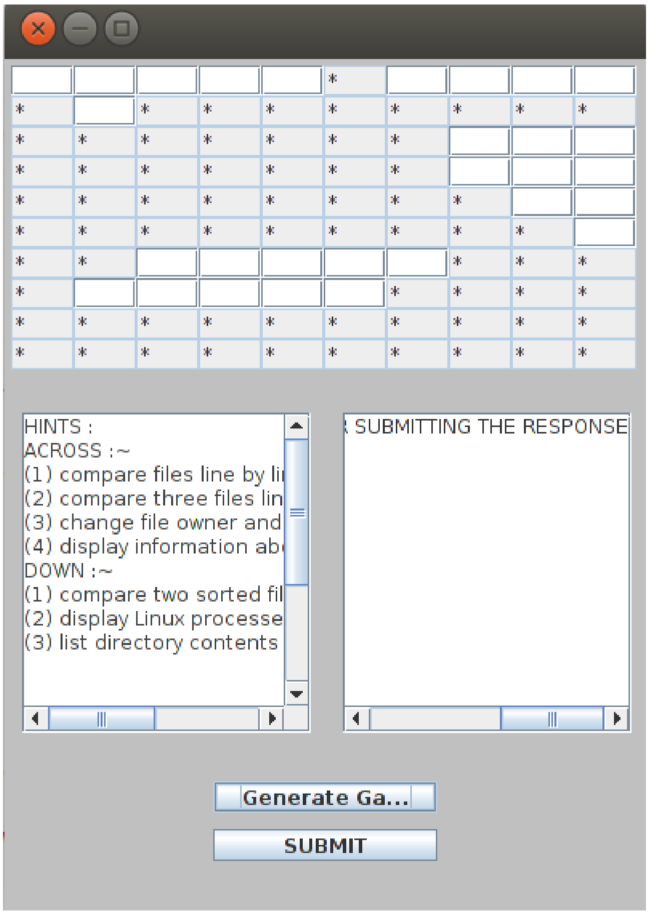


Welcome To CrossWord Ga...

Figure 3.2: Welcome Page

3.6.2 Generated Puzzle

This the second window where user will be directed after clicking the "start" button of first window. On clicking the "Generate Game", the puzzle and the hints will be generated.



					*				
*		*	*	*	*	*	*	*	*
*	*	*	*	*	*	*			
*	*	*	*	*	*	*			
*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*	*	*	
*	*						*	*	*
*						*	*	*	*
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*

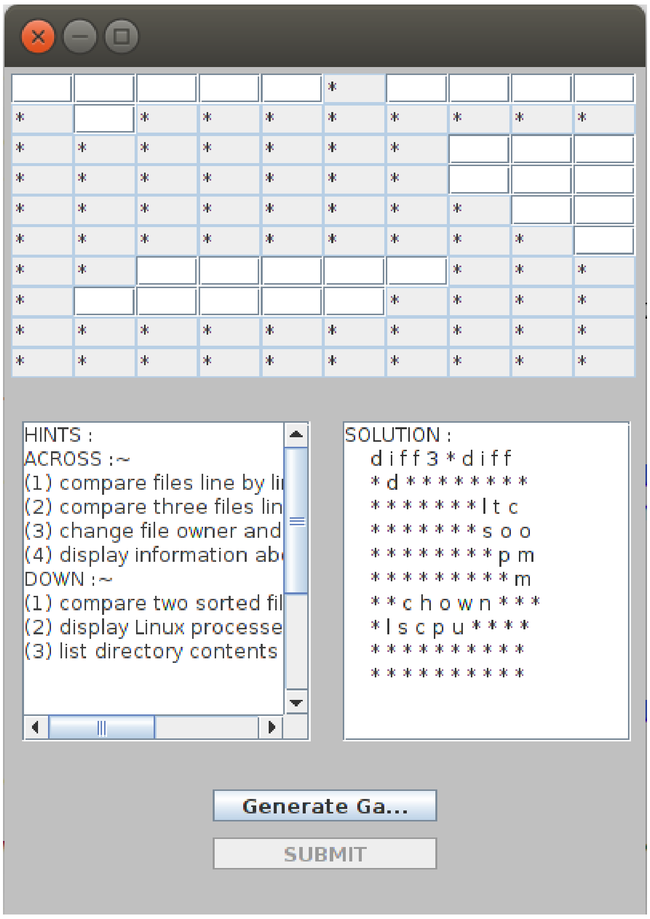
HINTS :
ACROSS :~
(1) compare files line by li
(2) compare three files lin
(3) change file owner and
(4) display information ab
DOWN :~
(1) compare two sorted fil
(2) display Linux processe
(3) list directory contents

SUBMITTING THE RESPONSE

Figure 3.3: Generated Puzzle with hints

3.6.3 Generated Puzzle with solution

Answer for the puzzle will be generated on submitting the guessed answer by user in second window itself as shown below.



					*				
*		*	*	*	*	*	*	*	*
*	*	*	*	*	*	*			
*	*	*	*	*	*	*			
*	*	*	*	*	*	*	*		
*	*	*	*	*	*	*	*	*	
*	*						*	*	*
*						*	*	*	*
*	*	*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*	*	*

```
HINTS :
ACROSS :~
(1) compare files line by li
(2) compare three files lin
(3) change file owner and
(4) display information ab
DOWN :~
(1) compare two sorted fil
(2) display Linux processe
(3) list directory contents
```

```
SOLUTION :
   d i f f 3 * d i f f
   * d * * * * * * * *
   * * * * * * * l t c
   * * * * * * * s o o
   * * * * * * * * p m
   * * * * * * * * * m
   * * c h o w n * * *
   * l s c p u * * * *
   * * * * * * * * * *
   * * * * * * * * * *
```

Figure 3.4: Puzzle with solution

Chapter 4

Testing

4.1 Used techniques

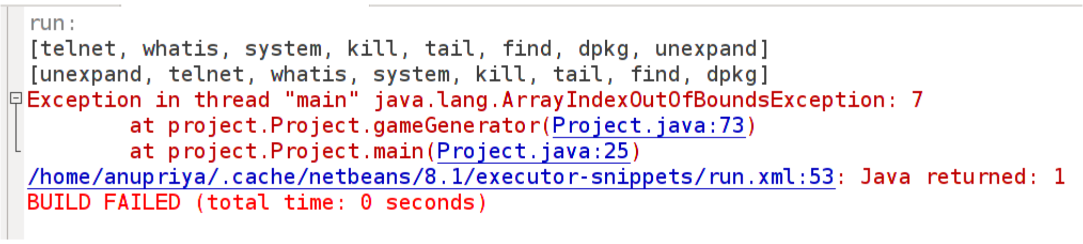
UNIT TESTING :

Various test cases were used to test the implemented system. Different combinations of words were choosed to check the extent upto which the system can work properly and also to check the limitations.

4.2 Test cases

One of the combination of words not working for the implemented system is :

1. telnet, whatis, system, kill, tail, find, dpkg, unexpand



```
run:
[telnet, whatis, system, kill, tail, find, dpkg, unexpand]
[unexpand, telnet, whatis, system, kill, tail, find, dpkg]
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 7
        at project.Project.gameGenerator(Project.java:73)
        at project.Project.main(Project.java:25)
/home/anupriya/.cache/netbeans/8.1/executor-snippets/run.xml:53: Java returned: 1
BUILD FAILED (total time: 0 seconds)
```

Figure 4.1: Result for test case 5

The reason why this test case fails is that, it contain words whose first character doesn't matches the character of longest word placed in first row.

Chapter 5

Conclusion

A crossword puzzle generator system has been designed to generate many random crossword puzzles.

5.1 Future work

The algorithm can be midified to make it better crossword puzzle generator as the implemented algorithm has some limitations. The designed system generate puzzles of 10x10 matrix, it can be extended for nxn matrix in future. Attempt can be made to overcome the limitations and make the algorithm advanced.

5.2 Acknowledgement

We would like to extend gratitude to our mentor Dr. Navanath Saharia, Assistant professor at IIIT Manipur, for providing such a golden opportunity to do this wonderful project ”CROSSWORD PUZZLE GENERATION SYSTEM”.

Anupriya Sinha
Kshitiz Kumar