Word Search Puzzle (P9)

Venkhat V, Pinkesh Barsopia

Department of Electrical Engineering
Indian Institute of Technology Bombay
GitHub link: https://github.com/pinkeshb/SDES_wordpuzzle/tree/pdev

April 29, 2015

Word Search Puzzle (P9) - Project Aim

- Interactive game based on Classic Word Search Puzzle
- Search a set of words embedded in a $N \times N$ matrix of characters

Different features

1. User selectable difficulty levels, Dictionary, Grid Size
2. Score
3. Time limit

Game Specifications

User Input

1. Size of grid - N
2. Difficulty level - Easy, Medium Hard
3. Dictionary - Animals, Cars

Game parameters

1. Length of longest word, L
2. No. of words
3. Intersection of words, I
4. Display the set of words
5. Random filling of characters

	Easy	Medium	Hard
	I- No Display - Yes	I- Yes Display - Yes	I- Yes Display - No
N = 8	WL - 6 No.of words =6	WL- 8 No. of words =8	WL - 8 No. of words =8
N = 12	WL - 8 No.of words =8	WL- 10 No. of words =12	WL - 10 No. of words =12

PyGame- GUI , Python - algorithm, Wx - for options widget

Modules

Modules

1. **Options GUI** - Gets input from user
2. **Game Settings** - Stores user input and calculates Game Design parameters
3. **Character Matrix** - holds matrix and get and set word and random fill function
4. **Wordlist**
   - holds word list and strategically populates matrix with words
   - chooses words and places them according to the difficulty levels
5. **Game GUI** - sets up the GAME and monitors the user actions
6. **Game Status** - stores the user game state
   Eg.current - score, time, word found

main function - which integrates all and runs the game

Intermediate Work Done

Intermediate Work Done - Until 24^{th} March

- Completed 3 modules with unit testing
  1. Character Matrix
  2. Word List
  3. Game GUI
- Difficulty level 0 - ensuring no overlaps and even spread of words
- GUI with fixed sized window, 2 mouse clicks selects the word

ScreenShots

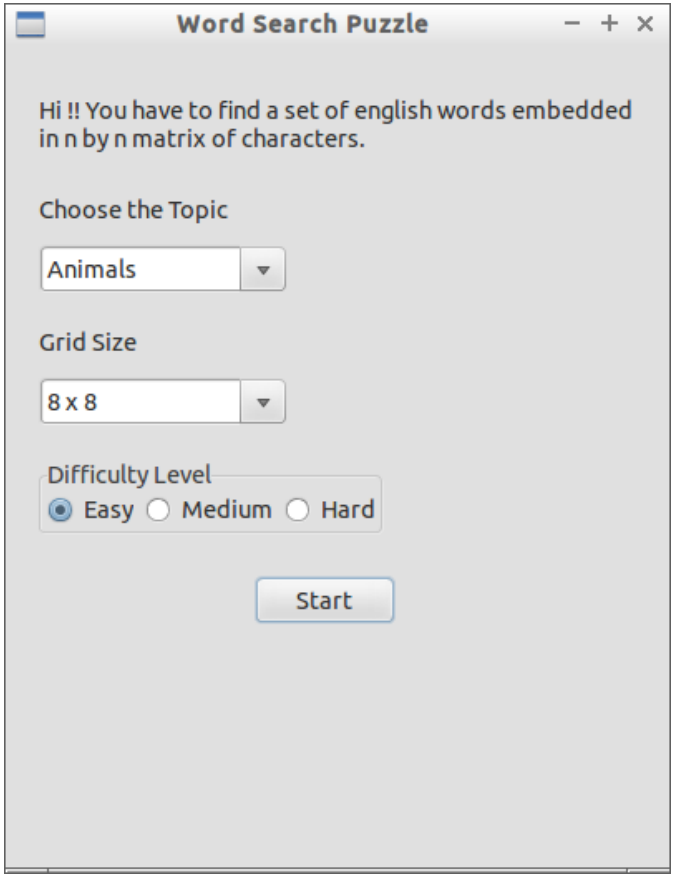


Figure : Options Menu

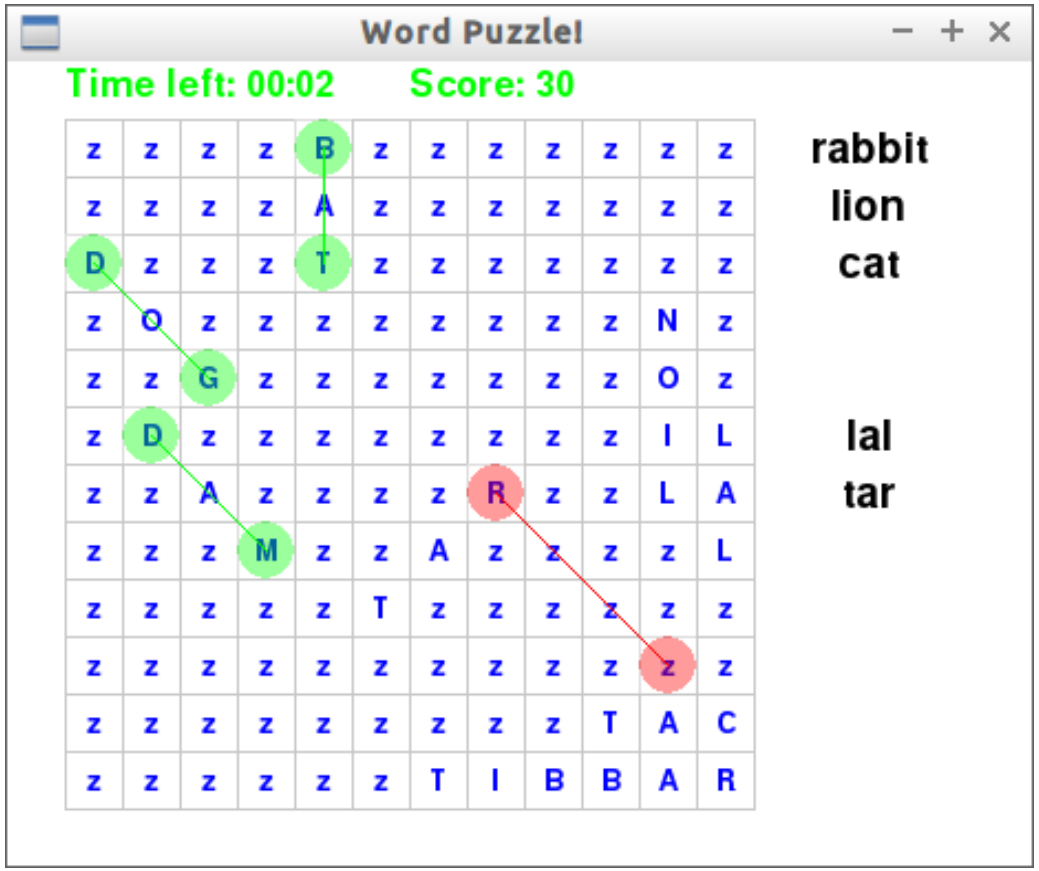
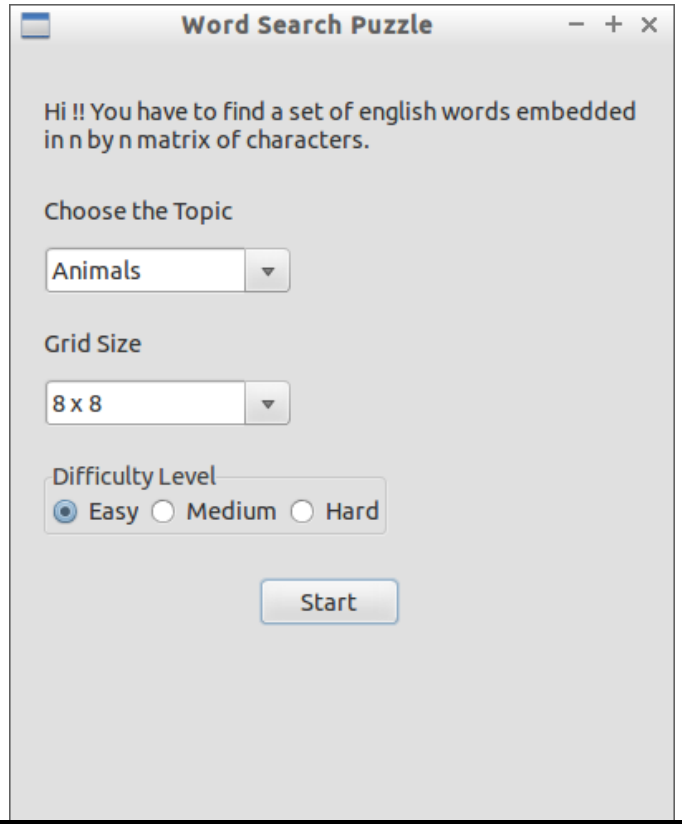


Figure : Puzzle Running

Game Flow

1

Start Options Menu GUI



2

Store in Game Settings

3

Generate Word List and their Positions

goldfish
hamster
kitten
bear
lion
cow
dog
owl

Game Flow - Continued

4

Generate Character Matrix

5

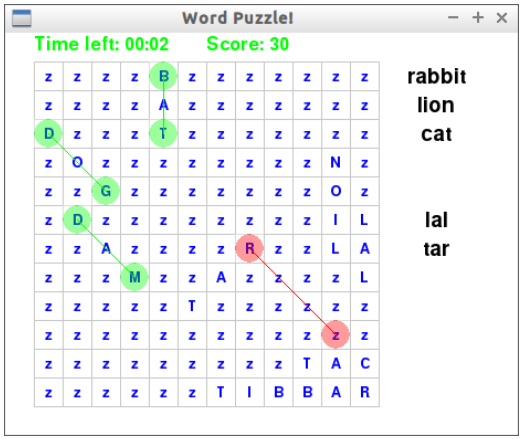
Initialize Game State

6

Start Game GUI

Time left: 00:00 **Score: 0**

			M								
	L			O							
		E			N						
H			R			K	T		Y	L	F
S				R			E	A			
I					I		H	Y	C		
F						U		S			
D							Q		I		
L				X	O	F		S		F	
O		R	E	G	I	T					
G											



Design Strategy

Design Strategy

- MVC -Model View Controller for GAME GUI
- Divide into modules -oop design
- Top down Approach

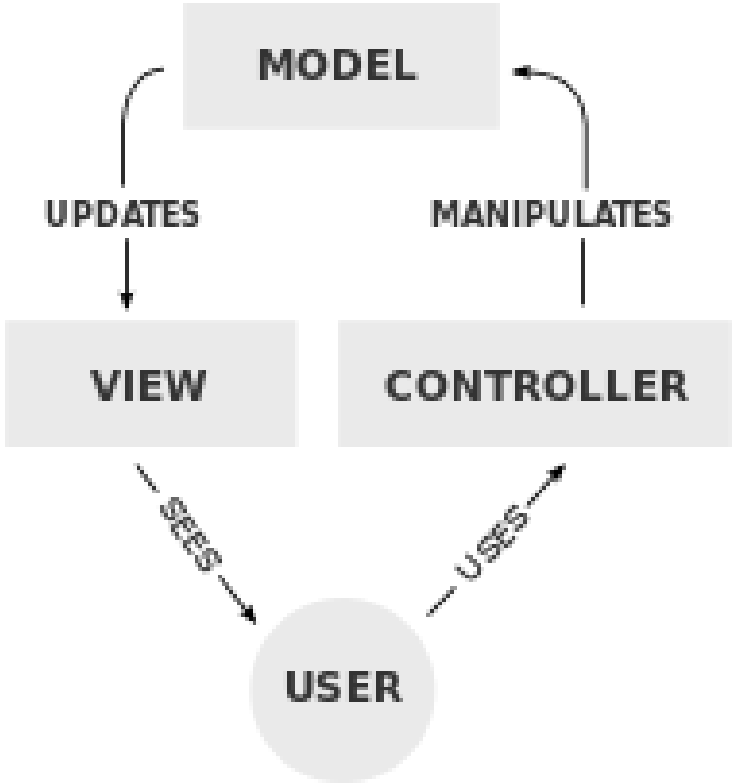


Figure : Model View Controller

Image Source: Wikipedia

Algo Features and Testing

Algo Features

1. Sticky Mouse Postitioning in Game GUI
2. Algorithm to find position of words to be placed.

Testing

- Logic and GUI has been separated as modules and developed separately
- Separation of Concern
- The logic has been tested using unit test.
- Inside the modules, divided into functions and tested individually.

Possible Upgrades

- Enhance User experience - by improving GUI
- Generate More diffuculty levels
- User Memory- store the specific user settings
- level by level unlocking

Resources

1. Word Postitioning Algorithm
   - http://stackoverflow.com/questions/6332652/a-fast-algorithm-for-creating-a-puzzle
2. PyGame
   - inventwithpython.com/pygame/chapter2.html
3. Wx Widget
   - http://wiki.wxpython.org/Getting%20Started
   - http://zetcode.com/wxpython/

Thank you