

**“School of Computer Science & Engineering”**

**Lovely Professional University**

**PROJECT REPORT**

**WORD PUZZLE GAME**

**PROJECT PARTICIPANTS:**

1. **Pentakota Jaswanth (12111901) - Roll.NO.: B52**
2. D Praveen (12115126) - Roll.NO.: A03
3. Shibu Prakash (12116263) - Roll.NO.: A28

**PROJECT SUBMITTED TO:**

* **Navpreet Rupal**

**PROJECT TITLE:**

Design a Puzzle Game in which, various letters will be shown in a table and he/she can move horizontally, vertically or diagonally in order to make meaningful words.

**Contents:**

1. Introduction
2. Product Description
3. Objectives
4. Design
5. Code
6. Result
7. Role of the participants
8. Gantt Chat
9. Conclusion
10. Reference

**INTRODUCTION:**

Word games (also called word game puzzles) are spoken or board games often designed to test ability with language or to explore its properties.

Word games are generally used as a source of entertainment, but can additionally serve an educational purpose. Young children, adults and other age group people can enjoy playing games such as Hangman, word scrambling game while naturally developing important language skills like spelling. While Hangman is a seriously dark game, what we like to focus on is the development of the children.

Researchers have also found that adults, kids, and other age group people who regularly solved crossword puzzles, which require familiarity with a larger vocabulary, had better brain function later in life.

This project will be a dependent on user only, as there will be many English meaningful words that will be hidden and user have to find them one by one.

**Project Description**

This game will consist of table of with many alphabets set in a random order and many English meaningful words will be hidden between them user must find them one by one as soon as possible.

Either horizontally or vertically. At the right we will have “Submit” and “Quit” button.

They can then use submit button to check if they are correct.

At the last on finding all the words user can sit with the quit button to know their final score.

**OBJECTIVE:**

• The main objective of the game is to consult with a list of words supplied with the grid, and find all the words on that list. The puzzle is usually given a theme, and the words on the list typically fit in with this particular theme.

• Puzzlers must spend time investigating the grid and applying various strategies to locate the words on the grid and type them in the space given to gain scores.

• The objective of this word puzzle game is to be the first player to search the meaningful words from the given words.

• It is not just about passing the time, but also about deriving some enjoyment and benefits from it. Just like many other games and puzzles, there is always an objective. Most of these word games and puzzles are designed to provide entertainment, but that is not all.

• This game will consist of table with many alphabets set in a random order and many English meaningful words will be hidden between them. User have to find them one by one as soon as possible.

• At the top of the game, name of the game will be shown which indicates the user that which game is been played by him/her

• Researchers have found that adults who regularly solved crossword puzzles, which require familiarity with a larger vocabulary, had better brain function later in life.

**GRAPHICAL USER INTERFACE:**

A Graphical user interface is an interface through which a user can interact with electronic devices such as computers and other applications with the help of a mouse there are so many graphical user interfaces Tkinter is mostly used as it is fast and easy to create GUI applications. This interface uses icons, menus, and other visual indicator representations to display information and related user controls, unlike text-based interfaces, where data and commands are in the text.

**DESIGN:**

This game is designed to play word puzzle. This game shows a mixed and jumbled words and hint will be given below.

When the game is opened the interface show the game and the hints that should be done.

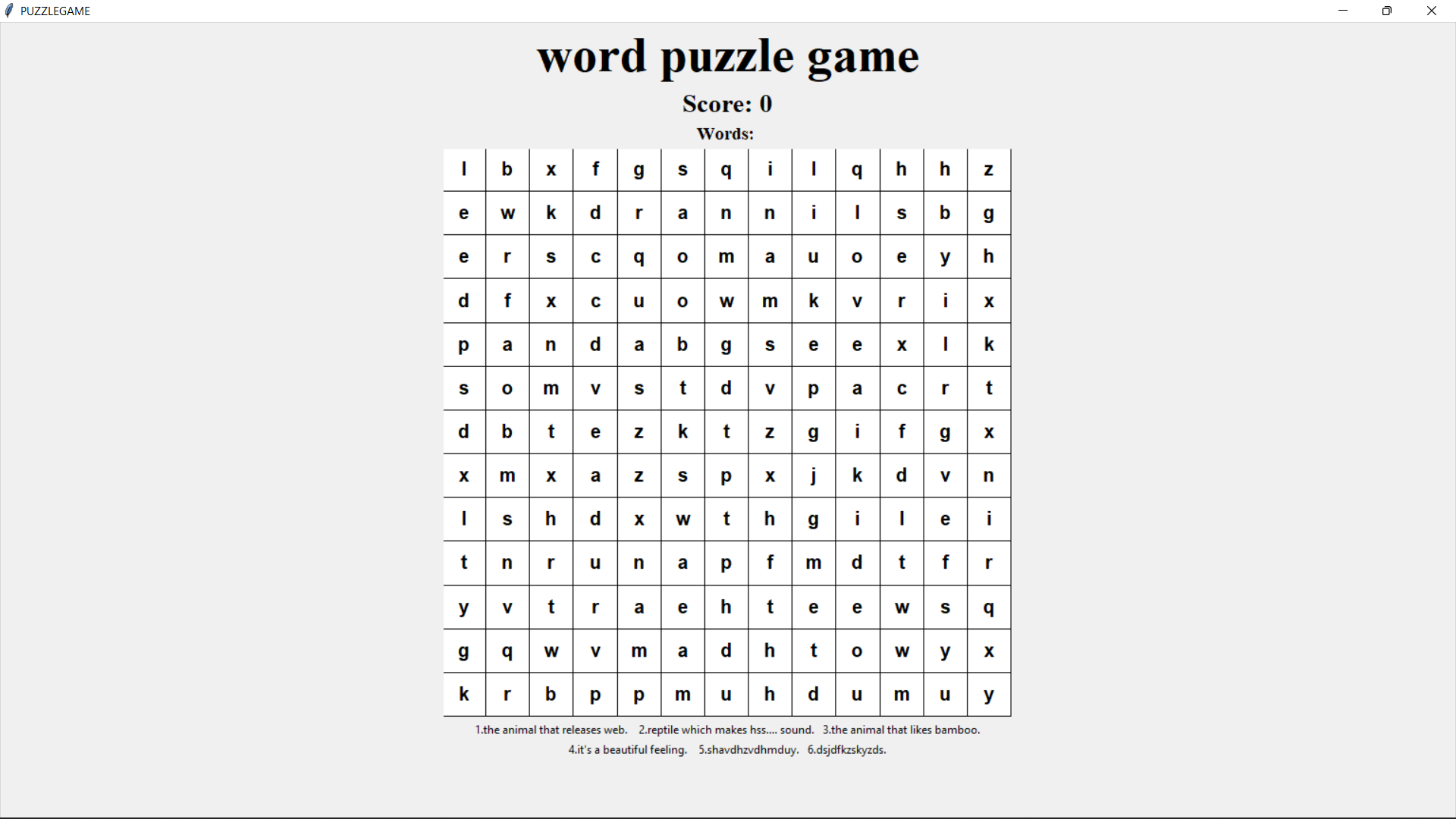
After the game starts, he/she should find the words and select the words.

The selected words will be marked in red and the data base will store the data selected and it will check the selected words with the stored information.

If the selected word gets matched with the database information, then the selected word will be displayed at the top and the score will be increased.

The process goes on until the words in the grid are completed.

This game will be changed for every play and the words in the game will not change because this game is not designed in that way.



**TABLE USED:**

In this project we have took various function and keywords in order to make it look attractive and proper functioning without getting any error and as per requirement and maintain the basic requirement as per the topic and basic requirement given to us.

**Different modules have been used in this python project which can be given and described as:**

1. Tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications.

1. Python 3.7.0:

Python is a general-purpose programming language. Hence, you can use the programming language for developing both desktop and web applications. Also, you can use Python for developing complex scientific and numeric applications. Python is designed with features to facilitate data analysis and visualization

1. NLTK:

Natural Language Toolkit, is a Python package that you can use for NLP. A lot of the data that you could be analysing is unstructured data and contains human-readable text. Before you can analyse that data programmatically, you first need to pre-process it.

.

**CODE OF THE PROJECT:**

**Source Code:**

import tkinter

import random

width = 600

height = 600

gridsize = 13

extra = 100

grid = []

root = tkinter.Tk()

root.title("PUZZLEGAME")

screenwidth = root.winfo\_screenwidth()

screenheight = root.winfo\_screenheight()

root.geometry(f"{width}x{height+extra}+{(screenwidth-width)//2}+{(screenheight-height-extra)//2}")

selected = []

wordlist = []

gotwords = tkinter.StringVar()

gotwords.set("Words: ")

score = tkinter.StringVar()

scoreint = 0

score.set("Score: 0")

restricted = ""

occupied = dict()

allwords = ["spider","snake","panda","Navpreet","Lovely","Mobile"]

'''c.pack()

scores = tkinter.Label(root, textvariable=score)

scores.pack()

'''

l9=tkinter.Label(root,text="word puzzle game")

l9.configure(font=("Times New Roman",40,"bold"))

l9.pack()

scores = tkinter.Label(root, textvariable=score)

scores.configure(font=("Times New Roman",20,"bold"))

scores.pack()

words = tkinter.Label(root, textvariable=gotwords)

words.configure(font=("Times New Roman",14,"bold"))

words.pack()

for i in range(gridsize):

grid.append([])

for j in range(gridsize):

grid[i].append("")

print(grid)

def checkpossible(w,i,j,ch):

if ch == "h":

countj = j

for letter in w:

if grid[i][countj] == "" or grid[i][countj]==letter:

pass

else:

return False

countj = countj+1

return True

elif ch == "v":

counti = i

for letter in w:

if grid[counti][j] == "" or grid[counti][j]==letter:

pass

else:

return False

counti = counti+1

return True

elif ch == "du":

counti = i

countj = j

for letter in w:

if grid[counti][countj] == "" or grid[counti][countj]==letter:

pass

else:

return False

counti = counti-1

countj = countj+1

return True

elif ch == "dd":

counti = i

countj = j

for letter in w:

if grid[counti][countj] == "" or grid[counti][countj]==letter:

pass

else:

return False

counti = counti+1

countj = countj+1

return True

def assign(w):

i = random.choice(range(gridsize))

j = random.choice(range(gridsize))

ch = random.choice(["h","v","du","dd"])

if ch == "h":

if j+len(w)-1<gridsize:

if checkpossible(w, i, j, ch):

count = j

for letter in w:

grid[i][count]=letter

count = count+1

else:

return assign(w)

else:

return assign(w)

elif ch == "v":

if i+len(w)-1<gridsize:

if checkpossible(w, i, j, ch):

count = i

for letter in w:

grid[count][j]=letter

count = count+1

else:

return assign(w)

else:

return assign(w)

elif ch == "du":

if (i-len(w)+1>=0 and j+len(w)-1<gridsize):

if checkpossible(w, i, j, ch):

counti = i

countj = j

for letter in w:

grid[counti][countj]=letter

counti = counti-1

countj = countj+1

else:

return assign(w)

else:

return assign(w)

elif ch == "dd":

if (i+len(w)-1<gridsize and j+len(w)-1<gridsize):

if checkpossible(w, i, j, ch):

counti = i

countj = j

for letter in w:

grid[counti][countj]=letter

counti = counti+1

countj = countj+1

else:

return assign(w)

else:

return assign(w)

for w in allwords:

assign(random.choice([w,w[::-1]]))

for i in range(gridsize):

for j in range(gridsize):

if grid[i][j] == "":

grid[i][j] = random.choice("abcdefghijklmnopqrstuvwxyz")

def printpos(event):

gridcol = event.x\*gridsize//width

gridrow = event.y\*gridsize//height

print("%d %d" % (gridrow, gridcol))

print(grid[gridrow][gridcol])

selectbox(gridrow, gridcol)

def checkvalid(gridrow, gridcol):

global selected

global restricted

if len(selected) == 0:

return True

elif len(selected) == 1:

if abs(selected[0][0] - gridrow)<=1 and abs(selected[0][1] - gridcol)<=1:

if abs(selected[0][0] - gridrow)==1 and abs(selected[0][1] - gridcol)==1:

if selected[0][1]-gridcol+selected[0][0]-gridrow==0:

restricted = "diagonalup"

elif selected[0][1]-gridcol==selected[0][0]-gridrow:

restricted = "diagonaldown"

elif abs(selected[0][0] - gridrow)==1 and abs(selected[0][1] - gridcol)==0:

restricted = "vertical"

elif abs(selected[0][0] - gridrow)==0 and abs(selected[0][1] - gridcol)==1:

restricted = "horizontal"

return True

else:

selected = sorted(selected)

if restricted == "horizontal":

if gridrow == selected[0][0] and ( abs(selected[0][1]-gridcol)==1 or abs(selected[-1][1]-gridcol)==1):

return True

if restricted == "vertical":

if gridcol == selected[0][1] and ( abs(selected[0][0]-gridrow)==1 or abs(selected[-1][0]-gridrow)==1):

return True

if restricted == "diagonalup":

if (selected[0][0]-gridrow==1 and selected[0][1]-gridcol==-1) or (selected[-1][0]-gridrow==-1 and selected[-1][1]-gridcol==1):

return True

if restricted == "diagonaldown":

if (selected[0][0]-gridrow==1 and selected[0][1]-gridcol==1) or (selected[-1][0]-gridrow==-1 and selected[-1][1]-gridcol==-1):

return True

print(restricted)

return False

def selectbox(gridrow, gridcol):

if [gridrow,gridcol] in selected:

c.create\_rectangle(gridcol\*width/gridsize, gridrow\*height/gridsize, gridcol\*width/gridsize + width/gridsize, gridrow\*height/gridsize + height/gridsize, fill="white", outline="black")

c.create\_text(width\*(2\*gridcol+1)/(2\*gridsize), height\*(2\*gridrow+1)/(2\*gridsize), text=grid[gridrow][gridcol], fill="black", font=('Helvetica 15 bold'))

selected.remove([gridrow,gridcol])

else:

if checkvalid(gridrow, gridcol):

c.create\_rectangle(gridcol\*width/gridsize, gridrow\*height/gridsize, gridcol\*width/gridsize + width/gridsize, gridrow\*height/gridsize + height/gridsize, fill="red", outline="black")

c.create\_text(width\*(2\*gridcol+1)/(2\*gridsize), height\*(2\*gridrow+1)/(2\*gridsize), text=grid[gridrow][gridcol], fill="white", font=('Helvetica 15 bold'))

selected.append([gridrow,gridcol])

def evaluate(event):

global selected

global score

global wordlist

global scoreint

selected = sorted(selected)

word = ""

for pos in selected:

word = word + grid[pos[0]][pos[1]]

if word not in wordlist:

if word in allwords:

scoreint = scoreint +1

score.set("Score: "+ str(scoreint))

gotwords.set("Words: "+" ".join(wordlist)+" " +word)

wordlist.append(word)

if word[::-1] in allwords:

scoreint = scoreint +1

score.set("Score: "+ str(scoreint))

gotwords.set("Words: "+" ".join(wordlist)+" " +word[::-1])

wordlist.append(word[::-1])

refreshgrid()

c = tkinter.Canvas(root, bg="white",height=height, width=width)

def refreshgrid(event=1):

global selected

selected = []

for i in range(gridsize):

for j in range(gridsize):

c.create\_rectangle(j\*width/gridsize, i\*height/gridsize, j\*width/gridsize + width/gridsize, i\*height/gridsize + height/gridsize, fill="white", outline="black")

for i in range(gridsize):

for j in range(gridsize):

c.create\_text(width\*(2\*j+1)/(2\*gridsize), height\*(2\*i+1)/(2\*gridsize), text=grid[i][j], fill="black", font=('Helvetica 15 bold'))

c.bind("<Button-1>", printpos)

root.bind("<Return>", evaluate)

root.bind("<Escape>", refreshgrid)

c.pack()

refreshgrid(1)

l1=tkinter.Label(root,text="1.The animal that releases web? 2.Reptile which makes hss.... sound? 3.The animal that eats bamboo?")

l1.configure(font=("Times New Roman",15,"bold"))

l1.pack()

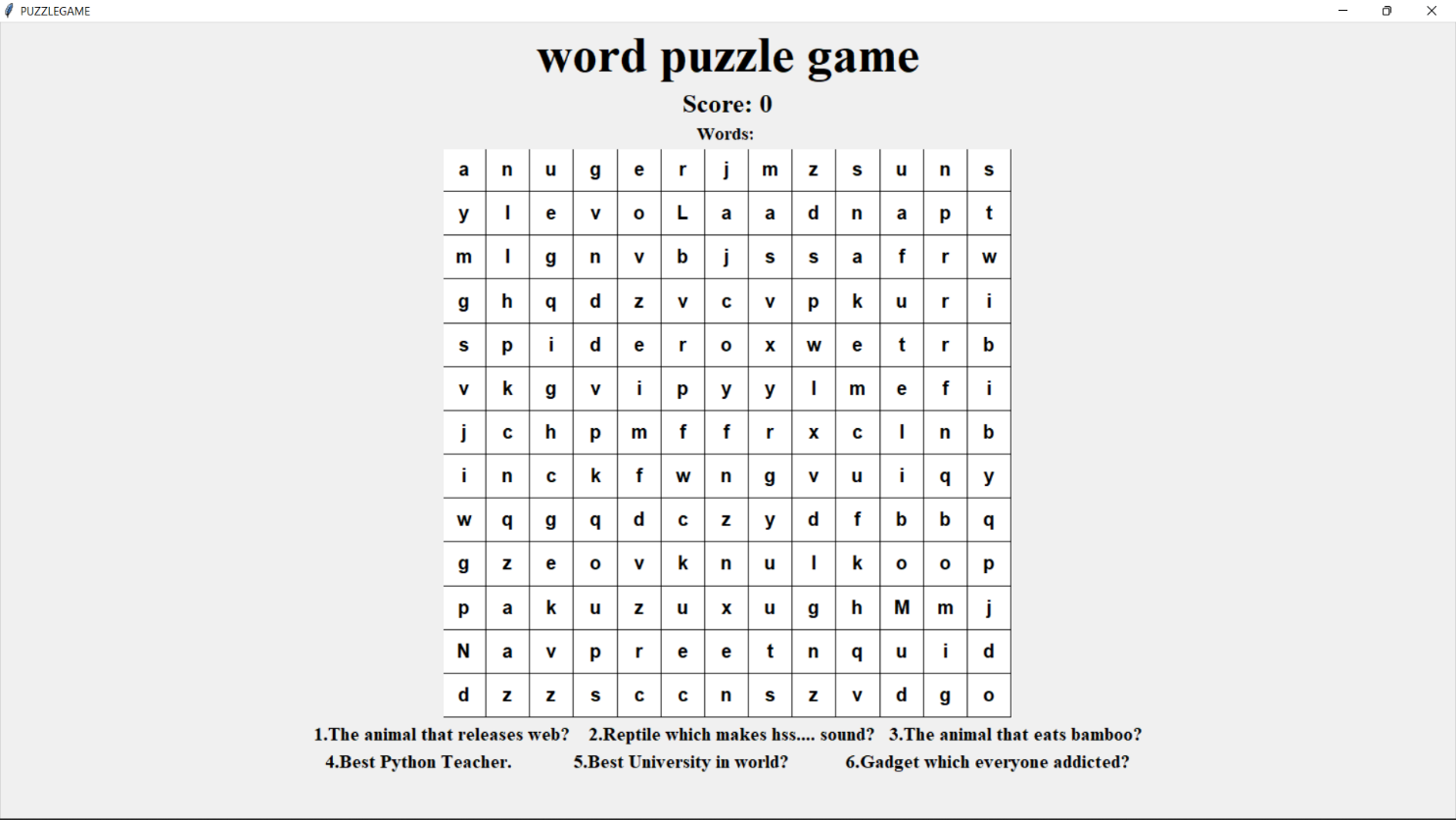
l2=tkinter.Label(root,text="4.Best Python Teacher. 5.Best University in world? 6.Gadget which everyone addicted?")

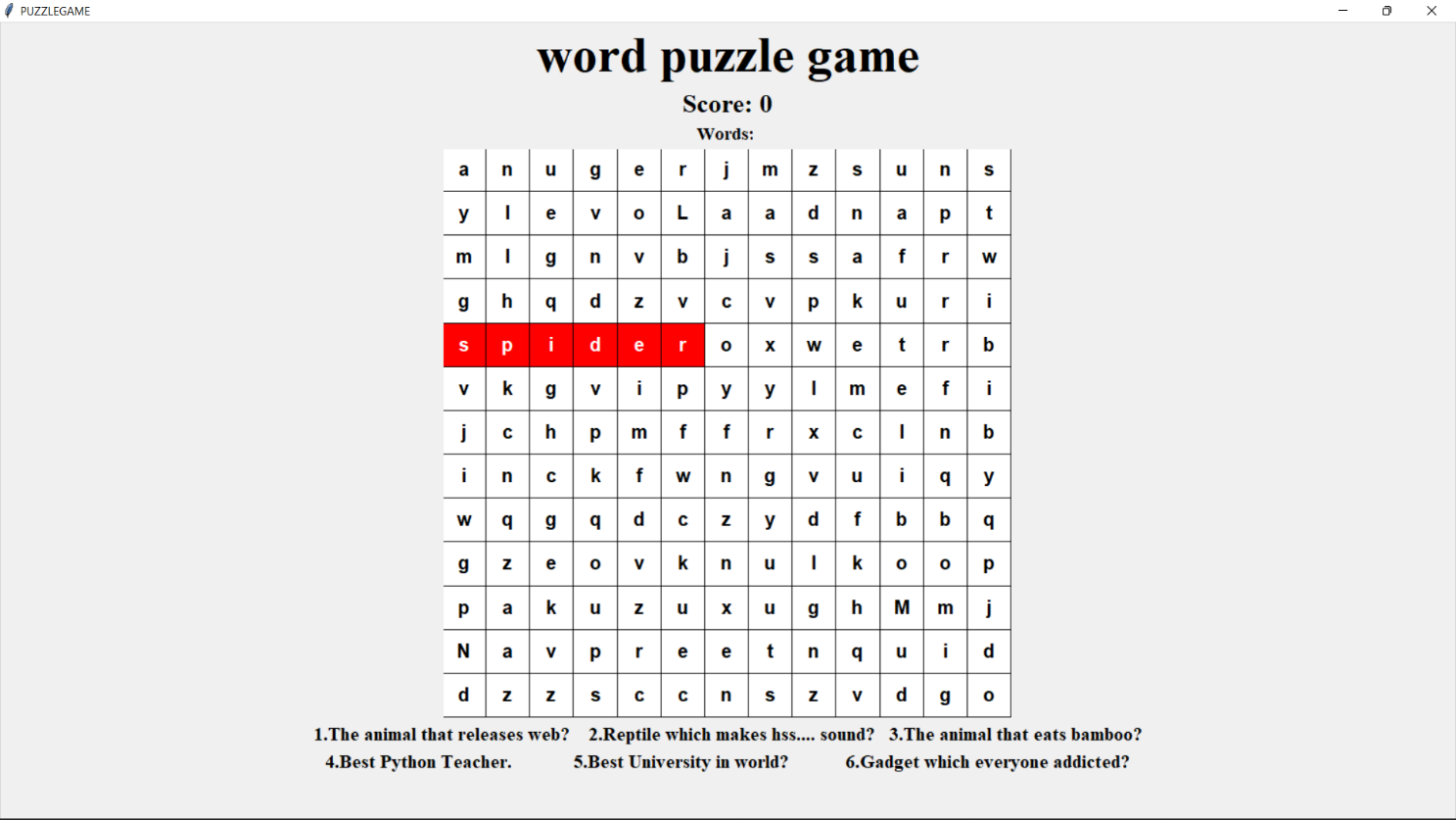
l2.configure(font=("Times New Roman",15,"bold"))

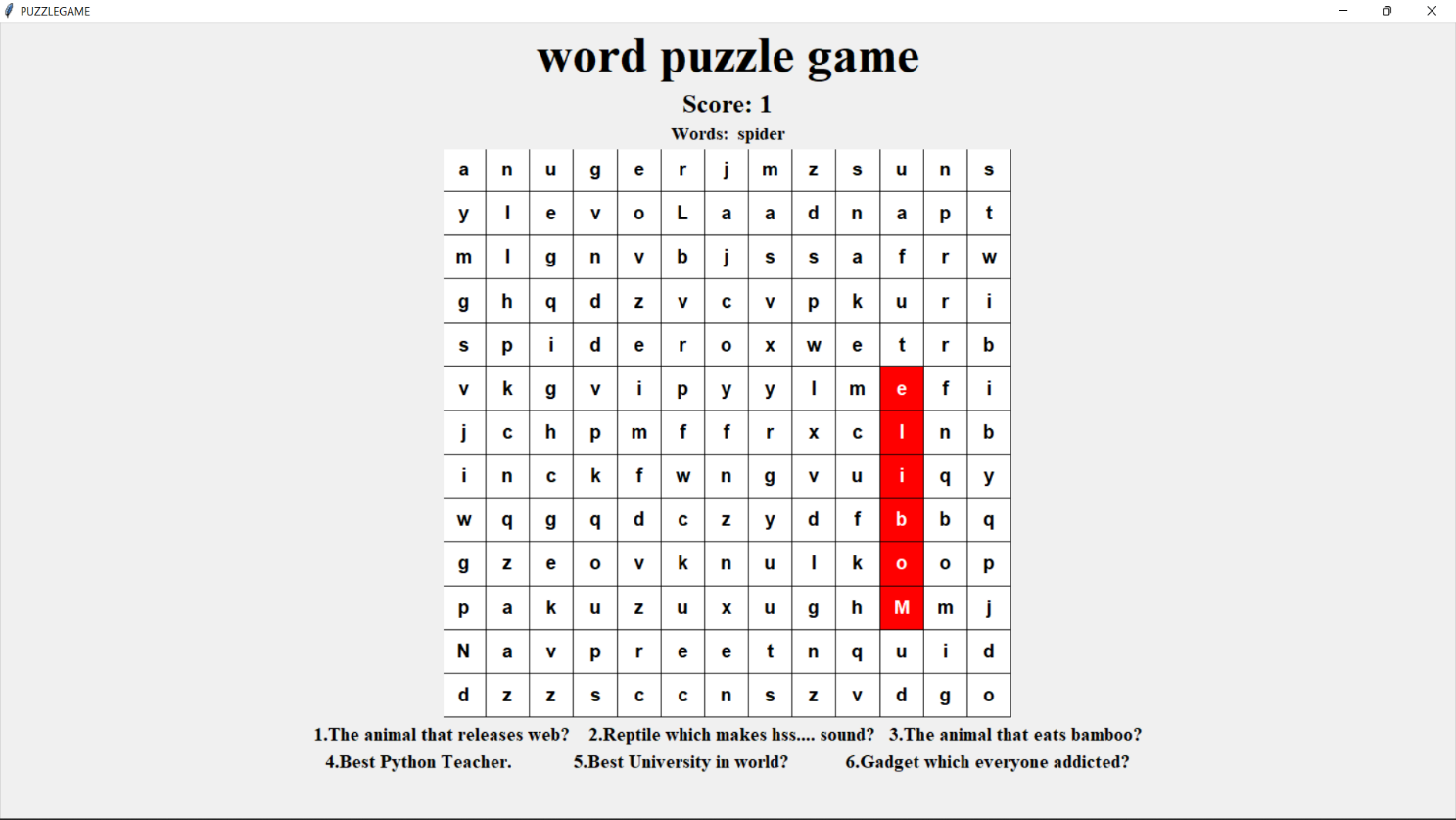
l2.pack()

root.mainloop()

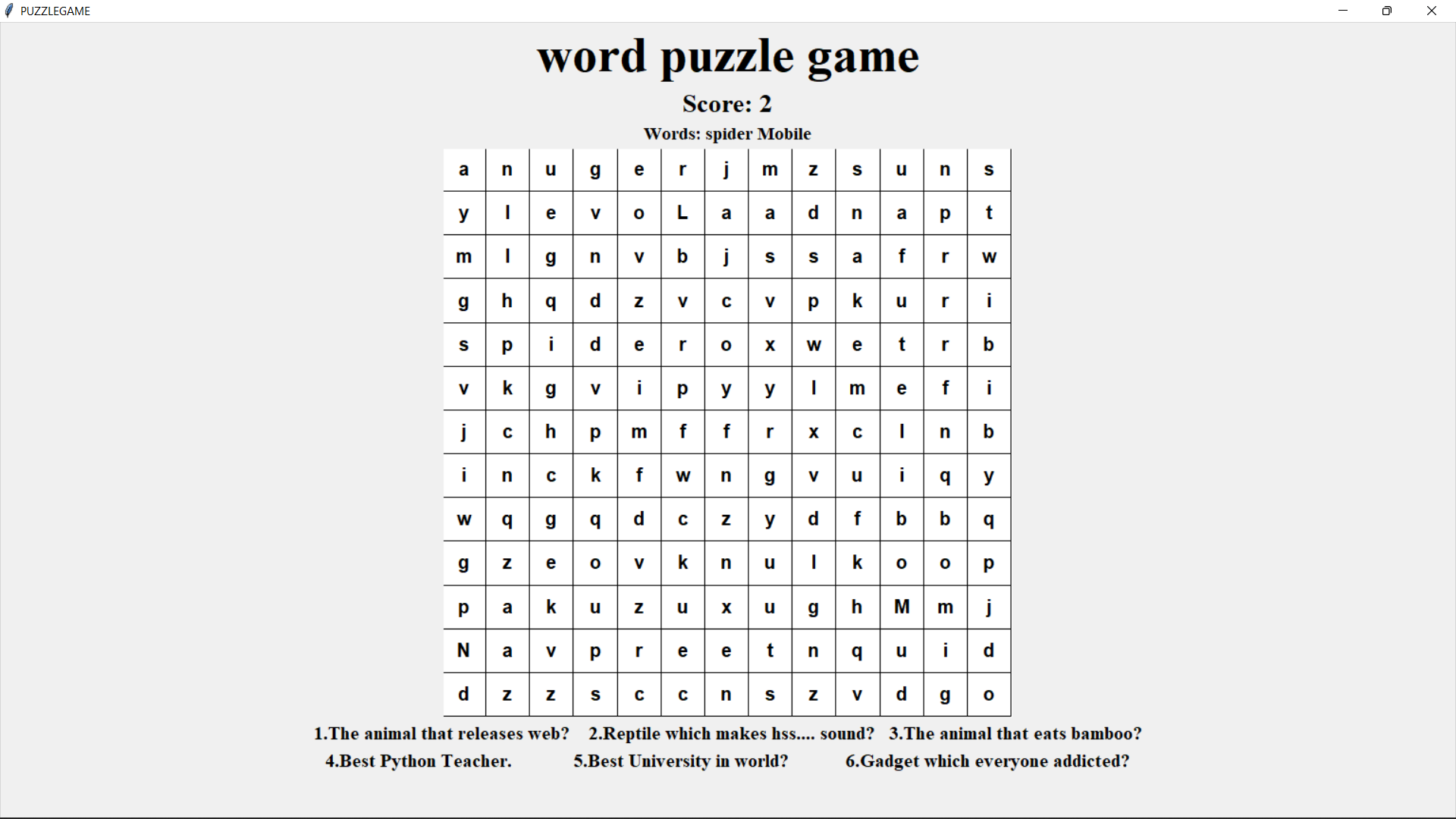
**RESULT SCREENSHOT:**







**Score and Words selected will be shown in the top.**



**ROLES AND RESPONSIBILITIES:**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of the Participants | Registration Number | Role of the Person | Responsibility of the Person |
| Pentakota Jaswanth | 12111901(B52) | Backend Development | Coding |
| D Praveen | 12115126 (A03) | Frontend Development | Requirements Analysis |
| Shibu Prakash M | 12116263 (A28) | Designing | Testing |

**GANTT CHART:**

Progress of work schedule in terms of Gantt Chart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Start Date** | **End Date** | **Duration (Days)** | **Days Complete** | **Precent complete** |
| Requirement Analysis | 11/09/22 | 13/09/22 | 3 | 3 | 100% |
| Designing | 22/09/22 | 25/09/22 | 4 | 4 | 100% |
| Coding | 26/09/22 | 20/10/22 | 20 | 20 | 100% |
| Testing | 26/09/22 | 31/10/22 | 2 | 2 | 100% |

**CONCLUSION**

Puzzle word games are an incredible mental exercise and provide us with engaging and thought-provoking gameplay. It is great for the development of a child and increases memory retention and improves cognitive ability. It is one of the best ways to keep your child occupied while benefiting his development. Puzzle game development is booming right now. Presently, it is a promising market and a great investment platform.

**REFFERENCE:**

• The Art of Game Design: A Book of Lenses. Jesse Schnell.

• On Game Design: Andrew Rollings and Ernest Adams.

• Video Game Explosion: A History from PONG to PlayStation (R) and beyond.

• Wikipedia: Dates of games not found in Video Game Explosion were taken from their entries in Wikipedia.