WORD SEARCH GAME

* ABSTRACT

This paper describes a process for inserting a list of random lettters.The output produce Two levels .In first level three stages and in stage 1, it will shows animal letters if we give 5 correct letters than it display we won.if failed to find all animal names it will show failed to find all.

In second stage birds name we have to find and like first stage all process is same.

And in third stage we have to find places name.if we got all 5 places we won the game otherwise it display failed to find.

In level second,here again two stages

Can u guess-in this level again five questions will display randomly we shoud give all answers otherwise it will shown fail to give answers.

And last one stage character fill-some character are display some dashes will be here you have to fill and type the correct word seven words are here.

* INTRODUCTION

A word search matrix has a number of potential;it can be both educational as well as entertaining.the primary goal in designing a program code to produce such a matrix is it be flexible enough to handle word list,and as well as skills levels.a word search ,mystery word puzzle is a word game.the objectivemof this puzzle is to find and mark all the word hidden inside the matrics .the words may be placed horizontally,vertically or diognally.its intresting but also mind storming game.

* LITERATURE SURVEY
* EXISTING SYSTEM AND PROPOSED SYSTEM
* EXISTING SYSTEM

In this system we have to perform a program which is based on searching.word search game in this we have to perform a game which works on random character matrix user should be knowing the method for each type of stages how to play.it not only for educational purpose but entertaining also.

If you are playing this you shoud know how to find letters.because some time it is very difficult to found words it has difficult stages hence brain storming.here somany products are in market we have too many options we can play it online offline also

* PROPOSED SYSTEM

In this system,as we told that a huge number of similar products we can easily download and play.but in this we modified it.not only you can play the word searching puzzels but here we included some new intresting effects.two levels are here as we know.

In the second level two different stages included

In first stage randomly 5 questions displays user should give answers of all five.

In second stage,character it will display some word with missing characters user should fill in th blanks.it is different to other product or game .

* **TOOLS AND TECHNOLOGIES USED**
* **TURBO C**

Turbo C was an integrated development environment (IDE) for programming in the C language. It was developed by Borland and first introduced in 1987. At the time, Turbo C was known for its compact size, comprehensive manual, fast compile speed and low price. It had many similarities to an earlier Borland product, Turbo Pascal, such as an IDE, a low price and a fast compiler, but was not as successful because of competition in the C compiler market.

* **Turbo C Features**
* Inline assembly with full access to the C language symbolic structures and names -- This allowed programmers to write some assembly language codes right into their programs without the need for a separate assembler.

Support for all memory models -- This had to do with the segmented memory architecture used by 16-bit processors of that era, where each segment was limited to 64 kilobytes (Kb). The models were called tiny, small, medium, large and huge, which determined the size of the data used by a program, as well as the size of the program itself. For example, with the tiny model, both the data and the program must fit within a single 64-Kb segment. In the small model, the data and the program each used a different 64-Kb segment. So in order to create a program larger than 64 Kb or one that manipulates data larger than 64 Kb, the medium, large and huge memory models had to be used. In contrast, 32-bit processors used a flat memory model and not have this limitation

* **HARDWARE & SOFTWARE REQUIREMENTS**
* **SOFTWARE CONFIGURATION**

Platform : Windows 7

IDI : Turbo C

* **HARDWARE CONFIGURATION**

System Type : INTEL

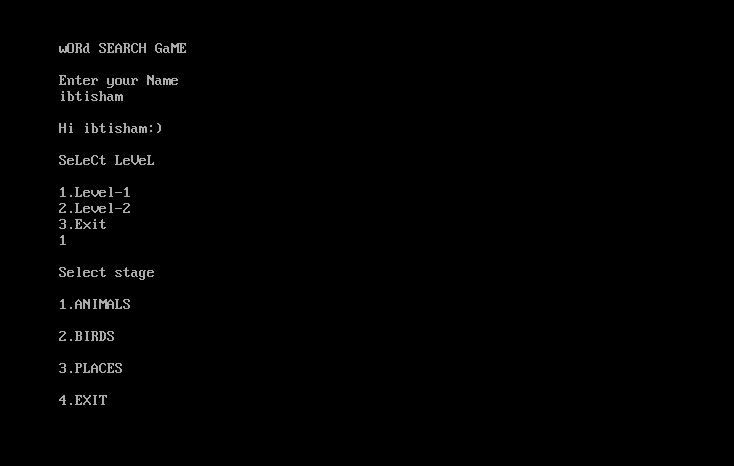
Processor : Pentium 4

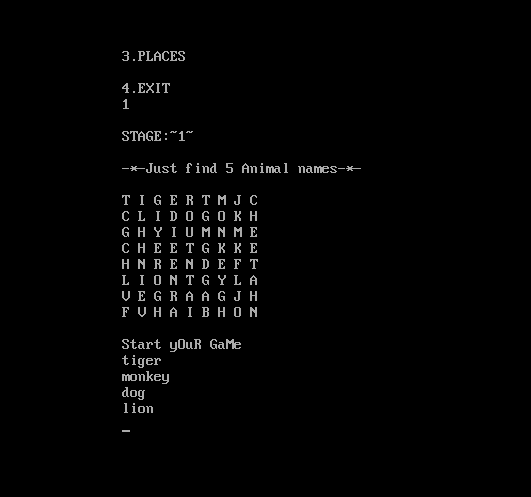
Processor Speed : 2.8 GHZ

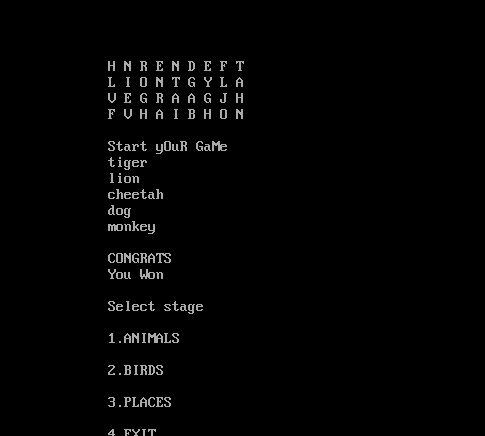
Hard Disk : 40 GB

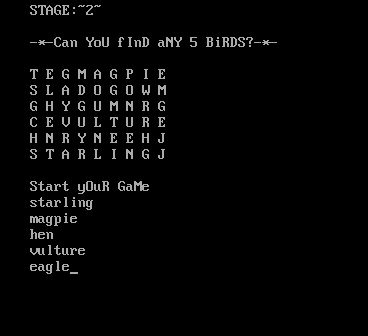
Memory Size : 128 MB

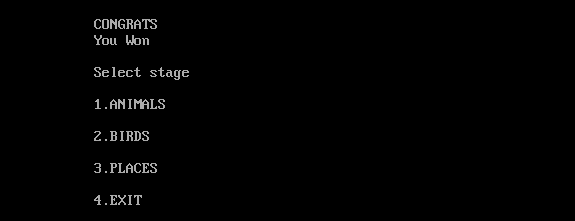
* IMPLEMENTATION

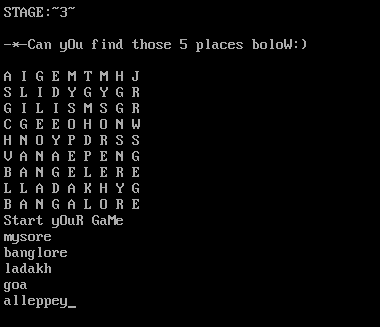


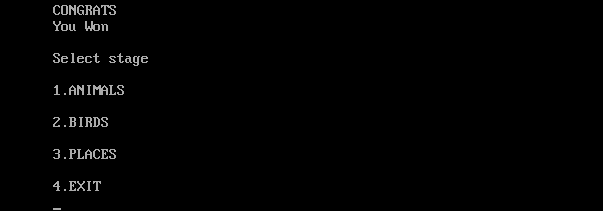


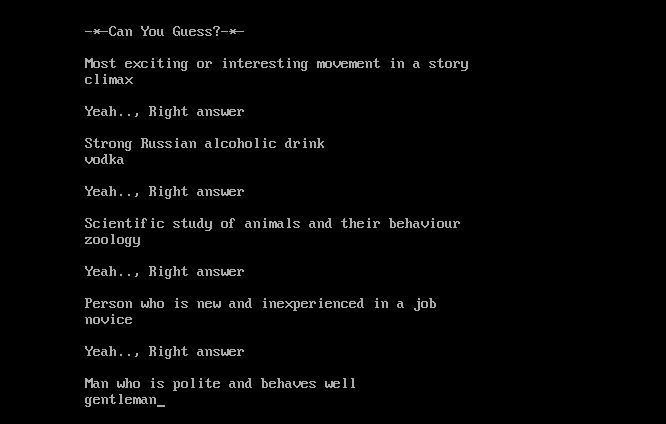


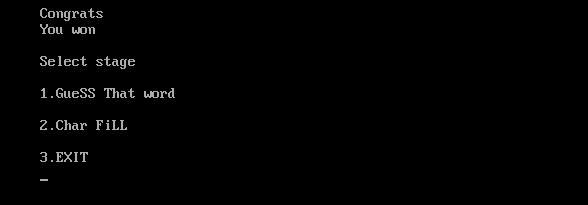


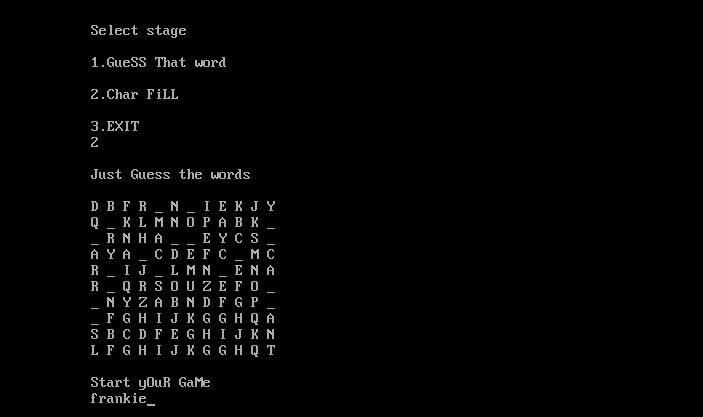


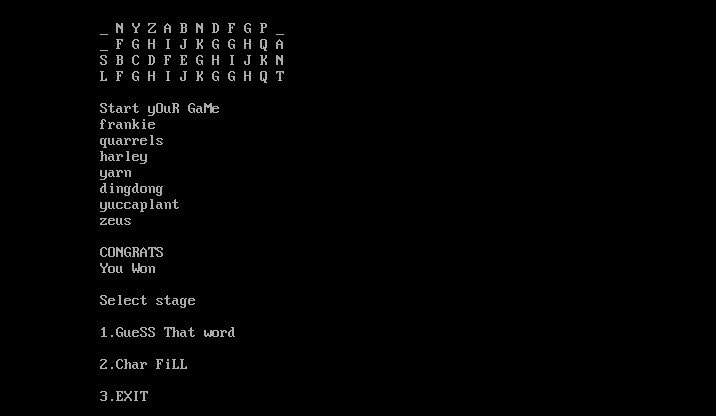












* SAMPLE CODING

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

#include<ctype.h>

#define MAX 5

#define LMAX 7

#define SIZE 50

char name[MAX][SIZE];

int i,j;

int e1,e2,e3,e4,e5,e6,e7;

void Level1()

{

int ch,n,count=0;

while(1)

{

printf("\nSelect stage\n");

printf("\n1.ANIMALS\n");

printf("\n2.BIRDS\n");

printf("\n3.PLACES\n");

printf("\n4.EXIT\n");

scanf("%d",&ch);

switch(ch)

{

case 1: {

printf("\nSTAGE:~1~\n");

printf("\n-\*-Just find 5 Animal names-\*-\n");

printf("\nT I G E R T M J C");

printf("\nC L I D O G O K H");

printf("\nG H Y I U M N M E");

printf("\nC H E E T G K K E");

printf("\nH N R E N D E F T");

printf("\nL I O N T G Y L A");

printf("\nV E G R A A G J H");

printf("\nF V H A I B H O N\n");

printf("\nStart yOuR GaMe\n");

for(i=0;i<MAX;i++)

{

scanf("%s",name[i]);

}

for(i=0;i<MAX;i++)

{

for(j=0;j<=strlen(name[i]);j++)

{

if(name[i][j]>=65 && name[i][j]<=90)

{

name[i][j]=name[i][j]+32;

}

}

}

e1=0;

e2=0;

e3=0;

e4=0;

e5=0;

count=0;

for(i=0;i<MAX;i++)

{

if((strcmp(name[i],"tiger")==0) && e1==0)

{

e1=1;

count+=1;

}

else if((strcmp(name[i],"cheetah")==0) &&e2==0)

{

e2=1;

count+=1;

}

else if((strcmp(name[i],"dog")==0) &&

e3==0)

{

e3=1;

count+=1;

}

else if((strcmp(name[i],"lion")==0) && e4==0)

{

e4=1;

count+=1;

}

else if((strcmp(name[i],"monkey")==0) &&

e5==0)

{

e5=1;

count+=1;

}

}

if(count==5)

{

printf("\nCONGRATS \nYou Won\n");

}

else

{

printf("\nfound only %d\n",count);

printf("\nFailed to find %d more animal names\n",5-count);

}

}

break;

case 2:{

printf("\nSTAGE:~2~\n");

printf("\n-\*-Can YoU fInD aNY 5 BiRDS?-\*-\n");

printf("\nT E G M A G P I E");

printf("\nS L A D O G O W M");

printf("\nG H Y G U M N R G");

printf("\nC E V U L T U R E");

printf("\nH N R Y N E E H J");

printf("\nS T A R L I N G J\n");

printf("\nStart yOuR GaMe\n");

for(i=0;i<MAX;i++)

{

scanf("%s",name[i]);

}

for(i=0;i<MAX;i++)

{

for(j=0;j<=strlen(name[i]);j++)

{

if(name[i][j]>=65 && name[i][j]<=90)

{

name[i][j]=name[i][j]+32;

}

}

}

e1=0;

e2=0;

e3=0;

e4=0;

e5=0;

count=0;

for(i=0;i<MAX;i++)

{

if((strcmp(name[i],"eagle")==0) && e1==0)

{

e1=1;

count+=1;

}

else if((strcmp(name[i],"hen")==0) && e2==0)

{

e2=1;

count+=1;

}

else if((strcmp(name[i],"vulture")==0) &&

e3==0)

{

e3=1;

count+=1;

}

else if((strcmp(name[i],"magpie")==0) && e4==0)

{

e4=1;

count+=1;

}

else if((strcmp(name[i],"starling")==0) &&

e5==0)

{

e5=1;

count+=1;

}

}

if(count==5)

{

printf("\nCONGRATS \nYou Won\n");

}

else

{

printf("\nfound only %d\n",count);

printf("\nFailed to find %d more birds names\n",5-count);

}

}

break;

case 3:{

printf("\nSTAGE:~3~\n");

printf("\n-\*-Can yOu find those 5 places boloW:) \n");

printf("\nA I G E M T M H J");

printf("\nS L I D Y G Y G R");

printf("\nG I L I S M S G R");

printf("\nC G E E O H O N W");

printf("\nH N O Y P D R S S");

printf("\nV A N A E P E N G");

printf("\nB A N G E L E R E");

printf("\nL L A D A K H Y G");

printf("\nB A N G A L O R E");

printf("\nStart yOuR GaMe\n");

for(i=0;i<MAX;i++)

{

scanf("%s",name[i]);

}

for(i=0;i<MAX;i++)

{

for(j=0;j<=strlen(name[i]);j++)

{

if(name[i][j]>=65 && name[i][j]<=90)

{

name[i][j]=name[i][j]+32;

}

}

}

e1=0;

e2=0;

e3=0;

e4=0;

e5=0;

count=0;

for(i=0;i<MAX;i++)

{

if((strcmp(name[i],"bangalore")==0) && e1==0)

{

e1=1;

count+=1;

}

else if((strcmp(name[i],"ladakh")==0) &&

e2==0)

{

e2=1;

count+=1;

}

else if((strcmp(name[i],"goa")==0) && e3==0)

{

e3=1;

count+=1;

}

else if((strcmp(name[i],"mysore")==0) &&

e4==0)

{

e4=1;

count+=1;

}

else if((strcmp(name[i],"alleppey")==0) && e5==0)

{

e5=1;

count+=1;

}

}

if(count==5)

{

printf("\nCONGRATS \nYou Won\n");

}

else

{

printf("\nfound only %d\n",count);

printf("\nFailed to find %d more places

names\n",5-count);

}

}

break;

case 4:exit(0);

default:printf("\nINVALID CHOICE\n");

}

}

}

void Level2()

{

int ch,n,count=0;

char str[SIZE];

while(1)

{

printf("\nSelect stage\n");

printf("\n1.GueSS That word\n");

printf("\n2.Char FiLL\n");

printf("\n3.EXIT\n");

scanf("%d",&ch);

switch(ch)

{

case 1: {

printf("\nSTAGE:~2~\n");

printf("\n-\*-Can You Guess?-\*-\n");

printf("\nMost exciting or interesting movement in a

story\n");

scanf("%s",str);

if(strcmpi(str,"climax")==0)

{

printf("\nYeah.., Right answer\n");

count+=1;

}

else

{

printf("\nOoooo No!!!! Wrong anwer\n");

}

printf("\nStrong Russian alcoholic drink\n");

scanf("%s",str);

if(strcmpi(str,"vodka")==0)

{

printf("\nYeah.., Right answer\n");

count+=1;

}

else

{

printf("\nOoooo No!!!! Wrong anwer\n");

}

printf("\nScientific study of animals and their

behaviour\n");

scanf("%s",str);

if(strcmpi(str,"zoology")==0)

{

printf("\nYeah.., Right answer\n");

count+=1;

}

else

{

printf("\nOoooo No!!!! Wrong anwer\n");

}

printf("\nPerson who is new and inexperienced in a job\n");

scanf("%s",str);

if(strcmpi(str,"novice")==0)

{

printf("\nYeah.., Right answer\n");

count+=1;

}

else

{

printf("\nOoooo No!!!! Wrong anwer\n");

}

printf("\nMan who is polite and behaves well\n");

scanf("%s",str);

if(strcmpi(str,"gentleman")==0)

{

printf("\nYeah.., Right answer\n");

count+=1;

}

else

{

printf("\nOoooo No!!!! Wrong anwer\n");

}

if(count==5)

{

printf("\nCongrats \nYou won\n");

}

else

{

printf("\nfound only %d\n",count);

printf("\nFailed to answer %d more

questions\n",5-count);

}

}

break;

case 2:{

printf("\nJust Guess the words\n");

printf("\nD B F R \_ N \_ I E K J Y");

printf("\nQ \_ K L M N O P A B K \_");

printf("\n\_ R N H A \_ \_ E Y C S \_");

printf("\nA Y A \_ C D E F C \_ M C ");

printf("\nR \_ I J \_ L M N \_ E N A");

printf("\nR \_ Q R S O U Z E F O \_");

printf("\n\_ N Y Z A B N D F G P \_");

printf("\n\_ F G H I J K G G H Q A");

printf("\nS B C D F E G H I J K N");

printf("\nL F G H I J K G G H Q T\n");

printf("\nStart yOuR GaMe\n");

for(i=0;i<LMAX;i++)

{

scanf("%s",name[i]);

}

e1=0;

e2=0;

e3=0;

e4=0;

e5=0;

e6=0;

e7=0;

count=0;

for(i=0;i<LMAX;i++)

{

if((strcmp(name[i],"dingdong")==0) && e1==0)

{

e1=1;

count+=1;

}

else if((strcmp(name[i],"frankie")==0) &&

e2==0)

{

e2=1;

count+=1;

}

else if((strcmp(name[i],"yarn")==0) && e3==0)

{

e3=1;

count+=1;

}

else if((strcmp(name[i],"zeus")==0) && e4==0)

{

e4=1;

count+=1;

}

else if((strcmp(name[i],"quarrels")==0) && e5==0)

{

e5=1;

count+=1;

}

else if((strcmp(name[i],"harley")==0) && e6==0)

{

e6=1;

count+=1;

}

else if((strcmp(name[i],"yuccaplant")==0) && e7==0)

{

e7=1;

count+=1;

}

}

if(count==7)

{

printf("\nCONGRATS \nYou Won\n");

}

else

{

printf("\nfound only %d\n",count);

printf("\nFailed to find %d more\n",7

count);

}

}

break;

case 3:exit(0);

default:printf("\nInvalid Choice\n");

}

}

}

void main()

{

char str[10];

int Lev;

clrscr();

printf("\nwORd SEARCH GaME\n");

printf("\nEnter your Name\n");

scanf("%s",str);

printf("\nHi %s:)\n",str);

printf("\nSeLeCt LeVeL\n");

printf("\n1.Level-1\n2.Level-2\n3.Exit\n");

scanf("%d",&Lev);

switch(Lev)

{

case 1:Level1();

break;

case 2:Level2();

break;

case 3:exit(0);

default:printf("\nInvalid Choice\n");

}

getch();

}

* **SOFTWARE TESTING**
* **Software Testing**

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of A software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

* **Verification**

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

* **Validation**

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

* **Basics of software testing**

There are two basics of software testing: black box testing and white box testing.

* **Black box Testing**

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

* **White box Testing**

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing.

Black box testing is often used for validation and white box testing is often used for verification.

* CONCLUSION

The finding of this study shows that games can afford a valuable technique in language classroom for students at intermidiate level and hence can be used to faciliate the processor of vocabulary learning .

The existing nature of playing games can faciliate the vocabulary learning process.language games are suitable evaluatuion tools in the hands of teachers.they quickly reveal the depth of students understanding and reinforce their previous knoledge.the teacher needs to argue that , through games,learners are given opportunities to meet and explore new vocabulary without direct teacher assistance.