DSA Project Synopsis on

**2048 Game in C**

Submitted to Manipal University, Jaipur

Towards the partial fulfillment for the Award of the Degree of

**BACHELORS OF TECHNOLOGY**

In Computer and Communication Engineering (16pt.)

2023-2024

By

Aayush Bindal,Atharva Deshmukh

229311202,229311049



Under the guidance of

Dr. Abhay Sharma

**Department of Computer and Communication Engineering**

**Manipal University Jaipur**

**Jaipur, Rajasthan**

**Introduction**

Welcome to the exciting world of the "2048" game project in C! In this project, we dive into the realm of coding to recreate a popular and addictive puzzle game. "2048" is all about merging tiles with the same numbers to reach the elusive number 2048.

Our project aims to introduce you to the fundamentals of programming in the C language while building this engaging game. You'll learn about data structures, algorithms, and user interfaces in a fun and interactive way.

With simple words and straightforward code, we'll guide you through the process of creating the game's logic, handling user input, and displaying the game board. As you progress through this project, you'll gain valuable coding skills and a deeper understanding of how games like "2048" work under the hood.

So, let's embark on this coding journey, where you'll not only have a blast playing "2048" but also develop your C programming skills along the way. Get ready to code, merge, and conquer!

**Motivation**

To learn more about c language and data structure through fun means by implementing them in a game that can be played by all.

**Problem Statement**

To create a 2048 game in c and learn about the various functions and methods used for making the same.

**Methodology**

►Libraries Used:

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

#include <time.h>

►Datatype Used:

int arr[4][4]={0},c[4],temp=0,len=0,score=0,highscore=0,count=0,ch=0;//COUNT WILL COUNT THE NO OF STEPS

►Function to print: print()

void print(){

        int i,j,k,len1;

        printf("\n\t\t\t\t\t===============2048==============\n");

          printf("\t\t\t\t\tYOUR SCORE=%d\n\t\t\t\t\t",score);

          if(score<highscore){

              printf("HIGH SCORE=%d\t\t\t\t\t\n",highscore);

          }

          else{

              highscore=score;

              printf("HIGH SCORE=%d\t\t\t\t\t\n",highscore);

          }

    printf("\t\t\t\t\t---------------------------------\n");

    for (i=0;i<4;i++){

        for(j=0;j<4;j++){

            if(j==0){

                printf("\t\t\t\t\t|");

            }

            if(arr[i][j]!=0){

                len1=findlen(arr[i][j]);

                //printf("%d:",len);

                for(k=0;k<4-len;k++){

                    printf(" ");

                }

                printf("%d",arr[i][j]);

                for(k=0;k<4-len;k++){

                    printf(" ");

                }

                for(k=0;k<len-1;k++){

                    printf(" ");

                }

                printf("|");

            }

            else

            {

                for(k=0;k<8-2\*len-1;k++){

                printf(" ");}printf("|");

            }

            len=0;

        }

        if(i!=3){

        printf("\n");

    printf("\t\t\t\t\t---------------------------------\n");

        }

    }

    printf("\n\t\t\t\t\t---------------------------------\n");

     printf("\t\t\t\t\tPREV-> P\t\t\t\t\t\n");

     printf("\t\t\t\t\tRESTART-> R\t\t\t\t\t\n");

     printf("\t\t\t\t\tEXIT-> U\t\t\t\t\t\n");

    printf("\t\t\t\t\tENTER YOUR CHOISE -> W,S,A,D\n\t\t\t\t\t");

}

►Function to move value from one block to another: movevalue()

void movevalue(int k){

    int i;

    if(k==3 ){

        return;

    }

    for(i=k;i<4;i++){

        if(c[i]!=0){

            movevalue(i+1);

            for(;i<4;i++){

                if(c[i+1]!=0){

                    break;

                }

                    c[i+1]=c[i];

                    c[i]=0;

            }

        }

    }

}

►Function to find length: findlen()

int findlen(int n){

        if(n==0){

            return len;

        }

        else{

            len++;

            findlen(n/10);

        }

}

►Function to add a random number at a random block: addrandno()

void addrandomno(){

    int no;

    srand(time(NULL));

    int i,j; //RANDOM INDEX

    do{

    i=(rand ())%(MAXRANDOMVALUE+1);

    j=(rand ())%(MAXRANDOMVALUE+1);

    }

    while(arr[i][j]!=0);

   //printf("%d %d",i,j);

    no=2\*((rand ()%10)+1);

    if(no==3 || no==2){

        arr[i][j]=4;

    }

    else{

        arr[i][j]=2;

    }

}

►Function to update the highscore: rupdate()

void rupdate(){

    int i,j;

    for(i=3;i>0;i--){

        if(c[i]==c[i-1] ){

            c[i]+=c[i-1];

            score=score+c[i];

            if(score>highscore){

            }

            temp=1;

            c[i-1]=0;

        }

        else if(c[i-1]==0 && c[i]!=0 ){

            c[i-1]=c[i];

            c[i]=0;

            temp=1;

        }

        else if(c[i]==0){

            temp=1;

        }

    }

        movevalue(0);

}

►Function to store and create the previous move: createprev()

void createprev(int \*\*\*p){

      int i,j,k;

      FILE \*ptr;

      ptr=fopen("hstr.txt","a");

      fprintf(ptr,"%d ",score);

      fclose(ptr);

    if(count==MAXPREV+1){

        for(i=MAXPREV;i>0;i--){

            for(j=0;j<4;j++){

                for(k=0;k<4;k++){

                        p[i][j][k]=p[i-1][j][k];

                }

            }

        }

        count=MAXPREV;

    }

    for(i=0;i<4;i++){

        for(j=0;j<4;j++){

            p[MAXPREV-count][i][j]=arr[i][j];

        }

    }

}

►Function to update the array to the previous moveset: updatearrtoprev()

void updatearrtoprev(int \*\*\*p){

    int data,i,j;

    if(count==0){

        printf("\n\t\t\t\t\t\*\*\*\*\*\*FURTHER MORE PREV NOT POSSIBLE\*\*\*\*\*\*\*\*");

        return;

    }

    FILE \*ptr=fopen("hstr.txt","r+");

        for(i=0;i<count;i++){

            fscanf(ptr,"%d ",&data);

        }

        score=data;

    for(i=0;i<4;i++){

        for(j=0;j<4;j++){

            arr[i][j]=p[MAXPREV-count][i][j];

        }

    }

    count--;

}

►Function to reset the game: resetgame()

void resetgame(){

            int i,j;

            for(i=0;i<4;i++){

                for(j=0;j<4;j++){

                    arr[i][j]=0;

                }

            }

            system("cls") ;

            score=0;

            addrandomno();

}

►Main function: main()

int main(){

    int i,j,k,m,n,same=0;

    char choise,s=-33,reschk;

    printf("\n\t\t\t\t\t===============2048==============\n");

    printf("\n\t\t\t\t\tWELCOME TO PUZZLE 2048");

    printf("\n\t\t\t\t\t> Made by: Atharva Deshmukh , Aayush Bindal");

    printf("\n\t\t\t\t\t> CONTROLS");

    printf("\n\t\t\t\t\t  FOR MOVES:- 'W','S','A','D'");

    printf("\n\t\t\t\t\t  GO BACKWARD:- 'P'");

    printf("\n\t\t\t\t\t  RESTAT THE GAME:- 'R'");

    printf("\n\t\t\t\t\t  EXIT:-'U'");

    printf("\n\n\t\t\t\t\tPRESS ANY KEY TO START THE GAME....\n\t\t\t\t\t");

    getch();

    system("cls") ;

    printf("\n\t\t\t\t\t===============2048==============\n");

    printf("\n\n\t\t\t\t\tLOADING...\n\n\t\t\t\t\t");

    for(int i=0,j;i<35;i++){

        printf("%c",s);

        j=i;

        if(i%2!=0 && i<20){

            sleep(1);

        }

    }

    sleep(1);

    system("cls") ;

    int \*\*\*p;

    p=(int \*\*\*)malloc(sizeof(int \*)\*(MAXPREV+1));

    for(int i=0;i<MAXPREV+1;i++){

        \*(p+i)=(int \*\*)malloc(sizeof(int \*)\*4);

    }

    for(int i=0;i<MAXPREV+1;i++){

        for(int j=0;j<4;j++){

            p[i][j]=(int \*)malloc(sizeof(int)\*4);

        }

    }

    createprev(p);

    FILE \*ptr;

    ptr=fopen("highscore.txt","r");

    fscanf(ptr,"%d",&highscore);

    fclose(ptr);

    ptr=fopen("hstr.txt","w");

    fclose(ptr);

    addrandomno();

    print();

    while(1){

            if(score>highscore){

            ptr=fopen("highscore.txt","w");

            fprintf(ptr,"%d",score);

            fclose(ptr);

            }

        choise=getch();

        if(choise=='D' || choise =='d'){

            count++;ch++;

              createprev(p);

            for(i=0;i<4;i++){

                for(j=0;j<4;j++){

                    c[j]=arr[i][j];

                }

                rupdate();

                for(k=0;k<4;k++){

                    arr[i][k]=c[k];

                }

            }

        }

        else if(choise=='a' || choise=='A'){

            count++;

            ch++;

            createprev(p);

            for(i=0;i<4;i++){

                for(j=3;j>=0;j--){

                    c[3-j]=arr[i][j];

                }

                rupdate();

                for(k=0;k<4;k++){

                    arr[i][3-k]=c[k];

                }

            }

        }

        else if(choise=='s' || choise =='S'){

            count++;ch++;

            createprev(p);

            for(i=0;i<4;i++){

                for(j=0;j<4;j++)

                {

                    c[j]=arr[j][i];

                }

                rupdate();

                for(k=0;k<4;k++)

                {

                    arr[k][i]=c[k];

                }

            }

        }

        else if(choise=='w' || choise=='W'){

            count++;ch++;

            createprev(p);

            for(i=0;i<4;i++){

                for(j=3;j>=0;j--){

                    c[3-j]=arr[j][i];

                }

                rupdate();

                for(k=0;k<4;k++){

                  arr[3-k][i]=c[k];

                }

            }

        }

        else if(choise=='p' || choise =='p'){

            updatearrtoprev(p);

            temp=8;

        }

        else if(choise=='R' || choise=='r'){

            count=0;

            resetgame();

            print();

            continue;

        }

        else if(choise=='u'||choise=='U'){

            exit(0);

        }

            if(temp==1){

                temp=0;

                system("cls") ;

                printf("\n%c\n",choise);

                addrandomno();

                print();

            }

            else if(temp==8){

                temp=0;

                print();

            }

            else{

                for(m=0;m<4;m++){

                    for(n=3;n>0;n--){

                        if(arr[m][n]==arr[m][n-1] || arr[m][n]==0 || arr[m][n-1]==0){

                                same=1;

                                break;

                        }

                        if(arr[n][m]==arr[n-1][m] || arr[m][n]==0 || arr[m][n-1]==0){

                            same=1;

                            break;

                        }

                    }

                    if(same==1)

                        break;

                }

                if(same==1){

                    printf("\n\t\t\t\t\t============INVALID KEY==========\n");

                    same=0;

                }

                else{

                    printf("\n\t\t\t\t\t=============GAME OVER============");

                    printf("\n\n\t\t\t\t\tWANT TO PLAY MORE?? Y/N??\n\t\t\t\t\t");

                    reschk=getch();

                    switch (reschk){

                        case 'Y':

                        case 'y':resetgame(); print();break;

                        case 'n':

                        case 'N':exit(0);

                    }

                    continue;

                }

            }

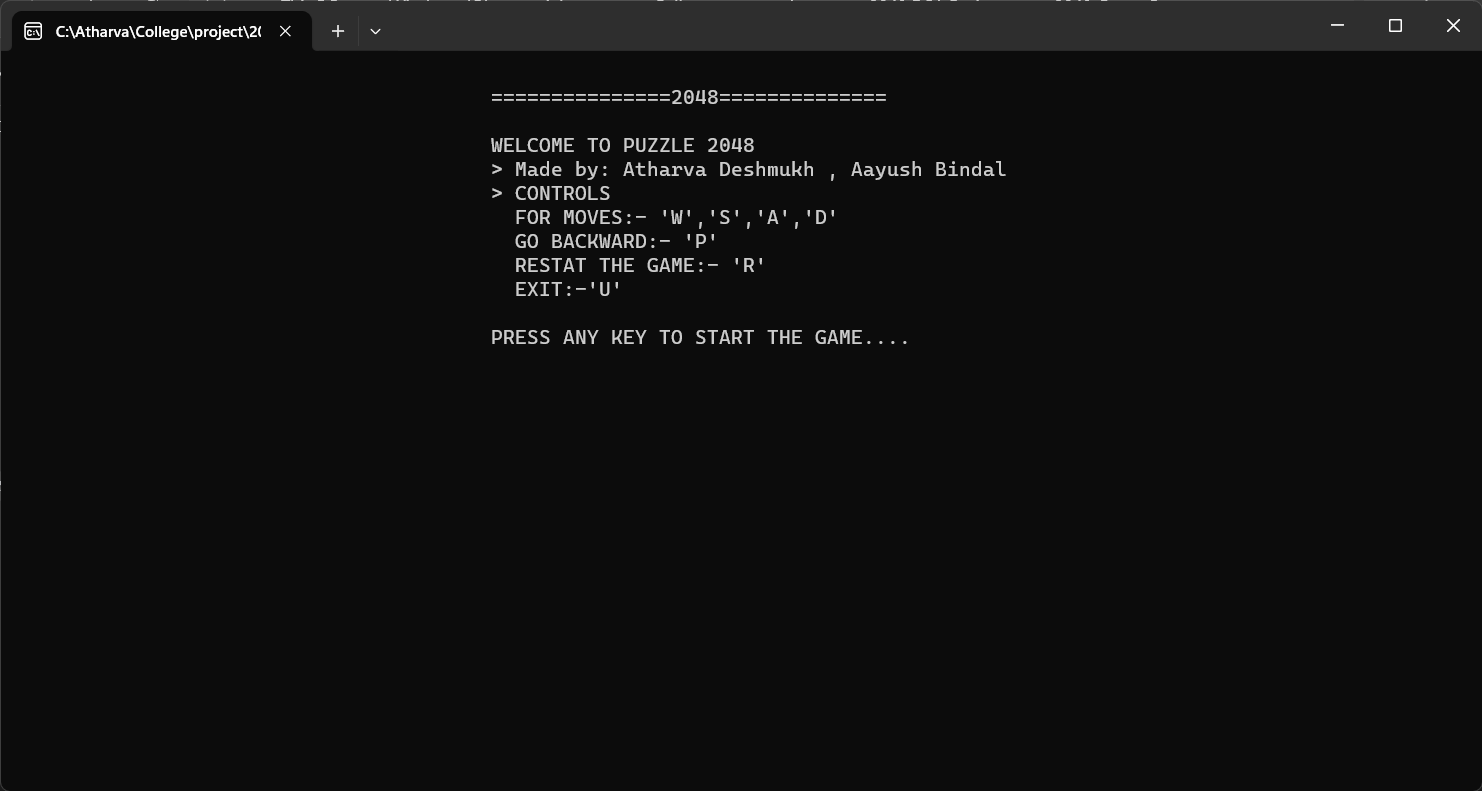
    }

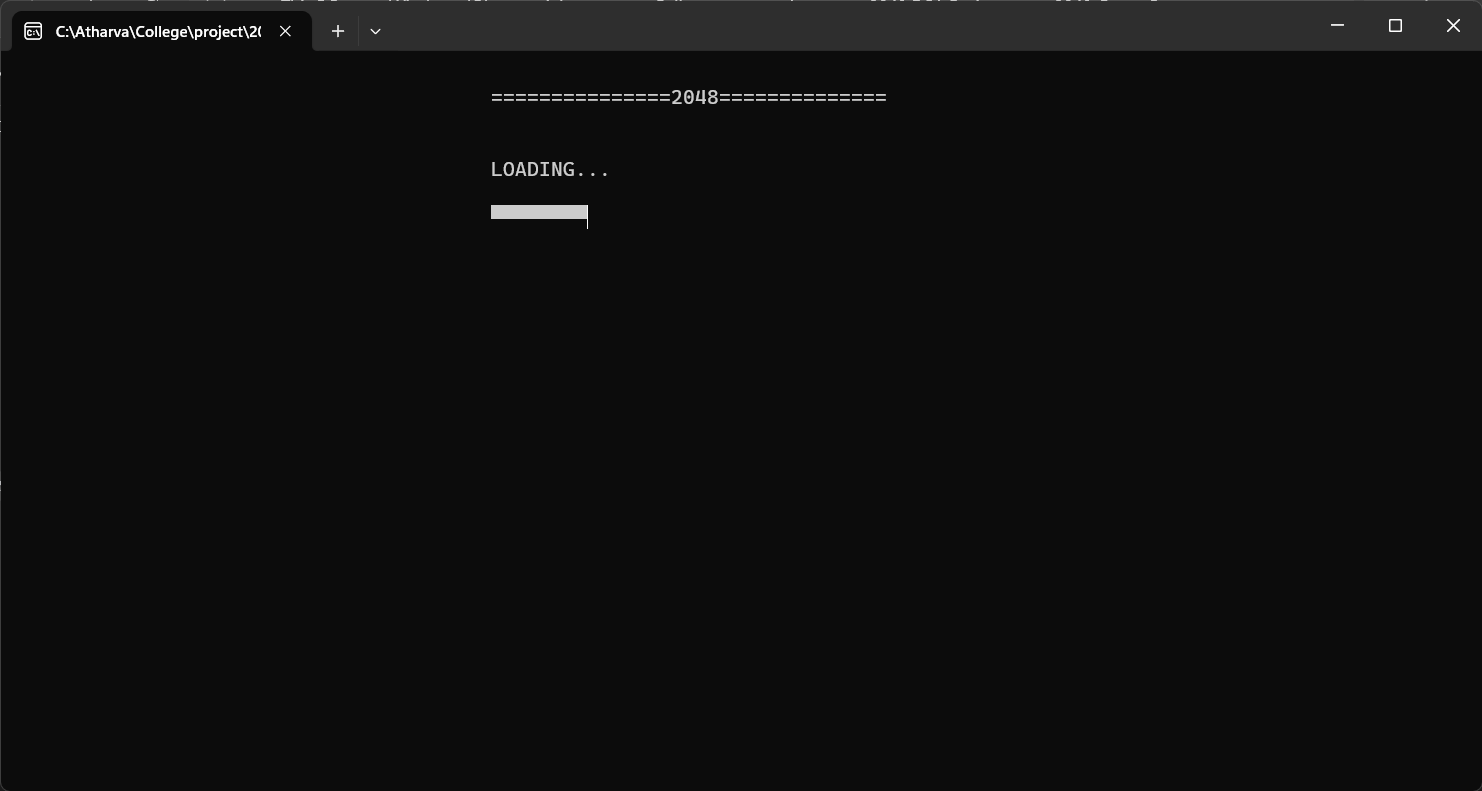
    return 0;

}

**Showcase**

►Starting Screen for game:

****

►****Loading Phase:

►Game Phase:

****

**Facilities**

* Libraries that need to be present before playing the game:

<stdio.h>,<stdlib.h>, <conio.h> ,<time.h>

* gcc compiler needs to be present if the code needs to be changed
* C redistributable
* VSCode/CodeBlocks
* Ming64/Msys2