

RUBIKS CUBE

CS101 PROJECT

PROJECT MEMBERS

1.NEIL gEORGE

2.RISHIKESH THAKRE

3.MAHESH SONI

4.HARISH MEENA

CONTENT

1.INTRODUCTION

2.BASIC OUTLINE

3.WORK COMPLETION

4.REFERENCES

1.INTRODUCTION

A rubiks cube is a 3dimensional puzzle game.

Rubiks cube contain 6 sides of different colours and each face contains 9 small boxes. Now the cube is shuffled randomly. For the puzzle to be solved all the 6 sides must be solved .

solving one side means making all 9 small boxes of same colour .A pivot mechanism enables each face to turn independently.

Here we are introducing a 3x3 rubiks cube game which can be solved by various combination of moves. We provide user to make several moves and solve the puzzle of Rubiks cube

2.BASIC OUTLINE

void rf(face *p,int n):this function takes an object of type face and an integer.It rotates a face which the pointer is pointing to as per the the integer passed to the function .

void swap(int*a,int*b,int*c,int*d,int n):this function 4 integers are passed by reference to it and other integer as a value it swaps the 4 values n times anticlockwise.

int*getelement(int i,int j): this function accepts two integers and stores it in variables i,j and returns the address of elements present in ith row and jth column of array to which the object of face points to.

Class rubicscube is defined which has the following variables and member functions

face*front

face*left

face*right

face*rear

face*top

face*bottom

these 6 pointers points to 6 faces of rubiks cube.

we define a constructor

public:rubiccube(): default constructor.

void rs(face*s,int n):this function accepts a reference of face and stores it in s.Also accepts an integer and stores it in n.It rotates the side corosponding to the face which the pointre points to and rotate it n times anticlockwise.

int num (string m): accepts string and returns an integers corosponding to the given string.

the pointer of the face and displays a name of face pointerr pointing to.

int num(face*s): Accepts the pointer of the face and returns an integer corresponding to the given face.

face*num(int n): Accepts an integer and returns the reference of the face corresponding to that integer.

int*L(face*s,int n),int*R(face*s,int n),int*S(face*s,int n): These functions help in rotation of a side by returning the corresponding address which would change during rotation

GRAPHICAL PART

We use simple cpp for the graphics.

these are the following functions we used:

void map_code_to_color(int code,int&r,int&g,int&b): ***This takes in code of color of a given square and returns the color as r,g,b by reference.***

void printcube(face*s,polygon s[3][3]):

This function accepts the pointer to a face and an array of polygons and prints a face of a cube.

WORK COMPLETION:

We divided the functions among our group members.

Neil George: graphical part was mostly done by him.

structure face ,int*getelement function ,class rubiks cube,

function void word.

Rishikesh Thakre:void rf

void swap ,void rs, function rubicscube ,

class rubicscube.

Mahesh Soni: int*L,int*R,int*S.

Harish Meena: int

num(face*s),face*num(int n) int num(face*s).

References: youtube, Abhiram Ranade,

also some internet references.

