Indian Institute of Information Technology, Allahabad

Demonstration of Sorting Algorithm

A BACHELOR'S GRAPHICS PROJECT

Submitted in fulfillment

Of the requirements for the completion of the ***5th*** *semester*

Of the

UNDER GRADUATE PROGRAM

In

INFORMATION TECHNOLOGY
(B. Tech in IT)
(Bachelor of technology in Information Technology)

Submitted by -

IWM2014006 (Surendra pal uikey)
BIM2014010 (Sanghpriya)

Under the Guidance of:

Dr. Pavan Chakraborty
And
Dr. Satish Singh
Associate Professor, Information Technology
IIIT-Allahabad

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD – 211012 (INDIA)

September, 2016

CANDIDATE’S DECLARATION

We hereby declare that the work presented in this project entitled “Demonstration of sorting Algorithm”, submitted in the fulfillment of the completion of the semester V of Bachelor of Technology (B. Tech) program, in Information Technology at Indian Institute of Information Technology, Allahabad, is an authentic record of our original work carried out under the guidance of **Dr. Pavan Chakraborty and Dr. Satish singh.** Due acknowledgements have been made in the text of the project to all other material used. This semester work was done in full compliance with the requirements and constraints of the prescribed curriculum.

Place: Allahabad	BIM2014010 (Sanghpriya)
Date: November 24, 2016	IWM2014006 (Surendra Pal)

CERTIFICATE FROM SUPERVISOR

We do hereby recommend that the graphics project report prepared under my supervision titled "Demonstration of sorting Algorithm" to be **accepted** in the complete fulfillment of the requirements of the completion of 5th semester Graphics Subject of Bachelor of Technology in Information Technology **for End Semester Examination.**

Date: November 17, 2016
Place: Allahabad

Dr. Pavan Chakraborty
Associate Professor, IIIT-A

Dr. Satish Singh
Assistance Professor, IIIT-A

Introduction

The Project is aimed at “Demonstration of Sorting Algorithm”.

We have used “bubble Sorting algorithm” and “insertion Sorting algo.”

Problem Solution and Methodology:

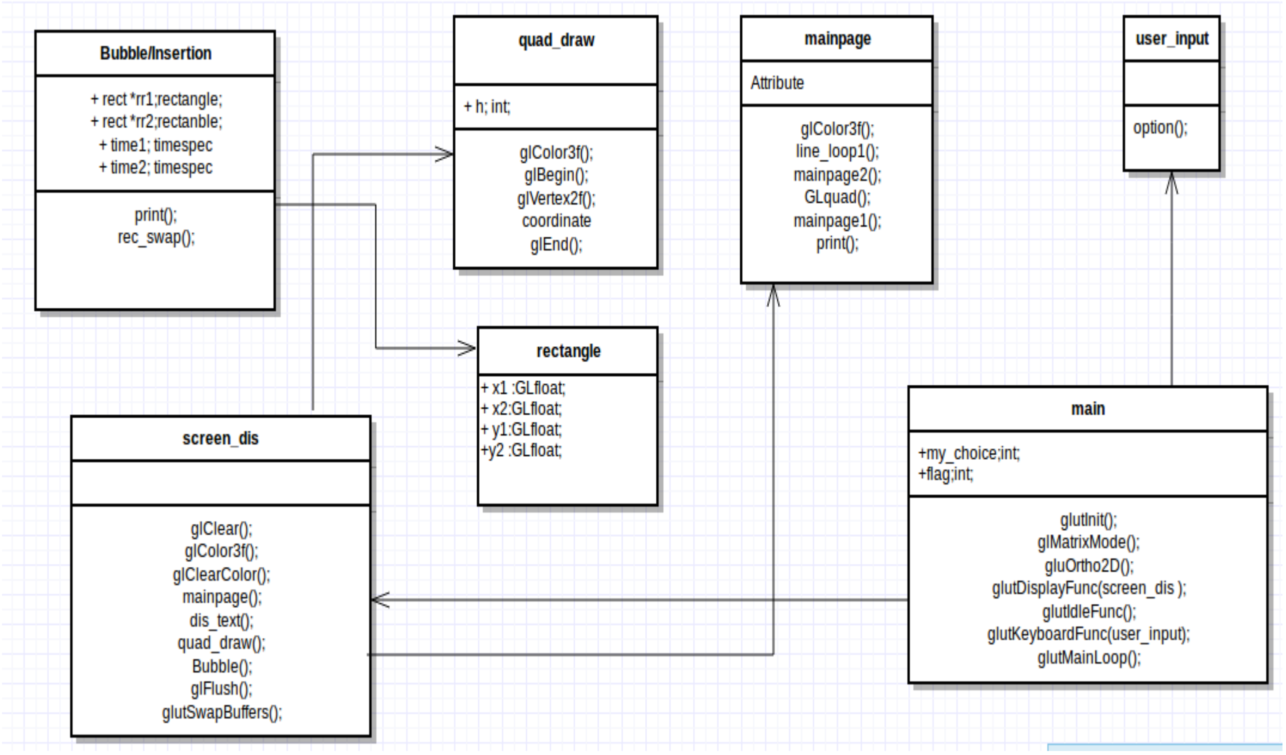
WORKING:

- On Successful compilation of the code and after running the code window will pop up and ask the user to hit space button to start.
- After hitting the space button of user the new window will pop up and provide option whether do you wants bubble sort or insertion sort.
- On hitting the button “b” algorithm of bubble sort will start and user can demostrate bubble sort algorithm.
- On hitting the button “Esc” program will terminate.

IMPLEMENTATION :

- The Rectangle is made using glBegin(GL_POLYGON) function.
- Translation of the Rectangle done by Arithmatics and timer(nanosleep()) function;
- To draw Lines we use glBegin(GL_LINE_LOOP).

CLASS-DIAGRAM:



*note: code has been attached

x

```
#include<bits/stdc++.h>
#include<cstdlib>
#include<cstdio>
#include<cmath>
#include<GL/glut.h>
#include<ctime>
#include<cstring>
#include<cstdlib>
using namespace std;

#define maximum 10

// define variable
int k=0;
GLfloat start_x11,start_x12, start_x21, start_x22;
int I = 0, J = 0,flag=0,N,srtng = 0,swp = 0;



// print character in window
void print(int x, int y, string str1, void *font)
{
        int length, k;
          glRasterPos2f(x, y);
          length = str1.length();
           k= 0;
           while(k<length){
                    glutBitmapCharacter(font, str1[k]);
```

```
            k++;
        }

}

// declration of rectaingle
typedef struct rectangle
{
        GLfloat x1;
        GLfloat x2;
        GLfloat y1;
        GLfloat y2;
} rect;

// declration  of arrays
int arr[maximum];
int start[maximum] = {4,10,8,3,6,9,7,2,5,1};
rect rect1[maximum];

// declaraion of display of line loop
void line_loop1(){
      glBegin( GL_LINE_LOOP );
        glVertex2f(0, 541);
        glVertex2f(1000, 541);
        glEnd();
 }

// defining coordinate

void coordinate(rect r){
      glVertex3f(r.x1, r.y1, 0.0);
        glVertex3f(r.x2, r.y1, 0.0);
        glVertex3f(r.x2, r.y2, 0.0);
        glVertex3f(r.x1, r.y2, 0.0);
 }

// displaying main page content part 1
void mainpage1(){

      glColor3f(248, 248, 255);
        print(200, 300, " DEMONSTRATION OF  SORTING
ALGORITHM",GLUT_BITMAP_TIMES_ROMAN_24);
        glBegin(GL_LINE_LOOP);
        glVertex2f(198, 295);
        glVertex2f(700, 295);
        glEnd();

}

// drawing quadrilateral
void quad_draw(rect r)
{
        glColor3f(0.10, 1.0, 0.10);
        glBegin(GL_POLYGON);
        coordinate(r);
        glEnd();
        glColor3f(0.765, 0.767, 0.225);

        int h = (r.y2 - 50 )/ 20;
        stringstream ss;
        ss << h;
        string str = ss.str();
        glColor3f(0.0, 0.0, 0.0);
        print(r.x1 + 2.5, r.y1, str , GLUT_BITMAP_9_BY_15);
}


// dislaying  mainpage content 2
 void mainpage2(){

    glColor3f(0.502, 0.000, 0.000);
   // print(400, 375, "WELCOME", GLUT_BITMAP_TIMES_ROMAN_24);
        glBegin( GL_LINE_LOOP );
        glVertex2f(0,122);
        glVertex2f(1000,122);
        glEnd();

 }
// swapping or translation in this function
void rec_swap(rect *rr1,rect *rr2)
{

      vector<int>v2;
      struct timespec tim1, tim2;
      tim1.tv_sec = 0;
      tim1.tv_nsec = 20000000L;

        if (swp == 0)
        {
            vector<int> v1;
```

```
     v1.push_back(0);
        start_x11 = rr1->x1;
        start_x12 = rr1->x2;
     v1.clear();
        start_x21 = rr2->x1;
        start_x22 = rr2->x2;
        swp = 1;
   }
 // cheaking conditions
  v2.clear();
  v2.push_back(0);
 //cheaking left bottom
 vector<int>v3;
 v3.clear();
   if (start_x11 <= rr2->x1){

        rr2->x1 -= 0.5;
      v3.push_back(v2[0]);

   }
 // cheaking right bottom
   if(start_x12 <= rr2->x2){

        rr2->x2 -= 0.5;
      v3.push_back(v2[0]);
  }

//  cheaking for right top
   if (start_x21 >= rr1->x1){
```

```
                        rr1->x1 += 0.5;
                    v3.push_back(v2[0]);
            }
    // cheaking for left top
              if (start_x22 >= rr1->x2){

                        rr1->x2 += 0.5;
                    v3.push_back(v2[0]);
            }



            v3.clear();
            v2.clear();
  // swapping of rectaingle takes place
              if (abs(start_x11-rr2->x1) < 0.3
            && abs(start_x12 - rr2->x2) < 0.3
            && abs(start_x21-rr1->x1) < 0.3
            && abs(start_x22 - rr1->x2) < 0.3)
              {
                        swp = 0;
   //            swaping of first coordinate
                        int temp = rr1->x1;
                        rr1->x1 = rr2->x1;
                        rr2->x1 = temp;

   //             swapping of second codrdinates

                        temp = rr1->x2;
                        rr1->x2 = rr2->x2;
                        rr2->x2 = temp;
```

```
//              swapping of third coordinate
                     temp = rr1->y1;
                     rr1->y1 = rr2->y1;
                     rr2->y1 = temp;
//           swapping of fourth quardinate of rectaingle
                     temp = rr1->y2;
                     rr1->y2 = rr2->y2;
                     rr2->y2 = temp;


           }
    // function to speed of translation
        nanosleep(&tim1, &tim2);
}

// displaing line loop again
 void line_loop(){


        glBegin(GL_LINE_LOOP);
          glVertex2f(198+50, 95);
          glVertex2f(700, 95);
          glEnd();


}


 void println1(){


                glColor3f(0.5,0.5,0.5);
                     print(10, 455, "  SORTING... ",GLUT_BITMAP_9_BY_15);
                     glColor3f(0.0,0.0,0.0);
 }

// implementation of bubble sort algorithm
void Bubble()
{
    if( !swp )
    {
        if( flag == 0 )
        {
            for(I = 0; I < N; I++)
            {

                J = I;
                for(; J < N-1; J++)
                {
                    if (arr[J] > arr[J+1])
                    {
                        int temp = arr[J];
                        arr[J] = arr[J+1];
                        arr[J+1] = temp;
                        goto SWAP;
                    }
                }
            }
        }
        if( flag == 1 ){
            for(I = 0; I < N; I++)
            {
                J = I;
                for(; J < N-1; J++)
                {
                    if (arr[J] < arr[J+1])
                    {
                        int temp = arr[J];
                        arr[J] = arr[J+1];
                        arr[J+1] = temp;
                        goto SWAP;
                    }
                }
            }

        }

    }

SWAP:
        print(10, 375, "swp ->",GLUT_BITMAP_9_BY_15);
        rec_swap(&rect1[J],&rect1[J+1]);
}



// implemention of insertion sort
void Insertion()
{
    int pos;

    if( !swp )
    {
        if(flag == 0 )
        {
```

```
    for(I = 0; I < N-1; I++)
    {
        pos = I;
        for(J = I + 1; J < N; J++)
        {
            if(arr[pos] > arr[J] )
                pos = J;
        }

        int temp = arr[I];
        arr[I] = arr[pos];
        arr[pos] = temp;
        goto SWAP;
    }
}
if(flag == 1 )
{
    for(I = 0; I < N-1; I++)
    {
        pos = I;
        for(J = I + 1; J < N; J++)
        {
            if( arr[pos] < arr[J] )
                pos = J;
        }
        int temp = arr[I];
        arr[I] = arr[pos];
        arr[pos] = temp;
        goto SWAP;
    }
```

```
                  }
         }
SWAP:
                print(10, 375, "swp ->",GLUT_BITMAP_9_BY_15);
                  rec_swap(&rect1[I],&rect1[pos]);


}


//  function to print values
     void println(){
              // print on screeen
              print(10, 455, "*******OPTIONS*********",GLUT_BITMAP_9_BY_15);
                  print(10, 435, " b to BUBBLE SORT",GLUT_BITMAP_9_BY_15);
                  print(10, 415, " i to INSERTION SORT",GLUT_BITMAP_9_BY_15);
                  print(10, 395, " Esc to QUIT",GLUT_BITMAP_9_BY_15);
    }
// printing demo version
void print_demo(){


       glBegin(GL_LINE_LOOP);
         glVertex2f(98, 560);
         glVertex2f(840, 560);
         glEnd();
  }


// drawing quad


void GLquad(){


       glBegin(GL_QUADS);
    glVertex2f(0,125);
   glVertex2f(1000,125);
   glVertex2f(1000,540);
   glVertex2f(0,540);
    glEnd();
}

void dis_text()
{
    print(100, 565, "*************************DEMONSTRATION OF SORTING 
ALGORITHM********************",GLUT_BITMAP_TIMES_ROMAN_24);
    print_demo();
    if (srtng == 0)
    {
        println();
    }
    else if (srtng == 1)
    {
        println1();
    }
}

// displaying option

void option(unsigned char button){

     switch (button)
        {
            case 27 : exit (0);
            case 'b' : srtng = 1; break;
```

```
                              case 'i' : srtng = 2; break;


          }


}
// displaying main page
void mainpage()


{
        //mainpage1();
          glColor3f(0.000, 0.749, 1.000);
        line_loop1();
        mainpage2();
        GLquad();
          mainpage1();
          glColor3f(0.000, 0.749, 1.000);
          print(250, 100, "PRESS SPACE BUTTON  FOR SORTING "
,GLUT_BITMAP_TIMES_ROMAN_24);
        line_loop();



}


// function for user input


void user_input( unsigned char button, int x, int y)
{
          if(button==32)
                    k=1;
          if (k==1)
```

```
    {
        option(button);
    }
}

// displaying  screen
void screen_dis(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.647, 0.165, 0.165);
    glClearColor(0.0, 0.0, 0.0, 0.0);

    if (k==0){

        mainpage();
   }
    else
    {
        dis_text();
        glPointSize(2.0);
        for (int i=0;i<N;i++)
        {
            glColor3f(1.0,0.0,0.0);
            quad_draw(rect1[i]);
        }
        if ( srtng == 1 ){
             Bubble();
      }
        else if( srtng == 2){
            Insertion();
            }
                else{
                  srtng = 0;
            }
        }
        glFlush();
        glutSwapBuffers();


}



int main(int argc, char ** argv)
{

        glutInit(&argc,argv);
        glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
        glutInitWindowPosition(50,50);
        glutInitWindowSize(950,700);
        glutCreateWindow("SORTING  ALGORITHM IMPLEMENTATION");
        glClearColor(1.0,1.0,1.0,0.0);
        glMatrixMode(GL_PROJECTION);
        gluOrtho2D(0.0,900.0,0.0,600.0);
        cout << "do u want to enter element Or filling default value" << endl;
        int my_choice;
        cout << "1. Manual" << endl;
        cout << "2. Auto" << endl;
        cin >> my_choice;
```

```
     if( my_choice == 1 )
     {
             cout << "maximum U can stored 10 values " << endl;

             cout << "enter N for No. of elment" << endl;
             cin >> N;

             cout << "enter the element" << endl;
             for(int i = 0; i < N; i++)
             {
                     cout << "Enter "<< i +1 << ": ";
                     cin >> start[i];
             }
       cout << " enter the 0 for ascending and 1 for descending \n";
       cin >> flag;
     }
     else if(my_choice == 2 )
   {
    N = maximum;
    cout << " enter the 0 for ascending and 1 for descending \n";
    cin >> flag;
     }
     else
     {
             exit(0);
     }

     I = J = swp = 0;
// initialisation of rctaingle object
     for (int k=0;k<N;k++)
```

```
        {
                arr[k] = start[k];
                rect1[k].y2 = arr[k]*20.0 + 50.0;
                rect1[k].y1 = 50.0;
                if (k == 0){
                        rect1[k].x1 = 45.0;
                   rect1[k].x2 = 65.0;
            }
                else{
                        rect1[k].x1 = rect1[k-1].x1 + 60.0;
                   rect1[k].x2 = rect1[k-1].x2 + 60.0;
            }
        }
        glutDisplayFunc(screen_dis);
        glutIdleFunc(screen_dis);
        glutKeyboardFunc(user_input);
        glutMainLoop();
        return 0;
}
```