**PROJECT REPORT**

**PROJECT TITLE:**

**RENDERING OF KEYS THROUGH OPENGL**

**Prepared By:**

**Name**: Ashish Naik **Name**: Kunal Naickar

**Roll. No**.: 62 **Roll. No**.: 63

**Name**: Anjali Pai **Name**: Yuvraj Patadia

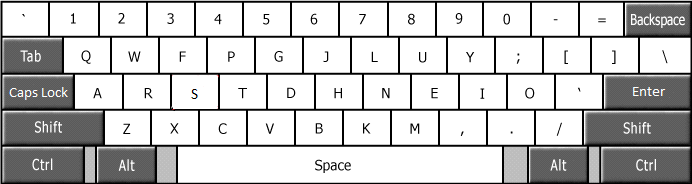
**Roll. No**.: 67 **Roll. No**.: 71

**Brief Introduction:**

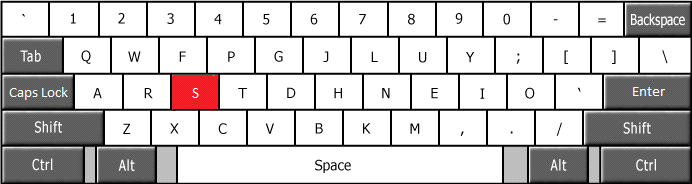
In this project we are going to design a layout of the standard keyboard using OpenGL. This layout will look similar to the virtual keyboard layout but will function differently. This keyboard layout will highlight the key on the layout which is pressed on the keyboard.

**Design of Solution:**

The virtual keyboard layout will look as follows:



On pressing, *assume the key 'S',* the keyboard would look as follows with the block lettered S highlighted with the color red.



**Methodology:**

In this mini project i.e., designing a keyboard layout we will make use of the functions provided within the openGL library. We will create the layout with the help of functions like GL\_POLYGON etc thereby creating each key. Each key will have the same default color. Whenever the program is executed and any valid key is pressed then that particular key will be highlighted for approximately less than one second of time. The highlighting will be done by changing the color of the polygon representing the pressed key.

**Design & Implementation**

**Algorithm of the program:**

Step 1: Start

Step 2: Set window size, window position.

Initialize keyy and key1 to 0.

Step 3: Display( )

3.1: for every key represented as polygon

Check if keyy is equal to ascii value of that key

If true set color to red

Otherwise set default color for key.

3.2: Draw al key polygons on the screen.

Step 4: keys( )

4.1: Check if arrow key is pressed

If true store ascii value in key, go to step 3.1

Else check if special key (shift, control, alt) is pressed using glutGetModifiers( )

If pressed store state in mod variable.

If either of shift, control, alt is pressed then

We set user defined value to keyy and store the ascii value of key1 pressed in combination with special key, go to 3.1.

Else

Normal key is pressed

Assign ascii value of key pressed to keyy.

Go to step 3.1

Step 5: Stop

**Code:**

#include <stdio.h>

#include <GL/glut.h>

#include <GL/freeglut.h>

#include <stdlib.h>

char keyy,key1;

void display();

void keyup(int key,int x,int y)

{

key1=0;

switch(key)

{

case GLUT\_KEY\_LEFT: keyy=20;

break;

case GLUT\_KEY\_UP: keyy=21;

break;

case GLUT\_KEY\_DOWN: keyy=22;

break;

case GLUT\_KEY\_RIGHT: keyy=19;

break;

default: return;

}

display();

}

void keys(unsigned char key,int x,int y)

{

int mod;

glutSpecialFunc(keyup);

mod = glutGetModifiers();

if(mod==GLUT\_ACTIVE\_SHIFT)

{

keyy=16;

key1=key;

}

else if(mod==GLUT\_ACTIVE\_CTRL)

{

keyy=17;

key1=key;

}

else if(mod==GLUT\_ACTIVE\_ALT)

{

keyy=18;

key1=key;

}

else

{

keyy=key;

key1=0;

}

display();

}

void display(void)

{

int i,j;

char str1[10]="Backspace",str2[4]="Tab",str3[9]="Capslock",str4[6]="Enter",str5[6]="Shift",str6[5]="Ctrl",str7[4]="Alt",str8[5]="DELL",str9[3]="\\",str10[3]="<-",str11[3]="->",str[2]="\\", fstr1[56]="Project Title: Keyboard Layout with Key Press Detection", fstr2[90]="Submitted By: Ashish Naik, Yuvraj Patadia, Anjali Pai & Kunal Naickar", fstr3[34]="Guided By : Prof. Lissa Rodrigues";

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,1.0,1.0);

glBegin(GL\_LINE\_LOOP); // White Border

glVertex2f(-0.92,0.92);

glVertex2f(0.98,0.92);

glVertex2f(0.98,0.00);

glVertex2f(-0.92,0.00);

glEnd();

glFlush();

glColor3f(1.0,0.0,0.0);

glBegin(GL\_LINES); //Red Line

glVertex2f(-0.92,-0.01);

glVertex2f(0.98,-0.01);

glEnd();

glFlush();

glColor3f(0.0,1.0,0.0);

glBegin(GL\_LINES); //Green Line

glVertex2f(-0.92,-0.02);

glVertex2f(0.98,-0.02);

glEnd();

glFlush();

glColor3f(0.0,0.0,1.0);

glBegin(GL\_LINES); //Blue Line

glVertex2f(-0.92,-0.03);

glVertex2f(0.98,-0.03);

glEnd();

glFlush();

if(key1=='~' || keyy=='`')

glColor3f(1.0,0.0,0.0);

else

glColor3f(1.0,1.0,1.0);

glBegin(GL\_POLYGON); //key ~

glVertex2f(-0.9,0.9);

glVertex2f(-0.78,0.9);

glVertex2f(-0.78,0.76);

glVertex2f(-0.9,0.76);

glEnd();

glFlush();

glColor3f(0.0,0.0,0.0);

glRasterPos2f(-0.84,0.85);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,'~');

glRasterPos2f(-0.84,0.78);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,'`');

if(keyy=='1' || key1=='!' ||key1=='1')

glColor3f(1.0,0.0,0.0);

else

glColor3f(1.0,1.0,1.0);

glBegin(GL\_POLYGON); //key 1

glVertex2f(-0.77,0.9);

glVertex2f(-0.65,0.9);

glVertex2f(-0.65,0.76);

glVertex2f(-0.77,0.76);

glEnd();

glFlush();

glColor3f(0.0,0.0,0.0);

glRasterPos2f(-0.71,0.8);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,'1');

if(key1=='A' || keyy=='a' || key1=='a')

glColor3f(1.0,0.0,0.0);

else

glColor3f(1.0,1.0,1.0);

glBegin(GL\_POLYGON); //key A

glVertex2f(-0.71,0.60);

glVertex2f(-0.59,0.60);

glVertex2f(-0.59,0.46);

glVertex2f(-0.71,0.46);

glEnd();

glFlush();

glColor3f(0.0,0.0,0.0);

glRasterPos2f(-0.67,0.50);

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,'A');

glColor3f(1.0,0.0,0.0);

glRasterPos2f(-0.03,0.03);

for(i=0;i<4;i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,str8[i]);

glColor3f(0.0,0.5,0.8);

glRasterPos2f(-0.35,-0.30);

for(i=0;fstr1[i]!='\0';i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,fstr1[i]);

glColor3f(0.0,0.5,0.8);

glRasterPos2f(-0.35,-0.50);

for(i=0;fstr2[i]!='\0';i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,fstr2[i]);

glColor3f(0.0,0.5,0.8);

glRasterPos2f(-0.35,-0.70);

for(i=0;fstr3[i]!='\0';i++)

glutBitmapCharacter(GLUT\_BITMAP\_HELVETICA\_18,fstr3[i]);

glColor3f(1.0,1.0,1.0);

glBegin(GL\_LINES);

glVertex2f(-0.40,-0.20);

glVertex2f(-0.05,-0.20);

glVertex2f(-0.38,-0.18);

glVertex2f(-0.38,-0.45);

glVertex2f(0.63,-0.75);

glVertex2f(0.3,-0.75);

glVertex2f(0.6,-0.78);

glVertex2f(0.6,-0.45);

glEnd();

glFlush();

// glutKeyboardFunc(keys);

}

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

{

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(1250,600);

glutInitWindowPosition(50,50);

glutCreateWindow("Keyboard Layout");

glutDisplayFunc(display);

glutKeyboardFunc(keys);

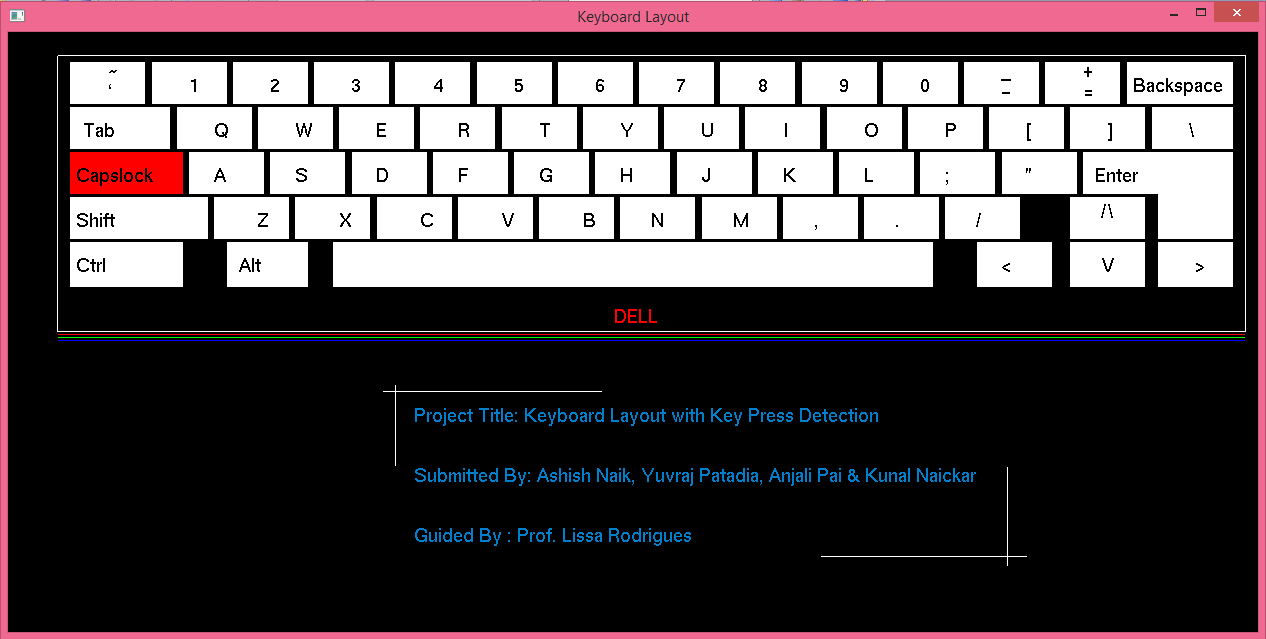
glutMainLoop();

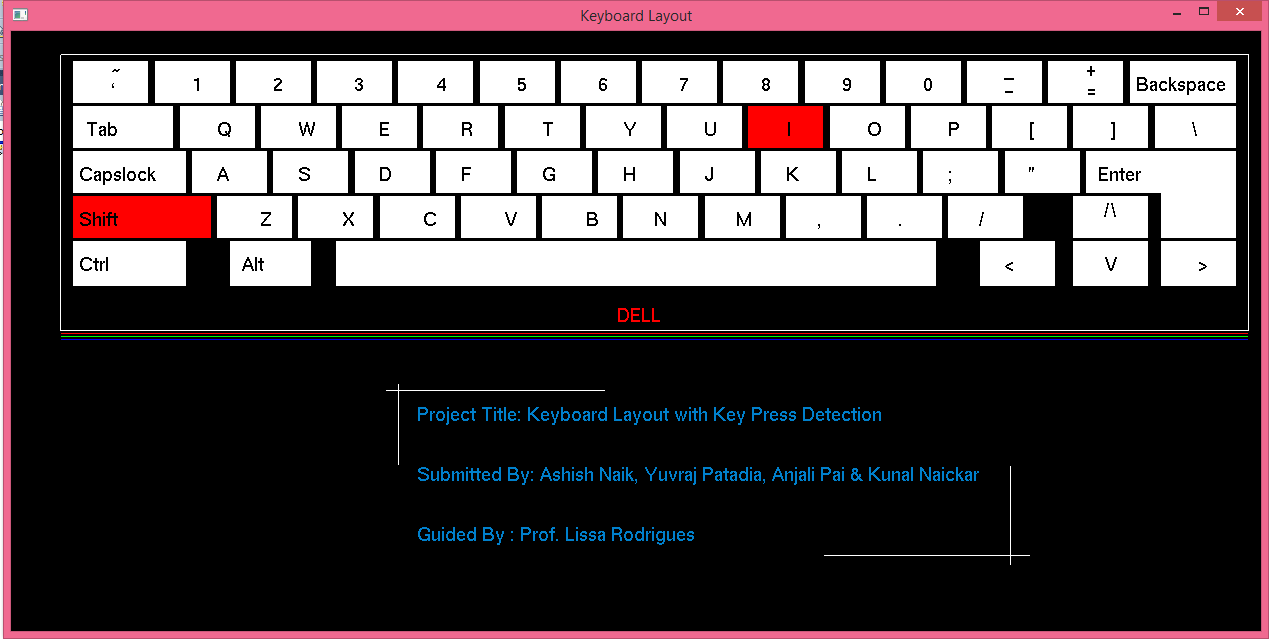
}

**Results**

The displaying of the keyboard layout and detecting the key pressed to highlight the key which is pressed. The Snapshots are as showed below.

**Snapshots:**

****

****

**Conclusion**

OpenGL serves as an efficient platform to create graphic applications. We have demonstrated following:

* Displaying a keyboard layout.
* Taking input as to which key is pressed i.e., implementing keyboard callback.
* Highlighting the pressed key on the layout.
* In future this project can be further improvised to include:
* Implementing it as a virtual keyboard by using mouse callback.
* Key pop up when key pressed.

So to conclude we have tried to display a keyboard layout with the main keys on keyboard and hence to capture the key pressed and then to highlight the respective key on the layout.