

Draw different borders in different colors

Ang Lee coolpix@cc.gatech.edu

```
int i, loop;  
int StartCorner;
```

```
/* create a temporary oo table, copy the O-Table */
```

```
temp-oo-table [] = o-table[i]
```

```
for (i = 0; i < ccount; i++)  
    oo[i] = o[i];
```

```
/* find a corner located on the border */
```

Search in the V-table, go backward since the newer corner tend to be on the border.

When find the first corner whose oo entry is not -1, terminate the search and start dealing with it.

And set the oo entry to -2, to prevent it being considered as border corner again in the future.

```
loop = -1;
```

```
for (i = ccount-1; i >= 0; i--)
```

```
{
```

```
    if (oo[i] == -1)
```

```
    {
```

```
        oo[i] = -2; //set its 'oo' to -2 indicating it has been drawn, prevent being drawn next time.
```

```
        loop = n(i); // found one loop
```

```
        break;
```

```
    }
```

```
}
```

```
/* Draw a border loop */
```

When there is a corner on the border found loop != -1, there is a border to be drawn.

```
while(loop != -1)
```

```
{
```

After finding a corner, draw the loop. Remember the corner which we start.

```
StartCorner = loop;
```

```
/* This is the beginning of drawing a new border, change to a different color */
```

```
ChangeColor()
```

```
/* Start drawing*/
```

```
glBegin(GL_LINE_LOOP);
```

```
/* the following is a do-while loop, when the next corner is not the corner we started, go to it and draw  
a segment of border */
```

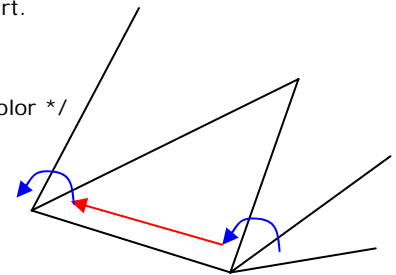
```
do{
```

```
    if(o[p(loop)] != -1) // Blue arrow case above
```

```
        loop = p(o[p(loop)]); // move to the next corner sharing the same vertex
```

```
    else // Red arrow case above
```

```
{
```



```

        loop = n(loop);

        oo[n(loop)] = -2; //set its 'oo' to -2 indicating it has been drawn, prevent being drawn next time.

        glVertex2f( gx[v[loop]], gy[v[loop]]);
    }
}

while (loop!=StartCorner);

glEnd(); //Finish drawing a border loop.

/* find next corner located on the border */
Search in the V-table, go backward since the newer corner tend to be on the border.
When find the first corner whose oo entry is not -1, terminate the search and start dealing with it.
And set the oo entry to -2, to prevent it being considered as border corner again in the future.

loop = -1;
for (i = ccount-1; i>=0; i--)
{
    if (oo[i] == -1)
    {
        oo[i] = -2; //set its 'oo' to -2 indicating it has been drawn, pevent being drawn next time.
        loop = n(i); // found one loop
        break;
    }
}
}
}

```