Lab-02: 2D object Coloring with OpenGL

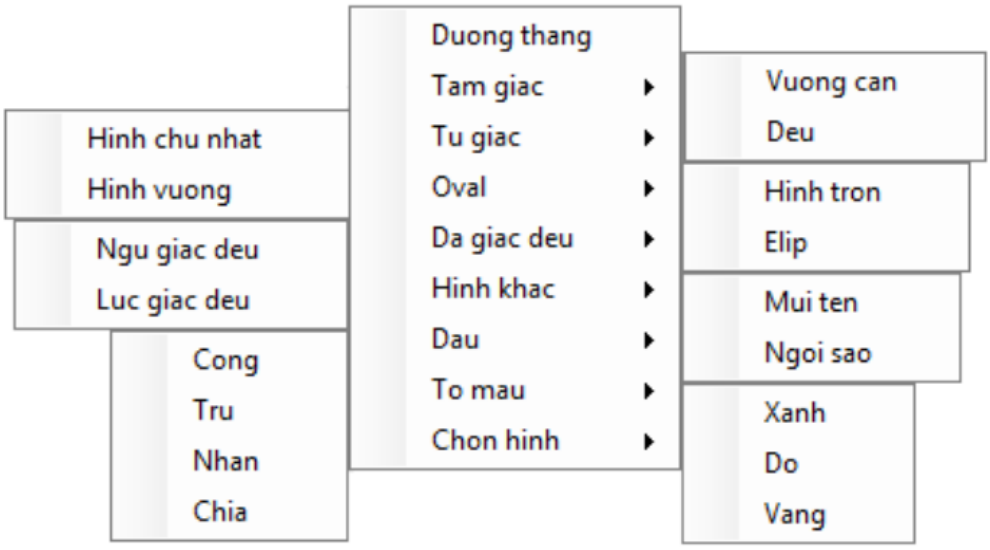
CSC411: Computer Graphics
Vo Hoai Viet (vhviet@fit.hcmus.edu.vn)

Objective

In this lab, the students will implement the algorithms to color 2D Object using OpenGL Library.

Requirements

- Write an OpenGL program as follows:
    - Allow to draw one of any basic shape includes (circle, ellipse, rectangle, polygon) using context menu
    - Allow to coloring this shape using seed point selected by clicking inside the polygon
    - Right click on draw window will prompt context menu as below with all the mentioned functions.



- **Note:** The boundary fill is recursive algorithm. So beware of stack overflow error, which could avoid by drawing small polygons instead of big ones.
    - Boundary Fill is algorithm used for the purpose of coloring figures in computer graphics. Here area gets colored with pixels of a chosen color as boundary this giving the technique its name. One can see the difference in the conditions that

are there for planting the seeds. Boundary fill fills the chosen area with a color until the given colored boundary is found.

- This algorithm is recursive in nature as the function returns when the pixel to be colored is the boundary color or is already the fill color. The below code demonstrate boundary fill coloring algorithm using OpenGL.

```
void BoundaryFill (int x, int y, RGBColor F_Color, RGBColor B_Color)
{
    RGBColor currentColor;

    currentColor = GetPixel(x, y);

    if(!IsSameColor(currentColor, B_Color) && !IsSameColor(currentColor, F_Color))
    {
        PutPixel(x, y, F_Color);
        FillLeft(x - 1, y, F_Color, B_Color);
        FillTop(x, y + 1, F_Color, B_Color);
        FillRight(x + 1, y, F_Color, B_Color);
        FillBottom(x, y - 1, F_Color, B_Color);
    }
}
```

In which

- RGBColor is (R,G,B) struct defined as:

```
typedef struct _RGBColor
{
    unsigned char r;
    unsigned char g;
    unsigned char b;
}RGBColor;
```

- IsSameColor is self-implementation function to compare 2 colors
- FillLeft, FillRight, FillTop, FillBottom could be implemented as same as BoundaryFill function.

- GetPixel(x, y): implement GetPixel function to retrieve color as specific location (x, y)

```
RGBColor GetPixel(int x, int y)
{
    unsigned char * ptr = new unsigned char [3];

    glReadPixels(x, h -  y, 1, 1, GL_RGB, GL_UNSIGNED_BYTE, ptr);

    RGBColor color;
    color.r = ptr[0];
    color.g = ptr[1];
    color.b = ptr[2];

    return color;
}
```

- PutPixel(x, y)

  Implement PutPixel function to set color as specific location (x, y)

```
void PutPixel(int x, int y, RGBColor color)
{
    unsigned char * ptr = new unsigned char [3];
    ptr[0] = color.r;
    ptr[1] = color.g;
    ptr[2] = color.b;

    glRasterPos2i(x, y);
    glDrawPixels(1, 1, GL_RGB, GL_UNSIGNED_BYTE, ptr);

    glFlush();
}
```

- Mouse click event

```
void XuLyMouse(int button, int state, int x, int y)
{
    if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        BoundaryFill(x, y, F_Color, B_Color);
    }
}
```

- **Self-research:**
  - Flood Fill Coloring Algorithm
  - Scan Line Coloring Algorithm
- **Note:**
  - Preset: gluOrtho2D(0.0, w, h, 0.0);
  - w, h is window's width & height
  - F_Color, B_Color are fill color & boundary color
  - Because glReadPixels using bottom-top direction for Oy axis, so we need to inverse the y value from location point (y' = h - y), with h is the window height.

Submission

Create 3 folder, compressed in 1 file MSSV_Lab1.zip

- Document
- Release

- Source

Resources

- Graham Sellers, OpenGL SupperBidle, Cheapter 1, 2, and 3.
- gear.kku.ac.th/~rujchai/files/CG471-07.ppt
- http://www.codeproject.com/KB/GDI/QuickFill.aspx