



## CSC16001 - ĐỒ HỌA MÁY TÍNH

### BÀI TẬP VỀ NHÀ

#### BTCN-02: VẼ VÀ TÔ MÀU ĐỐI TƯỢNG 2D

##### I. Thông tin chung

|                            |  |
|----------------------------|--|
| Mã số bài tập:             | BTCN-01  |
| Thời lượng dự kiến:        | 48 tiếng   |
| Deadline nộp bài:          |  |
| Hình thức:                 | Bài tập cá nhân  |
| Hình thức nộp bài:         | Nộp qua Moodle môn học   |
| GV phụ trách:              | Võ Hoài Việt   |
| Thông tin liên lạc với GV: | <a href="mailto:vhviet@fit.hcmus.edu.vn">vhviet@fit.hcmus.edu.vn</a> |

##### II. Chuẩn đầu ra cần đạt

Bài tập này nhằm mục tiêu đạt được các chuẩn đầu ra sau:

- Vẽ được các đối tượng đồ họa 2 chiều
- Thuật toán tô màu 2D
- Lập trình tương tác giao diện

##### III. Mô tả bài tập

Nội dung:

Xây dựng chương trình OpenGL có chức năng (tham khảo phần mềm MS Paint để nắm hiểu rõ các thao tác về sự kiện mouse)

- Tự động vẽ các đối tượng hình học cơ bản theo bảng kê bên dưới từ Menu chính của chương trình [6]
- Khi người dùng click chuột vào bên trong của đối tượng thực hiện chọn đối tượng đó [7]
- Dùng Menu tô màu để chọn màu tô cơ bản cho đối tượng đã chọn



**Lưu ý:** Thuật boundary fill là thuật toán đệ quy, để tránh lỗi tràn stack (stack overflow) nên vẽ các đối tượng tự động với kích thước pixel nhỏ.

#### **Các thuật toán tự tham khảo:**

- Flood Fill Coloring Algorithm
- Scan Line Coloring Algorithm

#### **Gợi ý tra cứu:**

- <http://www.codeproject.com/KB/GDI/QuickFill.aspx>
- <https://docs.microsoft.com/en-us/dotnet/framework/winforms/controls/walk-through-providing-standard-menu-items-to-a-form>

## **IV. Các yêu cầu & quy định chi tiết cho bài nộp**

### **Thuật toán tô màu theo Đường bao (Boundary Fill)**

Đường biên của vùng tô được xác định bởi tập các đỉnh của một đa giác, đường biên trong thuật toán được mô tả bằng một giá trị duy nhất đó là màu của tất cả các điểm thuộc về đường biên.

Bắt đầu từ điểm nằm bên trong vùng tô, ta sẽ kiểm tra các điểm lân cận của nó đã được tô màu hay có phải là điểm biên hay không, nếu không phải là điểm đã tô và không phải là điểm biên ta sẽ tô màu nó. Quá trình này được lặp lại cho tới khi nào không còn tô được điểm nào nữa thì dừng. Bằng cách này, toàn bộ các điểm thuộc vùng tô được kiểm tra và sẽ được tô hết.

#### **Mã giả (C++):**

```
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);
    }
}
```

Trong đó

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

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

- IsSameColor: hàm tự cài đặt so sánh 2 giá trị màu
- FillLeft, FillRight, FillTop, FillBottom: được cài đặt tương tự hàm BoundaryFill.

### GetPixel(x, y)

Hàm GetPixel lấy giá trị màu RGB tại tọa độ (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;
}
```

**Lưu ý:** Do hàm glReadPixels đi từ dưới lên theo trục Oy, nên chúng ta phải lật ngược giá trị y ( $y' = h - y$ ), trong đó h là chiều cao của cửa sổ vẽ.

## PutPixel(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);
    }
}
```

## Đầu ra:

- Chương trình được build dưới dạng **Release**, có file dll đi kèm
- Màn hình **OpenGL**: đối tượng được vẽ & tô màu, thời gian tô màu tính bằng mili-second (ms)
- File báo cáo (Word):
  - o Nhận xét về độ chính xác của các hàm tô màu đã cài đặt.
  - o Nêu các ưu nhược điểm của thuật toán này so với các thuật toán lý thuyết khác được học.

## V. Cách đánh giá

| STT | Tên kết quả            | Tỉ lệ điểm | Ghi chú   |
|-----|------------------------|------------|---|
| 1   | Mã nguồn, chương trình | 60%        | Cung cấp các thư viện và mã nguồn đầy đủ để biên dịch, thực hiện đầu đủ chức năng yêu cầu.  |
| 2   | Phong cách lập trình   | 20%        | Cấu trúc chương trình rõ ràng, hàm/ biến đặt tên dễ hiểu và gọi nhớ và tuân thủ các qui tắc lập trình.  |
| 3   | Báo cáo                | 20%        | Báo cáo kết quả thực hiện của các thuật toán, giải thích rõ ràng các thuật toán, thiết kế chương trình, bảng phân công công việc, slide trình bày về sản phẩm. Các nội dung tham khảo phải có liên kết. |



## VI. Tài liệu tham khảo

[1]

<https://www.geeksforgeeks.org/how-to-setup-opengl-with-visual-studio-2019-on-windows-10/>

[2] <https://github.com/dwmkerr/sharpgl>

[3] <https://www.nuget.org/packages/SharpGL>

[4] <http://www.opengl-redbook.com/>

[5] <https://learnopengl.com/>

[6] <https://followtutorials.com/2012/05/glut-tutorial-creating-menus-and-submenus-in-glut.html>

[7] <https://www.opengl.org/resources/libraries/glut/spec3/node50.html>

## VII. Các quy định khác

- Chương trình phải có hướng dẫn sử dụng (Không có hướng dẫn sử dụng sẽ bị trừ 50% số điểm của phần phần chương trình).
- Chương trình nên có giao diện trực quan thể hiện các kết quả tính toán.
- Tất cả các bài làm sai quy định đều bị 0 điểm cho mỗi bài.
- Hai bài giống nhau từ 80% trở lên sẽ bị 0 điểm cho cả hai bất kể ai là tác giả.
- Các trường hợp sử dụng mã nguồn không ghi rõ nguồn tham khảo sẽ bị điểm 0 cho tất cả các bài và các tác giả).
- Không nhận bài nộp trễ qua mail.