



## CSC16001 - ĐỒ HỌA MÁY TÍNH

### BÀI TẬP VỀ NHÀ

#### BTCN-03: VẼ VÀ TÔ MÀU ĐỐI TƯỢNG 3D

##### I. Thông tin chung

Mã số bài tập:	BTCN-03
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
- Thực hiện các phép biến đổi 2D

##### III. Mô tả bài tập

Nội dung: Mở rộng nội dung bài tập 2 bằng cách thêm các phép biến đổi 2 chiều

##### ○ Ma trận biến đổi Affine 2D

$$\begin{pmatrix} P'_x \\ P'_y \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha & \beta & 0 \\ \theta & \omega & 0 \\ t_x & t_y & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

Trong đó:

- $\alpha, \beta, \theta, \omega$ : giá trị biến đổi tuyến tính (do phép co giãn, quay, trượt)
- $t_x, t_y$ : giá trị biến đổi tịnh tiến (do phép tịnh tiến)
- $P$ : điểm gốc
- $P'$ : điểm mới

## ■ Phép tịnh tiến (Translate)

Function

$$\begin{cases} P'_x = P_x + t_x \\ P'_y = P_y + t_y \end{cases}$$

Matrix

$$M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix}$$

## ■ Phép quay (Rotate)

Function

$$\begin{cases} P'_x = \cos \alpha P_x - \sin \alpha P_y \\ P'_y = \sin \alpha P_x + \cos \alpha P_y \end{cases}$$

Matrix

$$M = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## ■ Phép co giãn (Scale)

Function

$$\begin{cases} P'_x = s_x P_x \\ P'_y = s_y P_y \end{cases}$$

Matrix

$$M = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## ■ Phép trượt (Shear)

Function

$$\begin{cases} P'_x = P_x + h_y P_y \\ P'_y = P_y + h_x P_x \end{cases}$$

Matrix

$$M = \begin{pmatrix} 1 & h_y & 0 \\ h_x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

### ● Đặc tả lớp Matrix

- Là lớp đảm nhận vai trò lưu giữ và tạo ra ma trận biến đổi affine 2D.
- **Với mỗi 1 phép biến đổi  $M_k$  sinh ra, giá trị ma trận biến đổi mới là tích của ma trận biến đổi đang lưu giữ với ma trận biến đổi  $M_k$ .**
- Mô phỏng theo lớp System.Drawing.Drawing2D.Matrix trong C#
  - Tham khảo lớp gốc tại:

<http://msdn.microsoft.com/en-us/library/w8s1ct0z.aspx#Y2700>

#### ○ Hàm tạo

<a href="#">Matrix</a>	Initializes a new instance of the Matrix class as the identity matrix.
<a href="#">Matrix(Double, Double, Double, Double, Double, Double)</a>	Initializes a new instance of the Matrix class with the specified elements.

#### ○ Thuộc tính

<a href="#">Elements</a>	Gets an array (3x3) of double-point values that represents the elements of this Matrix.
--------------------------	---

#### ○ Phương thức

<a href="#">Reset</a>	Resets this Matrix to have the elements of the identity matrix.
-----------------------	---



<a href="#">Invert</a>	Inverts this Matrix
<a href="#">Rotate(Double)</a>	Prepend to this Matrix a clockwise rotation, around the origin and by the specified angle.
<a href="#">Translate( Double , Double )</a>	Applies the specified translation vector (offsetX and offsetY) to this Matrix by prepending the translation vector.
<a href="#">Scale( Double , Double )</a>	Applies the specified scale vector to this Matrix by prepending the scale vector.
<a href="#">Shear( Double , Double )</a>	Applies the specified shear vector to this Matrix by prepending the shear transformation.
<a href="#">Multiply(Matrix)</a>	Multiplies this Matrix by the matrix specified in the matrix parameter, by prepending the specified Matrix.
<a href="#">TransformPoints(vector&lt;Point&gt;)</a>	Applies the geometric transform represented by this Matrix to a specified array of points.
<a href="#">TransformPoint(Point)</a>	Applies the geometric transform represented by this Matrix to a point.

### ○ Ví dụ

```
Matrix myMatrix;  
myMatrix.Rotate(30);  
myMatrix.Scale(1, 2);  
myMatrix.Translate(5, 0);  
  
vector<Point> pointsNew = myMatrix.TransformPoints(points);
```

### ○ Lưu ý

Khi thao tác với mảng trong C++ thay vì cách sử dụng mảng truyền thống có thể sử dụng kiểu dữ liệu `vector<Type>` để tránh phải quản lý vùng nhớ và dễ thực hiện các thao tác truy xuất, chèn, xóa, cắt...

Tham khảo tại:

- <http://www.cppreference.com/wiki/container/vector/start>
- <http://www.cplusplus.com/reference/stl/vector/>

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

##### Yêu cầu:

- Sử dụng các hình đã vẽ trong bài tập 2.
- Thực hiện **chọn 1 đối tượng đã vẽ từ menu chuột phải** và thực hiện biến đổi đối tượng chọn bằng bàn phím (chỉ biến đổi đối tượng chọn, các đối tượng khác giữ nguyên).
  - o Quay trái phải (mỗi lần 1 độ): <L>, <R>
  - o Tịnh tiến theo 4 chiều mũi tên (mỗi lần 1 đơn vị tọa độ): 4 phím mũi tên
  - o Phóng to thu nhỏ cả 2 chiều (mỗi lần 10%): <+>, <->

##### **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)

#### 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.