

Đồ họa



Tuần 3

Giảng viên: Trần Đức Minh

# Nội dung bài giảng



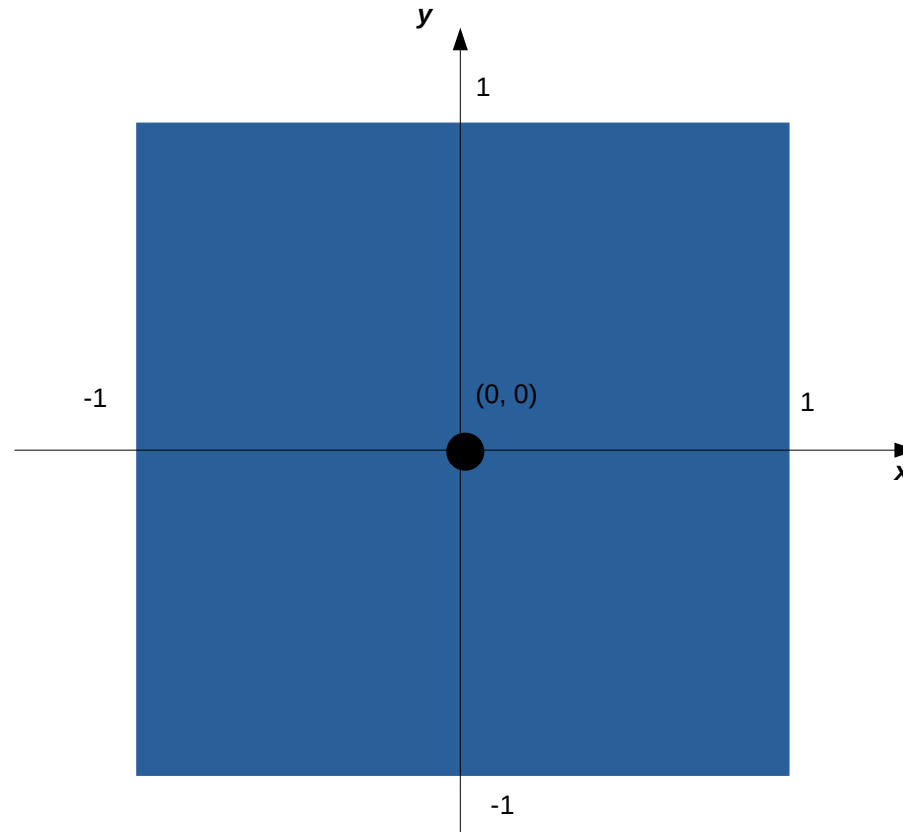
- Biến đổi tọa độ người dùng
- Khung nhìn
- Phép biến đổi Affine
- Phép biến đổi hình 2D
- Các ví dụ



# Biến đổi tọa độ người dùng



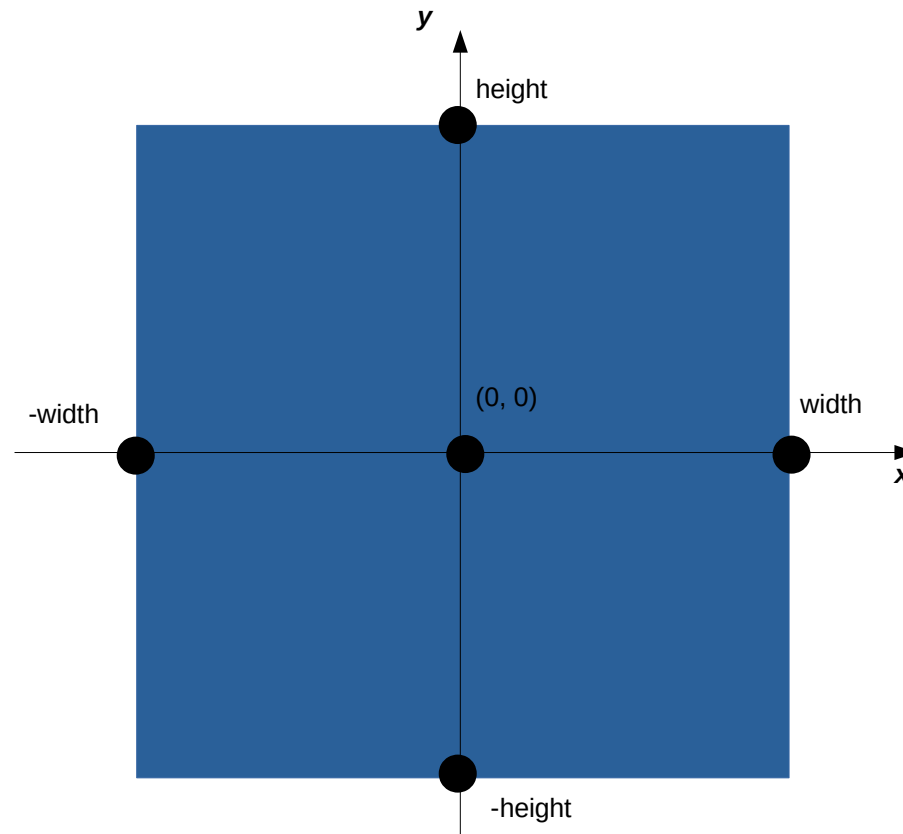
- Trục tọa độ của WebGL



# Biến đổi tọa độ



- Trục tọa độ của người sử dụng



# Biến đổi tọa độ



- Trục tọa độ của người sử dụng
  - **Thuận tiện** hơn cho người sử dụng trong việc dựng hình và tính toán.
- Cần **biến đổi** từ trục tọa độ người sử dụng sang trục tọa độ WebGL **để hiển thị thông tin** lên màn hình cho đúng.
- Công thức:
  - $x_{\text{webgl}} = x_{\text{user}} / \text{width}; \quad // -128 / 512 = -0.25 \text{ webgl}$
  - $y_{\text{webgl}} = y_{\text{user}} / \text{height};$

# Hàm trong GLSL



- Hàm trong GLSL được định nghĩa giống như trong C/C++.
  - **Phải được khai báo** trước khi đem ra sử dụng (nguyên mẫu hàm).
  - **Không được phép gọi đệ quy.**
  - Khi **truyền đối số cho hàm là một mảng**, ta cần phải xác định rõ số lượng phần tử của mảng.
    - Ví dụ: `void tinhToan(vec3 values[20]);`

# Ví dụ



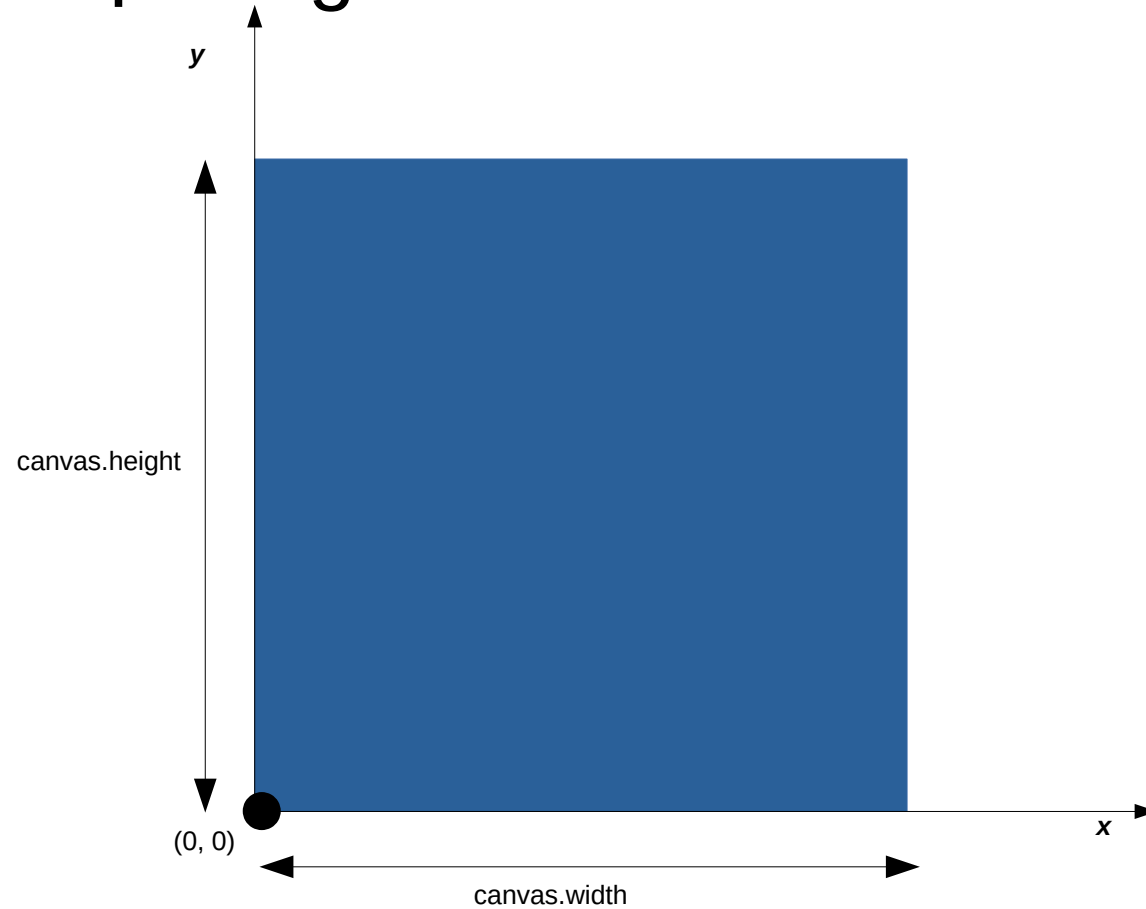
- Ví dụ 1
  - Vẽ một hình tam giác dựa trên tọa độ người sử dụng.
  - Xây dựng một **hàm** để biến đổi từ tọa độ người sử dụng sang tọa độ WebGL trong GLSL.
- Ví dụ 2
  - Vẽ một chữ T lên màn hình dựa trên tọa độ người sử dụng



# Khung nhìn



- Trục tọa độ thế giới

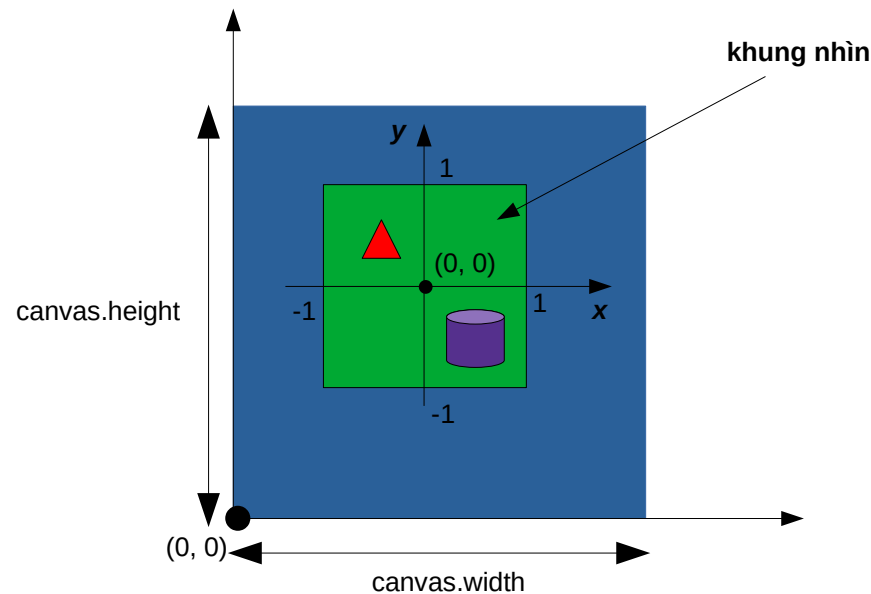




# Khung nhìn



- **Khung nhìn (viewport)** là một khung cửa sổ nằm trong trục tọa độ thế giới
  - Dùng để chứa toàn bộ các đối tượng hình ảnh được vẽ trên trục tọa độ WebGL (tức là toàn bộ trục tọa độ WebGL sẽ nằm bên trong khung nhìn).



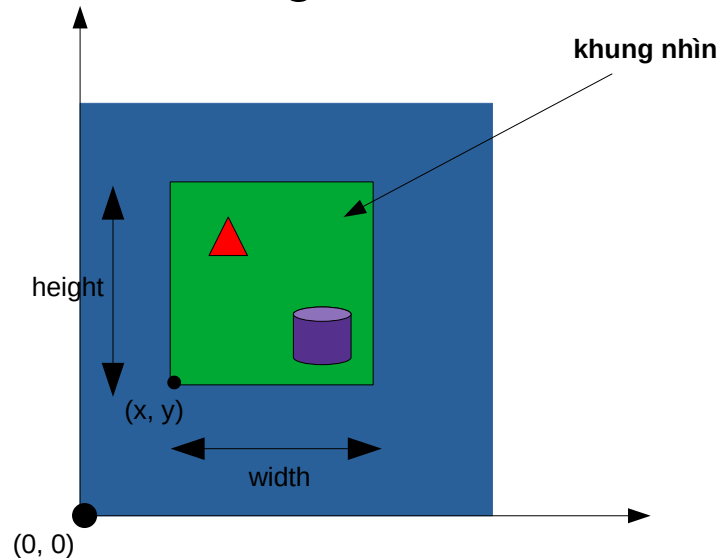
# Khung nhìn



- Để đặt khung nhìn, ta thực hiện câu lệnh sau

**`gl.viewport(x, y, width, height);`**

- **(x, y)**: Điểm tại góc dưới trái của khung nhìn trong trục tọa độ thế giới.
- **width**: Chiều rộng của khung nhìn.
- **height**: Chiều cao của khung nhìn.



# Khung nhìn



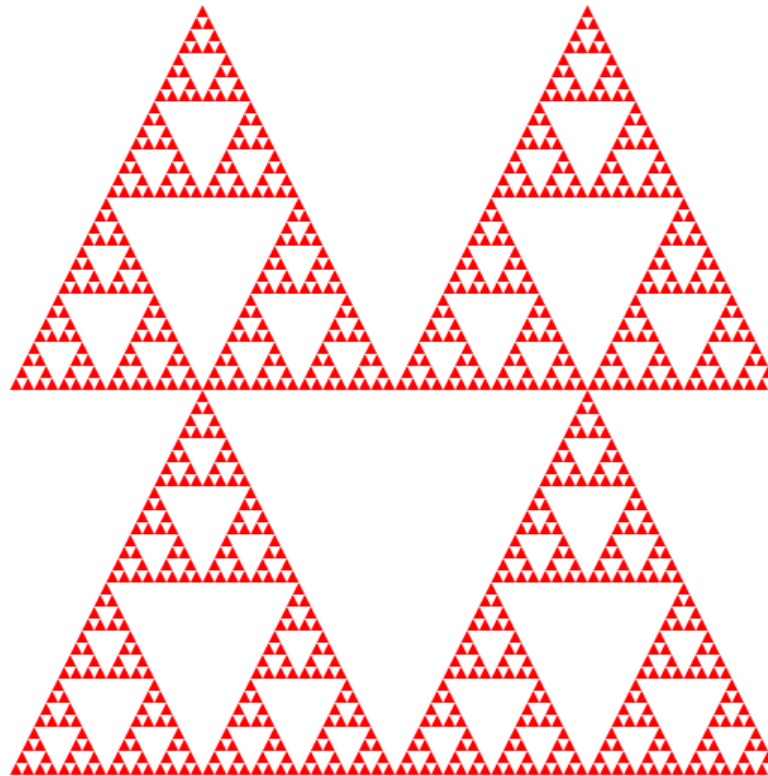
- Một số thủ thuật với khung nhìn
  - Tạo ra nhiều khung hình để vẽ nhiều hình vẽ giống nhau.
  - Thay đổi chiều rộng và chiều cao của khung hình để co kéo hình vẽ bên trong.



## Ví dụ 3



- Sử dụng khung nhìn, vẽ 4 tam giác Sierpinski ở 4 góc của màn hình như sau

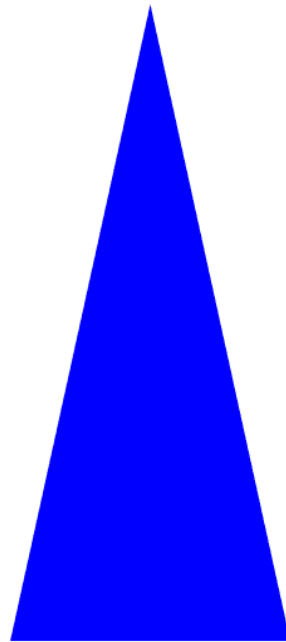


## Ví dụ 4



- Viết chương trình thay đổi hình dạng của tam giác bằng cách thay đổi chiều rộng và chiều cao của khung nhìn.

Width: 1  512  
Height: 1  512

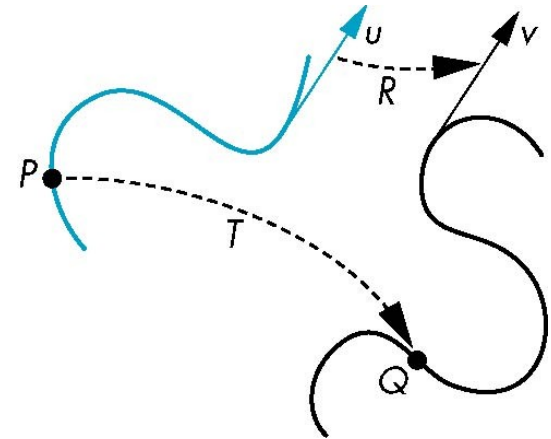


# Các phép biến đổi



- Các phép biến đổi (transformation) **thực hiện ánh xạ** một điểm đến một điểm khác hoặc một véc-tơ đến một véc-tơ khác.

- $Q = T(P)$
- $v = R(u)$

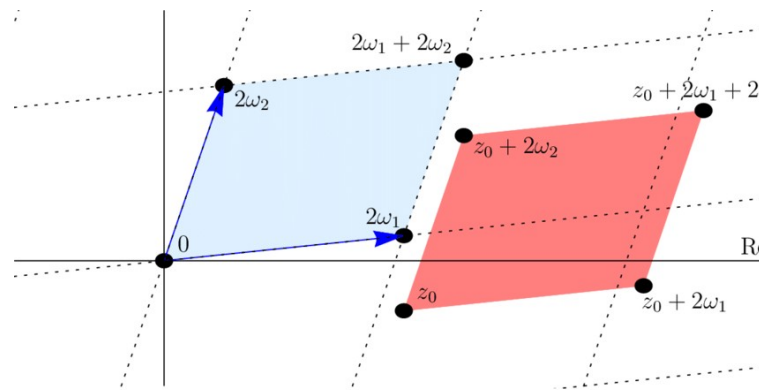


- **T** là một ánh xạ nào đó được dùng để biến đổi điểm P thành điểm Q
- **R** là một ánh xạ nào đó được dùng để biến đổi véc-tơ u thành véc-tơ v

# Phép biến đổi Affine



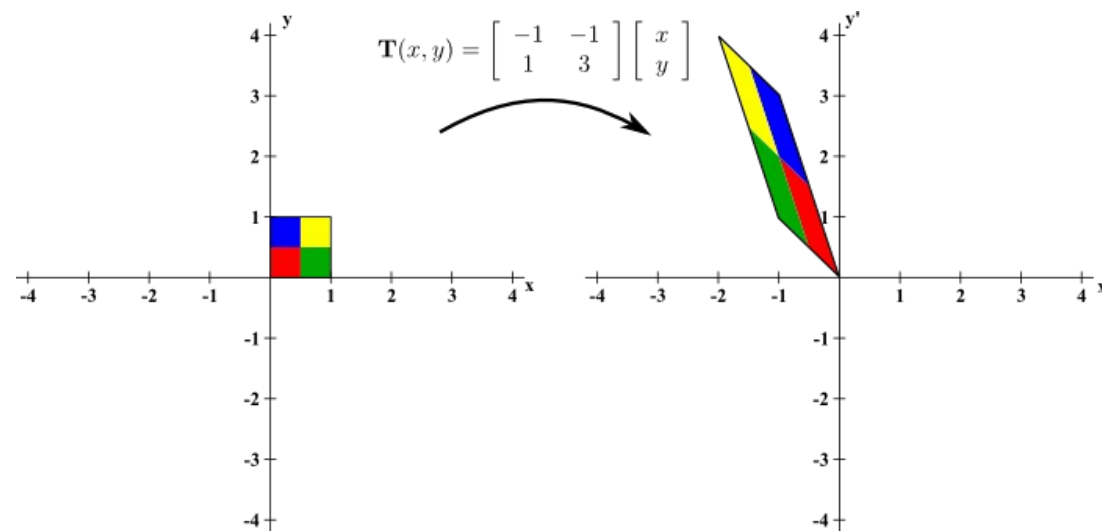
- Một phép biến đổi Affine là một phương pháp ánh xạ tuyến tính bảo toàn điểm, đường thẳng và mặt phẳng.
  - Tập hợp các đường thẳng song song sẽ vẫn song song với nhau sau phép biến đổi affine.
    - Ví dụ: Hình bình hành sau phép biến đổi affine thì vẫn là hình bình hành.



# Phép biến đổi Affine



- Các tỷ lệ trong phép biến đổi affine cũng được bảo toàn
  - Ví dụ: Trung điểm của các đoạn thẳng sau phép biến đổi affine vẫn sẽ là trung điểm của đoạn thẳng mới.

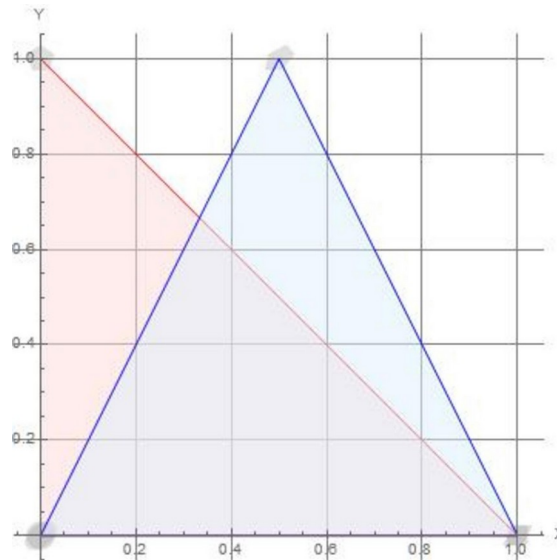




# Phép biến đổi Affine



- Tuy nhiên phép biến đổi Affine **không nhất thiết bảo toàn góc và độ dài**.
  - Ví dụ: Một tam giác sau phép biến đổi affine vẫn là một tam giác, nhưng có thể không đồng dạng với tam giác gốc.



# Phép biến đổi Affine



- Giả sử trên không gian 2 chiều ta một điểm  $P$  có tọa độ  $(P_x, P_y)$ .
- Thông qua một phép biến đổi  $T$  nào đó, ta cần biến đổi điểm  $P$  thành một điểm  $Q$  có tọa độ là  $(Q_x, Q_y)$
- Trong không gian Affine, tọa độ của  $Q$  là tổ hợp tuyến tính của tọa độ  $P$ .

$$Q_x = m_{11}P_x + m_{12}P_y + m_{13}$$

$$Q_y = m_{21}P_x + m_{22}P_y + m_{23}$$

# Phép biến đổi Affine



- Có 2 cách biểu diễn dựa trên ma trận để thể hiện mối quan hệ giữa P và Q

- Cách 1:

$$\begin{pmatrix} Q_x \\ Q_y \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} P_x \\ P_y \end{pmatrix} + \begin{pmatrix} m_{13} \\ m_{23} \end{pmatrix}$$

- Cách 2:

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$

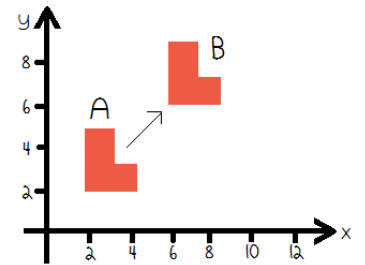
- Trong khóa học này ta chọn cách thứ 2.

# Phép biến đổi hình 2D



- Phép tịnh tiến

$$\begin{aligned} Q_x &= P_x + m_{13} \\ Q_y &= P_y + m_{23} \end{aligned} \quad \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & m_{13} \\ 0 & 1 & m_{23} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



- Ví dụ: Dịch chuyển một điểm theo trục hoành 5 đơn vị và trục tung 2 đơn vị ta sử dụng ma trận sau

$$\begin{pmatrix} 1 & 0 & 5 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

## Ví dụ 5



- Vẽ một chữ T lên màn hình, sau đó **tịnh tiến chữ T** bằng việc thiết lập các giá trị dịch chuyển cho trục tung và trục hoành.

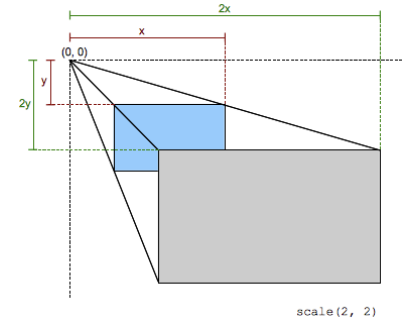


# Phép biến đổi hình 2D



- Phép biến đổi tỉ lệ

$$\begin{aligned} Q_x &= S_x P_x \\ Q_y &= S_y P_y \end{aligned} \quad \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



- Ví dụ: Biến đổi tỷ lệ một hình theo tỷ lệ đối với trục hoành là 1.5 và với trục tung là 2.1 ta sử dụng ma trận sau

$$\begin{pmatrix} 1.5 & 0 & 0 \\ 0 & 2.1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

## Ví dụ 6



- Vẽ một chữ T lên màn hình, sau đó **biến đổi tỷ lệ chữ T** bằng việc thiết lập các giá trị tỷ lệ cho trục tung và trục hoành.
  - Chú ý: Với giá trị nằm trong khoảng  $(0, 1)$ , hình sẽ bị thu nhỏ.

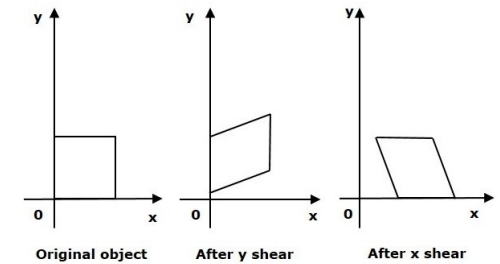


# Phép biến đổi hình 2D



- Phép biến dạng

$$\begin{aligned} Q_x &= P_x + R_x P_y \\ Q_y &= R_y P_x + P_y \end{aligned} \quad \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & R_x & 0 \\ R_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



- Ví dụ: Biến dạng một hình theo trục hoành là 1.5 và với trục tung là -1.5 ta sử dụng ma trận sau

$$\begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \mathbf{1.5} & 0 \\ \mathbf{-1.5} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



## Ví dụ 7



- Vẽ một chữ T lên màn hình, sau đó **biến dạng chữ T** bằng việc thiết lập các giá trị biến dạng cho trục tung và trục hoành.

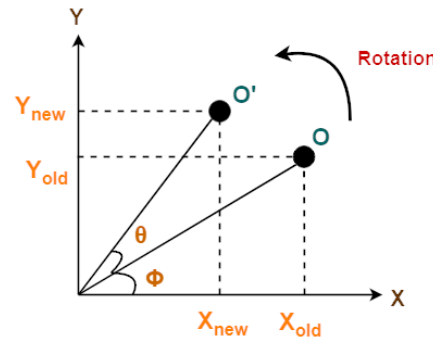


# Phép biến đổi hình 2D



- Phép quay quanh gốc tọa độ

$$\begin{aligned} Q_x &= \cos(\theta)P_x - \sin(\theta)P_y \\ Q_y &= \sin(\theta)P_x + \cos(\theta)P_y \end{aligned} \quad \begin{pmatrix} Q_x \\ Q_y \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_x \\ P_y \\ 1 \end{pmatrix}$$



- Ví dụ: Quay một điểm quanh gốc tọa độ một góc 45 độ ta sử dụng ma trận sau

$$\begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ \sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

# Phép biến đổi hình 2D



- Phép quay quanh gốc tọa độ

- Chứng minh

- Ta có:

$$\cos(\theta + \phi) = Q_x / r \text{ và } \sin(\theta + \phi) = Q_y / r$$

$$\cos(\phi) = P_x / r \text{ và } \sin(\phi) = P_y / r$$

- Tức là:

$$Q_x = r * \cos(\theta + \phi) \text{ và } Q_y = r * \sin(\theta + \phi) \quad (1)$$

$$P_x = r * \cos(\phi) \text{ và } P_y = r * \sin(\phi) \quad (2)$$

- Từ (1) ta có :

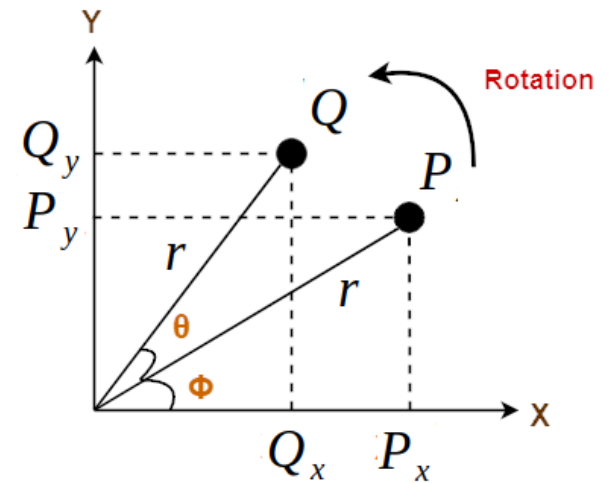
$$Q_x = r \cos(\theta) \cos(\phi) - r \sin(\theta) \sin(\phi) \quad (3)$$

$$Q_y = r \sin(\theta) \cos(\phi) + r \sin(\phi) \cos(\theta)$$

- Thế (2) vào (3) ta có:

$$- Q_x = \cos(\theta) P_x - \sin(\theta) P_y$$

$$- Q_y = \sin(\theta) P_x + \cos(\theta) P_y$$



## Ví dụ 8



- Vẽ một chữ T lên màn hình, sau đó **quay chữ T** quanh gốc tọa độ bằng việc thiết lập các góc quay.
  - Chú ý:
    - Thiết lập góc quay từ 0 độ cho đến 360 độ.
    - Sử dụng các hàm **Math.cos**, **Math.sin** và **Math.PI** có sẵn trong thư viện của Javascript.
      - Đầu vào của các hàm này là các tham số dạng Radian  
 $1 \text{ radian} = 180 \text{ độ} / \pi$   
Do đó:  $x \text{ radian} = y \text{ độ} * \pi / 180$

# Phép biến đổi hình 2D



- Phép biến đổi tổ hợp
  - Trong thực tế, ta thường phải thực hiện **các phép biến đổi phức tạp**, tuy nhiên mọi phép biến đổi phức tạp đều có thể được tạo thành từ các phép biến đổi cơ sở ở trên, do đó ta gọi đây là **phép biến đổi tổ hợp**.
    - Ví dụ:
      - Vừa di chuyển hình, vừa xoay hình.
      - Vừa xoay hình, vừa biến đổi tỉ lệ hình
      - Quay một hình quanh một điểm không phải là gốc tạo độ.

# Phép biến đổi hình 2D



- Phép biến đổi tổ hợp

- Giả sử ta có

- Điểm **X** qua phép biến đổi **f** ta nhận được điểm **Y**

- $Y = f(X) = f_{\text{matrix}} * X$

- Điểm **Y** qua phép biến đổi **g** ta nhận được điểm **Z**

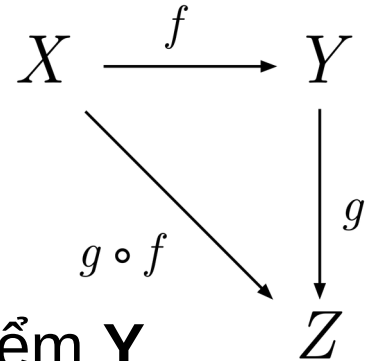
- $Z = g(Y) = g_{\text{matrix}} * X$

- Câu hỏi:

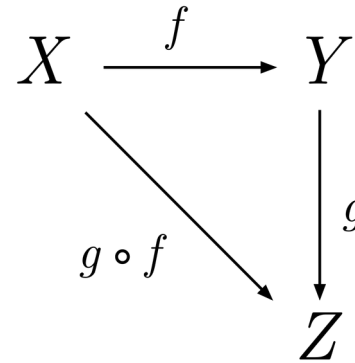
- **Có tồn tại phép biến đổi nào để biến đổi ngay lập tức điểm X thành điểm Z được không ?**

- Trả lời:

- **Có tồn tại.** Đó là hợp của hai phép biến đổi g và f



# Phép biến đổi hình 2D



- Phép biến đổi tổ hợp

- Ta cần phải **tìm ma trận biến đổi  $T_{\text{matrix}}$**  để ánh xạ điểm X thành điểm Z.

- Cách tìm:

- Ánh xạ  $f$  là ma trận biến đổi điểm X thành điểm Y.

- Ánh xạ  $g$  là ma trận biến đổi điểm Y thành điểm Z.

- Ta có:  $Z = g_{\text{matrix}} * Y = g_{\text{matrix}} * (f_{\text{matrix}} * X) = (g_{\text{matrix}} * f_{\text{matrix}}) * X$

- Vậy  $T_{\text{matrix}} = g_{\text{matrix}} * f_{\text{matrix}}$

- Kết luận:

- Ta thấy rằng để tính ma trận biến đổi  $T_{\text{matrix}}$  ta chỉ cần lấy ma trận ánh xạ  $g$  nhân với ma trận ánh xạ  $f$  (theo đúng thứ tự  $g$  đứng trước,  $f$  đứng sau)

# Phép biến đổi hình 2D



- Phép biến đổi tổ hợp
  - Phương pháp tìm phép biến đổi tổ hợp
    - B1: Phân tích cách thức biến đổi ra thành các phép biến đổi cơ bản.
      - Đã có ma trận cho các phép biến đổi cơ bản ở phần trên.
    - B2: Lấy ma trận của các phép biến đổi cơ bản nhân với nhau.
      - Lưu ý về trình tự: **Phép biến đổi cơ bản nào thực hiện trước thì việc nhân ma trận sẽ được thực hiện sau.**

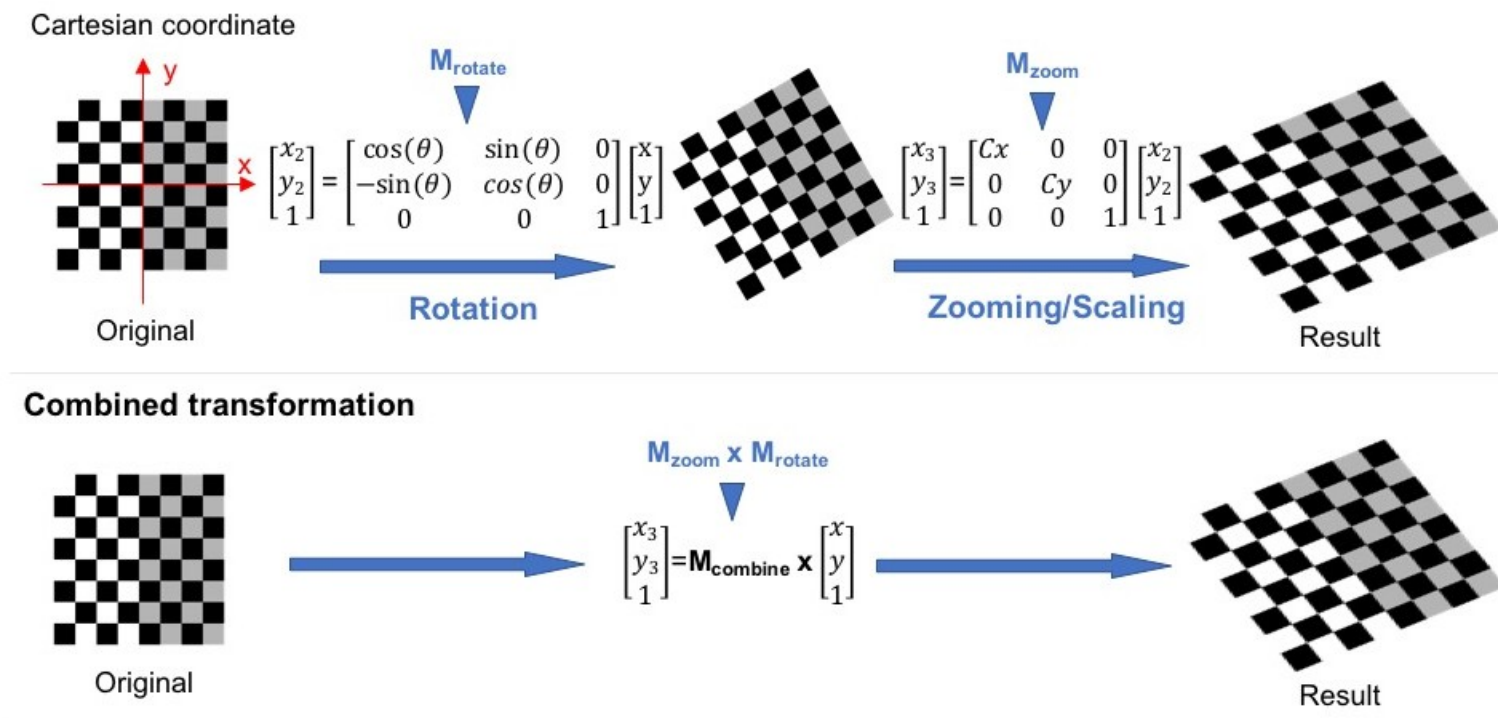




# Phép biến đổi hình 2D



- Phép biến đổi tổ hợp
  - Ví dụ 1:



# Phép biến đổi hình 2D

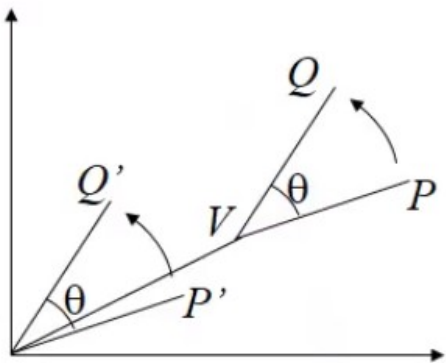


- Phép biến đổi tổ hợp

- Ví dụ 2: Xây dựng ma trận biến đổi để xoay một điểm  $P(P_x, P_y)$  quanh một điểm  $V(V_x, V_y)$  không phải gốc tọa độ.

- **Tìm ma trận tổ hợp T:**

- B1: Sau khi phân tích, cách thức biến đổi này có 3 phép biến đổi cơ bản.



- Thực hiện tịnh tiến điểm P theo véc-tơ  $v = (-V_x, -V_y)$  thành điểm  $P'$  (ký hiệu ma trận biến đổi là  $\mathbf{f}_{matrix}$ )
- Quay điểm  $P'$  quanh gốc tọa độ thành điểm  $Q'$  (ký hiệu ma trận biến đổi là  $\mathbf{g}_{matrix}$ )
- Thực hiện tịnh tiến điểm  $Q'$  véc-tơ  $v' = (V_x, V_y)$  ta được điểm Q (đây chính là điểm ta muốn điểm P xoay đến) (ký hiệu ma trận biến đổi này là  $\mathbf{h}_{matrix}$ )

# Phép biến đổi hình 2D

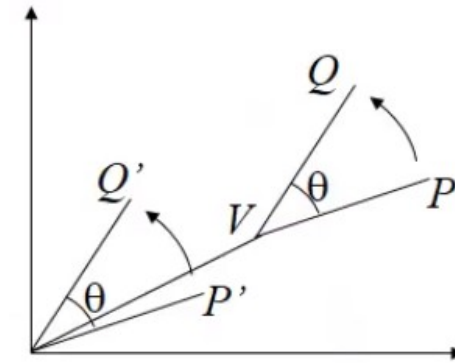


- Phép biến đổi tổ hợp

- Ví dụ 2:

- **Tìm ma trận tổ hợp T:**

- B2:  $T_{\text{matrix}} = h_{\text{matrix}} \times g_{\text{matrix}} \times f_{\text{matrix}}$



$$\begin{pmatrix} 1 & 0 & V_x \\ 0 & 1 & V_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -V_x \\ 0 & 1 & -V_y \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} \cos(\theta) & -\sin(\theta) & (1 - \cos(\theta)) * V_x + \sin(\theta) * V_y \\ \sin(\theta) & \cos(\theta) & -\sin(\theta) * V_x + (1 - \cos(\theta)) * V_y \\ 0 & 0 & 1 \end{pmatrix}$$

## Ví dụ 9



- Vẽ một chữ T lên màn hình, sau đó **quay chữ T** quanh một điểm cho trước bằng việc thiết lập các góc quay.
  - Chú ý:
    - Thiết lập góc quay từ 0 độ cho đến 360 độ.



# Hết Tuần 3



Cảm ơn các bạn đã chú ý lắng nghe !!!