

Đồ họa



Tuần 4

Giảng viên: Trần Đức Minh

Nội dung bài giảng




- Giới thiệu về HTML
- Giới thiệu về ngôn ngữ Javascript
- Cơ bản về xử lý sự kiện trong Javascript
- Các phương pháp nắm bắt sự kiện
- Sự kiện của các đối tượng cơ bản
- Các ví dụ



Giới thiệu về HTML



- **HTML** (Hypertext Markup Language) là ngôn ngữ đánh dấu siêu văn bản có sử dụng các **tag** để định nghĩa các thành phần bên trong trang Web.

**THANG LONG UNIVERSITY**

Ngành Khoa học máy tính

Khoa học máy tính là ngành học đóng vai trò rất quan trọng trong nghiên cứu và giải quyết các vấn đề lý luận cũng như các kỹ thuật nền tảng của CNTT và truyền thông. Ngành học này nghiên cứu những kiến thức cơ bản trong tin học nhằm mục đích áp dụng các thuật toán và toán học vào các bài toán trong thực tế. Cụ thể, những vấn đề được ngành học quan tâm nghiên cứu như: Nguyên lý làm việc của máy tính, biểu diễn cấu trúc của dữ liệu trong máy tính cùng các lý thuyết cơ bản trong lĩnh vực tin học, v.v. Các vấn đề này là cơ sở cho các ngành công nghệ thông tin - truyền thông khác.

Cử nhân khoa học máy tính là những người có hiểu biết rất cơ bản về các vấn đề nêu ở trên. Với kiến thức nền tảng của ngành học này, họ có đầy đủ tri thức để có thể tham gia nhiều mảng trong lĩnh vực tin học như nghiên cứu chuyên sâu, trở thành những nhà phát triển ứng dụng phần mềm chuyên nghiệp, những nhà phát triển các ứng dụng trí tuệ nhân tạo cũng như kiến trúc sư trưởng về hệ thống công nghệ thông tin và truyền thông, v.v.

Cử nhân khoa học máy tính do vậy là những người có kiến thức chuyên sâu và khả năng phân tích giải quyết vấn đề trong tin học một cách cơ bản. Họ có tư duy sâu sắc về hoạt động của máy tính, bao gồm cả phần cứng lẫn phần mềm, họ có khả năng tạo ra các sản phẩm phần mềm đáp ứng nhu cầu của thị trường một cách chuyên nghiệp.

Mã ngành
7340201

Thời gian học
04 năm

Tổ hợp môn thi
A00, A01

Học phí
22.000.000vnd/năm

Điểm trúng tuyển các năm gần đây (thang điểm 30)

Năm 2019	16.00
Năm 2018	15.00

Đối tác của TLU

view-source:https://thanglong.edu.vn/khoa-toan-tin-hoc/nganh-khoa-hoc-may-tinh

```
<li><a href="/khoa-toan-tin-hoc/nganh-cong-nghe-thong-tin">Ngành Công nghệ thông tin</a></li>
</div>
</div>
<div class="col-md-8 col-xl-6">
<div class="layout_content">
<h1 class="page-big-title d-none d-md-block">Ngành Khoa học máy tính</h1>
<div class="page-single-entry">
<div class="paragraph paragraph--type--section-text paragraph--view-mode--default">
<div class="clearfix text-formatted field field--name-field-paragraph-text field--type-text-long field--labe
"><p>Khoa học máy tính là ngành học đóng vai trò rất quan trọng trong nghiên cứu và giải quyết
truyền thông. Ngành học này nghiên cứu những kiến thức cơ bản trong tin học nhằm mục đích áp dụng các thuật to
để được ngành học quan tâm nghiên cứu như: Nguyên lý làm việc của máy tính, biểu diễn cấu trúc của dữ liệu tro
.v.v. Các vấn đề này là cơ sở cho các ngành công nghệ thông tin - truyền thông khác.</p>
</div>
<div class="clearfix text-formatted field field--name-field-paragraph-text field--type-text-long field--labe
"><p>Cử nhân khoa học máy tính là những người có hiểu biết rất cơ bản về các vấn đề nêu ở trên. Với kiến thức nê
nhiều mảng trong lĩnh vực tin học như nghiên cứu chuyên sâu, trở thành những nhà phát triển ứng dụng phần mề
cũng như kiến trúc sư trưởng về hệ thống công nghệ thông tin và truyền thông, .v.v.</p>
</div>
<div class="clearfix text-formatted field field--name-field-paragraph-text field--type-text-long field--labe
"><p>Cử nhân khoa học máy tính do vậy là những người có kiến thức chuyên sâu và khả năng phân tích giải quyết à
động của máy tính, bao gồm cả phần cứng lẫn phần mềm, họ có khả năng tạo ra các sản phẩm phần mềm đáp ứng nhu
</div>
</div>
<div class="paragraph paragraph--type--section-text paragraph--view-mode--default">
<div class="clearfix text-formatted field field--name-field-paragraph-text field--type-text-long field--labe
"><h2>Mục tiêu đào tạo</h2>
</div>
</div>
<div class="paragraph paragraph--type--section-text paragraph--view-mode--default">
<div class="clearfix text-formatted field field--name-field-paragraph-text field--type-text-long field--labe
"><p>Sau khi tốt nghiệp ngành Khoa học máy tính, sinh viên sẽ đạt được những yêu cầu sau:</p>
</div>
<div class="list-item">
<ul>
<li><li>Nắm vững các nguyên lý và lý luận về tính toán bên trong máy tính; Biểu diễn, quản lý dữ liệu và tri t
dụng vào các bài toán thực tế; Có hiểu biết sâu sắc về một số lĩnh vực trong tin học để có thể đưa ra giải ph
<li>Nắm vững một số ngôn ngữ lập trình phổ biến và hiện đại; Có kỹ năng lập trình vững vàng để giải quyết
các công cụ mới liên quan đến lĩnh vực lập trình. </li>
<li>Nắm bắt và vận dụng được các công cụ phát triển phần mềm để xây dựng được các sản phẩm phần mềm theo m
<li>Có khả năng tiếp cận các vấn đề liên quan đến trí tuệ nhân tạo theo cả hướng nghiên cứu chuyên sâu và
<li>Có đầy đủ tri thức để trở thành một kiến trúc sư trưởng về hệ thống thông tin; Tích hợp hệ thống, chủa
doanh nghiệp hay tổ chức trên nền tảng sử dụng các yếu tố công nghệ thông tin.</li>
<li>Thiết lập cũng như lập trình các hệ thống liên quan đến mạng máy tính; Giao tiếp tư x, hội thảo trực
</li>
</ul>
</div>
</div>
<div class="paragraph paragraph--type--section-text paragraph--view-mode--default">
```

Giới thiệu về ngôn ngữ Javascript



- Là một **ngôn ngữ lập trình kịch bản** (không đòi hỏi biên dịch trước khi thực thi) với cách viết tựa C.
- Mã lệnh của Javascript được **thực thi bởi Web Browser**.
- Được **sử dụng kết hợp với HTML** nhằm tăng hiệu quả thiết kế và hoạt động trên giao diện Web.



Giới thiệu về ngôn ngữ Javascript



- Cách nhúng Javascript vào HTML
 - **Cách 1**: Viết mã lệnh Javascript bên trong phần nội dung của tag

<script type="text/javascript"> ... </script>

```
<!DOCTYPE html>
<html>

<head>
  <title>Show warning</title>
</head>

<body>
  <button id="show-warning">Show</button>
</body>

<script type="text/javascript">

  let show_warning = function () {
    alert("Day la canh bao !!!");
  }

  let button = document.getElementById("show-warning");
  button.onclick = show_warning;

</script>

</html>
```

Giới thiệu về ngôn ngữ Javascript



- Cách nhúng Javascript vào HTML
 - **Cách 2**: Viết mã lệnh Javascript bên trong một **file có đuôi .js**, sau đó thiết **đặt giá trị của thuộc tính src** trong tag **script** là đường dẫn đến file này.

```
<script type="text/javascript" src="../../Common/initShaders.js"></script>  
<script type="text/javascript" src="../../Common/MVnew.js"></script>  
<script type="text/javascript" src="tlu.js"></script>
```

Cơ bản về xử lý sự kiện trong Javascript



- **Lập trình hướng sự kiện** (Event-driven programming) là mô hình trong đó luồng chương trình được **xác định bởi các hành động** của người sử dụng như nhấp chuột, nhấn phím, ...
- Trong ứng dụng hướng sự kiện, luôn có **một vòng lặp chạy liên tục** để nghe sự kiện. Khi phát hiện có một sự kiện nào đó xảy ra, hệ thống sẽ tự động gọi đến **hàm callback** liên quan đến sự kiện đó.

Event Loop

Callback Queue



onClick

onLoad

onDone

Cơ bản về xử lý sự kiện trong Javascript



- Hàm callback

- Trong javascript, hàm callback là một hàm bất kỳ, được truyền vào hàm khác như một tham số và hàm callback sẽ được thực hiện ở bên trong hàm đó.

- Ví dụ:

```
function f1() { ... }  
function f2(fcb) {  
    fcb();  
}
```

```
f2(f1);
```


Ví dụ 1



- Nghiên cứu các trường hợp xảy ra khi có và không có hàm callback.



Cơ bản về xử lý sự kiện trong Javascript



- Để xử lý sự kiện trong Javascript, ta cần hiểu 2 khái niệm sau
 - **Event handlers** là các **hàm do người lập trình tạo ra**, hàm này được gọi khi một sự kiện nào đó được kích hoạt.
 - Ví dụ: Hàm được gọi khi click chuột, hoặc nhấn một phím trên bàn phím, hoặc thay đổi kích thước cửa sổ, ...
 - Mỗi **event listener** là một thủ tục dùng để “**nghe**” (**đợi**) **sự kiện xuất hiện**.
 - Mỗi event listener **cần được đăng ký trước** để nắm bắt sự kiện.
 - Khi sự kiện xuất hiện event handler gắn với event listener sẽ được thực hiện.
-

Các phương pháp nắm bắt sự kiện



- **Phương pháp 1:** Gán trực tiếp event handler trong thuộc tính của đối tượng điều khiển
 - Không nên sử dụng cách này khi lập trình do khó quản lý code.
 - Code này trộn lẫn HTML và mã Javascript nên sẽ gây khó khăn trong việc phân tích và sửa đổi chương trình.

```
<!DOCTYPE html>
<html>

<head>
  <script type="text/javascript">
    function show_warning() {
      alert("Đây là cảnh báo !!!");
    }
  </script>
</head>
<body>
  <button onclick="show_warning()">Show</button>
</body>

</html>
```

Các phương pháp nắm bắt sự kiện



- **Phương pháp 2:** Gán event handler cho thuộc tính của đối tượng.

```
<!DOCTYPE html>
<html>

<head>
|   <title>Show warning</title>
</head>

<body>
|   <button id="show-warning">Show</button>
</body>

<script type="text/javascript">
|
|   let show_warning = function () {
|       alert("Day la canh bao !!!");
|   }
|
|   let button = document.getElementById("show-warning");
|
|   button.onclick = show_warning;
|
</script>

</html>
```

Các phương pháp nắm bắt sự kiện



- **Phương pháp 3:** Đối với mỗi sự kiện, sử dụng hàm **addEventListener()** để đăng ký event listener, đồng thời gắn kèm event handler để khi sự kiện được kích hoạt, event handler gắn kèm sẽ tự động thực hiện.

```
<body>
  <button id="show-warning">Show</button>
</body>

<script type="text/javascript">

  let show_warning = function () {
    alert("Đây là cảnh báo !!!");
  }

  let button = document.getElementById("show-warning");

  button.addEventListener("click", show_warning);

</script>
```

Các phương pháp nắm bắt sự kiện



- **Phương pháp 3:**

- Ta có thể gắn kèm nhiều event handler vào một event listener. Khi sự kiện được kích hoạt, tất cả event handler sẽ cùng được thực hiện.

```
<body>
  <button id="show-warning">Show</button>
</body>

<script type="text/javascript">

  let show_warning = function () {
    alert("Day la canh bao !!!");
  }

  let show_comment = function () {
    alert("Day la chu thích !!!");
  }

  let button = document.getElementById("show-warning");

  button.addEventListener("click", show_warning);
  button.addEventListener("click", show_comment);

</script>
```

Các phương pháp nắm bắt sự kiện



- **Phương pháp 3:**

- Sử dụng hàm **removeEventListener()** để xóa event handler của một event listener.

```
<body>
  <button id="show-warning">Show</button>
</body>

<script type="text/javascript">

  let show_warning = function () {
    alert("Day la canh bao !!!");
  }

  let show_comment = function () {
    alert("Day la chu thich !!!");
  }

  let button = document.getElementById("show-warning");

  button.addEventListener("click", show_warning);
  button.addEventListener("click", show_comment);

  button.removeEventListener("click", show_comment);

</script>
```

Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với **đối tượng Button**

- Tạo Button trong file HTML

`<button id="idButton">Thông tin hiển thị</button>`

Thông tin hiển thị

- Sử dụng tag có tên là **button**
 - id của button là **idButton** được sử dụng để nhận diện Button
 - Dòng chữ **Thông tin hiển thị** sẽ được hiển thị bên trên Button

Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với đối tượng Button
 - Xử lý **sự kiện click chuột** của Button

- Cách 1:

```
document.getElementById("idButton").onclick = function () {  
    .....  
};
```

- Cách 2:

```
var myButton = document.getElementById("idButton");  
  
myButton.addEventListener("click", function() {  
    .....  
});
```

Ví dụ 2



- Viết chương trình cho chữ T xoay liên tục quanh gốc tọa độ. Sử dụng 2 Button làm nhiệm vụ:
 - Điều khiển chữ T xoay theo chiều kim đồng hồ.
 - Điều khiển chữ T xoay ngược chiều kim đồng hồ.



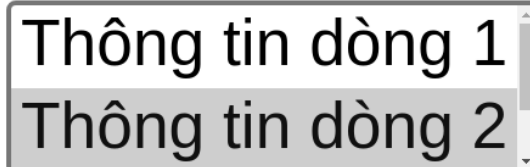
Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với **đối tượng Select**

- Tạo đối tượng Select trong file HTML

```
<select id="idSelect" size="2">  
  <option value="0">Thông tin dòng 1</option>  
  <option value="1">Thông tin dòng 2</option>  
  <option value="2">Thông tin dòng 3</option>  
</select>
```



- Sử dụng tag có tên là **select**
- Đối tượng Select sẽ chỉ hiển thị trực tiếp 2 dòng do size=2
 - Nếu để **size=1** hoặc **không thiết lập thuộc tính size** thì đối tượng select sẽ chuyển sang dạng **đối tượng combobox**

Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với đối tượng Select
 - Xử lý **sự kiện click chuột** của Select

```
var mySelect = document.getElementById("idSelect");
```

```
mySelect.addEventListener("click", function() {  
    switch(mySelect.selectedIndex) { // Lấy chỉ số dòng  
        case ...  
    }  
  
    // Lấy giá trị của chỉ số dòng  
    switch(mySelect.options[mySelect.selectedIndex].value) {  
        case ...  
    }  
});
```

Ví dụ 3



- Dựa trên Ví dụ 2, thêm một hộp thoại lựa chọn để điều khiển tốc độ quay của chữ T. Giả sử ta có 5 tốc độ:
 - Quay siêu nhanh
 - Quay nhanh
 - Quay vừa
 - Quay chậm
 - Quay siêu chậm



Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với bàn phím

- Xử lý **sự kiện keydown** của bàn phím

```
document.addEventListener("keydown", function(event) {  
    switch(event.key) {  
        case "ArrowLeft": // Phím mũi tên bên trái  
            .....  
        case "1": // Phím số 1  
            .....  
        case "a": // Phím chữ a (cả hoa và thường)  
            .....  
    }  
});
```



Ví dụ 4



- Dựa trên Ví dụ 3, điều khiển hướng quay và tốc độ của chữ T dựa trên bàn phím. Giả sử ta định nghĩa các phím như sau:
 - Phím mũi tên trái và phải trên bàn phím điều khiển việc quay trái và phải.
 - Phím mũi tên lên và xuống trên bàn phím điều khiển việc tốc độ quay.
- **Chú ý:** Sử dụng **hàm console.log** để in thông tin của các phím chức năng khác vào cửa sổ console của browser mỗi khi được nhấn.

Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với **đối tượng Slider**

- Tạo đối tượng Slider trong file HTML

```
<input id="idSlider" type="range" min="0" max="100" step="10" value="50" />
```



- Sử dụng tag có tên là **input** cùng với thuộc tính **type="range"**
- Thuộc tính **min** và **max** thể hiện **khoảng giá trị** của Slider
- Thuộc tính **step** là **giá trị mỗi bước di chuyển** của Slider
- Thuộc tính **value** là **giá trị ban đầu** của Slider

Làm việc với sự kiện



- Xử lý sự kiện với đối tượng Slider
 - Xử lý **sự kiện thay đổi giá trị** của Slider

```
var mySlider = document.getElementById("idSlider");
```

```
mySlider.addEventListener("change", function() {  
    var value = parseFloat(event.target.value);  
});
```



Ví dụ 5



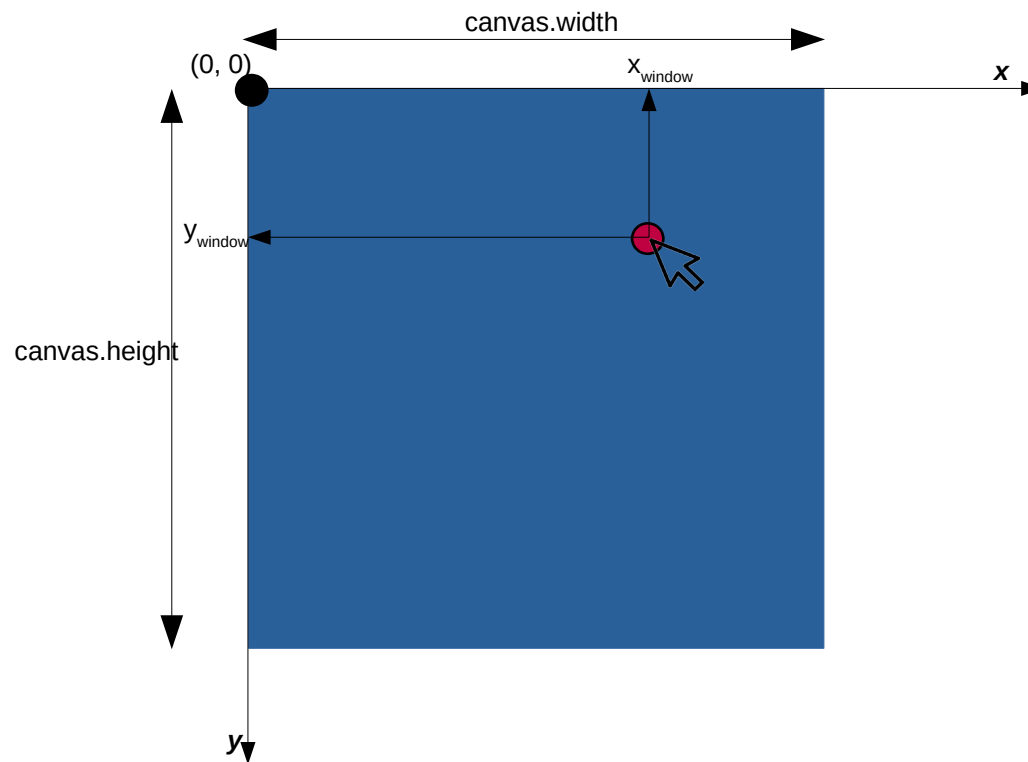
- Dựa trên Ví dụ 4, thêm vào chương trình một Slider làm nhiệm vụ phóng to, thu nhỏ chữ T trong khi đang quay.
- **Chú ý:** Giá trị trong khoảng $(0, 1)$ là thu nhỏ chữ T.



Sự kiện của các đối tượng cơ bản



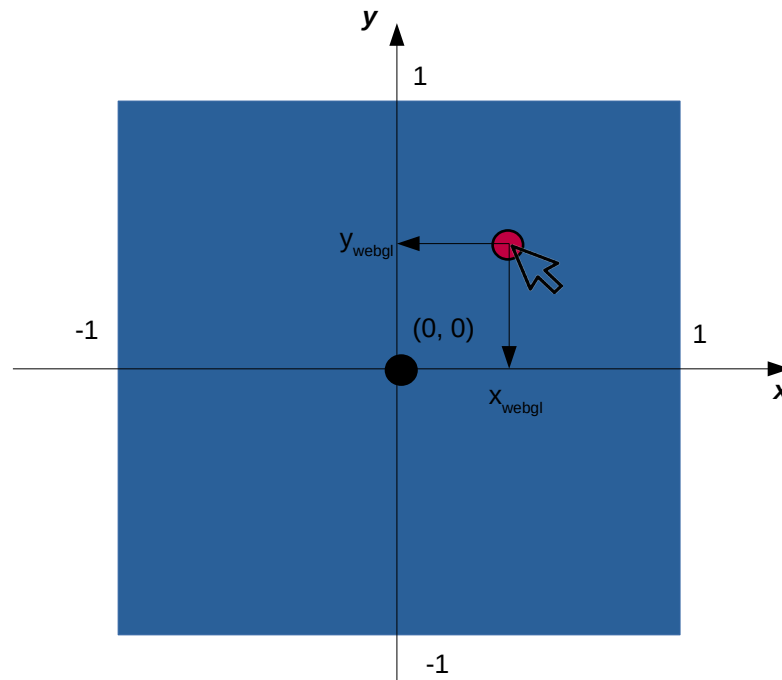
- Trục tọa độ cửa sổ
 - Tọa độ con trỏ chuột trên trục tọa độ cửa sổ



Biến đổi tọa độ con trỏ chuột



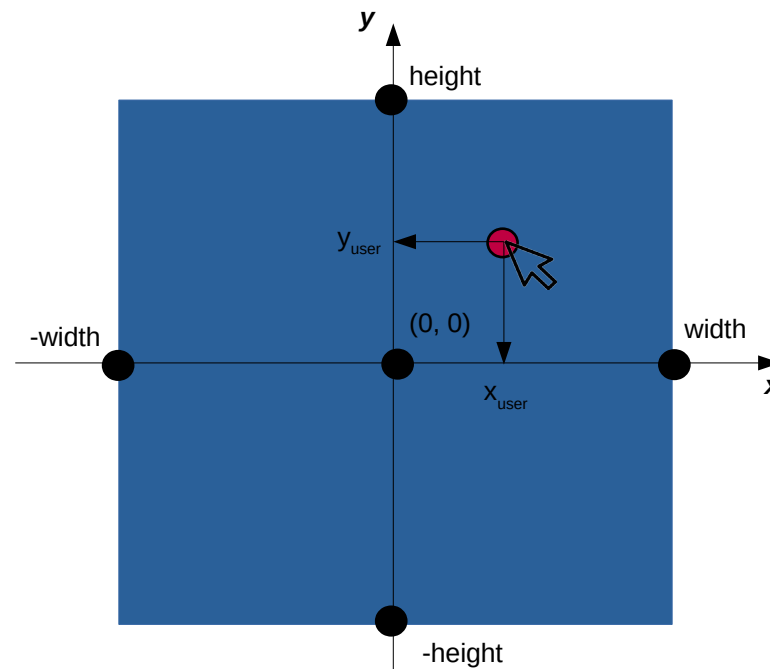
- Trục tọa độ của WebGL
 - Tọa độ con trỏ chuột trên trục tọa độ của WebGL



Biến đổi tọa độ con trỏ chuột



- Trục tọa độ của người sử dụng
 - Tọa độ con trỏ chuột trên trục tọa độ của người sử dụng



Biến đổi tọa độ con trỏ chuột



- Ta cần **biến đổi tọa độ con trỏ chuột**
 - Từ trục tọa độ cửa sổ sang trục tọa độ WebGL.
 - Từ trục tọa độ cửa sổ sang trục tọa độ của người sử dụng.
- Ta có **công thức biến đổi tọa độ từ trục tọa độ cửa sổ sang trục tọa độ WebGL** như sau:

$$x_{webgl} = -1 + \frac{2 * x_{window}}{width_{canvas}}$$

$$y_{webgl} = -1 + \frac{2 * (height_{canvas} - y_{window})}{height_{canvas}}$$

Biến đổi tọa độ con trỏ chuột



- Ta có công thức biến đổi tọa độ từ trục tọa độ người sử dụng sang trục tọa độ WebGL như sau:

$$x_{webgl} = \frac{x_{user}}{width_{canvas}}$$

$$y_{webgl} = \frac{y_{user}}{height_{canvas}}$$



Biến đổi tọa độ con trỏ chuột



- Để biến đổi tọa độ từ trục tọa độ cửa sổ sang trục tọa độ người sử dụng ta rút gọn công thức sau (được rút ra từ 2 công thức trên):

$$\frac{x_{user}}{width_{canvas}} = -1 + \frac{2 * x_{window}}{width_{canvas}}$$

$$\frac{y_{user}}{height_{canvas}} = -1 + \frac{2 * (height_{canvas} - y_{window})}{height_{canvas}}$$

Biến đổi tọa độ con trỏ chuột



- Sau khi rút gọn ta có công thức biến đổi tọa độ từ trục tọa độ cửa sổ sang trục tọa độ người sử dụng như sau:

$$x_{user} = 2 * x_{window} - width_{canvas}$$

$$y_{user} = height_{canvas} - 2 * y_{window}$$



Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với chuột
 - Lấy vị trí click chuột **trên trục tọa độ cửa sổ**

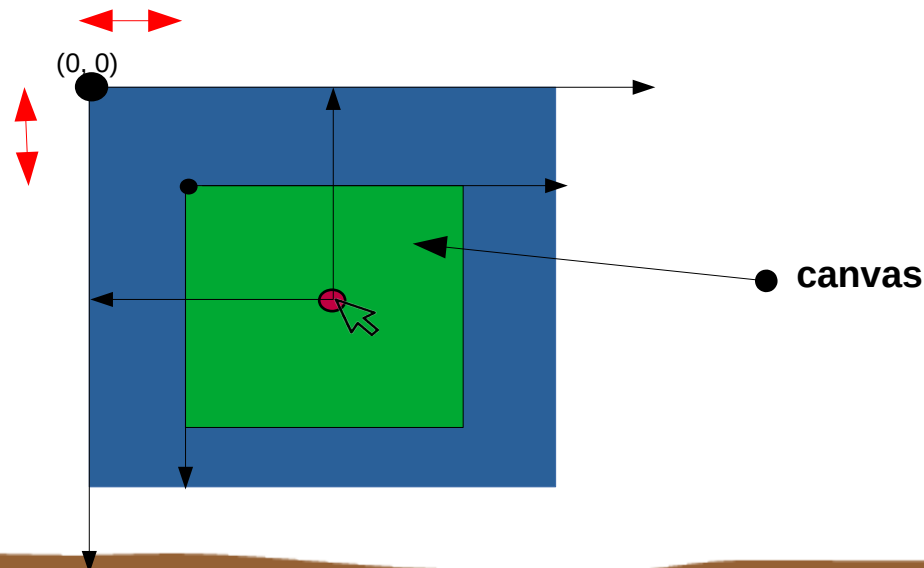
```
canvas.addEventListener("click", function(event) {  
    var x = event.clientX; // Tọa độ x của điểm click  
    var y = event.clientY; // Tọa độ y của điểm click  
});
```



Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với chuột
 - Do vùng hiển thị của canvas nằm bên trong nội dung của browser, do đó để xác định chính xác vị trí click chuột bên trong canvas, ta **cần phải hiệu chỉnh lại tọa độ** bằng cách trừ đi khoảng cách giữa cạnh canvas và mép nội dung của browser.



Ví dụ 6



- Viết chương trình mỗi khi click chuột lên màn hình, chương trình sẽ hiển thị một điểm ảnh màu đỏ tại vị trí vừa click chuột.



Sự kiện của các đối tượng cơ bản



- Xử lý sự kiện với cửa sổ
 - Xử lý **sự kiện resize** (thay đổi kích thước) của cửa sổ

```
window.addEventListener("resize", function() {  
    var newWidth = innerWidth; // Độ rộng mới của cửa sổ  
    var newHeight = innerHeight; // Chiều cao mới của cửa sổ  
});
```



Ví dụ 7



- Dựa trên ví dụ 5, viết chương trình xử lý sự kiện resize cửa sổ sao cho mỗi lần thực hiện resize, viewport tự động điều chỉnh chiều rộng và chiều cao của vùng vẽ bằng $\frac{1}{2}$ chiều cao của cửa sổ.



Hết Tuần 4



Cảm ơn các bạn đã chú ý lắng nghe !!!