

## Programmation web en JS :

**Paul-Antoine BISGAMBIGLIA**

*Faculty of Science*

UMR-CNRS-6134

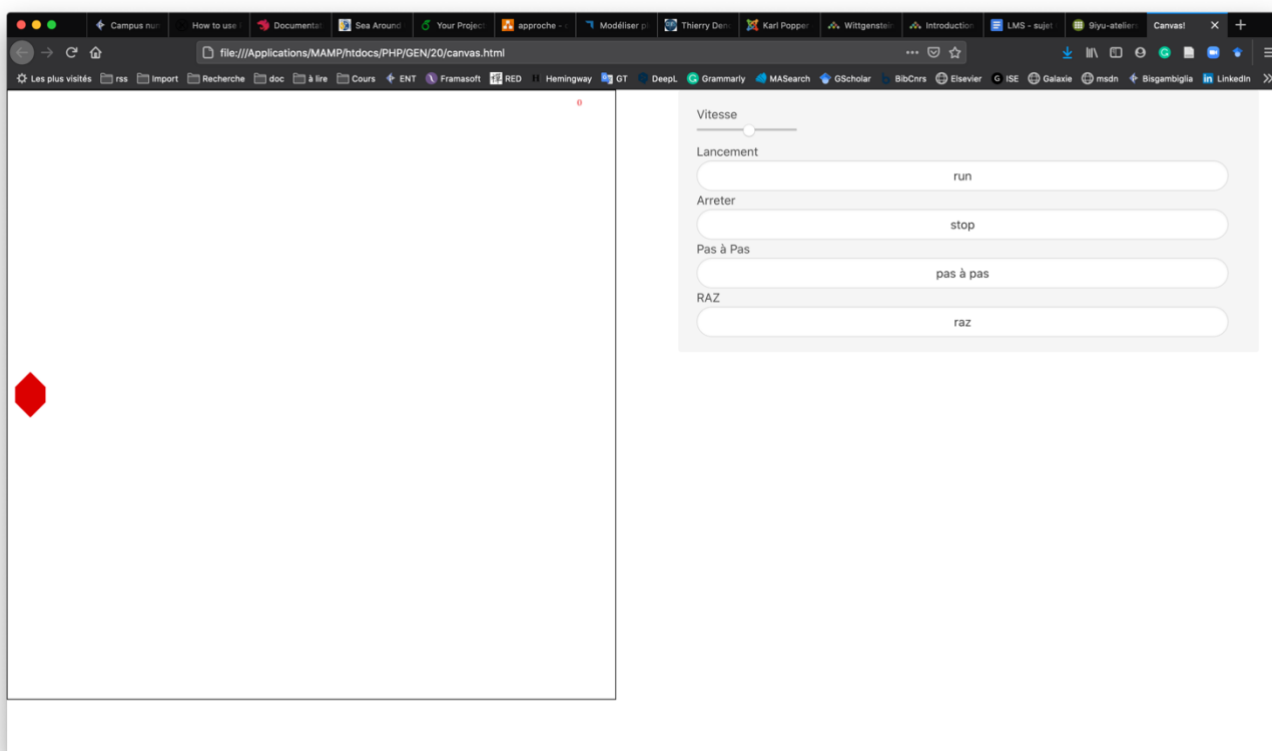
University of Corsica – 20250 Corte

*bisgambiglia@univ-corse.fr*

*Mots-clés : JavaScript, JSON, IHM, CANVAS, ANIMATION*

Vous allez devoir réaliser une petite interface et ajouter un canvas qui permet de faire évoluer un objet.

Voici un exemple d'interface :

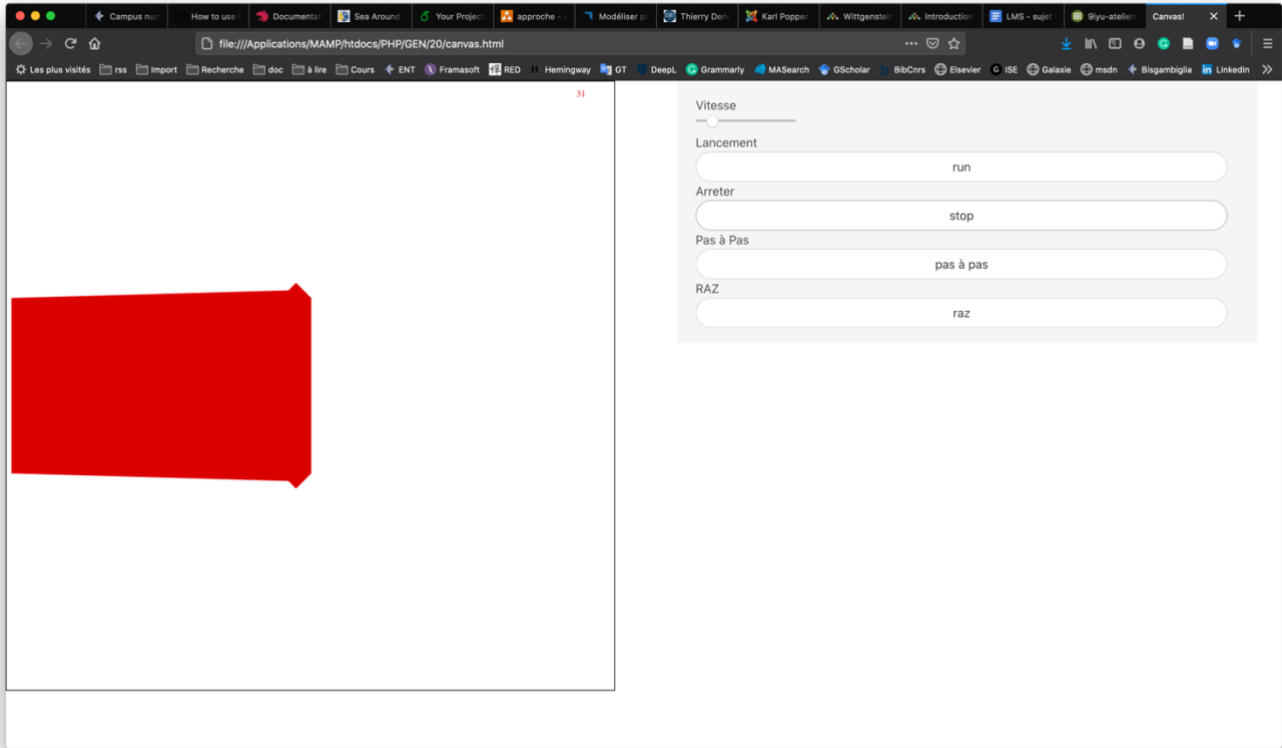


Elle contient :

- Un canvas
- Un menu avec :
  - o input type="range"
  - o 4 input type="submit"
- **Vitesse** permet de gérer la vitesse de l'animation.
- **Run** de lancer l'animation en mode automatique
- **Stop** d'arrêter l'animation.

- **Pas de pas** de lancer l'animation en mode pas à pas
- **Raz** de replacer le composant à sa place initiale.

En haut à droite du bord du canvas il y a un compteur de temps ou d'itération.



Tout doit être fait en javascript.

Ici la forme des composants est basée sur le framework CSS bulma.

1. Vous devez réaliser la même interface.
2. Vous devez ajouter une rose des vents qui permet de fixer une direction à l'animation.

Le code ci-dessous est donné à titre indicatif, pour vous aider si vous avez des difficultés, il est préférable de réaliser ce TP sans aide et surtout sans copier-coller trouvés sur le web :

```
<!DOCTYPE HTML>
<html lang="fr">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta charset="UTF-8">
    <title>Canvas!</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.8.0/css/bulma.min.css">
    <script defer src="https://use.fontawesome.com/releases/v5.3.1/js/all.js"></script>
    <title>Mon premier document HTML</title>
  </head>
  <body>
```

```

<div class="columns is-desktop">
  <div class="column">
    <canvas id="myCanvas" width="800" height="800" style="border:1px solid #000000;"></canvas>
  </div>
  <div class="column">
    <div class="container is-fluid">
      <div class="notification">
        <label for="vitesse">Vitesse</label><br>
        <input type="range" id="vitesse" name="vitesse" min="1" max="20"><br>
        <label for="run">Lancement</label><br>
        <input type="submit" id="run" name="run" value="run" class="input is-rounded" /><br>
        <label for="stop">Arreter</label><br>
        <input type="submit" id="stop" name="stop" value="stop" class="input is-rounded" /><br>
        <label for="pap">Pas à Pas</label><br>
        <input type="submit" id="pap" name="pap" value="pas à pas" class="input is-rounded" /><br>
        <label for="pap">RAZ</label><br>
        <input type="submit" id="raz" name="raz" value="raz" class="input is-rounded" /><br>

      </div>
    </div>
  </div>
</div>

</body>
<script>
document.querySelector('#raz').addEventListener('click', function (event) {
  window.location.reload();
},false);

document.querySelector('#vitesse').addEventListener('click', function (event) {
  speed = parseInt(document.querySelector('#vitesse').value);
},false);

document.querySelector('#stop').addEventListener('click', function (event) {
  clearInterval(nIntervId);
},false);

document.querySelector('#run').addEventListener('click', function (event) {
  nIntervId = setInterval(function(){
    myTime += 1;
    move(xPoint,yPoint,can,xt=speed,yt=speed,time=myTime);
  },900);
},false);

```

```
document.querySelector("#pap").addEventListener('click', function (event) {  
    myTime += 1;  
    move(xPoint,yPoint,can,xt=speed,yt=speed,time=myTime)  
},false);
```

```
var nIntervId;  
var speed = 10;  
var myTime = 0;  
const xPoint = [10,20,30,40,50,50,40,30,20,10];  
const yPoint = [390,380,370,380,390,410,420,430,420,410];  
const can = document.getElementById("myCanvas");  
initial(xPoint,yPoint,can);
```

```
function xAjust(x,speed)  
{  
    let output = x;  
    if (x<=10)  
    {  
        output -=speed*(1/100);  
    }  
    else  
    {  
        output +=speed;  
    }  
    return output;  
}
```

```
function yAjust(y,speed)  
{  
    let output = y;  
    if (y<=400)  
    {  
        output -=speed*(3/10);  
    }  
    else  
    {  
        output +=speed*(3/10);  
    }  
    return output;  
}
```

```
function initial(px,py,can,time=0)
```

```

{
  let ctx=can.getContext("2d");
  ctx.font="50px Arial";
  ctx.fillStyle = "rgb(200,0,0)";
  ctx.font = '12px serif';
  ctx.fillText(time, 750, 20);
  ctx.beginPath();
  ctx.moveTo(px[0],py[0]);
  for (let index = 1; index < px.length; index++) {
    ctx.lineTo(px[index],py[index]);
  }
  ctx.fill();
}

function move(px,py,c,xt,yt,time=0)
{
  console.log("execution");
  var xPoint = px;
  var yPoint = py;
  for (let index = 0; index < xPoint.length; index++) {
    xPoint[index] = xAjust(xPoint[index],xt);
    yPoint[index] = yAjust(yPoint[index],yt);
  }
  let ctx=c.getContext("2d");
  ctx.font = '12px serif';
  ctx.clearRect(0, 0, 800, 800);
  ctx.fillText(time, 750, 20);
  ctx.fillStyle = "rgb(200,0,0)";
  ctx.beginPath();
  ctx.moveTo(xPoint[0],yPoint[0]);
  for (let index = 1; index < xPoint.length; index++) {
    ctx.lineTo(xPoint[index],yPoint[index]);
  }
  ctx.fill();
}
</script>
</html>

```