

La vache est dans le pré.

PHILIPPE Lucas & BENBETKA Logan dans le groupe « Unix > Windaube ».

Pas de commande de compilation nécessaire pour générer l'exécutable, tout se passe sur le site web avec les entrées du nombre de piquets et des coordonnées et cliquer sur un bouton.

Lien github :

<https://github.com/Xurama/unix-windaube/>

Lien du site :

<https://xurama.github.io/unix-windaube/>

Conclusion :

Ce projet fut extrêmement intéressant d'un point de vue mathématiques et logique, cela nous a permis de nous faire travailler nos ménages. Le fait de nous « forcer » à faire des fonctions de 15 lignes maximum nous a permis de prendre une bonne habitude à faire des fonctions courtes et claires avec un seul objectif dans chaque fonction. Nous avons rencontré des difficultés par rapport aux déclarations de variables avec Javascript. En effet, on ne peut pas initialiser des variables en float ou en int. De ce fait on a rencontré des problèmes de compréhension lors de retour de variable comme le centre de gravité qui renvoyait « NaN ». Du fait qu'on travaillait en groupe, nous avons dû forcer encore plus sur la clarté des variables pour qu'on puisse se comprendre lors d'ajouts de code. Nous devons au départ, réaliser l'affichage sur un serveur discord via un bot. Ayant eu énormément de complications, nous sommes restés sur l'idée de faire le projet en javascript mais via un site internet.

Annexe :

Fichier projet.html :

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPPrVMTNDbiYZCxYbOOI7+AMvyTG2x" crossorigin="anonymous">
  <link href="design.css" rel="stylesheet" type="text/css">
</head>
<body>
  <h1 class="text-center">Projet tutoré. La vache est dans le pré.</h1>
  <h2 class="text-center">PHILIPPE Lucas - BENBETKA Logan</h2>
  <form class="text-center">
    <label for="name">Nombre de piquets ( max : 50 ) </label>
    <input type="text" id="nbPiquets" name="name" required size="10">
    <br><br>
    <div class="container"></div>
    <button type="button" class="btn btn-primary btn-lg text-center" value="Rechercher" id="boutonEnvoi" OnClick
    = "nombrePiquets()">Appuyez</button>
    <button type="button" class="btn btn-primary btn-lg text-center" value="Calcul" id="boutonEnvoiCalcul" OnClick
    = "main()" style="display:none;">Calculez</button>
    <button type="button" class="btn btn-primary btn-lg text-center" value="Reset" id="boutonReset" OnClick="res
    et()" style="display:none;">Reset</button>
  </form>

  <script src="http://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></script>
  <script type="text/javascript">
    function nombrePiquets(){
      const nbPiquets = document.getElementById("nbPiquets").value;
      const myBtnSearch = document.getElementById("boutonEnvoi");
      const myBtnCalcul = document.getElementById("boutonEnvoiCalcul");
      conditionNombrePiquets(nbPiquets, myBtnSearch, myBtnCalcul);
    }

    function conditionNombrePiquets(nbPiquets, myBtnSearch, myBtnCalcul){
      if(nbPiquets>50)
        $(".container").append(
          "<p> Vous n'avez pas le droit de prendre plus de 50 piquets</p>");
      else{
        $(".container").append(
          "<div>" + "<p>Donnez les coordonnées ( x et y ) de chaque piquets</p>");
        for(i=0; i<=nbPiquets-1;i++)
          $(".container").append(
            "<div class='marginBottom'> <label " + " for='piquet-" + i + "'>Piquet n°" + (i+1) + " : </label>" +
            "<input name=arrayX[] placeholder=Coordonnées-X " + (i+1) + ">" +
            "<input class='marginLeft' name=arrayY[] placeholder=Coordonnées-Y " + (i+1) + ">" + "</div>" + "
          <br>");
        myBtnSearch.style.display = "none";
        myBtnCalcul.style.display = "block";
      }
    }
  </script>
</body>
```

```

    }
}

```

```

function main(){
    const coordonneeX = document.getElementsByName("arrayX[]");
    const coordonneeY = document.getElementsByName("arrayY[]");
    const nbPiquets = document.getElementById("nbPiquets").value;
    const myBtnCalcul = document.getElementById("boutonEnvoiCalcul");
    let x = [];
    let y = [];
    for(i=0; i<=nbPiquets-1; i++){
        x[i] = parseFloat(coordonneeX[i].value);
        y[i] = parseFloat(coordonneeY[i].value);
    }
    myBtnCalcul.style.display = "none";
    pointGravite = calculCentreGravite(x, y, nbPiquets);
    calculAppartenancePointPolygone(x, y, nbPiquets, pointGravite)
}

```

```

function calculAire(x, y, nbPiquets){
    let aire = 0;
    for(i=0; i<=nbPiquets-1; i++){
        aire += x[i] * y[(i+1)%nbPiquets] - x[(i+1)%nbPiquets] * y[i];
    }
    aire = aire * (1/2);

    return aire;
}

```

```

function calculCentreGravite(x, y, nbPiquets){
    aire = calculAire(x, y, nbPiquets);
    let centreDeGraviteX = 0;
    let centreDeGraviteY = 0;
    for(i=0; i<=nbPiquets-1; i++){
        centreDeGraviteX += ( x[i] + x[(i+1)%nbPiquets] ) * ( x[i] * y[(i+1)%nbPiquets] - x[(i+1)%nbPiquets] * y[i] );
        centreDeGraviteY += ( y[i] + y[(i+1)%nbPiquets] ) * ( x[i] * y[(i+1)%nbPiquets] - x[(i+1)%nbPiquets] * y[i] );
    }
    centreDeGraviteX = (1/(6 * aire)) * centreDeGraviteX;
    centreDeGraviteY = (1/(6 * aire)) * centreDeGraviteY;
    pointGravite = affichageAireCentreGravite(centreDeGraviteX, centreDeGraviteY);
    return pointGravite;
}

```

```

function affichageAireCentreGravite(centreDeGraviteX, centreDeGraviteY){
    let pointGravite = new Boolean("false");
    if(isNaN(centreDeGraviteX)&&isNaN(centreDeGraviteY))
        pointGravite = "true";
    else{
        $(".container").append(
            "<div class='marginBottom'><p>Aire : (" + Math.abs(aire) + ") </p>" +
            "<div class='marginBottom'><p>Centre de gravité : (" + centreDeGraviteX + ", " + centreDeGraviteY + ") </p>";

```

```

        pointGravite = "false";
    }
    return pointGravite;
}

```

```

function calculAppartenancePointPolygone(x, y, nbPiquets, pointGravite){
    let determinant = determinant2Vecteurs(x, y, nbPiquets);
    let vacheDansEnclos = new Boolean("false");
    let somme=0;
    if(pointGravite==="true")
        $(".container").append(
            "<div class='marginBottom'><p>Les piquets ne sont pas donnés dans le bon ordre.</p>");
    else{
        theta = calculTheta(x, y, nbPiquets);
        for(i=0;i<=nbPiquets-1;i++){
            if(determinant<=0)
                somme[i] += theta;
            else
                somme[i] -= theta;
        }
        conditionAppartenancePointPolygone(somme);
    }
}

```

```

function conditionAppartenancePointPolygone(somme){
    if(somme==0){
        vacheDansEnclos="false";
        $(".container").append(
            "<div class='marginBottom'><p>Position de la vache : intérieur </p>");
    }
    else{
        vacheDansEnclos="true";
        $(".container").append(
            "<div class='marginBottom'><p>Position de la vache : extérieur </p>");
    }
}

```

```

function calculTheta(x, y, nbPiquets)
{
    let theta;
    produitScalaire = produitScalaireVecteur(x, y, nbPiquets);
    norme = normeVecteur(x, y, nbPiquets);
    for(i=0;i<=nbPiquets-1;i++){
        theta = Math.acos(produitScalaire[i]/(norme[i]*norme[(i+1)%nbPiquets]));
    }
    return theta;
}

```

```

function produitScalaireVecteur(x, y, nbPiquets){
    vecteur = coordonneesVecteur(x, y, nbPiquets);
    vecteurX = vecteur[0];
    vecteurY = vecteur[1];
}

```

```

    let produitScalaireVecteur = [];
    for(i=0;i<=nbPiquets-1;i++)
        produitScalaireVecteur[i] = vecteurX[i] * vecteurX[(i+1)%nbPiquets] + vecteurY[i] * vecteurY[(i+1)%nbPiquets]
;
    return produitScalaireVecteur;
}

function normeVecteur(x, y, nbPiquets){
    vecteur = coordonneesVecteur(x, y, nbPiquets);
    vecteurX = vecteur[0];
    vecteurY = vecteur[1];
    let normeVecteur = [];
    for(i=0;i<=nbPiquets-1;i++)
        normeVecteur[i] = Math.sqrt((Math.pow(vecteurX[i], 2)) + (Math.pow(vecteurY[i], 2)));
    return normeVecteur;
}

function determinant2Vecteurs(x, y, nbPiquets){
    vecteur = coordonneesVecteur(x, y, nbPiquets);
    vecteurX = vecteur[0];
    vecteurY = vecteur[1];
    let determinantVecteur = [];
    for(i=0;i<=nbPiquets-1;i++)
        determinantVecteur[i] = vecteurX[i] * vecteurY[(i+1)%nbPiquets] - vecteurY[i] * vecteurX[(i+1)%nbPiquets];
    return determinantVecteur;
}

function coordonneesVecteur(x, y, nbPiquets){
    let vecteurX = [];
    let vecteurY = [];
    for(i=0;i<=nbPiquets-1;i++){
        vecteurX[i] = (x[(i+1)%nbPiquets]) - (x[i]);
        vecteurY[i] = (y[(i+1)%nbPiquets]) - (y[i]);
    }
    return [vecteurX, vecteurY];
}
</script>
</body>
</html>

```

Fichier design.css :

```

.text-center{
    margin-left:auto;
    margin-right:auto;
}

.pad30{
    padding-top:30px;
}

```

```
.marginLeft{  
  margin-left:5px;  
}
```

```
.marginBottom{  
  margin-bottom:5px;  
}
```