Rapport : une vache dans le pré

Groupe : Martino vincent G2

Ossart Gabriel G2

Url Git : https://github.com/Madshadows/PreAVache

Compilation sous CodeBlock : bouton compiler et run dans main.cpp

Conclusion

Ce projet était intéressant car il nous a permit en plus de l'utilisation de différentes formules mathématiques et du langage technique, de combiner les deux afin d'obtenir un résultat innovant. Nous avons pu rencontrer des problèmes au cours du développement de ce programmes plus particulièrement lors des calculs des angles lors de la dernière partie.

Annexe :

```
#include <iostream>
#include <cmath>

using namespace std;



bool AppartenancePointPolygone(int n,double Polygone[50][2],double Gravite[2])
{
    double Somme, Thetai;
    int i;

    Somme = 0.0;

    for(i=0;i<=n-1;i++)
    {
        //calcul des coordonnées des 2 vecteurs
        double vect1X,vect1Y,vect2X,vect2Y;
```

```
            vect1X = Polygone[i][0]-Gravite[0];

            vect1Y = Polygone[i][1]-Gravite[1];

        if(i!=n-1)

        {

            vect2X = Polygone[i+1][0]-Gravite[0];

            vect2Y = Polygone[i+1][1]-Gravite[1];

        }

        else{

            vect2X = Polygone[(i+1)%n][0]-Gravite[0];

            vect2Y = Polygone[(i+1)%n][1]-Gravite[1];

        }


        // calculs pour trouver l'angle

        double PScal = vect1X*vect2X+vect1Y*vect2Y;

        double NormVect = sqrt(pow(vect1X,2)+ pow(vect1Y,2))*sqrt(pow(vect2X,2)+
pow(vect2Y,2));

        if(vect1X*vect1Y-vect2X*vect2Y >= 0)

        {

             Thetai = acos(PScal/NormVect);

        }

        else{

            Thetai = acos(PScal/NormVect)*-1;

        }


        Somme = Somme + Thetai;


    }

    //renvoie du résultat

    if(Somme == 0)

    {

        return false;

    }

    else{

        return true;

    }

}




int main()

{
```

```
int nombrePiquet;
double Aire=0.0;
double AbscisseG=0.0;
double OrdonneeG=0.0;
bool ResultAppartenance;

double piquet[50][2];  // tableau contenant les valeurs des piquets

double PointGravite[2];
//Remplissage du tableau pour récupérer toutes les valeurs des piquets
do
{
    cout << "Saisir le nombre de piquets" << endl;
    cin >> nombrePiquet;
    if(nombrePiquet < 3 || nombrePiquet > 50)
    {
        cout<< "le nombre de piquet dois être compris entre 3 et 50"<< endl;
    }
}while(nombrePiquet < 3 || nombrePiquet > 50);


for(int i=0; i <= nombrePiquet-1; i++)
{

    cout<< "saisir le piquet "<< i+1 <<endl;
    cin >>piquet[i][1] ;
    cin >>piquet[i][2] ;
}


//Calcul de l'air des piquet avec exception pour la dernière valeur fermant le pré
for(int j=0;j<nombrePiquet;j++)
{
    if( j!= nombrePiquet-1)
    {
        Aire=Aire+(piquet[j][1]*piquet[j+1][2]-piquet[j+1][1]*piquet[j][2]);
    }
    else{
        Aire=Aire+(piquet[j][1]*piquet[0][2]-piquet[0][1]*piquet[j][2]);
```

```
        }
    }
    Aire=Aire/2;
    cout<<"Aire = "<<Aire<<endl;

    //Calcul de l'abcisse du centre de gravité
    for(int m=0;m<nombrePiquet;m++)
    {
        if( m!= nombrePiquet-1)
        {
        AbscisseG=AbscisseG+(piquet[m][1]+piquet[m+1][1])*(piquet[m][1]*piquet[m+1][2]-
piquet[m+1][1]*piquet[m][2]);
        }
        else{
          AbscisseG=AbscisseG+(piquet[m][1]+piquet[0][1])*(piquet[m][1]*piquet[0][2]-
piquet[0][1]*piquet[m][2]);
        }
    }
    AbscisseG=AbscisseG/(6*Aire);

    //Calcul de l'ordonnée du centre de gravité
    for(int m=0;m<nombrePiquet;m++)
    {
        if( m!= nombrePiquet-1)
        {
        OrdonneeG=OrdonneeG+(piquet[m][2]+piquet[m+1][2])*(piquet[m][1]*piquet[m+1][2]-
piquet[m+1][1]*piquet[m][2]);
        }
        else{
          OrdonneeG=OrdonneeG+(piquet[m][2]+piquet[0][2])*(piquet[m][1]*piquet[0][2]-
piquet[0][1]*piquet[m][2]);
        }
    }
    OrdonneeG=OrdonneeG/(6*Aire);

    //Affichage du point de gravité
    PointGravite[0]=AbscisseG;
    PointGravite[1]=OrdonneeG;
    cout<<"Centre de gravité : "<<PointGravite[0]<< ", " <<PointGravite[1]<<endl;

    //Appel de fonction et affichage final
```

```
    ResultAppartenance = AppartenancePointPolygone(nombrePiquet,piquet,PointGravite);
    if(ResultAppartenance == true)
    {
        cout<<"Pas de problème, la vache est dans le pré"<< endl;
    }
    else{
        cout<<"Attention, la vache est hors du pré"<< endl;
    }

    return 0;
}
```