

```

//
// Polygon_PS1.cpp
// Assignment1
//
// Created by Ekrar Efaz on 22/3/23.
//

#include "Polygon.h"
#include "Matrix3x3.h"

float Polygon::getSignedArea() const noexcept
{
    float area = 0.0f;

    for (size_t i = 0; i < fNumberOfVertices; i++)
    {
        // handle last vertex
        if (i == fNumberOfVertices-1)
        {
            const Vector2D& firstVertex = fVertices[0];
            const Vector2D& lastVertex = fVertices[fNumberOfVertices-1];
            area += lastVertex.x() * firstVertex.y() - firstVertex.x() * lastVertex.y();
        }
        else{
            const Vector2D& currentVertex = fVertices[i];
            const Vector2D& adjacentVertex = fVertices[(i + 1)];

            area += currentVertex.x() * adjacentVertex.y() - adjacentVertex.x() * currentVertex.y();
        }
    }
    area = area * 0.5f;
    return area;
}

//float Polygon::getSignedArea() const noexcept
//{
//    float area = 0.0;
//    for (size_t i = 0; i < fNumberOfVertices; i++)
//    {
//        const Vector2D& vertex1 = fVertices[i];
//        const Vector2D& vertex2 = fVertices[(i + 1) % fNumberOfVertices];
//        float crossProduct = vertex1.x() * vertex2.y() - vertex2.x() * vertex1.y();
//        area += crossProduct;
//    }
//    return area / 2.0;
//}

Polygon Polygon::transform( const Matrix3x3& aMatrix ) const noexcept{
    Polygon aTransform(*this);
    for (size_t i = 0; i < fNumberOfVertices; i++){
        aTransform.fVertices[i] = static_cast<Vector2D> (aMatrix * aTransform.fVertices[i]);
    }
    return aTransform;
}

```