

## Übungsblatt 5

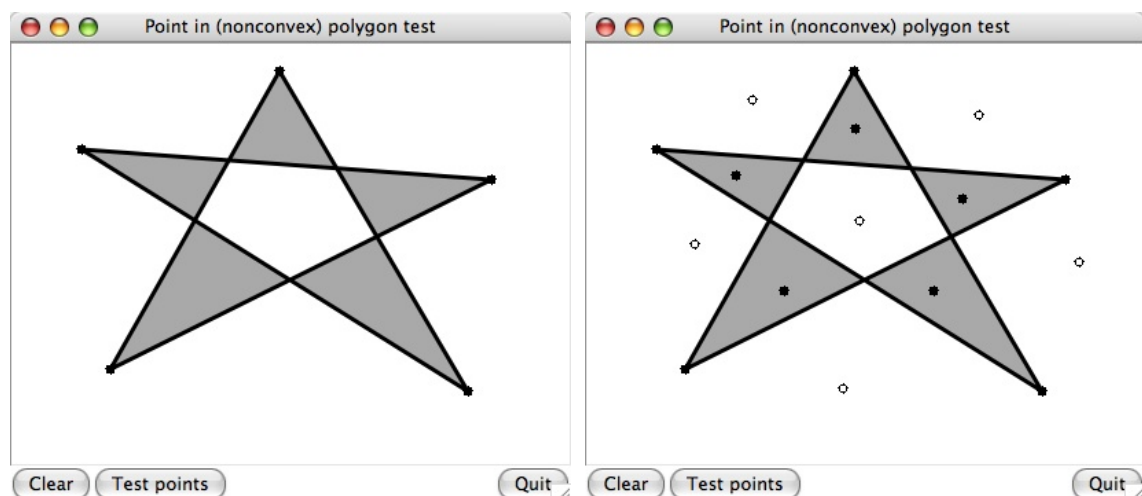
**Aufgabe 1.** Auf der Webseite zur Veranstaltung Generative Computergrafik finden Sie unter <http://www.mi.hs-rm.de/~schwan/Vorlesungen/GenCG> das Python Skript `pointInPolygonTestTemplate.py`.

Nach dem Starten des Programms können Sie mit Hilfe der Maus Punkte eingeben. Diese definieren solange ein Polygon, bis Sie den Button `Test points` drücken (siehe Abbildung). Die anschließend eingegebenen Punkte sollen dann darauf getestet werden, ob sie innerhalb oder ausserhalb des zuvor definierten Polygons liegen. Punkte die *innerhalb* des Polygons liegen sind *gefüllt*, Punkte *ausserhalb* *ungefüllt* darzustellen.

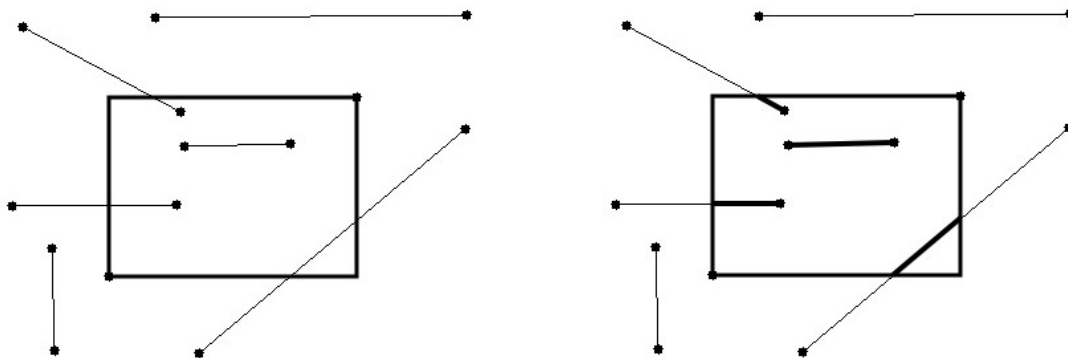
Um zu testen, wo ein Punkt  $p$  liegt, soll das Programm die, in der Vorlesung besprochene Beobachtung nutzen, dass ein beliebiger von  $p$  ausgehender Strahl genau dann eine ungerade Anzahl von Schnittpunkten mit dem Polygon aufweist, wenn  $p$  innerhalb des Polygons liegt.

Implementieren Sie die Funktion `intersect(11, 12)` in dem Python Programm `pointInPolygonTestTemplate.py`, welche überprüft, ob sich die beiden *Liniensegmente* 11 und 12 schneiden. Ein direkter Weg zur Implementierung dieser Funktion besteht darin, den Schnittpunkt der beiden, durch die Liniensegmente definierten Geraden zu berechnen und anschließend zu überprüfen, ob dieser Punkt zwischen den Endpunkten der beiden Strecken liegt. Ein weiterer Ansatz, welcher den *Dreh-sinn* dreier Punkte benutzt, findet sich zum Beispiel in dem Buch *Algorithmen und Datenstrukturen von R. Sedgewick*.

Die nachfolgenden Abbildungen zeigen ein Beispiel. Auf der linken Seite ist ein nicht konvexes Polygon zu sehen. Die rechte Seite entstand durch Eingabe weiterer Punkte, nachdem der Button `Test points` gedrückt wurde. Wie zu erkennen ist, werden Punkte innerhalb des Polygons gefüllt dargestellt, während Punkte ausserhalb des Polygons ungefüllt gezeichnet werden.



**Aufgabe 2.** Auf der Webseite zur Veranstaltung Generative Computergrafik finden Sie unter <http://www.mi.hs-rm.de/~schwan/Vorlesungen/GenCG> das Python Skript `lineClippingTemplate.py`. Das Programm erlaubt es, mit Hilfe der Maus Punkte zu setzen. Die ersten beiden Punkte definieren ein Rechteck. Von den weiteren definieren jeweils zwei eine Gerade. Implementieren Sie in diesem Programm die Funktion `calcNewLine` so, dass ein Lineclipping nach dem Algorithmus von Cohen-Sutherland durchgeführt wird. Die geclippten Linien sollen, wie in den nachfolgenden Abbildungen gezeigt, dicker dargestellt werden.



**Aufgabe 3.** Auf der Webseite zur Veranstaltung Generative Computergrafik finden Sie unter <http://www.mi.hs-rm.de/~schwan/Vorlesungen/GenCG> das Python Skript `bresenhamTemplate.py`. Das Programm erlaubt es mit Hilfe der Maus Punkte in einem Raster zu setzen. Je zwei Punkte werden dabei mit einer dünnen Linie verbunden (siehe Abbildung unten links). Implementieren Sie in dem Programm die Funktion `drawBresenhamLine`. Diese soll mit Hilfe des Algorithmus von Bresenham die Kästchen (Pixel) des Gitters setzen, welche zu den jeweiligen Linien gehören. In der Abbildung unten rechts ist ein gewünschtes Ergebnis zu sehen.

