HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Konstanz, 26.09.2016

Übungsblatt 1

„Computergrafik“

Besprechung und Abgabe spätestens am 08.11.2016, F033.

Programmgerüst für die Übung:
Laden Sie sich das Programmgerüst für die Übung von der Webpage der Vorlesung herunter. Legen Sie im Visual Studio ein neues Projekt an und führen Sie das Programm aus. Es erscheint ein Fenster wie in Abbildung 1.

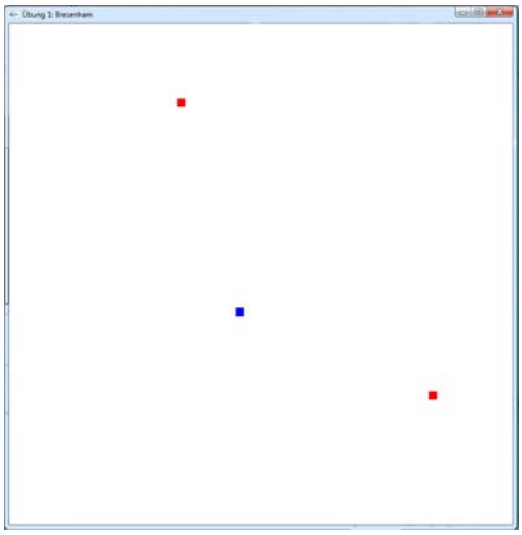

Abbildung 1: Programmgerüst für Übung 1

Wenn Sie die Datei öffnen werden Sie feststellen, dass sich darin einiges an Programmcode befindet. Nicht alle Sourcen sind für die Lösung der Aufgabe wichtig. Moderne Monitore besitzen eine große Auflösung. Daher können Sie die Korrektheit Ihres Programms nur sehr eingeschränkt testen, wenn Sie die Algorithmen auf Pixelbasis programmieren, weil man ein einzelnes Pixel kaum erkennt. Durch das Programmgerüst wird eine Vergrößerung vorgenommen.
Wichtig für die Übung sind lediglich die Funktionen `clearImage(Color c)`, `setPoint(Point p,Color c)` und `display()` sowie `bhamLine(Point p1, Point p2, Color c)` und `bhamCircle (Point p, int r, Color c)`. Benutzen Sie die Funktion `setPoint`, um ein "Pixel" zu setzen. Der zeichenbare Bereich ist auf die Koordinaten $[-50; 49]^2$ beschränkt. Wenn Sie außerhalb dieser Koordinaten einen Punkt zeichnen, wird im Konsolefenster eine entsprechende Fehlermeldung ausgegeben.

HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
Fakultät für Informatik
Labor für Computergrafik
Prof. Dr. G. Umlauf

Aufgabe 1 (Der Bresenham Algorithmus für Linien) **5 Punkte**

Teil 1:
Implementieren Sie Bresenhams Algorithmus für Linien im ersten Oktanten, indem Sie die Funktion

```
bhamLine(Point p1, Point p2, Color c)
```

entsprechend erweitern. Der Aufruf der Funktion erfolgt dann in der Funktion `display()`.

Teil 2:
Verallgemeinern Sie die Funktion Bresenham auf beliebige Oktanten durch Spiegelung von Start- und Endpunkt an der X- bzw. Y-Achse.

Aufgabe 2 (Der Bresenham Algorithmus für Kreise) **5 Punkte**

Teil 1:
Implementieren Sie den Bresenham Algorithmus für Kreise zunächst für den zweiten Oktanten, indem sie die Funktion

```
bhamCircle(Point p, int r, Color c)
```

entsprechend erweitern. Der Aufruf der Funktion erfolgt dann in der Funktion `display()`.

Teil 2:
Erweitern Sie die Funktion dann durch Spiegelung der Ausgabepunke auf einen Vollkreis und fügen Sie zum Schluss die Möglichkeit hinzu den Kreis beliebig zu positionieren.

Besprechung und Abgabe spätestens am 08.11.2016, F033.