



Grundlagen der Computergrafik — Praktikum

Bresenham-Line-Algorithmus

Ziel

Am Ende dieses Praktikumsteils haben Sie das Zeichnen von Linien mit dem Bresenham-Line-Algorithmus für alle Oktanten selbst implementiert.

Vorbereitung für den Praktikumstermin

Lesen Sie bitte als Vorbereitung für den Praktikumstermin sowohl diese PDF als auch die PDF mit den *allgemeinen Hinweisen* aufmerksam durch. In den allgemeinen Hinweisen finden Sie Softwareempfehlungen für das Praktikum und Informationen über die Struktur der Projektmappe. Machen Sie sich zudem bitte schon mit dem bereitgestellten Praktikumsframework vertraut. Öffnen Sie dafür die JavaScript-Datei `bresenham.js` in einem Editor Ihrer Wahl und zeigen Sie die Html-Datei `bresenham.html` in einem Webbrowser an. Wenn Sie sich die JavaScript-Datei `bresenham.js` anschauen, werden Sie am Ende des Codes folgende Beispielfunktion finden:

```
function example(i)
{
  let y = i + 2;
  for(let x = 0; x < 400; x++)
  {
    y--;
    if(y < -i)
    {
      y = i;
    }
    setPixel(x, Math.abs(y));
  }
}
```

Stellen Sie sicher, dass Sie zum Praktikumstermin genau verstanden haben, wie dieses Beispiel funktioniert und was es für ein Ergebnis liefert. Das Beispiel enthält bereits alle Befehle, die Sie für die Lösung der Praktikumsaufgabe benötigen.

Bearbeitung, Abgabe, Bewertung

Die Abgabe erfolgt gruppenweise. Ein Gruppenmitglied lädt dazu ein Zip-Archiv der Dateien `bresenham.js`, `bresenham_1.js` und `bresenham_1458.js` auf der [moodle-Seite](#) des Kurses hoch. Wichtig: Tragen Sie unbedingt Ihre Gruppe, den Studiengang sowie die einzelnen Gruppenmitglieder im führenden Kommentarblock der Datei ein. Benennen Sie die Datei folgendermaßen:

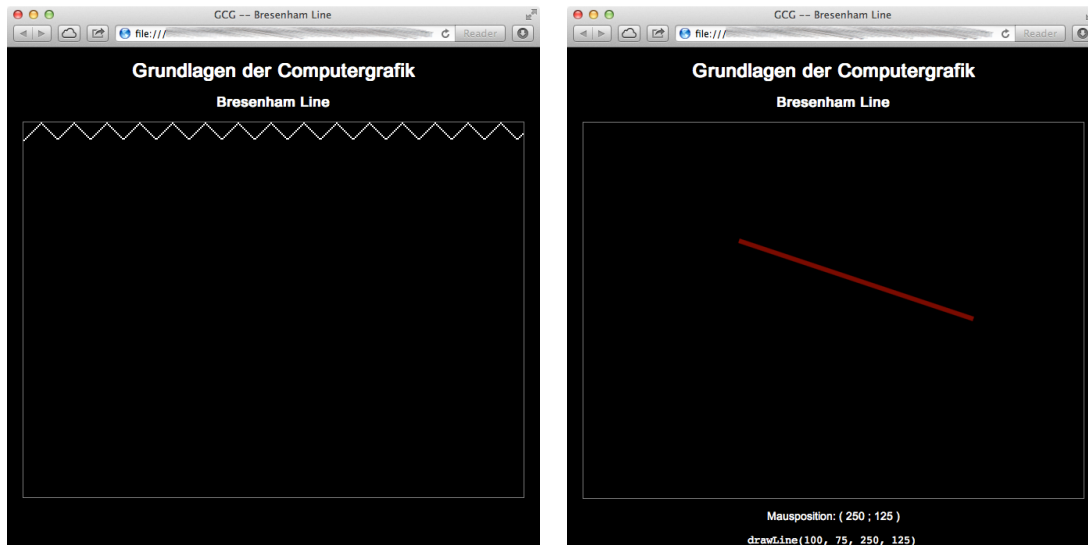
STUDIENGANG-GRUPPE-NAME1-NAME2-bresenham.zip (Beispiel: BMT-A-Mayer-Richter-bresenham.zip)

Für die Bearbeitung haben Sie einen Praktikumstermin sowie anschließend 7 Tage Zeit. Für diesen Praktikumsteil erhalten Sie maximal 10 Punkte. Achten Sie auf die Vollständigkeit Ihrer Abgabe. Verspätete Abgaben werden nicht berücksichtigt. Bei Feststellung eines Täuschungsversuchs wird das Praktikum für alle beteiligten Gruppen mit 0 Punkten bewertet.

Praktikumsaufgaben

Öffnen Sie die Datei `bresenham.html` in Ihrem Browser. Sie sollten auf der Html-Seite einen Rahmen sehen, in den ein Zickzack-Muster gezeichnet wurde (Abbildung links). Wenn Sie die Maus innerhalb des Rahmens bewegen, wird Ihnen die aktuelle Mausposition angezeigt (Abbildung rechts). Ein Mausklick startet das Zeichnen, ein weiterer beendet das Zeichnen der Linie. Die breite, rote Linie dient zur visuellen Kontrolle Ihrer Implementation. Die zugrunde liegende Black-Box ruft zum Zeichnen automatisch die von Ihnen implementierte Funktion `drawLine(...)` in der Datei `bresenham.js` auf. Diese soll eine dünne, weiße Linie zeichnen. Die korrekten Parameter für Anfangs- und Endpunkt werden automatisch übergeben. Sind die rote und die weiße Linie für alle Oktanten deckungsgleich, ist Ihre Implementation korrekt. Zur Erläuterung wird Ihnen der Aufruf von

`drawLine(...)` inklusive der Parameter auf der Html-Seite angezeigt.



1. Erster Oktant

Implementieren Sie in der Funktion `drawLine(...)` in der Datei `bresenham.js` das Zeichnen von Linien in den ersten Oktanten mit dem Bresenham-Line-Algorithmus, den Sie in der Vorlesung kennen gelernt haben.

Sämtliche Befehle, die Sie dazu benötigen, finden Sie in der Funktion `example(...)` in der Datei `bresenham.js`, die beim Laden der Seite ein Zickzack-Muster erstellt. Kopieren Sie den aktuellen Stand in eine Datei namens `bresenham_1.js`.

Hinweis:

Die Aufteilung der Oktanten verläuft in der WebGL-Darstellung am Bildschirm anders als im Koordinatensystem der Vorlesung. Beachten Sie, dass die Y-Achse im Fenster des Programms nach unten zeigt!

2. Oktanten 4, 5 und 8

Implementieren Sie zusätzlich das Zeichnen in die Oktanten 4, 5 und 8. Machen Sie durch Kommentare im Code deutlich, welche grundlegenden Änderungen nötig sind, um in den jeweiligen Oktanten zeichnen zu können. Kopieren Sie den aktuellen Stand in eine Datei namens `bresenham_1458.js`.

3. Oktanten 2, 3, 6 und 7

Implementieren Sie zusätzlich das Zeichnen in die Oktanten 2, 3, 6 und 7. Machen Sie durch Kommentare im Code deutlich, welche grundlegenden Änderungen nötig sind, um in den jeweiligen Oktanten zeichnen zu können. Alle Oktanten sollten nun in der Datei `bresenham.js` gezeichnet werden.

Punkteverteilung (10 Punkte)

- | | |
|--|----------|
| • Linien werden korrekt gezeichnet: | 6 Punkte |
| • Aussagekräftige Kommentierung & Benennung der Variablen: | 1 Punkt |
| • Symmetrie/Performance: | 3 Punkte |

Erklärung zur Symmetrie:

Nutzen Sie unbedingt die Symmetrie-Eigenschaften des Algorithmus aus. Eine Implementierung in nachfolgender Form zeichnet zwar in alle Oktanten, führt aber zu deutlichem Punktabzug, da Symmetrie-Eigenschaften nicht berücksichtigt werden:

```
function drawLine(x0, y0, x1, y1) {
  if (erster oktant) { ... setPixel( ... ); ... }
  else if (zweiter oktant) { ... setPixel( ... ); ... }
  ...
  else if (siebter oktant) { ... setPixel( ... ); ... }
  else if (achter oktant) { ... setPixel( ... ); ... }
}
```