

Dokumentation Pi Simulator

von Sabrina Leuenberger WIBB2.61

Wie der Name schon sagt, schätzt dieses Programm den Wert von Pi über eine Simulation ab. Der Grundgedanke hinter der Schätzlogik ist, dass wenn in einem Quadrat mit der Seitenlänge 1 ein Viertelkreis mit Radius = 1 hineingesetzt wird und dann zufällig Punkte in diesem Quadrat generiert werden, kann Pi aus dem Verhältnis zwischen den Anzahl Punkten innerhalb des Viertelkreises und der totalen Anzahl Punkte gemäss folgender Formel¹ geschätzt werden: $\text{Pi} = (\text{Anzahl Punkte innerhalb des Viertelkreises} / \text{Anzahl aller Punkte}) * 4$. Für die Berechnung, ob sich ein Punkt innerhalb oder ausserhalb des Viertelkreises befindet wurde folgendermassen vorgegangen¹: Wenn $x^2 + y^2 < 1$, wobei x und y die Koordinaten des Punktes sind, befindet sich der Punkt innerhalb des Viertelkreises, ansonsten ausserhalb.

Das Programm basiert auf dem Model-View-Controller Pattern. Das Modell wurde mit einer Klasse Point ergänzt, welche auch einen Teil der Logik enthält.

Die Klasse View definiert das GUI, welches in vier Teile gegliedert ist:

Im oberen Teil werden die Anzahl der generierten Punkte angezeigt und die aktuelle Schätzung von Pi, die auf diesen Punkten basiert.

Darunterliegend wird das Quadrat angezeigt, innerhalb dessen die simulierten Punkte angezeigt werden. Die Punkte, welche sich innerhalb des Viertelkreises befinden erscheinen blau, die anderen gelb. Das führt dazu, dass der Vierteilkreis umso deutlicher erscheint, je mehr Punkte generiert werden.

Darunter wird angezeigt, wie weit die Schätzung vom wirklichen Pi entfernt ist. Die rote Linie steht für Pi, die grüne Linie gibt an, wie weit das geschätzte Pi vom reellen Pi entfernt ist, sowohl im positiven Bereich, wie auch im negativen Bereich. Für die Skala musste dabei ein Kompromiss zwischen der kompletten Sichtbarkeit der generierten Kurve und der Präzision der Kurve nahe bei Pi eingegangen werden. Aktuell ist es so eingestellt, dass am Anfang der Simulation, die Kurve ausserhalb des sichtbaren Bereiches zu liegen kommen kann, dafür kann man die Differenz zwischen realem Pi und simulierten Pi bis auf die zweite Nachkommastelle sehen. Wenn die Skala so gewählt worden wäre, dass die Differenz bis auf die dritte Nachkommastelle deutlich sichtbar gewesen wäre, wäre die Linie im Schnitt noch öfter ausserhalb des sichtbaren Bereichs zu liegen kommen, was nicht gewünscht war.

Und schliesslich befinden sich im untersten Teil des Fensters die Interaktionsmöglichkeiten des Benutzers mit der Simulation. Der Benutzer kann die Simulation starten und pausieren. Im pausierten Zustand, lässt sich die Simulation dann auch wieder zurücksetzen. Ausserdem hat der Benutzer die Möglichkeit die Geschwindigkeit der Simulation zu variieren.

Das Programm wird durch das Schliessen des Fensters beendet.

¹ Gemäss Anleitung zu den Bonuspunktaufgaben, Vorlesung SWE2 von Brad Richards, FHNW