AR Billiard mit OpenCV und OpenGL

Sommersemester 2018

Friedemann Runte, Moritz Ludolph, Diyar Omar, Robin Mertens

Fakultät für Informatik

14. Juli 2018

Darstellung des Spielfeldes

- Hintergrundfarbe Dunkel-Grün mit der OpenGL ClearColor
- Löcher und Kugeln mit Triangle-Fan

Triangle-Fan

- Struktur aus Dreiecken
- Mittelpunkt c
- Radius r
- Auflösung k ist die Anzahl der Dreiecke, aus denen unser Kreis hinterher besteht

Triangle-Fan

Berechnung:

1. Berechnen vom Winkel $\delta = \frac{2 \cdot \pi}{k}$, der den Abstand zwischen den Punkten auf dem Kreis darstellt
2. Mittelpunkt $= (0, 0)$, Objekt wird später an die Stelle c transformiert
3. Erste Punkt auf dem Kreis ist $(r, 0)$
4. Dreieck wird vollendet mit dem Punkt $x = \cos(\delta \cdot i) \cdot r$
   $y = \sin(\delta \cdot i) \cdot r$, mit $i =$ Anzahl der Iterationen von $i = 0$ bis k
5. Jeder neu hinzugefügte Punkt wird mit dem Mittelpunkt und dem letzten Punkt auf dem Umriss zu einem Dreieck verbunden

Abbildung: Erstes Dreieck eines Triangle-Fans, $i = 1$, $k = 36$

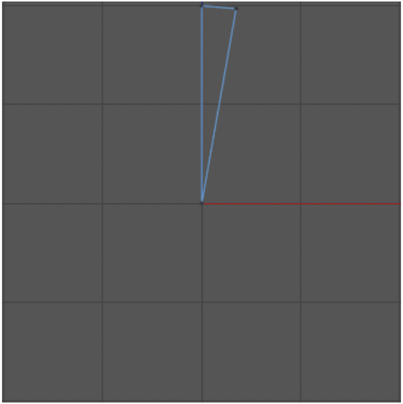


Abbildung: $i = 2$, $k = 36$

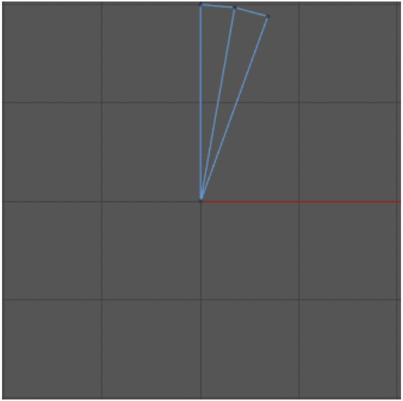


Abbildung: $i = 3,\ k = 36$

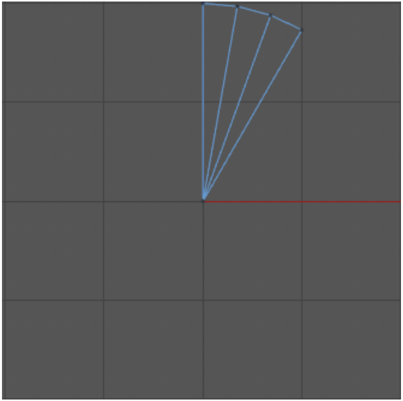
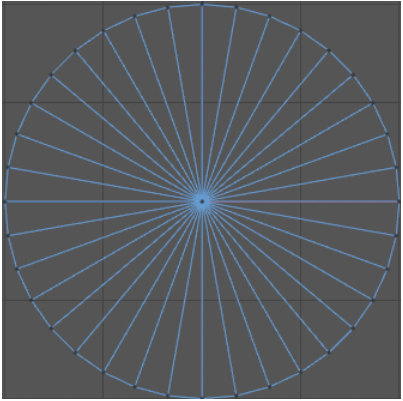


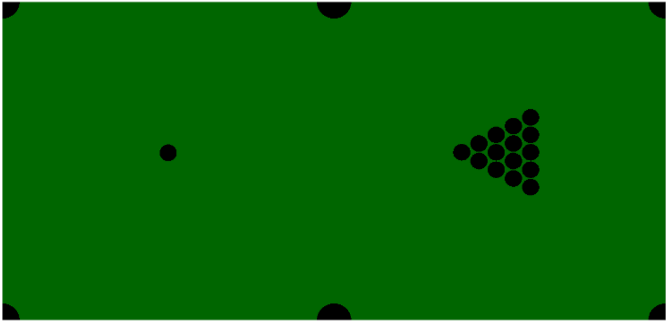
Abbildung: Vollständiger Triangle-Fan, $i = k = 36$



Zwischenstand

Wir haben nun ein Spielfeld mit Löchern und Kugeln. Jedoch sind die Kugeln noch nicht voneinander zu unterscheiden.

Abbildung: Spielfeld ohne Texturen

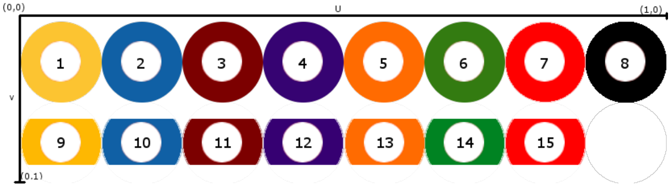


Grundkonzepte:

- Eine Textur besteht aus Koordinaten auf der u (waagerecht) und der v (vertikal) Achse
- Die beiden Achsen sind immer im Bereich $[0, 1]$, egal wie groß die Textur ist
- Man gibt beim Erstellen von Objekten für jeden Knoten die Texturkoordinaten in u und v an, um sie auf das Objekt abzubilden

Texturierung

Abbildung: Textur mit u- und v-Achse



Es fällt auf, dass eine Kugel $\frac{1}{8}$ Durchmesser hat auf u, aber $\frac{1}{2}$ auf v

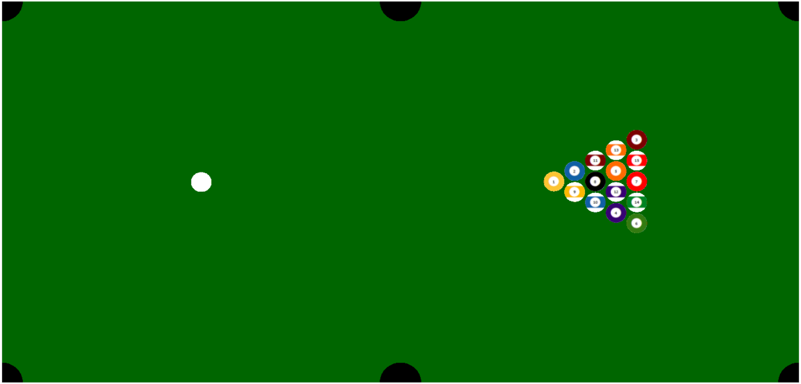
Wir müssen nun berechnen, welcher Ausschnitt für welche Kugel ist.

- Die Farben sind geordnet auf der Textur und bekommen Werte von 0 bis 7
- Die Vollen Kugeln sind in der ersten Reihe und die Halben in der Zweiten und bekommen damit die Werte 0 (voll) und 1 (halb)
- Wir können jetzt eine Funktion erstellen, die uns anhand von Farbe und Fülle den Textur Mittelpunkt ausgibt

Beispiel: Mittelpunkt der gelben, halben Kugel auf der Textur:
$u = \frac{1}{16} + \frac{1}{8} \cdot 0 = \frac{1}{16}$
$v = \frac{1}{4} + \frac{1}{2} \cdot 1 = \frac{3}{4}$
Dabei ist $\frac{1}{16}$ der Radius einer Kugel auf u, und $\frac{1}{4}$ der Radius auf v. Die Funktion ist dann:
$u(b) = \frac{1}{16} + \frac{1}{8} \cdot c(b), c(b) =$ Farbe der Kugel
$v(b) = \frac{1}{4} + \frac{1}{2} \cdot f(b), f(b) =$ Fülle der Kugel

Wir haben nun ein fertiges Spielfeld mit Texturierten Kugeln.

Abbildung: Fertiges Spielfeld



Die Physik-Berechnungen des Spiels lassen sich aufteilen in 3 Bereiche:

- Kollision von Kugel und Wand
- Kollision von Kugeln mit anderen Kugeln
- Kollision vom Queue mit der weißen Kugel

Kollisionen: Kugel mit Wand

Prinzip sehr einfach:

1. Definiere die Wände als Achsenabschnitte: Linke Wand ist $x = 0$, rechte Wand $x = w$, obere Wand $y = 0$ und untere Wand $y = h$, mit $w =$ Breite des Spielfeldes und $h =$ Höhe des Spielfeldes
2. Überprüfen ob eine der Koordinaten der Kugel zusammen mit dem Radius eine der Wände schneidet
   z.B. für $r = 5$ wäre $x = 5 - r = 0$ und würde somit die Linke Wand schneiden
3. Geschwindigkeit auf der Achse die Geschnitten wurde invertieren, also bei $x = 0$ oder $x = w$ wird $vx = -vx$ gesetzt, analog für y

Nun fehlen noch die Löcher:

- Wir setzen zusätzliche Bedingungen für die Kollision mit Wänden ein
- Wir überprüfen beim Schnitt mit einer der Wände, ob der Ausschnitt auf bei einem Loch liegt

$$vx = \begin{cases} -vx, & \text{falls } y \leq (h-l) \wedge y \geq l, l = \text{Radius eines Lochs} \\ vx, & sonst \end{cases}$$ Analog für vy, nur mit 2 Wandabschnitten auf der X-Achse aufgrund der 3 Löcher auf X.

Wenn der Mittelpunkt einer Kugel von der Distanz her weniger weit entfernt ist von dem Mittelpunkt eines Lochs, als der Radius von diesem, dann lassen wir die Kugel verschwinden und zählen sie als Punkt.

Kollision mit anderen Kugeln

- Die Kollision von 2 Kreisen war uns bereits gegeben durch das Airhockey-Spiel.
- In Airhockey kollidiert ein Puck mit einem der beiden Schläger und bekommt dadurch eine neue Geschwindigkeit
- Der Schläger wird von der Kollision nicht verändert
- Wir brauchen aber, dass sich beide Kugeln bei Kollision verändern

Kollision mit anderen Kugeln

Lösung:

- Wir berechnen für jede Kugel die Kollision mit jeder anderen Kugel
- Wir nehmen die Methode aus dem Airhockey und wählen unsere Kugel die sich bewegen soll als Puck und berechnen die Kollision mit allen anderen Kugeln als Schläger
- Wenn wir das in beide Richtungen ausführen, sodass jede Kugel sozusagen einmal Schläger und einmal Puck ist, wird jede Kugel von einer Kollision getroffen

Kollision mit Queue

Die eigentliche Kollision bleibt gleich wie beim Airhockey.
Problem: Framerate der Kamera
$\Longrightarrow$ Unschärfe lässt Kollisionen verloren gehen

Literatur