

Feuersimulation

Praktikum „Parallele Algorithmen“
Kristin Schmidt
Sebastian Höffner

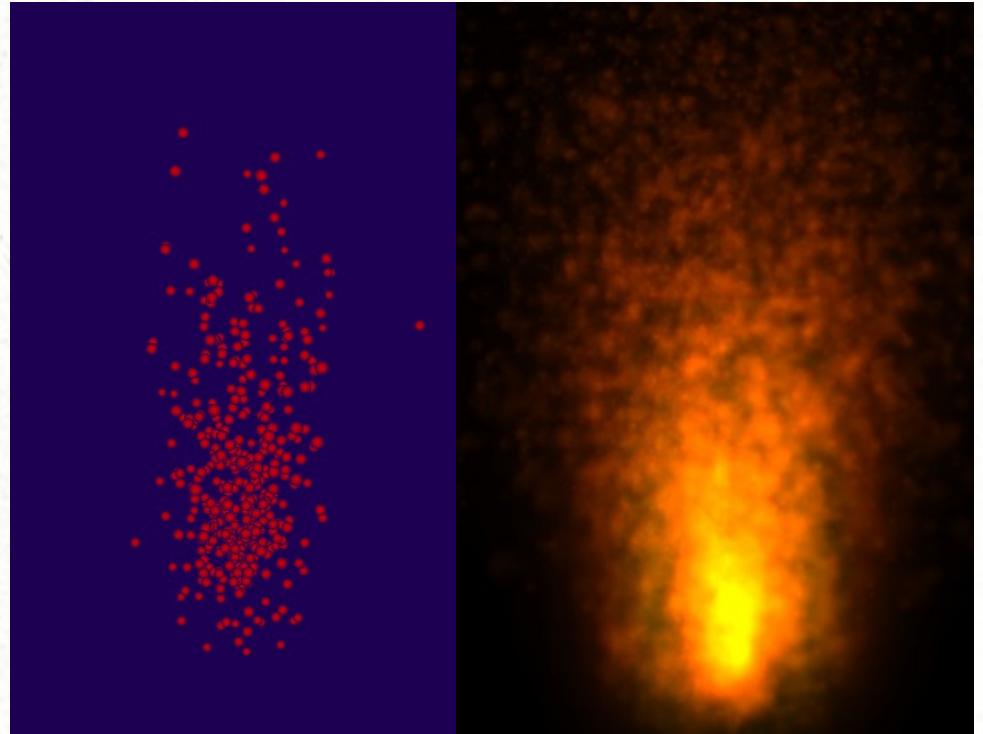
Motivation

- *Aufbau interaktiver 3D-Engines:*
Fackelschimmer



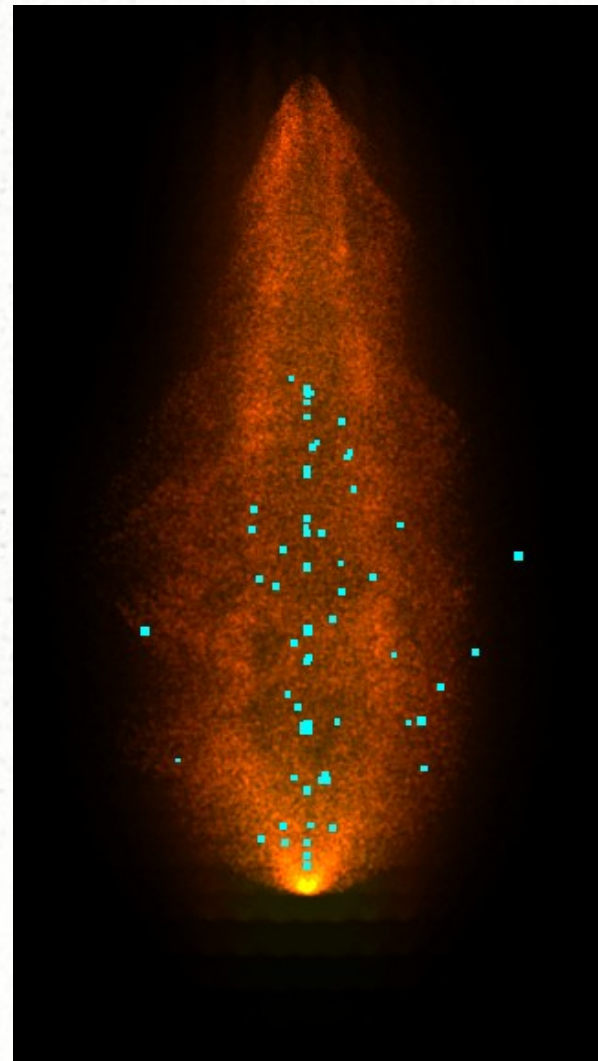
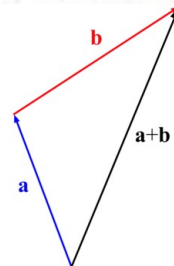
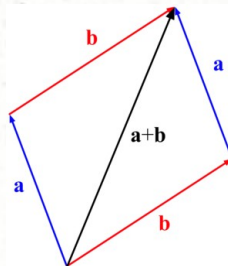
Umsetzungsschwerpunkte

- Partikelsystem
- Zusammenspiel
OpenCL + OpenGL
- Visualisierung von
Partikeln



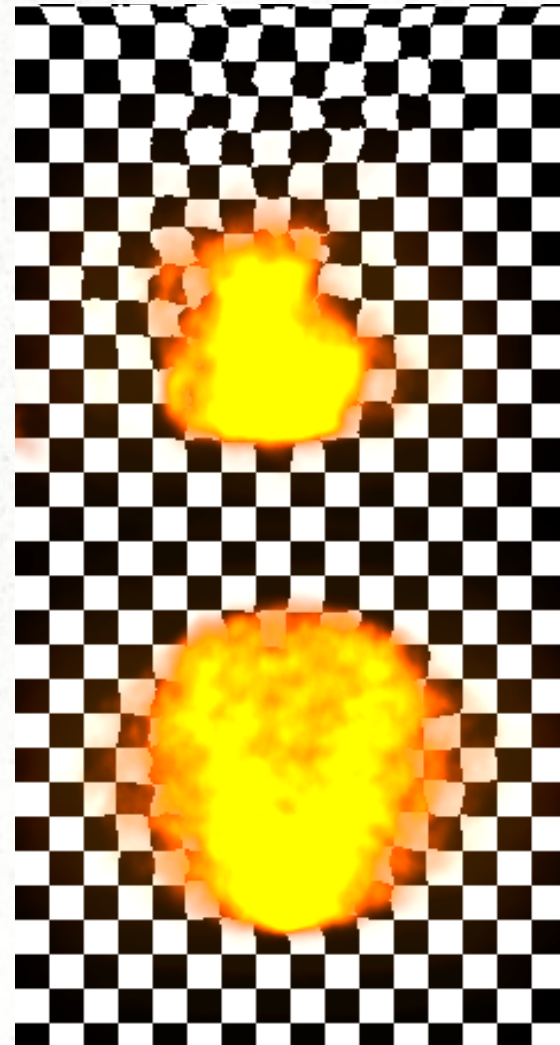
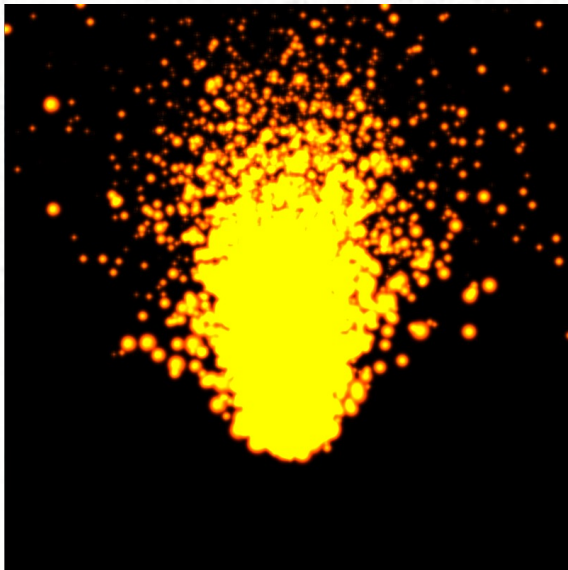
Partikelsystem

- Partikel haben Eigenschaften
 - Richtung,
Position, Alter
- Bewegung durch "Krafteinwirkungen"
 - Low Pressure Areas



Respawn

- Ziel: Alte Partikel recyceln
- Problem: Was sind "alte" Partikel?

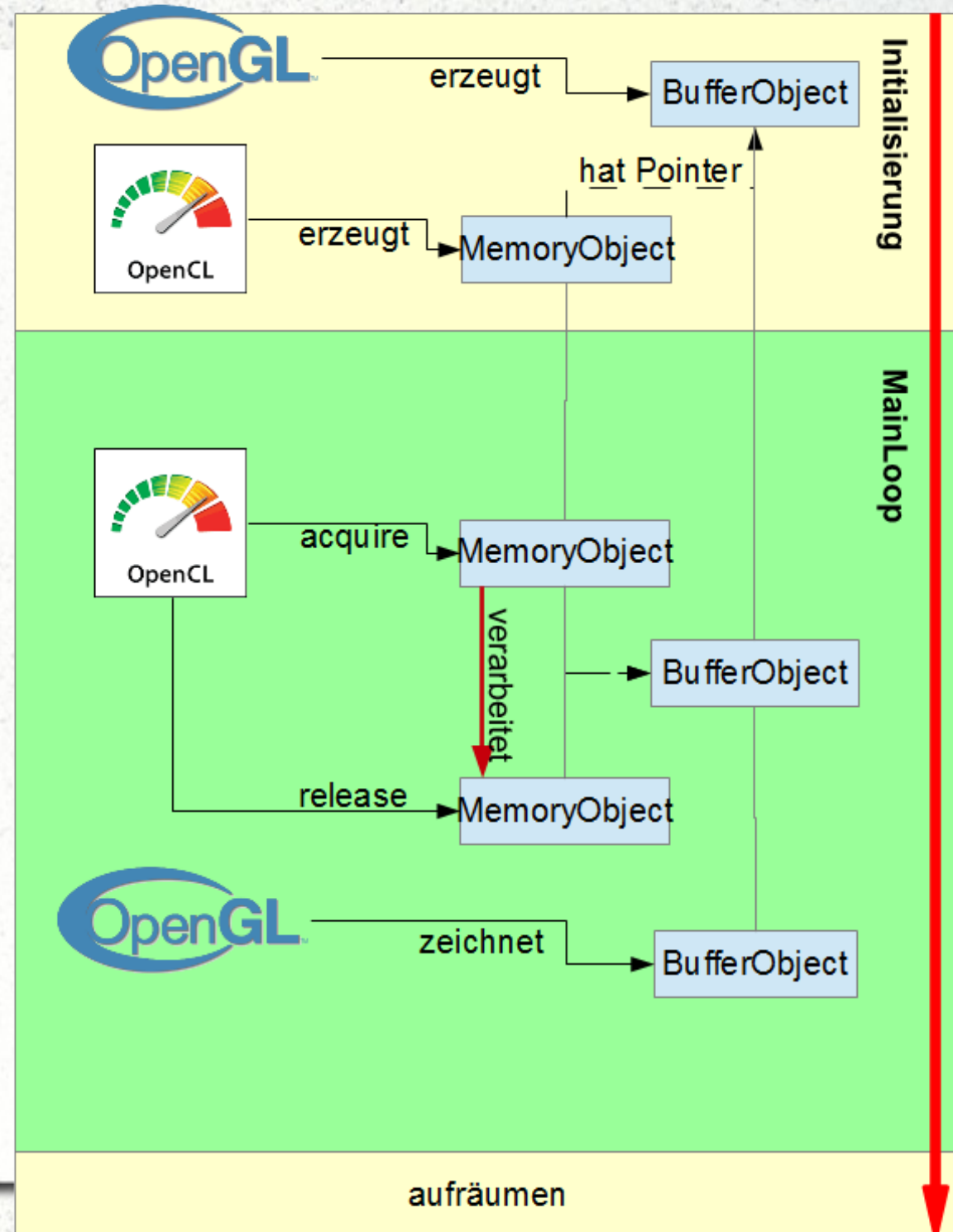


Respawn

- Lösung 1: Sortieren
 - Beliebige Reihenfolge möglich
 - Langsam (Bitonic Sort: $O(\log^2 n)$)
- Lösung 2: ArrayShift
 - Nur $O(2n)$ Speicherzugriffe
 - Schwierigkeit
Synchronisation:
Lesen/Schreiben
- Lösung 3: Erneuere m Partikel ab Offset
 - nur $O(m)$ schreibende Speicherzugriffe
 - Unter Umständen nicht älteste Partikel
- **Gewinner: Lösung 3**
 - Sehr schnell
 - Nur Schreiben

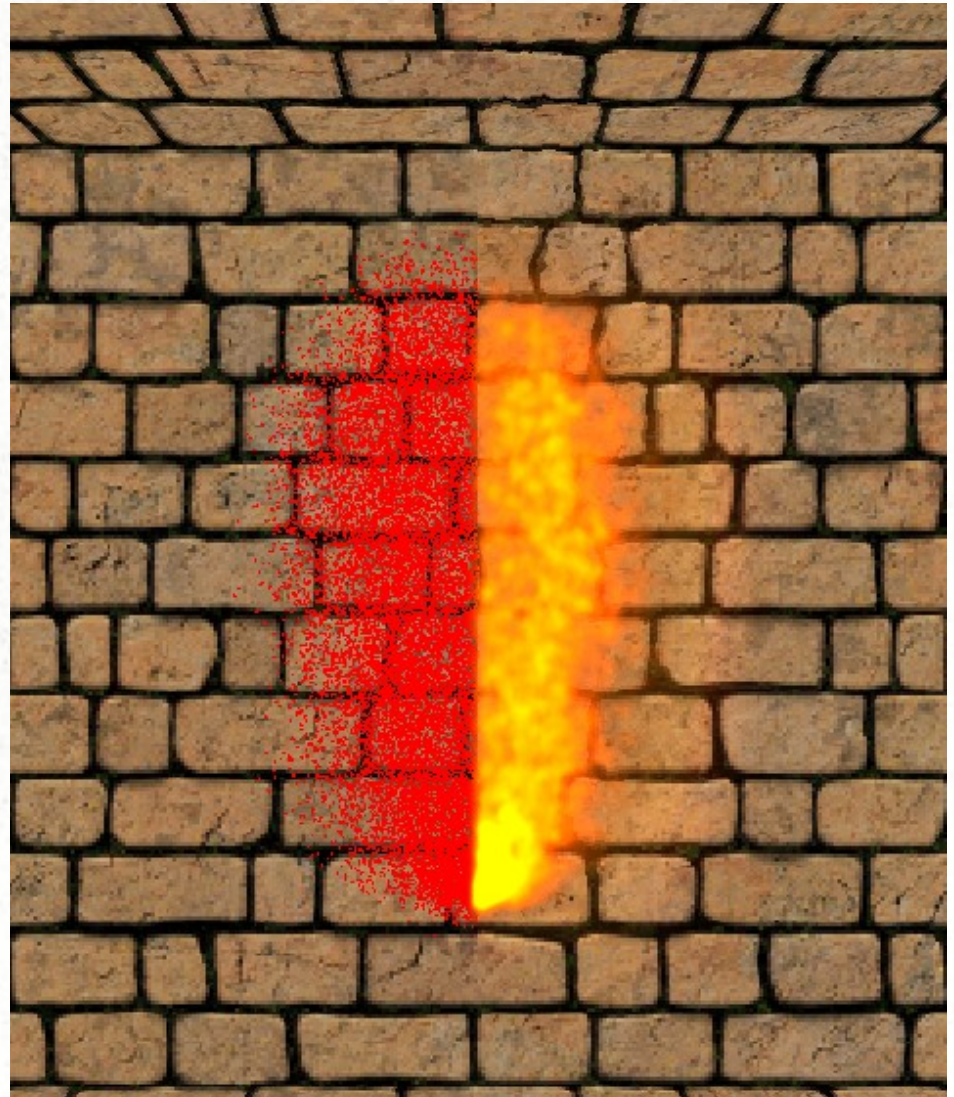
OpenCL + OpenGL

- Effizienz durch geteilte Ressourcen
 - OpenCL arbeitet auf dem von OpenGL reservierten Speicher
- Einfacher Zugriff sowohl im Kernel als auch im Shader



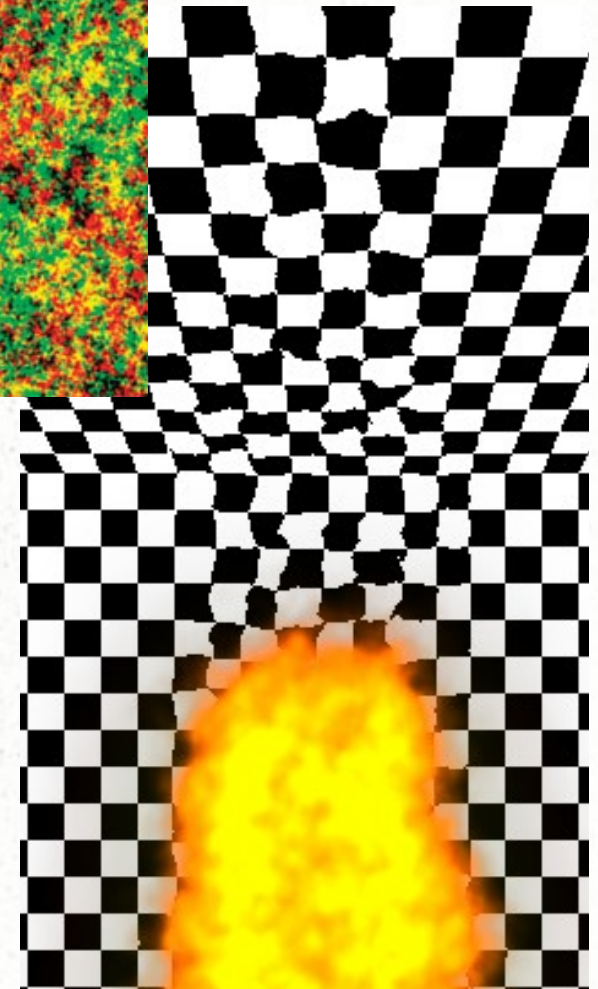
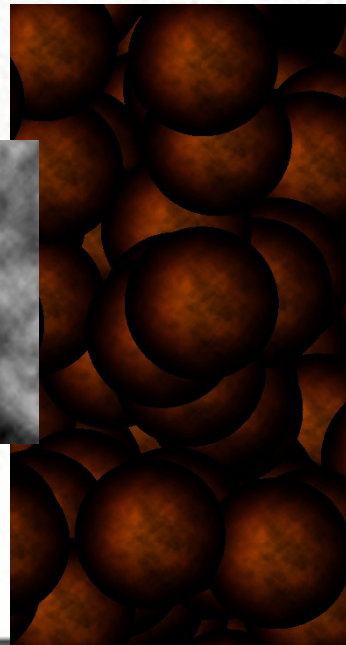
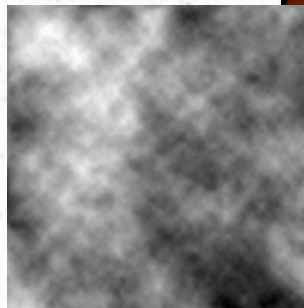
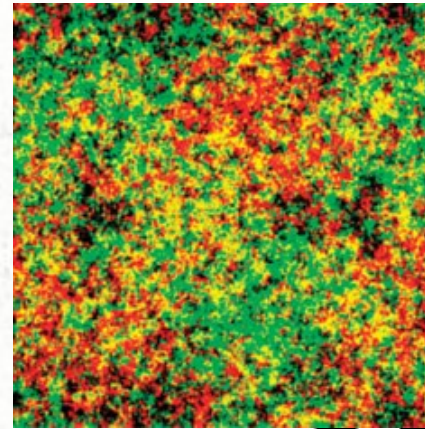
Visualisierung

- Partikel sollen als Feuer erkennbar werden
- Blurfilter, Beleuchtung, Blending, ...
- Special Effects
 - Hitzeblimmern



Noise

- Einfache Methode um Zufall zu erzeugen
- Anwendung beim Hitzeflimmern und als Struktur



DEMO

