Tennis mit Leap Motion

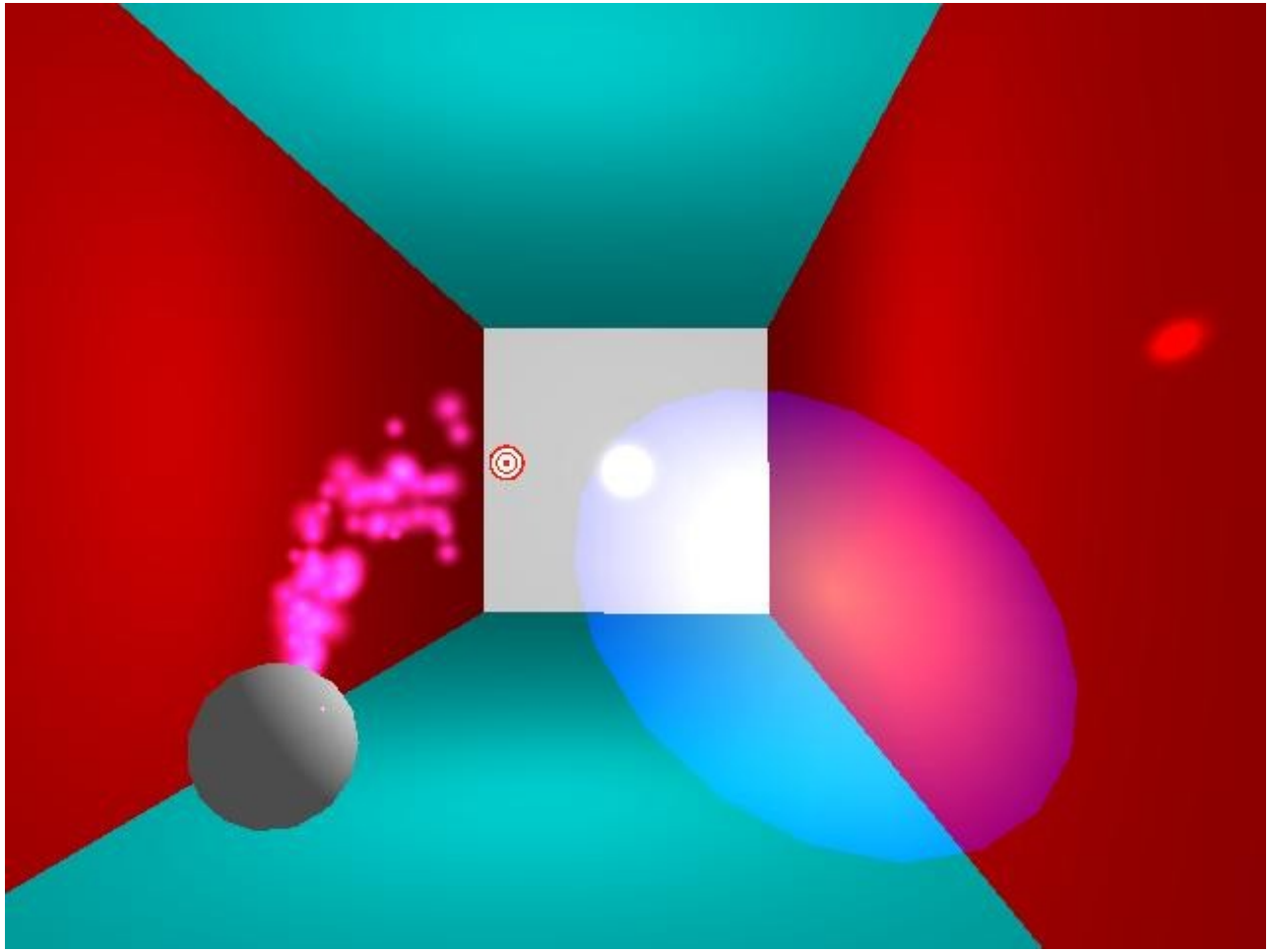
V. Reckendrees, A. Ljulin, M. Greco, E. Arpaci

Fachprojekt Visual Computing

24. Juli 2017

Inhaltsverzeichnis

Worum geht's?



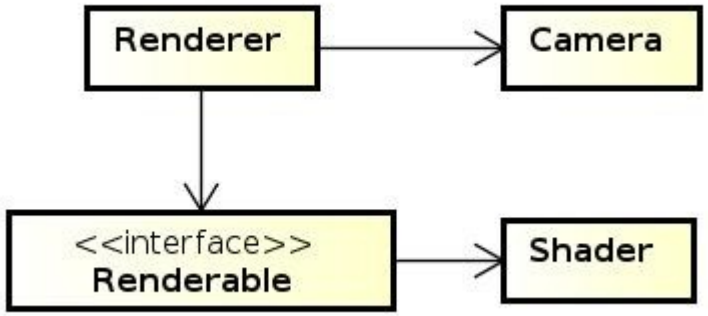
Inhaltsverzeichnis

Rendering Architektur

- Klassen der Spielobjekte:

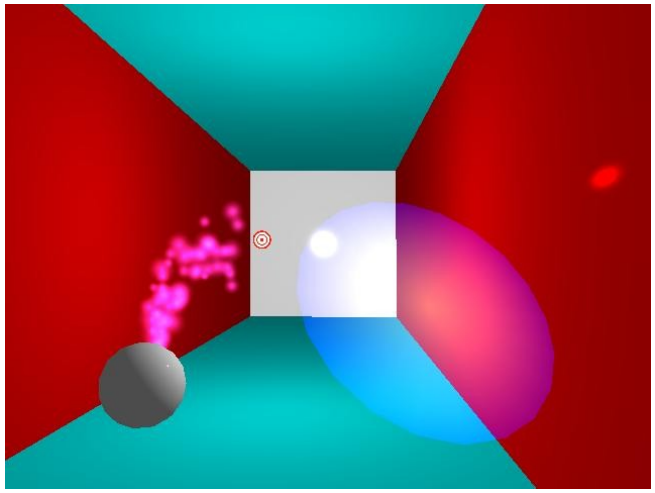
  => Ball, Racket, Box, Scorefield

- Enhalten Member, die Renderable Interface implementieren
- Jedes Renderable zuständig für
  - Erzeugung der Render-Daten, der Buffer und des Shader-Programms
  - Rendern des Modells



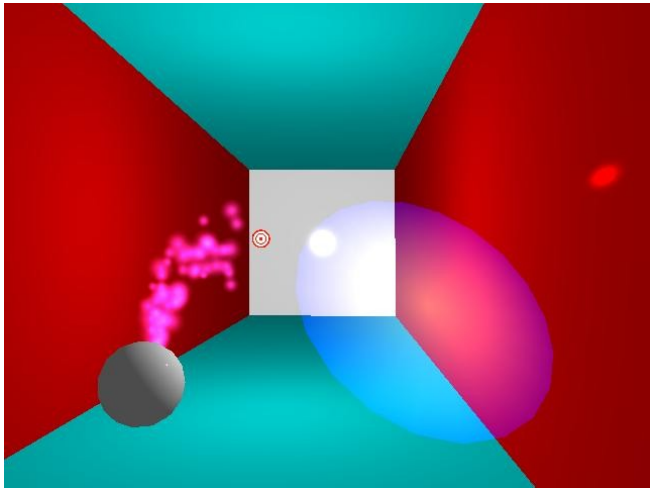
3D-Modelle

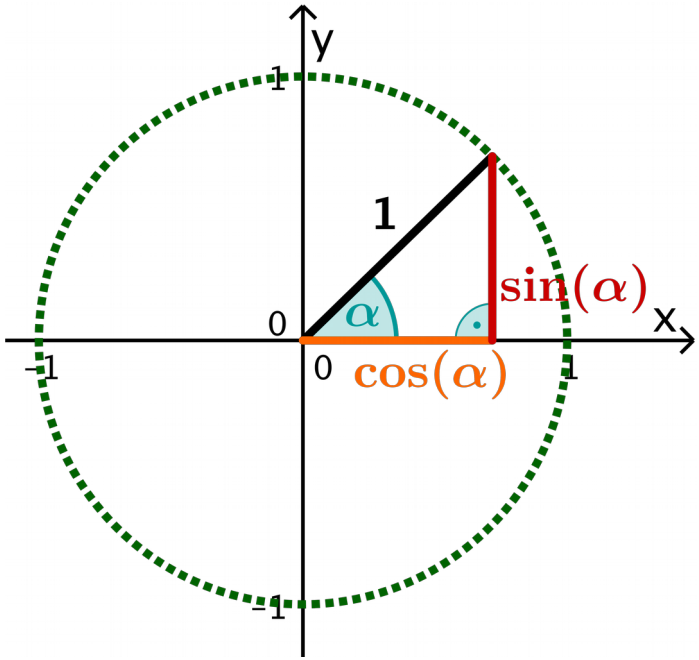
- Box (Spielfeld)
  - Quaderförmig
  - Wände als Ebenen in Hessescher NF $\vec{n} \cdot \vec{x} = d$
  - Koordinaten der Vertices aus den Abstandswerten



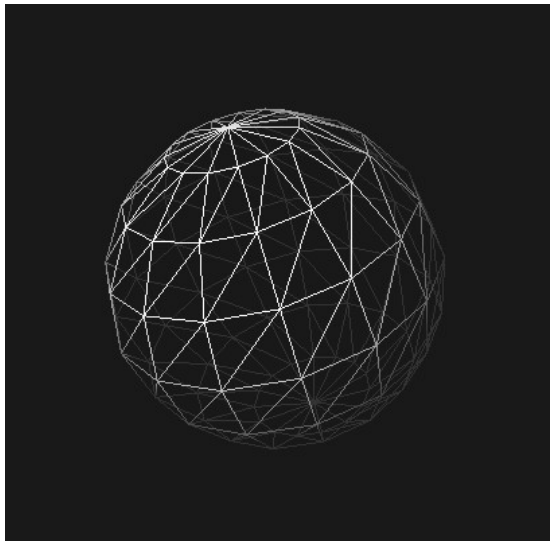
3D-Modelle

- Racket (Schläger)
  - Regelmäßiges Polygon / n-Eck
  - N+1 Vertices
  - Erzeugung der Vertices durch Berechnung der Koordinaten auf dem Einheitskreis





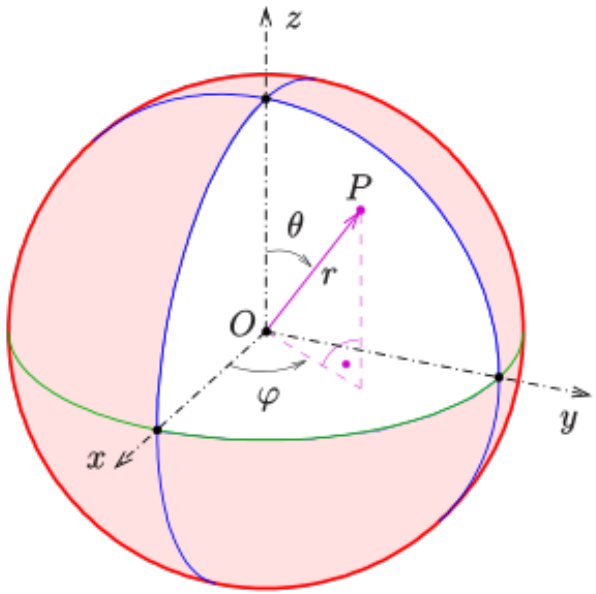
3D-Modelle



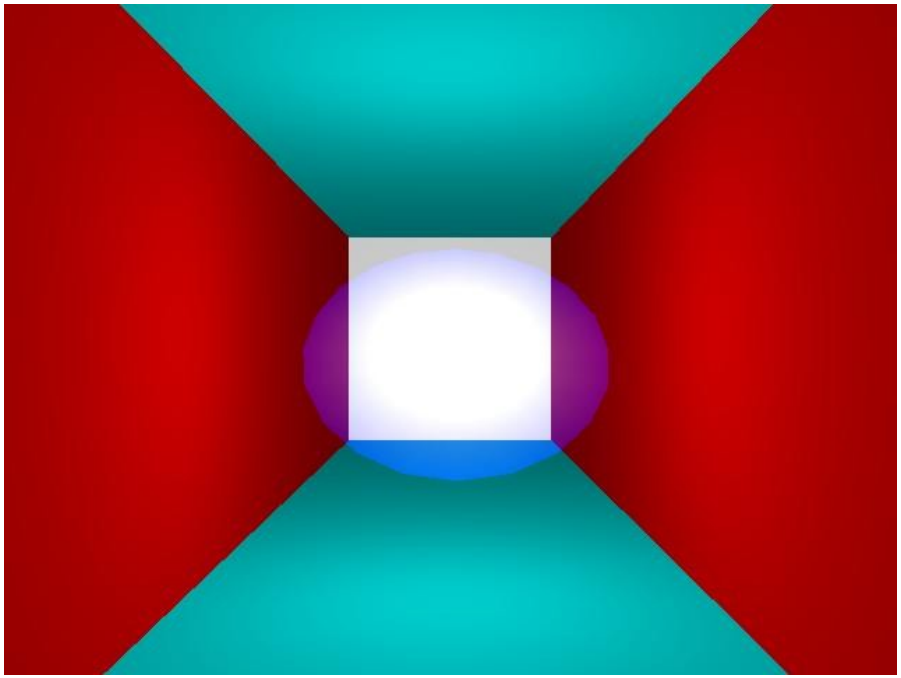
- Ball
  - Erzeugung der Vertices durch Berechnung der Koordinaten auf der Einheitskugel
  - Formeln:

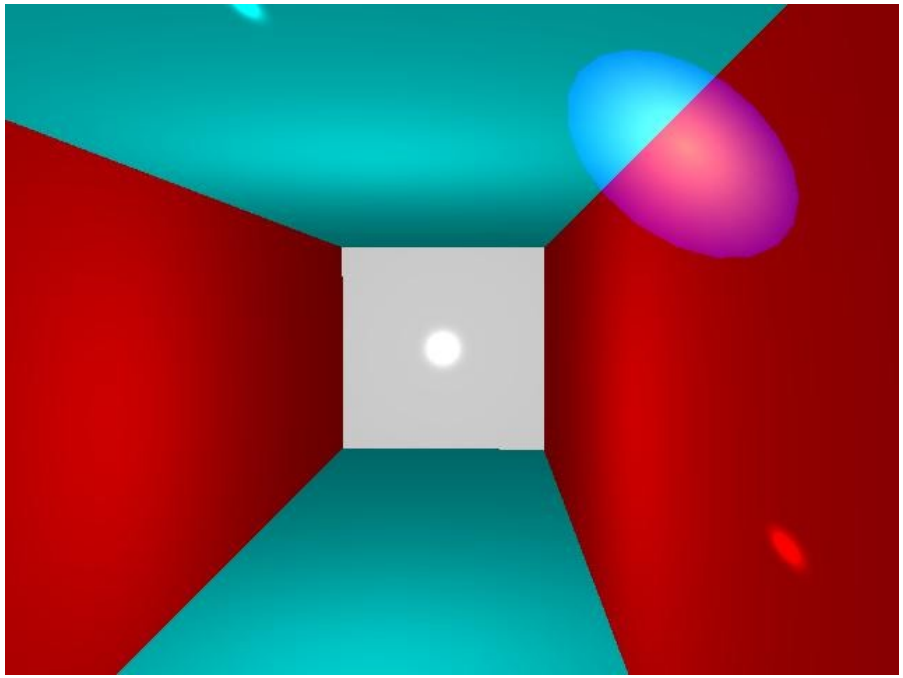
$$x = \sin(\theta) * \cos(\phi)$$
$$y = \sin(\theta) * \sin(\phi)$$
$$z = \cos(\theta)$$

  - Winkelkoordinaten (intuitiv):
    - θ bestimmt den Ring
    - ϕ bestimmt den Punkt auf dem Ring

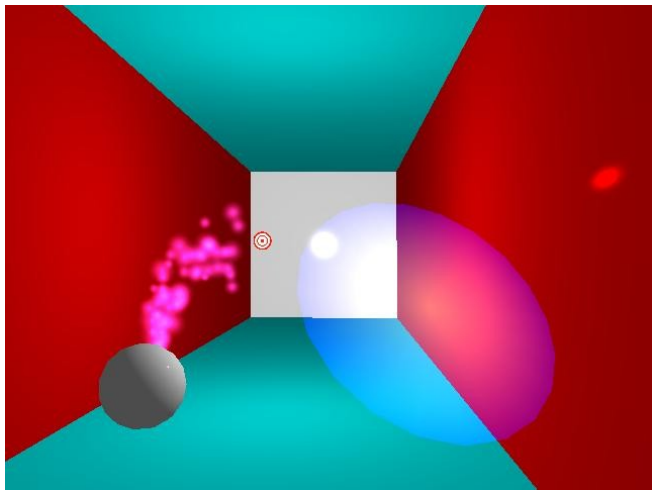


Kamerabewegung





Partikelsystem



- Komponenten des Systems:
  - Particle-Klasse
    - Position
    - Geschwindigkeit
    - Lebensdauer
    - Größe
  - BallParticleRenderable-Klasse
    - Verwaltung der Partikel (emittieren, aktualisieren, löschen)
    - Rendern aller Partikel

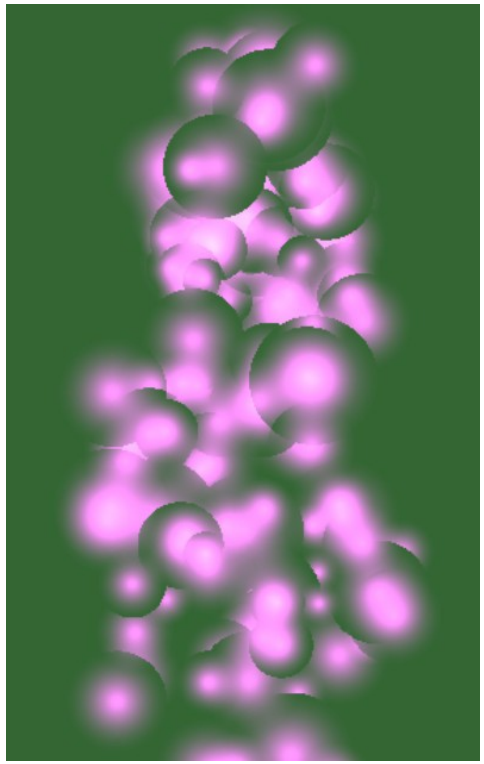
Partikelsystem

- Instanced Rendering
  - Rendering von mehreren Instanzen desselben Modells mit nur einem Render-Call
  - Anlegen eines Array-Buffers für Model-Matrizen der Partikel in BallParticleRenderable
  - Bei jedem Update Model-Matrizen neu berechnen und an Buffer senden

0	v_1	v_2	v_3	v_4	v_5	v_6
1	t_1	t_2	t_3	t_4	t_5	t_6
2	M_1	M_2	• • •			M_n

Partikelsystem

- Blending
  - Erlaubt es Farben miteinander zu mischen
  - Teil der Fragmentverarbeitung in der Rendering-Pipeline
  - Sich überlagernde Partikel sollen einen leuchtend weißen Bereich bilden



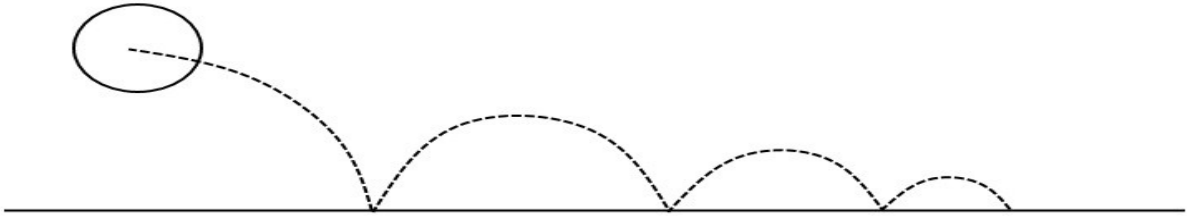
Partikelsystem

- Tiefentest
  - Transparenz eines Fragments spielt keine Rolle für Tiefentest
  - Lösung: Tiefenpuffer non-writeable schalten

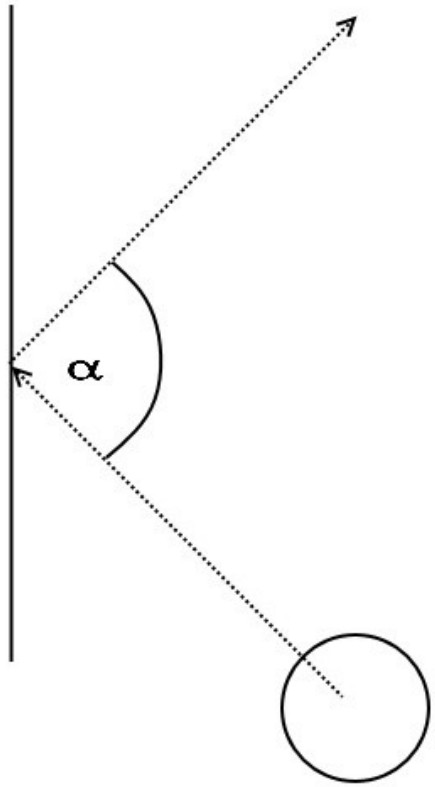
Inhaltsverzeichnis

Physik

- Ziel
  - Realistische Ballbewegungen

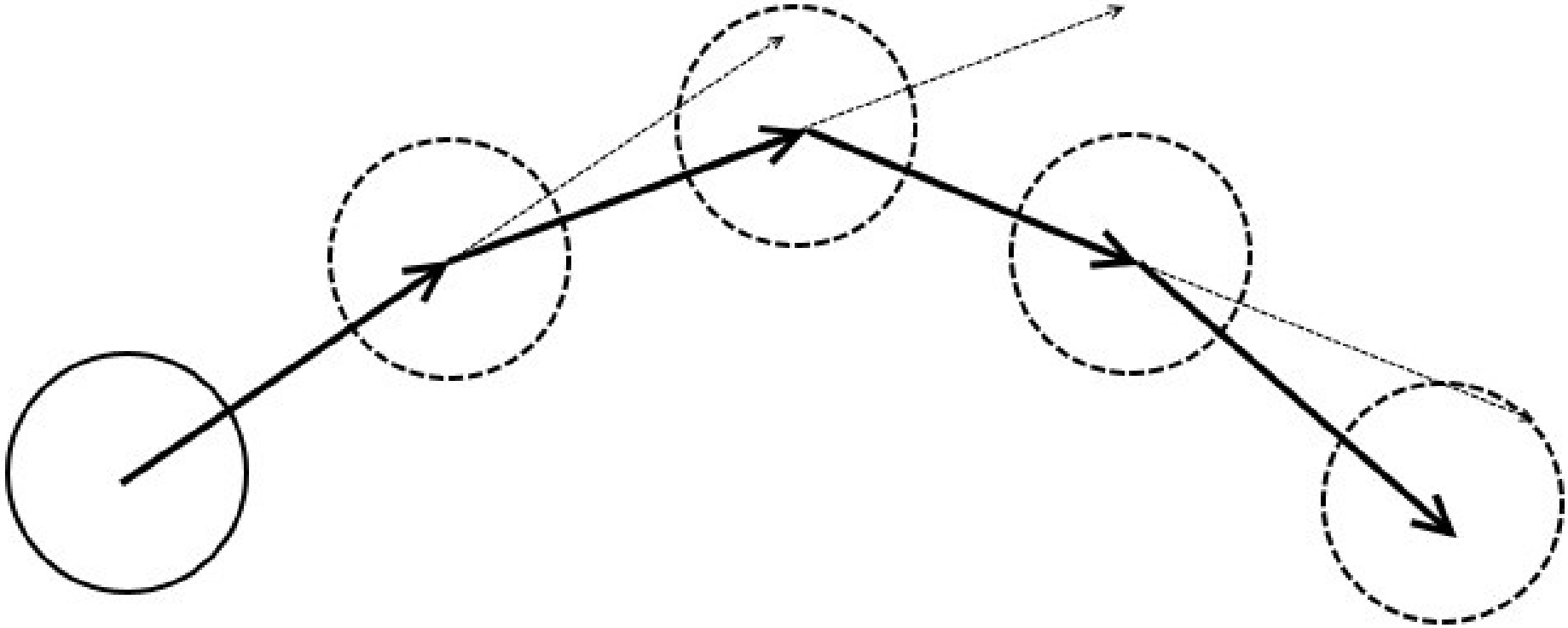


  - Nachvollziehbare Reflexion



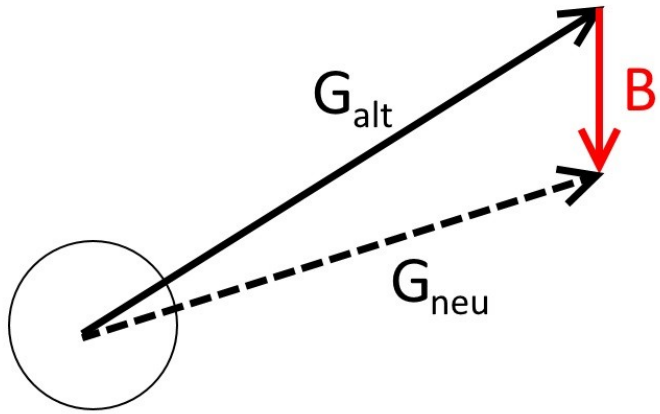
Ballbewegungen

- Gravitationsfaktoren



Ballbewegungen

- Bei jedem Tick:
  - Berechnung des Geschwindigkeitsvektors



$$Beschleunigung = \left(\frac{1}{Masse} + Gravitation\right) * Zeit$$

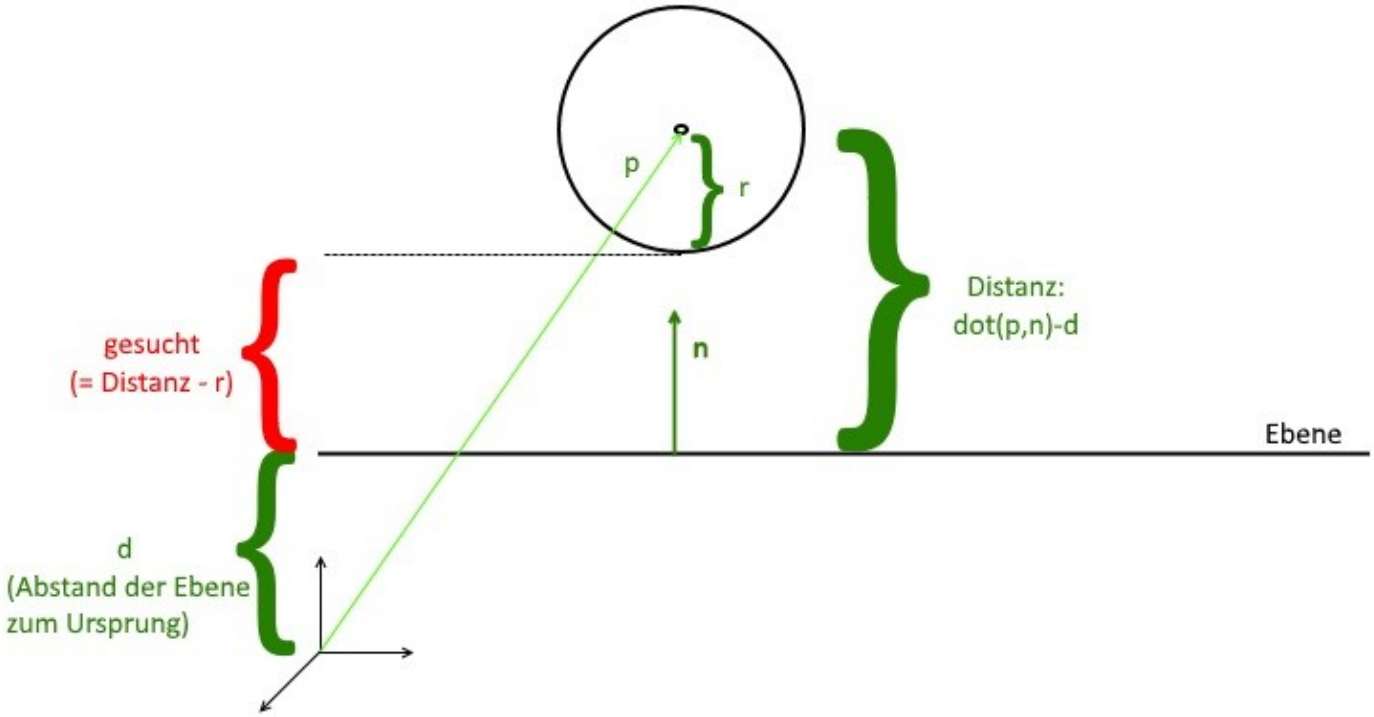
$$Geschwindigkeit_{Neu} = Geschwindigkeit_{Alt} + Beschleunigung$$

  - Berechnung der Neuen Position

$$Position_{Neu} = Position_{Alt} + Geschwindigkeit_{Neu} * Zeit$$

Kollisionen und Reflexionen

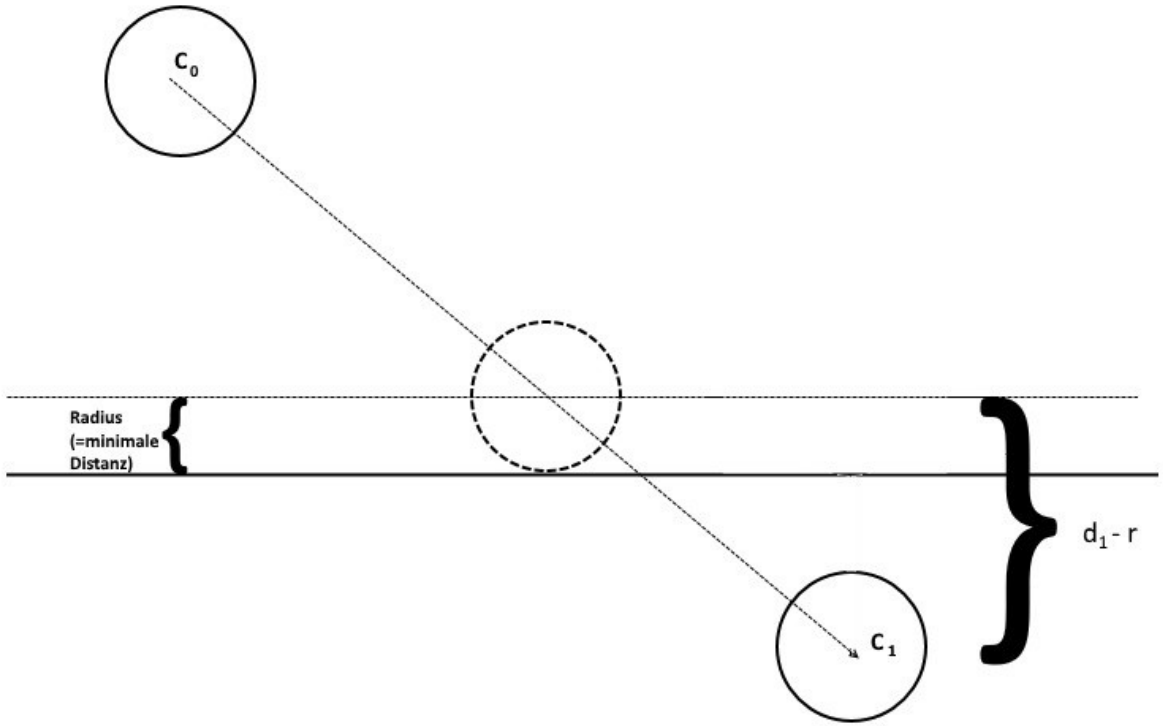
- Berechnung der Abstand zur Ebene



Kollisionen und Reflexionen

- Falls der Ball „hinter“ die Ebene gelangt (bei hohen Geschwindigkeiten)

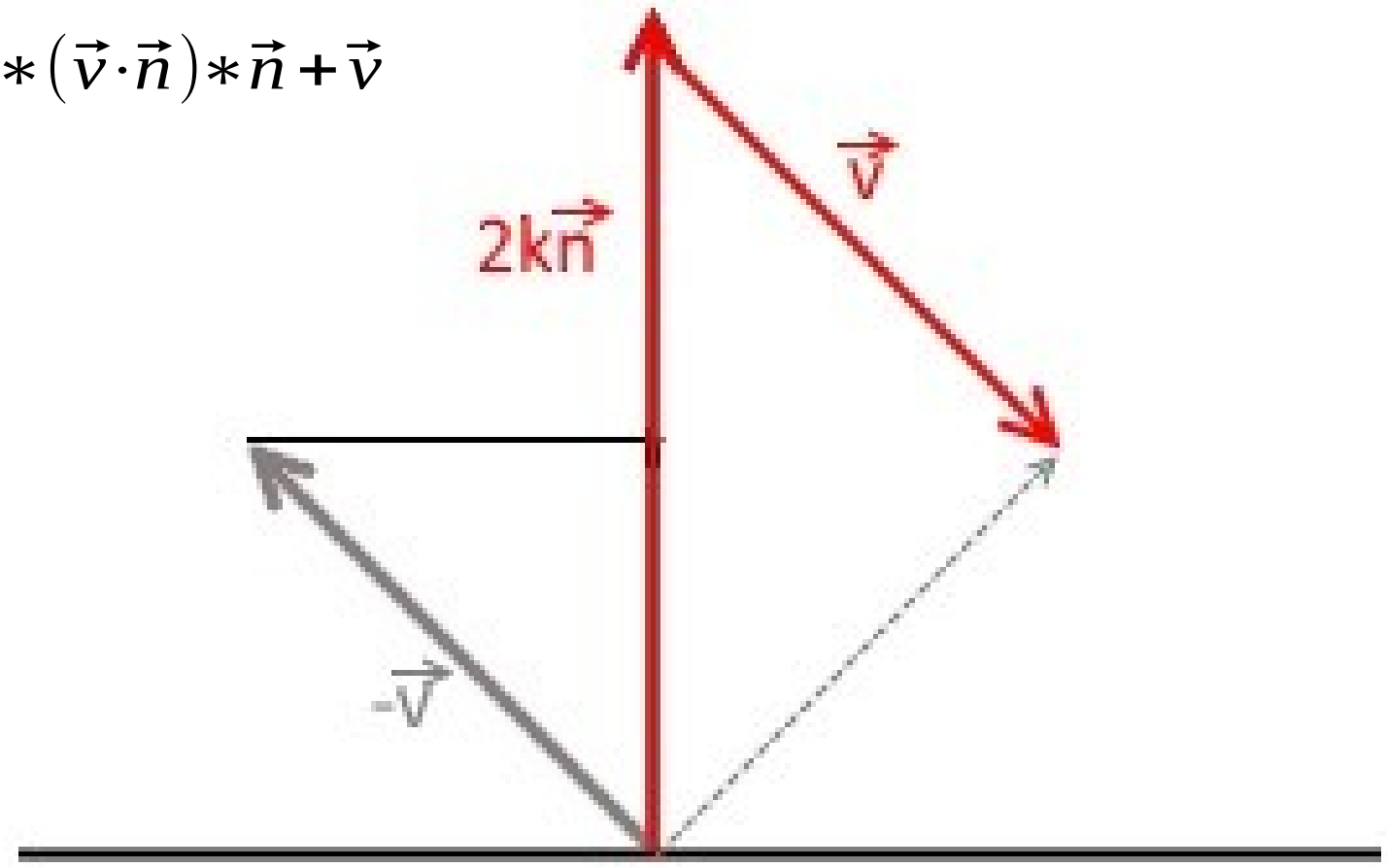
  => Interpolation der Ballposition



Kollisionen und Reflexionen

- Bei Kollision: Reflexion gemäß Reflexionsgesetz

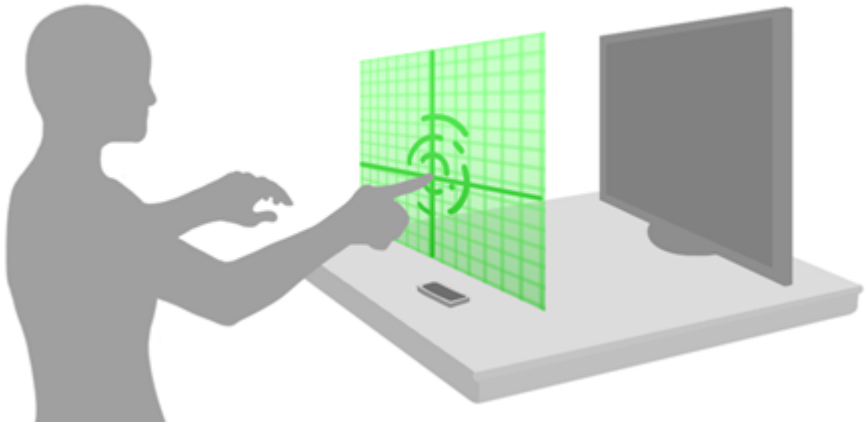
$$\vec{u} = -2 * (\vec{v} \cdot \vec{n}) * \vec{n} + \vec{v}$$



Inhaltsverzeichnis

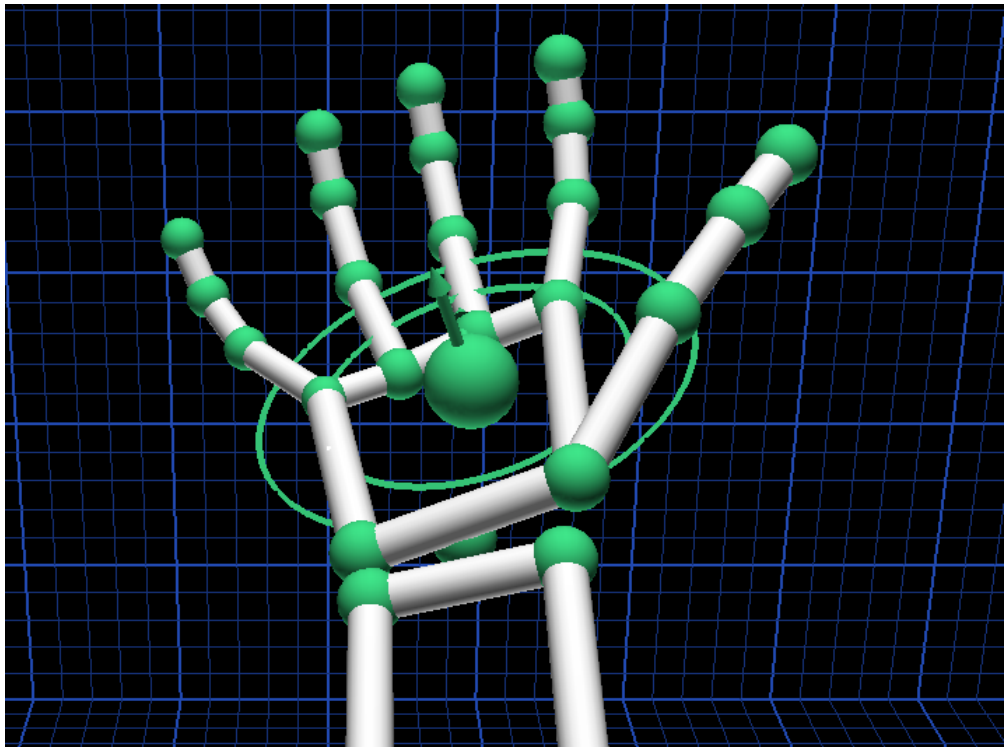
Steuerung

- Eingabe durch Leap Motion
- Rechte Hand steuert Schläger
- Linke Hand kann Ball zurücksetzen



Rechte Hand

- Steuert Position des Schlägers
- Normalvektor bestimmt Ausrichtung
- Geschwindigkeit wird erfasst



Linke Hand

- Kann mit einer Kreisgeste den Ball zurücksetzen