

# 05 Visuelle Effekte und Feedback-Systeme

---

Ein intuitives visuelles Feedback ist entscheidend, um dem Benutzer Orientierung in der komplexen 3D-Welt zu geben. Nodges implementiert hierfür ein mehrschichtiges System aus Highlights und Animationen.

## 05.1 Das Highlight-System (**HighlightManager**)

Der **HighlightManager** ist die zentrale Instanz für alle visuellen Hervorhebungen. Er verwaltet, welches Objekt gerade warum hervorgehoben wird und stellt sicher, dass sich Effekte nicht gegenseitig stören.

### Architektur der Highlight-Verwaltung

Das System basiert auf einer **Highlight-Registry** (`Map<Object3D, HighlightData>`). Jedes aktive Highlight wird dort registriert, zusammen mit Metadaten (Typ, Zeitstempel, Original-Material).

- **Vermeidung von Konflikten:** Wenn ein Objekt bereits markiert ist (z.B. durch eine Selektion), verhindert die Registry, dass ein niedriger priorisierter Effekt (z.B. ein Hover-Effekt einer anderen Komponente) den Status überschreibt.
- **Material-Sicherheit:** Bevor ein Objekt visuell verändert wird, legt der Manager ein Backup des originalen Materials an. Beim Entfernen des Highlights wird dieser Originalzustand exakt wiederhergestellt. Dies verhindert "Geister-Effekte", bei denen Objekte versehentlich die Highlight-Farbe behalten.
- **Cleanup:** Eine automatische Bereinigung entfernt Highlights, die nicht mehr gültig sind (z.B. wenn ein Objekt gelöscht wird).

### Priorisierung

Effekte haben implizite Prioritäten. Ein **SELECTION**-Effekt ist "stärker" als ein **HOVER**-Effekt. Der Manager sorgt dafür, dass ein selektiertes Objekt seinen Status behält, auch wenn die Maus darüber hin- und herbewegt wird.

## 05.2 Highlight-Modi im Detail

Das System unterscheidet fünf spezifische Modi, die unterschiedliche semantische Bedeutungen haben und visuell unterscheidbar sind.

### 1. HOVER-Modus (Temporäre Interaktion)

- **Zweck:** Signalisiert "Interaktivität". Zeigt an, welches Element bei einem Klick ausgewählt würde.
- **Aktivierung:** Automatisch durch Raycasting bei Mausbewegung.
- **Visuell (Nodes):** Helligkeit +20%, Cyan-transparenter Halo-Umriss, Opacity 0.3.
- **Visuell (Edges):** Cyan-blauer Umriss (TubeGeometry mit größerem Radius), Opacity 0.8.

### 2. SELECTION-Modus (Fokus)

- **Zweck:** Dauerhafter Fokus auf ein Element zur Detailansicht. Triggert das Info-Panel.
- **Aktivierung:** Linksklick auf ein Objekt.
- **Visuell:** Grüner Glow (RGB 0,1,0), erhöhte Intensität (0.4-1.0).

- **Besonderheit:** Nutzt eine **animierte Pulsation**, um die Aufmerksamkeit dauerhaft zu binden (siehe 05.3).

### 3. SEARCH-Modus (Finden)

- **Zweck:** Hervorhebung von Ergebnissen einer Suchanfrage (z.B. "Finde Node 'Server-01'").
- **Aktivierung:** Programmatisch durch Suchfunktion.
- **Visuell:** Hochkontrastierendes Gelb (0xFFFF00) mit Cyan-Glow. Bleibt aktiv, bis die Suche gelöscht wird.

### 4. PATH-Modus (Analyse)

- **Zweck:** Visualisierung von Verbindungen, z.B. der kürzeste Weg zwischen zwei Knoten.
- **Aktivierung:** Algorithmen (Dijkstra, BFS).
- **Visuell:** Cyan (0x00FFFF) für alle Elemente entlang des Pfades. Ermöglicht das Verfolgen von Linien durch das "Dickicht" des Graphen.

### 5. GROUP-Modus (Clustering)

- **Zweck:** Kennzeichnung von Communities oder Kategorien.
- **Aktivierung:** Cluster-Algorithmen oder manuelle Gruppierung.
- **Visuell:** Benutzerdefinierte Farbe (Standard: Magenta), die konsistent auf alle Mitglieder der Gruppe angewendet wird.

## 05.3 Der Glow-Effekt (**GlowEffect**)

Nodges nutzt keine teuren Post-Processing Shader (wie Unreal Bloom) für den Standard-Glow, um Performance auf mobilen Geräten zu sparen. Stattdessen wird ein geometrischer und material-basierter Ansatz verfolgt.

### Technik: Emissive Materials & Halos

- **Emissive Property:** Three.js Materialien besitzen eine **emissive** Eigenschaft (Selbstleuchten). Diese wird genutzt, um die Grundfarbe des Objekts zu überstrahlen.
- **Halo-Meshes:** Für den "Schein" um ein Objekt herum (Outline) wird ein separates, transparentes Mesh erzeugt, das etwas größer ist als das Originalobjekt ("Shell"-Technik).

### Animierte Pulsation

Aktivierte Objekte (besonders im Selection-Modus) "atmen". Diese Animation wird vom **StateManager** gesteuert.

#### Funktionsweise:

- Eine **animate()**-Schleife läuft mit 60 FPS.
- Ein Oszillator berechnet die Intensität basierend auf einer Sinus-Kurve oder einem Ping-Pong-Algorithmus (0.0 -> 1.0 -> 0.0).
- Formel: **newIntensity = currentIntensity + deltaTime \*  $\pi$  \* 0.2 \* frequency \* direction**.
- Dieser Wert steuert direkt die **emissiveIntensity** des Materials.

Dies erzeugt einen organischen, lebendigen Eindruck, der den Benutzer subtil daran erinnert, welches Objekt gerade aktiv ist.

---

*Ende Kapitel 05*