Technische Hochschule Rosenheim

Entwicklung von Computerspielen: Angewandte Renderoptimierung in Unity

Fakultät Informatik
FWPM

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Übersicht

- UI, häufige Probleme
- Batching
- Light/ Shadow Baking
- Occlusion

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
UI Probleme in Unity

- Damit in Unity UI Elemente gerendert werden, müssen diese auf einem Canvas Objekt liegen (Ausnahme manche TexMeshPro Elemente)

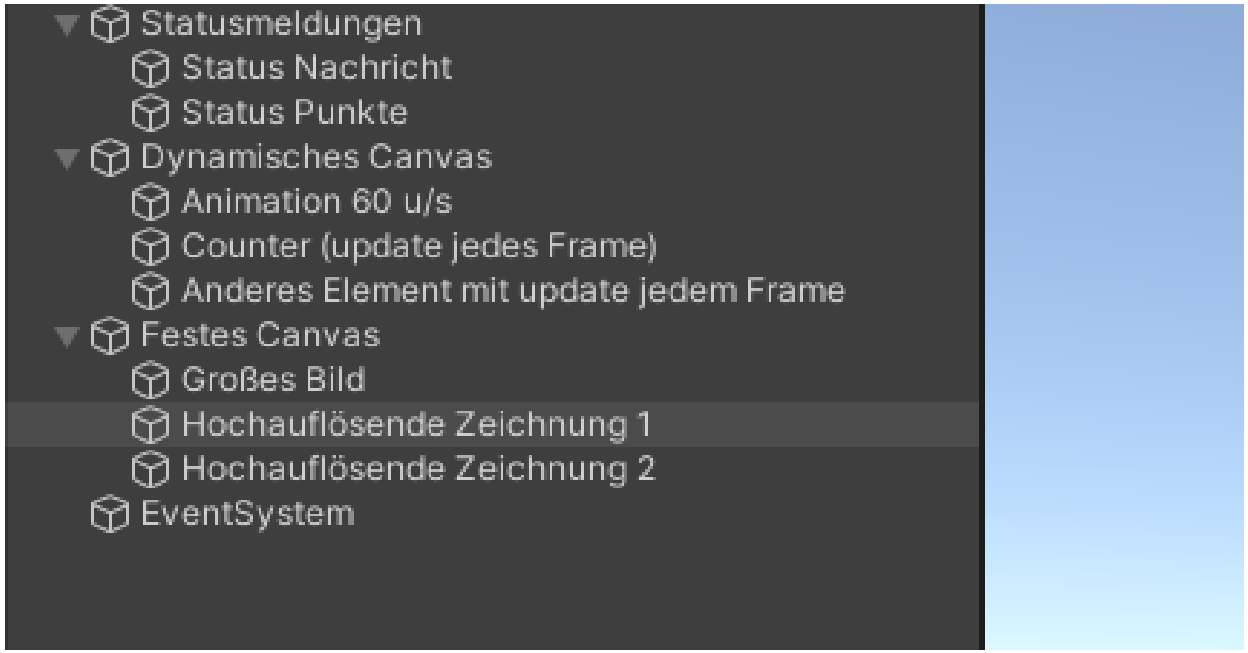
  Ein Objekt liegt dann auf einem Canvas, wenn es ein Kindobjekt von diesem ist.

- Das Canvas Objekt wird jedes mal neu gebaut, wenn eine Änderung bei dessen Kindobjekten ausgelöst wird. (z.B. Animation, Counter, etc.)

  Das erneute Bauen eines großen Canvas verwendet sehr viele Ressourcen und beeinträchtigt die Performance erheblich.
  → Gruppierung von Elementen mit ähnlicher Updatezeit in eigenem Canvas
  → aufgeblähte Canvas Objekte nur, wenn diese nie verändert werden.

Angewandte Renderoptimierung in Unity
UI Probleme in Unity

- Beispiel zu Canvas Verteilung:



Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Batching

- Was ist Batching?

  Vereinfacht gesagt reduziert Batching die Menge an Vertices / Meshes, welche gerendert werden sollen in größere Meshes / Vertice Gruppen, sodass Draw Calls gespart werden können.

- Was sind Draw Calls?

  Draw Calls sind Anweisungen der Engine an die Grafik Api ein Objekt auf den Bildschirm zu Zeichnen. Im Normalfall entsteht durch jeden Draw Call ein signifikanter CPU Overhead in der Grafik API.
  → Weniger Draw Calls eigentlich immer besser

Angewandte Renderoptimierung in Unity
Batching; Arten von Batching

- Dynamisches Batching

  Dynamisches Batching kombiniert mehrere Vertices von Objekten mit gleichem Material und Single Pass Shader (Sehr einfache Shader) und schickt diese zusammen an die API.

  → Funktioniert Automatisch, verursacht einen geringen CPU Overhead. (Bei Apple Metal normalerweise nicht zielführend, da API sehr effizient)

- Statisches Batching

  Kombiniert mehrere Meshes in ein größeres Mesh, wenn das Material das Selbe ist (Instanzen von Materials sind gleich, aber nicht das Selbe). Funktioniert nur bei Meshes, welche als Static deklariert wurden und sich daher garantiert nicht Bewegen oder Skalieren.

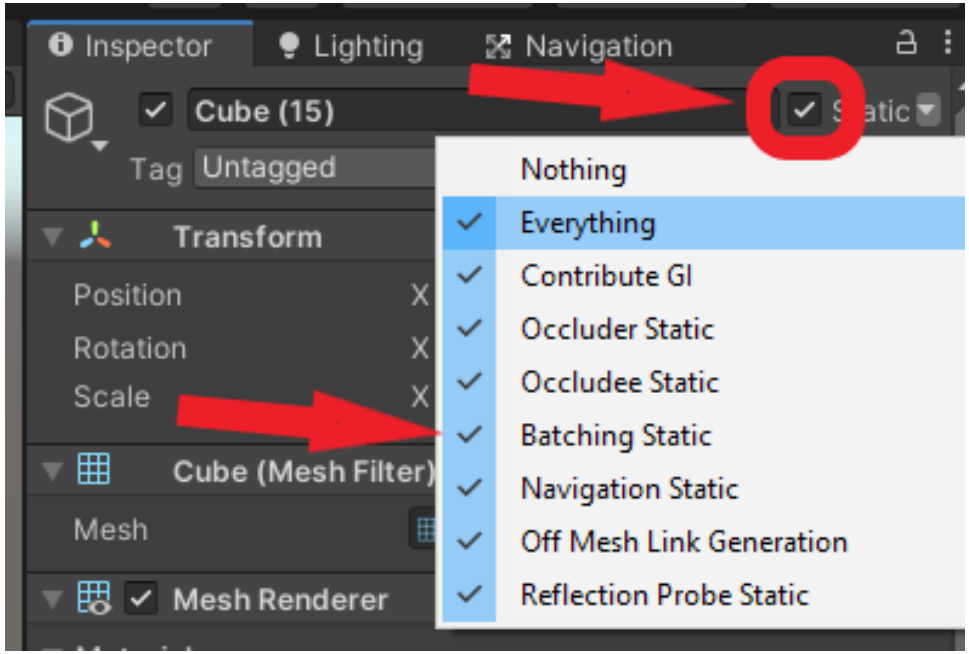
  → Verursacht einen Memory Overhead

  → Obergrenze des Batchings liegt bei 64k Vertices (OpenGLES 48k, Mac OS 32k)

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Batching; Static Batching

- Static Batching. Wie?



Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Batching; Kombination von Meshes

- Alternative zu Batching: Manuelles Kombinieren von Meshes
  - Entweder bereits beim Modell
  - Oder in Unity mit .CombineMeshes
    - https://docs.unity3d.com/ScriptReference/Mesh.CombineMeshes.html

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Licht / Schatten

- Grundsätzlich: Licht in Spielen funktioniert nicht direkt wie echtes licht, sondern eher wie eine strukturierte Einfärbung von Texturen an Stellen, an denen diese das Licht erreicht. Genauso Schatten, nur anders herum.

  Physikbasierte Ansätze versuchen das Verhalten von Licht zu simulieren
  → Global Illumination (GI)
  → PBR (Physics Based Rendering)
  → Raytracing (In Unity nur in High Defenition Rendering Pipeline (HDRP))

Angewandte Renderoptimierung in Unity
Licht / Schatten

- Beleuchtung und Schatten müssen die Geometrie aller beteiligten Objekte in die Berechnung des Schattenwurfs miteinbeziehen. Auch auf Seiten des Schattennehmers muss die Geometrie miteinbezogen werden.

  → Je nach Detailgrad/ Approximationsstufe des Lichts, sehr leistungsintensiv

Angewandte Renderoptimierung in Unity
Licht / Schatten

- Daher: Verschiedene Ansätze Schatten (und Licht) zu berechnen:
  - Realtime (Echtzeit)
  - Baked
  - Mixed
- Echtzeit
  - Berechnung findet parallel zum Spiel statt. Größte Performancekosten, kaum Mehraufwand in der Entwicklung des Spiels (Standard), Große Anpassungsfähigkeit.
- Baked
  - Die Meisten / Alle Schatten- und Beleuchtungsberechnungen finden zur Compilezeit oder davor statt. (In der Regel) geringste Performancekosten. Wenig Anpassungsfähigkeit, keine bewegten Objekte mit Schatten. Erheblicher Mehraufwand zur Entwicklungszeit.

Angewandte Renderoptimierung in Unity
Licht / Schatten

- Mixed:

  Licht und Schatten für statische Objekte (Häuser ...) werden vorberechnet, während dynamische Objekte(Spieler, Gegner etc.) zur Echtzeit Schatten werfen.

  → reduzierte Performance Kosten
  → bewegte Objekte bekommen Schatten
  → trotzdem: erheblicher Aufwand bei der Entwicklung
  → klare Einteilung in Statische und Dynamische Objekte nötig

Technische Hochschule Rosenheim

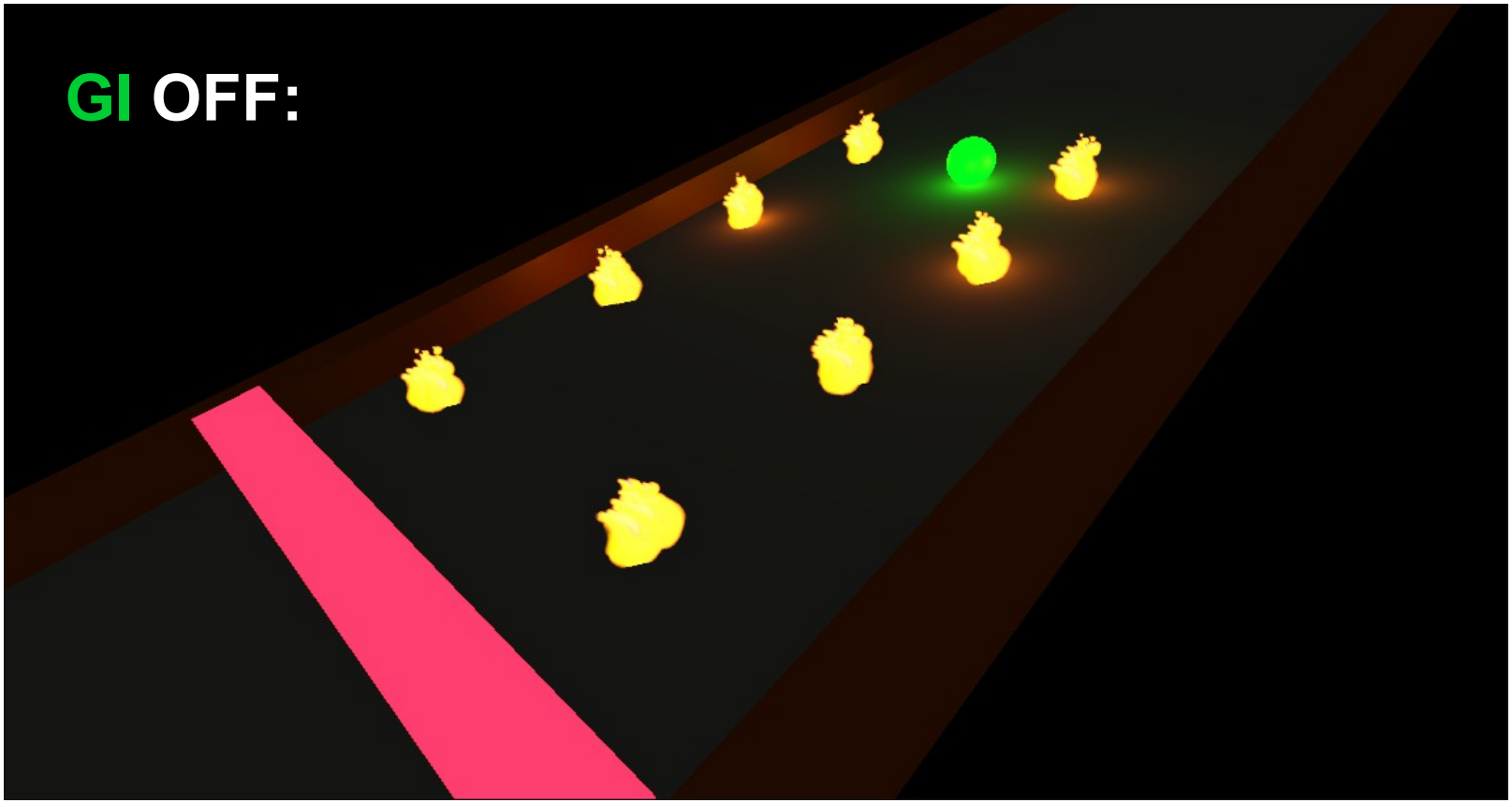
Angewandte Renderoptimierung in Unity
Licht / Schatten Lightmapping

- Lighting Fenster über Window > Rendering > Lighting aufrufbar
  Mehr dazu auf https://docs.unity3d.com/Manual/Lightmapping.html

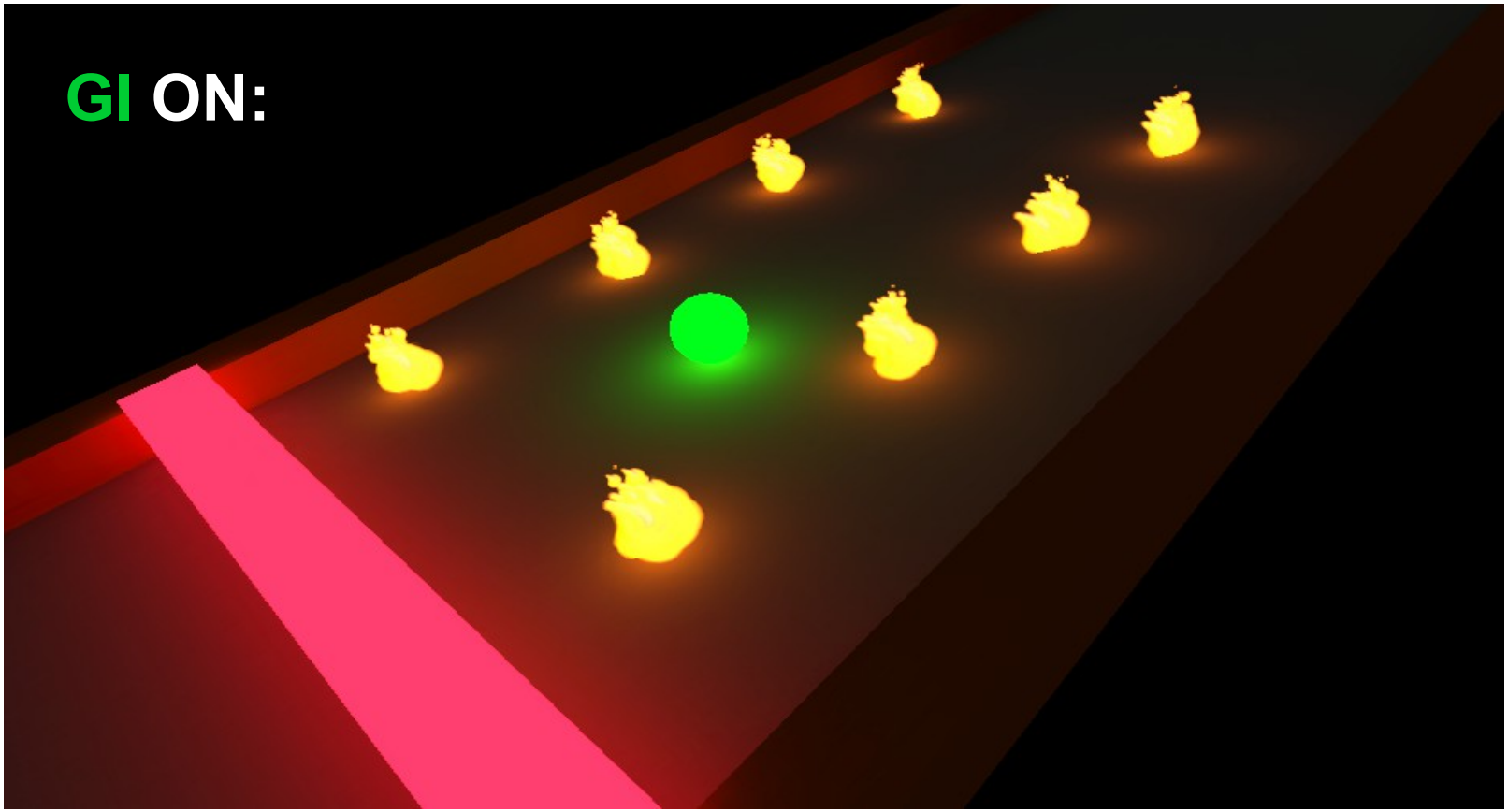
- Verschiedene Lighting Modes für Global Illumination:
  - Shadowmask
  - Baked Indirect
  - Subtractive

Warum GI?:
Lässt Spiele wesentlich besser aussehen, als die Modelle es sonst ermöglichen würden

Angewandte Renderoptimierung in Unity
Warum Global Illumination;



Angewandte Renderoptimierung in Unity
Warum Global Illumination;



Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Licht / Schatten Shadowmask

- Shadowmask benutzt vorberechnete Indirekte Lichtberechnung und Echtzeit Schatten.
- Eigenschaften von Shadowmask Mode zusammengefasst:

  Sehr Hohe Performancekosten
  Beste / realistischte Schatten und Lichtberechnung
  Schatten in der Distanz möglich
  Speichert zusätzliche Daten in Lightprobes

  → Shadowmask ist die beste Wahl für realistische Schattenberechnung und hat erhöhte Performanceansprüche

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Licht / Schatten Baked Indirect

- Baked Indirect lässt alle Mixed Lichter wie Echtzeit Lichter funktionieren. Lediglich die Indirekte Beleuchtung der GI wird vorberechnet. Anders als Shadowmask werden Schatten nur bis zur Schadow Distance (Definierbar in Project Settings) berechnet

- Zusammengefasst:
  - Relativ Hohe Ansprüche an die Performance
  - Effektiv kaum Unterschied zwischen Mixed Lights und Realtime
  - Lightprobes werden verwendet für die Beleuchtung von Dynamischen Objekten mit indirektem Licht

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Licht / Schatten Subtractive

- Subtractive erlaubt nur einem Directional Light Schatten von Dynamischen Objekten Casten zu lassen (z.B. Sonne). Alle Schatten von Statischen Objekten werden vorberechnet und in einer Lightmap gespeichert, genauso wie direktes und indirektes Licht aller Lichtquellen.

- Zusammengefasst:
  - Geringe Ansprüche an die Performance
  - Alle Mixed Lichter werden Effektiv zu Baked Lichtern, bis auf die Sonne
  - Kombination von Echtzeitschatten und Baked Schatten erfolgt nicht in der Engine und muss durch stilistische Mittel verschleiert werden.
  - Nicht sonderlich Realistisch, aber für Stilisierte Artstyles durchaus Sinnvoll

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Occlusion Culling

- Culling verringert, wie mittlerweile bekannt die Komplexität der zu rendernden Geometrie durch Weglassen unnötiger / unsichtbarer Teile.
- Occlusion Culling ermöglicht die Entfernung von Objekten, welche im View Frustrum der Kamera nicht sichtbar sind.

- Siehe auch: https://docs.unity3d.com/Manual/OcclusionCulling.html

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Occlusion Culling

- Culling verringert, wie mittlerweile bekannt die Komplexität der zu rendernden Geometrie durch Weglassen unnötiger / unsichtbarer Teile.
- Occlusion Culling ermöglicht die Entfernung von Objekten, welche im View Frustrum der Kamera nicht sichtbar sind.

- Siehe auch: https://docs.unity3d.com/Manual/OcclusionCulling.html

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Occlusion Culling; Verwendung

- Im Occlusion Culling Fenster: Window > Rendering > Occlusion Culling

- Bei jedem Mesh Renderer gibt es eine Option, ob Unity diesen in die Verdeckung einbeziehen soll als: Verdecker oder als Verdeckter.

- Im Occlusion Culling Fenster die Occlusion dann Baken

- Das ganze Kann aus der Szenensicht getestet werden mit der Visualisierung des Occlusion Cullings

Technische Hochschule Rosenheim

Angewandte Renderoptimierung in Unity
Zusammenfassung

- Großer Bereich. In jedem der genannten Bereiche kann man sich stark vertiefen, je nach Interesse / Bedarf
- Jeden Teil dieser Vorlesungseinheit muss man selbst ein wenig ausprobiert haben, um ein Händchen dafür zu entwickeln.
- Alle besprochenen Themen sind eine Gradwanderung und benötigen Übung. Genauso ist jeden der Teilbereiche richtig einzustellen eine Kunst für sich und es gibt dabei keinen Schlaghammer für Alles.

Vorsicht: Kreativ begeisterungsfähige Menschen können sich in diesen Einstellungen verlieren und das gesamte Zeitbudget nur für die genannten Punkte aufwenden. An einem Punkt **muss** man diese Arbeit **beenden**.