

Ray Tracing on Android

Kagerer Miriam
Baldinger Andreas
Shehata Abdu

20.06.2021

Version 1.1



1 Einleitung

1.1 Zweck des Dokuments

Das Dokument dient dazu den Verlaufs des Projekts zu dokumentieren und auf den Source Code genauer einzugehen. Weiters werden mit dem Dokument die Bedingungen, Ziele und Vorgaben festgelegt. Zusätzlich sollen die Endresultate, Messungen/Benchmarks, tatsächliche Kosten und tatsächliche Dauer zur Evaluierung enthalten sein.

1.2 Gültigkeit des Dokuments

Der Rahmen dieses Dokuments umfasst das gesamte Projekt "Raytracing on Android".

Änderungen können und werden von den drei Hauptverantwortlichen vorgenommen.

Die Aktualität des Dokuments ist anhand der Versionsnummer und des Versionsdatums prüfbar. Die aktuellste Version nimmt immer Bezug auf die aktuellste Version des Projekt-Codes und der Projektdokumentation.

1.3 Begriffsbestimmungen und Abkürzungen

OpenGL - Open Graphics Library - Spezifikation einer plattform- und programmiersprachen übergreifenden Programmierschnittstelle zur Entwicklung von 2D- und 3D-Computergrafik Anwendungen.

ES - Embedded Systems - Die Version von OpenGL die unter anderem für mobile Endgeräte entwickelt wurde und auf Android nutzbar ist.

VCS - Version Control System (Beispiel folgt)

Git - Eine freie Software zur verteilten Versionsverwaltung von Dateien

FPS - Frames Per Second - Maß für die Bildfrequenz

API - Application Programming Interface - Programmteil, der von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird.

GUI - Graphical User Interface - bezeichnet eine Form von Benutzerschnittstelle eines Computers.

IDE - integrated development environment - eine Sammlung von Computerprogrammen, mit denen die Aufgaben der Softwareentwicklung möglichst ohne Medienbrüche bearbeitet werden können.

1.4 Überblick über das Dokument

Der folgende Teil des Dokumentes befasst sich mit den unten angeführten Punkten:

- 2) Allgemeinen Beschreibung der Produkts
- 3) Detaillierte Beschreibung der Produktmerkmale
- 4) Vorgaben an die Projektabwicklung
- 5) Verpflichtungen des Auftraggebers
- 6) Verwendete Literatur

2 Allgemeine Beschreibung des Produkts

2.1 Zweck des Produkts

Der Zweck der App ist es zu zeigen, dass Raytracing auf Android implementiert werden kann und dass aktuelle, mobile Hardware performant genug ist Realtime-Raytracing zu betreiben.

Der entscheidende Vorteil gegenüber der etablierten Render-Methode "Rasterization" ist, dass sich mit Raytracing naturgetreue Lichtberechnungen durchführen lassen, die in einem realistischer aussehenden Bild resultieren.

In der Filmindustrie profitiert man schon lange von den Vorzügen die Raytracing mit sich bringt. In Realtime-Computer-Grafikanwendungen wie Spielen (Bilder können nicht zu einem früheren Zeitpunkt vor-gerendert werden) war dies aber mit der bisherigen Hardware-Leistung nicht möglich, auf mobilen Endgeräten nahezu undenkbar.

2.2 Abgrenzung und Einbettung des Produkts

Das Produkt soll folgende Merkmale aufweisen:

- Siehe Kapitel 3
- Ausführbarkeit auf OpenGL ES 3.1 (oder höher) kompatiblen Android-Smartphone-Systemen
- Echtzeit-Fähigkeit
- Support für beliebig viele Szenen und Objekte (unter Einhaltung der Echtzeit-Fähigkeit)
- Bedienbarkeit durch Nutzer-Gesten

Das Produkt soll folgende Merkmale nicht aufweisen:

- Vom Gerät abhängige Ausführung (Android als Betriebssystem hat alleinig die Aufgabe die Hardware anzusprechen)
- Von Softwareversionen abhängige Ausführung (die oben definierten Android und OpenGL Abgrenzungen sind einzuhalten, weitere Versionsunterschiede basierend auf zum Beispiel Hersteller oder Standort des Nutzers dürfen aber keine Auswirkungen haben)

2.3 Überblick über die geforderte Funktionalität

Die App soll fähig sein, eine simple 3D-Szene darzustellen, welche durch Steuerung der Kamera betrachtet werden kann.

2.4 Allgemeine Einschränkungen

Schnittstellen:

- Android native Komponenten

Standards:

- OpenGL ES 3.1

Methoden/Klassen:

- Activity
- GLSurfaceView
- Renderer
- Eigene Kameraklasse

2.5 Vorgaben zu Hardware und Software

Die App soll auf Android Geräten mit OpenGL ES 3.1 und aufwärts laufen. Die API wird erst ab Android 5 (API Level 21) verfügbar und läuft somit nicht auf älteren Geräten. Die App wird in der Entwicklungsumgebung Android Studio entwickelt. Als Version Control System wird git benutzt.

2.6 Benutzer des Produkts

Die Zielgruppe des Projektes sind hauptsächlich Technikinteressierte, es benötigt aber keine besondere Ausbildung oder Vorkenntnisse, da es ein möglichst benutzerfreundliches Design haben soll.

3 Detaillierte Beschreibung der geforderten Produktmerkmale

3.1 Lieferumfang

Das Produkt ist eine App und enthält somit keine Lieferbaren Produkte.

3.2 Abläufe (Szenarien) von Interaktionen mit der Umgebung

Use Case Nummer: 1

Der Benutzer startet das Programm und steuert die Kamera der Szene mithilfe der in 3.4.1 beschriebenen Funktion.

3.3 Ziele des Benutzers

In Use Case Nummer 1 kann der Benutzer unter Verwendung der in 3.4.1 beschriebenen Funktion die gesamte Szene überblicken. Er kann verdeckte Objekte sichtbar machen und die durch den Betrachtungswinkel verursachte Echtzeit-Veränderung des Lichteinfalls betrachten.

3.4 Geforderte Funktionen des Produkts

Die App soll fähig sein, eine simple 3D-Szene darzustellen, welche durch Steuerung der Kamera betrachtet werden kann.

3.4.1 Kamerasteuerung

3.4.1.1 Wirkungsweise der Kamera-Steuerung

Durch Streichen über den Bildschirm lässt sich die Kamera und damit die Ansicht/Szene rotieren.

3.4.1.2 Abhängigkeiten / Randbedingungen

Die Geschwindigkeit der Kamera-Rotation wird in einer zentralen Kamera-Klasse einstellbar sein und kann sich zum Beispiel szenen-spezifisch ändern. Dies gilt auch für weitere in Verbindung stehende Kamera-Eigenschaften wie zum Beispiel "Field-of-View" und "View Frustum" (beschreiben Größe und Tiefe der Ansicht).

3.4.2 Szenendarstellung

3.4.2.1 Wirkungsweise der Szenendarstellung

Es soll eine Szene bestehend aus verschiedenen Objekten gerendert und schattiert werden.

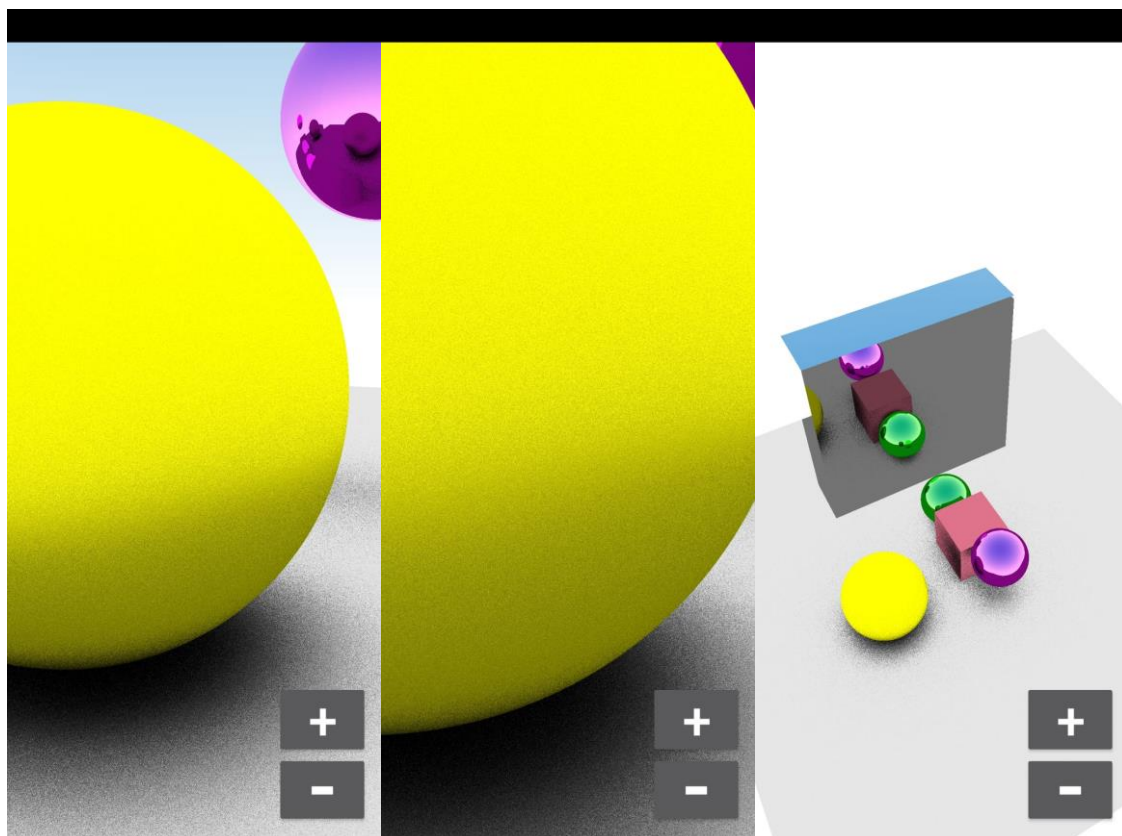
3.4.2.2 Abhängigkeiten / Randbedingungen

Je nach Material wird ein Objekt unterschiedlich schattiert und kann so zum Beispiel einen metallischen Glanz oder eine matte Schattierung erhalten.

3.5 Externe Schnittstellen des Produkts

3.5.1 Benutzerschnittstellen (User Interfaces)

Die GUI besteht aus der "Raytrayed Surface" und 2 Buttons. Die Surface enthält eine Szene durch Gestensteuerung gesteuert werden kann. Dies erlaubt eine Rotation mit 2 Fingern und Translation mit 1 Finger. Die Buttons erlauben es in die Szene rein und aus der Szene raus zu zoomen.



3.5.2 Systemschnittstellen

- Android
 - Activity: Wird benötigt um eine App zu starten.
 - GLSurfaceView: Diese View ist eine Erweiterung der SurfaceView um OpenGL zu ermöglichen.
 - Renderer: Ermöglicht das Zeichnen der Szene auf dem GLSurfaceView

3.6 Sonstige geforderte Produktmerkmale

Es folgt eine Beschreibung der nicht funktionalen Anforderungen:

3.6.1 Geschwindigkeitsmerkmale (performance)

- Reaktionszeiten/Inputlag (auf zum Beispiel die in 3.4.1 beschriebene Kamerasteuerung/Fingerbewegung)
- Anlaufzeiten (App-Start, Szenen-Ladezeit)
- Durchsatzrate/Bildwiederholrate (FPS, Logic-Updates/Sekunde)

3.6.2 Ressourcenmerkmale (resources)

Die folgenden Punkte fallen alle unter den Überpunkt bzw. die Resource "Smartphone)

- Datenmengen
- CPU-Bedarf
- CPU-Auslastung (möglicherweise Per-Core-Auslastung bei Multithreading)
- GPU-Bedarf
- GPU-Auslastung
- Speicher
- Arbeitsspeicher

3.6.3 Schutzmerkmale (security)

Es bestehen keine besonderen Notwendigkeiten von zusätzlichen Schutz des Produktes, weswegen die App nur die Standard Schutzfunktion von Android-Studio verwendet.

3.6.4 Sicherheitsmerkmale (safety)

Smartphones, die OpenGL nicht verwenden, können diese App nicht herunterladen, da dies im Manifest der App festgelegt wurde.

Beim Beenden der App werden sämtliche Ressourcen wieder freigegeben, um nicht unnötig Speicherplatz zu verwenden.

3.6.5 Portabilitätsmerkmale (portability)

Die Portabilität des Produktes liegt in seiner Natur als Smartphone-App. Es ist allerdings auf Android begrenzt und nicht auf andere mobile Betriebssysteme übertragbar.

Die App kann einen Eintrag im Google Play Store erhalten, um dort als Download zur Verfügung gestellt zu werden bzw. auf diversen Websites erhältlich sein.

3.6.6 Wartungsmerkmale (maintenance)

Die App enthält nur 2 native Komponenten (Buttons) auf die geachtet werden muss, da Android bei den OpenGL Surface Funktionen keine Änderungen einbaut, die nicht abwärtskompatibel sind. Da Buttons essenzielle Elemente in Android Apps sind, können mit jedem neuen Update öfters Design- und Funktionsänderungen aufkommen. Da OpenGL immer abwärtskompatibel bis zu Version 2 ist, erleichtert dies die Wartungsarbeit.

3.6.7 Wiederverwendbarkeitsmerkmale (reuse)

Ist eine spätere Wiederverwendung von Produktteilen gewünscht, werden die Anforderungen daran festgelegt.

3.6.8 Benutzbarkeitsmerkmale (usability)

Aufgrund des benutzerfreundlichen Designs dürften keine Bedienungsfehler auftreten.

4 Vorgaben an die Projektabwicklung

4.1 Anforderungen an die Realisierung

- Hardware
 - Smartphone
- Software
 - Android Betriebssystem
 - OpenGL
 - Java als Programmiersprache

4.2 Fertige und zugekaufte Komponenten

Für die Tests sind bereits private PCs vorhanden auf denen gratis Software (Android Studio als IDE) und von der FH zur Verfügung gestellte Software (Microsoft Office) installiert wurde.

Es sind außerdem private Smartphones vorhanden, die für die Applikations-Testungen auf echten Endgeräten verwendet werden. Dies ist manchmal notwendig, weil der in Android Studio integrierte Emulator OpenGL ES Version 3 nicht unterstützt.

4.3 Unterauftragnehmer

Es sind keine Unterauftragnehmer vorhanden.

4.4 Abnahmebedingungen

Das Projekt wird abgenommen, sobald alle Rahmenbedingungen erfüllt sind.

4.5 Lieferbedingungen

Die Software wird per Git übergeben, sobald alle Rahmenbedingungen erfüllt sind.

5 Verpflichtungen des Auftraggebers

Der Auftraggeber ist verpflichtet einen Projektbetreuer, sowie Dokumentation zur Verfügung zu stellen. Zusätzlich erhalten alle Projektteilnehmer Schulungsmaßnahmen in Form der Vorlesung "Projektmanagement und Tools".

6 Verwendete Literatur

Shireley, P. (2020). Ray Tracing in One Weekend. v3.2.3
<https://raytracing.github.io/books/RayTracingInOneWeekend.html>

Joe de Vries (2020). Learn OpenGL.
<https://learnopengl.com>

Displaying graphics with OpenGL ES, 2020-11-18 UTC.
<https://developer.android.com/training/graphics/opengl>

OpenGL ES, 2021-01-11 UTC.
<https://developer.android.com/guide/topics/graphics/opengl>