WebGPU Game Engine

Autoren

Abd El Rahaman Shehata

Datum: 8.06.2022, Version 1

Table of Contents

[1 Einleitung 4](#_Toc108375630)

[*1.1* Zweck des Dokuments 4](#_Toc108375631)

[*1.2* Gültigkeit des Dokuments 4](#_Toc108375632)

[*1.3* Begriffsbestimmungen und Abkürzungen 4](#_Toc108375633)

[1.4 Zusammenhang mit anderen Dokumenten 4](#_Toc108375634)

[*1.5* Überblick über das Dokument 5](#_Toc108375635)

[2 Allgemeine Beschreibung des Produkts 6](#_Toc108375636)

[*2.1* Zusammenhang mit bereits laufenden Projekten 6](#_Toc108375637)

[*2.2* Zusammenhang mit Vorgänger- und Nachfolgeprojekten 6](#_Toc108375638)

[*2.3* Zweck des Produkts 6](#_Toc108375639)

[*2.4* Abgrenzung und Einbettung des Produkts 6](#_Toc108375640)

[*2.5* Überblick über die geforderte Funktionalität 6](#_Toc108375641)

[*2.6* Allgemeine Einschränkungen 6](#_Toc108375642)

[*2.7* Vorgaben zu Hardware und Software 6](#_Toc108375643)

[*2.8* Benutzer des Produkts 6](#_Toc108375644)

[3 Detaillierte Beschreibung der geforderten Produktmerkmale 7](#_Toc108375645)

[*3.1* Lieferumfang 7](#_Toc108375646)

[*3.2* Abläufe (Szenarien) von Interaktionen mit der Umgebung 7](#_Toc108375647)

[*3.3* Ziele des Benutzers 7](#_Toc108375648)

[*3.4* Geforderte Funktionen des Produkts 7](#_Toc108375649)

[3.4.1 Funktion 3D Modell anzeigen 7](#_Toc108375650)

[3.4.2 Funktion Spiel Entwickeln 7](#_Toc108375651)

[*3.5* Externe Schnittstellen des Produkts 8](#_Toc108375652)

[3.5.1 Benutzerschnittstellen (User Interfaces) 8](#_Toc108375653)

[3.5.2 Systemschnittstellen 8](#_Toc108375654)

[*3.6* Sonstige geforderte Produktmerkmale 8](#_Toc108375655)

[3.6.1 Geschwindigkeitsmerkmale (performance) 8](#_Toc108375656)

[3.6.2 Ressourcenmerkmale (resources) 8](#_Toc108375657)

[3.6.3 Schutzmerkmale (security) 8](#_Toc108375658)

[3.6.4 Sicherheitsmerkmale (safety) 8](#_Toc108375659)

[3.6.5 Portabilitätsmerkmale (portability) 8](#_Toc108375660)

[3.6.6 Zuverlässigkeit (reliability) 8](#_Toc108375661)

[3.6.7 Wartungsmerkmale (maintenance) 8](#_Toc108375662)

[3.6.8 Wiederverwendbarkeitsmerkmale (reuse) 8](#_Toc108375663)

[3.6.9 Benutzbarkeitsmerkmale (usability) 9](#_Toc108375664)

[4 Vorgaben an die Projektabwicklung 10](#_Toc108375665)

[*4.1* Anforderungen an die Realisierung 10](#_Toc108375666)

[*4.2* Fertige und zugekaufte Komponenten 10](#_Toc108375667)

[*4.3* Unterauftragnehmer 10](#_Toc108375668)

[*4.4* Abnahmebedingungen 10](#_Toc108375669)

[*4.5* Lieferbedingungen 11](#_Toc108375670)

[*4.6* Gewährleistung 11](#_Toc108375671)

[5 Verpflichtungen des Auftraggebers 12](#_Toc108375672)

[6 Literaturverweise 12](#_Toc108375673)

[7 Anhang 12](#_Toc108375674)

# Einleitung

## Zweck des Dokuments

Dieses Dokument dient zur Spezifikation der Mindestziele des Projekts. Zusätzlich dient es als Grundlage zur Bewertung des Semester Projekts.

## Gültigkeit des Dokuments

Dieses Dokument besitzt eine Gültigkeit für das Projekt „WebGPU Game Engine“ im Sommersemester 2022. Sofern das Projekt im nächsten Semester weiterverfolgt wird, wird die Gültigkeit verlängert und die Spezifikation dementsprechend angepasst. Dieses Projekt setzt auf kein bereits vorhandenes Pflichtenheft auf. Für die allfällige Änderungen ist das gesamte Projektteam verantwortlich unter Absprache mit dem Projektleiter.

## Begriffsbestimmungen und Abkürzungen

**Abkürzungen**

* glTF**:** Graphics Language Transmission Format
* FPS: Frames Per Second
* OpenGL: Open Graphics Library
* WGSL: WebGPU Shading Language

**Begriffsbestimmungen**

* WebGPU: Eine standardisierte, moderne GPU API für Native- und Web-Anwendungen
* WGSL: Beschreibt Programme, die auf der GPU über WebGPU laufen.
* glTF: Ein standardisiertes Dateiformat für 3D Szenen und Modelle.
* Rust: Eine moderne, performante und zuverlässige System-Programmiersprache
* VS Code: Visual Studio Code ist ein kostenloser Quelltext-Editor von Microsoft.
* FPS: Maß für die [Bildfrequenz](https://de.wikipedia.org/wiki/Bildfrequenz)
* OpenGL: [Spezifikation](https://de.wikipedia.org/wiki/Spezifikation) einer plattform- und programmiersprachen übergreifenden [Programmierschnittstelle](https://de.wikipedia.org/wiki/Programmierschnittstelle) zur Entwicklung von 2D- und 3D-[Computergrafik Anwendungen](https://de.wikipedia.org/wiki/Computergrafik)
* Wireframe Modell: 3D Körper, welcher nur durch Kanten (Edges) repräsentiert/angezeigt wird.

## Zusammenhang mit anderen Dokumenten

Die in diesem Pflichtenheft angeführten Aspekte des Umfangs und die genauen Spezifikationen unterliegen keinem Lastenheft.

Weitere Dokumente welche gelegentlich erwähnt werden können sind:

* Systemarchitektur Dokumentation
* Kurzbeschreibung des Projekts
* Die Projektbeschreibung

## Überblick über das Dokument

Die exakte Auflistung dieses Dokuments mit allen Unterpunkten findet sich im Inhaltsverzeichnis wieder.

Im Anschluss werden die wichtigsten Unterpunkte dieses Dokuments kurz zusammengefasst.

Unter Punkt 1 findet sich eine Allgemeine Übersicht des Dokuments.

Unter Punkt 2 findet sich eine erste Beschreibung des Projekts. Diese beinhaltet unter anderem den Zweck des Projekts sowie eine Abgrenzung.

Unter Punkt 3 findet sich eine detaillierte Beschreibung aufbauend auf der Beschreibung unter Punkt 2. Zusätzlich wird auf die externen Schnittstellen eingegangen.

In Abschnitt 4, finden sich schließlich die Vorgaben an die Projektabwicklung, wie Anforderungen an die Realisierung.

Schließlich finden sich hier auch Informationen über die Abnahme- und Lieferbedingungen.

# Allgemeine Beschreibung des Produkts

Die App soll ist ein Experiment und soll es ermöglichen Objekte (3D Modelle) anzeigen können und auch ein minimaler „Game Engine“ sein. Der Fokus liegt aber auf das Experimentieren und das Einlesen in die Thematik.

## Zusammenhang mit bereits laufenden Projekten

Das Thema Game Engine wurde von einer anderen Gruppe auch gewählt, jedoch benutzen die Kollegen OpenGL und Java für das Projekt.

## Zusammenhang mit Vorgänger- und Nachfolgeprojekten

Es besteht kein Vorgänger Projekt. Das Nachfolgeprojekt erweitert die Funktionen der App.

## Zweck des Produkts

Das Projekt dient dazu da Erfahrungen zum Thema „Graphics Programming“ zu sammeln und gleichzeitig die WebGPU API zu testen.

## Abgrenzung und Einbettung des Produkts

Der Game Engine ist kein Produktionsreifes Produkt. Es ist ausschließlich zur Wissenserweiterung und zum Testen entwickelt worden. Dies kann sich in Zukunft ändern, aber ist nicht der Fokus.

## Überblick über die geforderte Funktionalität

Die geforderte Funktionalität ist das Anzeigen eines Objekts mithilfe der WebGPU API.

## Allgemeine Einschränkungen

Die einzige Einschränkung ist die Nutzung der WebGPU API.

## Vorgaben zu Hardware und Software

Es gab keine Vorgaben, aber das System ist in der Lage auf jedem gängigen Betriebssystem zu laufen, da WebGPU für Browser entwickelt wurde.

## Benutzer des Produkts

Der Game Engine ist für Softwareentwickler gedacht, die entweder schon Erfahrung mit Grafikprogrammierung haben oder damit anfangen wollen.

# Detaillierte Beschreibung der geforderten Produktmerkmale

## Lieferumfang

Das Produkt ist eine Software-Bibliothek bzw. eine App.

## Abläufe (Szenarien) von Interaktionen mit der Umgebung

* 3D Modell anzeigen
* Spiel entwickeln
* WebGPU Performance testen

## Ziele des Benutzers

Der Benutzer soll in der Lage sein ein Modell anzuzeigen, wie in Blender, Unity, Unreal Engine, etc.

## Geforderte Funktionen des Produkts

Für dieses Semester soll das Projekt in der Lage sein ein Objekt anzuzeigen. Die Quelle des Objekts ist unwichtig. Aufbauend auf diese Funktion kann ein Spiel entwickelt werden.

### Funktion 3D Modell anzeigen

#### Wirkungsweise

Der Pfad zu einer glTF-Datei wird im Code angeben und dem glTF-Loader übergeben und dann damit die App gestartet. Die Datei wird in diesem Prozess eingelesen und dann in einem Fenster als Wireframe angezeigt.

#### Abhängigkeiten / Randbedingungen

Ein 3D Modell im glTF Format gespeichert.

### Funktion Spiel Entwickeln

#### Wirkungsweise

Das Projekt ist derzeit noch nicht in dazu in der Lage. Der Entwickler müsste immer noch viele Teile selbst programmieren, welche normalerweise von einem Game Engine abstrahieren werden.

#### Abhängigkeiten / Randbedingungen

Derzeitig keine Abhängigkeiten

## Externe Schnittstellen des Produkts

### Benutzerschnittstellen (User Interfaces)

Derzeit besitzt der Game Engine keine GUI, bis auf das Fenster, welches das Modell anzeigt. Es muss alles im Code bearbeitet werden.

### Systemschnittstellen

#### WebGPU

WebGPU kommuniziert mit der Grafikkarte über Metal/Vulkan/DirectX abhängig vom Betriebssystem.

## Sonstige geforderte Produktmerkmale

Beschreiben der nicht funktionalen Anforderungen.

### Geschwindigkeitsmerkmale (performance)

Die Performance des Systems und der API sind ziemlich vielversprechend. Derzeit laufen Animationen (Z-Achsen Rotation und rein-/rauszoomen) mit 60FPS.

### Ressourcenmerkmale (resources)

Die minimalen Ressourcenmerkmale wurden nicht getestet, jedoch ist das derzeitige System in der Lage auf integrierten GPUs zu laufen und dies mit 60FPS

### Schutzmerkmale (security)

Das System benutzt keine externen Schnittstellen und greift auch nicht auf das Netzwerk zu, was Eingriffe erschwert. Aber es sind keine Sicherheitsmechanismen eingebaut.

### Sicherheitsmerkmale (safety)

Ist nicht im Rahmen des Projektes spezifiziert worden.

### Portabilitätsmerkmale (portability)

Das Projekt ist Plattformunabhängig, da die benutzten Bibliotheken alle gängigen Betriebssysteme und Hardwarearchitekturen unterstützen. Rust macht das Kompilieren für jedes Betriebssystem und beliebte Hardwarearchitektur ziemlich Trivial.

### Zuverlässigkeit (reliability)

Keine Abstürzte bis jetzt bekannt.

### Wartungsmerkmale (maintenance)

Keine Vorhanden

### Wiederverwendbarkeitsmerkmale (reuse)

Derzeit sind viele Komponenten noch verknüpft, aber dies soll sich in Zukunft ändern

### Benutzbarkeitsmerkmale (usability)

Das Projekt ist noch klein und kann über die Dokumentation und das Lesen des Codes verstanden werden.

# Vorgaben an die Projektabwicklung

## Anforderungen an die Realisierung

* Hardware
  + Entwicklungsrechner
    - Windows Rechner
    - Macbook Pro 15“ 2013
* Software
  + Betriebssysteme (Host und Target)
    - Host:
      * MacOS/Linux/Windows
    - Target:
      * MacOS/Linux/Windows
      * Auch Möglich sind Android/IOS
  + Compiler, Bibliotheken
    - Rustup (inkludiert Rustc, Cargo)
    - Wgpu
  + Entwicklungsumgebung
    - VS Code

## Fertige und zugekaufte Komponenten

Es werden keine Komponenten benötigt

## Unterauftragnehmer

Es gab keine externen Unterauftragnehmer.

## Abnahmebedingungen

* Rahmenbedingungen
  + Wogegen wird abgenommen (Pflichtenheft)?

Es wird nach der Requirements Specification abgenommen.

* + Wie wird abgenommen (gemeinsam beim Kunden, oder Kunde testet)  
    Die Abnahme erfolgt durch Johannes Lugstein
  + Wer stellt Testdaten bereit? Wann sind diese Daten bereitzustellen?

Das Projekt enthält mehrere Testmodelle und auch GIFs zum herzeigen.

* + Wo wird abgenommen?

Das komplette Projekt befindet sich zur Abnahme auf Gitlab

* + Wer unterzeichnet das Abnahmeprotokoll.

Johannes Lugstein

* Abnahmekriterien
  + Festlegung des Abnahmetests

-

* + Wann ist die Abnahme erfolgreich (Resfehlerquote)?  
    Wenn das Projekt auf GitLab gepusht wurde.
  + Werden die nicht funktionalen Anforderungen erfüllt?  
    Ja.
* Abnahmenunterlagen  
  -
* Gutachten, Sicherheitsnachweise  
  -
* Erfüllung von Vorschriften und Normen

-

## Lieferbedingungen

Das Projekt ist über das Herunterladen des Source-Codes zur Verfügung gestellt.

## Gewährleistung

Das Projekt ist nur für Testzwecke und zum Experimentieren und hat damit auch keine Gewährleistung.

# Verpflichtungen des Auftraggebers

Das Projekt hat keinen Auftraggeber.

# Literaturverweise

Joe de Vries (2020). Learn OpenGL.

<https://learnopengl.com>

Learn Wgpu

[https://sotrh.github.io/learn-wgpu](https://sotrh.github.io/learn-wgpu/)

Alain Galvan (2019). Raw WebGPU

<https://alain.xyz/blog/raw-webgpu>

WebGPU Spezifikation

<https://gpuweb.github.io/gpuweb>

WGSL Spezifikation

<https://gpuweb.github.io/gpuweb/wgsl>

# Anhang