**Belegarbeit: Internettechnologien**

Erstellung einer Musikvisualisierungsanwendung in JavaScript unter Nutzung der Three.js Bibliothek

**Vorgelegt am:**

**Von:**  Josia Rudolph, Martin Görner

**Studiengang:** Technische Informatik

**Studienrichtung:** Technik

**Matrikelnummer:** s4004602,

**Gutachter:** Prof. Dr. Mathias Sporer (Staatliche Studienakademie Glauchau)

Vorwort

Inhaltsverzeichnis

[Vorwort II](#_Toc150589713)

[Inhaltsverzeichnis III](#_Toc150589714)

[Abbildungsverzeichnis IV](#_Toc150589715)

[Abkürzungsverzeichnis V](#_Toc150589716)

[1. Zielsetzung 1](#_Toc150589717)

[2. Grundlagen 2](#_Toc150589718)

[2.1. Programmierumgebung 2](#_Toc150589719)

[2.2. Das Three.js-Modul 2](#_Toc150589720)

[2.2.1. WebGL 2](#_Toc150589721)

[2.2.2. Three.js 2](#_Toc150589722)

[2.3. Programmierstil 2](#_Toc150589723)

[3. Evaluierung 3](#_Toc150589724)

[3.1. Visualisierungskonzept 3](#_Toc150589725)

[3.2. Umsetzung in Typescript 3](#_Toc150589726)

[3.3. Installationsanleitung 3](#_Toc150589727)

[3.4. Musikvisualisierer 3](#_Toc150589728)

[4. Fazit 4](#_Toc150589729)

Abbildungsverzeichnis

Abkürzungsverzeichnis

HTML Hyper-Text-Markup-Language

API Application Programming Interface

UML Unified Modelling Language

1. Zielsetzung

Musikvisualisierungsapp mit javascript

Musik abspielen

Coole Animationen auf Musik reagierend

Mit Three.js auseinandersetzten

Konzept Entwicklung

Plattformübergreifend und auf jedem Browser nutzbar

1. Grundlagen
   1. Programmierumgebung

Node.js: (npm Paketmanager und Liveserver/Runtime Environment) -> Laufzeitumgebung und LiveServer, Ausführung von JS ohne Browser auf Server oder Befehlszeile; kommt mit Paketmanager; verarbeitung großer datenmengen durch skalierbarkeit dank Asynchronem Verhalten -> für effiziente und Leistungsfähige Platformen, Nachteil: muss installiert werden, Laufzeitumgebung mit Compiler bzw. Transpier für typescript in JavaScript

Installation der benötigten Module und Bibliotheken per Befehlszeile

Ausführung der Kompilierung und Liveserver

* 1. Das Three.js-Modul
     1. WebGL

Effektiver grphik renderer

* + 1. Three.js

Modul für js, das unteranderem webgl nutzt/vereinfacht

Wie funktioniert Musik in Three.js: Laden (THREE.AudioLoader), abspielen (THREE.Audio), -> über THREE.AudioListener (wie Mikro) dann hörbar

Möglichkeiten Musik zu analysieren

Was ist gleich nochmal fft?

* 1. Programmierstil

Typescript, was das

Vorteil gegenüber normalen JS

Was bringt Typescript in Bezug auf OOP zusätzlich mit sich

1. Evaluierung
   1. Visualisierungskonzept

Die Visualisierung zielt darauf ab, die Laustärke der analysierten Frequenzbereiche durch die Bewegung und Verfärbung von Objekten darzustellen.

Die Darstellung der Musikvisualisierung ist in 4 Einzelbereiche unterteilt.

Der Hintergrund wird von einem Balkendiagramm ausgefüllt. Die Aussteuerung der Balken entspricht der Lautstärke der analysierten Frequenzbereiche.

Im Vordergrund ist die Visualisierung in drei Frequenzbereiche aufgeteilt für die Darstellung der Bässe, der Mitten und der Höhen.

In einem Kreis um den Mittelpunkt herum befinden sich Objekte, welche sich, nach der Lautstärke der Frequenzen im Bassbereich richtung Mitte bewegen. Diese sollen sich je lauter sie sind, umso röter verfärben.

Aus dem Nullpunkt der Szene heraus entspringen Objekte, welche nach den analysierten Lautstäken des mittleren Frequenzbereiches animiert sind. Sie verteilen sich in einem Kreis um den Mittelpunkt. Die Austeuerung nach außen soll proportional zur Laustärke sein. Die roten, grünen und blauen Farbanteile verändern sich nach Frequenz und Lautstärke.

Von dem äußeren Bildschirmrand soll Objekte in Spiralförmigen Mustern nach inne Pulsiert kommen. Diese Animation wird durch die Lautstärken der hohen Frequenzanteile gesteuert. Für diesen Bereich sollen hellere Farben verwendet werden.

Die Visualisierungen der drei Frequenzanteile sollen ebenfalls rotieren. Die Objekte, welche die Bässe und die Höhen darstellen im gegen den Uhrzeigersinn und die Objekte der Mitten im Uhrzeigersinn.

Für einen Intensiveren visuellen Eindruck bewegt dich perspektive je nach dem niedrigsten verfügbaren Frequenzbereich von den Objekten weg. Aus den Bässen lässt sich zumeist ein Rhythmus der Musik ableiten. Ist der Abgespielte Ton im Bassbereich Laut, was darauf schließen lässt, dass sich die Musik um ihren stimmungstechnischen Höhepunkt befindet, sind alle Objekte vollumfänglich zu sehen.

Die konzeptionellen Anforderungen an die eingeteilten Bereiche der Visualisierung sind in Typescript umgesetzt wurden.

* 1. Umsetzung in Typescript

Die Umsetzung der Evaluierung in Typescript erfolgt in drei Schritten. Zunächst entsteht die Oberfläche eines Musikplayers mit Steuerelementen in HTML, mit Anpassungen im Aussehen und Aufrufen von Funktionen in Javascript durch ausgelöste Events beim Betätigen der Steuerelemente.

Der nächste Schritt ist die Erstellung einer THREE.js Szene in Typescipt. Es soll daneben eine Klasse entstehen, welche über die Audio-API von THREE.js verfügt. Die ausgelösten Events sollen auf die Methoden dieser Klasse zugreifen.

Im letzten Schritt wird die Visualisierung in einer separaten Klasse für Szenarien implementiert. Diese Klasse enthält Methoden zum Erschaffen von Objekten, zum Animieren dieser Objekte, sowohl wenn Musik gespielt wird als auch bei Pausen und einer Methode zum Zurücksetzten.

Ein grober Überblick über die Einbettung der Implementierungen ist in Abbildung 1 zu sehen.



**Abbildung 1,** Verschachtelungs-Skizze der Programmcodedateien

Am oberen Rand wird eine Leiste mit Steuerelementen erzeugt. Dies geschieht in HTML mit Symbolen in einer Tabelle.

Der Typescript-Quellcode ist in einem HTML-DIV-Tag eingebettet. In dem Programmcode befindet sich die THREE.js Szene und weitere Szenendefinitionen zum Erstellen einer Grundszene. Die Klassen für die Events, die Audio-API und die einzelnen Szenarien werden aus weiteren Typescript-Dateien eingebunden.

Der strukturelle Aufbau des Typescriptcodes ist in einem UML-Klassendiagramm in Abbildung 2 beschrieben. Jede Klasse ist in einer separaten Datei ausgelagert.

Die Event\_Handler-Klasse listet im Konstruktor alle benötigten Events für das Betätigen der Steuerelemente und Tastatureingaben und verknüpft diese mit den entsprechenden Funktionen. Die Methoden rufen Funktionen aus der Audio\_Processing klasse auf.

Die Audio\_Processing-Klasse initialisiert die Elemente aus dem THREE.js Modul zum Verwalten des Audios, zum Laden des Audios und zum Analysieren des Audios. Die Methoden zum laden, abspielen, anhalten und zurücksetzten des Audios werden von den Events aufgerufen. Die Klasse verfügt außerdem über eine Methode zum aufrufen der richtigen Visualisierungsanimationsfunktion basierend auf dem Status der Musik, ob diese abgespielt wird oder gerade pausiert ist.

Die Scenaries-Klasse ist ein Prototyp. Die erzeugten Kindklassen erben den Funktionsumfang und Membervariablen, um das initialisieren verschiedener Visualisierungsmethoden, projiziert auf die vier einzelnen Bereiche unter 3.1 Visualisierungskonzept beschrieben, zu vereinfachen.



**Abbildung 2,** Klassendiagramm Typescript Quellcode

* 1. Installationsanleitung
  2. Musikvisualisierer

Screenshots vom Programm mit kurzer Erklärung und Bezug auf 3.2

1. Fazit

3D-Charakter kommt nicht ganz raus

Three.js einfache Handhabung zum Erstellen von Szenen

Visualisierer Luft nach oben, aber funktioniert gut auch Plattformübergreifend