



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Audio- Video Programmierung Wintersemester 2018/19

„Nature meets Piano“

**Prof. Dr. Andreas Plaß
Jakob Sudau**

**Manuel Patti 2217273
Anas Garwal 2195430**

Die Programmierung des Web Audio Projekts besteht aus drei Teilen: HTML, JavaScript und dem CSS teil. Die Website ist strategisch so aufgebaut, dass der Benutzer ohne großen Aufwand herausfinden kann, wie die Seite zu bedienen ist. Im ersten Teil der Seite ist eine kurze Einleitung bzw. Motivation wieso wir eine solche Website erstellt haben und im zweiten Teil gibt es eine Schritt für Schritt Anleitung wie die Sounds zu kontrollieren sind.

Naturgeräusche:

Für die Naturgeräusche werden 2 Objekte je Sound im HTML-Teil angegeben. Es gibt zwei Slider. Objekte.

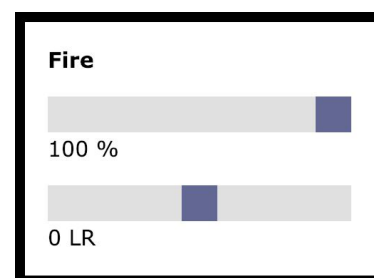
```
<input type="range" min="0" max="100" value="100" class="w3-slider" id="rainVS" step="10">  
<br>  
<span type="number" id="rainVO">100 %</span>
```

Der Slider startet mit dem Wert 100 und kann in zehnerschritten bis 0 herunter und bis 100 wieder hoch skaliert werden. Dieses wurde nur aus ästhetischen Gründen gewählt. Die Klasse w3-slider ist eine CSS Komponente die den Slider verschönert. Wie man unschwer erkennen kann ist der Slider für die Lautstärke des Sounds zuständig.

```
<input type="range" min="0" max="100" value="50" class="w3-slider" id="rainPS">  
<br>  
<span id="rainPO">0 LR</span>
```

Der zweite Slider hat den Startwert 50, was in dem Fall die Mitte des Sliders ist. In diesem Fall wird die Ausgabe der Sounds reguliert. Bei 50 wird der Sound gleichmäßig ins linke und rechte Ohr ausgegeben. Der Wert lässt sich von 50 runtersetzen auf 0, was dazu führt, dass der Sound mehr im Linken Ohr zu hören ist. Das Gleiche gilt für das rechte Ohr, nur mit den Werten 50 bis 100.

Am Ende sieht es so aus.



Um die Naturgeräusche zu hören bzw. abspielen zu lassen, muss man lediglich den Playbutton betätigen. Der Playbutton zeigt “Play” an wenn alles pausiert ist und “Pause” wenn die Sounds laufen.

```
<div class="w3-container w3-center w3-padding">  
  <button class="w3-button w3-white w3-margin-bottom w3-round" id="btnPLay">PLAY</button>  
</div>
```

Für die einzelnen Pianotasten wurden per HTML div-boxen implementiert.

```
<div class="w3-klavierKey w3-hover-opacity" id="key1">  
  <h3 class="w3-center w3-margin-bottom" id="keyId">A</h3>  
</div>
```

In der Mitte jeder einzelnen Taste wird die Note angezeigt. Jede Taste hat selbstverständlich eine eigene ID. Die einzelnen Tasten ändern ihre Farbe, wenn man mit der Maus drüber geht. Für den JavaScript teil wird die ID am wichtigsten. Bei mehreren ID's haben wir uns gedacht, von links nach rechts die Tasten hochzählen. Somit hat die Taste ganz links, die ID “key1” und einen rechts daneben die ID “key2”. Es geht so weiter, je nachdem wie viele Keys man nutzen möchte.

Das JavaScript ist folgendermaßen aufgebaut: am Kopf des Skriptes werden alle globalen Variablen deklariert. Einige von denen werden auch sofort Initialisiert. Es werden die Pfade der ganzen Sounds initialisiert.

Bewusst werden die AudioContext 'e in einer Funktion instanziiert und alle dazugehörigen Komponenten initialisiert.

```
function createAudioContext() {  
  if (!isCreated) {  
    windSound = new Audio("sound/wind.wav");  
    windSound.loop = true;  
    windSound.crossOrigin = "anonymous";  
    var windCon = new AudioContext();  
    windSource = windCon.createMediaElementSource(windSound);  
    windGain = windCon.createGain();  
    windStereoPanner = windCon.createStereoPanner();  
    windSource.connect(windGain);  
    windGain.connect(windStereoPanner);  
    windStereoPanner.connect(windCon.destination);  
  }  
}
```

Der AudioContext soll erst erstellt werden, wenn der Benutzer das Ausdrücklich möchte. Mit dem drücken des PlayButton will der Benutzer die gewünschten Sounds hören und somit kann die Erstellung der Kontexte beginnen.

Darauffolgend werden die Listener zugewiesen und die Logik wird implementiert.

```
//volume slider listener  
rainVS.addEventListener("input", function (e) {  
  var rainVolume = (this.value / 100);  
  rainVO.innerHTML = (rainVolume * 100) + " %";  
  rainGain.gain.value = rainVolume;  
});
```

Hier als Beispiel der Volume Slider des Regengeräusches. Wir haben den Wert für den Sound so gewählt, dass er von 0-100 geht. In unserem Fall wäre die 100 die volle Lautstärke. Aus Erfahrung können wir sagen, dass es die entspannteste Lautstärke ist. Außerdem kann es nicht passieren, dass die Geräusche zu laut sind. Es passt sehr gut zu unserem Geschäftsmodell. Die anderen Naturgeräusche

Die Ausgabe der Sound Ausrichtung wurde wie folgt implementiert:

```
//panning slider listener
rainPS.addEventListener("input", function (e) {
  var pValue = (this.value - 50) / 50;
  rainPO.innerHTML = pValue + " LR";
  rainStereoPanner.pan.value = pValue;
});
```

Hier gilt wieder, dass es sich bei der Grundeinstellung um die für uns angenehmste Lautstärke handelt. In diesem Fall ist der Ausgangswert bei 0. Die Werte sind änderbar von -1 bis 0, dass das für die linke Ausgabe gehört. Für die rechte Ausgabe geht der wert von 0 bis 1.

Für die Piano Töne wird ein simpler aufruf des Tons aufgerufen. Der ID wird einer Funktion übergeben in der ein Switch/Case Anweisung prüft, welche Taste auf der Tastatur gedrückt wird.

```
document.body.addEventListener("keydown", function (e) {
  var key_code = e.keyCode;
  keySoundPlay(key_code);
});
```

```
function keySoundPlay(key_code) {
  switch (key_code) {
    case a:
      key1Sound.play();
      break;
    case s:
      key2Sound.play();
      break;
  }
```