

Dal Suono allo Spazio

5EL - TPSIT

5 febbraio 2026

Prof. CAPOLUPO

Analizzeremo 3 aree chiave basate sui nostri esperimenti:

1 Audio Engineering (Web Audio API)

File: oscilloscopio.html, spettroscopio.html

2 Il Mondo 3D (Three.js)

File: monitor2.html, monitor3.html

3 Dati Reali (API NASA)

File: monitor1.html

1. Catturare la Voce

Come fa il browser a sentirci?

- **Il Ponte:** Usiamo `navigator.mediaDevices.getUserMedia`.
- **La Privacy:** Il browser chiede *sempre* il permesso.
- **Il Flusso:** I dati arrivano come **Stream** continuo.

Attenzione al Microfono

In `oscilloscopio.html` questo è il comando essenziale per rilasciare l'hardware (microfono / webcam / led) e liberare le risorse.

```
stream.getTracks().forEach(t => t.stop())
```

2. Tempo vs Frequenza

Differenze tra i due visualizzatori:

Oscilloscopio (Tempo)

- Metodo: `getBytesTimeDomainData`
- Cosa vedi: L'onda fisica.
- Disegno: `moveTo` → `lineTo`.

Spettroscopio (Frequenza)

- Metodo: `getBytesFrequencyData`
- Cosa vedi: Bassi, Medi, Alti.
- Disegno: Colori HSL (Arcobaleno).

Il Trucco della Scia (Ghosting)

In `spettroscopio.html`, usiamo un trucco per non far sparire subito le barre.

```
// Invece di cancellare tutto (clearRect)...  
// Disegniamo un velo nero semitrasparente (0.2)  
ctx.fillStyle = "rgba(0, 0, 0, 0.2)";  
ctx.fillRect(0, 0, width, height);
```

Questo crea un effetto di dissolvenza naturale ("scia"), molto più gradevole da vedere.

1. La Ricetta del 3D

Per creare un mondo 3D servono sempre 3 ingredienti:

- ❶ **SCENE (Scena):** L'universo vuoto.
- ❷ **CAMERA (Telecamera):** L'occhio che guarda (*Perspective*).
- ❸ **RENDERER (Motore):** Il pittore che disegna i pixel (WebGL).

2. Regex e Sicurezza

In `monitor3.html`, vogliamo vietare le lettere 'P' e 'R'.

```
// Evento 'input': controlla mentre scrivi
if (/[PR]/.test(val)) {
    // Regex: trova P o R e rimuovi (replace con vuoto)
    val = val.replace(/[PR]/g, '');
}
```

Le **Regex** (Espressioni Regolari) sono filtri potenti per validare i dati.

3. Tipi di Dati: Stringhe vs Numeri

Perché usiamo `parseFloat` nello slider della luce?

- HTML restituisce testo: "10.5"
- Three.js vuole numeri: 10.5

```
// monitor3.html
const x = parseFloat(sliderX.value); // Converte "10.5" -> 10.5
pointLight.position.x = x;
```

In `monitor1.html`, scarichiamo dati veri sugli asteroidi.

Concetti Chiave

- **Async/Await:** La NASA è "lontana". Mettiamo il codice in pausa (`await`) finché i dati non arrivano.
- **.map():** Prendiamo l'array complesso della NASA e ne creiamo uno nuovo, pulito, solo con i dati utili.

Come disponiamo gli asteroidi in orbita circolare?

```
// Trigonometria: seno e coseno  
mesh.position.x = Math.cos(angle) * distanza;  
mesh.position.y = Math.sin(angle) * distanza;
```

Senza `Math.cos` e `Math.sin`, non esisterebbero orbite nei videogiochi o nelle simulazioni.

HSL Hue, Saturation, Lightness. Ruota cromatica (0-360). Usata per l'arcobaleno audio.

Canvas La "tela" digitale HTML5 dove disegniamo pixel via codice.

Buffer Memoria temporanea (array) che contiene i dati audio grezzi.

Mesh Oggetto 3D = *Geometria* (forma) + *Materiale* (colore/riflessi).

API Il "cameriere" che porta la richiesta al server (NASA) e torna con i dati.

Buon divertimento