

Spherical Video Player

27436A Nava Stefano

2026

1 Architettura del sistema

1.1 initPlayerCommands.js

Gestisce la pulsantiera dei comandi ed i relativi eventi. Funzionalità:

- Start, pausa (bottone o barra spaziatrice) e stop
- Volume (si può mutare l'audio anche premendo "m")
- Fullscreen
- Zoom (con rotella del mouse).
- Rotazione (con trascinamento)
- Barra di progresso di audio e video. E' possibile visualizzare il tempo tenendo il mouse su un certo punto della barra e spostarvisi cliccandoci sopra.

1.2 SpatialAudioManager.js

Classe che gestisce l'audio. Componenti:

- **audioCtx** - AudioContext (Web Audio API) passato all'inizializzazione
- **videoElement** - Elemento video a 360°, passato all'inizializzazione
- Sorgenti audio. Ognuna è definita da **audioFilePath**, il percorso locale dove si trova la clip audio, e **options**, oggetto composto da:
 - **rotations**: composto da **starting** ed **ending**, che indicano rispettivamente l'angolo di partenza e di arrivo del suono in movimento. Se i due parametri corrispondono, il suono rimane fermo per tutta la sua durata.
La posizione viene gestita ruotando l'audio Ambisonics, impostando i parametri **yaw**, **pitch** e **roll**.
 - **volumes**: gestione analoga a **rotations**, con volumi rispettivamente di partenza e di arrivo.
 - **timing**: gestione analoga ai precedenti, con la specifica del secondo in cui l'audio inizia ad essere eseguito ed il secondo in cui l'audio termina.

Un esempio dell'uso dei parametri **starting** ed **ending** è la clip dei passi dell'uomo che si sposta: mentre si avvicina viene avviato l'audio, il volume si incrementa fino al punto massimo (in cui l'uomo è più vicino all'ascoltatore) per poi cominciare a diminuire fino a sparire, quando l'uomo è ormai lontano.

- Gain node **master** per gestire il volume generale. Ogni sorgente ha comunque il suo gain node per gestire i singoli volumi.

Il suono originale del video viene silenziato per poter sfruttare meglio le clip audio.

1.3 _main.js

File JS principale. Si occupa di inizializzare tutto l'ambiente (elemento video, camera, elementi audio). Istanzia un oggetto SpatialAudioManager, aggiunge le sorgenti audio presenti fin dall'inizio, gestisce l'orientamento del suono in base alla rotazione della camera e la sincronizzazione del video e delle clip audio.

1.4 index.html

Pagina principale, include:

- a-scene che contiene il video a 360 gradi
- pulsantiera dei comandi

1.5 player.css

Gestisce l'estetica del player e della pulsantiera dei comandi.

2 Risorse utilizzate

- Video a 360 gradi, formato mp4, rappresentato tramite a-scene.
- 5 clip audio Ambisonics B-format prelevate da <https://library.soundfield.com/>

3 Problemi affrontati

3.1 Sincronizzazione audio-video

Per mantenere la sincronizzazione è stato implementato un sistema di controllo periodico con `checkSynchronization` che controlla, per ogni sorgente, ogni 500 ms, se è ancora sincronizzata con il video. Se lo scarto supera i 150 ms, l'audio viene re-sincronizzato con il video.

3.2 Gestione della rotazione

Dato che A-Frame usa un sistema di rotazione con coordinate x, y e z ma gli audio Ambisonics usano un formato con matrice di rotazione, i valori vengono convertiti (`eulerToRotationMatrix`) per mantenere coerenza tra la posizione dell'ascoltatore nel video e quella dei suoni circostanti.

3.3 Audio in movimento

La gestione dell'audio dinamico (nel caso specifico dei passi) non è stata complessa ma ha richiesto l'introduzione del formato `starting/ending` e di una funzione che, durante l'esecuzione, calcoli i valori necessari e li applichi a volume e posizione. Ad esempio, se il volume iniziale è 0 ed il finale è 10, a metà dell'esecuzione della clip audio il volume dovrà essere 5.