

## Scienze e Tecnologie Multimediali

Laboratorio di Tecnologie Audio-Video Interattive 2017-18



## INTRODUZIONE

Il progetto “*Rhythm Game*” svolto per il corso di “*Laboratorio di Tecnologie Audio-Video Interattive*” è un’applicazione basata sul tempismo.

Lo scopo del gioco è quello di premere i cerchi che compaiono nello schermo quando entrano nell’area delimitata da un altro cerchio, che rimane fisso nella schermata.

Il gioco è strutturato su due livelli di diversa difficoltà:

- “**Cinema**”: contiene due aree obiettivo da premere e i cerchi che vengono generati periodicamente hanno una velocità bassa. Il punteggio minimo per considerare il livello superato è 20 cerchi colpiti correttamente.
- “**Web**”: contiene tre aree obiettivo da premere. Il punteggio minimo per considerare il livello superato è 40 cerchi colpiti correttamente.

Come tema grafico del gioco sono stati utilizzati i cerchi, integrando anche il tema universitario richiamandolo nel sottotitolo, nelle istruzioni e utilizzando il logo dell’Università (visibile nello scrolling dello sfondo). Tutti gli elementi grafici sono stati ideati e realizzati appositamente per questa applicazione dai componenti del gruppo. Le musiche presenti sono state scaricate da un sito specializzato ed editate per assecondare le esigenze dell’applicazione (a tutte le tracce audio dei livelli è stato aggiunto un effetto di fade-in e fade-out per renderle meno brusche).

- **Schermata principale**

Nella scena principale, il giocatore si trova di fronte al titolo del gioco. Può decidere se accedere al menù o uscire dalla applicazione.



- **Schermata menù**

Nella scena menù, il giocatore ha la possibilità di leggere le istruzioni del gioco, selezionare il livello che vuole giocare, oppure ritornare alla schermata principale.



- **Schermata livello 1**

In questa scena sono presenti l'indicatore del punteggio, il pulsante "Start" (con cui, una volta cliccato, si potrà iniziare il livello) e l'indicazione del punteggio minimo da raggiungere.

Una volta premuto "Start" i due cerchi bianchi diventeranno ascoltatori, e il giocatore potrà interagirci.



- **Schermata livello 2**

Questo livello ricalca le stesse funzionalità del primo.

A differenza del precedente, i cerchi sono situati sotto lo “Start” e sono tre al posto di due.



## CODICE

- **LIBRERIA COMPOSER:** la struttura generale del codice è stata creata utilizzando la libreria composer che deve essere richiamata ad ogni nuova scena e permette al giocatore di spostarsi da una zona all'altra del gioco.
- **CONFIGURAZIONE:** nel file del progetto “*config.lua*” è stato scelto di impostare il valore “*scale =*” a “*zoomEven*”. Questo cambiamento è motivato da un fattore estetico. In questo modo i cerchi creati dal centro dello schermo escono dal display senza passare dalle bande nere, che altrimenti sarebbero presenti.
- **LIBRERIA WIDGET:** questa libreria permette di creare un effetto grafico migliore sui tasti. Grazie ad essa il giocatore, quando preme un tasto, avrà la sensazione di averlo premuto veramente grazie ad un cambiamento della grafica. Quando si costruisce un widget si indicano due immagini. La prima è quella che si visualizzerà di default nello schermo, mentre la seconda è quella che si sostituirà alla prima quando premuta. Alle due immagini viene assegnata la stessa grandezza per essere sicuri di ottenere un effetto pulito. Il parametro “*onRelease =*” indica quale funzione verrà eseguita una volta rilasciato il click. I tasti widget vanno appositamente eliminati in fase di “*destroy*”, altrimenti, se ci si limita a disattivarli in fase “*hide*” questi rimarrebbero visibili anche nelle scene successive.
- **AUDIO:** le tracce audio presenti nel progetto sono tre, una per titolo e menù, una per il livello 1 e una terza per il livello 2. La musica delle scene titolo e menù è caricata sul canale 1 e vi rimarrà per tutta la durata dell'applicazione. Per mantenere un effetto di continuità nell'audio tra la scena titolo e la scena menù abbiamo inserito nel codice una funzione di controllo per far partire la musica da capo solo se questa non è già in fase di riproduzione. Questa scelta è stata fatta per evitare che, quando il giocatore passa da scene diverse a

quella del titolo, la musica non venga riprodotta più volte e sovrapposta a quella già in riproduzione. Nel canale 2 viene caricata in automatico la musica del livello che si sceglie in menù. Una volta usciti dal livello grazie alla funzione *"tornaMenu"*, che prevede al suo interno anche *"audio.stop()"*, la musica viene fermata, il canale liberato e quando il giocatore sceglierà un nuovo livello, la musica corrispondente si caricherà automaticamente nel canale 2. Nel passaggio dal menù al livello selezionato il volume della musica menù viene abbassato fino a raggiungere lo zero e una volta che il giocatore premerà start e avvierà la partita, la musica del livello sarà fatta partire. Una volta usciti dal livello il volume della musica menù viene fatto ritornare al valore di partenza.

- **FUNZIONE *"tornaMenu()"***: questa funzione, utilizzando il comando *"composer.gotoScene()"*, permette al giocatore di passare da una scena all'altra. Prima di effettuare questa operazione vengono rimosse tutte le scene che potrebbero essere state create in precedenza in modo da evitare che si creino sovrapposizioni. Quando questa funzione è all'interno dei livelli di gioco presenta delle aggiunte. Come prima cosa è bene specificare che la funzione è adatta ad essere utilizzata sia se il giocatore torna al menù mentre la partita è in corso, sia se torna al menù quando la partita è terminata. Questa funzione rialza il volume del canale 1, interrompe la musica del livello grazie al codice *"audio.isChannelPlaying(2)"* se questa è in riproduzione ed effettua un controllo sulla presenza o meno delle scritte di vittoria e sconfitta. Se queste, alla fine del gioco sono presenti le rende invisibili, mentre, se si torna al menù senza terminare la partita vengono rese prima visibili obbligatoriamente e poi la funzione ha gli elementi necessari per poter fare un controllo di tipo if (se si torna al menù prima di terminare la partita e viene applicato lo stop sulla musica si attiva in ogni caso la funzione *"esci"* che fa comparire a video le scritte di vittoria o sconfitta. Per questo motivo la funzione deve essere in grado di coprire anche questa opzione). L'ultima aggiunta specifica per i livelli è quella che permette di liberare la memoria dai cerchi creati e che non sono più utili ai fini del gioco. Ogni cerchio che si genera dal centro del display viene inserito nella tabella *"gruppoCC"* e ogni volta che da un livello torno al menù questa tabella viene svuotata. Il primo passaggio di questa operazione è stampare a console il numero di elementi presenti nella tabella, la seconda fase è costituita da un costrutto for che permette di eliminare ogni elemento della tabella e a fine ciclo stampo nuovamente il numero di elementi presenti per avere conferma dell'avvenuta cancellazione. Eseguite queste fasi la funzione *"tornaMenu"* fa passare il giocatore alla scena selezionata.
- **FUNZIONE *"scrolling()"***: è stata fatta la scelta stilistica di utilizzare uno scrolling per animare lo sfondo del titolo e del menù. Le due immagini da utilizzare vengono posizionate inizialmente una visibile a display e l'altra subito alla destra dello schermo (il posizionamento avviene nella fase *will di show*). Le due immagini vengono spostate dalla funzione scrolling di 1 pixel (il parametro *"speed"* è fissato a 1) ogni frame verso sinistra, così che parte della prima immagine esca dallo schermo sostituita da quella affiancata, che man mano ne prende il posto. Una volta che la prima immagine è uscita completamente dallo schermo viene riposizionata dove in origine c'era la seconda creando un senso di

circolarità. Questa funzione viene associata ad un ascoltatore Runtime ed eseguita ad ogni nuovo frame.

- **FUNZIONE “esci()”**: La funzione “*native.requestExit()*” permette di chiudere l’applicazione direttamente dal display premendo il tasto exit. Questa funzione è presente solo nella scena titolo.
- **FUNZIONI “giocaLivello1()”, “giocaLivello2()” e “tornaTitolo()”**: queste funzioni ci permettono di accedere alla scena prescelta in modo analogo alla funzione “*tornaMenu()*”. Alle prime due è stato scelto di aggiungere un valore opzionale (“*options*”) che permette di personalizzare la transizione tra le due scene con un effetto tecnicamente chiamato “*zoomInOutFade*”.
- **FUNZIONE “creaCC()”**: questa funzione ha il compito di generare i cerchi in maniera automatica e casuale, in modo che il giocatore non possa imparare a memoria l’ordine dei cerchi creati, ma si trovi davanti sempre a nuove combinazioni. Questa funzione può generare solo due tipi di cerchio, uno che va verso il cerchio obiettivo2 e uno che va verso il cerchio obiettivo1. La decisione di quale cerchio creare è affidata al “*math.random()*” che può generare come risultati i valori 1 e 2. Se viene generato 1 si crea un cerchio c1, se risulta 2 viene generato un cerchio c2. I due cerchi vengono da prima creati con un “*display.newCircle()*”, vengono colorati di bianco e li si imposta come completamente trasparenti. Per darli movimento viene utilizzata una transizione. Durante questa fase il cerchio passa progressivamente da trasparente a opaco, attraversa il cerchio obiettivo, raddoppia le sue dimensioni e raggiunge un punto al di fuori dello schermo. Queste impostazioni della transizione sono state scelte per motivi estetici e per evitare una sovrapposizione di cerchi una volta arrivati al loro cerchio obiettivo, in modo tale da rendere chiara la distinzione tra cerchi colpiti e cerchi non colpiti. Alla fine della creazione ogni cerchio viene inserito nella tabella “*gruppoCC*” per poi essere successivamente eliminato. Questa logica è stata applicata anche alla funzione da utilizzare per il livello 2.
- **FUNZIONI “controlloTap()”**: questa funzione permette di assegnare o meno un punto. Per fare questo calcola in valore assoluto la distanza tra il punto toccato dal giocatore e il cerchio in arrivo. Se questa distanza rispetta i parametri del controllo viene assegnato un punto (che viene aggiornato anche a schermo) e il cerchio in movimento diventa di colore nero, per comunicare al giocatore il successo della sua azione. Questa funzione è associata all’ascoltatore cerchio obiettivo in modo che, se il giocatore non preme nell’area delimitata, non possa avvenire il controllo sopra descritto.
- **FUNZIONE “fine()”**: la funzione “*fine()*” viene eseguita una volta che l’audio del livello ha terminato la sua riproduzione. Il compito di questa funzione è stampare a video la scritta “*hai vinto!*” o “*hai perso!*” in base all’effettivo punteggio raggiunto dal giocatore (il controllo avviene grazie ad un ciclo if). Per motivi estetici la funzione rimuove i cerchi obiettivo.

- **FUNZIONE “inizioGioco()”:** Una volta attivata questa funzione (associata all’ascoltatore “start”) il punteggio viene inizializzato a 0, viene fatto scomparire il bottone “start” e vengono creati dei cerchi trasparenti c1 e c2 (nel livello 2 anche c3) in modo tale da evitare un mal funzionamento della funzione “controllaTap” che rischierebbe di avere i valori legati a questi oggetti nulli ( nel caso l’utente preme un cerchio obiettivo quando il “math.random()” non ha ancora creato il corrispettivo cerchio ). Successivamente viene attivato il timer che attiva la funzione “creaCC()”, cerchio1 e cerchio2 diventano gli ascoltatori della funzione “controlloTap()” e viene fatta partire la musica del livello.