



FONDAZIONE
ANNA MARIA
CATALANO



SAPIENZA
UNIVERSITÀ DI ROMA

Progetto “DISEGNO + SMARTPHONE = MUSICA” - DSM
REPORT TRIMESTRALE N. 2 – 29 MAGGIO 2024

CALL FOR
IDEAs

INCLUSIONE • DIDATTICA • EDUCAZIONE • ARTE | sostenibili



FONDAZIONE
TIM

Sommario

Organizzazione e team di progetto	3
Gantt	3
Approvvigionamenti e affidamenti	3
Borse di studio e incarichi professionali	3
Stato di avanzamento delle attività	4
Spese sostenute	4
Attività tecniche.....	4
T1.1 – Sviluppo del software (backend)	4
T1.2 – Sviluppo del software (frontend).....	6
T1.3 - Acquisto e testing dei tools	6
Attività di comunicazione del progetto.....	7

Organizzazione e team di progetto

La composizione del team di progetto indicata nel verbale del 7/12/2023 dello Steering Committee è invariata.

Gantt

Il GANTT allegato all'Accordo sottoscritto in data 6 dicembre 2023 risulta confermato.

Approvvigionamenti e affidamenti

Rispetto al Report n. 1 non sono stati effettuati ulteriori approvvigionamenti e affidamenti.

Borse di studio e incarichi professionali

Rispetto al Report n. 1 non sono stati assegnati ulteriori incarichi professionali e borse di studio.

Stato di avanzamento delle attività

Spese sostenute

La Fondazione, a fronte di un totale di commissionamenti e assegnazioni di € 178.921,56, ha liquidato alla data del 31/5/2024, mediante bonifici bancari, Euro 53.111,56.

Il Dipartimento di Informatica ha consuntivato, alla stessa data, spese per il personale dedicato al progetto pari a € 65.850,00.

Lo stato di avanzamento in termini economici è pari al 31,72%.

Attività tecniche

Di seguito vengono riportate le attività tecniche svolte secondo il GANTT durante il periodo indicato dalla finestra temporale che coinvolge questo report.

T1.1 – Sviluppo del software (backend)

Lo sviluppo del backend si è concentrato sui seguenti aspetti:

- Keyboard detector
- MediaPipeUnityPlugin
- Distanza mani e foglio dal telefono

Keyboard detector

Keyboard detector è il nome che abbiamo dato al modulo che, preso un frame dalla webcam del telefono, trova il pianoforte disegnato su carta.

I miglioramenti apportati al nuovo detector sono molteplici e riguardano gli aspetti di facilità d'uso, velocità e accuratezza. Il nuovo detector è facile da usare perché sfrutta al meglio l'interfaccia interattiva dello smartphone per semplificare l'inquadratura del foglio all'interno di un'area designata, e questo serve a ridurre il numero di passaggi da eseguire, che passano da 2 a 1. Il detector non ha più bisogno di effettuare la fase di background subtraction (la fase in cui il foglio viene isolato dal resto dell'immagine) e questo riduce notevolmente anche il tempo richiesto all'elaborazione. Per la seconda fase, quella di rilevamento dei tasti, l'algoritmo è stato rivisto nella sua interezza ed è stato reso molto più snello. Oltre ad una fase di preprocessing (che serve ad aumentare il contrasto tra il nero dei tasti disegnati e il bianco del foglio ed è di poco diversa rispetto alla sua versione precedente) la nuova procedura di rilevamento dei tasti utilizza un solo algoritmo di computer vision rispetto ai 3 usati nella versione precedente, riducendo drasticamente il tempo di computazione e allo stesso tempo aumentandone notevolmente la precisione. In questo nuovo algoritmo i tasti vengono rilevati per intero in base alla loro area e al loro contorno visti come un tutt'uno, non in base a righe orizzontali e verticali che possono facilmente essere mal interpretate. Questo permette

la creazione dell'immagine di una tastiera virtuale che viene mostrata in sovrapposizione al video dello smartphone, grazie alla quale l'utente può valutare direttamente il risultato ottenuto e, se insoddisfatto, ripetere l'operazione.

MediaPipeUnityPlugin

MediaPipe è una libreria sviluppata da Google che sfrutta il machine learning per, tra le altre cose, eseguire il tracciamento delle mani in tempo reale. La libreria può essere usata in modo nativo con Python, o su Android e iOS grazie alle API di Google. Tuttavia da parte di Google non c'è un supporto diretto per C#, il linguaggio che stiamo usando noi su Unity.

La soluzione arriva da un repository su github chiamata MediaPipeUnityPlugin (MUP). È un plugin per Unity che implementa MediaPipe e permette di usarlo nei progetti.

In una prima fase sono stati svolti dei test per vedere il funzionamento e le prestazioni di MUP. Gli interessi erano molteplici: verificare se fosse utilizzabile sui vari telefoni, e controllare le prestazioni sui telefoni di varie fasce di prezzo (e quindi con potenze di calcolo differenti, dai più ai meno potenti). In seguito è iniziato lo sviluppo del software per il nostro programma, che sfrutta MUP. In altre parole, il modulo che si occupa di tracciare le mani. Questa fase ha richiesto diverso tempo, per due motivi in particolare. Il primo è che MUP, come spesso accade con le librerie software, non è ben documentato, e quindi capire come sviluppare anche la più semplice delle operazioni richiede impegno e tempo. Il secondo è che MUP si interfaccia con MediaPipe a basso livello, e questo significa che per sfruttare a pieno MUP bisogna sapere bene anche come funziona MediaPipe.

Distanza mani e foglio dal telefono

Informazioni necessarie per capire quando un utente vuole suonare una nota sono la distanza della mano e del foglio di carta su cui è disegnato il pianoforte dal telefono.

MediaPipe stima la distanza della mano dalla fotocamera. Questo dato però, come appena citato, viene stimato e non è molto preciso; in particolare il valore risulta estremamente instabile (flickering). Per ovviare a questo problema sono stati testati due metodi di interpolazione dei dati, per far sì che il valore nel tempo si stabilizzi.

I due metodi di interpolazione sono Simple Moving Average (SMA) ed Exponential Moving Average (EMA). Entrambi i metodi tengono conto dello storico di un dato (in particolare gli ultimi n valori) e lo interpolano facendo una media. La differenza tra SMA e EMA è che SMA dà lo stesso peso a tutti i valori nello storico, mentre EMA dà esponenzialmente meno peso man mano che un valore diventa più vecchio, dando quindi più importanza ai valori più recenti. Entrambi i metodi di interpolazione sono sembrati validi, e per testarli si è quindi impostato il seguente test: viene presa la profondità stimata da MediaPipe, poi interpolata in maniera indipendente sia con SMA che con EMA e i tre valori sono stati salvati per essere poi disegnati su un grafico, in modo da mostrare visivamente con SMA e EMA come cambiano i valori rispetto a MediaPipe senza interpolazione.

Si è visto inoltre, sperimentalmente, che SMA e EMA non bastano per rendere affidabile il dato di profondità stimato da MediaPipe, per questo è stato avviato uno studio sull'irrobustimento di questo dato, ed eventualmente combinandolo con altri dati per mappare meglio la scena.

Progressi

Il keyboard detector è stato migliorato sia in precisione che in velocità. È stato aggiunto un feedback visivo immediato per l'utente.

Con MediaPipeUnityPlugin siamo in grado di tracciare efficacemente le mani nella scena.

È iniziato lo studio di una parte critica del software, la stima della profondità della scena.

Lavoro da svolgere

La parte di keyboard detection è conclusa.

È necessario migliorare la stima della distanza mano e foglio dal telefono. I grafici prodotti per osservare l'interpolazione del dato stimato da Mediapipe sono serviti in una prima fase di studio, ma quando questa parte del programma sarà più completa occorrerà fare test empirici per vedere se, nell'utilizzo concreto dell'applicazione, l'interpolazione sarà effettiva.

T1.2 – Sviluppo del software (frontend)

Insieme al backend, che gestisce il funzionamento del programma, è necessario anche un frontend, ovvero l'interfaccia grafica (UI) con cui l'utente interagisce. Due sono gli aspetti importanti su cui ci si sta focalizzando: la UI deve essere intuitiva e di facile utilizzo e deve essere bella da vedere.

Lavoro svolto

Si è iniziato a progettare la UI, prima con diverse bozze su Photoshop, per avere un'idea di massima, e poi con alcune prove su Unity.

Una difficoltà in questo tipo di applicazione è determinata dall'orientamento del telefono e da schermi diversi; il problema non risiede nella risoluzione degli schermi, ma il rapporto d'aspetto (base:altezza). Infatti, esistono una pleora di rapporti d'aspetto tra i telefoni, e quindi dopo aver fatto alcune prove in 16:9, si è iniziato a riadattare la UI in modo tale che automaticamente si adatti allo schermo di ogni telefono (e.g. 18:9, 19:9, 19.5:9).

Si è inoltre deciso di seguire le linee guida di Material Design di Google, design pensato per creare applicazioni con interfacce grafiche pulite e funzionali.

Progressi

È iniziato lo sviluppo dell'interfaccia grafica e sono state prese le decisioni che ne guideranno la direzione.

Lavoro da svolgere

Bisogna completare la UI.

T1.3 - Acquisto e testing dei tools

Non sono stati effettuati ulteriori acquisti.

Attività di comunicazione del progetto

Il sito dedicato alla comunicazione <https://www.musikeyrtual.it> è stato attivato in data 16/3/2024 e sono state caricate le video interviste a:

- Marini Marco Raoul Principal Investigator del progetto
- Tronci Enrico Direttore del Dipartimento di Informatica
- Princic Andrea Programmatore SW Borsista della Fondazione A.M. Catalano.

E. Tronci e M.R. Marini sono intervenuti Venerdì 17 maggio nel Panel "**Ricerca musicale, innovazione e programmazione radiofonica**" del Festival delle Radio Universitarie 16 - 18 MAGGIO 2024 - SAPIENZA UNIVERSITÀ DI ROMA".

[https://www.dropbox.com/scl/fi/b1qf1go5obrtlrc5hvvhs/FRU-2024 Tronci-e-Marini.mp4?rlkey=3u2wtjkwac140y8h0zdxyo54&dl=0](https://www.dropbox.com/scl/fi/b1qf1go5obrtlrc5hvvhs/FRU-2024%20Tronci-e-Marini.mp4?rlkey=3u2wtjkwac140y8h0zdxyo54&dl=0)