FONDAZIONE ANNA MARIA CATALANO

SAPIENZA UNIVERSITÀ DI ROMA

Progetto “DISEGNO + SMARTPHONE = MUSICA” - DSM

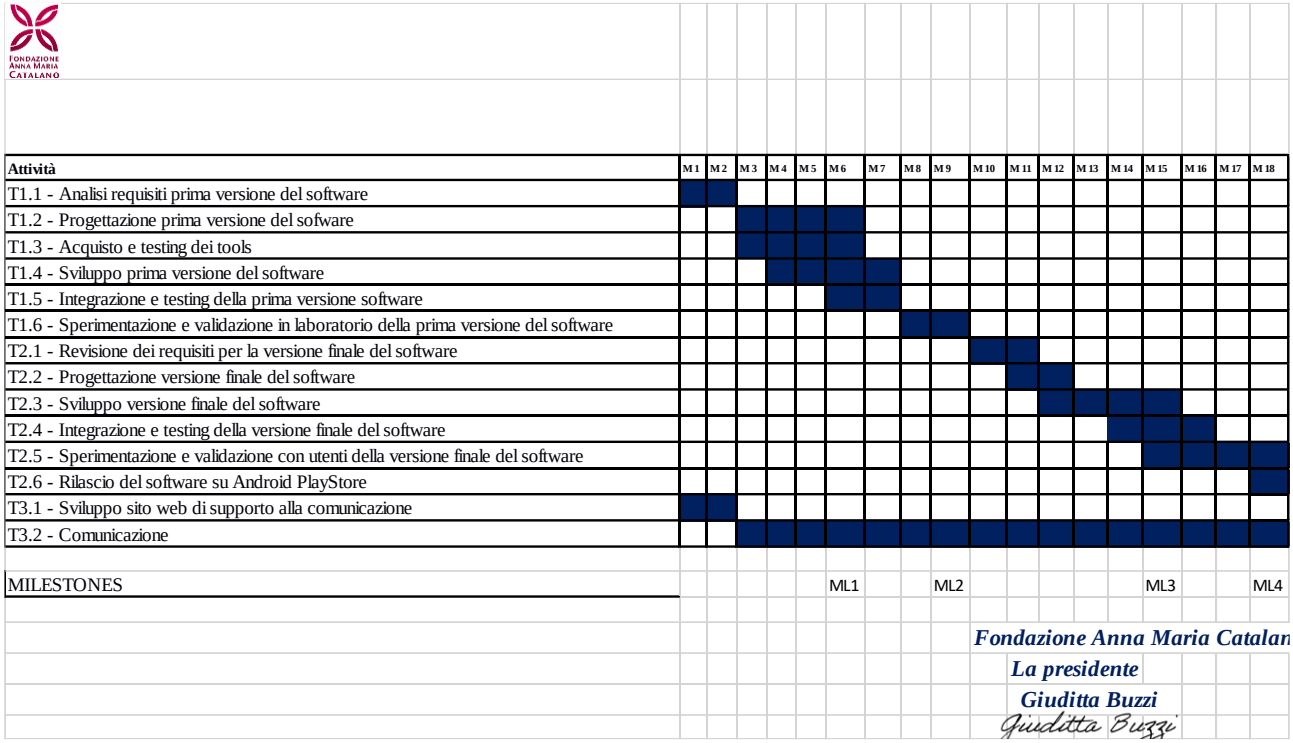
STATO DI AVANZAMENTO AL 5 MARZO 2024

Sommario

Data inizio progetto

Il progetto ha avuto inizio il giorno 11 dicembre 2023.

Gantt



Stato di avanzamento delle attività

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 - Analisi requisiti prima versione del software

Operazione svolta secondo il cronoprogramma. Sono state raccolte le funzionalità base secondo i seguenti use-case:

- Mostra tutorial
- Calibrazione del sistema
- Avvio sistema di riproduzione musicale
- Opzioni per la gestione del suono (volume ed eventualmente tipologia di suono)

T1.2 - Progettazione prima versione del software

Operazione svolta quasi nella sua interezza. È stata individuata la libreria per sfruttare gli algoritmi di Computer Vision, ma sono in corso i test per verificare l'efficacia della libreria per l'estrazione dello skeleton della mano.

Lavoro svolto
Coordinamento del gruppo di lavoro. È stato realizzato in collaborazione con la Fondazione Anna Maria Catalano il video ed il materiale per la trasmissione Geo&Geo. Sono state raccolte le specifiche dei requisiti per iniziare il lavoro di porting del codice dalla versione tramite camera RGB-D a quella per smartphone. Sono stati individuati i software necessari per iniziare lo sviluppo Android-centered ed eventualmente iOS tramite cross-platform build. Sono state suggerite le strategie operative legate

agli algoritmi da utilizzare ai due borsisti che attualmente lavorano tramite la Fondazione sul progetto.

Progressi
Sul lato gestionale, tutte le attività sono state svolte secondo cronoprogramma. L'unica attività che è fortemente legata a quella dello sviluppo è la progettazione del software, poiché si richiedono test per avere una risposta definitiva sulle strategie operative.

Lavoro da svolgere
Attualmente non sono presenti attività pendenti sul lato del coordinamento. Sono invece aperte alcune problematiche legate allo sviluppo, ma sono in fase di risoluzione.

T1.3 - Acquisto e testing dei tools

Acquisti effettuati secondo cronoprogramma. Sono stati acquisiti smartphone di varie tipologie e fasce di prezzo, sistemi per reggere i suddetti, macchine potenti per poter sviluppare il codice e telecamere di profondità di ultima generazione per effettuare correttamente il porting dalla versione Desktop alla versione Mobile.

T1.4 - Sviluppo prima versione del software

Attività in corso, secondo cronoprogramma.

Lavoro svolto

Sviluppo dell'algoritmo di "keyboard detection", ovvero un modulo del programma che permette di individuare la tastiera (pianoforte) una volta inquadrata con il telefono. Come base è stato usato un algoritmo che avevamo già sviluppato e pubblicato su un articolo, ed è stato migliorato. Durante i test il gruppo di lavoro ha risolto il seguente problema: quando si cercano le linee del pianoforte molte di queste vengono trovate duplicate (i.e. una singola linea viene trovata come tante linee vicine) o peggio ancora spezzate. A partire da queste linee duplicate e/o spezzate, bisogna ricostruire una singola linea che rappresenti una linea del pianoforte. L'algoritmo trova delle semirette con HoughLinesP (i.e. Hough Lines probabilistico), che vengono date con notazione punto-punto (i.e. (x1, y1, x2, y2)), e le converte in rette con la notazione (ρ, θ). Viene presa ogni coppia di rette e si valuta se sono sufficientemente simili, usando un threshold per valutare distanza e rotazione delle rette. Le semirette simili tra loro vengono messe in un unico cluster, si calcola la media dei punti per creare una nuova retta unica. Il problema delle righe viene attenuato facendo pre-processing sull'immagine: in particolare riducendone le dimensioni e applicando blur.

L'applicazione viene sviluppata usando il motore di gioco Unity, con il linguaggio di programmazione C#. Esistono due plugin per Unity che permettono di usare all'interno dei progetti OpenCV, una delle più importanti librerie con algoritmi di computer vision. I due plugin presi in considerazione sono OpenCVForUnity e Emgu. Il lavoro si è concentrato sul determinare quale fra i due plugin sia più adatto al nostro lavoro. La scelta sarebbe stata tra uno dei due, o addirittura entrambi, per ottenere il meglio dei due mondi. Sulla carta Emgu sembrava essere il migliore, essendo un wrapper di OpenCV

dovrebbe garantire prestazioni migliori di OpenCVForUnity, che invece riscrive le librerie di OpenCV (e quindi interpretato, non compilato). Sono stati definiti dei test che, presa in input un'immagine, le applicavano semplici algoritmi usati in computer vision; i.e. Canny, HoughLinesP e Kmeans. I test, in primis eseguiti su computer, hanno sin da subito rivelato una completa superiorita di OpenCVForUnity su Emgu. Non solo è stato molto più veloce, spesso con uno scarto di diversi ordini di grandezza, ma è stato anche più semplice da usare, avendo dato meno errori e avendo dato meno problemi circa l'inizializzazione delle strutture dati e in generale dei parametri passati in input alla funzione. L'idea quindi di usare entrambi i plugin è andata sfumando per diversi motivi, il primo di cui sopra. Il secondo motivo è che Emgu è un plugin molto pesante, che avrebbe reso l'applicazione a sua volta pesante (oltre 2 1GB). In ultimo, il codice sarebbe stato molto più complicato, dovendo tenere traccia delle strutture dati che sono le stesse per entrambi i plugin, ma implementate diversamente; quindi le chiamate a OpenCVForUnity avrebbero richiesto una certa struttura dati implementata dentro OpenCVForUnity, mentre Emgu avrebbero richiesto la stessa struttura dati, ma implementata dentro Emgu, non compatibili, e quindi con eventuali conversioni. Già convinti a questo punto che OpenCVForUnity sarebbe stato l'unico plugin in uso, sono arrivati i test su telefono. Il codice, così come è stato testato su computer, è stato portato su un *apk* e installato su un telefono Android. Il test con OpenCVForUnity e stato eseguito senza problemi, mentre su Emgu si e fermato, con un errore che recitava "image not found": Emgu era convinto che il file contenente l'immagine non esistesse, file che pero OpenCVForUnity era aveva trovato, e quindi sicuramente esistente. Quest'ultimo test ci ha definitivamente convinti che OpenCVForUnity sarebbe stata la scelta migliore.

Progressi

L’algoritmo che trova la tastiera è stato scritto, testato, e quasi pronto. Sebbene l’algoritmo non sia ancora perfetto, è già utilizzabile per eseguire eventuali test. Inoltre, è stato scelto il plugin per usare gli algoritmi di computer vision nel progetto.

Lavoro da svolgere

Il threshold per valutare se due rette vadano nello stesso cluster o meno può essere raffinato per discriminare con più precisione. Una volta clusterizzate le righe, si può usare un algoritmo migliore per unirle in un’unica riga. L’algoritmo dovrà in seguito essere implementato nel progetto del programma. Per fare questo andrà riscritto usando il linguaggio e i framework usati per il progetto; in altre parole l’algoritmo scritto in Python andrà riscritto in C# usando il plugin per Unity OpenCVForUnity.
Proseguono i test per la verifica del miglior sistema di cattura dello skeleton della mano.