0121

Documentazione

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 5

2.4 Pianificazione 1

2.5 Analisi dei mezzi 1

2.5.1 Software 1

2.5.2 Hardware 1

3 Progettazione 2

3.1 Design dell’architettura del sistema 2

3.2 Design dei dati e database 4

3.3 Design delle interfacce 5

3.4 Design procedurale 5

4 Implementazione 6

4.1 Menu 6

4.1.1 Interfaccia 6

4.1.2 Creazione file 6

4.1.3 Caricamento immagini 6

5 Test 8

5.1 Protocollo di test 8

5.2 Risultati test 9

5.3 Mancanze/limitazioni conosciute 9

6 Consuntivo 9

7 Conclusioni 9

7.1 Sviluppi futuri 9

7.2 Considerazioni personali 9

8 Bibliografia 9

8.1 Sitografia 9

9 Allegati 9

# Introduzione

## Informazioni sul progetto

|  |  |
| --- | --- |
| Allievi coinvolti nel progetto : | Karim Galliciotti, Zeno Darani, Stefano Mureddu e Sara Bressan |
| Docente responsabile : | Luca Muggiasca |
| Classe : | 3 AA-AC della sezione di Informatica della Scuola Arti e Mestieri di Trevano |
| Data di inizio del progetto : | 2021-01-14 |
| Data di fine del progetto : | Indefinita |

## Abstract

E’ una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* Background/Situazione iniziale
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.

## Scopo

Lo scopo di questo progetto consiste nello sviluppo di una tavolozza virtuale sulla quale sarà possibile disegnare a mano libera utilizzando un Leap Motion. Nella tavolozza sarà quindi possibile selezionare gli strumenti di disegno (ad esempio pennelli, matite ...) e selezionare il colora da utilizzare nel disegno. I movimenti delle mani corrisponderanno a determinate azioni sulla tavolozza (es. tracciare una riga), inoltre i disegni creati saranno salvabili ed esportabili su file separati.

Lo scopo di progetto consiste inoltre nell'imparare a gestire dei progetti in team e imparare ad utilizzare nuove risorse.

# Analisi

## Analisi del dominio

È stato richiesto lo sviluppo di un'applicazione per la creazione di contenuti grafici, attraverso l'utilizzo di un sensore Leap Motion. L'applicazione dovrà permettere dunque l'interazione dell'utente con essa tramite il solo movimento delle mani. Il programma dovrà essere simile a Paint e semplice da utilizzare, inoltre dovrà dare la possibilità di salvare ed esportare le opere artistiche create. L'applicazione sarà stand alone e non necessiterà particolari conoscenze da parte degli utenti.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| ID: REQ-001 | |
| Nome | Tracciamento delle mani |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Utilizzo del sensore Leap Motion |

|  |  |
| --- | --- |
| ID: REQ-002 | |
| Nome | Possibilità di selezione di diversi strumenti |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| ID: REQ-003 | |
| Nome | Possibilità di selezione il colore con cui disegnare |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| ID: REQ-004 | |
| Nome | È possibile disegnare a mano libera |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| ID: REQ-005 | |
| Nome | I disegni si possono salvare |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| ID: REQ-006 | |
| Nome | Applicativo stand alone |
| Priorità | 1 |
| Versione | 1.0 |
| Note |  |

|  |  |
| --- | --- |
| ID: REQ-007 | |
| Nome | Possibilità di importare immagini da modificare |
| Priorità | 2 |
| Versione | 1.0 |
| Note |  |

## Use case



## Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

|  |
| --- |
| Macintosh HD:Users:Guest:Downloads:VirtualPainter-main 2:Documentazione:Immagini:VirtualPainter.png  Figura 1: Esempio di diagramma di Gantt. |

## Analisi dei mezzi

### Software

* Google Chrome
* Opera
* MacDown
* Visual Studio Versione 2019
* Unity Versione 2019.4.1
* Libreria SDK di Leap Motion
* Git Hub
* Trello

### Hardware

* Mac A422-05 scolastico
* Mac A422-06 scolastico
* Mac A422-07 scolastico
* Mac A422-08 scolastico
* PC personale di Karim (HP Omen)
* PC personale di Sara (Acer Aspire)
* Leap Motion Controller fornito dal docente

# Progettazione

## Design dell’architettura del sistema



La applicazione si avvia e per prima cosa controlla che il Leap Motion sia collegato.

Se è collegato entra nel menu altrimenti da un errore e chiede di riavviare il programma.



L’applicazione entra nel menu e chiede se si vuole creare un nuovo file o caricarne uno.

Se si crea un nuovo file chiede di definirne le dimensioni e il tipo e poi entra nella pagina di disegno.

Se si carica un file controlla se è in regola e poi entra nella pagina di disegno.



L’applicazione apre le impostazioni e permette di selezionare con l’indice della mano dominante una delle seguenti cose, selezione strumento, selezione colore, impostazioni generali, salvataggio, chiusura.

Se viene selezionato salvataggio controlla se è il primo salvataggio del file.

In caso sia il primo apre l’explorer per definire dove salvarlo, altrimenti sovrascrive il file.



L’applicazione apre l’area di disegno e controlla se la mano debole ha il palmo girato verso l’alto.

Se sì apre le impostazioni, altrimenti permette alla mano di fare varie azioni.

Nello stesso momento controlla se la mano dominante è completamente aperta.

Se si permette a questa di fare varie azioni, altrimenti controlla la posizione delle dita.

Se queste sono chiuse a pugno si potrà cancellare, altrimenti se l’indice e il pollice sono collegati si potrà disegnare.



L’applicazione si chiude.

Prima di chiudersi controlla se c’è stata una nuova modifica dopo l’ultimo salvataggio.

Se sì apre le impostazioni di salvataggio, e una volta selezionate chiude il programma, altrimenti il programma viene chiuso direttamente.

## Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l’ultima versione, mentre le vecchie saranno sui diari.

## Macintosh HD:Users:Guest:Downloads:interfacce.pngDesign delle interfacce

La nostra idea sulle interfacce dell’applicazione consiste nel avere un menu iniziale per caricare o creare un nuovo foglio, e successivamente una tela su cui disegnare.

Il menu iniziale, il primo in alto a sinistra nell’immagine, consiste in soli due grandi bottoni. Uno per creare un nuovo foglio bianco, e uno per caricare un immagine esistente. In caso si clicchi su crea nuovo foglio la schermata del menu cambierà in quella in basso a sinistra, in cui potranno venir inseriti grandezza e altezza desiderati, e continua per creare un nuovo foglio.

Una volta caricato o creato il foglio su cui lavorare ci troveremo sulla tela dove potremmo disegnare. La tela si trova in alto a destra nell’immagine. Sulla tela potremmo vedere il foglio di lavoro e le nostre mani e, sulla mano forte, in questo caso rappresentata da quella di destra, potremmo vedere il colore e lo strumento selezionato.

Il menu, nell’immagine in basso a destra, partendo da sopra verso il basso, permetterà di selezionare lo strumento, il colore, la mano forte, salvare, salvare con nome e uscire per tornare al desktop. Il menu comparirà automaticamente sulla mano debole quando questa è rivolta con il palmo verso l’alto.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# Implementazione

L’implementazione parte con la creazione del progetto su Unity con le prime impostazioni di base che poi verrà esportato su gli altri pc. Ogni volta che qualcuno completerà qualcosa sul proprio pc questa verrà esportata e implementata negli altri.

## Menu

### Interfaccia

Il menu implementato segue fedelmente il Design progettato e comprende i due bottoni con il titolo.

Inizialmente l’unico bottone funzionante è quello per spostarsi sulla schermata per selezionare la grandezza del nuovo foglio visto che nessuno script oltre a quello per spostarsi tra questi due menu è stato implementato.

Il menu di creazione del file segue anch’esso fedelmente la progettazione con i due form di testo e il pulsante continua.

Entrambe le interfacce sono state create unicamente utilizzando l’editor di unity.

Lo script per spostarsi non fa altro che nascondere il primo menu e mostrare il secondo al click del bottone tramite dei boolean settati a false o true.

### Creazione file

Per la creazione file si utilizza uno script richiamato cliccando il bottone continua del menu di creazione.

Lo script al momento del click prende i dati inseriti dai form e li converte in int, questi vengono poi forniti come parametri ai metodi per modificare la grandezza della tela e per creare una nuova texture da applicare ad essa.

Al momento del click il programma apre automaticamente l’explorer e fa salvare all’utente il foglio appena creato con già un nome e una posozione. Successivamente crea una texture di dimensioni volute e la assegna all’oggetto.

Una volta fatto questo trasforma i dati che sono stati inseriti nei form come pixel e li trasforma in unità di unity dividendoli per 100 così da poter modificare la grandezza dell’oggetto in modo corretto e scalato.

### Caricamento immagini

Per il caricamento di immagini

DA FINIRE

### Caricamento tela disegnabile

Il caricamento della tela disegnabile avviene come anticipato prima tramite una scalatura di pixel. In partenza la nostra tela non è altro che un GameObject a forma di rettangolo bianco senza nulla di particolare. Una volta che i dati per la grandezza vengono inseriti nei form sulla tela verrà applicata una texture delle dimensioni volute. Per far si che la tela e la texture applicata ad essa siano proporzionate bisogna ridimensionare la tela. Per farlo si rende la tela alta e larga quanto i px inseriti divisi per 100. Inoltre per fare in modo che la tela occupi tutto lo schermo, in orizzonate o in verticale, bisogna scalarla nuovamente ingrandendola o diminuendola mantenendo però sempre le proporzioni.

## Leap Motion

### Prova Leap Motion Controller

Per testare il funzionamento del Leap Motion lo abbiamo attaccato al portatile e abbiamo avviato il programma di test ufficiale scaricato insieme ad i pacchetti di base necessari a questo per funzionare.

I test sono stati un successo e Leap Motion vedeva correttamente le mani.

### Tracciamento mani

Per il tracciamento delle mani inizialmente bisogna scaricare gli Unity module packages[[1]](#footnote-1), ovvero i pacchetti per gli sviluppatori utilizzanti un Leap Motion. Una volta scaricati e importati si procede ad installare il plugin “XR plugin Management[[2]](#footnote-2)”, nel nostro caso la versione 3.2.16, e a selezionarlo sotto la scheda “Project Settings”.

Prima di creare il programma da zero si è provato a testare il funzionamento delle configurazioni settate, usando la scena di test già fornita dal Leap Motion “Capsule Hands (Desktop)[[3]](#footnote-3)”. In questo scenario abbiamo verificato che il Leap Motion interagisse con le mani presenti nella scena.

Una volta confermato il successo dei test siamo passati a ricreare la scena da soli.

Per prima cosa si trascina il prefab LeapHandController[[4]](#footnote-4) nella scena e si crea un nuovo GameObject di nome “HandsModel”. Questo dovrà contenere “RigidRoundHand\_L[[5]](#footnote-5)”, “RigidRoundHand\_R”, “Capsule Hand Left” [[6]](#footnote-6)e Capsule Hand Right”.

HandsModel dovrà poi utilizzare lo script “Hand Model Manager” e dopo aver modificato il campo Size di Models Pool da 0 a 2 dovranno comparire 2 nuovi gruppi di campi. Questi dovranno venir chiamati “Graphics\_Hands” e “Physics\_Hands”. Sotto “Graphics\_Hands, nei campi Left Model e Right Model dovranno venir selezionati “RigidRoundHand\_L” e “RigidRoundHand\_R”.

In entrambi i gruppi andranno inoltre spuntate le caselle “Is enable” e “Can duplicate”.

### Lettura movimenti

La lettura dei movimenti avviene tramite metodi già presenti nel package del Leap Motion, che riconoscono se la mano è in una determinata posizione. Questo può esserci molto utile, visto che non dovremo fare altro che inserire il metodo per controllare la posizione voluta nella condizione dell’if per gestire, quando fare, o non fare, una determinata azione.

## Disegno

### Azione Disegno

L’azione di disegnare è una se non la parte più importante del progetto. Per farlo utilizza i RayCast, ovvero dei vettori orizzontali che partono dalla mano e arrivano fino alla tela. Nel momento in cui la mano soddisfa la condizione isPinching(), ovvero ci sono una o più dita il punto toccato sulla tela viene colorato del colore selezionato. Il metodo si chiama DrawCircle e per disegnare il puntino si entra in un ciclo definito dalla grandezza di esso che cambia tutti i pixel al suo interno.

### Selezione colore

#### Tavolozza dei colori

La tavolozza dei colori è la componente che permette di definire i colori da usare per dipingere. È infatti possibile scegliere le singole componenti RGBA dei colori, e avere fino a un massimo di 12 colori.

#### Creazione della GUI

In un primo momento è stata pensata ad una GUI 2D strutturata con 3 slider in una parte superiore e una serie di pallini rappresentanti i colori nella parte inferiore.



Poi, in seguito, si è optato per una versione 3D così da poter implementare facilmente l’interattività con il Leap Motion. Infatti la versione 3D è composta da componenti presi dai progetti Unity d’esempio di utilizzo del Leap Motion. Gli slider hanno la componente InteractionSlider mentre i button hanno la InteractionButton. La struttura è stata modificata poiché ritenuta più comoda da usare con il Leap Motion. Adesso è divisa in due parti verticali dove a sinistra ci sono gli slider uno sotto all’altro e a destra i button sono disposti in una matrice 3 per 4.



#### La classe ColorPicker

La classe ColorPicker serve a gestire le funzioni della paletta di colori ed è un suo componente.

Ogni volta che un colore (parte destra) viene selezionato, cliccandoci sopra con un dito, quello diventa il colore che verrà usato per disegnare. Il colore selezionato è messo in evidenza poiché leggermente più grande degli altri. Quando un colore selezionato gli slider prendono i valori delle componenti RGBA[[7]](#footnote-7) di quel colore. Questa serie di istruzioni vengono eseguite dal metodo OnColorSelected() che viene aggiunta come azione ai button.

button.GetComponent<InteractionButton>().OnPress = new Action(OnColorSelected);

Ad ogni cambiamento di valore degli slider viene aggiornato sia il valore numerico rappresentato a destra che il valore effettivo del colore. Per fare ciò ci sono i tre metodi che eseguono questi aggiornamenti di valori: OnChangedRed(), OnChangedGreen(), OnChangedBlue(), OnChangedAlpha(). Anche in questo caso i metodi vengono aggiunti come azione da eseguire con la modifica del valore degli slider.

SliderRed.HorizontalSlideEvent = new Action<float>(OnChangedRed);

SliderGreen.HorizontalSlideEvent = new Action<float>(OnChangedGreen);

SliderBlue.HorizontalSlideEvent = new Action<float>(OnChangedBlue);

SliderAlpha.HorizontalSlideEvent = new Action<float>(OnChangedAlpha);

### Selezione strumento e dimensione

## Azioni varie

### Salvataggio

Il salvataggio rapido avviene tramite il metodo salva() richiamato al momento del clic del bottone. Essendo che al momento della creazione della tela avviene già un pre salvataggio, non è necessario controllare ed eventualmente far salvare con nome nel caso sia il primo salvataggio. Lo script funziona in modo molto semplice, per prima come controlla la path dell’ultimo file aperto nel file json così da sapere che file sovrascrivere, e successivamente prende la nuova texture dalla tela e la sovrascrive al file. Visto che alla tela possono venir assegnate texture prese da file sia jpg che png prima di salvare fa un controllo per vedere di rimanere con la sterssa estensione, visto che i metodi di salvataggio per jpg e quelllo per png sono due metodi distinti.

### Salvataggio con nome

Il salvataggio con nome avviene appena viene premuto l’apposito pulsante, prendendo per prima cosa la texture attuale dalla tela, poi apre l’explorer e ritorna la path selezionata, dopo di che controlla l’estensione e salva la texture. Una volta fatto questo va a scrivere all’ultimo index nel file json la path così da sapere dove fare un eventuale salvataggio rapido in futuro ed elimina eventuali doppioni all’interno di esso.

### Ctrl + Z

Il ctrl Z o Undo funziona tramite una lista grande fino a 15 texture su cui vengono salvate e instanziate le texture dopo ogni modifica. Una volta che viene salvata la 16 la 1 viene eliminata facendo si che ce ne siano sempre 15. Al momento del ctrl Z viene assegnata alla tela la penultima texture salvata e poi questa viene eliminata dalla lista così che al prossimo ctrl z la texture presa sarà quella ancora prima.

### Riempimento

L’azione di fill, o riempimento, è stata sviluppata in modo simile all’azione disegno, infatti prevedeva il controllo tramite isPinching() della mano destra per sapere quando farlo e prendeva il punto utilizzando i RayCast per saper dove farlo.

Per sapere quando fare il riempimento e quando disegnare è stata aggiunto uno strumento selezionabile nella sezione di selezione strumento.

Il funzionamento non prevedeva altro che partire dal punto indicato e continuare a colorare in tutte le direzioni finchè non si fosse incontrato un pixel di colore diverso.

Questa azione tuttavia è risultata troppo pensate e abbiamo dunque deciso di rimuoverla dal progetto.

### Esci

Per uscire all’interno del menu c’è un pulsante che porta ad una schermata di uscita. Qui si potrà scegliere se annullare e quindi tornare a disegnare, o uscire. Per uscire si potrà scegliere inoltre se salvare gli ultimi progressi o non salvarli tenendo l’ultimo salvataggio. Se si sceglie di salvare prima di chiudere l’applicazione il programma non farà altro che richiamare il metodo salva come quando si clicca sul bottone di salvataggio.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case:  Riferimento: | TC-001  REQ-001 | Nome: | Tracciare le mani |
| Descrizione: | Controllare che prenda correttamente le mani. | | |
| Prerequisiti: | Il programma Virtual Painter ed il sensore Leap Motion | | |
| Procedura: | 1. Attaccare il Leap Motion al pc e avviare il programma. 2. Mettere le mani sopra il Leap Motion e muoverle guardando se nell’applicazione si muovono correttamente. | | |
| Risultati attesi: | Le mani nell’applicazione si muovono come le mani reali. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case:  Riferimento: | TC-002  REQ-002 | Nome: | Cambio strumento |
| Descrizione: | Controllare che il cambio dello strumento avvenga correttamente. | | |
| Prerequisiti: | Il programma Virtual Painter ed il sensore Leap Motion | | |
| Procedura: | 1. Disegnare una riga. 2. Aprire il pannello gestione strumenti e selezionare la gomma. 3. Cancellare la riga. | | |
| Risultati attesi: | Lo strumento selezionato passa da matita a gomma, e la linea inizialmente disegnata è cancellata. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case:  Riferimento: | TC-003  REQ-003 | Nome: | Cambio colore di disegno |
| Descrizione: | Controllare che il cambio del colore avvenga correttamente. | | |
| Prerequisiti: | Il programma Virtual Painter ed il sensore Leap Motion | | |
| Procedura: | 1. Tracciare una linea con il colore di default d’avvio. 2. Aprire il menu girando il braccio sinistro. 3. Selezionare con l’indice della mano destra il pulsate per aprire la tavolozza (secondo dall’alto) 4. Cambiare le componenti di rosso, verde, blu e trasparenza trascinando gli slider. 5. Rischiacciare il pulsante del punto 3 e disegnare una linea 6. Verificare che colore selezionato e quello della linea sia lo stesso. 7. Aprire nuovamente la tavolozza come nei punti 2 e 3, e selezionare un altro slot colore a destra. Cambiare le componenti e quindi disegnare una linea. 8. Verificare che il colore sia quello nuovo. 9. Riaprire la tavolozza e selezionare il primo colore utilizzato nel punto 6. 10. Disegnare una linea e verificare che il colore sia effettivamente quello usato ne punto 6. | | |
| Risultati attesi: | Nella tavolozza vengono impostati due colori e sulla tela vengono utilizzati per disegnare. La tavolozza memorizza i colori precedenti e possono venire riutilizzati e modificati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case:  Riferimento: | TC-001  REQ-004 | Nome: | Disegno a mano libera |
| Descrizione: | Controllare la possibilità di disegnare a mano libera. | | |
| Prerequisiti: | Il programma Virtual Painter ed il sensore Leap Motion | | |
| Procedura: | 1. Con la mano destra unire le dita e muoverla per tracciare una linea. | | |
| Risultati attesi: | Una linea viene disegnata. | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Sitografia

* <https://www.ultraleap.com/> , Sito ufficiale del Leap Motion Controller, 14-01-2021

# Allegati

* Diari di lavoro
* Codici sorgente
* Progetto in Unity
* QdC

1. https://developer.leapmotion.com/unity [↑](#footnote-ref-1)
2. Windows -> Package Manager [↑](#footnote-ref-2)
3. Cartella->Assets/Plugins/LeapMotion/Core/Example [↑](#footnote-ref-3)
4. Cartella->Assets/Plugin/LeapMotion/Core/Prefab [↑](#footnote-ref-4)
5. Cartella->Assets/Plugin/LeapMotion/Core/Prefab/HandModelPhysical [↑](#footnote-ref-5)
6. Cartella->Assets/Plugin/LeapMotion/Core/Prefab/HandModelNonHuman [↑](#footnote-ref-6)
7. RGBA **R**ed **G**reen **B**lue **A**lpha [↑](#footnote-ref-7)