



# Documentazione Virtual Painter

Titolo del progetto: Virtual Painter  
Alunno/a: Karim Galliciotti, Zeno Darani, Stefano Mureddu e Sara Bressan  
Classe: I3AA-AC  
Anno scolastico: 2020/2021  
Docente responsabile: Luca Muggiasca

---

1	Introduzione .....	4
1.1	Informazioni sul progetto .....	4
1.2	Abstract .....	4
1.3	Scopo .....	4
2	Analisi .....	5
2.1	Analisi del dominio .....	5
2.2	Analisi e specifica dei requisiti .....	5
2.3	Use case .....	6
2.4	Pianificazione .....	7
2.4.1	Analisi.....	8
2.4.2	Progettazione .....	8
2.4.3	Implementazione.....	8
2.4.4	Test .....	8
2.4.5	Documentazione .....	8
2.4.6	Consegna.....	8
2.5	Analisi dei mezzi.....	9
2.5.1	Software .....	9
2.5.2	Hardware.....	9
3	Progettazione .....	10
3.1	Design dell'architettura del sistema .....	10
3.2	Design dei dati e database.....	14
3.3	Design delle interfacce .....	15
3.4	Design procedurale .....	16
4	Implementazione .....	21
4.1	Menu e Interfacce Grafiche.....	21
4.1.1	Start Page GUI.....	22
4.1.2	GUI di configurazione del foglio (GUI "Configurazione Tela") .....	23
4.1.3	Creazione file .....	24
4.1.4	Impostazioni Main Camera .....	25
4.1.5	Caricamento immagini .....	25
4.1.6	Caricamento tela disegnabile.....	25
4.2	Leap Motion.....	26
4.2.1	Prova Leap Motion Controller .....	26
4.2.2	Tracciamento mani .....	27
4.2.3	Lettura movimenti .....	29
4.2.4	HandReset e HandFinish.....	29
4.3	Disegno .....	30
4.3.1	Azione Disegno .....	30
4.3.2	Selezione colore .....	30
4.3.3	Selezione strumento e dimensione della penna .....	32
4.3.3.1	Visualizzazione dello strumento in uso .....	32
4.4	Azioni varie ed inventario .....	33
4.4.1	Selezione strumento .....	33
4.4.2	ColorPicker.....	33
4.4.3	Exit .....	34
4.4.4	Salvataggio .....	34
4.4.5	Salvataggio con nome .....	34
4.4.6	Impostazioni .....	35
4.4.7	Ctrl + Z .....	35
4.4.8	Riempimento .....	35
4.4.9	Rotazione della tela .....	35
4.4.10	Zoom della tela .....	35
4.4.11	Info Panel .....	36
5	Test.....	37
5.1	Protocollo di test.....	37
5.2	Risultati test.....	40

5.2.1	Tracciare le mani – TC-001 .....	40
5.2.2	Cambio Strumento – TC-002 .....	41
5.2.3	Cambio colore di disegno – TC-003 .....	42
5.2.4	Disegno a mano libera – TC-004 .....	44
5.2.5	Salvataggio disegno – TC-005.1 .....	44
5.2.6	Salvataggio con nome – TC-005.2 .....	46
5.2.7	Importare un'immagine – TC-006 .....	47
5.2.8	Applicativo stand alone - TC-007 .....	49
5.2.9	Tabella dei risultati .....	49
5.3	Mancanze/limitazioni conosciute .....	50
6	Consuntivo .....	51
6.1	Analisi .....	52
6.2	Progettazione .....	52
6.3	Implementazione .....	52
6.4	Test .....	52
6.5	Documentazione .....	52
6.6	Consegna .....	52
7	Conclusioni .....	53
7.1	Sviluppi futuri .....	53
7.2	Considerazioni Personali .....	53
7.2.1	Sara Bressan .....	53
7.2.2	Karim Galliciotti .....	53
7.2.3	Stefano Mureddu .....	53
7.2.4	Zeno Darani .....	53
8	Bibliografia .....	54
8.1	Sitografia .....	54
9	Indice delle figure .....	55
10	Allegati .....	55
11	Indice analitico .....	56

## 1 Introduzione

### 1.1 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:	2021-05-20

### 1.2 Abstract

*During the second semester of the 2020-2021 school year, our client requested the realization of a virtual drawing board that works with a Leap Motion Controller. The program must grant the possibility to change the drawing instrument, the colour with a colour picker and allow you to draw freehand. After creating the painting, it will be possible to export it as an image, in addition the program will have to be standalone.*

*To create a graphically beautiful and functional application, we decided to use the Unity cross-platform graphics engine and the C# programming language, which has been present in our studies since the first year in the SAM school in Trevano, using C# we have the opportunity to improve our knowledge in this field.*

*To use Unity without compatibility issues we decided to work with Windows operating systems.*

*We were able to create a working product, despite the presence of some known bugs. The program allows you to do what is required in the specifications given by the customer, so you can use our program to paint freehand drawings using the Leap Motion Controller, which performs a tracking of the movements of the user's hands and allows you to show the drawing in progress on the screen.*

### 1.3 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 colore 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.

## 2 Analisi

### 2.1 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 standalone, e non necessiterà particolari conoscenze da parte degli utenti.

### 2.2 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 standalone.
Priorità	1
Versione	1.0
Note	

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

### 2.3 Use case

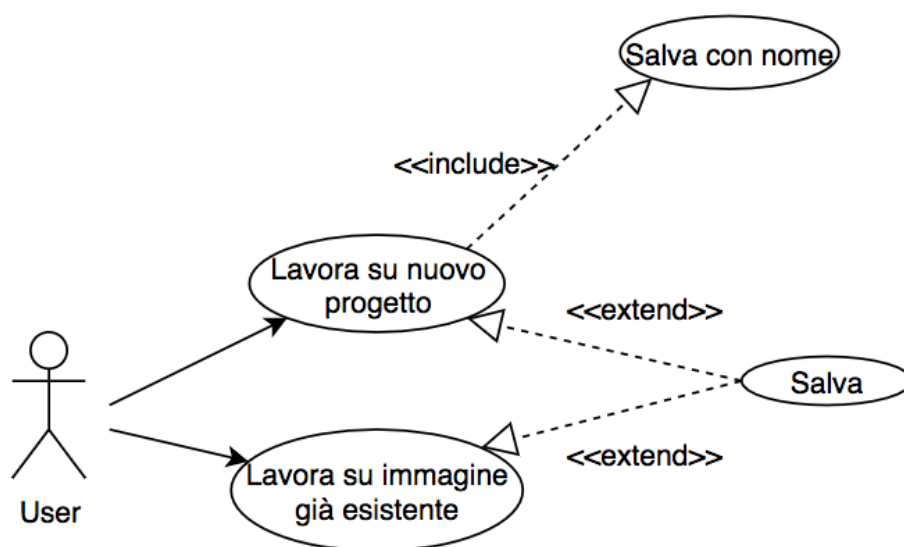


Figura 1 - Use Case

## 2.4 Pianificazione

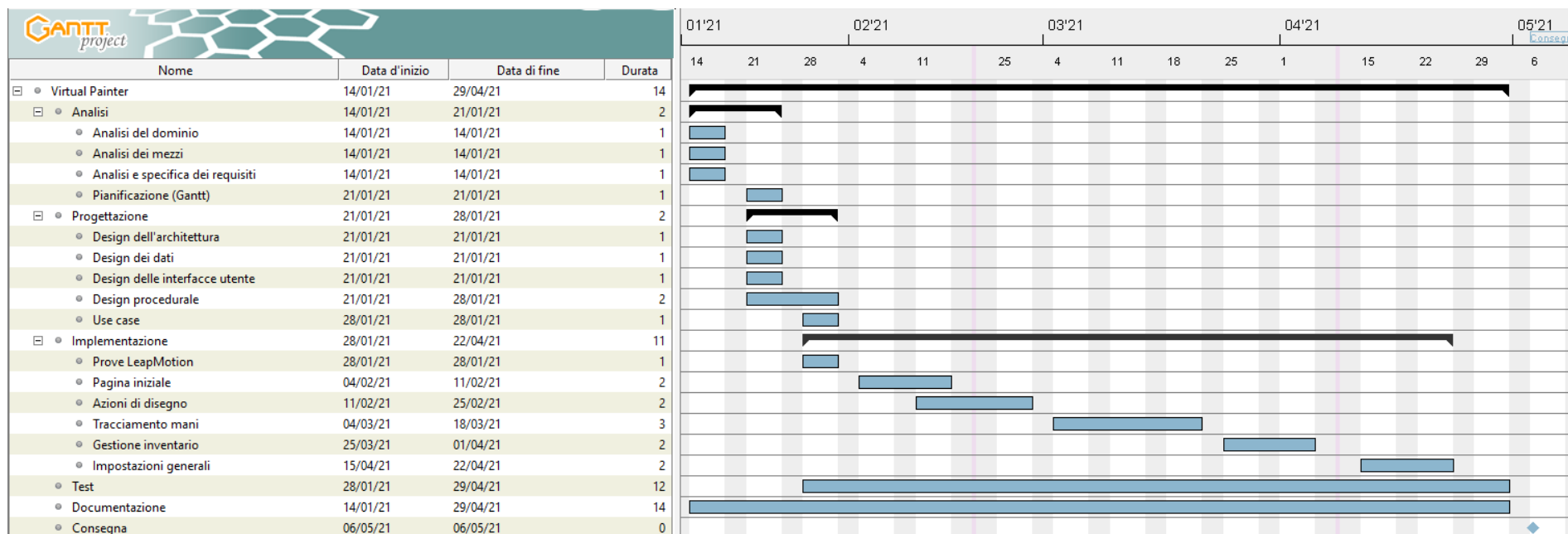


Figura 2 - Gantt preventivo

### 2.4.1 Analisi

La fase di analisi consente al gruppo di capire in cosa consiste il progetto e quali sono le specifiche date dal nostro docente responsabile, durante questa fase eseguiremo l'analisi del dominio, l'analisi dei mezzi, l'analisi e la specifica dei requisiti e la pianificazione descritta dal Gantt preventivo.

### 2.4.2 Progettazione

Durante la fase di progettazione faremo il design dell'architettura del sistema, il design della gestione dei dati, il design delle interfacce, il design procedurale e lo Use Case.

La progettazione è una parte importante del nostro progetto, in quanto determina l'organizzazione generale del progetto, oltre che a definire in parte come diventerà il progetto e come verrà implementato.

### 2.4.3 Implementazione

L'implementazione è suddivisa in sei importanti fasi del nostro progetto, inizialmente eseguiremo dei test e delle prove per capire il funzionamento del sensore Leap Motion che ci è stato fornito, in questo modo potremo capire se ci sono dei problemi di compatibilità con i nostri sistemi oppure se ci sono imprevisti che potrebbero compromettere il nostro progetto.

Dopo aver creato il progetto di Unity inizieremo a creare le diverse interfacce grafiche utili all'utilizzo del nostro programma (menu principale), questa parte è presente nel Gantt con il nome "Pagina Iniziale".

In parallelo alla creazione del menu una parte del team si dedicherà all'implementazione dell'azione di disegno che consentirà all'utente di disegnare, dopodiché potremo iniziare a fare un tracciamento delle mani tramite il Leap Motion e potremo implementare l'azione disegno creata in precedenza con l'utilizzo del Leap Motion Controller.

Quando saremo sicuri che l'opzione di disegno funzionerà potremo iniziare a gestire l'inventario (menu che consentirà all'utente di salvare il disegno, cambiare strumento e colore con la quale si disegnerà). Infine nella pianificazione abbiamo aggiunto una parte chiamata "impostazioni generali" la quale comprende eventuali funzionalità aggiuntive.

### 2.4.4 Test

I test verranno eseguiti durante le fasi d'implementazione secondo i protocolli di test descritti nei capitoli seguenti.

### 2.4.5 Documentazione

La documentazione è una parte importante del nostro progetto e consiste sia nella creazione di questa documentazione che tenere traccia di tutto ciò che si è fatto giornalmente sui diari presenti su Git Hub, inoltre tutto verrà tracciato anche su Trello così da consentire al docente responsabile del progetto i nostri progressi. La documentazione è uno strumento utile anche per capire cosa gli altri componenti del gruppo stanno implementando.

### 2.4.6 Consegna

La consegna è fissata al 20 maggio 2021, siccome inizialmente la data di consegna non era definita abbiamo inserito nel Gantt una data plausibile per la consegna del progetto.



## 2.5 Analisi dei mezzi

### 2.5.1 Software

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

### 2.5.2 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)
- PC personale di Zeno (Acer)
- Leap Motion Controller fornito dal docente

### 3 Progettazione

#### 3.1 Design dell'architettura del sistema

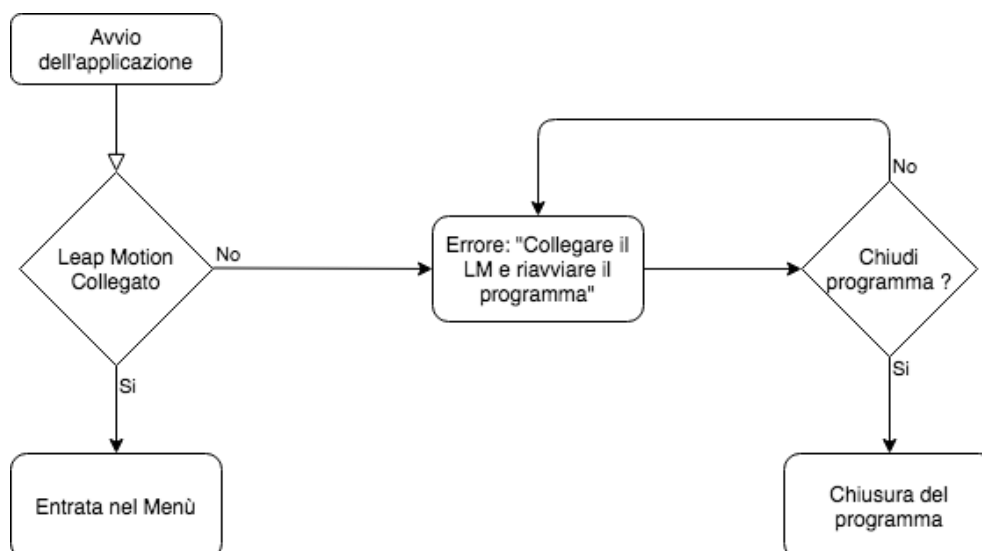


Figura 3 - Design dell'architettura del sistema

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.

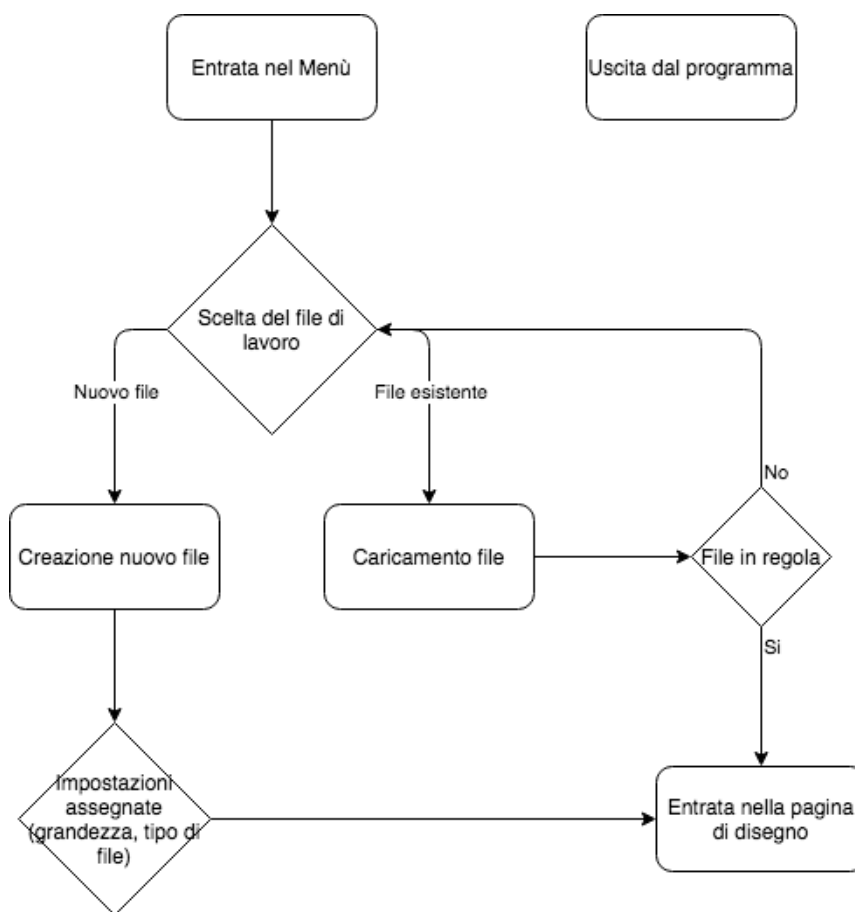


Figura 4 - Gestione del menu

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.

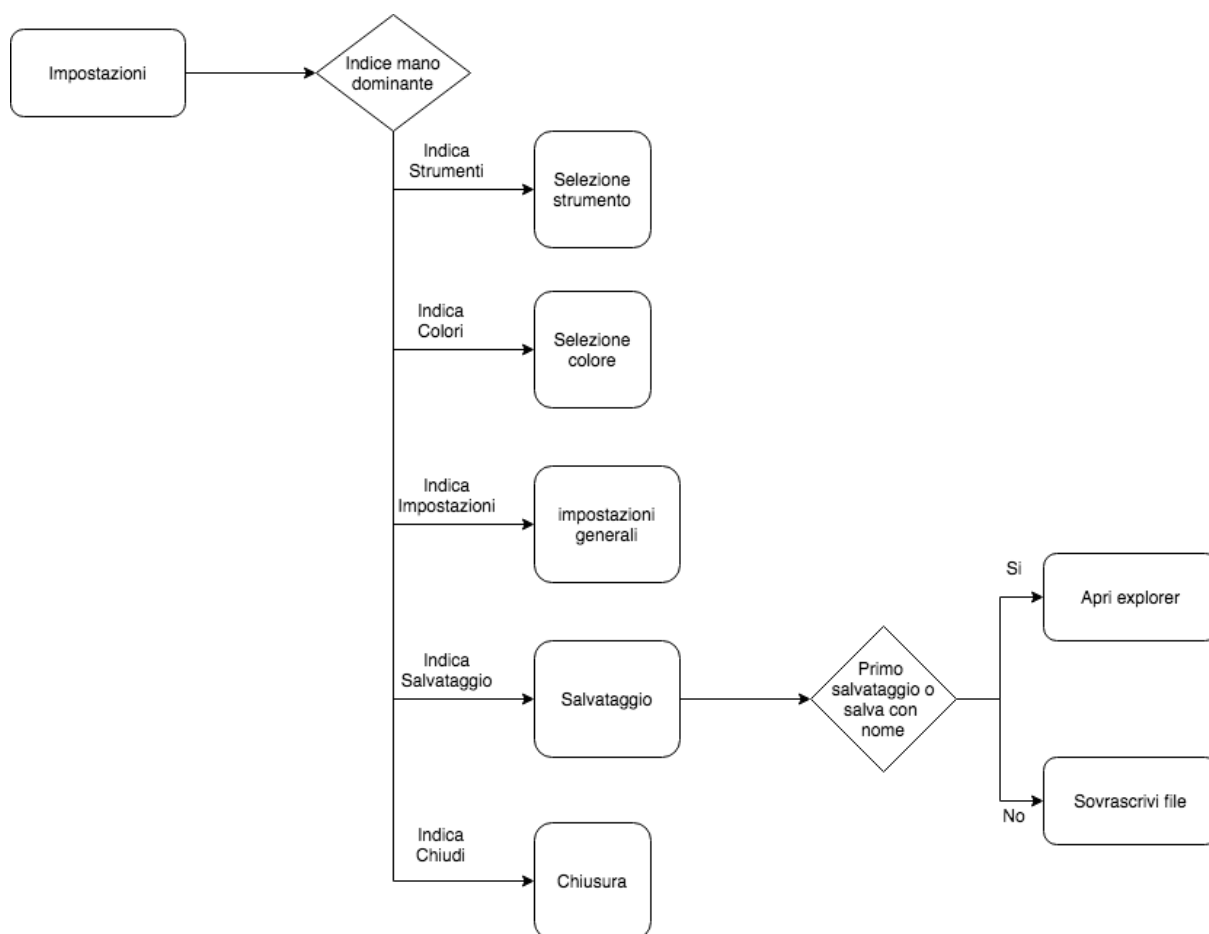


Figura 5 - Design impostazioni

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.

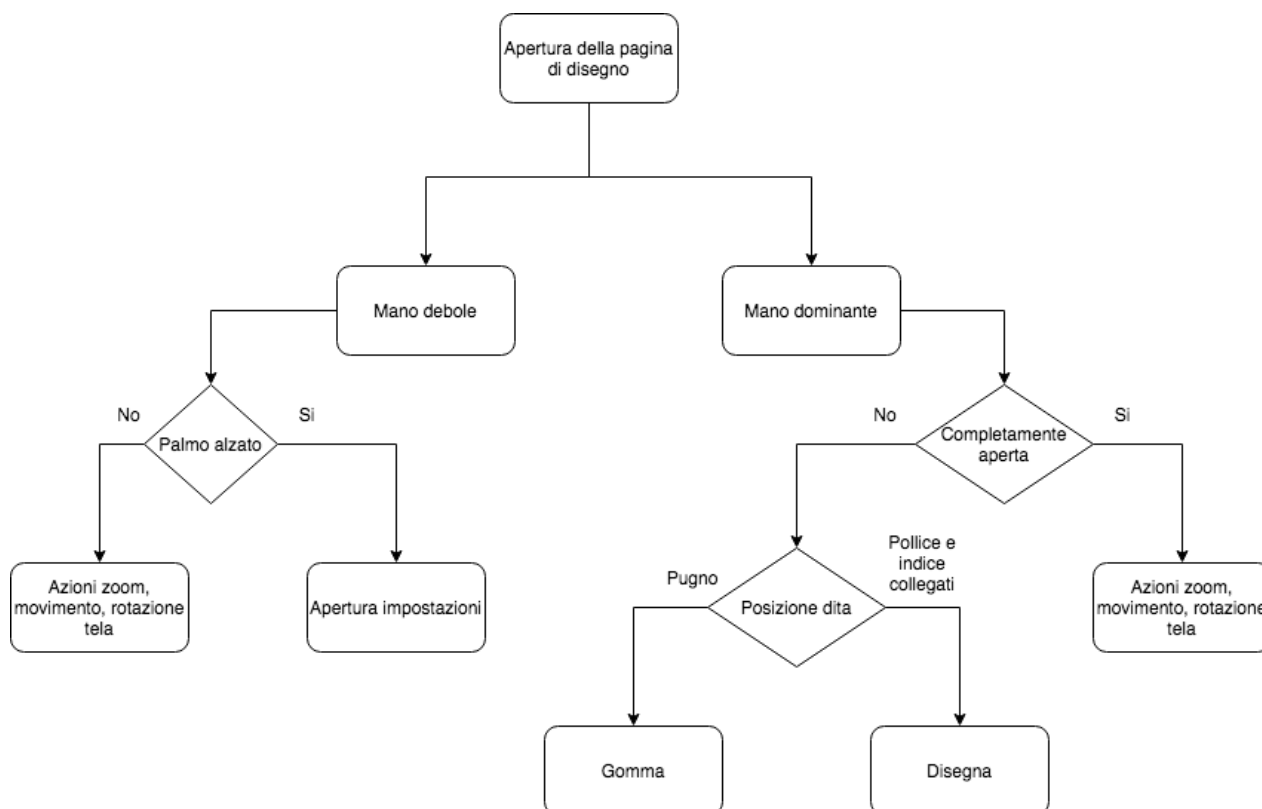


Figura 6 - Design apertura area di disegno

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 sì 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.

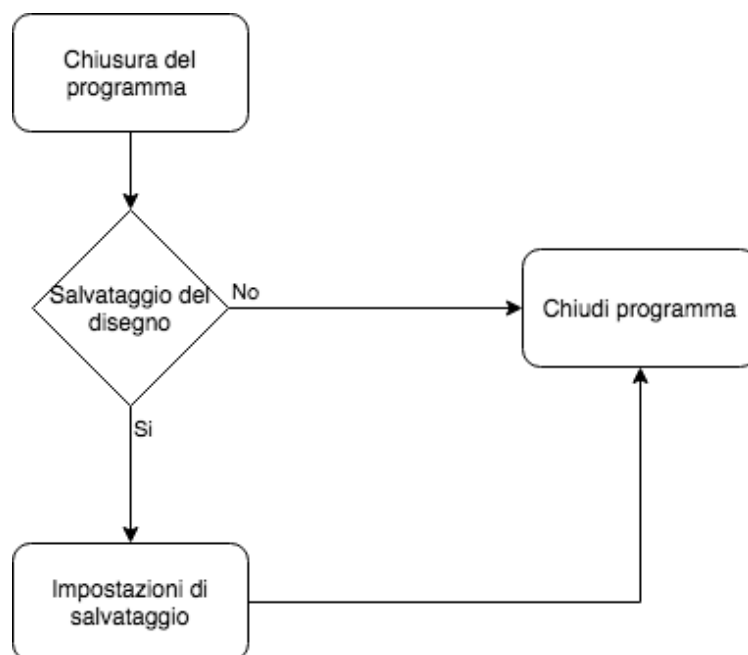


Figura 7 - Design chiusura del programma

L'applicazione si chiude.

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

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

### 3.2 Design dei dati e database

Non abbiamo preso in considerazione l'uso di database. I file recenti verranno visualizzati in un file json.

### 3.3 Design delle interfacce

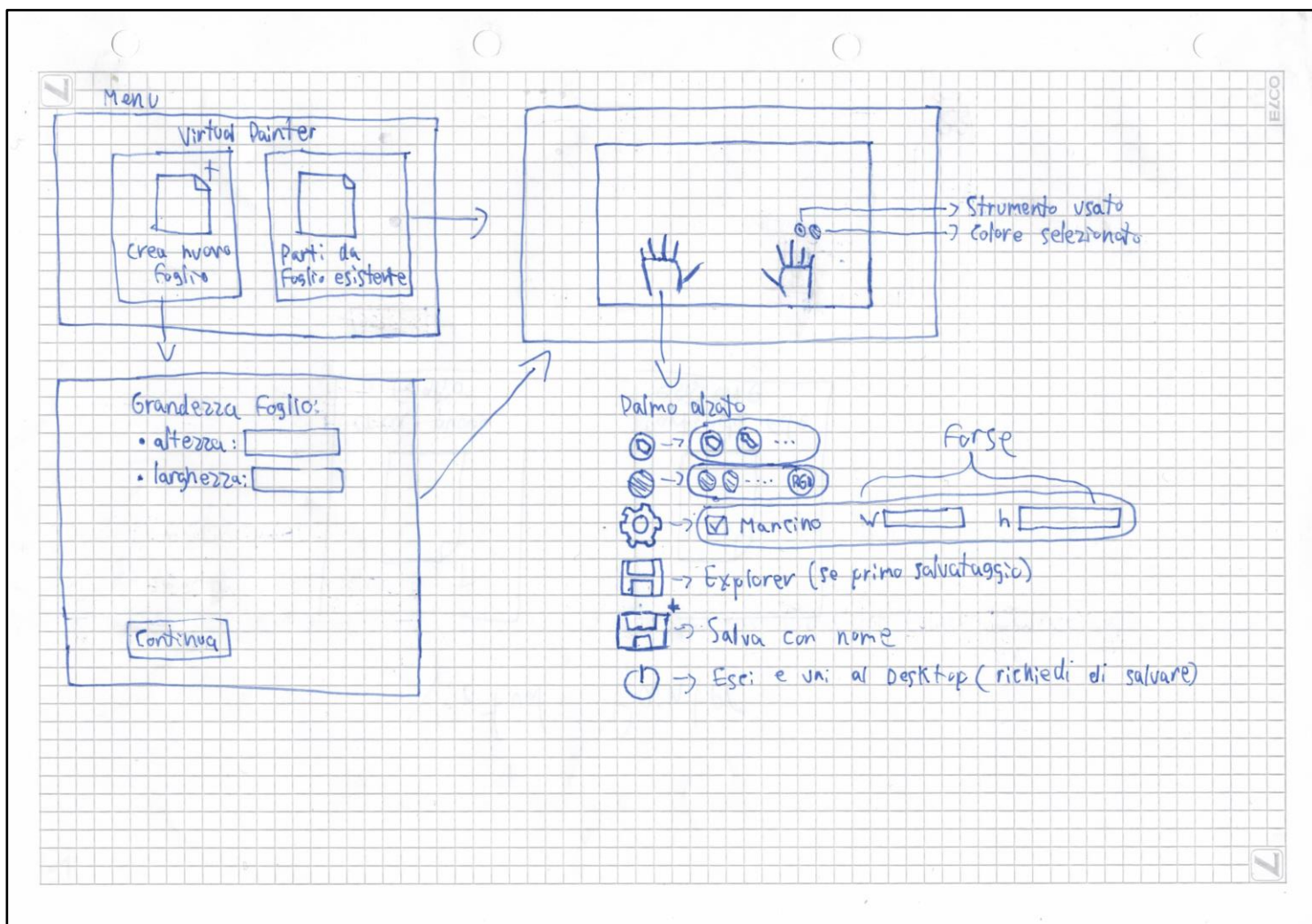


Figura 8 - Design delle interfacce

La nostra idea sulle interfacce dell'applicazione consiste nell'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.

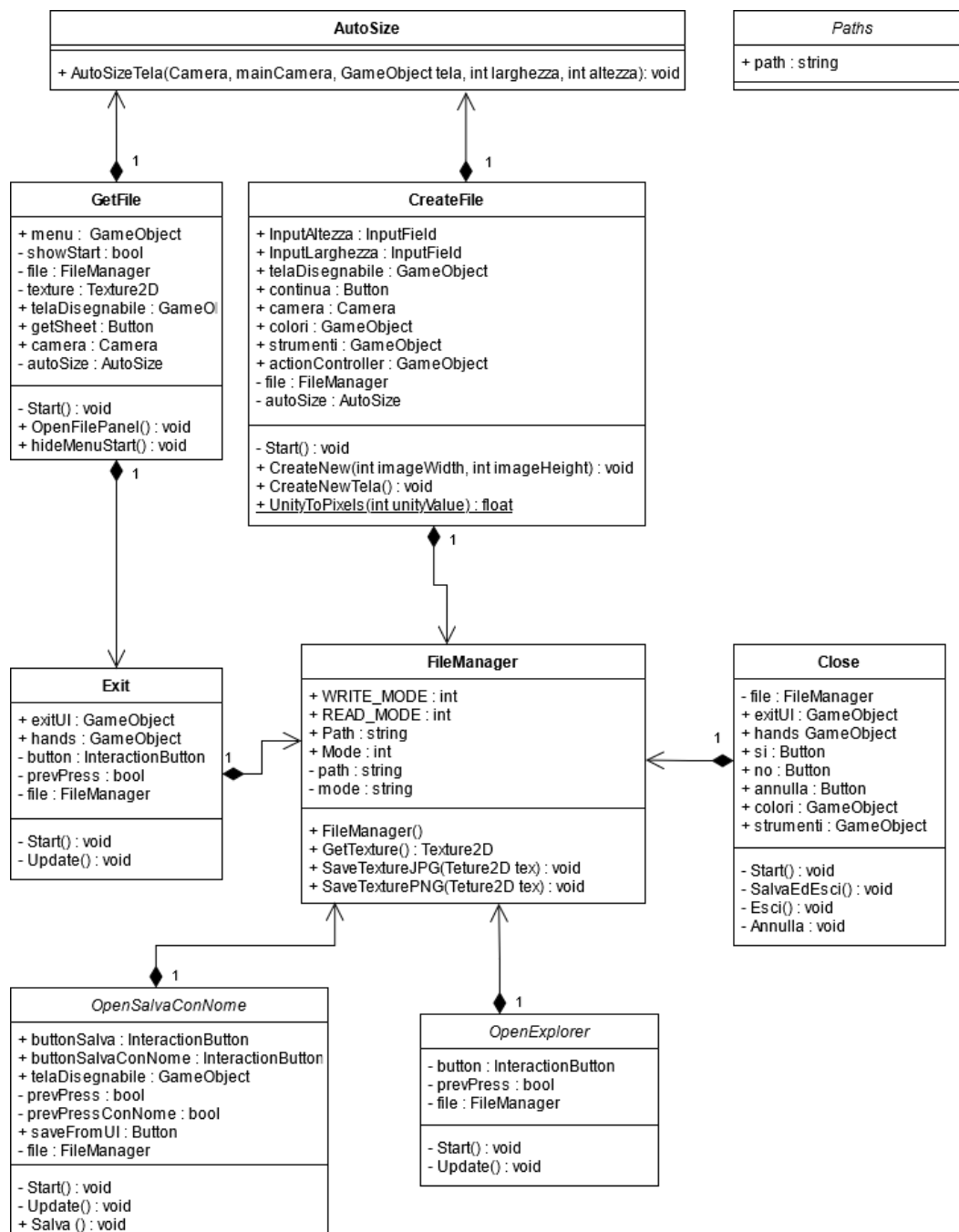
Una volta caricato o creato il foglio su cui lavorare ci troveremo sulla tela dove potremmo disegnare. La tela si trova in alto a destra nell'immagine. Sulla tela potremmo vedere il foglio di lavoro e le nostre mani e, sulla mano forte, in questo caso rappresentata da quella di destra, potremmo vedere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

### 3.4 Design procedurale

In questo capitolo sono mostrate le classi necessarie al funzionamento del programma. Per una descrizione più dettagliata consultare [la documentazione delle classi](#).

Le classi seguenti servono per la creazione, salvataggio e caricamento della tela.





Queste classi sono utili per gestire i vari menu del programma.

<i>OpenStrumenti</i>
<ul style="list-style-type: none"> <li>- button : InteractionButton</li> <li>- prevPress : bool</li> <li>- active : bool</li> <li>+ instrument : GameObject</li> <li>+ rightHand : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> </ul>

<i>OpenImpostazioni</i>
<ul style="list-style-type: none"> <li>- button : InteractionButton</li> <li>- prevPress : bool</li> <li>+ configurazione_tela : GameObject</li> <li>+ hands : GameObject</li> <li>+ tela : GameObject</li> <li>+ actionController : GameObject</li> <li>+ colorPicker : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> </ul>

<i>OpenColori</i>
<ul style="list-style-type: none"> <li>- button : InteractionButton</li> <li>- prevPress : bool</li> <li>- active : bool</li> <li>+ colorPicker : GameObject</li> <li>+ rightHand : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> </ul>

<i>ShowMenu</i>
<ul style="list-style-type: none"> <li>+ menu : GameObject</li> <li>+ configMenu : GameObject</li> <li>+ colorPicker : GameObject</li> <li>+ instruments : GameObject</li> <li>- showStart : bool</li> <li>- showConfig : bool</li> <li>+ nuovofoglio : Button</li> <li>+ hands : GameObject</li> <li>+ telaDisegnabile : GameObject</li> <li>+ exitMenu : GameObject</li> <li>+ info : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> <li>+ ChangeConfig() : void</li> <li>+ CreateJson() : void</li> </ul>

<i>ChangeInstrument</i>
<ul style="list-style-type: none"> <li>+ rightHand : GameObject</li> <li>+ pen : InteractionButton</li> <li>+ eraser : InteractionButton</li> <li>+ fill : InteractionButton</li> <li>+ colorPicker : GameObject</li> <li>- prevPenPress : bool</li> <li>- activePen : bool</li> <li>- prevErasePress : bool</li> <li>- activeErase : bool</li> <li>- prevFillPress : bool</li> <li>- activeFill : bool</li> </ul>
<ul style="list-style-type: none"> <li>- Update() : void</li> </ul>

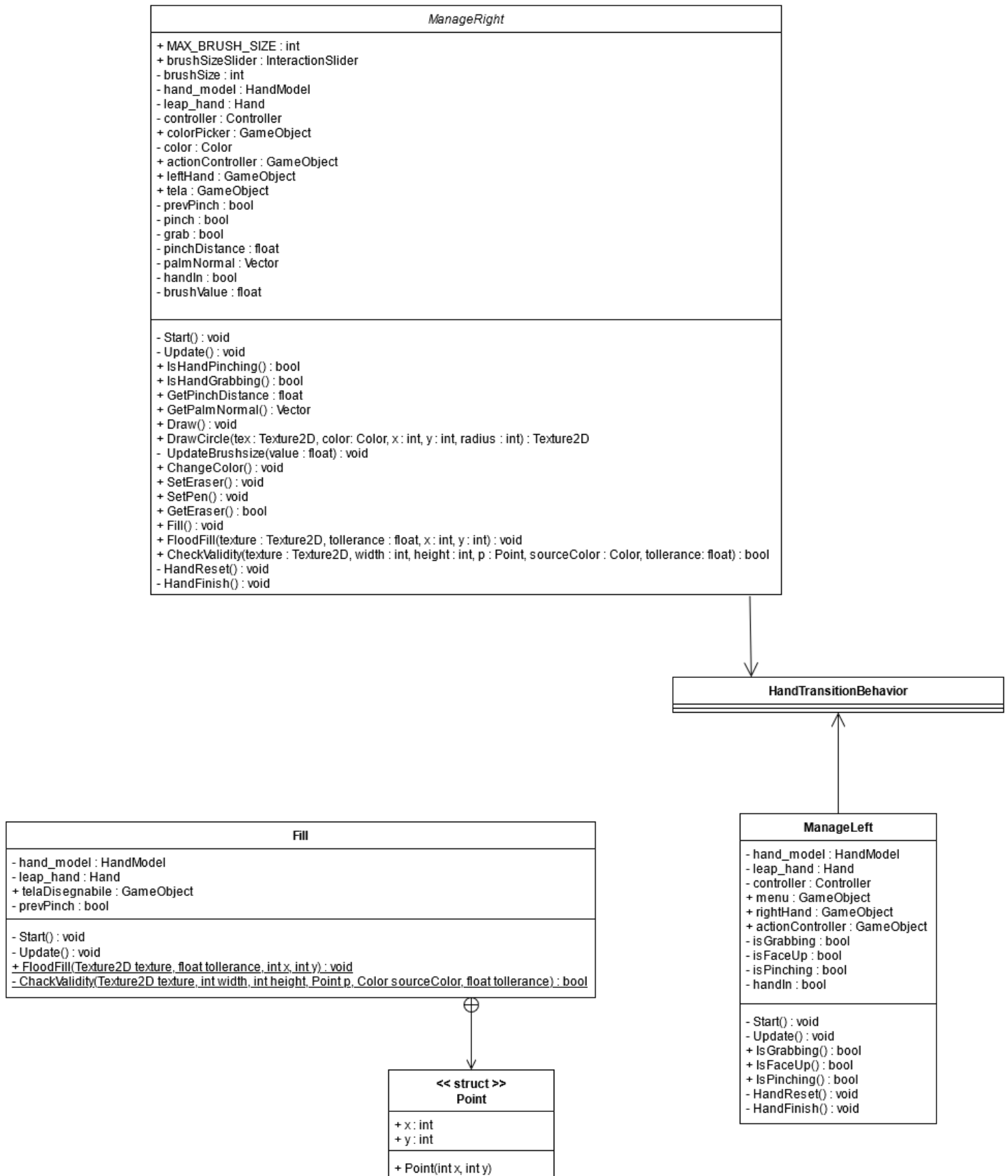
<i>ColorPicker</i>
<ul style="list-style-type: none"> <li>+ ROWS_OF_SELECTABLE_COLORS : int</li> <li>+ COLS_OF_SELECTABLE_COLORS : int</li> <li>+ COLORS_DISTANCE : float</li> <li>+ SELECTED_COLOR_SIZE_INCR : float</li> <li>+ DefaultColor : Color</li> <li>+ Color : GameObject</li> <li>+ SliderRed : InteractionSlider</li> <li>+ SliderGreen : InteractionSlider</li> <li>+ SliderBlue : InteractionSlider</li> <li>+ SliderAlpha : InteractionSlider</li> <li>+ RValue : TextMesh</li> <li>+ GValue : TextMesh</li> <li>+ BValue : TextMesh</li> <li>+ AValue : TextMesh</li> <li>+ SelectedColor : Color</li> <li>- activeColor : GameObject</li> <li>- colori : List&lt;GameObject&gt;</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- OnColorSelected(float value) : void</li> <li>- OnChangedAlpha(float value) : void</li> <li>- OnChangedBlue(float value) : void</li> <li>- OnChangedGreen(float value) : void</li> <li>- OnChangedRed(float value) : void</li> </ul>

Queste classi permettono il disegno sulla tela.

<i>StartDrawing</i>
<ul style="list-style-type: none"> <li>+ tela : GameObject</li> <li>+ configMenu : GameObject</li> <li>+ caricaFoglio : Button</li> <li>+ creaNuovaTela : GameObject</li> <li>+ hands : GameObject</li> <li>+ telaDisegnabile : GameObject</li> <li>+ actionController : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> <li>+ HideCaricamento() : void</li> <li>+ HideNuovaTela() : void</li> <li>+ ShowHands() : void</li> </ul>

<i>Undo</i>
<ul style="list-style-type: none"> <li>- last : bool</li> <li>- pressed : bool</li> <li>- hand_model : HandModel</li> <li>- leap_hand : Hand</li> <li>- textures : List&lt;Texture2D&gt;</li> <li>+ telaDisegnabile : GameObject</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> </ul>

<i>DrawRayCast</i>
<ul style="list-style-type: none"> <li>+ MAX_BRUSH_SIZE : int</li> <li>+ BrushSizeSlider : InteractionSlider</li> <li>+ telaDisegnabile : GameObject</li> <li>+ leftHand : GameObject</li> <li>- brushSize : int</li> <li>- hand_model : HandModel</li> <li>- leap_hand : Hand</li> <li>- controller : Controller</li> <li>- color : Color</li> </ul>
<ul style="list-style-type: none"> <li>- Start() : void</li> <li>- Update() : void</li> <li>+ DrawCircle(Texture2D tex, Color color, int x, int y, int radius) : Texture2D</li> <li>- UpdateBrushSize(float value)</li> </ul>





## 4 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.

Prima di iniziare a creare il progetto bisogna impostare il timestep a 0.0111111 (l'equivalente di 90 frames per secondo) come consigliato dalla documentazione ufficiale del Leap Motion Controller. Questa impostazione è da cambiare in "Edit → Project Settings → Time".

Per il corretto funzionamento del programma è necessario impostare la risoluzione dello schermo a 19:6, in questo modo la tela andrà a coprire interamente lo schermo.

### 4.1 Menu e Interfacce Grafiche

Inizialmente pensavamo di creare diverse scene per ogni interfaccia grafica, ma in seguito per evitare eventuali errori durante il passaggio di informazioni tra scene abbiamo deciso di collassare tutto il programma in un'unica Scene (scena di Unity) chiamata "Tela". Abbiamo quindi copiato e incollato i due Canvas già creati ("start\_page" e "configurazione\_tela") nella scena appena creata.

A questo punto abbiamo creato un GameObject vuoto chiamato "ActionController" che si occupa di gestire e far partire i diversi script del programma non strettamente legati ad un oggetto (ad esempio lo script per creare la tela).

Per fare in modo che si possano selezionare i bottoni dei vari Canvas è necessario metterli in Layer differenti (uno più avanti e l'altro dietro).

Dopo aver creato le due GUI descritte nei capitoli seguenti abbiamo creato uno script nominato "ShowMenu" il quale gestisce la visualizzazione a schermo della GUI iniziale e nasconde ciò che non deve essere visto inizialmente come ad esempio le mani virtuali.

#### 4.1.1 Start Page GUI

Per prima cosa è stato creato il Canvas “start\_page”, che corrisponde alla GUI del primo menù che apparirà all’avvio del nostro programma.

Al canvas è stato impostato il render mode: Screen Space - Camera, in questo modo il canvas sarà sempre grande come la Main Camera, quindi grande come l'applicazione. Il Canvas ha inoltre un Vertical Layout Group, che consentirà di disporre il Label con il titolo del programma e i bottoni uno sopra l'altro, e sia l'altezza che la larghezza dei componenti all'interno di questo layout sono definiti dalla grandezza del Canvas (altezza e larghezza sono spuntate nelle impostazioni: "Control Child Size" e "Child Force Expand").

In seguito, abbiamo aggiunto nel Canvas un Label con scritto “Virtual Painter” e un pannello che andrà a contenere i due bottoni con la quale caricare o creare un nuovo file (tela).

Il pannello ha a sua volta un Horizontal Layout Group che serve a disporre i due bottoni in fila in modo che si adattino alla dimensione del pannello, in questo caso è stato inserito anche un padding di 10 su tutti i lati (in questo modo i bottoni sono separati tra loro e dai bordi della finestra).

I due bottoni nominati in precedenza consistono nel bottone “NuovoFoglio”, il quale servirà a creare una nuova tela, e il bottone “FoglioEsistente”, che servirà invece ad aprire l'Explorer per selezionare un immagine preesistente da inserire nella tela.

Per inserire le immagini nei bottoni abbiamo importato un package dall'Asset Store di Unity (il pacchetto è gratuito ed è installabile dalla seguente pagina: <https://assetstore.unity.com/packages/2d/gui/icons/simple-icon-pastel-tone-107568>), il pacchetto è stato quindi importato tramite l'Unity Hub.

Ad ogni bottone è stato poi assegnato un Grid Layout Group (con “Child Alignment” settato a “Middle Center”) nella quale sono stati inseriti un'immagine (presa dal Package installato in precedenza) e un testo.

La GUI appena creata serve all'utente per scegliere se aprire un file esistente sulla quale disegnare oppure se creare un nuovo file.

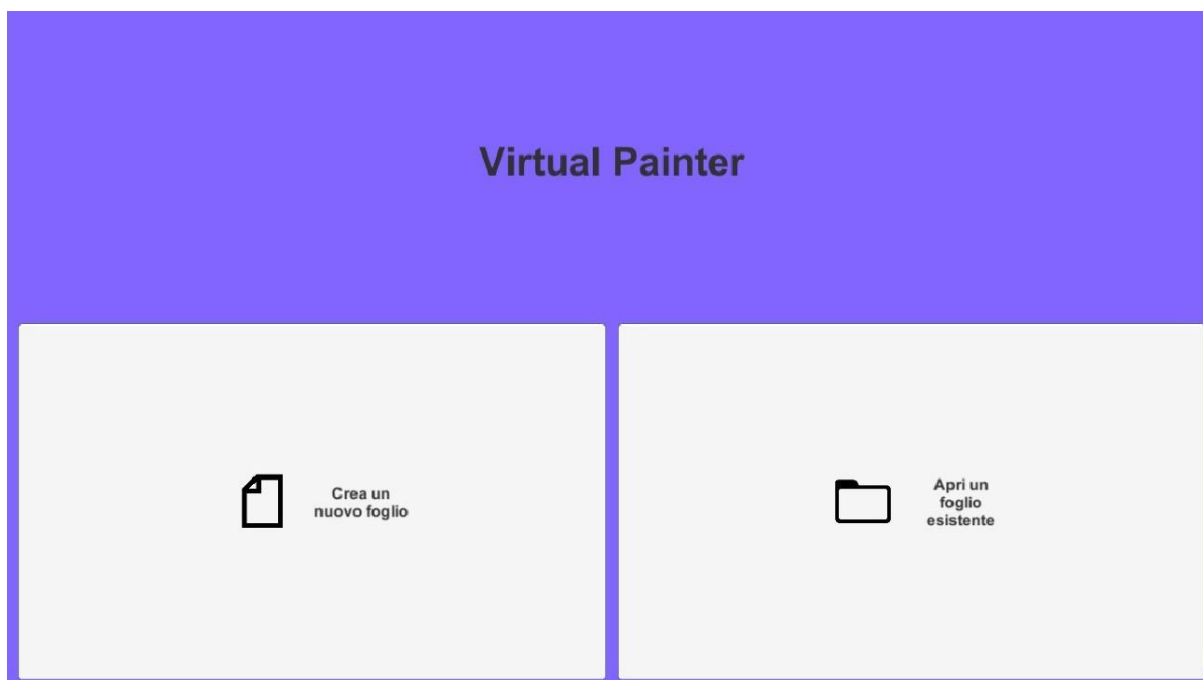


Figura 9 - Start Page GUI

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

#### 4.1.2 GUI di configurazione del foglio (GUI “Configurazione Tela”)

Nella scena del progetto è stato aggiunto il Canvas “configurazione\_tela” fungerà da GUI per l’inserimento delle dimensioni del nuovo disegno che andrà creato (vedi capitolo “Creazione file”).

Abbiamo poi aggiunto al Canvas appena creato un Vertical Layout Group, il quale permette agli elementi al suo interno di essere messi uno sotto l’altro. All’interno del Canvas abbiamo quindi aggiunto i seguenti elementi in ordine: un Label (testo “Grandezza foglio”), un pannello che va a contenere gli Input Field e i Label di descrizione per definire le dimensioni della tela (disegno), infine abbiamo aggiunto il bottone “continua”, la dimensione delle componenti appena descritte viene definita dal Canvas che le contiene e hanno la grandezza massima contenibile dal Canvas stesso.

Gli Input Field che definiscono altezza e larghezza della tela accettano solamente numeri interi, il tipo di input accettato è stato selezionato nel componente sotto “Input Field” dove c’è scritto “Content Type”.

Fintanto che questa GUI è visibile bisogna disattivare l’azione “MoveTela”, altrimenti il premere i tasti sul tastierino numerico del PC potrebbe traslare la tela.

Figura 10 - Configurazione Tela GUI

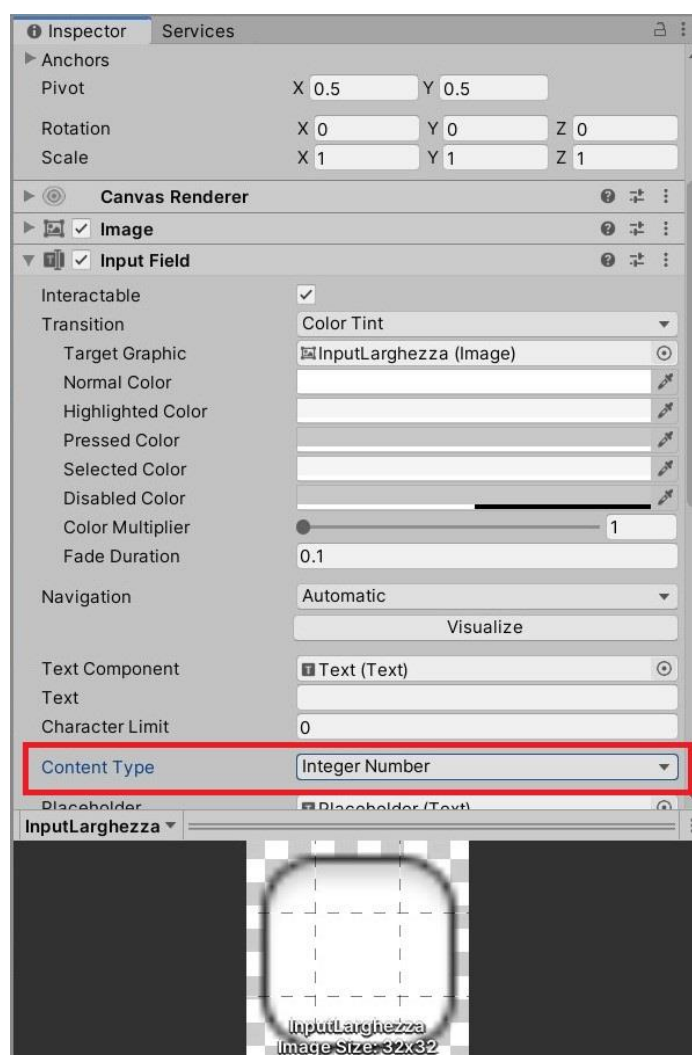


Figura 11 - Impostazioni Input Field

#### 4.1.3 Creazione file

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

Lo script che si chiama "CreateFile" al momento del click prende i dati inseriti dai form "configurazione\_tela" 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 del sistema operativo utilizzato e fa salvare all'utente il foglio appena creato con già un nome e una posizione. 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.



#### 4.1.4 Impostazioni Main Camera

La telecamera principale mostra ciò che l'utente finale vedrà a schermo, per questo tutti i componenti che andremo ad inserire devono essere centrati nella sua visuale. Per poter visualizzare sia le mani che la tela contemporaneamente abbiamo messa la telecamera principale in posizione  $x = 0$ ,  $y = 0$  e  $z = 3.066$ .

#### 4.1.5 Caricamento immagini

Per il caricamento delle immagini si utilizza uno script, che si chiama "GetFile", che viene richiamato quando viene cliccato il bottone nel menu principale per importare immagini esterne. Al momento del click apre l'Explorer del sistema operativo e permette di importare qualunque immagine PNG o JPG. Appena viene scelta l'immagine lo script si prende la path e, se non è mai stata utilizzata, la salva nel file "paths.json", dove vengono salvati tutti i percorsi delle immagini modificate. Se invece il percorso è già presente nella lista lo sposta solamente all'ultima posizione. Dopodiché crea la texture basata sull'immagine presa, la applica alla tela, nasconde il menu principale e infine scala la tela in modo da avercela della dimensione massima possibile.

#### 4.1.6 Caricamento tela disegnabile

Il caricamento della tela disegnabile avviene scalando i pixel. In partenza la nostra tela non è altro che un GameObject a forma di rettangolo bianco chiamato "TelaDisegnabile" ed è posizionata nella scena principale in posizione  $x = 0$ ,  $y = 140$ . La rotazione del GameObject corrisponde invece a  $x = 90$ ,  $y = 180$  e  $z = 0$ , mentre la scala iniziale (che verrà modificata automaticamente in seguito) corrisponde a  $x = 12$ ,  $y = 14$  e  $z = 12$ . Una volta che i dati per la grandezza vengono inseriti nel form "configurazione\_tela" sulla tela verrà applicata una texture delle dimensioni volute. Per fare in modo 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 pixels inseriti diviso 100. Inoltre, per fare in modo che la tela occupi tutto lo schermo, in orizzontale o in verticale, bisogna scalarla nuovamente ingrandendola o diminuendola mantenendo però sempre le proporzioni.

## 4.2 Leap Motion

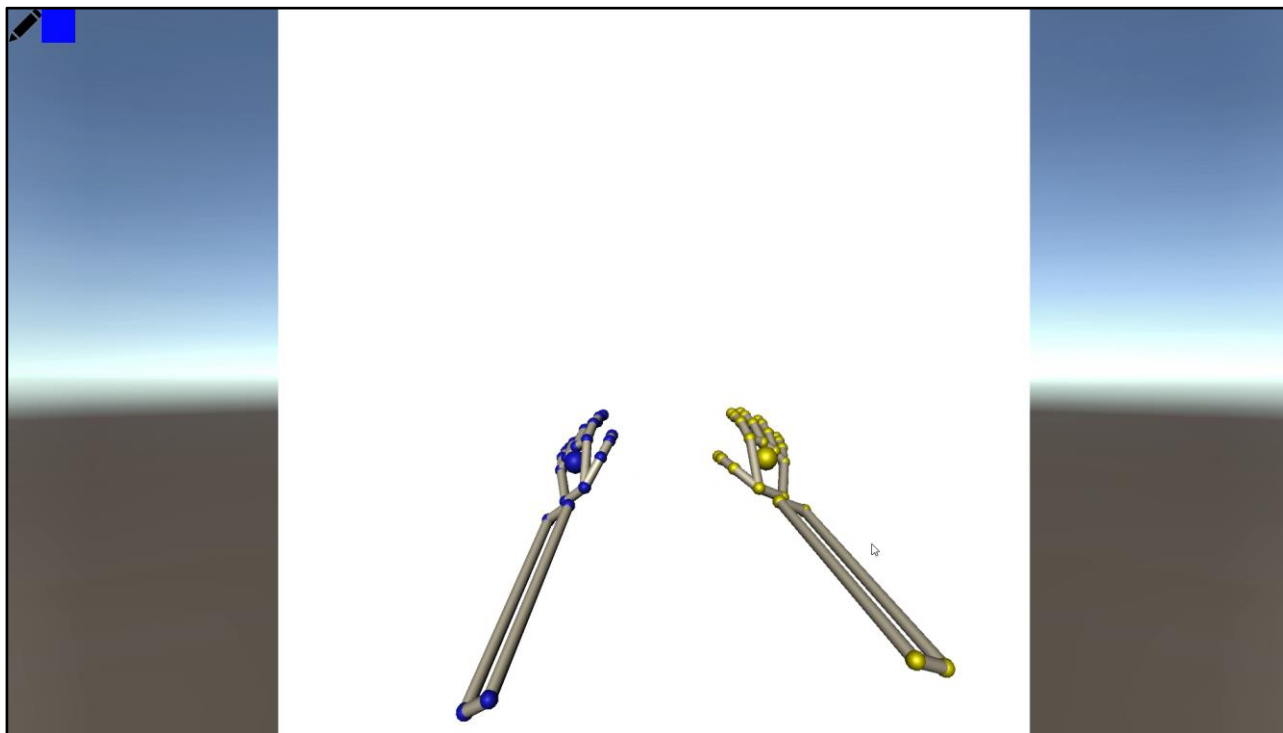


Figura 12 - Mani e tela

### 4.2.1 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.

## 4.2.2 Tracciamento mani

Per il tracciamento delle mani inizialmente bisogna scaricare gli Unity module packages<sup>1</sup>, ovvero i pacchetti per gli sviluppatori utilizzanti un Leap Motion. Una volta scaricati e importati si procede ad installare il plugin “XR plugin Management<sup>2</sup>”, 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)<sup>3</sup>”. 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<sup>4</sup> nella scena e si crea un nuovo GameObject di nome “HandsModel”. Questo dovrà contenere “RigidRoundHand\_L<sup>5</sup>”, “RigidRoundHand\_R”, “Capsule Hand Left”<sup>6</sup> 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 essere chiamati “Graphics\_Hands” e “Physics\_Hands”. Sotto “Graphics\_Hands”, nei campi Left Model e Right Model dovranno venir selezionati “RigidRoundHand\_L” e “RigidRoundHand\_R”.

Bisogna inoltre spuntare in entrambi i gruppi solo la casella “IsEnabled”.

### 4.2.2.1 HandsModel

Dopo aver eseguito la prova descritta in precedenza ed esserci assicurati il corretto tracciamento dei movimenti abbiamo inserito l’HandsModel e il LeapHandController creato in precedenza all’interno di un GameObject vuoto chiamato “Hands” (Il quale funge quindi da contenitore per le due mani virtuali). La posizione di “Hands” equivale a  $x = -1.04$ ,  $y = 0$  e  $z = -0.04$ .

<sup>1</sup> <https://developer.leapmotion.com/unity>

<sup>2</sup> Windows -> Package Manager

<sup>3</sup> Cartella->Assets/Plugins/LeapMotion/Core/Example

<sup>4</sup> Cartella->Assets/Plugin/LeapMotion/Core/Prefab

<sup>5</sup> Cartella->Assets/Plugin/LeapMotion/Core/Prefab/HandModelPhysical

<sup>6</sup> Cartella->Assets/Plugin/LeapMotion/Core/Prefab/HandModelNonHuman

#### 4.2.2.2 Attachment Hands

All'interno del HandsModel abbiamo inserito il prefab "Attachment Hands" che si trova nel progetto nella directory "Assets/Plugins/LeapMotion/Core/Prefabs" (vedi prefab presenti nel pacchetto del Leap Motion). All'interno di questo prefab si trovano due Attachment Hand (Left e Right), uno per la mano sinistra e uno per la mano destra. Nella mano sinistra abbiamo inserito un GameObject chiamato "Palm Forward Transform", il quale verrà utilizzato in seguito come target per indicare la posizione della mano e del suo palmo.

In seguito nella mano sinistra, abbiamo inserito il GameObject "Palm UI Pivot Animation", il quale controlla la rotazione della mano per gestire la visualizzazione (o la non visualizzazione) dell'inventario (pannello di bottoni attaccato alla mano sinistra). All'interno del GameObject appena creato abbiamo messo due GameObject (Hidden e Visible) contenenti le coordinate che indicano la posizione per la quale l'inventario risulta visibile.

"Palm UI Pivot Animation" ha al suo interno lo script "Transform Tween Behaviour" fornito dal Package scaricato in precedenza del LeapMotion Controller, le proprietà inserite sono:

- Target transform: Palm Forward Transform (GameObject creato in precedenza)
- Start Transform: Hidden
- End Transform: Visible
- Le altre impostazioni sono quelle di default.

Dopo aver eseguito i passaggi precedenti abbiamo creato il "Palm UI Pivot Anchor" il quale andrà a contenere la grafica dell'inventario, al suo interno ha lo script "Simple Facing Camera Callbacks" il quale ha le seguenti caratteristiche:

- To face camera: Palm Forward Transform
- Camera to face: none
- On Begin Facing Camera(): → Runtime Only → Palm UI Pivot Animation → TranformTweenBehaviour.PlayForward
- On End Facing Camera(): → Runtime Only → Palm UI Pivot Animation → TranformTweenBehaviour.PlayBackward

All'interno di questo contenitore abbiamo messo il GameObject "Palm UI" avente lo script "Ignore Collisions In Children" che serve a prevenire collisioni tra gli elementi della UI.

Dentro "Palm UI" abbiamo inserito un GameObject chiamato "Button Panel" che corrisponde all'inventario. Dentro l'inventario ho inserito una serie di bottoni 3d (6 bottoni con immagini diverse per ogni funzione dell'inventario) creati con dei cubi alla quale è stato assegnato lo script "Interaction Button" (Script del Package del LeapMotion), il Manager indicato delle proprietà dello script corrisponde all'InteractionManager (vedi capitoli seguenti).

Per gestire in modo più efficace l'orientamento della mano e la conseguente visualizzazione del menù inventario abbiamo aggiunto una script chiamato "ManageLeft" nella mano "RigidRoundHand\_L" il quale gestisce i movimenti della mano sinistra, il codice controlla l'orientamento del palmo della mano (la direzione y del palmo deve essere maggiore o uguale a 0.6), a questo punto se il palmo risulta girato verso l'alto l'inventario risulta attivo (cioè visibile), altrimenti viene disattivato. Se la mano non è visualizzata di conseguenza bisogna rendere invisibile il menù inventario.

Sempre nello script "ManageLeft" dopo aver controllato che la mano sinistra sia effettivamente visibile, viene controllato se entrambe le mani hanno il pollice e l'indice che si toccano (Pinching). Se le dita si toccano allora viene controllata la distanza tra le due mani:

- Se la distanza è maggiore di 0.38 unità di Unity allora viene eseguito un ZoomOut
- Se la distanza è minore di 0.2 unità di Unity allora viene eseguito un ZoomIn

Vedi metodi pubblici della classe Zoom per ZoomIn e ZoomOut.

Abbiamo in seguito creato il nuovo script "ManageRight" che si occupa di gestire i movimenti della mano destra. Usando la stessa metodologia di "ManageLeft", ManageRight è stato assegnato alla mano destra "RigidRoundHand\_R" e tramite il controller Leap Motion viene captato l'oggetto mano. Dopo aver controllato che la mano esista nel contesto corrente e che sia effettivamente la mano destra viene salvata in una variabile la posizione della mano. In seguito, viene controllato se l'indice tocca il pollice:

- Se è vero la variabile pinch viene settata a true ed inoltre viene salvata la distanza tra le due dita.
- Se è falso controlla la rotazione della mano e a dipendenza di quanto è ruotata richiama il metodo RotateRight o RotateLeft della classe Rotate. Se la mano è a pugno chiuso la rotazione torna come era inizialmente con il metodo NormalizeRotation della classe Rotate.

Questa classe ha inoltre i tre metodi pubblici: IsHandPinching, GetPinchDistance, GetPalmNormal che ritornano le tre variabili settate in precedenza.

In seguito, abbiamo constatato che è meglio applicare lo Zoom e la Rotation solo attraverso i tasti della tastiera e abbiamo quindi disattivato queste funzionalità negli script "ManageLeft" e "ManageRight".

#### 4.2.2.3 InteractionManager

All'interno dell'HandModel abbiamo creato il GameObject "InteractionManager" inserendo al suo interno lo script "Interaction Manager" e lasciando le impostazioni di default, in seguito all'interno del GameObject appena creato abbiamo inserito i Prefab: "Interaction Hand Left" ed "Interaction Hand Right" entrambe queste Prefab sono trovabili nel progetto in "Assets/Plugins/LeapMotion/Modules/InteractionEngine /Prefabs/Interaction Controllers".

A questo punto è possibile modificare lo script "ShowMenu" aggiungendo come parametri in entrata "Hands" e "TelaDisegnabile", nel metodo Start dello script questi due GameObject sono stati impostati come non attivi (in questo modo non risultano visibili in contemporanea al menù iniziale).

Inoltre, si può aggiungere al GameObject "ActionController" i seguenti script:

- "StartDrawing" che serve a disattivare i menù iniziali (Start\_Page e impostazioni) e, a mostrare mani e tela quando si pigia il bottone "Continua" oppure il bottone "FoglioEsistente".
- "CreateFile"
- "GetFile"

#### 4.2.3 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.

#### 4.2.4 HandReset e HandFinish

Sia "ManageLeft" che "ManageRight" devono implementare "HandTransitionBehavior", a questo punto bisogna importare i due seguenti metodi:

- "HandReset" viene richiamato all'inserimento della mano nella scena
- "HandFinish" viene richiamato all'uscita della mano dalla scena

Questi due metodi notificano se le mani vengono inserite oppure se vengono levate dalla scena.

## 4.3 Disegno

### 4.3.1 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.



Figura 13 - Azione di disegno

### 4.3.2 Selezione colore

#### 4.3.2.1 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.

#### 4.3.2.2 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.



Figura 14 - 2D Color Picker

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.



Figura 15 - 3D Color Picker

#### 4.3.2.3 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<sup>7</sup> 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);`

<sup>7</sup> RGBA Red Green Blue Alpha

### 4.3.3 Selezione strumento e dimensione della penna

La selezione dello strumento e della dimensione della penna è eseguita tramite un pannello composto da tre bottoni ed uno slider. Selezionando uno dei tre bottoni è possibile cambiare lo strumento (penna, gomma o filler), purtroppo abbiamo dovuto disattivare la funzione filler in quanto rallentava troppo l'esecuzione del programma.

Selezionando la penna lo script "ManageRight" imposta il colore selezionato dal Color Picker come colore dei pixel da modificare (vedi capitolo sull'azione disegno), mentre selezionando la gomma il colore diventa bianco così da dare l'impressione di cancellare il disegno.

Attraverso lo slider invece si modifica la grandezza del punto che si andrà a disegnare (quantità di pixel).

La seguente immagine mostra il pannello, il primo bottone è il bottone penna, il secondo la gomma, mentre il terzo è il filler.



Figura 16 - Pannello Selezione Strumento

#### 4.3.3.1 Visualizzazione dello strumento in uso

Per prima cosa bisogna creare un Canvas nella scena, questo Canvas ha "RenderMode" in modalità "MainCamera", in questo modo pur cambiando la dimensione dello schermo lo strumento selezionato sarà sempre in alto a sinistra.

Nel canvas bisogna in seguito creare tre oggetti immagini (gomma, penna e riempi) le quali mostrano ognuna uno strumento (le immagini sono le stesse che vengono mostrate nel pannello di selezione dello strumento visto in precedenza), ogni qualvolta viene selezionato uno strumento diverso dal precedente verrà mostrato solamente l'oggetto immagine dell'oggetto in uso. In seguito, è stato aggiunto un altro oggetto "RawImage" alla quale viene cambiato il colore secondo il colore del pennello attualmente in uso.



#### 4.4 Azioni varie ed inventario

Come annunciato nei precedenti capitoli l'inventario consiste in un pannello di bottoni attaccati alla mano. Ogni bottone ha uno script dedicato, il quale ascolta quando il bottone viene premuto ("isPressed") dalla mano destra, ogni bottone ha un diverso scopo. Per fare in modo che i bottoni funzionano bisogna assicurarsi prima che nell'Interaction Hand della mano destra sia attivo il "Contact Enabled".

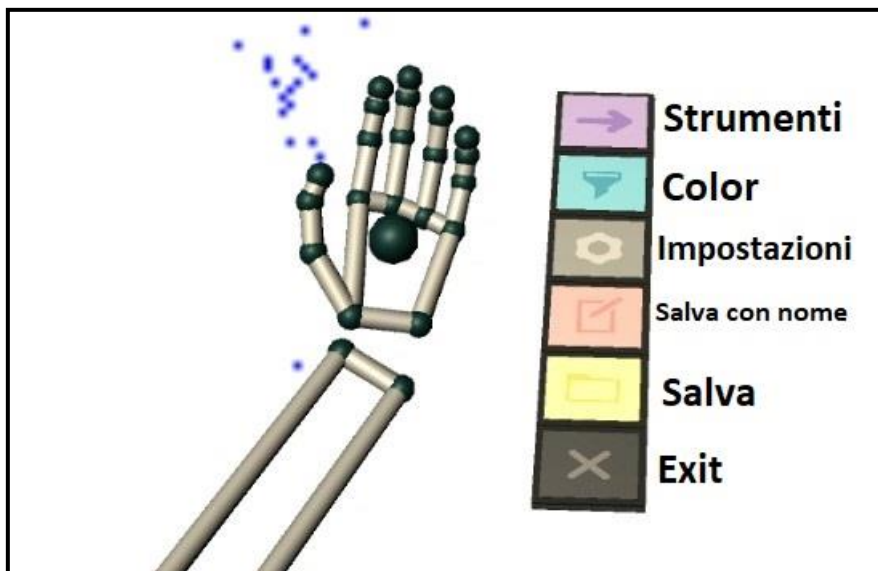


Figura 17 - Pannello Inventario

##### 4.4.1 Selezione strumento

Selezionando il bottone per la selezione dello strumento verrà visualizzato pannello per la scelta dello strumento corrente e la scelta della grandezza della penna, vedi capitoli precedenti. Nel caso in cui il bottone sia nuovamente cliccato il pannello scomparirà applicando le impostazioni settate.

##### 4.4.2 ColorPicker

Selezionando il bottone per la selezione dei colori verrà visualizzato il Color Picker descritto in precedenza, se il bottone viene cliccato nuovamente il Color Picker scomparirà e verrà impostato il colore scelto.

#### 4.4.3 Exit

Cliccando questo bottone viene visualizzata la GUI di uscita tramite la quale si può scegliere se uscire dal programma e salvare oppure non salvare il disegno in corso, in alternativa si può annullare l'azione di uscita. Lo script "Close" è stato quindi creato ed assegnato all'ActionController. Se il bottone "si" (che indica se si vuole salvare e uscire) viene cliccato viene richiamato il metodo "salvaEdEsci" il quale salva la texture (tela) e chiude il programma. Se invece viene cliccato il bottone "no" (non salvare, ma uscire ugualmente) il programma viene chiuso. Infine, se viene cliccato il tasto "Annulla", le mani vengono nuovamente mostrate e l'UI non viene più visualizzata.



Figura 18 - Exit GUI

L'uscita dal programma viene eseguita con "Application.Quit()".

Non è possibile testare l'uscita dal programma (Application.Quit()) se non è ancora stata eseguita una build del programma.

#### 4.4.4 Salvataggio

Il salvataggio rapido avviene tramite il metodo salva() richiamato al momento del clic del bottone dedicato. 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 cosa 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 stessa estensione, visto che i metodi di salvataggio per jpg e quello per png sono due metodi distinti.

#### 4.4.5 Salvataggio con nome

Il salvataggio con nome avviene appena viene premuto l'apposito bottone, 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.

#### 4.4.6 Impostazioni

Il bottone impostazioni consente di mostrare nuovamente il la GUI “Configurazione Tela” tramite la quale è possibile ridimensionare il disegno.

#### 4.4.7 Ctrl + Z

Il ctrl Z o Undo funziona tramite una lista grande fino a 15 texture su cui vengono salvate e istanziate le texture dopo ogni modifica. Una volta che viene salvata la 16 la 1 viene eliminata facendo sì 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.

#### 4.4.8 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 finché non si fosse incontrato un pixel di colore diverso.

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

#### 4.4.9 Rotazione della tela

Per gestire la rotazione della tela è stato creato un nuovo script chiamato "Rotate". All'interno di questa tela ho messo 3 metodi pubblici che ruotano la tela cambiando la rotazione con "tela.transform.Rotate(new Vector3(0, {rotazione}, 0));"

La tela è ruotabile anche premendo i bottoni Right e Left Arrow della tastiera e la barra spaziatrice (resetta la rotazione).

Questo script è stato aggiunto all'ActionController.

#### 4.4.10 Zoom della tela

Per gestire lo Zoom della tela abbiamo creato un nuovo script chiamato "Zoom". All'interno di questa tela sono stati inseriti tre metodi pubblici che avvicinano/allontanano la tela dalla telecamera così da ottenere un effetto di zoom. Questo script è stato aggiunto all'ActionController.

#### 4.4.11 Info Panel

Abbiamo creato un pannello che mostra i comandi principali per l'utilizzo dell'applicazione, questo menù è visibile selezionando il tasto "i" e diventa invisibile utilizzando la stessa modalità.

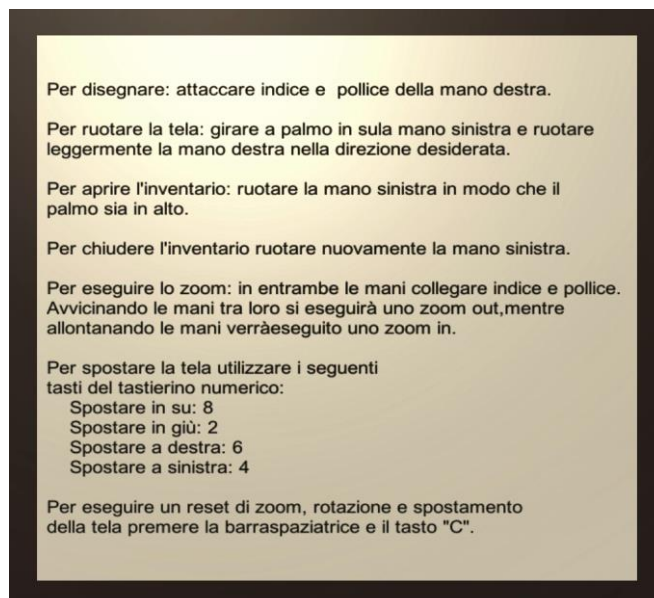


Figura 19 - Info Panel

## 5 Test

### 5.1 Protocollo di test

Test Case:	TC-001	Nome:	Tracciare le mani
Riferimento:	REQ-001		
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:	TC-002	Nome:	Cambio strumento
Riferimento:	REQ-002		
Descrizione:	Controllare che il cambio dello strumento avvenga correttamente.		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion		
Procedura:	Punto 1. Disegnare una riga. Punto 2. Aprire il pannello gestione strumenti e selezionare la gomma. Punto 3. Cancellare la riga.		
Risultati attesi:	Lo strumento selezionato passa da matita (penna) a gomma, e la linea inizialmente disegnata è cancellata.		

Test Case:	TC-003	Nome:	Cambio colore di disegno
Riferimento:	REQ-003		
Descrizione:	Controllare che il cambio del colore avvenga correttamente.		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion		
Procedura:	<p>Punto 1. Tracciare una linea con il colore di default d'avvio.</p> <p>Punto 2. Aprire il menu girando il braccio sinistro.</p> <p>Punto 3. Selezionare con l'indice della mano destra il pulsante per aprire la tavolozza (secondo dall'alto)</p> <p>Punto 4. Cambiare le componenti di rosso, verde, blu e trasparenza trascinando gli slider.</p> <p>Punto 5. Rischiacciare il pulsante del punto 3 e disegnare una linea</p> <p>Punto 6. Verificare che colore selezionato e quello della linea sia lo stesso.</p> <p>Punto 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.</p> <p>Punto 8. Verificare che il colore sia quello nuovo.</p> <p>Punto 9. Riaprire la tavolozza e selezionare il primo colore utilizzato nel punto 6.</p> <p>Punto 10. Disegnare una linea e verificare che il colore sia effettivamente quello usato ne punto 6.</p>		
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:	TC-004	Nome:	Disegno a mano libera
Riferimento:	REQ-004		
Descrizione:	Controllare la possibilità di disegnare a mano libera.		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion con i suoi driver e software.		
Procedura:	Punto 1. Con la mano destra unire le dita e muoverla per tracciare una linea.		
Risultati attesi:	Una linea viene disegnata.		

Test Case:	TC-005.1	Nome:	Salvataggio disegno
Riferimento:	REQ-005.1		
Descrizione:	Controllare che il disegno venga salvato correttamente		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion con i suoi driver e software.		
Procedura:	Punto 1. Modificare il disegno disegnandoci. Punto 2. Aprire il menu girando il braccio sinistro Punto 3. Selezionare con l'indice della mano destra il pulsante per salvare (quarto dall'alto) Punto 4. Verificare che il file sia stato salvato correttamente nel percorso scelto in precedenza		
Risultati attesi:	Un file JPG o PNG contenente il disegno è presente nel percorso corretto		

Test Case:	TC-005.2	Nome:	Salvataggio con nome
Riferimento:	REQ-005.2		
Descrizione:	Controllare che il disegno venga salvato correttamente nel giusto percorso		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion con i suoi driver e software.		
Procedura:	Punto 1. Modificare il disegno disegnandoci. Punto 2. Aprire il menu girando il braccio sinistro Punto 3. Selezionare con l'indice della mano destra il pulsante per salvare con nome (quinto dall'alto) Punto 4. Verificare che il file sia stato salvato correttamente nel percorso scelto.		
Risultati attesi:	Un file JPG o PNG contenente il disegno è presente nel percorso corretto		

Test Case:	TC-007	Nome:	Importare un'immagine
Riferimento:	REQ-007		
Descrizione:	Verificare che l'importazione di un'immagine funzioni e che sia disegnabile		
Prerequisiti:	Il programma Virtual Painter ed il sensore Leap Motion con i suoi driver e software. con i suoi driver e software		
Procedura:	Punto 1. All'avvio del programma selezionare "Apri un foglio esistente" (Il pulsante a destra) Punto 2. Selezionare un file JPG o PNG Punto 3. Modificare l'immagine disegnandoci		
Risultati attesi:	L'immagine viene caricata correttamente sul piano ed è possibile disegnarci		

Test Case:	TC-008	Nome:	Applicativo standalone.
Riferimento:	REQ-006		
Descrizione:	Controllare la possibilità eseguire l'applicazione al di fuori dell'ambiente di sviluppo		
Prerequisiti:	Una build del programma Virtual Painter ed il sensore Leap Motion con i suoi driver e software.		
Procedura:	Punto 1. Prendere la build dell'applicazione e avviarla sul pc.		
Risultati attesi:	L'applicazione si avvia e funziona senza problemi.		

## 5.2 Risultati test

### 5.2.1 Tracciare le mani – TC-001

Per testare che il sensore riproduca i movimenti delle mani basta attaccare il Leap Motion al PC, avviare il programma e infine muovere le mani sopra il sensore verificando che quelle a schermo riproducano gli stessi movimenti. In questo caso i movimenti eseguiti sono stati di apertura e chiusura della mano.

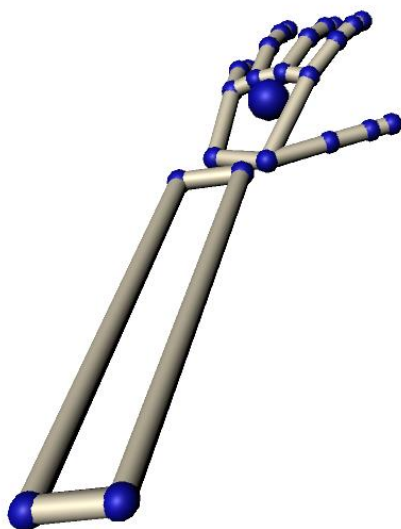


Figura 21 - Mano aperta

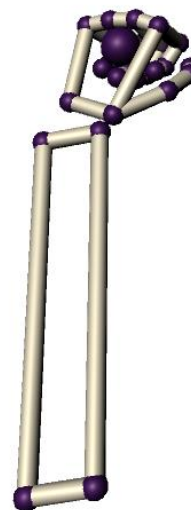


Figura 20 - Mano chiusa



### 5.2.2 Cambio Strumento – TC-002

Per verificare la possibilità di cambiare strumento (in questo caso da penna a gomma) per prima cosa si traccia una riga sulla tela. In seguito, si gira la mano sinistra facendo apparire l'inventario e si preme il tasto viola raffigurante una freccia. Si aprirà il menù per la selezione degli strumenti, dove si sceglierà lo strumento "gomma".

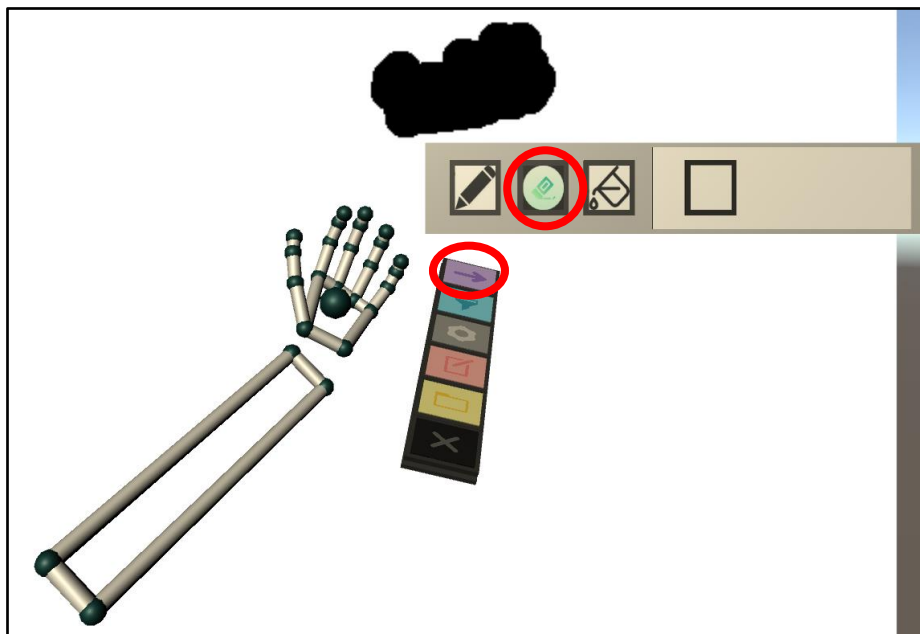


Figura 22 - Selezione strumenti

Dopo aver selezionato la gomma si prova a cancellare parte della riga.

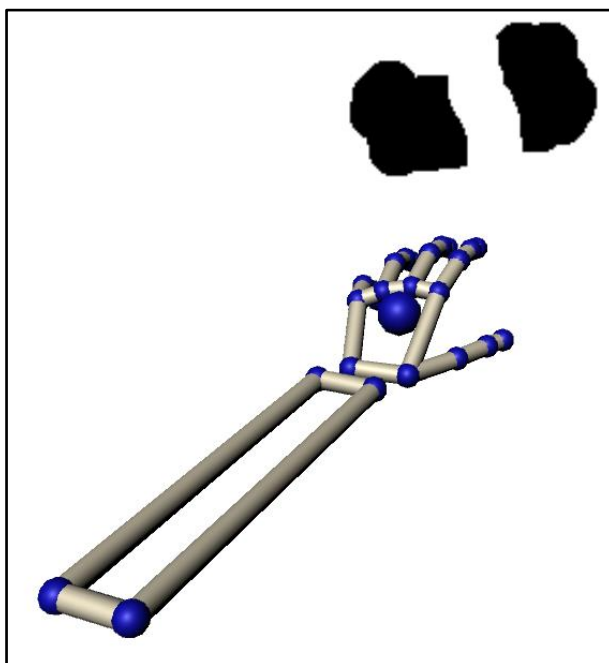


Figura 23 - Strumento gomma

Se parte della riga viene cancellata il test ha esito positivo.

### 5.2.3 Cambio colore di disegno – TC-003

Per verificare la possibilità di poter cambiare il colore con il quale disegnare si prova a disegnare usando il colore di default (Nero). In seguito, si gira la mano sinistra facendo apparire l'inventario e si preme il tasto blu raffigurante un imbuto, così facendo si aprirà il Color Picker.

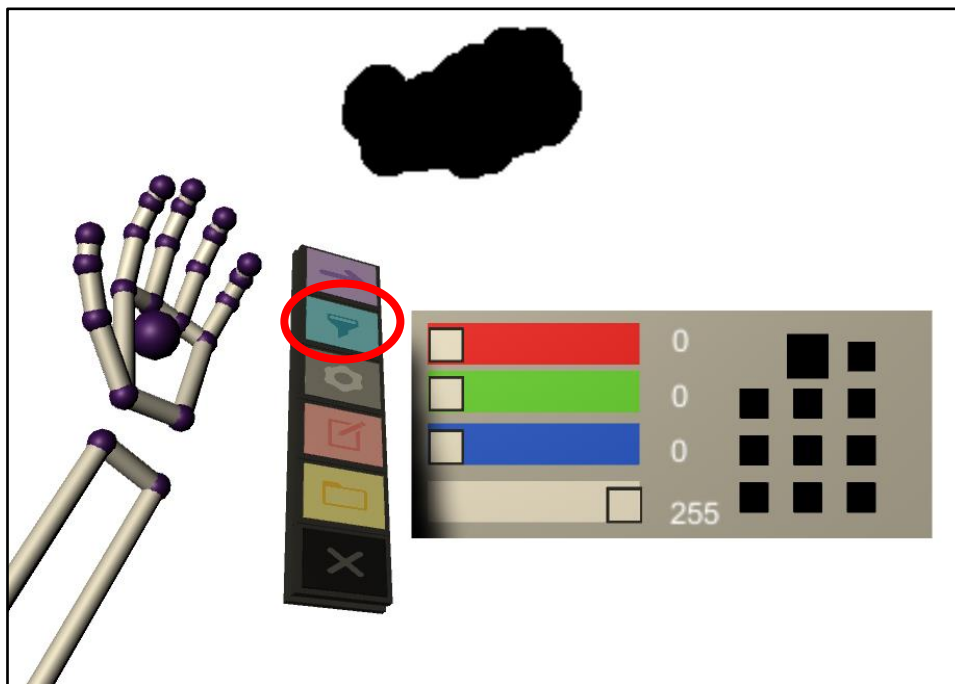


Figura 24 - Color Picker

Il ColorPicker permette di cambiare il colore modificando i valori tramite RGB. È, inoltre, possibile settare dei colori predefiniti premendo su uno dei quadrati sulla destra e cambiando il colore. In questo caso sono stati impostati i primi due slot sul rosso e sul blu (Lo slot selezionato è quello più grande).

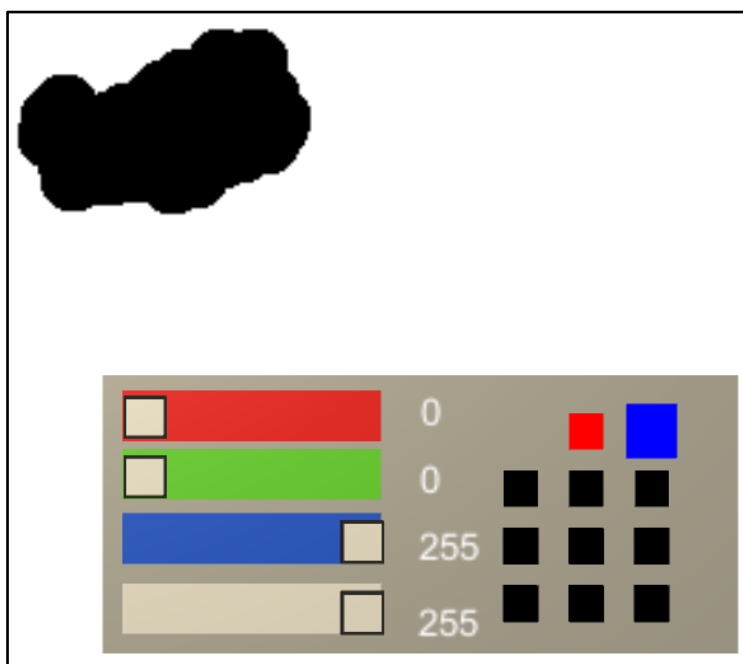


Figura 25 - Colori predefiniti

Ora che il blu è selezionato si prova a disegnare.

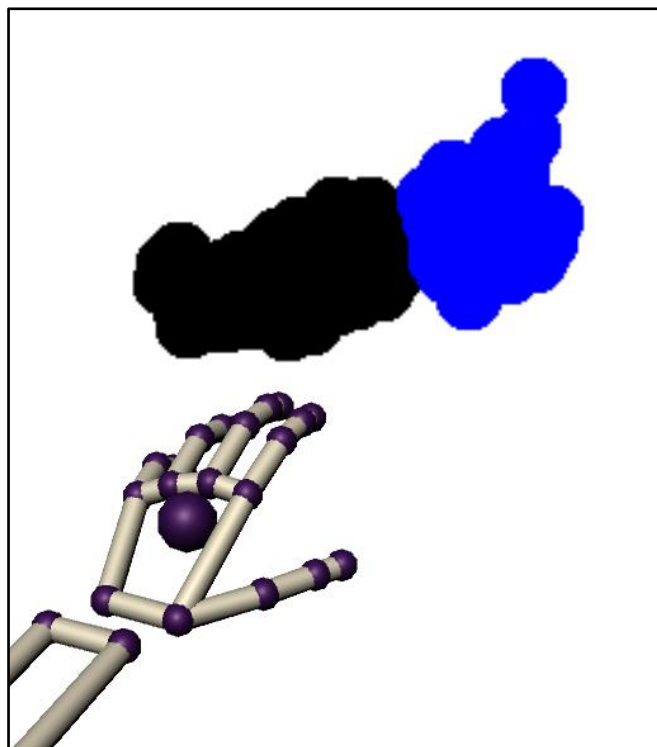


Figura 26 – Colore predefinito blu

Se il colore delle strisce è diventato blu il test ha esito positivo. Per verificare anche il funzionamento dei colori predefiniti si riapre il Color Picker e si seleziona il colore rosso.

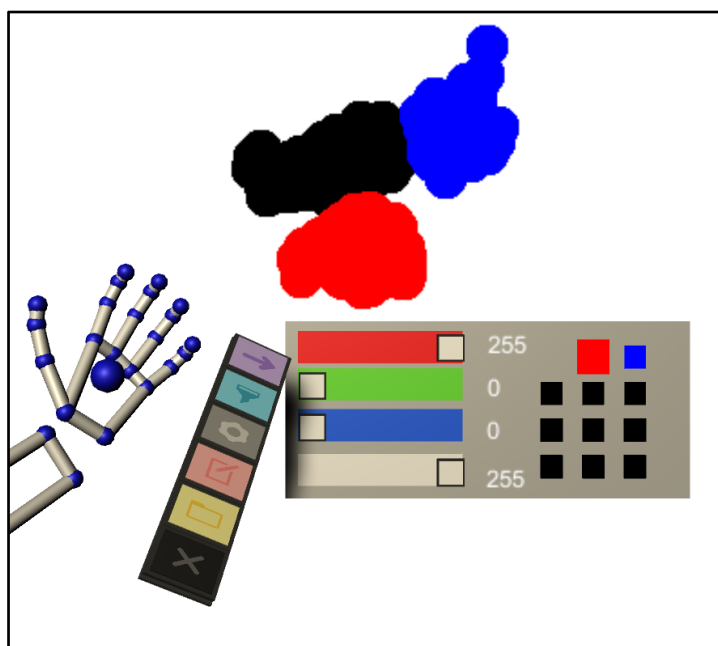


Figura 27 - Colore predefinito rosso

Se il colore delle strisce disegnate è diventato rosso il test ha esito positivo.

#### 5.2.4 Disegno a mano libera – TC-004

Per testare l'azione di disegno la mano deve assumere la giusta posizione, ovvero l'indice e il pollice devono toccarsi (pinching).

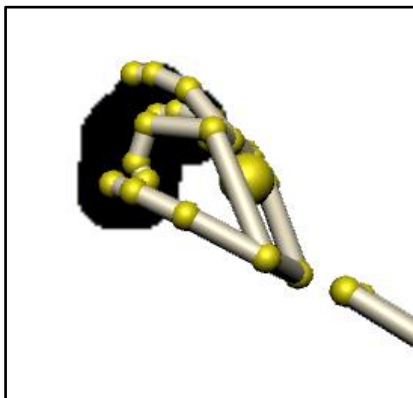


Figura 28 - Azione disegno

Di seguito si prova a muovere la mano tenendo la mano nella stessa posizione, tracciando delle linee.

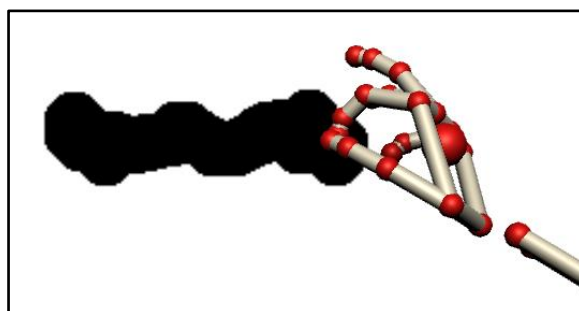


Figura 29 - Linea disegnata

Se la linea segue i movimenti della mano il test ha esito positivo.

#### 5.2.5 Salvataggio disegno – TC-005.1

Per testare la possibilità di salvare le immagini bisogna ricordarsi il percorso in cui è stata creata o importata l'immagine. L'immagine che si andrà a modificare è stata creata al momento all'avvio del programma e si chiama "MySaveFile.png" (nome di default).

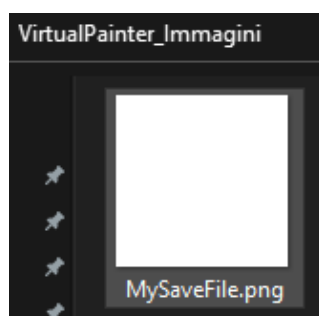


Figura 30 - File prima

Ora l'immagine è bianca. Perciò si andrà a modificarla e salvarla. Per poter salvare l'immagine si apre l'inventario e si preme il pulsante rosa raffigurante un foglio e una penna.

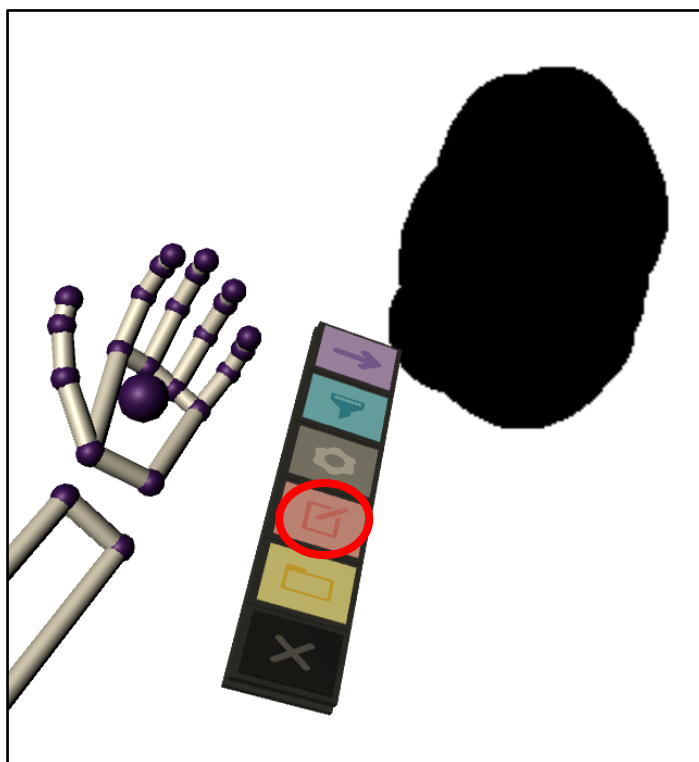


Figura 31 – Salva

Ora si va a vedere il file nel percorso e si verifica che sia stato sovrascritto.

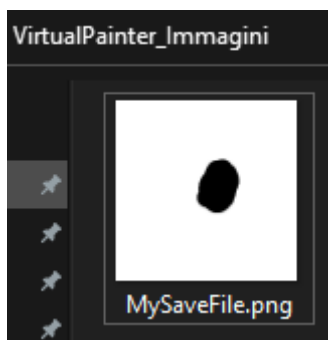


Figura 32 - File dopo

Se l'immagine presenta gli stessi disegni effettuati sul programma il test ha esito positivo.

### 5.2.6 Salvataggio con nome – TC-005.2

Per testare la possibilità di salvare con nome i procedimenti sono presso che gli stessi del test precedente, a parte che invece di premere il pulsante rosa si premerà il pulsante giallo raffigurante una cartella.

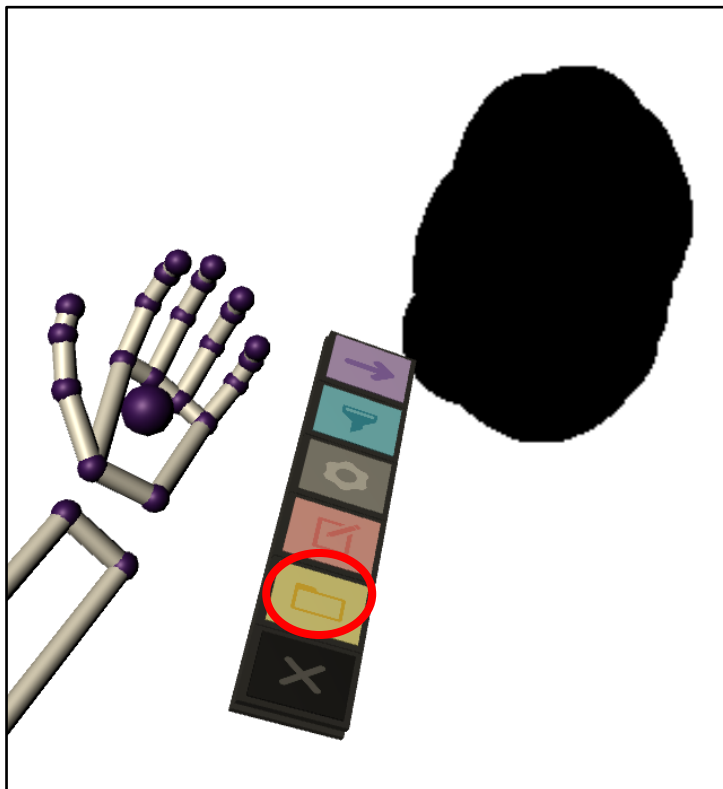


Figura 33 - Salva con nome

Ora si aprirà l'Explorer e chiederà dove salvare il nuovo file. In questo caso verrà salvata nella stessa cartella ma con un nome differente (NuovaImmagine.png).

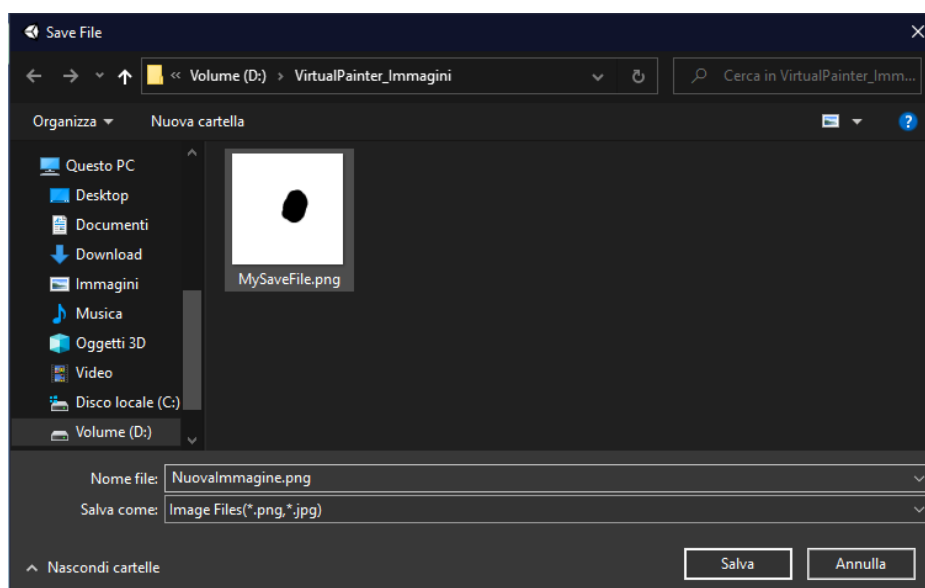


Figura 34 - Nuovo file

Ora si va a verificare che nella che il file sia presente con le modifiche corrette.

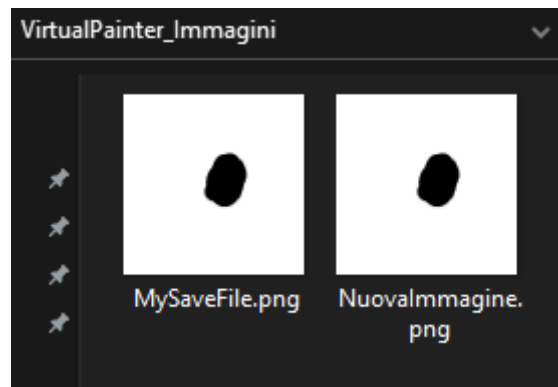


Figura 35 - File salvato

Se il file viene create correttamente il test ha esito positivo.

### 5.2.7 Importare un'immagine – TC-006

Per verificare la possibilità di poter importare un'immagine, all'avvio del programma, sul menu principale, si seleziona il pulsante "Apri un foglio esistente".

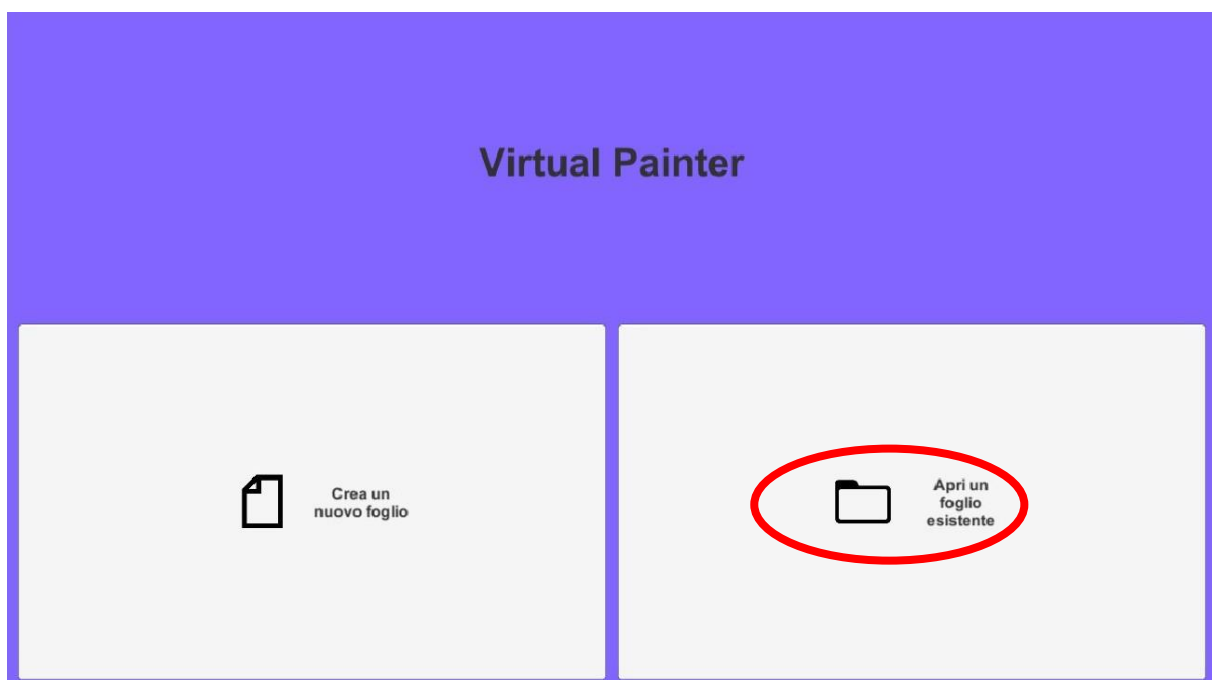


Figura 36 - Importare file

Ciò aprirà l'Explorer, consentendo di scegliere quale immagine importare.

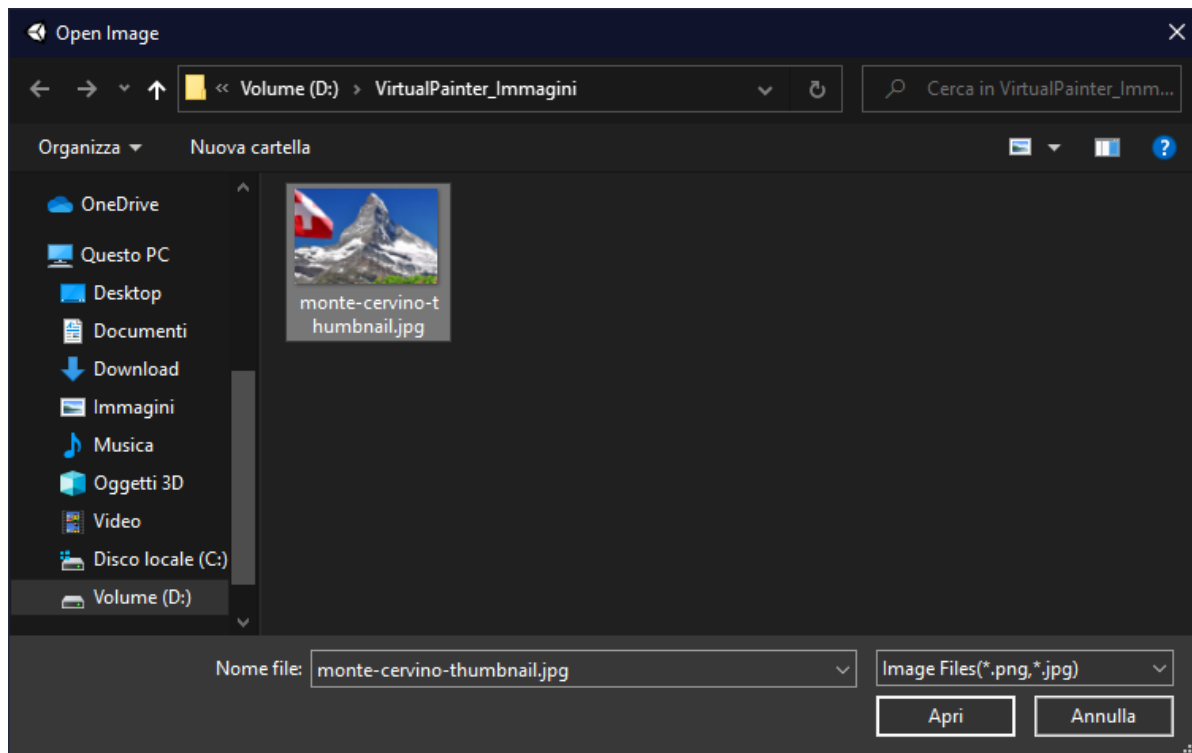


Figura 37 - File da importare

Dopo aver scelto l'immagine da importare essa sarà applicata sulla tela in questo modo.



Figura 38 - File importato



Oltre ad essere applicabile deve anche essere modificabile, perciò bisognerà provare a disegnarci sopra.



Figura 39 - File importato modificato

Se si riesce a disegnare sull'immagine importata senza problemi il test darà esito positivo.

### 5.2.8 Applicativo stand alone - TC-007

L'applicativo deve funzionare anche fuori dall'ambiente di sviluppo, perciò si prova a eseguire una build del progetto, far partire l'eseguibile e testarne il corretto funzionamento. Come prerequisito per il corretto funzionamento del programma devono essere presenti il sensore Leap Motion e i driver che ne permettano il corretto funzionamento. Se l'eseguibile consente l'esecuzione di tutte le azioni testate in precedenza il test dà esito positivo.

### 5.2.9 Tabella dei risultati

Questa tabella riassume l'esito dei test eseguiti con eventuali commenti.

Test Case ID	Nome	Risultato
TC-001	Tracciare le mani	Riuscito
TC-002	Cambio strumento	Riuscito
TC-003	Cambio colore di disegno	Riuscito
TC-004	Disegno a mano libera	Riuscito
TC-005.1	Salvataggio disegno	Riuscito
TC-005.2	Salvataggio con nome	Riuscito, ma apre due volte l'Explorer per salvare la stessa immagine siccome il pulsante rimane attivo più del dovuto.
TC-006	Importare un'immagine	Riuscito
TC-007	Applicativo stand alone	Riuscito

### 5.3 Mancanze/limitazioni conosciute

Alla fine del nostro progetto possiamo constatare la mancanza della gestione della mano dominante dell'utente in quanto il tempo a disposizione è stato sfruttato al miglioramento del funzionamento del programma e al risolvimento di problemi riscontrati durante la fase d'implementazione. Le funzioni zoom, rotazione della tela e movimento della tela (spostamento) sono state disattivate in quanto durante la modifica della posizione della tela quando si disegnava il disegno usciva non centrato (ad esempio se si disegnava in alto a destra a volte il disegno appariva in basso a sinistra). In quanto la tela è un oggetto di tipo texture che si applica su un piano, quando si va a disegnare ai lati della tela il disegno continua nella parte opposta (effetto Pac-Man).

## 6 Consuntivo

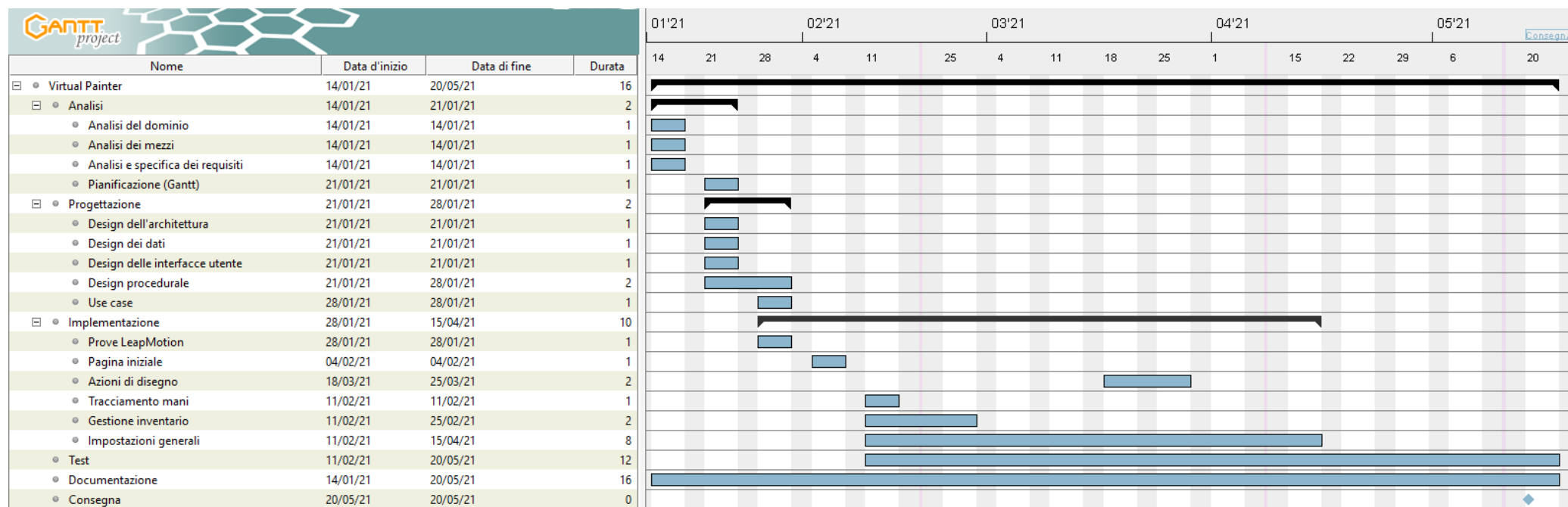


Figura 40 - Gantt consuntivo

## 6.1 Analisi

La fase di analisi è stata eseguita nei tempi definiti nel Gantt preventivo.

## 6.2 Progettazione

Anche la fase di progettazione è stata eseguita nei tempi previsti.

## 6.3 Implementazione

L'attività "Prove LeapMotion" è stata eseguita nei tempi previsti.

La durata dell'attività "Pagina iniziale" è durata una settimana in meno del previsto, ci siamo resi conto che la sua implementazione è semplice.

Abbiamo iniziato l'attività "Azione disegno" all'incirca un mese dopo di quanto previsto in precedenza, siccome per implementare la parte di disegno era necessario creare altre parti del progetto, come ad esempio il tracciamento delle mani.

Il Tracciamento delle mani, la gestione dell'inventario e le impostazioni generali sono iniziate, al contrario del Gantt preventivo, in contemporanea in quanto il gruppo si è spartito i compiti in modo differente da quanto previsto, in questo modo abbiamo potuto lavorare in parallelo riducendo i tempi di implementazione.

## 6.4 Test

Abbiamo iniziato a eseguire i test dopo aver creato la pagina iniziale, siccome abbiamo iniziato a creare la parte principale del progetto a partire dal tracciamento delle mani.

## 6.5 Documentazione

La documentazione è stata aggiornata in parallelo al progetto.

## 6.6 Consegna

La consegna che prima era in data da definire è stata in seguito fissata per il 20 maggio 2021.

## 7 Conclusioni

Questo progetto si può considerare un discreto successo. Vale a dire che la nostra soluzione risponde a tutti i requisiti. Tuttavia considerate le alternative già esistenti, in essa c'è ancora molto da migliorare. Punto di forza del prodotto sono sicuramente i pochi requisiti tecnici necessari all'utilizzo. Infatti altri prodotti richiedono attrezzatura più avanzata come visori di realtà virtuale, mentre il nostro funziona completamente grazie al sensore Leap Motion, mouse e tastiera.

### 7.1 Sviluppi futuri

In futuro sarebbe ideale sistemare le problematiche e mancanze conosciute, migliorare le prestazioni del programma e renderlo più fluido. Inoltre, se si presenterà l'occasione verrà migliorata la grafica del software.

### 7.2 Considerazioni Personali

#### 7.2.1 Sara Bressan

Durante questo lavoro abbiamo imparato le difficoltà e i vantaggi che comporta la collaborazione in un gruppo, inoltre, ho imparato l'importanza di eseguire una buona analisi e una buona progettazione del progetto. Nella fase di implementazione del software ho imparato ad utilizzare meglio il linguaggio di programmazione C# e ad utilizzare Unity, inoltre ho scoperto l'esistenza del Leap Motion e come utilizzarlo, cosa che ho trovato molto interessante.

Mi è piaciuto molto questo progetto e mi sono trovata bene con i miei compagni, nonostante alcuni momenti di sconforto abbiamo lavorato molto bene assieme e abbiamo creato una bella sinergia.

#### 7.2.2 Karim Galliciotti

Aver partecipato a questo progetto mi ha insegnato tante cose, tra cui l'importanza del lavoro di squadra mentre si lavora in gruppo.

Aver partecipato all'implementazione del progetto mi è anche servito per ampliare le mie conoscenze su C#, Unity, ma anche semplicemente sullo svolgimento di un lavoro a gruppi.

Il progetto è stato molto interessante ed eccitante da svolgere, anche coi compagni di lavoro mi sono trovato bene andando d'accordo con loro durante lo svolgimento dell'intero progetto.

#### 7.2.3 Stefano Mureddu

Grazie a questo progetto ho imparato quanto sia più complicato ma interessante lavorare in un gruppo di più di 2 persone, infatti spesso ci siamo dovuti mettere d'accordo su chi modificasse cosa in modo da non eliminare per sbaglio il lavoro di qualcuno, ma allo stesso tempo vedere parti del progetto fatte dagli altri membri funzionare perfettamente con la parte fatta da te da grande soddisfazione.

Altre cose interessanti che ho imparato è l'utilizzo di unity per progetti più grandi, di C# per gli script e, soprattutto, l'utilità di github visto che, lavorando solitamente da solo, non avevo mai veramente utilizzato.

Mi ritengo fortunato visto che con tutti gli altri componenti del gruppo mi sono trovato bene e il progetto essendo interessante è stato divertente da realizzare.

#### 7.2.4 Zeno Darani

Lo sviluppo del progetto ritengo sia andato bene. Anche se a tratti caotico si è sempre riuscito a sistemare tutto. Soprattutto all'inizio ci sono stati un po' di disguidi dovuti all'ambiente di sviluppo unity che anche se offre molte possibilità rimane comunque complesso da utilizzare bene. Sicuramente ho potuto familiarizzare di più con il software. Per quanto riguarda l'utilizzo di C# non c'è stato un vero miglioramento poiché molti degli scripts usavano classi prese dalla libreria del Leap Motion e le parti che ho realizzato non avevano un livello così alto di difficoltà. Al contrario i miei compagni hanno riscontrato problemi più articolati poiché le loro classi riguardavano alla parte di disegno vera e propria. Mi è piaciuto lavorare con loro e ritengo che ci sia stato una buona ed equa suddivisione del lavoro. Inoltre eravamo comunque sempre a conoscenza di quello che ognuno di noi faceva e ci siamo sempre dati una mano se vi erano problemi grossi.

## 8 Bibliografia

### 8.1 Sitografia

- <https://docs.ultraleap.com/>, Documentazione ufficiale Leap Motion Controller, 21-01-2021
- <https://www.youtube.com/watch?v=4LVVpl9tCNE>, Video dimostrativo all'uso del Leap Motion Controller al fine di disegnare, 21-01-2021
- <https://github.com/gkngkc/UnityStandaloneFileBrowser>, Standalone File Browser, 21-01-2021
- <https://www.ultraleap.com/>, Sito ufficiale del Leap Motion Controller, 14-01-2021
- <https://medium.com/@felixnoller/getting-started-with-simple-3d-buttons-using-unity-and-leap-motion-40d9efded317>, Guida alla creazione dei bottoni 3D, 04-02-2021
- <https://assetstore.unity.com/packages/2d/gui/icons/simple-icon-pastel-tone-107568>, Package contenente le immagini che sono state inserite nel progetto al fine di creare bottoni e interfacce grafiche, 04-02-2021
- <https://developer.leapmotion.com/unity>, Sito dalla quale installare gli Unity Module Packages, 04-02-2021
- <https://github.com/gkngkc/UnityStandaloneFileBrowser#~:text=%20Unity%20Standalone%20File%20Browser%20%201%20Works,Ricardo%20Rodrigues.%207%20Basic%20WebGL%20support.%20More>, Unity Standalone File Browser Project, 04-02-2021
- [https://developer-archive.leapmotion.com/documentation/v2/unity/devguide/Leap\\_Gestures.html](https://developer-archive.leapmotion.com/documentation/v2/unity/devguide/Leap_Gestures.html), Guida alla lettura dei movimenti delle mani tramite Leap Motion in Unity, 11-02-2021
- <https://unitycoder.com/blog/2015/11/04/leap-motion-get-finger-position-direction/>, Guida alla lettura dei movimenti delle dita tramite Leap Motion in Unity, 11-02-2021
- [https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap\\_Hand.html](https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Hand.html), Guida di utilizzo delle mani utilizzando il Leap Motion in Unity, 11-02-2021
- <https://leapmotion.github.io/UnityModules/interaction-engine.html>, Guida all'utilizzo del Leap Motion Controller in Unity, 11-02-2021
- <https://www.flaticon.com/search?word=fill>, Icone utilizzate nelle GUI, 25-03-2021
- <https://gist.github.com/ProGM/22a615b812c5a9f1183d43b536d14a42>, Codice per la creazione del filler, 25-03-2021
- <https://stackoverflow.com/questions/3944552/undo-for-a-paint-program>, CTRL + Z, 01-04-2021
- <https://docs.unity3d.com/Packages/com.unity.localization@0.4/manual/index.html>, Guida all'utilizzo del pacchetto Localization utilizzato per gestire più lingue, 01-04-2021
- <https://forum.unity.com/threads/localizing-strings-on-script.847000/>, Aiuto per utilizzare il pacchetto Localization, 01-04-2021
- <https://github.com/intuiface/LeapIA>, Leap Motion Interface Assets foe IntuiFace, 15-04-2021

## 9 Indice delle figure

Figura 1 - Use Case	6
Figura 2 - Gantt preventivo	7
Figura 3 - Design dell'architettura del sistema	10
Figura 4 - Gestione del menu	11
Figura 5 - Design impostazioni	12
Figura 6 - Design apertura area di disegno	13
Figura 7 - Design chiusura del programma	14
Figura 8 - Design delle interfacce	15
Figura 9 - Start Page GUI	22
Figura 10 - Configurazione Tela GUI	23
Figura 11 - Impostazioni Input Field	24
Figura 12 - Mani e tela	26
Figura 13 - Azione di disegno	30
Figura 14 - 2D Color Picker	31
Figura 15 - 3D Color Picker	31
Figura 16 - Pannello Selezione Strumento	32
Figura 17 - Pannello Inventario	33
Figura 18 - Exit GUI	34
Figura 19 - Info Panel	36
Figura 20 - Mano chiusa	40
Figura 21 - Mano aperta	40
Figura 22 - Selezione strumenti	41
Figura 23 - Strumento gomma	41
Figura 24 - Color Picker	42
Figura 25 - Colori predefiniti	42
Figura 26 – Colore predefinito blu	43
Figura 27 - Colore predefinito rosso	43
Figura 28 - Azione disegno	44
Figura 29 - Linea disegnata	44
Figura 30 - File prima	44
Figura 31 – Salva	45
Figura 32 - File dopo	45
Figura 33 - Salva con nome	46
Figura 34 - Nuovo file	46
Figura 35 - File salvato	47
Figura 36 - Importare file	47
Figura 37 - File da importare	48
Figura 38 - File importato	48
Figura 39 - File importato modificato	49
Figura 40 - Gantt consuntivo	51

## 10 Allegati

- Diari di lavoro
- Codici sorgente
- Progetto in Unity
- QdC
- Abstract
- Guida utente
- Documentazione classi (html)



## 11 Indice analitico

### A

ActionController · 21; 29; 35; 36  
Analisi · 9  
applicazione · 15; 37; 41

### B

bottone · 22; 23; 24; 25; 29; 33; 34; 35; 36  
bottoni · 15; 21; 22; 28; 33; 34; 36; 55

### C

Camera · 22; 25; 28  
Canvas · 21; 22; 23; 33  
    Canvas · 22  
Color Picker · 32; 33; 34  
colore · 5  
    colori · 12; 15; 31; 32; 33; 34; 36; 39

### D

Design · 10; 12; 13; 14; 15; 16; 22  
dipingere  
    disegnare · 31  
disegnare · 4; 5; 13; 15; 22; 31; 32; 33; 36; 39; 51; 55  
disegni · 4; 5  
disegno · 4; 11; 13; 23; 31; 33; 35; 36; 39; 40; 51  
Disegno  
    disegno · 31; 39

### E

errore · 10  
Explorer · 12; 24; 25; 35; 47; 49; 50

### F

file · 4; 11; 12; 22; 23; 24; 25; 35; 40  
form · 24; 25

### G

GameObject · 21; 25; 27; 28; 29  
gomma · 33; 38  
GUI · 21; 22; 23; 32; 35; 36; 55

### H

Hand  
    Hands · 27; 28; 29; 34; 55  
HandModel · 29  
Hands · 27; 28; 29  
HandsModel · 27; 28  
Hardware · 9

### I

immagini · 6; 22; 25; 28; 33; 55  
Implementazione · 21  
impostazioni · 12; 13; 14; 21; 22; 28; 29; 34; 36  
input  
    Input Field · 23  
InteractionManager · 28; 29  
interfacce  
    interfaccia · 15; 55  
interfaccia grafica, · 21  
inventario · 28; 34  
isPinching · 31; 36

### J

JPG · 25; 40  
    jpg · 35  
json · 25; 35

### L

Leap Motion · 4; 5; 9; 10; 21; 26; 27; 28; 29; 32; 38; 39; 40;  
    41; 55  
*Leap Motion Controller* · 4; 9; 21; 26; 55

### M

MainCamera · 33  
ManageLeft · 28; 29  
Manager · 27; 28; 29  
ManageRight · 29; 33  
mani · 4; 5; 15; 21; 25; 26; 27; 28; 29; 35; 38; 55  
mano · 4; 5; 12; 13; 15; 28; 29; 31; 34; 36; 39; 40; 51  
matita · 38  
menu · 10; 11; 15; 22; 24; 25; 39; 40  
Model · 27





---

**O**

oggetto · 21; 24; 29; 33; 51

---

**P**

pacchetti

pacchetto · 26; 27

package · 22; 29

pannello · 22; 23; 28; 33; 34; 37; 38

penna · 33; 34; 38

Pianificazione · 7

Pinching

pinch · 28

pixel · 24; 25; 31; 33; 36

pixels

pixel · 25

plugin · 27

PNG · 25; 40

png · 35

prefab · 27; 28

Progettazione · 10

programma · 5; 10; 14; 21; 22; 24; 26; 27; 33; 35; 38; 39; 40;  
41; 51

pulsante · 39; 40

---

**R**

RGBA · 31; 32

riempimento · 36

---

**S**

salvataggio · 12; 14; 35

Salvataggio

salvataggio · 35; 40

scena

scene · 21; 23; 25; 27; 29; 33

scene · 21

script · 21; 24; 25; 27; 28; 29; 33; 34; 35; 36; 55

slider · 32; 33; 39

Software · 9

*standalone* · 4; 5; 6; 41

strumenti · 5

---

**T**

tela · 15; 21; 22; 23; 24; 25; 26; 29; 31; 35; 36; 39; 51

telecamera · 25; 36

test · 26; 27; 38; 41

texture · 24; 25; 35; 36; 51

tracciamento · 27

---

**U**

UI · 28; 35

*Unity* · 4; 9; 21; 22; 24; 27; 28; 32; 55; 56

Use case · 6

---

**V**

Virtual Painter · 22; 38; 39; 40; 41

---

**W**

*Windows* · 4; 27

---

**Z**

Zoom · 28; 29; 36