

Documentazione Hand Gesture Recognition

Titolo del progetto: Hand Gesture Recognition
Alunno/a: Diego Muniz Pereira
Classe: I3BB
Anno scolastico: 2022/2023
Docente responsabile: Geo Petrini

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
	Background/Situazione iniziale:	3
	Descrizione del problema e motivazione:	3
	Approccio/Metodi:	3
	Risultati:	3
1.3	Scopo	3
	Scopi didattici:	3
	Scopi operativi:	3
2	Analisi	4
2.1	Analisi del dominio	4
2.2	Analisi e specifica dei requisiti	4
	Analisi:	4
	Specifica dei requisiti:	4
2.2.1	Spiegazione elementi tabella dei requisiti:	5
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 delle interfacce	2
4	Implementazione	4
4.1	Lista strumenti	4
4.2	GUI	4
4.3	Cattura immagine webcam	5
4.4	Selezione webcam	7
4.5	Selezione applicazione	7
5	Test	8
5.1	Protocollo di test	8
5.2	Risultati test	8
6	Consuntivo	1
7	Conclusioni	1
7.1	Sviluppi futuri	1
7.2	Considerazioni personali	1
8	Glossario	1
9	Bibliografia	2
9.1	Sitografia	2
10	Allegati	2

1 Introduzione

1.1 Informazioni sul progetto

Questo progetto è supervisionato dal docente Geo Petrini e sviluppato da: Diego Muniz Pereira.
Il progetto è svolto nella scuola SAMT di Canobbio.
Inizio: 09.09.2022 – Fine: 23.12.2022.

1.2 Abstract

Background/Situazione iniziale:

Questo progetto consiste in una applicazione che, tramite il movimento della mano, possa cambiare le slide di una presentazione PowerPoint, LibreOffice Impress, Google Slides, ...

Descrizione del problema e motivazione:

Per questo progetto, saranno necessarie buone conoscenze nella programmazione e manipolazione d'immagini per poter rilevare una mano dentro un'immagine e il suo movimento. Questo programma potrebbe essere molto utile per le persone che devono fare molte presentazioni (es. professori) usando PowerPoint, LibreOffice Impress, Google Slides, ... per presentare in modo più comodo e naturale.

Approccio/Metodi:

Per arrivare alla soluzione finale ho provato vari metodi con svariati livelli di successo per colpa dei problemi di compatibilità di EmguCV e la povera performance.
Prima di arrivare alla mia soluzione attuale ho provato a usare OpenPose per rilevare la mano. Questo approccio si è rilevato non effettivo per il povero frame rate, OpenPose è molto preciso e comodo ma non per del video in tempo reale.

Risultati:

Questo progetto è realizzato con C# App Windows Forms (.NET Framework), utilizza le librerie: EmguCV per il rilevamento della mano (v. 2.2.0.822) e AForge DirectShow (v. 2.2.5.0) per l'ottenimento dell'immagine dalla Webcam. Per rilevare la sagoma della mano ho utilizzato un metodo che consiste nella detezione dell'oggetto più grande del colore della pelle e calcolare l'involuppo convesso, sviluppata da Luca Del Tongo.

1.3 Scopo

Scopi didattici:

- Saper creare una progettazione e rispettarla.
- Saper documentare il lavoro e creare i diari.
- Saper creare Use Case e Diagrammi di Gantt.

Scopi operativi:

- Creare un progetto C# App Windows Forms (.NET Framework).
- Imparare a manipolare e visualizzare immagini da una webcam.
- Imparare a calcolare le coordinate di un punto in costante movimento.

2 Analisi

2.1 Analisi del dominio

Questo progetto consiste in una applicazione che, tramite il movimento della mano, possa cambiare le slide di una presentazione PowerPoint, LibreOffice Impress, Google Slides, ...

Questo progetto è designato per le persone che devono fare molte presentazioni (es. professori) per poter presentare in modo più comodo e naturale grazie all'abilità di cambiare slide con il movimento della mano senza il bisogno di toccare la tastiera.

Il programma è molto semplice da usare grazie alla sua interfaccia semplice e minimalista.

2.2 Analisi e specifica dei requisiti

Analisi:

Software che permette di utilizzare il movimento delle proprie mani per poter cambiare le slide in un PowerPoint in tempo reale grazie a una webcam. Muovo la mano verso destra e passa alla prossima slide, muovo la mano verso sinistra torna indietro.

Specifica dei requisiti:

ID: REQ-01	
Nome	Rilevamento mano
Priorità	1
Versione	1.0
Note	
Sotto requisiti	
001	Controllo presenza webcam
002	Movimento destra
003	Movimento sinistra

ID: REQ-02	
Nome	Funzionamento impostazioni
Priorità	1
Versione	1.0
Note	
Sotto requisiti	
001	Selezione applicazione, lingua
002	Selezione camera, modalità camera

2.2.1 Spiegazione elementi tabella dei requisiti:

ID: identificativo univoco del requisito

Nome: breve descrizione del requisito

Priorità: indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

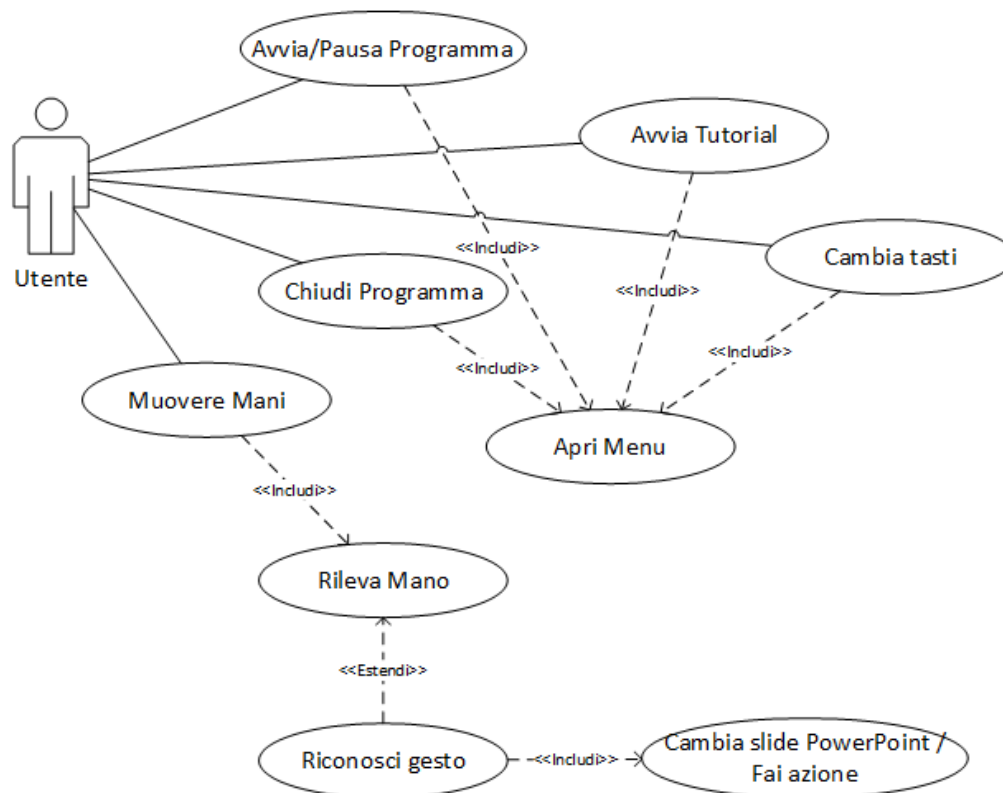
Versione: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l'ultima versione, mentre le vecchie dovranno essere inserite nei diari.

Note: eventuali osservazioni importanti o riferimenti ad altri requisiti.

Sotto requisiti: elementi che compongono il requisito.

2.3 Use case



2.4 Pianificazione

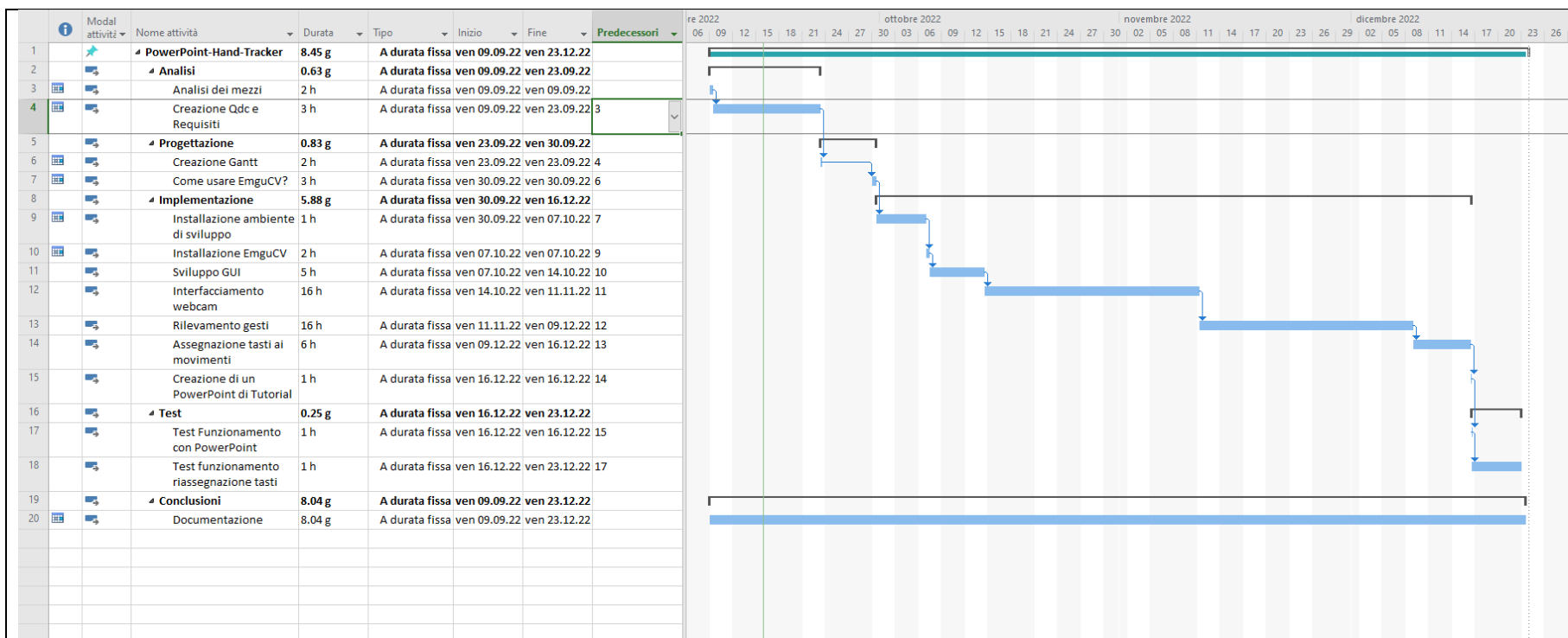


Figura 1: Diagramma di Gantt.

Titolo del progetto: Hand Gesture Recognition
Alunno/a: Diego Muniz Pereira
Classe: I3BB
Anno scolastico: 2022/2023
Docente responsabile: Geo Petrini

2.5 Analisi dei mezzi

2.5.1 Software

Per la creazione del programma ho utilizzato: Visual Studio 2022, .NET Framework 4.7.2, EmguCV (v. 2.2.0.822) e AForge DirectShow (v. 2.2.5.0).

Per la realizzazione dell'intero progetto: Microsoft Project, Microsoft Visio e Microsoft Word.

2.5.2 Hardware

Un PC con Windows 7+ 64bit e una Webcam.

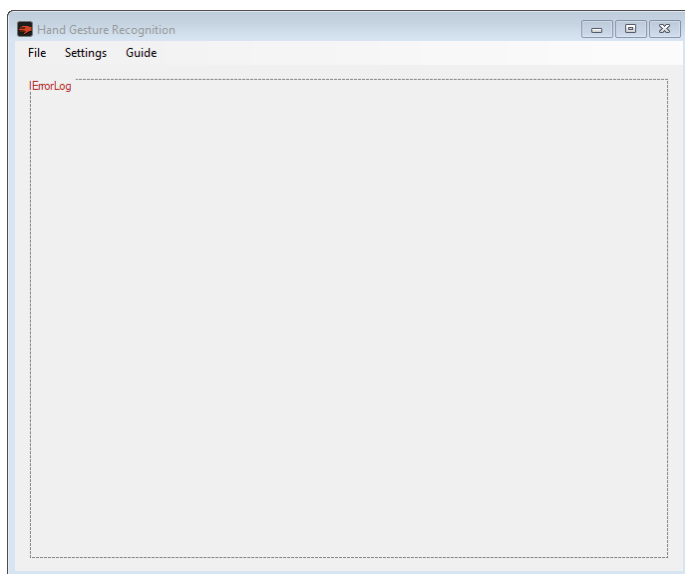
Titolo del progetto:	Hand Gesture Recognition
Alunno/a:	Diego Muniz Pereira
Classe:	I3BB
Anno scolastico:	2022/2023
Docente responsabile:	Geo Petrini

3 Progettazione

3.1 Design dell'architettura del sistema

La mano si muove davanti la webcam, la webcam trasmette l'immagine al programma e il programma fa azioni sul PowerPoint (oppure altro).

3.2 Design delle interfacce



L'UI è semplicemente composta da un MenuStrip dove si trovano i controlli dell'applicazione, un PictureBox dove verrà visualizzato l'immagine e un Label dove compariranno eventuali errori.

Il Menu Strip possiede 4 voci del menu le quali possiedono vari sotto voci

File (File)

Start (Avvia)

Avvia il funzionamento principale del programma (Cattura webcam e rilevamento mano).

Pause (Pausa)

Pausa il funzionamento principale del programma.

Settings (Impostazioni)

Application (Applicazione)

Mostra una lista di tutte le applicazioni aperte dalla quale si può scegliere quale si vuole che venga utilizzata dal programma.

Language (Lingua)

Mostra una lista di tutte le lingue disponibili dalla quale si può scegliere quale utilizzare.

Camera (Camera)

Mostra una lista di tutte le webcam collegate al PC dalla quale si può scegliere quale utilizzare.

Camera Mode (Modalità Camera)

Normal (Normale)

Mostra l'immagine della webcam con disegnato sopra la mano vari contorni.

Skin (Pelle)

Mostra l'immagine della webcam dove i pixel colore della pelle sono diventano bianchi e il resto nero.

Guide (Guida)

Mostra una guida al giusto utilizzo dell'applicazione.

Fps (Fps)

Mostra gli fps della webcam. All'inizio della applicazione è nascosta e viene mostrata solo quando la webcam è avviata.

4 Implementazione

4.1 Lista strumenti

IDE: Visual Studio 2022

Modello progetto: App Windows Forms (.NET Framework)

Librerie:

System.Management

System.Runtime.InteropServices

System.Diagnostics

Emgu.CV

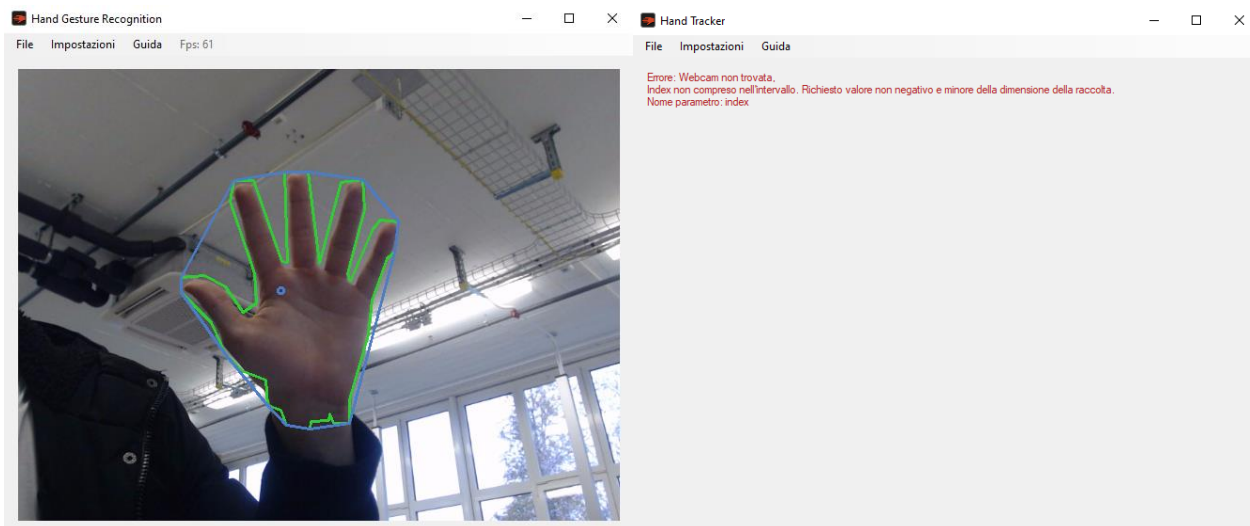
Emgu.CV.Structure

AForge.Video

AForge.Video.DirectShow

HandGestureRecognition.SkinDetector

4.2 GUI



Il progetto è basato sulla “App Windows Forms (.NET Framework)” rendendo la realizzazione della GUI molto più facile e veloce.

La GUI è stata creata con “App Windows Forms (.NET Framework)” grazie al suo facile e comprensibile modo di creare GUI (Forms).

L'applicazione è composta da un Form `fHandGestureRecognition`, un MenuStrip `msMenu` dove si trovano i controlli dell'applicazione, un PictureBox `pbCurrentFrame` dove verrà visualizzato l'immagine e un Label `lErrorLog` dove compariranno eventuali errori.

4.3 Cattura immagine webcam

Per ottenere l'immagine dalla webcam connessa ho usato la libreria `AForge.Video` e `AForge.Video.DirectShow`. Quando clicco il tasto File/Avvia controllo se `capture` è `null` se sì allora creo una variabile di tipo `FilterInfoCollection` chiamata `videoDevices` dove salvo tutte le webcam disponibili, poi salvo la webcam alla posizione `camIndex` dentro la variabile globale `capture` che è di tipo `VideoCaptureDevice`, dopo di che ho assegnato a `capture.NewFrame` il metodo `VideoCaptureDevice_NewFrame`. In fine richiamo `capture.Start()` per avviare la webcam.

```
if (capture == null)
{
    FilterInfoCollection videoDevices =
        new FilterInfoCollection(FilterCategory.VideoInputDevice);
    capture = new VideoCaptureDevice(videoDevices[camIndex].MonikerString);
    capture.NewFrame += new NewFrameEventHandler(VideoCaptureDevice_NewFrame);
}

capture.Start();
```

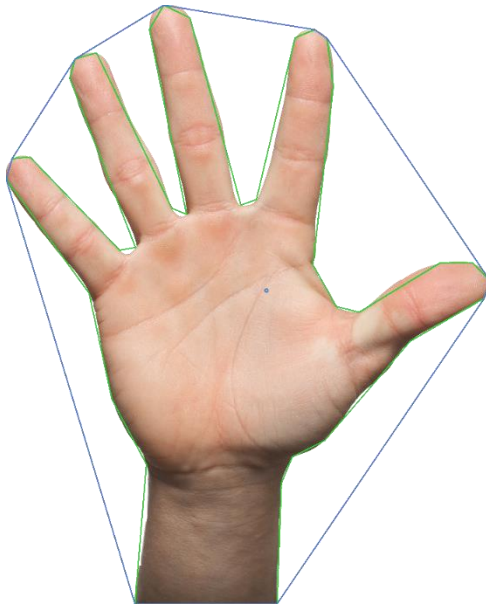
Una volta cominciata la cattura dell'immagine ad ogni frame viene chiamato il metodo `VideoCaptureDevice_NewFrame`. Il metodo salva nella variabile `currentBitmapFrame` il frame in versione `Bitmap` per poi specchiarlo e trasformarlo in `Image<Bgr, Byte>` e salvarlo dentro `currentFrame`. Dentro la variabile `skin` salvo l'immagine dove i pixel che rientrano tra $Y = 0, Cr = 131, Cb = 80$ e $Y = 255, Cr = 185, Cb = 135$ sono bianchi mentre il resto saranno neri. Finalmente passo `skin` al metodo `ExtractContourAndHull` che procederà con disegnare un contorno per l'oggetto di colore bianco più grande e lo disegnerà dentro `currentFrame`. In fine secondo la `cameraMode` (Modalità Camera) selezionata mostrerà sul PictureBox "pbCurrentFrame" `currentFrame/skin`, poi richiamo `Thread.Sleep` e aspetto `waitTime` (quanto ci ha messo a fare questa parte di codice) per non avere problemi con i thread.

```
currentBitmapFrame = (Bitmap)eventArgs.Frame.Clone();
currentBitmapFrame.RotateFlip(RotateFlipType.RotateNoneFlipX);
currentFrame = new Image<Bgr, Byte>(currentBitmapFrame);

Image<Gray, Byte> skin = skinDetector.DetectSkin(
    new Image<Bgr, Byte>(currentBitmapFrame), YCrCb_min, YCrCb_max);

ExtractContourAndHull(skin);
try
{
    if (cameraMode == 0)
        pbCurrentFrame.Image = currentFrame.Bitmap;
    else if (cameraMode == 1)
        pbCurrentFrame.Image = skin.Bitmap;
}
catch { }

Thread.Sleep(waitTime);
```



Alla fine del metodo `VideoCaptureDevice_NewFrame` esso chiama `checkHandPosition` passando `box.center.X` e `box.center.Y` come argomenti che sono le coordinate del punto centrale della mano. Il metodo controlla la attuale posizione della mano e la compara con la previa se supera la percentuale dettata da `CHANGEPERC` (9%) allora, a dipendenza della direzione del movimento della mano, invia un keystroke all'app selezionata (`selectedApp`).

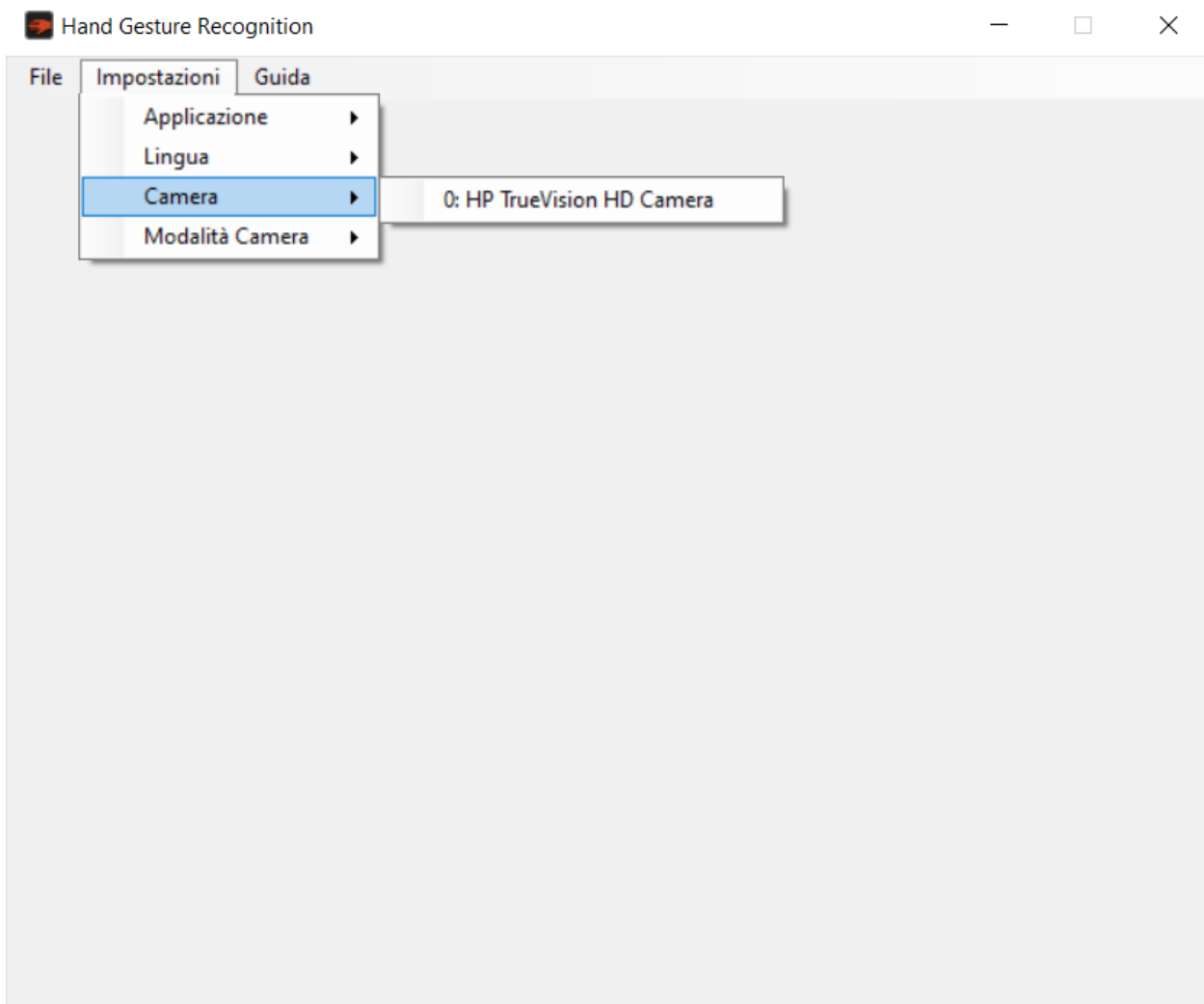
```
if (centerX > previousHandPosition.X) {
    currentPerch = (short)(100 - previousHandPosition.X / centerX * 100);
    if (currentPerch > CHANGEPERC)
        sendKey = 1;
}
else if (centerX < previousHandPosition.X) {
    currentPerch = (short)(100 - centerX / previousHandPosition.X * 100);
    if (currentPerch > CHANGEPERC)
        sendKey = 2;
}

if (previousPerch < CHANGEPERC && selectedApp != null) {
    if (sendKey == 1) {
        SetForegroundWindow(selectedApp.MainWindowHandle);
        SendKeys.SendWait("{RIGHT}");
        SetForegroundWindow(thisApp.MainWindowHandle);
    }
    else if (sendKey == 2) {
        SetForegroundWindow(selectedApp.MainWindowHandle);
        SendKeys.SendWait("{LEFT}");
        SetForegroundWindow(thisApp.MainWindowHandle);
    }
}

previousPerch = currentPerch;
previousHandPosition = new PointF(centerX, centerY);
```

4.4 Selezione webcam

Grazie al metodo `ManagementObjectSearcher` ottengo tutte le webcam connesse al PC, dopo di che le salvo in `miCamera` e le associo al metodo `SetCamIndex` che cambia la webcam selezionata.



4.5 Selezione applicazione

Grazie al metodo `Process.GetProcesses()` ottengo tutte le applicazioni aperte nel PC, dopo di che le salvo in `miApplication` e le associo al metodo `SetApplicationSelected` che cambia l'applicazione selezionata.

5 Test

5.1 Protocollo di test

Test Case:	TC-01	Nome:	Funzionamento webcam
Riferimento:	REQ-01		
Descrizione:	Verificare se il programma riconosce la webcam		
Prerequisiti:			
Procedura:	1. Andare sotto Impostazioni/Camera e scegliere la webcam da utilizzare 2. Andare sotto File e clicca Start		
Risultati attesi:	Si vede l'immagine della webcam.		

Test Case:	TC-02	Nome:	Rilevamento mano
Riferimento:	REQ-01		
Descrizione:	Verificare se il programma riconosce la webcam		
Prerequisiti:			
Procedura:	1. Andare sotto Impostazioni/Application e scegliere l'applicazione desiderata 2. Avviare la webcam (TC-01) 3. Muovere la mano a destra e sinistra davanti alla webcam		
Risultati attesi:	L'applicazione selezionata reagisce al movimento della mano.		

5.2 Risultati test

Test Case	Funzionamento	Commento	Data
TC-01	OK	Viene visualizzato l'immagine della webcam correttamente.	16.12.2022
TC-02	OK	Il programma selezionato reagisce al movimento della mano (es. PowerPoint cambia slide).	16.12.2022

6 Consuntivo

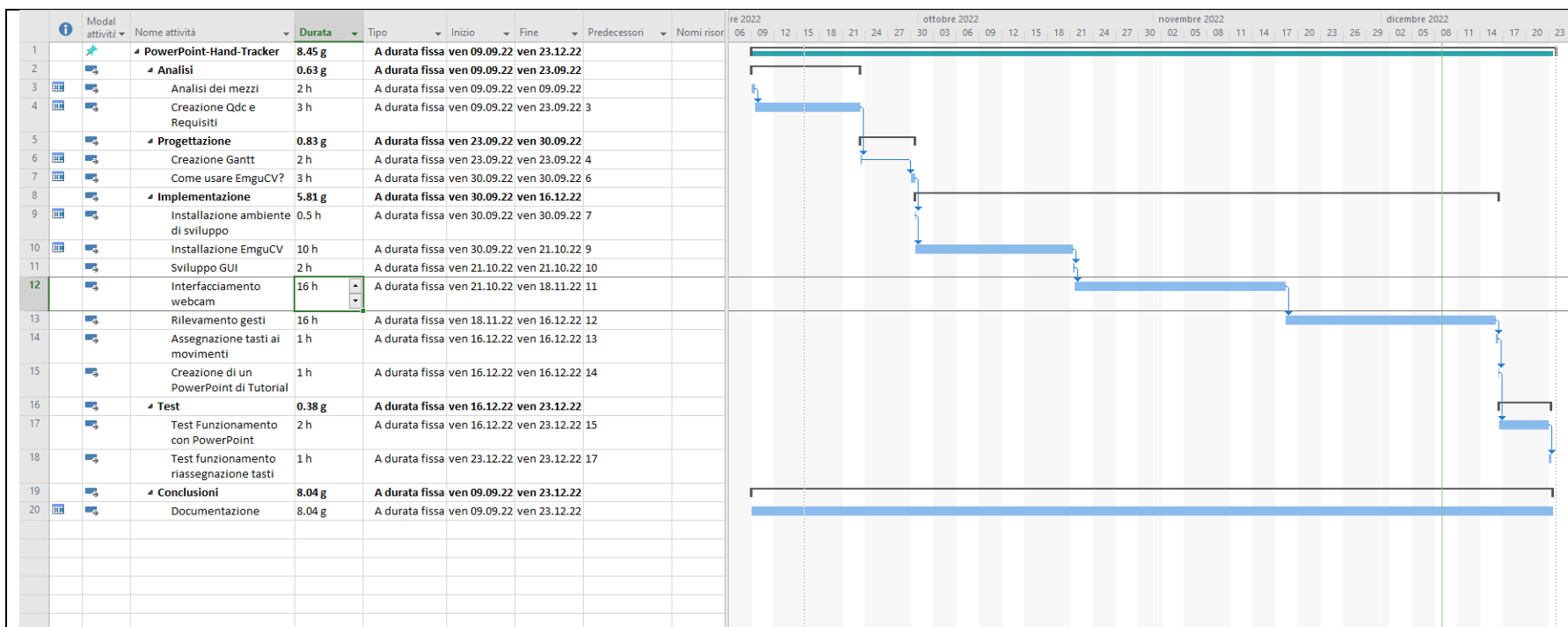


Figura 2: Diagramma di Gantt Reale

Titolo del progetto: Hand Gesture Recognition
Alunno/a: Diego Muniz Pereira
Classe: I3BB
Anno scolastico: 2022/2023
Docente responsabile: Geo Petrini

7 Conclusioni

7.1 Sviluppi futuri

Come sviluppo futuro si potrebbe migliorare la ricognizione della mano e del suo movimento dando la possibilità all'utente di selezionare il colore della propria pelle dato che non tutti lo stesso colore di pelle il che potrebbe creare problemi con certe persone.

7.2 Considerazioni personali

In conclusione, posso dire che il progetto mi ha insegnato molto. Dato che questo è stato il primo progetto serio che ho fatto ho imparato molto di come si fa e si porta avanti un progetto di questo peso sia sul lato teorico che pratico. Ho imparato come documentare il lavoro e creare i diari, creare Use Case e Diagrammi di Gantt ma anche come creare un progetto C# App Windows Forms (.NET Framework), come manipolare e visualizzare immagini da una webcam, come calcolare le coordinate di un punto in costante movimento, come installare e usare le librerie, come usare e convertire nuovi spazi di colori (YCrCb / Hsv), ...

8 Glossario

Termine	Descrizione
C#	C Sharp: è un linguaggio di programmazione del 2000 sviluppato da Mads Torgersen (Microsoft).
Windows Forms	Microsoft Windows Forms: è il nome dato alla parte di GUI del framework Microsoft.NET.
.NET Framework	.NET Framework: è l'ambiente di esecuzione runtime della piattaforma tecnologica .NET.
.NET	Microsoft .NET: è una piattaforma di sviluppo general purpose, ideata e sviluppata da Microsoft, che mette a disposizione varie funzionalità come il supporto per più linguaggi di programmazione.
YCrCb	YCrCb: indica una famiglia di spazi colore usata nei sistemi video a componenti e di fotografia digitale.
Hsv	Hue Saturation Value: indica sia un metodo additivo di composizione dei colori, sia un modo per rappresentarli in un sistema digitale.

Titolo del progetto: Hand Gesture Recognition
Alunno/a: Diego Muniz Pereira
Classe: I3BB
Anno scolastico: 2022/2023
Docente responsabile: Geo Petrini

9 Bibliografia

9.1 Sitografia

Esempio:

- <https://youtu.be/Fjj9ggTCTfc>, *Emgu CV Hand Gestures Recognition Advanced Tutorial*, 16.09.2022.
- <https://stackoverflow.com/questions/42655934/c-sharp-getting-a-list-of-names-of-the-webcams>, *C# Getting a list of names of the webcams*, 07.10.2022.
- http://web.archive.org/web/20100201125854/http://www710.univ-lyon1.fr/~bouakaz/OpenCV-0.9.5/docs/ref/OpenCVRef_StructAnalysis.htm, *Structural Analysis Reference*, 07.10.2022.
- <https://stackoverflow.com/questions/20470389/how-to-get-running-applications-in-windows>, *How to get running applications in windows?*, 14.10.2022.
- <https://stackoverflow.com/questions/15292175/how-to-send-a-key-to-another-application>, *How to send a key to another application*, 14.10.2022.

10 Allegati

Elenco degli allegati, esempio:

- Qdc
- Use-case
- Gantt preventivo
- Gantt consultivo
- Diari di lavoro
- Codici sorgente
- Applicazione