

Primary Flight Display

Soluzione software Primary Flight Display

Indice

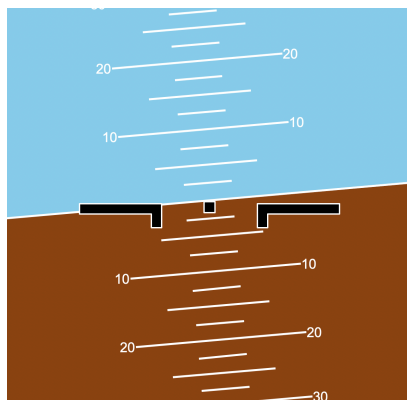
1. [Riepilogo Prodotto](#)
2. [Riepilogo Tecnico](#)
3. [Sviluppi Futuri](#)
4. [Crediti](#)

Crediti

Diego Ciucaloni
Matteo Fabbioni
Luca Niccià
Vittorio Piotti

[Repository GitHub.](#) [Video di Test](#)

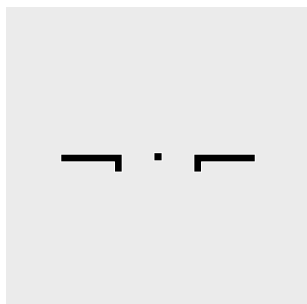
La schermata è stata acquisita dal progetto realizzato.



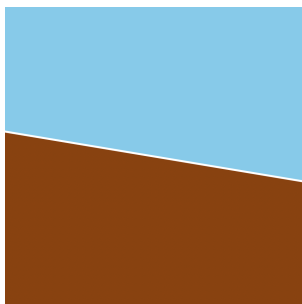
Riepilogo Prodotto

Funzionalità

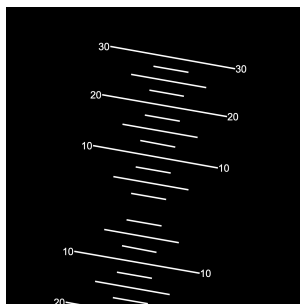
La schermate sono state acquisite dal progetto realizzato.



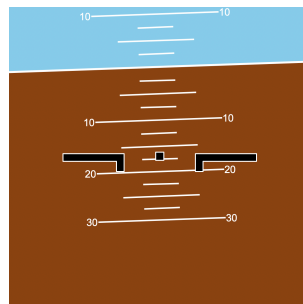
1



2



3



4

1. Mirino ([link](#))
2. Orizzonte ([link](#))
3. Scala del Pitch ([link](#))
4. Display funzionante ([link](#))

Copyright

Il codice è open-source.

Riepilogo Tecnico

Obbiettivo

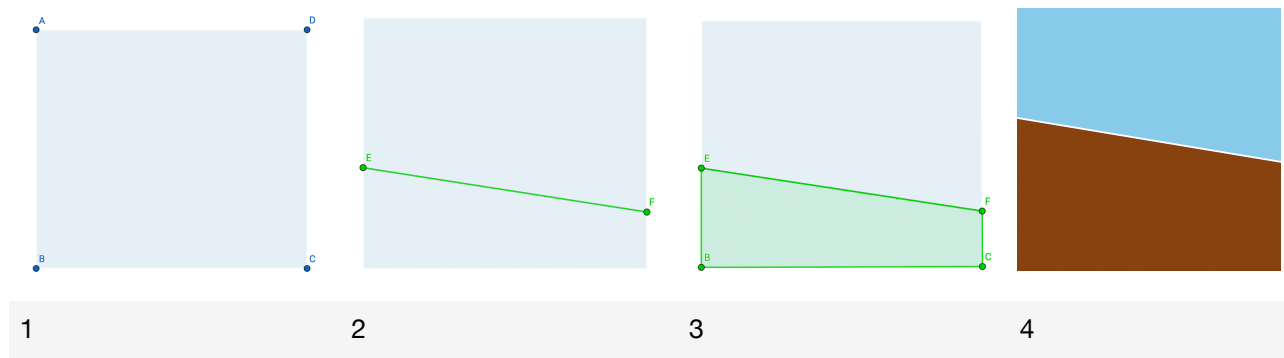
- Ricezione con una frequenza di 100 ms i dati in formato json dal server python.
- Elaborazione dei dati di pitch e roll per la visualizzazione dei componenti grafici

Indice

1. [Soluzione](#)
2. [Tecnologie](#)
3. [Note sul Codice](#)
4. [Riferimenti Matematici](#)
5. [Testing](#)
6. [Distribuzione Locale](#)
7. [Albero di Path](#)

Soluzione

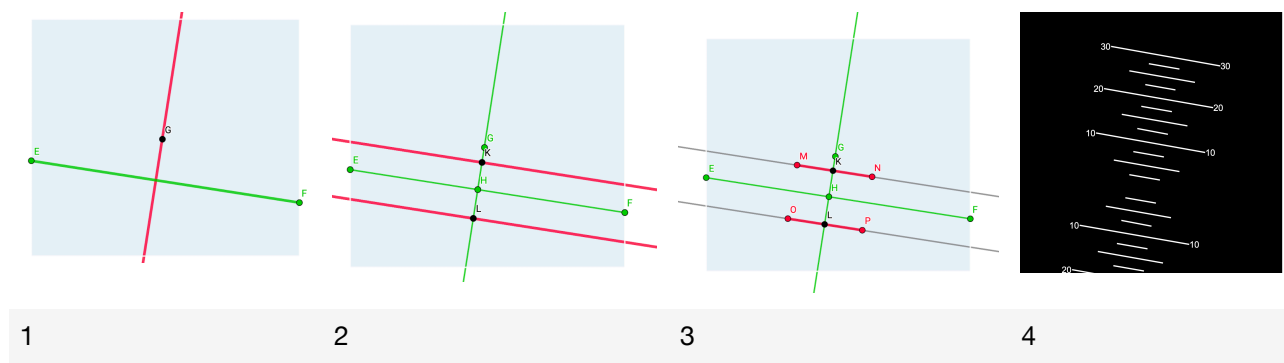
Orizzonte



Dati due punti qualsiasi agli estremi della larghezza del display crea l'orizzonte come:

1. Display quadrilatero di dimensione variabile
2. Orizzonte come segmento tra i punti degli estremi del display
3. Terra come quadrilatero compreso tra la base e l'orizzonte del display
4. Componente Realizzato

Scala del pitch



1. Data la retta dell'orizzonte ed il centro del display si calcola la retta passante per il centro e perpendicolare all'orizzonte
2. Data la retta dell'orizzonte e la retta ad esso perpendicolare si determina il punto di intersezione tra le parallele dell'orizzonte e la perpendicolare all'orizzonte
3. Date le rette parallele ed i punti di intersezione con la perpendicolare si calcolano per ciascuno 2 punti equidistanti dal punto di intersezione
4. Componente Realizzato

Tecnologie

Librerie linguaggi e codice:

- Linguaggio **Python** v.3.12 ([link](#))
- Libreria **TKinter** v.8.6 ([link](#))
- **Separazione dei compiti** per i componenti grafici in **classi distinte** garantendo la **scalabilità** e la **facile manutenzione** del codice.

Librerie linguaggi e codice:

- Connessione alla porta dell'ip del server
- Comunicazione tramite socket
- Architettura Client-Server

Note sul Codice

- Il codice è stato sviluppato in modo **compartimentato e modulare**
- È stata garantita una corretta **separazione dei compiti** e delle responsabilità delle classi.
- Ogni componente grafico del display è stato sviluppato in una classe autonoma
- Il codice è **scalabile ed ottimizzato** per la manutenzione
- Sono state implementate **classi di test** per ogni componente del display

Riferimenti Matematici

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1} \quad \text{Equazione parametrica della retta}$$

Utilizzato per calcolare l'equazione delle rette dei segmenti

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \text{Distanza euclidea tra due punti in un piano cartesiano}$$

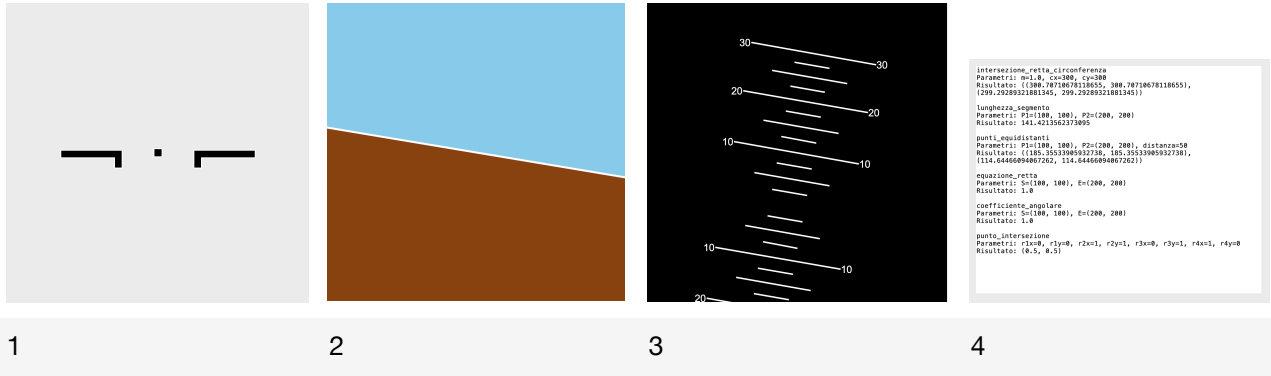
Utilizzato per calcolare la lunghezza del segmento che collega due punti

$$y = y_0 + m(x - x_0) \quad \text{Equazione del fascio di rette passante per un punto}$$

Utilizzato per calcolare l'equazione della retta dell'orizzonte

Testing

È stato creato uno script di test per ogni componente grafico o di logica da testare garantendo la scalabilità e la facile manutenzione del codice.



1. Test Mirino ([link](#))
2. Test Orizzonte ([link](#))
3. Test Scala del Pitch ([link](#))
4. Test Calcoli ([link](#))

- **Fatal Error** se il valore del **coefficiente angolare non è valido** perché non c'è nessun controllo per gestire questo caso.
- Rivedere ed ottimizzare la logica dei componenti grafici per visualizzare il PitchLadder

Tutti i test hanno dato **esito positivo** garantendo la **stabilità del software** solo se il **coefficiente angolare è valido**.

Distribuzione Locale

1. Configura **Python** ([link](#))

2. Configura ambiente virtuale:

2.1. Crea ambiente virtuale

```
-m venv myenv
```

2.2. attiva ambiente virtuale Mac:

```
source myenv/bin/activate
```

2.3. attiva ambiente virtuale Windows:

```
.\myenv\Scripts\Activate
```

3. scarica **tkinter** v.8.6 ([link](#)):

```
pip install tk
```

4. Caricare i file sorgenti eventualmente sostituendo quelli già presenti

Albero di Path

```
source
├── src
│   ├── Calc.py
│   ├── Horizon.py
│   ├── PitchLadder.py
│   └── Viewfinder.py
├── test
│   ├── Test_Calc.py
│   ├── Test_Horizon.py
│   ├── Test_PitchLadder.py
│   └── Test_Viewfinder.py
├── client.py
├── server.py
└── display.py
```

Sviluppi Futuri

Scala dello Yaw



Scala del Roll



Crediti

Diego Ciucaloni

Matteo Fabbioni

Luca Niccià

Vittorio Piotti