# Progetto RTOS

Berselli Werther

Di Blasi Fabrizio

# Titolo:

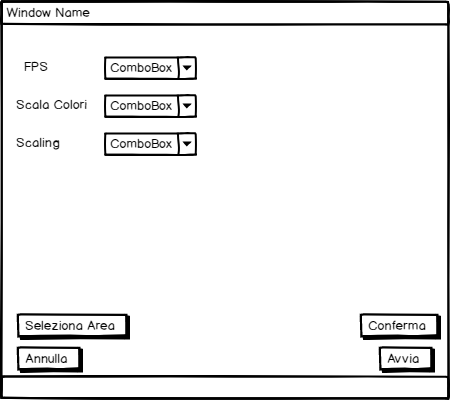
Image processing. Simulate a fixed camera looking at a given portion of an animated screen and write 4 periodic tasks performing different computations on the same image and displaying the results on other images. For example, task1 displays the histogram, task2 displays the image after applying a threshold, task3 displays the difference between consecutive frames, task4 displays the result of a filter. The user

must be able to vary some parameters of such tasks.

# Guida per l’utente:

Il programma ha il compito di analizzare uno stream video partendo da un riquadro tracciato a piacere dall’utente.

La schermata inziale è la seguente:



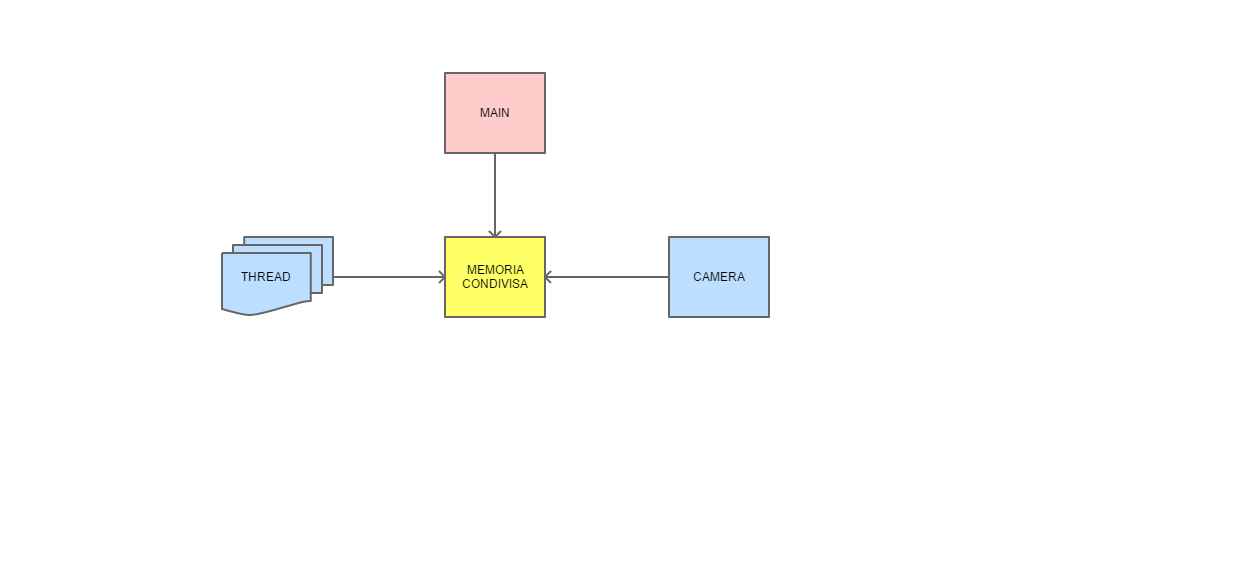
All’utente sarà permesso di scegliere alcuni iperparametri, che verranno utilizzati per la computazione.

All’inizio sarà “clickabile” solamente il tasto “**Seleziona Area**” che permetterà all’utente di selezionare la parte di schermo che si vuole processare.

Una volta selezionata la porzione di schermo, si potrà premere il tasto “**conferma**” il quale farà comparire a video le 5 aree di lavoro (videocamera, e 4 thread)

Infine col tasto **annulla** si ritorna alla scelta degli iperparametri, altrimenti si avvia la computazione.

# Struttura del codice ed organizzazione della memoria condivisa:



Main:

* Inizializza il frame dell’interfaccia utente iniziale vista sopra
* Permette la scelta degli iperparametri nelle modalità già descritte
* Mette in esecuzione i thread

Camera:

* Gestisce la memoria condivisa tramite i semafori. Produce il frame, disattiva il suo semaforo ed attiva i semafori dei thread
* Inserisce l’ultimo frame all’interno della memoria condivisa
* Visualizza sul suo frame GUI il video che si sta analizzando
* Cattura il video ad “X” FPS scelti dall’utente nella fase iniziale

Memoria Condivisa:

* Semafori dei thread
* Mutex mutua esclusione
* FPS (iper parametro)
* Scala Colori (iper parametro)
* Scaling (iper parametro)
* Immagine catturata dalla camera **NON** RIDIMENSIONATA DI **DIMENSIONI FISSE**
  + Altezza
  + Larghezza
* Immagine catturata dalla camera **RIDIMENSIONATA DI DIMENSIONI FISSE**
  + Altezza
  + Larghezza

Thread:

* Parametri variabili impostabili dall’utente
* Variabili personali

## Organizzazione della memoria condivisa:

Il main mette in esecuzione il thread videocamera ed inizia a campionare il video.

Il fotogramma (campione) verrà inserito in due celle di memoria condivisa.

* Una cella conterrà il fotogramma a dimensioni originali non alterate
* L’altra conterrà il fotogramma a dimensioni ridotte in base alla volontà dell’utente

In questo modo è possibile far prendere il fotogramma ridimensionato per operazioni in cui è accettabile perdere informazioni, come ad esempio l’applicazione di un filtro.

Mentre quello a dimensioni originali sarà utilizzato da thread che faranno operazioni *non lossy* come ad esempio il tracciamento di un’istogramma