Relazione Sistemi Multimediali

# Autori

Alexandru Stefan Andries 20018146

Nicolò Benella 20018152

# Descrizione Progetto

Effettuare la rotazione di un’immagine, sia a colori che bianco e nero, e mostrare la sua animazione in una nuova finestra.

Come extra, l’implementazione di un Double Buffer.

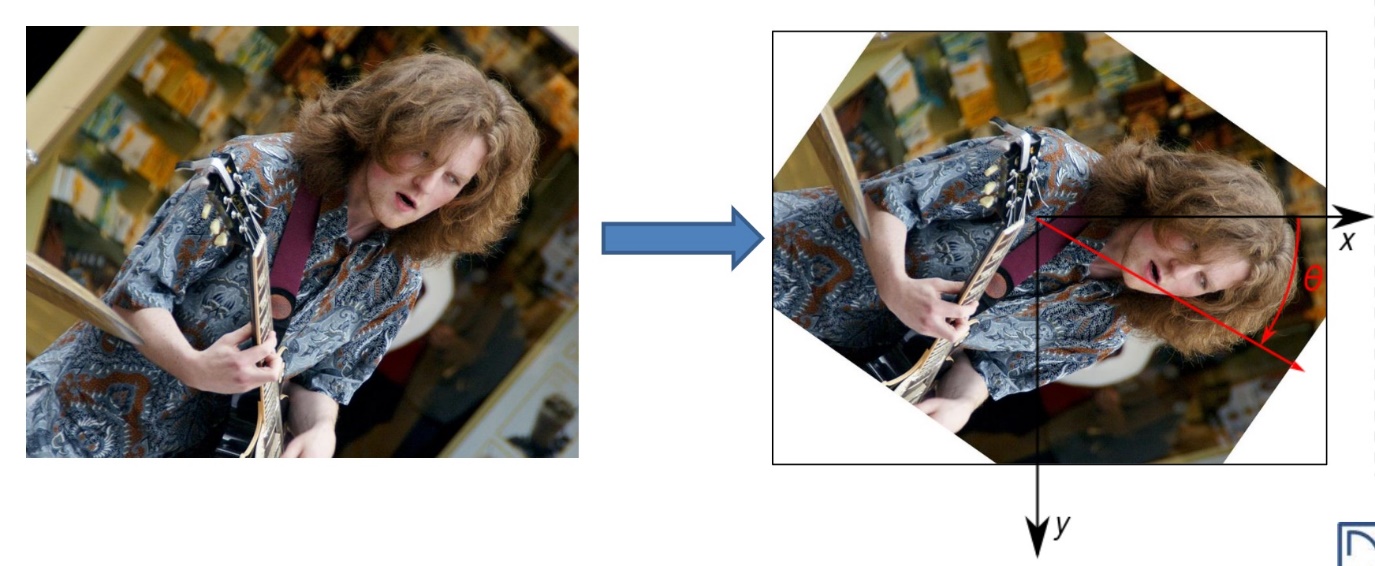
# Spiegazione Progetto

Per iniziare abbiamo seguito i consigli presenti sul pdf per la relazione finale, ovvero abbiamo diviso il progetto in 2 parti.

La prima consiste nell’eseguire l’effettiva rotazione di un’immagine a colori o in bianco e nero, dato un determinato angolo di rotazione.

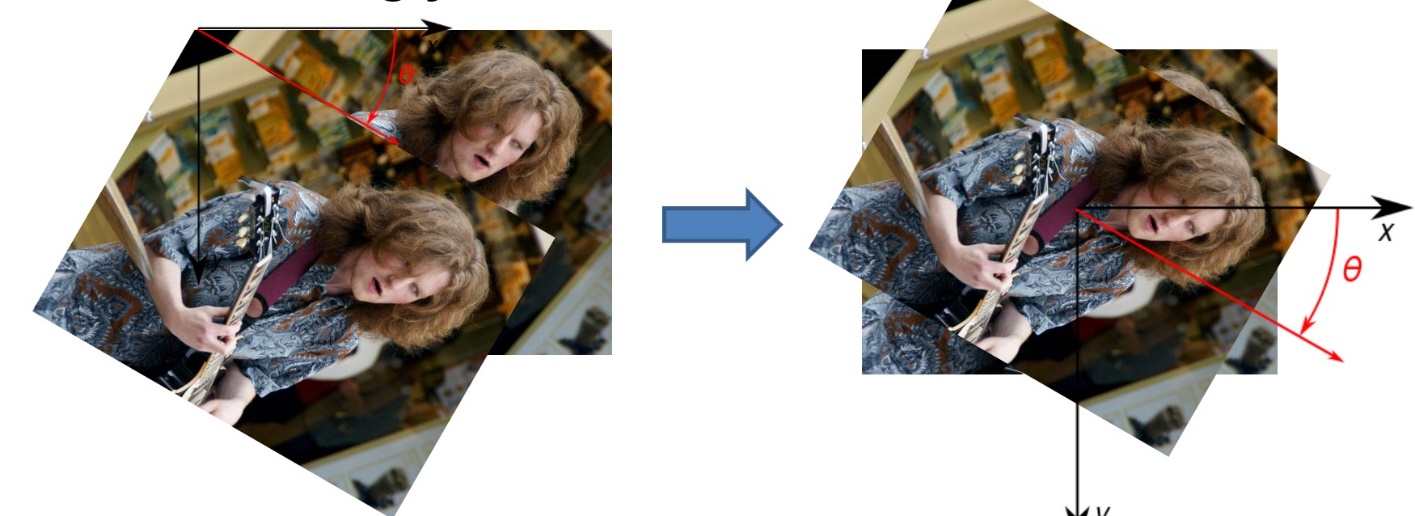
La seconda invece è la parte di animazione della rotazione nella finestra.

# Rotazione di un immagine



Per eseguire la rotazione abbiano dovuto risolvere i diversi problemi che erano già stati anticipati nel pdf finale.

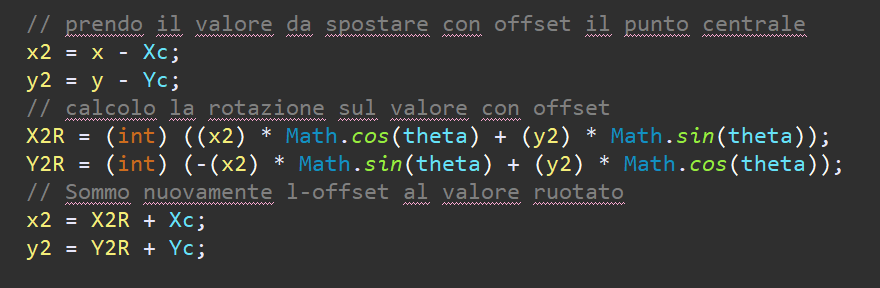
PROBLEMA 1



Pur avendo la formula di rotazione base, essa permetteva la rotazione di un’immagine solamente sul punto 0,0 ovvero l’angolo in alto a sinistra

Nel nostro progetto ci interessava la rotazione sul punto centrale dell’immagine.

Abbiamo quindi dovuto modificare a dovere la formula trovando infine questa:

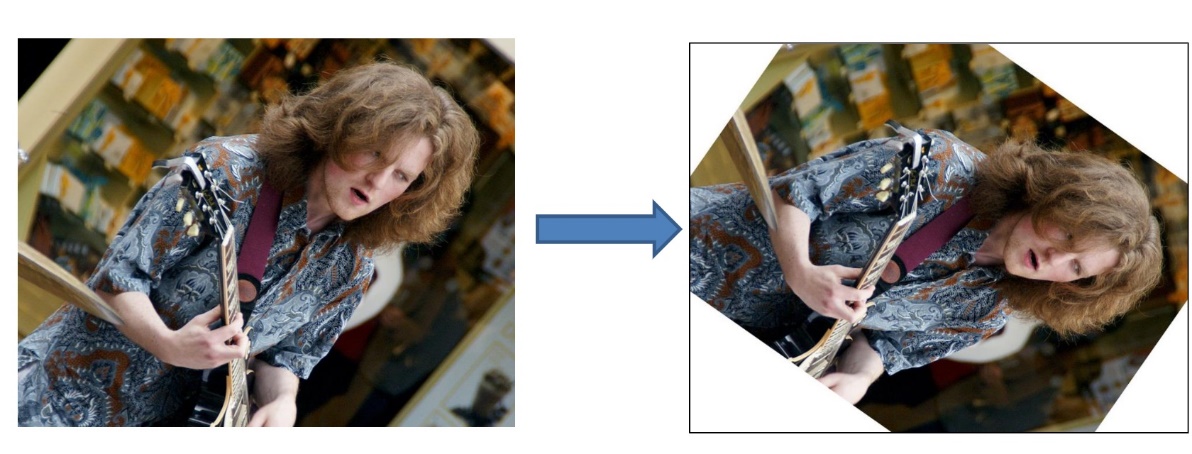


Anziché effettuare la formula di rotazione direttamente sul punto analizzato, possiamo aggiungere come offset il punto centrale dell’immagine ed eseguire la rotazione sul nuovo punto trovato.

In questo modo spostiamo il pixel intorno al centro e non al punto 0,0.

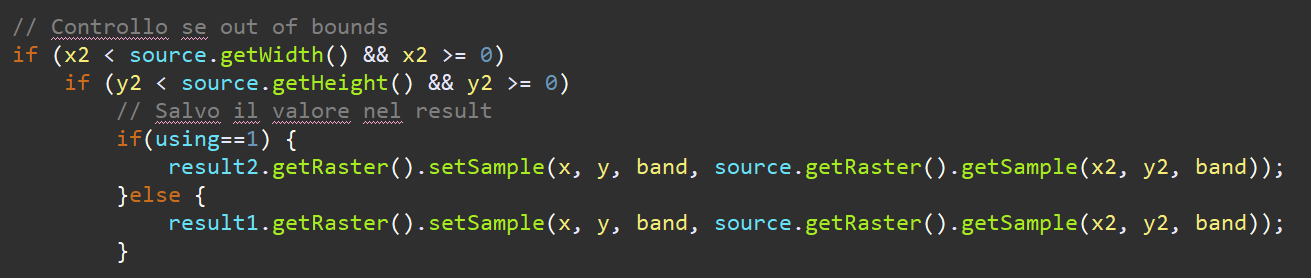
Per trovare poi nuovamente la sua posizione effettuare aggiungiamo nuovamente l’offset.

PROBLEMA 2



Il secondo problema che abbiamo dovuto risolvere è quello del clipping.

Ruotando l’immagine succede che i bordi possano uscire dalla grandezza originale dell’immagine e questo potrebbe portare a errori nel programma.



Per risolvere il problema abbiamo semplicemente fatto un controllo che prima di inserire effettivamente il pixel controlla se la sua posizione è compresa tra 0 e la larghezza/altezza dell’immagine. Dopodiché salva il valore ( using verrà spiegato in seguito ).

PROBLEMA 3

Il terzo problema è quello della mancanza di pixel o artefatti dopo aver ruotato un’immagine.

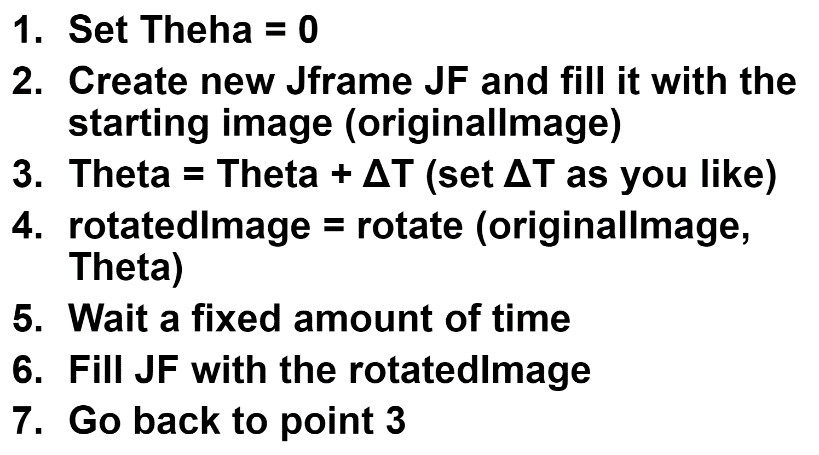
Per risolverlo ci è sembrato più veloce usare la soluzione numero 2, ovvero anziché fare upsample – rotation – downsample, abbiamo semplicemente trovato quella pixel dell’immagine originale dovrebbe stare in quella posizione.



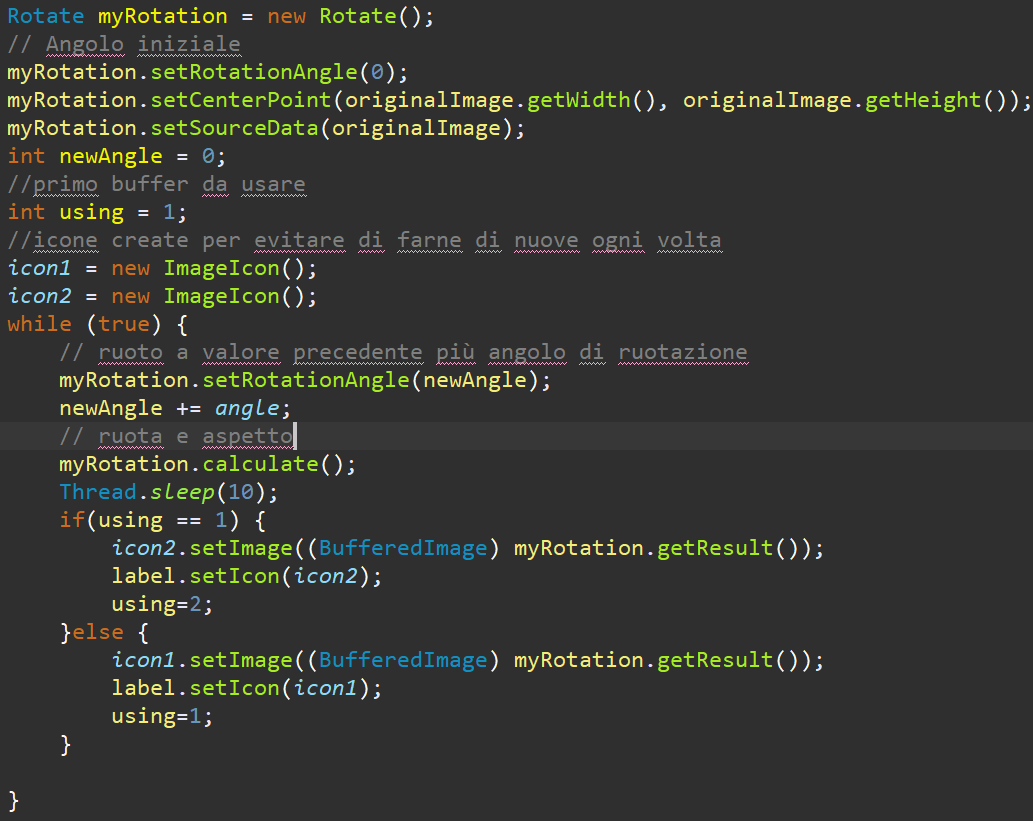
Con questo abbiamo risolto il problema della rotazione.

# Animazione

Per effettuare l’animazione abbiamo seguito l’algoritmo generale descrittoci sul pdf.



Tradotto in codice abbiamo:

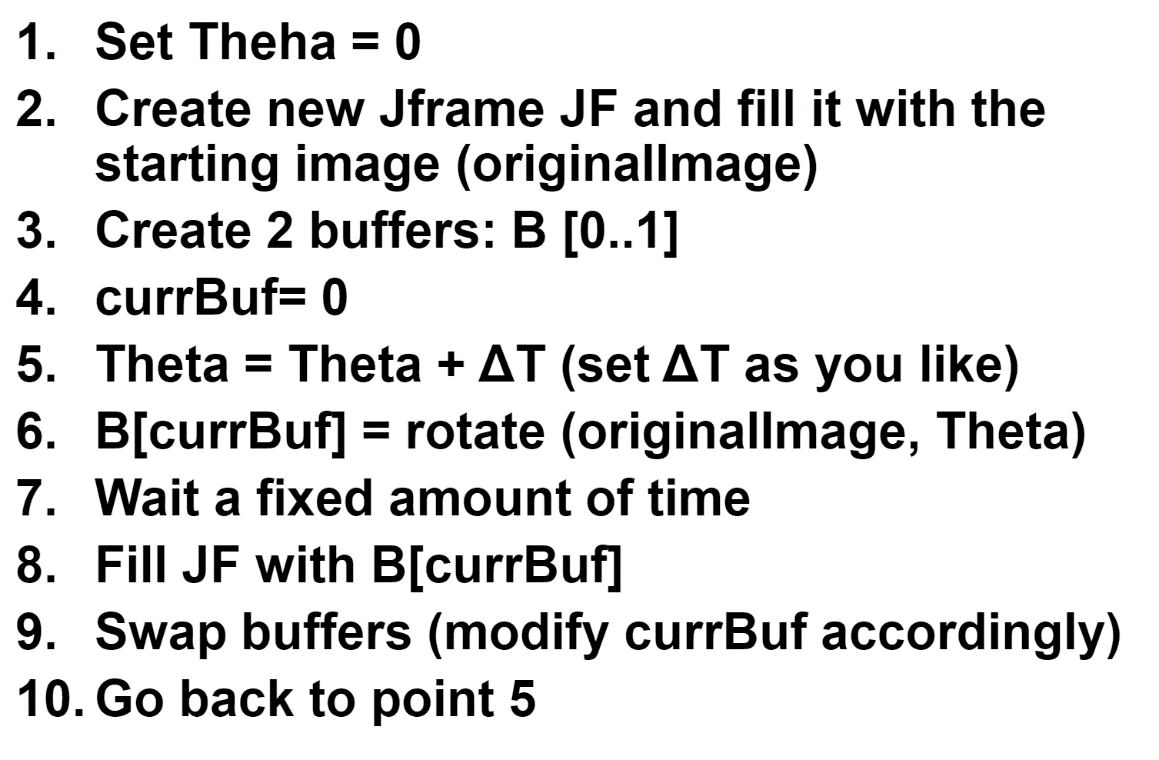


Il codice è molto semplice e rispecchia l’algoritmo generico scritto sopra. L’animazione prosegue fino a quando non viene chiusa la finestra.

# EXTRA - Double Buffer

Abbiamo aggiunto al progetto anche una semplice implementazione del double buffer.

Sempre seguendo l’algoritmo generico:



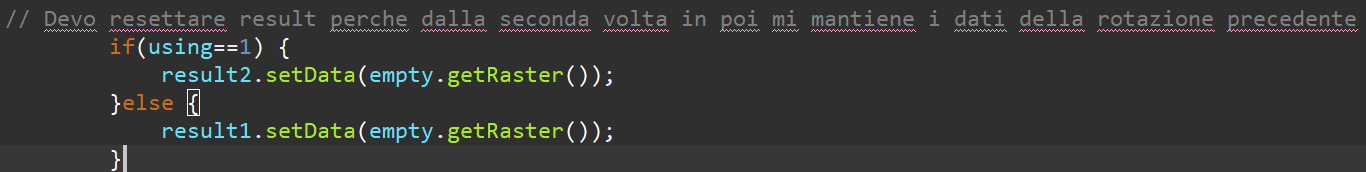
È possibile vedere questa implementazione anche nelle immagini precedenti ( [qui](#codiceOutOfBounds) e [qui](#codiceAnimazione)).

Praticamente nell’animazione facciamo in modo di alternare l-utilizzo del primo e secondo buffer per evitare di lavorare su quello visualizzato al momento.

Nella calculate abbiamo dovuto lavorare sullo stesso problema, infatti lavoriamo prima su result1 e poi su result2.

Un problema extra da risolvere è stato quello dell’immagine residua, ovvero rimaneva nella BufferedImage bordi dell’immagine ruotata in precedenza.

Per risolverlo copiamo una immagine vuota prima ogni volta che dobbiamo fare la rotation



# Considerazioni finali progetto

Il progetto è stato abbastanza semplice, abbiamo avuto leggere difficoltà durante l’implementazione del double buffer e non siano ancora pienamente convinti sia stato implementato nella maniera più efficace, ma tutto sommato siamo soddisfatti del risultato finale ottenuto.