



RAINBOW<sub>1.1</sub>



# COLORINO

- 105 grammi, batterie incluse
- 110x50x25 mm esclusa custodia
- Range di prezzi 199.00€ – 499.00€
- Poco pratico
- Necessita di contatto con le superfici
- Alcune importanti limitazioni

Come fare di meglio?

- Prezzo sempre in diminuzione
- Sensori CMOS sempre più sofisticati (8+ MP)
- Unità di calcolo veloci (400 Mhz)
- Chip dedicati a Java VM e accelerazione grafica
- Bacino di utenti globale
- Accessibilità





# Mobile Color Recognition System

ΜΙΟΒΙΗΓ ΓΟΙΟΛ ΚΕΓΟΘΥΠΙΟΥ ΠΛΗΓΩ

- E' un software. Sfrutta dispositivi già esistenti.
- E' scritto in Java (J2ME su Sun MMAPi 1.1)
- E' indipendente dal layer di accessibilità
- Risolve molti dei problemi legati alle alternative esistenti



VS



Costo minimo 199€ (medio 350€)	<del>Costo minimo 39€ (medio 150€)</del>
Dispositivo dedicato	Layer software su dispositivi esistenti
Necessita di contatto con le superfici	Sfrutta la fotocamera integrata
Ingombrante	Occupa lo stesso spazio del telefono
Assistenza limitata	Efficienti sistemi di assistenza remota



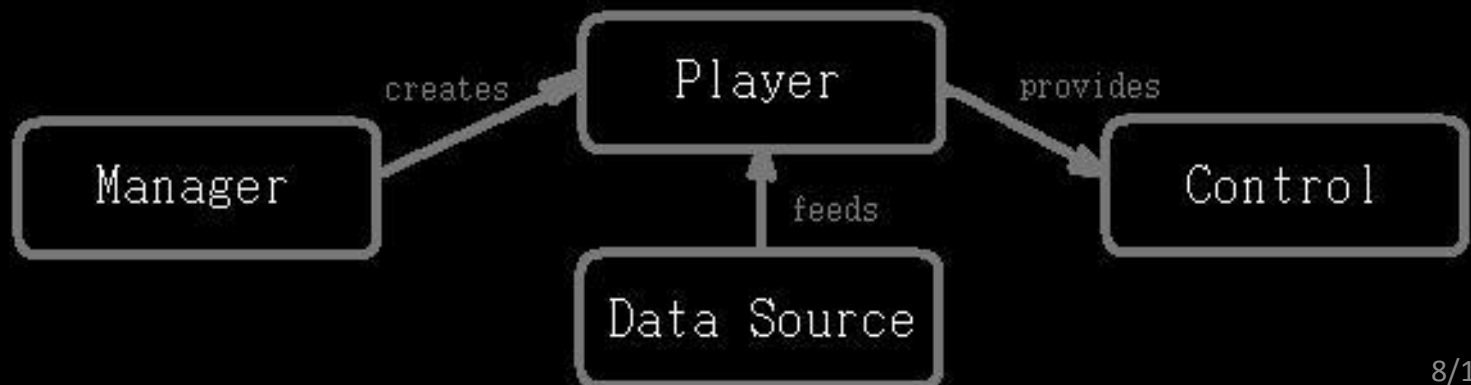
# Funzionamento

1. Avvio del programma sul dispositivo mobile
2. Puntamento dell'oggetto di cui si vuole conoscere il colore.  
Attivazione dispositivo mediante la pressione di un tasto singolo.
3. Fase di elaborazione automatica
4. Visualizzazione su schermo del nome del colore dominante dell'oggetto in formato riconoscibile dai lettori vocali



# Realizzazione

- Sfrutta Sun Multimedia Application Programming Interface 1.1 (MMAPI), accessibile da Java.
- Vantaggi della programmazione ad oggetti
- Struttura a pipeline in 6 fasi. L'output di ogni fase è l'input della fase successiva.
- Paradigma MPCDS







# Realizzazione

Pipeline di elaborazione – Fase 1 Acquisizione



- Creazione d un Player collegato al sensore CMOS, sfruttando la classe Manager
- Creazione di controlli per regolare diversi parametri sul sensore (White Balance, Exposure)
- Salvataggio dell'immagine in forma di Array di Byte, e successivamente convertito in formato PNG



# Realizzazione

Pipeline di elaborazione – Fase 2 Crop



- L'immagine ottenuta viene ritagliata a partire dal centro fino ad una superficie pari all'80% dell'immagine originale.



# Realizzazione

Pipeline di elaborazione – Fase 3 Filtro Gaussiano

- Applicazione di un Filtraggio Gaussiano
- Eliminazione del rumore. “Liscia” il segnale.
- Realizzazione tipica dell’algoritmo, 2 cicli ‘for’ annidati.
- Il filtro è separabile.
- Filtro di ampiezza 3 pixel.





# Realizzazione

Pipeline di elaborazione – Fase 4 Correzione Gamma

- Correzione Gamma dell'immagine così ottenuta.
- Parametro Gamma fissato a priori a 1.5





# Realizzazione

Pipeline di elaborazione – Fase 5 Calcolo Colore Dominante

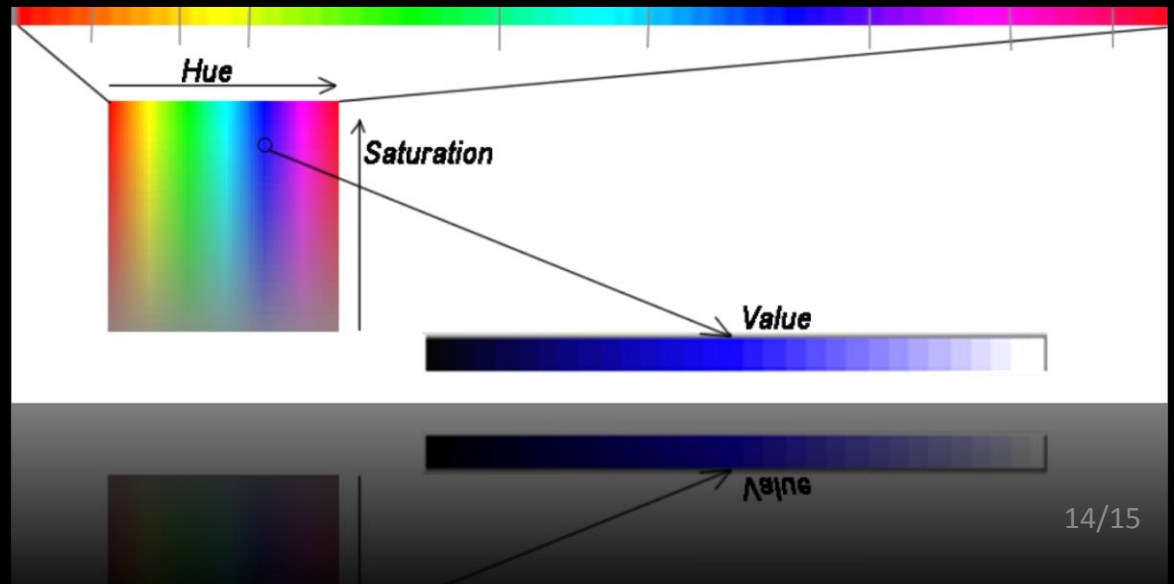
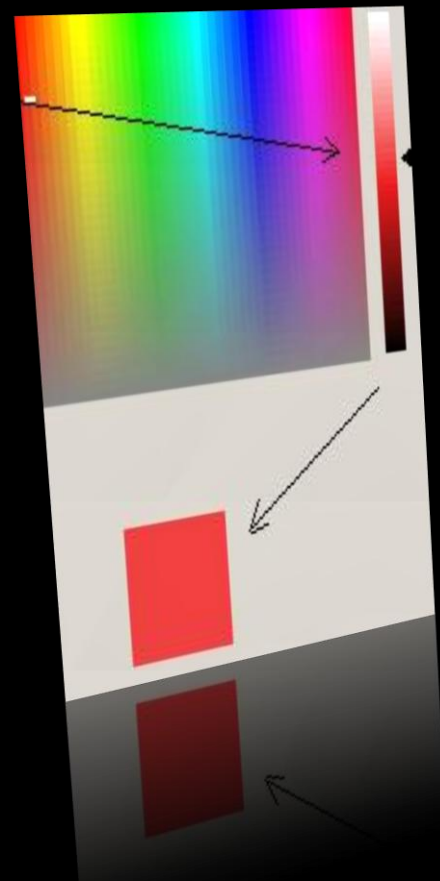
- Ridimensionamento immagine tramite sottocampionamento
- Calcolo della componente di colore media.
- Valutazione della ‘bontà’ dell’immagine. Eventuale visualizzazione di un messaggio all’utente.



# Realizzazione

Pipeline di elaborazione – Fase 6 Analisi HSV

- Conversione del colore trovato nello spazio HSV
- Analisi più intuitiva per l'uomo
- Visualizzazione a video del nome del colore trovato  
24 Colori + Sfumature



# Problematiche

Qualcosa su cui riflettere

- Bilanciamento del bianco (Java vs C++)



- Controllo esposimetro integrato
- Riconoscimento trame

