

## Attività2 - Indovina pixel

- **Materiale:** fantacolor modular 4 di Quercetti
- **Età:** a partire da 8 anni
- **Competenze acquisite a fine attività:**

### **Obiettivi di apprendimento al termine della classe terza della scuola primaria**

Ambito dati e informazione

- O-P3-D-1. scegliere ed utilizzare oggetti per rappresentare informazioni familiari semplici (es. colori, parole, ...)
- O-P3-D-2. definire l'interpretazione degli oggetti utilizzati per rappresentare l'informazione

### **Obiettivi di apprendimento al termine della classe terza della scuola primaria**

Ambito dati e informazione

- O-P5-D-2. utilizzare simboli per rappresentare semplici informazioni strutturate (es. immagini "bitmap", ...)

A fine attività è consigliata la visione del quesito Bebras Red, Green, Blue su <https://bebras.it/lib/libretto-esempi.pdf>

Preparazione: distribuisce una tavoletta ad ogni bambino oppure dividi in gruppi e distribuisce una ad ogni gruppo. Chiedi ai bambini di realizzare dei disegni inizialmente con i chiodini da 20mm, non usare i colori bianco, nero e rosa.

Immaginiamo di essere un computer e di disegnare delle immagini posizionando i pixel sulla griglia, i pixel saranno chiodini colorati.

Successivamente, a turno, ogni bambino (o gruppo) guardando il lavoro altrui deve indovinare le immagini rappresentate dagli altri. Senza farsi sentire deve comunicarlo al proprietario, se indovina, il proprietario del disegno acquisisce un punto. Vince il bambino che ha ottenuto più punti.

Cosa significa? C'è qualcuno che ha ottenuto 0 punti? Perché gli altri bambini hanno ottenuto meno punti? La loro immagine è risultata meno chiara e quindi difficilmente indovinabile dagli altri. Perché?

Proietta l'immagine:

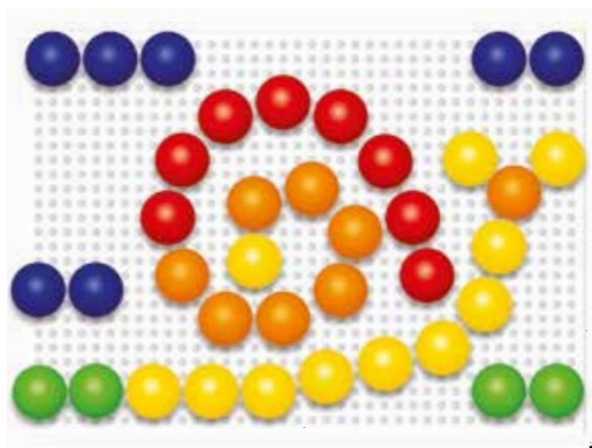


Figura 1

A colpo d'occhio riusciamo a capire cosa rappresenta l'immagine? Sembrerebbe essere una lumaca ma la risoluzione è molto bassa. Quanti pixel abbiamo utilizzato? 40.

Chiedi adesso di realizzare la stessa figura oppure una diversa con i chiodini da 15mm.  
Ripeti lo stesso gioco facendo indovinare le immagini degli altri.

Cosa è cambiato rispetto a prima? I bambini hanno ottenuto più punti rispetto alla volta precedente? Se qualcuno prima aveva ottenuto 0 punti, adesso quanti ne ha ottenuti?

Proietta l'immagine:

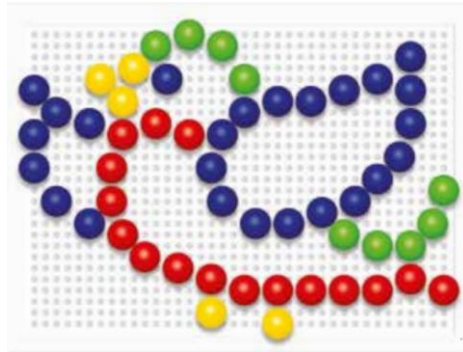


Figura 2

Cosa cambia rispetto alle figure precedenti? Quanti pixel abbiamo utilizzato? 56.

Utilizzando pixel più piccoli riusciamo ad utilizzarne di più. Forse l'immagine inizia a diventare più chiara.

Chiedi adesso di realizzare la stessa figura oppure una diversa con i chiodini da 10mm.

Ripeti lo stesso gioco facendo indovinare le immagini degli altri.

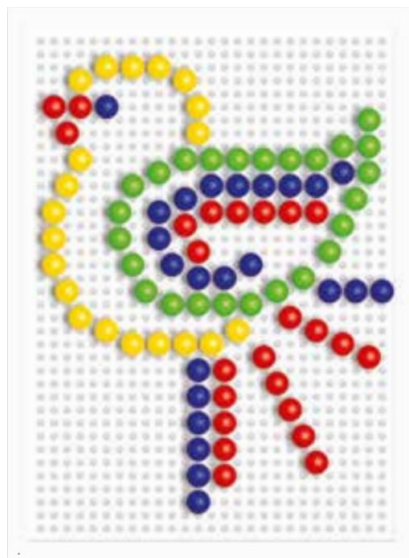


Figura 3

Cosa cambia rispetto alle figure precedenti\*? Quanti pixel abbiamo utilizzato? 92.

L'immagine inizia a diventare molto più chiara perché abbiamo aumentato la risoluzione: diminuendo la grandezza dei pixel aumenta il numero di pixel che possiamo inserire.

Per questo motivo le figure con i chiodini da 10mm risultano più definite rispetto a quelle con i chiodini da 20mm.

Chiedi adesso ai bambini di disegnare una figura con pixel neri di 10mm riempiendo lo sfondo con i chiodini bianchi e scrivere il codice dell'immagine, poi ripetere il procedimento per la figura con pixel da 20mm. Fai confrontare i due codici e fai riflettere i bambini sul perché uno è più lungo dell'altro.

Se aumentando la risoluzione aumenta il numero di pixel, aumenta anche la complessità della rappresentazione dell'immagine sul computer in quanto il codice è più lungo da scrivere (Figura 4).

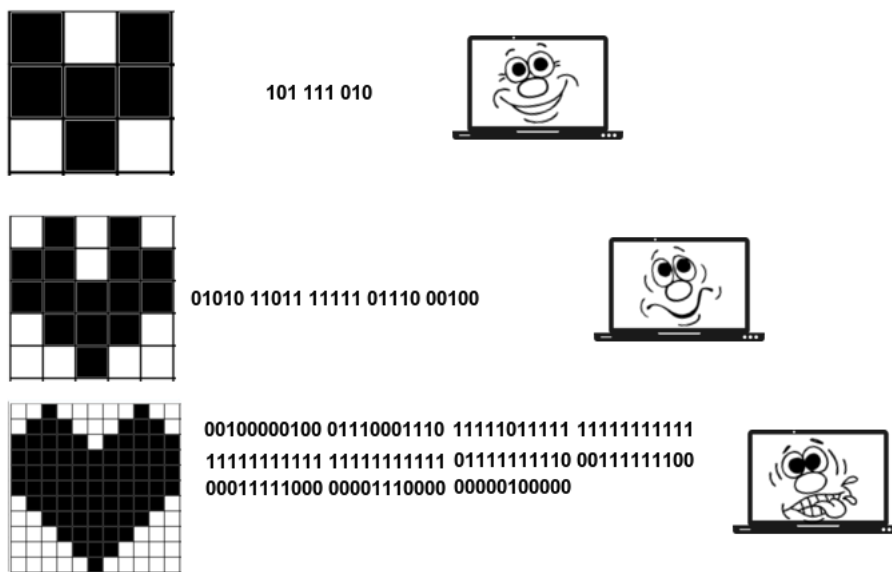


Figura 4

Abbiamo imparato fino ad ora a scrivere codici per le immagini in bianco e nero ma non abbiamo utilizzato i chiodini a colori, perché?

Fai mettere da parte ai bambini il codice dell'immagine da 20mm appena creata in bianco e nero, fagli sostituire i chiodini neri con quelli colorati, sempre da 20 mm, in modo da codificare la stessa immagine a colori (lasciando sempre i chiodini bianchi come sfondo).

Come codifichiamo l'immagine?

**0** → bianco

**1** → nero

**?** → rosso

Ricordiamoci che nella codifica binaria si possono utilizzare solo le cifre 0 e 1, come possiamo risolvere?

Qualcuno ha idee su come identificare il colore rosso?

Un'idea possibile...:

**00** → bianco

**11** → nero

**10** → rosso

\* Figura 1, Figura 2, Figura 3 prese da [https://quercetti.sintrasviluppo.it/Download/AlbumFantaColorAll\\_web\\_compressed\\_1.pdf](https://quercetti.sintrasviluppo.it/Download/AlbumFantaColorAll_web_compressed_1.pdf)

Ma ci sono altri colori...

- 00 → bianco
- 11 → nero
- 10 → rosso
- ? → verde
- ? → blu
- ? → giallo
- ? → arancione
- ? → azzurro

Proviamo allora tutte le combinazioni... →

- 00 → bianco
- 11 → nero
- 10 → rosso
- 01 → verde
- ? → blu
- ? → giallo
- ? → arancione
- ? → azzurro

→

Ci siamo bloccati, non ci sono più combinazioni possibili, qualcuno ha un'idea su come procedere?

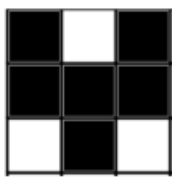
Se con 1 bit posso rappresentare 2 colori e, se con 2 bit posso rappresentare 4 colori, con 3 bit quanti colori posso rappresentare? Proviamo! →

- 000 → bianco
- 111 → nero
- 101 → rosso
- 011 → verde →
- 110 → blu
- 010 → giallo
- 001 → arancione
- 100 → azzurro

- 1 bit → 2 colori infatti  $2^1 = 2$  (2x1)
- 2 bit → 4 colori infatti  $2^2 = 4$  (2x2)
- 3 bit → 8 colori infatti  $2^3 = 8$  (2x2x2)

Quindi le potenze del 2 indicano quante informazioni, in questo caso le informazioni sono i colori, si possono rappresentare avendo a disposizione 2 bit

Dopo aver codificato l'immagine a colori, confrontate i due codici, quello appena ottenuto e quello precedente, cosa notate? Tanti più bit vengono usati per definire i colori, tanto più si potrà essere precisi ma tanto più sarà grande il codice.



101 111 010



101 000 101010 001010 000101000

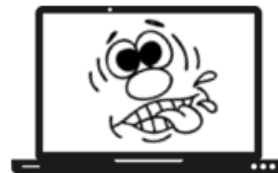


Figura 5

Nella realtà, per le immagini a colori si utilizza la codifica RGB: il computer usa combinazioni di Rosso, Verde e Blu per formare gli altri colori, sommandoli tutti e tre si ottiene il bianco mentre il nero è dato dall'assenza di colore.

Tale sistema prevede che il rosso, il verde e il blu siano codificati mediante un numero intero utilizzando 8 bit. Quindi ogni pixel per essere memorizzato richiede 24 bit (3 byte).

- Rosso: 11111111 00000000 00000000
- Verde: 00000000 11111111 00000000
- Blu: 00000000 00000000 11111111

Siccome con 8 bit posso rappresentare al massimo  $2^8 = 256$  combinazioni, il numero totale di colori distinti con questo sistema sarà  $256 \times 256 \times 256 = 16.777.216$ , cioè più di 16 milioni di colori.