

Homework

Informazioni generali

- Ogni studente deve **completare l'homework prima di sostenere la prova orale per il primo modulo.**
- Ogni homework dev'essere svolto da **un solo studente.**
- L'homework dev'essere **consegnato al più 2 giorni prima della data della prova orale** a cui lo studente intende partecipare.
- L'homework deve essere **inviato all'indirizzo email *m.oliverio@dimes.unical.it*.**
- L'homework si intende **consegnato solo se si riceve un'email di risposta.**

Il modulo "cImage"

Il modulo "cImage" è disponibile nella sezione "Homework" del sito Web del corso, insieme ad alcuni esempi di utilizzo.

Si richiamano di seguito alcune funzioni del modulo utili per la realizzazione dell'homework (si faccia comunque riferimento anche agli esempi forniti sul sito del corso).

- Le funzioni `getWidth()` e `getHeight()` applicate ad un'immagine (`cImage.Image`) ne restituiscono larghezza e altezza, rispettivamente.
- La funzione `cImage.EmptyImage(l,a)` restituisce una nuova immagine vuota di larghezza `l` e altezza `a`.
- La funzione `getPixel(j,i)` applicata ad un'immagine restituisce il pixel sulla riga `i` e colonna `j`.
- La funzione `setPixel(j,i,p)` applicata ad un'immagine imposta il pixel `p` sulla riga `i` e colonna `j`.
- La funzione `getRed()` applicata ad un pixel restituisce il valore della componente rosso. I valori sono compresi tra 0 e 255.
- La funzione `getGreen()` applicata ad un pixel restituisce il valore della componente verde. I valori sono compresi tra 0 e 255.
- La funzione `getBlue()` applicata ad un pixel restituisce il valore della componente blu. I valori sono compresi tra 0 e 255.
- La funzione `cImage.Pixel(r,g,b)` restituisce un nuovo pixel con i dati valori di rosso (`r`), verde (`g`) e blu (`b`). I valori sono compresi tra 0 e 255.
- La funzione `captureClicks(n)` applicata ad una finestra (`cImage.ImageWin`) cattura `n` click del mouse su un'immagine e restituisce la lista delle coordinate dei pixel su cui sono stati effettuati i click (si veda in particolare l'esempio "gestione_immagini_cattura_click.py").

Tracce

Traccia 1: Trasformazione di immagini

Si personalizzi la funzione “operazione(immagine_originale)” nel modulo “gestione_immagini.py”, implementando una trasformazione diversa da quelle esemplificate nei file “gestione_immagini_esempio_x.py”.

Di seguito alcune possibili trasformazioni suggerite (lo studente può comunque individuare e sviluppare una trasformazione a piacere):

- Aumentare o ridurre di una data percentuale la componente rossa (o verde o blu) di ognuno dei pixel dell'immagine originale.
- Convertire l'immagine originale in toni di rosso (o verde o blu) come esemplificato per i toni di grigio.
- Costruire un'immagine in cui la componente rossa di ogni pixel è uguale alla media delle componenti rosse dei pixel adiacenti nell'immagine originale (applicando la stessa logica alle componenti verde e blu).
- Dividere l'immagine originale in 4 quadranti uguali e ricomporli in ordine diverso.
- Costruire un'immagine che rappresenti quella originale ruotata rispetto all'asse orizzontale (o verticale).
- Convertire in toni di grigio una porzione dell'immagine originale, facendo scegliere all'utente la porzione mediante due click del mouse (corrispondenti ai pixel in alto a sinistra e in basso a destra).

Traccia 2: “Forza 4”

Si sviluppi un'interfaccia grafica per il gioco “Forza 4” (disponibile nella sezione “Homework” del sito Web del corso, come soluzione dell'esercizio 3 della traccia del 24/6/2015) mediante le funzioni del modulo “cImage”.

Traccia 3: Libera (con modulo cImage)

Si sviluppi un programma a piacere riguardante l'elaborazione di immagini mediante le funzioni del modulo “cImage”.

Traccia 4: Libera (con modulo turtle)

Si sviluppi un programma grafico a piacere mediante le funzioni del modulo “turtle”.