Diario di lavoro

|  |  |
| --- | --- |
| Luogo | Canobbio |
| Data | 03.03.2023 |

|  |
| --- |
| Lavori svolti |
| <https://kivy.org/doc/stable/api-kivy.uix.slider.html>  Aggiunto uno slider per la tolleranza del bordo nello schermo di modifica dell’immagine  <https://kivy.org/doc/stable-1.10.1/api-kivy.uix.screenmanager.html?highlight=get_screen#kivy.uix.screenmanager.ScreenManager.get_screen>  Risolto problema del caricamento dell’immagine e del passaggio tra screen  <https://kivy.org/doc/stable/api-kivy.uix.image.html>  <https://docs.opencv.org/4.x/d3/df2/tutorial_py_basic_ops.html>  Creato metodo per calcolare le coordinate all’interno dell’immagine di dove si preme il mouse.  Quando si preme il mouse viene colorato il pixel premuto.  <https://stackoverflow.com/questions/889333/how-to-check-if-a-file-is-a-valid-image-file>  Installato filetype e aggiunto il controllo per verificare se la path immessa per l’immagine è veramente un’immagine.  Continuata la documentazione |

|  |
| --- |
| Problemi riscontrati e soluzioni adottate |
| L’immagine quando si cambia la tolleranza del bordo, non viene ridisegnata a schermo, per risolvere questo problema abbiamo inizialmente cercato di pulire le cache, questa soluzione non ha funzionato. Chiedendo consigli al docente abbiamo notato che la versione di kivy da noi utilizzata (1.0.9) non era quella installata, questo limitava le nostre opzioni. Quindi abbiamo messo la versione installata (2.1.0) e abbiamo utilizzato un nuovo metodo per riaggiornare, ogni volta che si cambia valore, lo screen.  Inoltre, ogni volta che si accedeva alla schermata di modifica, tramite il pulsante “Modify”, o si modificava la tolleranza tramite lo slider veniva creato un nuovo schermo sopra a quello istanziato precedentemente. Grazie al docente abbiamo risolto questo problema cambiando il metodo richiamato e ora ci spostiamo semplicemente tra i vari screen.  Problemi a ottenere le dimensioni dell’immagine usando l’attributo size/texture, non davano la grandezza attuale ma solo quella iniziale. Risolto utilizzando l’attributo norm\_image\_size che ritorna la grandezza attuale dell’immagine.  Per colorare il pixel premuto ci sono stati problemi perché opencv leggeva l’immagine mettendo lo 0 in alto a sinistra mentre il metodo touch, che serve a prendere la posizione di dove è stato premuto il mouse sullo schermo, mette lo 0 in basso a sinistra, per questo abbiamo dovuto inserire dei calcoli per la coordinata y per poterla adattare a opencv. |

|  |
| --- |
| Punto della situazione rispetto alla pianificazione |
| Stiamo seguendo la pianificazione |

|  |
| --- |
| Programma di massima per la prossima giornata di lavoro |
| Completare l’algoritmo per la scelta delle parti dell’immagine da mantenere e migliorare la GUI |