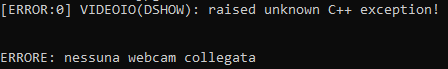
# Lavori svolti

|  |  |  |  |
| --- | --- | --- | --- |
| Orario | lavoro svolto | persone | ID Lavoro |
| 8:20 – 9:00 | Implementato gestione errori | Matteo, Bruno | 1 |
| 9:00 – 16:30 | Documentazione | Matteo | 2 |
| 8:20 – 16:30 | Script di installazione linux | Jonas | 3 |
| 8:20 – 16:30 | Implementate thread esecuzione Python | Luca, Bruno (dalle 9:00) | 4 |

# Lavori

1. Implementato il codice per fare in modo che se qualcuno avvia il programma senza avere una webcam ritorni un errore. Per fare questo abbiamo utilizzato un semplice try con un except (generale perché tanto quello è l’unico errore che possa avere). Adesso se si avvia il programma senza webcam esso ritorna queste due linee di errore:
2. Ho creato uno script per installare le varie dipendenze e ho lavorato soprattutto su python e java.
3. Su consiglio del perito abbiamo implementato una struttura multithreaded per la classe faceLock.py, permettendole di aggiornare le impostazioni, controllare la presenza di un volto conosciuto e bloccare il pc in modo autonomo, l’uno dall’altro.

Ciò è stato fatto usando la classe *Threading*, che permette di trasformare una funzione in una thread indipendente.

# Metodo richiamato per creare e fare partire le Threads del programma

def main(self):

updateSettingsThread = threading.Thread(target=self.updateSettings)

faceCheckThread = threading.Thread(target=self.checkFace)

lockThread = threading.Thread(target=self.lock)

updateSettingsThread.start()

faceCheckThread.start()

lockThread.start()

Per fare ciò è stato necessario inoltre dare una struttura più simile a quella di una classe a faceLock.

In questo modo, quando il pc viene bloccato e quando le impostazioni vengono aggiornate, non c’è bisogno di riavviare il codice.

# Problemi riscontrati

3) L’installazione di Java richiedeva un’accettazione dei termini dall’utente e bloccava lo script quindi ho dovuto cercare una soluzione per bypassarla.

Sto avendo problemi ad installare python e le sue librerie nelle versioni corrette.

4) Abbiamo avuto difficoltà nel capire il funzionamento di *self* in Python.

# Programma di massima per la prossima giornata di lavoro

Luca: Implementare l’esecuzione automatica del python

# Punto rispetto alla pianifica

Estremamente avanti.