RISULTATI

I risultati saranno messi a video con l’ausilio delle Map: esse avranno

-La chiave, che corrisponderà al valore della disparità

-Una stringa, “indovinate/totale”: totale dipende dall’algoritmo che calcola le disparità, mentre indovinate dal test (servirà una classe a parte per questa stringa fatta di 2 interi e “/”) -> singleResult.toString()

Ogni volta che l’algoritmo calcola 3 disparità, si incrementa il valore di totale per quella chiave di 1 e, se il candidato la indovina, si incrementa anche indovinate.

Oss: ci saranno MAX\_DISPARITY+1 chiavi (una anche per il valore 0, che indica l’assenza dell’immagine).

ALGORITMO PER CALCOLARE LE DISPARITà

Si prende un massimo, un minimo ed un valore medio. Gli altri 3 valori saranno messi a 0 (nessuna immagine sarà mostrata per quelle imageView che hanno associate tale disparità).

Inizialmente il massimo sarà fornito in input, mentre il minimo sarà fisso a 1 (vettore iniziale: 10-5-1).

ALGORITMO PER UNSORT IL VETTORE

Il vettore [disp1 disp2 disp3 0 0 0] sarà disordinato a partire dalle 2 vettori [disp1 disp2 disp3] calcolate e [0 0 0] con un algoritmo. Servono

-i 2 vettori

-2 contatori che contano quanti valori di ciascun vettore sono stati inseriti nel vettore finale

-il metodo della classe Random per generare un valore tra 0 e 1:

-se 0, inserisci uno 0

-se 1, inserisci un valore di disparità calcolato

Ciò viene fatto finchè non si riempie il vettore ( 6 ).

->A questo punto ho: disparità calcolate, vettore di 6 disparità disordinate, Map aggiornate sul totale

ESECUZIONE TEST

Quando viene eseguito il test bisogna verificare quali imageview sono state viste:

-Se all’interno c’è un valore di disparità, aggiornare la mappa per quella chiave

-Se all’interno c’è 0, aggiornare la mappa corrispondente (ad ogni test la chiave 0 vede il totale chiesto aumentare di 3 e indovinate a seconda dell’esecuzione del test).

COME PROSEGUIRE IL TEST

Il test prevede 6 imageView: 3 contenenti un’immagine con una certa disparità e 3 vuote.

-Se il test viene fatto correttamente (vengono viste solo immagini) si va avanti con il test per via di una sorta di ricerca dicotomica. Si prende il valore di disparità visto più basso ed il valore di disparità non visto più alto e si propone un nuovo test, fino al raggiungimento di una situazione in cui nel vettore ci sono 2 valori di disparità uguali e 1 diverso (come in provaVettore).

A qusto punto:

-Se vengono viste solo le 2 disparità uguali si termina il test

-Se vengono viste tutte e 3 le disparità si propone un nuovo test con NuovoMassimo=VecchioIntermedio e NuovoMinimo=1

-Si può essere flessibili se viene premuta un’immagine vuota, ma non troppo:

-se immagini vuote viste > 1 si torna indietro con NuovoMassimo=MAXDISPARITY e NuovoMinimo=VecchioMassimo

-se immagini vuote viste = immagini viste torno indietro con nuovoMassimo = MAXDISPARITY e NuovoMinimo=VecchioMassimo

-altrimenti si può proseguire incrementando le varie chiavi come prima (immagini vuote viste = 0 oppure = 1, ma immagini viste >= 2 ).

If (immagini con 0 cliccate >=2 || immagini con 0 cliccate == immagini con disparità cliccate) -> nuovo test (MAX, vecchio max)

If (test corretto ed immagini con 0 cliccate <=1) -> avanza con il test (nuovo max, nuovo min)

If (test scorretto) -> nuovo test (MAX, vecchio max)

If (! firstPhase)

-> se test giusto (2 disparità) fai fare 3 tentativi,

-> se 3 disparità, avanza con il test

-> se sbagliato, diminuisci tentativi o ritorna indietro con il test

RISULTATI

A video verranno mostrati, per ogni valore di disparità, la relativa stringa: in questo modo si potrà capire come è stato l’andamento del test:

Dati rilevanti:

-numero di volte che una disparità è stata indovinata rispetto a quanto è stata chiesta (%)

-valore della chiave 0

Se una chiave ha valore 100%, allora la disparità è certificata.

In funzione del valore della chiave 0, invece, si valutano le altre disparità: questo serve per impedire la casualità del test (premendo sempre tutte le immagini tutte le disparità chieste avrebbero il rateo del 100%).

L’obiettivo è minimizzare il valore della chiave 0, così che si possa comprendere che il test è stato fatto correttamente