# Esercitazione 2 di Grafica Computerizzata

Bimbati Alan, Ceravolo Fabrizio, Dainese Andrea,

# Contents

*Un designer sa che ha raggiunto la perfezione*
*non quando non cé piú niente da aggiungere,*
*ma quando non cé piú niente da togliere!*
*Antoine de Saint-Exupéry*

# 1 Implementazione

## 1.1 Esercitazione 1 - Realizzazione di una mano meccanica

### 1.1.1 Creazione della mano

La creazione di una mano meccanica parte dalla creazione di parallelepipedi come falangi, o meglio come dita. Infatti **partendo da un parallelepipedo abbiamo creato ogni dito spaziato di una costante con un altro**.
Ogni dito contiene una parte "Upper" e una parte "Under" per indicare la falange superiore ed inferiore, alla fine della creazione di tutte le falangi abbiamo collegato la parte superiore gerarchicamente con la parte inferiore, cosí che **muovendo la falange inferiore anche quella superiore, essendo "attaccata", si muove**. Ecco un esempio di come é stato creato il dito indice (`UnderIndex.add(UpperIndex);`)
Infine abbiamo creato un parallelepipedo piú grande che rappresenta il polso, che ricolleghiamo con tutte le dita seguendo lo stesso principio della falange superiore con quella inferiore (`Hand.add(UnderIndex);`)

### 1.1.2 Movimenti della mano

Attraverso ad uno *switch* e la tabella ASCII, abiamo collegato ad ogni tasto una rotazione verso z[1]. Per evitare che il braccio si rompa, *if* abbiamo inserito delle limitazioni:
Falange superiore verso l'alto di **30 gradi**
Falange superiore vero il basso di **90 gradi**
Falange inferiore verso l'alto di **60 gradi**
Falange inferiore vero il basso di **90 gradi**

Di seguito é riportata una tabella con i tasti per muovere le dita associate.

| Dito | Tasto1 | Tasto2 |
|---|---|---|
| Pollice | 1 | 2 |
| Indice Superiore | Q | A |
| Indice Inferiore | W | S |
| Medio Superiore | E | D |
| Medio Inferiore | R | F |
| Annulare Superiore | T | G |
| Annulare Inferiore | Y | H |
| Mignolo Superiore | U | J |
| Mignolo Inferiore | I | K |
| Mano (asse y) | O | L |
| Mano (asse x) | Z | X |
| Mano (asse z) | N | M1 |

[1] in realtá la rotazione é in y, ma nel codice utilizziamo z perché la rotazione avviene come perno in asse z, e come effetto in asse y

### 1.1.3 Altri accorgimenti

Volevamo creare anche il pollice, ma l'idea é stata scartata perché alla creazione avendo un valore diverso delle assi, l'asse y é invertito, ed é quindi diverso da zero, la rotazione seguiva come perno il punto 0 con un effetto *snodato*. Infatti a quel punto la falange superiore si muoveva in modo diverso dalle altre.
Per questo motivo abbiamo cancellato il dito oponibile, ma con piú tempo avremo potuto *"evolverci"* e creare il dito oponibile.

## 1.2 Esercitazione 2 - Sfera rimbalzante

Abbiamo creato un cubo trasparente al centro degli assi per visualizzare meglio la sfera posizionata all'interno, piú precisamente, nell'angolo in alto a destra del cubo.
Abbiamo creato un **HUD** dove é possibile visualizzare e settare la **velocitá di rimbalzo** sui tre assi cardinali (valore compreso tra 0 e 1).
La funzione di rimbalzo é una funzione incrementale:

```
step+=controls.bouncingspeed;
sphere.position.asse=L/2-1-(12*Math.abs(Math.sin(step)));
//L/2 = meta lato del cubo
```

Inoltre abbiamo aggiunto con i tasti W, A, S, D, Q, E, la rotazione del cubo per **visualizzare il rimbalzo della scena in tutte le prospettive possibili**

## 1.3 Esercitazione 3 - Material Show

Abbiamo strutturato il **main** in vari blocchi definiti da funzioni init e animate. Le funzioni principali che meritano una descrizione sono:

1. **init**, questa funziona inizializza la camera, la luce e la GUI.
2. **setupGui**, crea la GUI con le forme, i materiali e i colori del progetto. Di default si utilizza un cubo di colore smeraldo con tutte le luci attive.
3. **reset**, questa funzione semplicemente rimuove tutti i materiali, risulta molto comoda ogni volta che si vuole aggiungere un nuovo materiale, infatti il risultato sará un nuovo colore, ma in realtá la funzione reset toglie tutti i materiali, cosí poi da aggiungere solo quello interessato e rendere il programma leggero da eseguire.
4. **animate**, renderizza le immagini frame per frame.

Come descritto dalla consegna, abbiamo inserito tutti i valori della tabella per i materiali e le luci in degli array, in modo da rendere piú semplice la scelta del materiale o della luce.