**Università degli studi di Udine**

**Corso di Immagini e Multimedialità – 2018-19**

prof. Vito Roberto

**Relazione finale**

**Zanutta Riccardo**

Matricola 120169

# **ESERCIZIO 3.1**

**Traccia dell’esercizio**

Creare una scena con un color cube.

1. **Effettuare le traslazioni lungo i singoli assi x, y, z**

 Mantenendo fisso il punto di vista, effettuare le traslazioni:

* + Per valori positivi della x;
  + Per valori negativi della x;
  + Per valori positivi della y;
  + Per valori negativi della y;
  + Per valori positivi della z;
  + Per valori negativi della z;

Effettuare una traslazione in modo che il cubo sia visto in basso a dx e risulti più piccolo.

1. **Effettuare le rotazioni attorno ai singoli assi x, y, z**

 Mantenendo fisso il punto di vista effettuare le rotazioni:

* + Per valori positivi della x (provare con 90°, 180°, …);
  + Per valori negativi della x;
  + Per valori positivi della y;
  + Per valori negativi della y;
  + Per valori positivi della z;
  + Per valori negativi della z.

1. **Effettuare le scalature rispetto ai singoli assi x,y,z**

Mantenendo fisso il punto di vista effettuare le scalature:

* + Per valori positivi della x;
  + Per valori negativi della x;
  + Per valori positivi della y;
  + Per valori negativi della y;
  + Per valori positivi della z;
  + Per valori negativi della z.

**3.1.1 – Schermate Traslazioni**

|  |  |  |
| --- | --- | --- |
|  | **Valori positivi** | **Valori negativi** |
| **Coordinata** **x** |  |  |
| **Coordinata y** |  |  |
| **Coordinata z** |  |  |

In sintesi:

* Per valori positivi della coordinata x l’oggetto viene spostato a **destra**, per valori negativi a **sinistra**;
* Per valori positivi della coordinata y l’oggetto viene spostato verso **l’alto**, per valori negativi verso il **basso**;
* Per valori positivi della coordinata z l’oggetto viene **avvicinato**, per valori negativi viene **allontanato**.

Effettuare una traslazione in modo che il cubo sia visto in basso a dx e risulti più piccolo.

**3.1.1 - Codice Traslazioni**

**3.1.2 – Schermate Rotazioni**

|  |  |  |
| --- | --- | --- |
|  | **Valori positivi** | **Valori negativi** |
| **Coordinata** **x** |  |  |
| **Coordinata y** |  |  |
| **Coordinata z** |  |  |

**3.1.2 – Codice Rotazioni**

**3.1.3 – Schermate Scalature**

|  |  |  |
| --- | --- | --- |
|  | **Valori positivi** | **Valori negativi** |
| **Coordinata** **x** |  |  |
| **Coordinata y** |  |  |
| **Coordinata z** |  |  |

**3.1.3 – Codice Scalature**

# **ESERCIZIO 3.2**

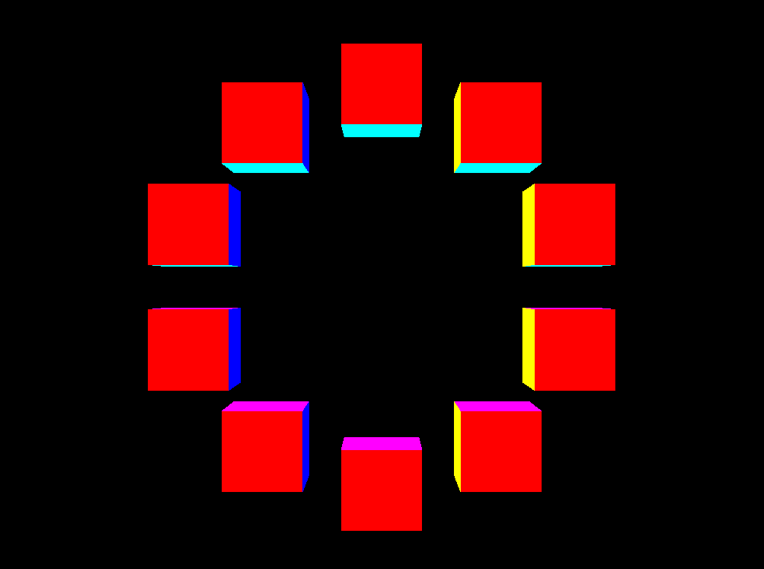
**Traccia dell’esercizio**

Utilizzando le trasformazioni, creare una scena con un numero arbitrario di cubi (diversi) disposti a cerchio.

**3.2 – Processo**

****Ho utilizzato una classe esterna *CubesInCircle* per rappresentare i cubi e distribuirli proporzionalmente entro le dimensioni di una sfera: applico una trasformazione proporzionale al numero di cubi.

**3.2 – Implementazione e risultato**



# **ESERCIZIO 3.3**

Utilizzare ***lookAt()*** per mostrare una scena da diversi punti di vista.

Trovare trasformazioni da applicare al ***ViewPlatform*** per cui si hanno:

1. 1 punto di fuga;
2. 2 punti di fuga;
3. 3 punti di fuga;

**3.3 – Processo**

Ho creato una funzione separata per traslare il punto di vista dell’osservatore.

|  |  |  |
| --- | --- | --- |
| 1 punto di fuga | 2 punti di fuga | 3 punti di fuga |
|  |  |  |

**3.3 - Implementazione**

# **ESERCIZIO 3.4**