المحاضرة الأولى

VRML ← هي لغة نصية اي تتكون من نص متخصصة في تصميم مجسمات

Virtual Reality Modeling language لاظهار ثلاثية الأبعاد وهي اختصار ل

اشكال الأساسية او رئيسية ه- 3D Model ه-

#Cylinder

Shape {

مظهر ← appearance Appearance { اي تنفيذ على مظهر ويتم كتابتها

Material Material {} في جزء هنا

}

geometry cylinder { نفذ بتعريف Compiler

height 2.0 على انه شكل هندسي

radius 1.5

} auto بنفذ عليها برنامج

}

}

Box {

Size 2.0 0.5 3.0

}

height

width

depth
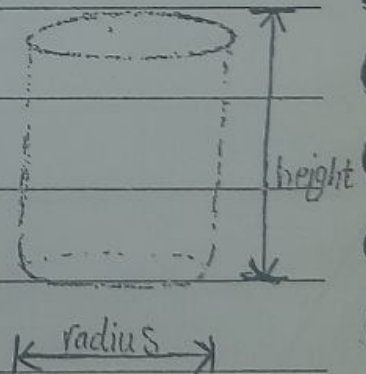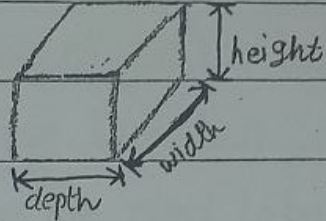
Sphere {

radius 1.0

}

Sphere شكله ياخذ كرة
فقط

Cone {

height 3.0

bottom Radius 0.75

}

height

bottom radius
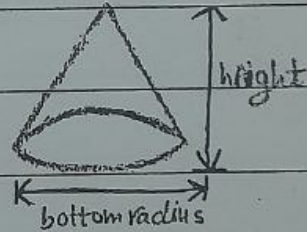
Text LIBYA   تكتب انه اضافة text

Transformation

تعني لانه يكون نفس
شكل ولكن بحجم مجوه   معايرة   Scaling   دوران   rotation   ازاحة   Translation.

نظرا نستعمل
محدوفة

3D Cortona ← Viewer يعرض عنه مجسم على web page مطلوب مني منه الجزئية، download s/w unity /unity hub، Laptop، VS

Wrl امتداد ← VRML

diffuse Color ← يتم من خلالها تغير لون شكل ويكون عن طريق RGb

0.0	0.0	0.0	→ اسود
0.1	0.1	0.1	→ ابيض
0.0	0.1	0.0	→ اخضر
0.1	0.0	0.0	→ احمر
0.0	0.0	0.1	→ ازرق
0.1	0.1	0.0	→ اصفر

G

green (0,1,0)   yello (110)

gray axis   white(1,1,1)

R

Black (0,0,0)   Red (1,0,0)

Blue (0,0,1)

B
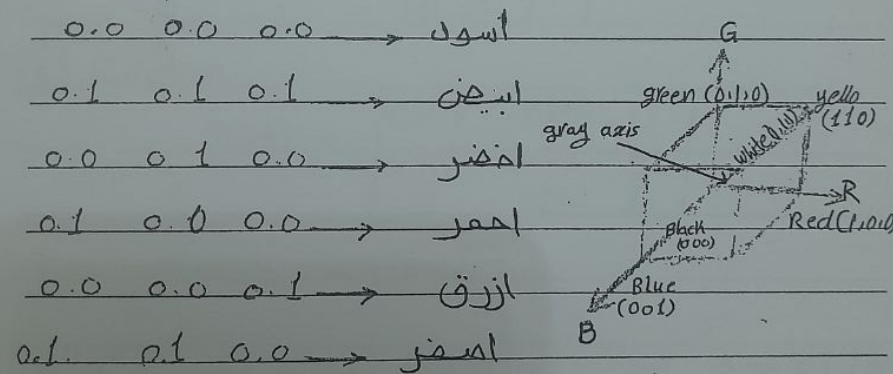
Defination of the ray parametric Form

$$R_o = \{ x_o \ y_o \ z_o \} \text{ (Ray origin)}$$

$$R_d = \{ x_d \ y_d \ z_d \} \text{ (Ray Direction}$$

direction of the ray we need to have

unit Vector $x_d^2 + y_d^2 + z_d^2 = 1$

How are transform homogeneous coordinates ?

[1] Translate

$$T(d_x, d_y) = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix}$$

[2] scale

$$S(s_x, s_y) = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[3] Rotation

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
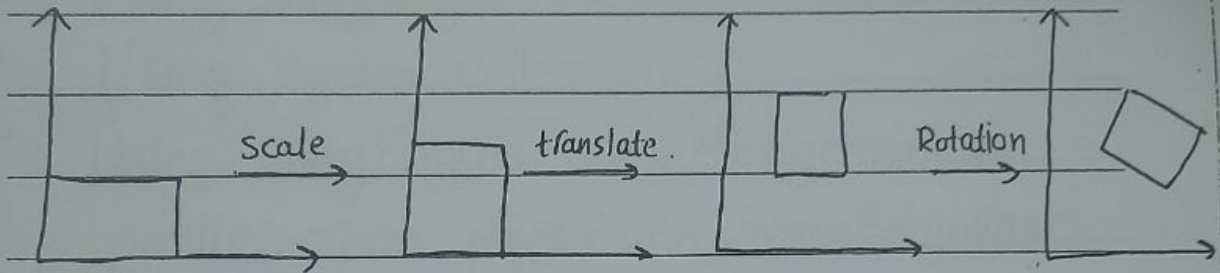
How are transforms Combined? scale, Translate, Rotation

Use Matrix multiplication $p' = T(sp) = Tsp = Ts$

[1] scaling

$x' = x \cdot Sx$

$y' = y \cdot Sy$

$\Rightarrow$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \cdot \begin{bmatrix} Sx \\ Sy \end{bmatrix}$$

[2] Translation

$x' = x + Tx$

$y' = y + Ty$

$\Rightarrow$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} Tx \\ Ty \end{bmatrix}$$

[3] Rotation

$x' = x \cdot \cos\theta - y \cdot \sin\theta$

$y' = x \cdot \sin\theta + y \cdot \cos\theta$

$\Rightarrow$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \cdot$$

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$P' = T + P$, $P' = R \cdot P$, $P' = S \cdot P$

Demonstrate projection in ray tracing, use
mathematical operations and matrixes?

يؤدي ضرب النقطة p=$[x\ y\ z\ 1]^T$ مع المصفوفة Mper
الى الحصول على نقطة جديدة متجانسة $[x, y, z, w]^T$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = Mper \cdot p = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$[x\ y\ z\ w]^T = [x\ y\ z\ z/d]^T$$

الان نقوم بالقسمة على w التي هي z/d واسقاط
احداثي الرابع للعودة الى 3D

$$\left[\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right] = (x_p, y_p, z_p) = \left[\frac{x}{z/d}, \frac{y}{z/d}, d\right]$$

$$\frac{x_p}{d} = \frac{x}{z+d}, \quad \frac{y_p}{d} = \frac{y}{z+d}$$

بالضرب في d

$$x_p = \frac{d \cdot x}{z+d} = \frac{x}{(z/d)+1}, \quad y_p = \frac{d \cdot y}{z+d} = \frac{y}{(z/d)+1}$$

```
Using unityEngine;
Using UnityEngine.UI;
Using System.Collections;
Using UnityEngine.SceneManagement;
public class playerHealth : monoBehaviour
{
        public int starthealth = 100;
        public int currenthealth;
        public slider healthSlider;
        AudioSource playerAudio;
        PlayerMovement playerMovement;
        playershooting playershooting;
        Bool isDead;
        bool damaged;
        public void TakeDamage (int amount)
        {

            damaged = true;
```
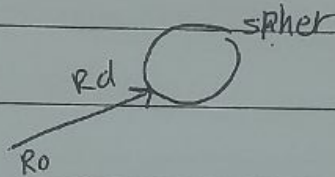
Calulate the intersection between a ray and sphere and find out the normal at the intersection point and solve?

Difination of the ray parametric Form

$R_0 = \{x_0 \ y_0 \ z_0\}$　　　$\{ray\ origine\}$

$R_d = \{x_d \ y_d \ z_d\}$　　　$\{ray\ direction\}$

Direction of ray we need one vector

$x_d^2 + y_d^2 + z_d^2 = 1$

parametric Form

$R_t = R_0 + R_d t$　　　$t > 0$

implicit Form

center $S_c = \{x_c \ y_c \ z_c\}$

Rudius S_r

Surface point $= \{x_s \ y_s \ z_s\}$

implicit equation for sphere

$(x_s - x_c)^2 + (y_s - y_c)^2 + (z_s - z_c)^2 = sr^2$

Write an unity C# script to move the player around
the room and jump when it find an object or
Wall. The values of the rigidbody?

```csharp
using    UnityEngine;
using    System. Collections;
public class moveplayer : monoBehaviuor
    {

        public Float spead;
        public float Jumphight;
        public Rigidbody rd;
        private vector3 direction;
        void start ()
        {

        direction = new vector (Input. GetAxis ("Horizon-
        ),0, Input. GetAxis ("vertical"));

        {
```

```
private void FexidUpdate()
{
    rd.move.position(transform.position + direction *
    spead * time.DaltTime());
    IF(Input.GetButtonDown("jump"))
    {
        rd.AddForce(vector3.up * Jumphight,
        ForceMode.velocitychange);
    }
}
}
```

Write a Unity C# script to controll the shooting
of balls through the gun and calculate how
many times has the target hit?
The sound and time classes must included.

```
using    UnityEngine;
using    System. Collections;
public class  Collision : monoBehaviour
{
    private void onCollisionEnter (Collision Collision)
    {
        If(obl.gameobject. comparetag (bullets))
        {
            Debug. Log ("player hit")
        {
        Time. DaltaTime ()
        {
{
```

Write an unity program to determine the next level and to kill an enemy?

```
using     UnityEngine;
using     System.Collections;
public class Killenemy : monoBehaviour
{
    public  int  CurrentLevel;
    public  int   EnemyCount;
    public  int   enemy Kill Count;
    void start ()
    {
        SpawnEnemy ();
    {
    void Update ()
    {
        If (enemy Kill Count == enemy count)
        {
```

```
        currentLevel ++;
        enemyCount ++;
        enemyKillCount = 0;
        spawnEnemy();
    }

    void spawnEnemy()
    {

    }

public void killEnemy()
    {
        enemyKillCount ++;
    }
}
```

Explain how to let the main camera moving
with the player ?

```
Using   UnityEngine ;
Using   System. Collections ;
public  class moving cam : monoBehaviuor
   {
   public float SENSTIVITY ;
   public float XRotation ;
   public float YRotation ;
   void start ()
      {

      ;  {
   void update ()
      {

      XRotation += Input.GetAxis ("mouse x")
         * SENSTIVITY ;
```

Explain how to let the main camera moving with the player?

```
Using   UnityEngine;
Using   System. Collections;
public  class movingcam : monoBehaviuor
{
    public float SENSTIVITY;
    public float XRotation;
    public float YRotation;
    void start()
    {

    };
    void update()
    {
        XRotation += Input.GetAxis("mouse x")
        * SENSTIVITY;
```

```
YRotation -= Input.GetAxis("mouse Y") * SENSTIVITY;
transform.eulerAngles = new Vector3(0, XRotation, 0);
```

```
CurrentHealth = amount;
healthSlider.value = CurrentHealth;
if (CurrentHealth <= 0 && !isDead)
{
    Death();
}
}

Void Death()
{
    isDead = true;
    playershooting.DisableEffects();
    Playermovement.enabled = false;
    playershooting.enabled = false;
}

public void Restartlevel()
{


}
}
```

```csharp
Using UnityEngine;
Using System.Collections;
public class mainmenu : monoBehaviour
{
    public void play()
    {
    }
    {
    public void Quit()
    {
        Application.Quit();
        Debug.Log("player has quit the game");
    }
        ;
```

```
Using    Unity Engine;
Using    System.Collections;
public  class shot {
        public Float upForce;
        public Float hitForce;
    }

public class ShotManager: monoBehaviour
    {
        public shot topspin;
        public shot Flat;
    }
```

Explain the main purpose of physical material,
Dynamic function, Static function Bounciness?

الغرض الرئيسي من مواد فيزيائية هو محاكاة خصائص

الفيزيائية للاشياء في العالم افتراضي

- الاحتكاك الديناميكي : يشير الى الاحتكاك الذي يحدث

بين جسمين عندما يكون في حركة

- الاحتكاك الثابت : هو الاحتكاك الذي يحدث بين جسمين

عندما يكونوا ثابتين

- الانعكاسية تحدد كمية الطاقة التي يحتفظ بها الجسم

Write an Unity C# script to control the
mouse and the keyboard ?

تعريفة