

## 1st Programme: Transformation for 3 game Objects

```
using UnityEngine;

public class GeoTransform : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float rotateSpeed = 50f;
    public float scaleSpeed = 0.5f;

    public static bool globalControl = false; // Toggle for common key
movement

    void Update()
    {
        // Toggle global mode using Spacebar
        if (Input.GetKeyDown(KeyCode.Space))
        {
            globalControl = !globalControl;
        }

        if (globalControl)
        {
            GlobalMovement(); // Move all objects together
        }
        else
        {
            IndividualMovement(); // Move objects separately
        }
    }

    void GlobalMovement()
    {
        // Move all objects using 1 (Forward), 2 (Backward), 3 (Left), 4
(Right)
        float moveX = 0, moveZ = 0;
        if (Input.GetKey(KeyCode.Alpha1)) moveZ += moveSpeed *
Time.deltaTime;
        if (Input.GetKey(KeyCode.Alpha2)) moveZ -= moveSpeed *
Time.deltaTime;
```

```

        if (Input.GetKey(KeyCode.Alpha3)) moveX -= moveSpeed *
Time.deltaTime;
        if (Input.GetKey(KeyCode.Alpha4)) moveX += moveSpeed *
Time.deltaTime;
        transform.Translate(new Vector3(moveX, 0, moveZ));

        // Rotate all using 5 (Left) and 6 (Right)
        if (Input.GetKey(KeyCode.Alpha5)) transform.Rotate(Vector3.up,
-rotateSpeed * Time.deltaTime);
        if (Input.GetKey(KeyCode.Alpha6)) transform.Rotate(Vector3.up,
rotateSpeed * Time.deltaTime);

        // Scale all using 7 (Increase) and 8 (Decrease)
        if (Input.GetKey(KeyCode.Alpha7)) transform.localScale +=
Vector3.one * scaleSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.Alpha8)) transform.localScale -=
Vector3.one * scaleSpeed * Time.deltaTime;
    }

    void IndividualMovement()
    {
        if (gameObject.name == "Cube")
        {
            MoveCube();
        }
        else if (gameObject.name == "Sphere")
        {
            MoveSphere();
        }
        else if (gameObject.name == "Plane")
        {
            MovePlane();
        }
    }

    void MoveCube()
    {
        // Cube moves using WASD
        float moveX = 0, moveZ = 0;
        if (Input.GetKey(KeyCode.W)) moveZ += moveSpeed * Time.deltaTime;

```

```

        if (Input.GetKey(KeyCode.S)) moveZ -= moveSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.A)) moveX -= moveSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.D)) moveX += moveSpeed * Time.deltaTime;
        transform.Translate(new Vector3(moveX, 0, moveZ));

        // Rotate using Q and E
        if (Input.GetKey(KeyCode.Q)) transform.Rotate(Vector3.up,
-rotateSpeed * Time.deltaTime);
        if (Input.GetKey(KeyCode.E)) transform.Rotate(Vector3.up,
rotateSpeed * Time.deltaTime);

        // Scale using R and F
        if (Input.GetKey(KeyCode.R)) transform.localScale += Vector3.one *
scaleSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.F)) transform.localScale -= Vector3.one *
scaleSpeed * Time.deltaTime;
    }

    void MoveSphere()
    {
        // Sphere moves using UHKJ
        float moveX = 0, moveZ = 0;
        if (Input.GetKey(KeyCode.U)) moveZ += moveSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.J)) moveZ -= moveSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.H)) moveX -= moveSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.K)) moveX += moveSpeed * Time.deltaTime;
        transform.Translate(new Vector3(moveX, 0, moveZ));

        // Rotate using Y and I
        if (Input.GetKey(KeyCode.Y)) transform.Rotate(Vector3.up,
-rotateSpeed * Time.deltaTime);
        if (Input.GetKey(KeyCode.I)) transform.Rotate(Vector3.up,
rotateSpeed * Time.deltaTime);

        // Scale using O and L
        if (Input.GetKey(KeyCode.O)) transform.localScale += Vector3.one *
scaleSpeed * Time.deltaTime;
        if (Input.GetKey(KeyCode.L)) transform.localScale -= Vector3.one *
scaleSpeed * Time.deltaTime;
    }

```

```

void MovePlane()
{
    // Plane moves using ZXCV
    float moveX = 0, moveZ = 0;
    if (Input.GetKey(KeyCode.Z)) moveZ += moveSpeed * Time.deltaTime;
    if (Input.GetKey(KeyCode.V)) moveZ -= moveSpeed * Time.deltaTime;
    if (Input.GetKey(KeyCode.X)) moveX -= moveSpeed * Time.deltaTime;
    if (Input.GetKey(KeyCode.C)) moveX += moveSpeed * Time.deltaTime;
    transform.Translate(new Vector3(moveX, 0, moveZ));

    // Rotate using B and N
    if (Input.GetKey(KeyCode.B)) transform.Rotate(Vector3.up,
-rotateSpeed * Time.deltaTime);
    if (Input.GetKey(KeyCode.N)) transform.Rotate(Vector3.up,
rotateSpeed * Time.deltaTime);

    // Scale using M and <
    if (Input.GetKey(KeyCode.M)) transform.localScale += Vector3.one *
scaleSpeed * Time.deltaTime;
    if (Input.GetKey(KeyCode.Comma)) transform.localScale -=
Vector3.one * scaleSpeed * Time.deltaTime;
}
}

// Controls
// ☒ Individual Mode (Default)
// Cube

// Move: W, A, S, D
// Rotate: Q (Left), E (Right)
// Scale: R (Increase), F (Decrease)
// Sphere

// Move: U, H, K, J
// Rotate: Y (Left), I (Right)
// Scale: O (Increase), L (Decrease)
// Plane

```

```
// Move: Z, X, C, V
// Rotate: B (Left), N (Right)
// Scale: M (Increase) , (Decrease)

// ☒ Global Mode (All Objects Move Together)
// Press Spacebar to toggle Global Mode
// Move all objects using 1 (Forward), 2 (Backward), 3 (Left), 4 (Right)
// Rotate all objects using 5 (Left), 6 (Right)
// Scale all objects using 7 (Increase), 8 (Decrease)
```

## Programme 2: To change the texture, materials and colors

```
using UnityEngine;
using UnityEngine.UI;

public class ObjectChanger : MonoBehaviour
{
    public GameObject[] objects; // Assign Cube, Sphere, and Plane
    public Button colorButton, materialButton, textureButton;
    public Material[] materials; // Assign materials in the Inspector
    public Texture[] textures; // Assign textures in the Inspector

    private int materialIndex = 0;
    private int textureIndex = 0;

    void Start()
    {
        colorButton.onClick.AddListener(ChangeAllColors);
        materialButton.onClick.AddListener(ChangeAllMaterials);
        textureButton.onClick.AddListener(ChangeAllTextures);
    }

    void ChangeAllColors()
    {
        foreach (GameObject obj in objects)
        {
            if (obj != null)
                obj.GetComponent<Renderer>().material.color = new
Color(Random.value, Random.value, Random.value);
        }
    }
}
```

```

    }
}

void ChangeAllMaterials()
{
    if (materials.Length > 0)
    {
        materialIndex = (materialIndex + 1) % materials.Length;
        foreach (GameObject obj in objects)
        {
            if (obj != null)
                obj.GetComponent<Renderer>().material =
materials[materialIndex];
        }
    }
}

void ChangeAllTextures()
{
    if (textures.Length > 0)
    {
        textureIndex = (textureIndex + 1) % textures.Length;
        foreach (GameObject obj in objects)
        {
            if (obj != null)
                obj.GetComponent<Renderer>().material.mainTexture =
textures[textureIndex];
        }
    }
}
}

```

## PROGRAM 4

```
using UnityEngine;
```

```
public class Program4 : MonoBehaviour
```

```
{
```

```
    // Start is called once before the first execution of Update after the MonoBehaviour is created
```

```

public float moveSpeed;
public float rotationSpeed;

void Start()
{
    moveSpeed = 15f;
    rotationSpeed = 100f;
}

// Update is called once per frame
void Update()
{
    float moveX = Input.GetAxis("Horizontal");
    float moveY = Input.GetAxis("Vertical");

    Vector3 move = (Vector3.right * moveX + Vector3.forward * moveY) * moveSpeed *
Time.deltaTime;
    transform.Translate(move);

    float mouseX = Input.GetAxis("Mouse X");
    float mouseY = Input.GetAxis("Mouse Y");

    transform.Rotate(Vector3.up, mouseX * rotationSpeed * Time.deltaTime, Space.World);
    transform.Rotate(Vector3.right, -mouseY * rotationSpeed * Time.deltaTime, Space.World);
}
}

```

## Programme 5: 2D Raycast

```

using UnityEngine;
using UnityEngine.InputSystem;

public class Prog5 : MonoBehaviour
{
    [SerializeField] float speed = 20f;
    [SerializeField] float distance = 10f;

    RaycastHit2D hit;

    void Start()
    {

```

```

        Debug.Log("Press 'Space' to shoot a raycast");
    }

    void FixedUpdate()
    {
        transform.Rotate(Vector3.forward * speed * Time.deltaTime);
        hit = Physics2D.Raycast(transform.position, transform.right,
distance);
        if (hit.collider != null)
        {
            Debug.DrawRay(transform.position, hit.point, Color.red);
            Debug.Log("Hit: " + hit.collider.name);
        }
        else
        {
            Debug.DrawRay(transform.position, transform.right * distance,
Color.green);
            Debug.Log("No hit");
        }
    }
}

```

**PROGRAM 7**

**PROGRAM 8**

```

using UnityEngine;

public class therrun : MonoBehaviour
{
    Animator anim;

    void Start()
    {
        anim = GetComponent<Animator>();
    }

    public void therun()
    {
    }
}

```



```

    {
        anim.SetTrigger("run");
    }
}

```

## PROGRAM 9

### COINCOLLECT

```

using UnityEngine
using System.Collections;
public class CoinCollect : MonoBehaviour
{
    public static int totalCoins = 0; // Points per coin

    void Awake()
    {
        //Make Collider as trigger
        GetComponent<Collider>().isTrigger = true;
        Debug.Log("Awake in CoinCollect");
    }
    void OnTriggerEnter(Collider other)
    {
        Debug.Log("OnTriggerEnter in CoinCollect");
        if (other.CompareTag("Player")) // Ensure only player can collect
coins
        {

            Debug.Log("Destroying coin...");
            // StartCoroutine(DestroyCoin()); // Destroy the coin after
collection
            Destroy(gameObject);
            totalCoins++;
            Debug.Log("Coin collected!" + " Total Coins: " + totalCoins);
        }
    }
}

```

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class SceneLoader : MonoBehaviour
{
    public static SceneLoader instance; // Singleton instance

    void Awake()
    {
        // Initialize the singleton instance
        if (instance == null)
        {
            instance = this;
        }
        else if (instance != this)
        {
            Destroy(gameObject); // Ensure there's only one instance
        }
    }

    public void LoadNextLevel()
    {
        Debug.Log("Loading next scene...");
        int nextSceneIndex = SceneManager.GetActiveScene().buildIndex + 1;
        SceneManager.LoadScene(nextSceneIndex);
    }
}

```

```

using UnityEngine;
using TMPro; // Required for TextMeshPro UI

public class ScoreManager : MonoBehaviour
{
    public TextMeshProUGUI scoreText; // Reference to the UI text component
    private int totalCoinsInScene;

    void Start()
    {

```

```
        totalCoinsInScene =
GameObject.FindGameObjectsWithTag("Coin").Length;
    }
    void Update()
    {

        Debug.Log("ScoreManager Update");
        if (scoreText.text != "Score: " +
CoinCollect.totalCoins.ToString())
        {
            Debug.Log("Updating score...");
            scoreText.text = "Score: " +
CoinCollect.totalCoins.ToString();
        }
        if (CoinCollect.totalCoins == totalCoinsInScene)
        {
            Debug.Log("Calling LoadNextLevel...");
            CoinCollect.totalCoins = 0;
            SceneLoader.instance.LoadNextLevel();
        }
    }
}
```