# UNITY GÜNLÜĞÜ

## FLY CAM KULLANIMI
## SWITCH CAM UYGULAMASI

# UNITY GÜNLÜĞÜ

by yselim

## FLY CAM KULLANIMI

```
public class flyCam : MonoBehaviour {

// Use this for initialization
void Start () {

}
   /*
  Made simple to use (drag and drop, done) for regular keyboard layout
   wasd : basic movement
   shift : Makes camera accelerate
   space : Moves camera on X and Z axis only.  So camera doesn't gain any height*/
    float mainSpeed = 10.0f; //std hız
    float shiftAdd = 150.0f; //shift ile koşturma
    float maxShift = 500.0f; //Maximum hız
    float camSens = 0.55f; //mouse hassasiyeti
    private Vector3 lastMouse = new Vector3(255, 255, 255); //kind of in the middle of the screen, rather than at the
top (play)
    private float totalRun = 1.0f;

    void Update()
    {
        lastMouse = Input.mousePosition - lastMouse;
        lastMouse = new Vector3(-lastMouse.y * camSens, lastMouse.x * camSens, 0);
        lastMouse = new Vector3(transform.eulerAngles.x + lastMouse.x, transform.eulerAngles.y + lastMouse.y, 0);
        transform.eulerAngles = lastMouse;
        lastMouse = Input.mousePosition;
        //Mouse  camera angle done.

        //Klavye komutları
        float f = 0.0f;
        Vector3 p = GetBaseInput();
        if (Input.GetKey(KeyCode.LeftShift))
        {
            totalRun += Time.deltaTime;
            p = p * totalRun * shiftAdd;
            p.x = Mathf.Clamp(p.x, -maxShift, maxShift);
            p.y = Mathf.Clamp(p.y, -maxShift, maxShift);
            p.z = Mathf.Clamp(p.z, -maxShift, maxShift);
        }
        else
        {
            totalRun = Mathf.Clamp(totalRun * 0.5f, 1f, 500f);
            p = p * mainSpeed;
        }

        p = p * Time.deltaTime;
        Vector3 newPosition = transform.position;
        if (Input.GetKey(KeyCode.Space))
        { //If player wants to move on X and Z axis only
            transform.Translate(p);
            newPosition.x = transform.position.x;
            newPosition.z = transform.position.z;
            transform.position = newPosition;
        }
        else
        {
            transform.Translate(p);
        }

    }
```

```
if (Input.GetKey(KeyCode.Space))
        { //If player wants to move on X and Z axis only
            transform.Translate(p);
            newPosition.x = transform.position.x;
            newPosition.z = transform.position.z;
            transform.position = newPosition;
        }
        else
        {
            transform.Translate(p);
        }

    }//Update Sonu

private Vector3 GetBaseInput()
    { //returns the basic values, if it's 0 than it's not active.
        Vector3 p_Velocity = new Vector3();
        if (Input.GetKey(KeyCode.W))
        {
            p_Velocity += new Vector3(0, 0, 1);
        }
        if (Input.GetKey(KeyCode.S))
        {
            p_Velocity += new Vector3(0, 0, -1);
        }
        if (Input.GetKey(KeyCode.A))
        {
            p_Velocity += new Vector3(-1, 0, 0);
        }
        if (Input.GetKey(KeyCode.D))
        {
            p_Velocity += new Vector3(1, 0, 0);
        }
        return p_Velocity;
    }
}
```

# SWITCH CAM UYGULAMASI

```
public class camKontrol : MonoBehaviour {

    public Camera[] cameras;
    private int currentCameraIndex;

void Start()
    {
        currentCameraIndex = 0;

        //ilki hariç tüm kameraları gizle
        for (int i = 1; i < cameras.Length; i++)
        {
            cameras[i].gameObject.SetActive(false);
        }

        //camera dizisine her yeni kamera aktif ediliyor..
        if (cameras.Length > 0)
        {
            cameras[0].gameObject.SetActive(true);
           // Debug.Log("Camera with name: " + cameras[0].camera.name + ", is now enabled");
        }
    }


    void Update()
    {
//C tuşuna basılınca tüm kamera dizisindeki kameralar geziliyor..Sonra en başa dönülüyor..
        if (Input.GetKeyDown(KeyCode.C))
        {
            currentCameraIndex++;
            Debug.Log("C basıldı.Sonraki kameraya geçiyor");
            if (currentCameraIndex < cameras.Length)
            {
                cameras[currentCameraIndex - 1].gameObject.SetActive(false);
                cameras[currentCameraIndex].gameObject.SetActive(true);
             //   Debug.Log("Camera with name: " + cameras[currentCameraIndex].camera.name + ", is now enabled");
            }
            else
            {
                cameras[currentCameraIndex - 1].gameObject.SetActive(false);
                currentCameraIndex = 0;
                cameras[currentCameraIndex].gameObject.SetActive(true);
             //   Debug.Log("Camera with name: " + cameras[currentCameraIndex].camera.name + ", is now enabled");
            }
        }
    }
}
```