BLM0364 Oyun Programlama

Ray & Line

Fizik Ray Yayımı (Raycasting)

Arş. Gör. Muhammed Ömer Faruk Selvi

Bursa Teknik Üniversitesi

Line

public static void DrawLine(Vector3 start, Vector3 end, Color color = Color.white, float duration = 0.0f, bool depthTest = true);

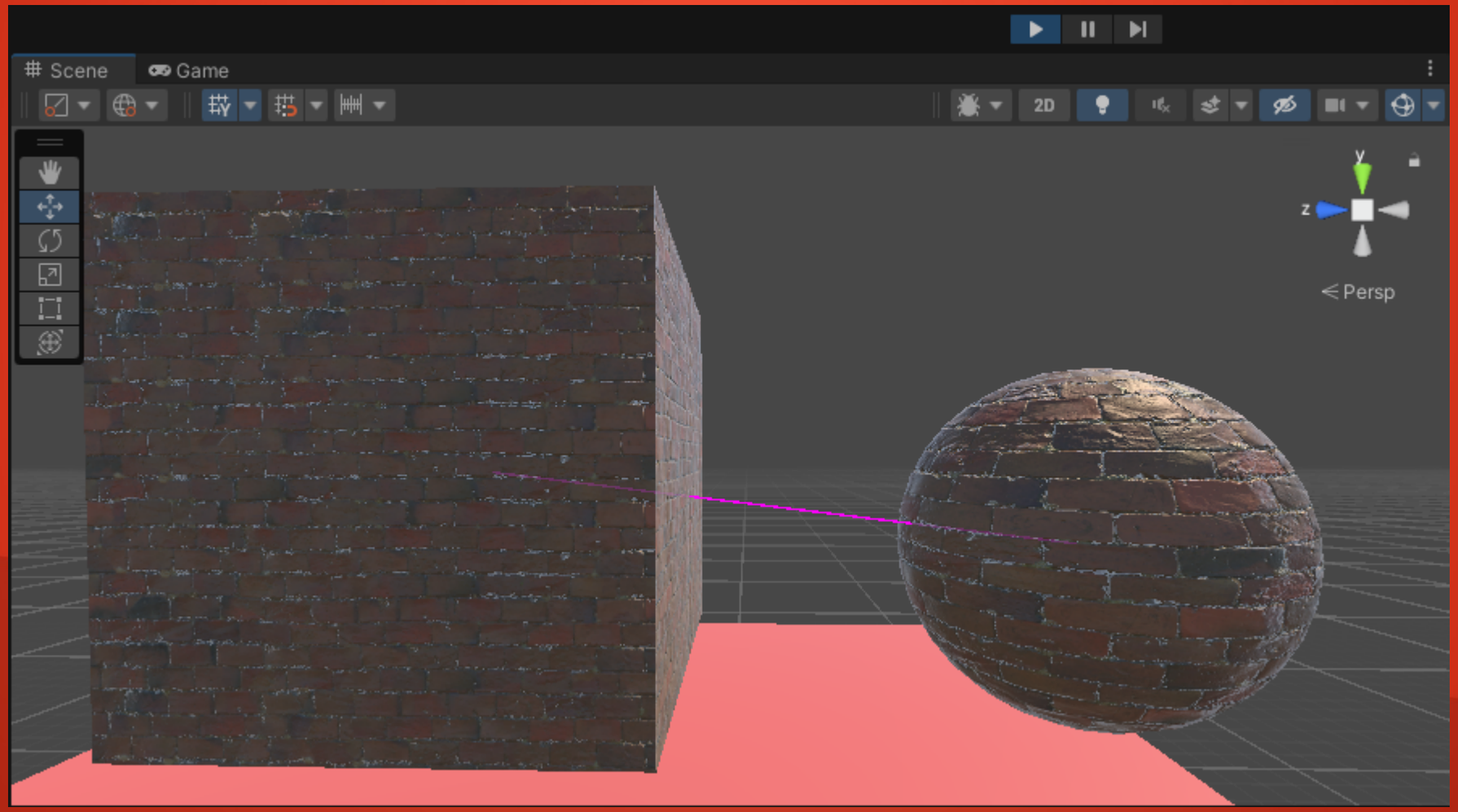
start ve end konumları arasında, Scene penceresinden görülebilen bir çizgi çizer.

duration=0 ise tek kare boyunca çizim yapılır. Saniye türünden verilen süre boyunca çizim ekranda tutulur.

depthTest=false ise tüm nesnelerin üzerinde çizdirilir. True ise nesnelerin arkasında kalan kısımlar daha silik görünür.

```
void Update()
{
  Debug.DrawLine(transform.position, target.position, Color.magenta);
}
```

Line



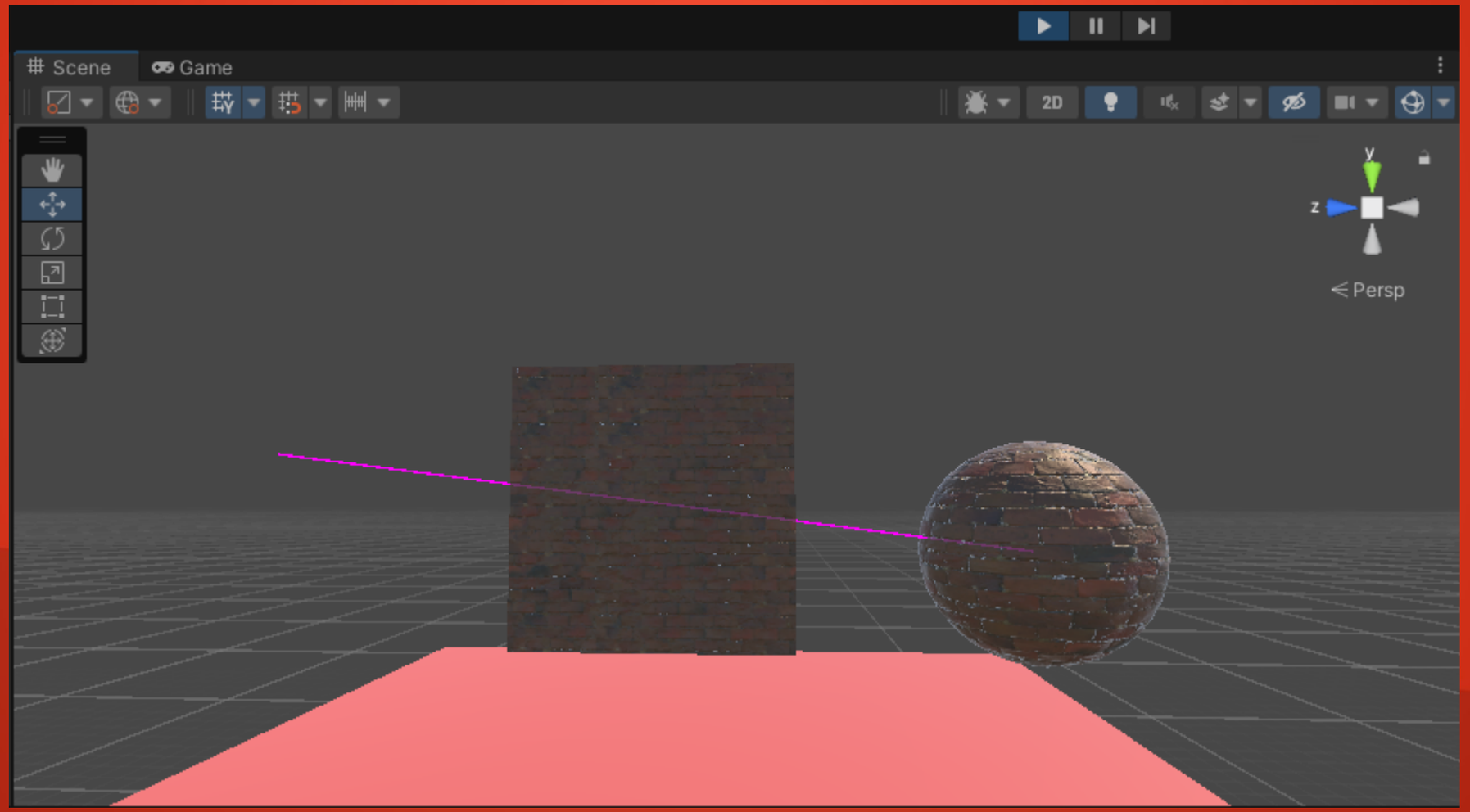
Ray

public static void DrawRay(Vector3 start, Vector3 dir, Color color = Color.white, float duration = 0.0f, bool depthTest = true);

start konumundan dir doğrultusunda bir çizgi çizdirir.

```
void Update()
{
  Debug.DrawRay(transform.position, (target.position-transform.position)*2, Color.magenta);
}
```

Ray



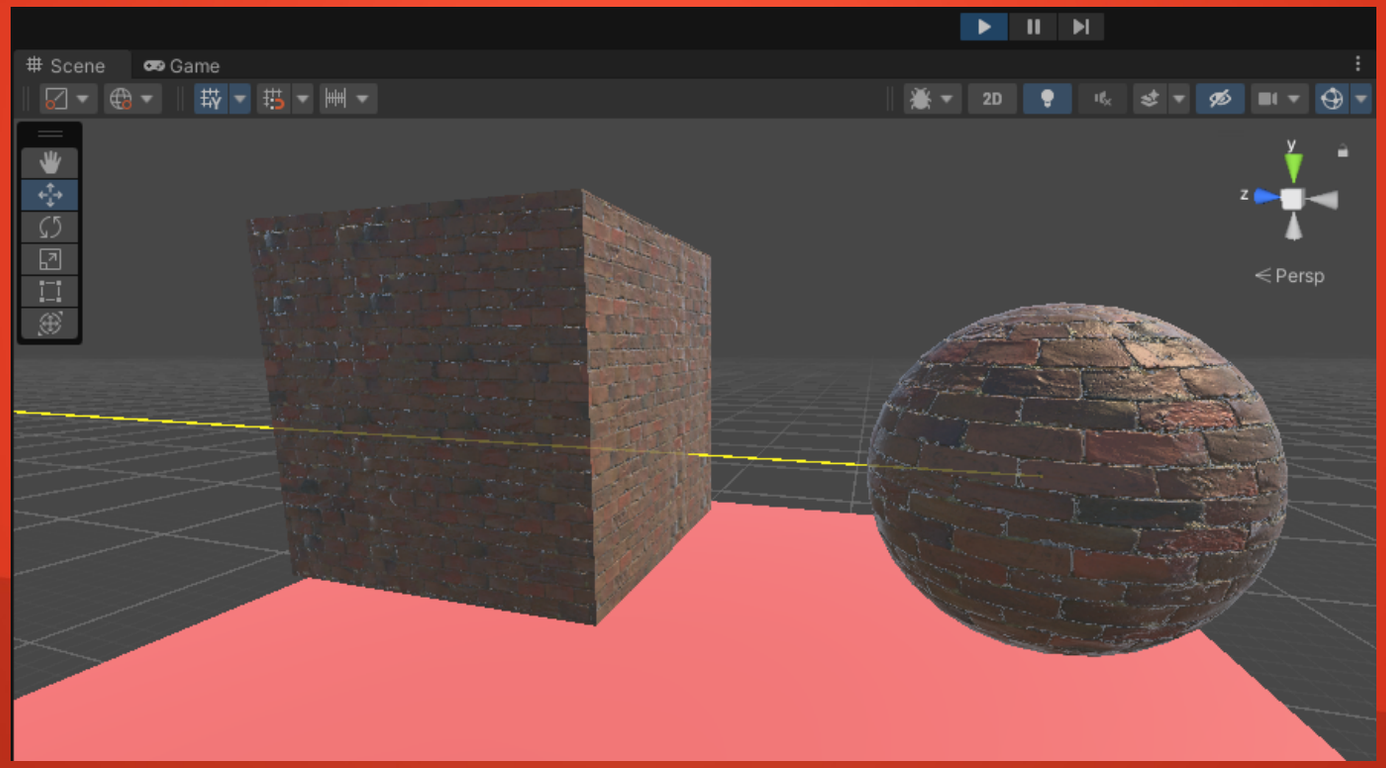
Raycasting

```
public static bool Raycast(Vector3 origin, Vector3 direction, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers,
QueryTriggerInteraction queryTriggerInteraction = QueryTriggerInteraction.UseGlobal);

public static bool Raycast(Vector3 origin, Vector3 direction, out RaycastHit hitInfo, float maxDistance, int layerMask, QueryTriggerInteraction
queryTriggerInteraction);

public static bool Raycast(Ray ray, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers, QueryTriggerInteraction
queryTriggerInteraction = QueryTriggerInteraction.UseGlobal);

public static bool Raycast(Ray ray, out RaycastHit hitInfo, float maxDistance = Mathf.Infinity, int layerMask = DefaultRaycastLayers,
QueryTriggerInteraction queryTriggerInteraction = QueryTriggerInteraction.UseGlobal);
```

Belirtilen ışın doğrultusunda çarpışma testi yapılır ve -varsa- çarpışma bilgileri dışarıya verilir. Yayılan ışın sadece collider'a sahip nesneler ile çarpışır.

```
void FixedUpdate()
{
    RaycastHit hit_info;
    if(Physics.Raycast(transform.position, transform.forward, out hit_info, 100.0f))
    {
        Debug.Log("Işının çarptığı nesne: "+hit_info.transform.name + ", @("+hit_info.point+") ... "+Time.time);
        Debug.DrawRay(transform.position, transform.forward*100.0f, Color.yellow);
    }
}
```

Raycasting



Raycasting


Project
Console
Project Settings
Sprite Editor
Animator
Clear
Collapse
Error Pause
Editor
999+
0
0
[13:22:25] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,3
UnityEngine.Debug:Log (object)
[13:22:25] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,32
UnityEngine.Debug:Log (object)
[13:22:25] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,34
UnityEngine.Debug:Log (object)
[13:22:26] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,36
UnityEngine.Debug:Log (object)
[13:22:26] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,38
UnityEngine.Debug:Log (object)
[13:22:26] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,4
UnityEngine.Debug:Log (object)
[13:22:26] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,42
UnityEngine.Debug:Log (object)
[13:22:26] Işının çarptığı nesne: Cube (1), @((0.51, 11.22, -0.40)) ... 111,44
UnityEngine.Debug:Log (object)

RaycastHit

collider: çarpılan nesnenin Collider komponentinin referansı
rigidbody: -varsa- çarpılan nesnenin Rigidbody komp. referansı
transform: çarpılan nesnenin Transform komp. referansı

point: çarpışma noktasının global eksendeki koordinatları
normal: çarpışmanın gerçekleştiği yüzeyin normal vektörü