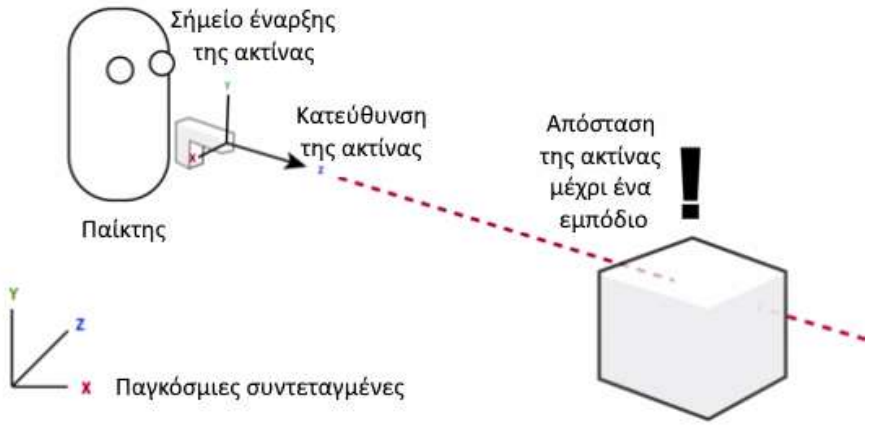
ΜΑΘΗΜΑ 5ο – ΝΟΗΤΕΣ ΑΚΤΙΝΕΣ (RAYS) & ΠΡΟΚΛΗΣΗ ΖΗΜΙΑΣ

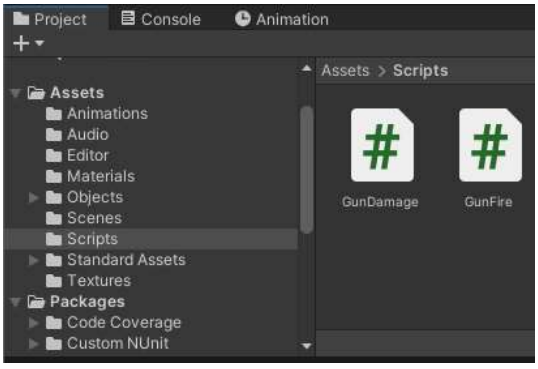
Νοητές ακτίνες

Θέλουμε να δημιουργήσουμε έναν μηχανισμό ο οποίος θα εξομοιώνει τον πυροβολισμό του όπλου και την εκτόξευση μιας σφαίρας από αυτό. Η πορεία της σφαίρας θα ακολουθεί την κατεύθυνση που έχει το όπλο τη στιγμή του πυροβολισμού και θα σταματά εάν αγγίξει έναν εχθρό ή κάποιο άλλο εμπόδιο.

Ένας τρόπος για να επιτευχθεί κάτι τέτοιο είναι να δημιουργήσουμε ένα μικρό 3Δ αντικείμενο (π.χ. μια μικρή σφαίρα) το οποίο θα τοποθετείται αρχικά στην κάνη του όπλου και κατόπιν θα κινείται με μεγάλη ταχύτητα προς την κατεύθυνση που «κοιτάζει» το όπλο. Κάτι τέτοιο θα μπορούσε να πραγματοποιηθεί αν ο βασικός μας στόχος είναι να έχουμε οπτική απεικόνιση της σφαίρας και της πορείας της.

Ένας άλλος τρόπος είναι η χρήση **νοητών ακτίνων (rays)**. Μία νοητή ακτίνα δημιουργείται σε ένα σημείο του 3Δ κόσμου μας και ακολουθεί μία ευθύγραμμη πορεία προς μία κατεύθυνση έως ότου συναντήσει ένα εμπόδιο.





Το Unity διαθέτει έναν τέτοιο μηχανισμό εκτόξευσης και ελέγχου νοητών ακτίνων. Για να ενσωματώσουμε τον μηχανισμό αυτόν στο παιχνίδι μας δημιουργούμε ένα νέο σενάριο στον φάκελο **Scripts** στα **Στοιχεία (Assets)** με όνομα **GunDamage** (δεξί κλικ → **Create** → **C# Script**).

Η εντολή που θα χρησιμοποιήσουμε για να επιτύχουμε τα παραπάνω είναι η **Physics.Raycast** και

συντάσσεται ως εξής:

public static bool **Physics.Raycast**(Vector3 **origin**, Vector3 **direction**, out RaycastHit **hitInfo**, float **maxDistance**);

όπου στα παραπάνω ορίσματα έχουμε:

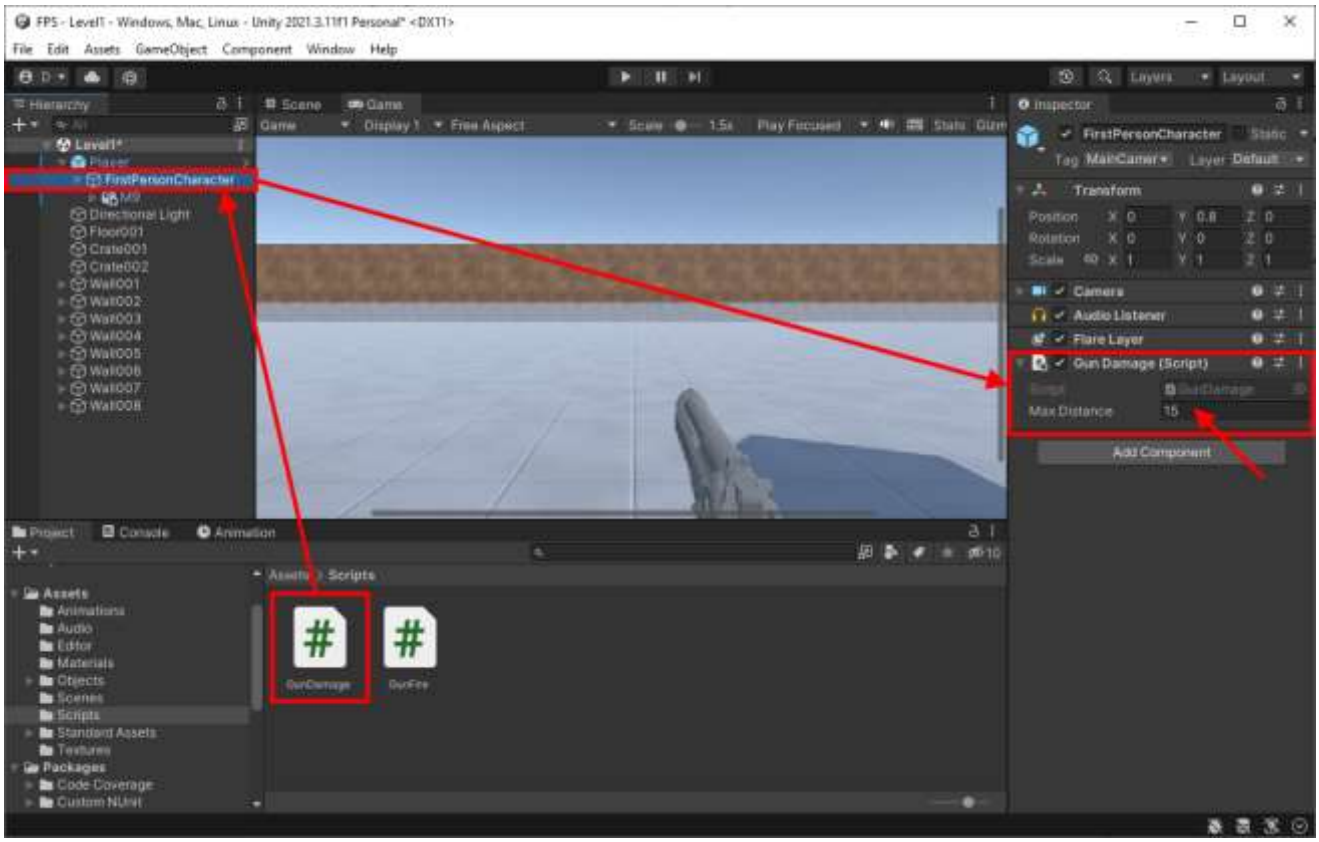
origin	Το σημείο έναρξης της ακτίνας
direction	Η κατεύθυνση της ακτίνας
hitinfo	Οι πληροφορίες που θα μας επιστραφούν εφόσον η ακτίνα συναντήσει εμπόδιο
maxDistance	Η μέγιστη απόσταση που θέλουμε να διανύσει η ακτίνα

Οπότε, τροποποιούμε τον κώδικα του **GunDamage** ως εξής:

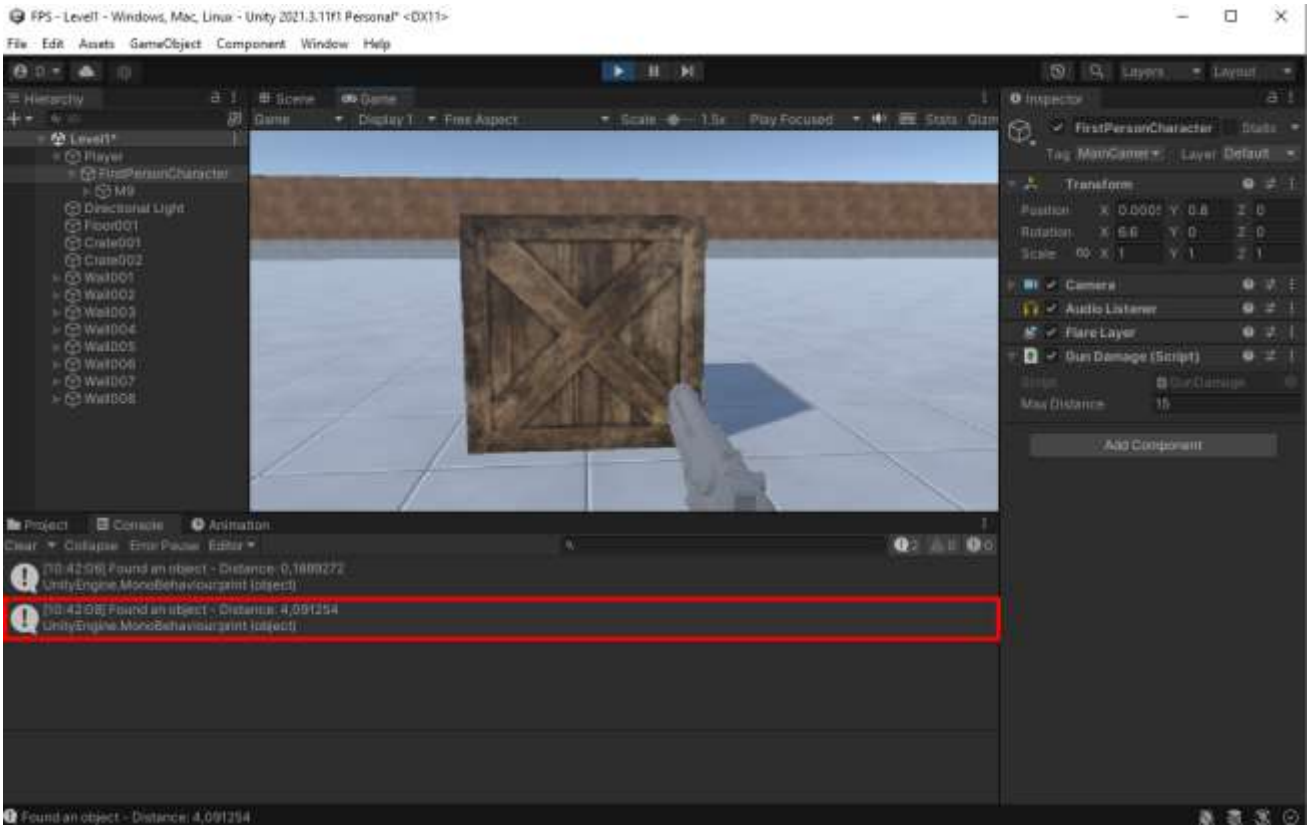
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GunDamage : MonoBehaviour
{
    public float MaxDistance = 15.0f;

    void Update()
    {
        if(Input.GetButtonDown("Fire1"))
        {
            RaycastHit hit;

            if (Physics.Raycast(transform.position, transform.forward, out hit, MaxDistance))
                print("Found an object - Distance: " + hit.distance);
        }

    }
}
```

Κατόπιν σύρουμε το σενάριο **GunDamage** και το αφήνουμε κάτω από το **FirstPersonCharacter** στο παράθυρο με την **Ιεραρχία** των αντικειμένων.

Στον κώδικα, η μεταβλητή **MaxDistance** δηλώθηκε ως πραγματική και χαρακτηρίστηκε ως **public**. Αυτό σημαίνει ότι μπορούμε να θέσουμε την τιμή της και μέσω του παραθύρου **Ιδιοτήτων** (**Inspector**) στο **FirstPersonCharacter** που εφαρμόζεται το σενάριο.



Αν πατήσουμε **Αναπαραγωγή** και εντός του παιχνιδιού πυροβολήσουμε προς μία κατεύθυνση, στην κονσόλα θα εμφανιστεί ένα μήνυμα εφόσον η ακτίνα μας συναντήσει κάποιο εμπόδιο το οποίο είναι σε απόσταση μικρότερη ή ίση της τιμής της μεταβλητής **MaxDistance**.



Πρόκληση ζημιάς

Στο παραπάνω σενάριο, θέλουμε να προσθέσουμε τη δυνατότητα η ακτίνα να δημιουργεί ζημιά στο σημείο που θα προσκρούσει (π.χ. σε έναν εχθρό) και να αφαιρεί ενέργεια από αυτό.

Έστω **DamageAmount** η μεταβλητή που περιγράφει το ποσό της ενέργειας που θέλουμε να αφαιρεθεί από το αντικείμενο. Για να αφαιρέσουμε το ποσό αυτό μπορούμε να καλέσουμε μία **μέθοδο** (**method**) που έχουμε κατασκευάσει εμείς για το αντικείμενο και η οποία θα ενεργοποιείται κάθε φορά που η ακτίνα προσπίπτει σε αυτό. Για να γίνει αυτό χρησιμοποιούμε την εντολή **SendMessage** η οποία στέλνει ένα μήνυμα προς το αντικείμενο προκειμένου αυτό να εκτελέσει μία μέθοδό του. Συντάσσεται ως εξής:

public void **SendMessage**(string **methodName**, object **value** = null, SendMessageOptions **options** = SendMessageOptions.RequireReceiver);

όπου στα παραπάνω ορίσματα έχουμε:

methodName	Το όνομα της μεθόδου που θα κληθεί.
value	Προαιρετική τιμή που θέλουμε να περάσει ως παράμετρος στην παραπάνω μέθοδο.
options	***SendMessageOptions.RequireReceiver****:* Θα επιστραφεί μήνυμα λάθους αν το μήνυμα δεν παραληφθεί από το αντικείμενο. ***SendMessageOptions.DontRequireReceiver****:* Δεν απαιτείται ενέργεια σε περίπτωση μη παραλαβής του μηνύματος.

Τροποιούμε ξανά τον κώδικα του **GunDamage** ως εξής:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class GunDamage : MonoBehaviour
{
    public float MaxDistance = 15.0f;
    public float DamageAmount = 5.0f;

    void Update()
    {
        if(Input.GetButtonDown("Fire1"))
        {
```

```
            RaycastHit hit;
            if (Physics.Raycast(transform.position, transform.forward, out hit, MaxDistance))
                if(hit.distance <= MaxDistance)
                {
                    hit.transform.SendMessage("DecreaseEnergy", DamageAmount,
                    SendMessageOptions.DontRequireReceiver);
                }
        }
    }
}
```

Στο παραπάνω σενάριο βλέπουμε ότι αποστέλλεται ένα μήνυμα προς το αντικείμενο στο οποίο η ακτίνα προσέκρουσε και ζητείται να εκτελεστεί σε αυτό η μέθοδος **DecreaseEnergy** και να περάσει σε αυτή η παράμετρος **DamageAmount**.

Το επόμενο βήμα μας είναι να δημιουργήσουμε ένα σενάριο το οποίο θα ονομάσουμε **EnemyScript** και θα περιλαμβάνει την μέθοδο αυτή (**DecreaseEnergy**). Όταν η μέθοδος **DecreaseEnergy** θα καλείται, η ενέργεια του αντικεμένου θα μειώνεται κατά **DamageAmount**. Αν η ενέργεια μηδενιστεί τότε το αντικείμενο θα «καταστρέφεται» και θα εξαφανίζεται από την σκηνή.

Δημιουργούμε στον φάκελο **Scripts** ένα νέο σενάριο με όνομα **EnemyScript** σε αυτό. Τροποποιούμε τον κώδικα του **EnemyScript** ως εξής:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnemyScript : MonoBehaviour
{
    public float EnemyHealth = 10.0f;

    void DecreaseEnergy(float DamageAmount)
    {
        EnemyHealth -= DamageAmount;
    }

    void Update()
    {
        if(EnemyHealth <= 0)
            Destroy(gameObject);
    }
}
```

Εφαρμόζουμε το σενάριο **EnemyScript** στα δύο ξύλινα κιβώτια (**Crate001** και **Crate002**). Πατώντας **Αναπαραγωγή**, αν πυροβολήσουμε 2 φορές στα αντικείμενα αυτά βλέπουμε ότι εξαφανίζονται.

Η σελίδα αυτή έμεινε σκόπιμα κενή