

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 2 – Unity 3D.

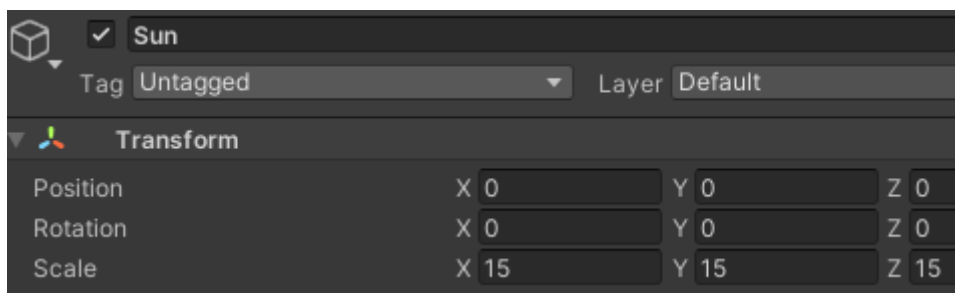
ΟΜΑΔΑ :

ΒΑΣΙΛΗΣ ΜΑΡΟΥΛΗΣ ΑΜ:4573

ΣΠΥΡΟΣ ΜΟΤΣΕΝΙΓΟΣ ΑΜ:4426

i) Η εφαρμογή μας έχει όνομα « **Ηλιακό Σύστημα** » και τρέχει με ανάλυση **1024x768**. Επίσης το *background* είναι μια σκηνή που αντιπροσωπεύει το διάστημα (*bonus ερώτημα*). Το κατασκευάσαμε φτιάχνοντας ένα skybox 6-sided με το texture 2k_stars_milky_way , υστέρτα το κάναμε equip στο περιβάλλον μας.

ii) Φτιάχνουμε μια σφαίρα ακτίνας 15 και την τοποθετούμε στο κέντρο. Έτσι το scale θα το κάνουμε 15 και στους 3 άξονες.



Κάνουμε Load το texture sun.jpg και το τοποθετούμε στον ήλιο.

Υστέρτα φτιάχνουμε ένα point light και το τοποθετούμε μαζί με τον ήλιο στο σημείο $x=0$ $y=0$ $z=0$ την εμβέλεια του την φτιάχνουμε έτσι ώστε να φωτίζει όλους τους πλανήτες που θα τοποθετήσουμε αργότερα. Στο χρώμα του φωτός επιλέγουμε ένα αχνό πορτοκαλί ώστε να αναπαραστήσουμε το φως του ηλίου όσο τον δυνατόν καλύτερα.

iii) Τοποθετούμε όλους τους πλανήτες όπως ζητείται από τον μικρότερο στον μεγαλύτερο. Εδώ σκεφτήκαμε να βάλουμε τις υφές των πλανητών όπως είναι στο ηλιακό μας σύστημα.

P1, ακτίνα =2, texture: mercury.jpg

P2, ακτίνα=4, texture: venus.jpg

P3, ακτίνα =6, texture: earth.jpg

P4, ακτίνα =7, texture: mars.jpg

P5, ακτίνα =8, texture: Jupiter.jpg

Η θέση των πλανητών έχει επιλεχθεί με τέτοιο τρόπο ώστε οι πλανήτες να μην ακουμπάνε μεταξύ τους.

Για την κυκλική τροχιά των πλανητών γύρω από τον ήλιο έχουμε φτιάξει το script **Rotation.cs**. Το εφαρμόζουμε σε όλους τους πλανήτες με target τον ήλιο, ώστε να γυρίζουν γύρω από αυτόν. Κάθε πλανήτης έχει ξεχωριστή ταχύτητα κυκλικής τροχιάς, μπορούμε να το αλλάξουμε ευκολά αφού έχουμε κάνει public την μεταβλητή rotation_speed και μπορούμε να αλλάξουμε την ταχύτητα από το inspector για κάθε πλανήτη. Οι ταχύτητες είναι τυχαίες.

Το ίδιο script εφαρμόζουμε και δεύτερη φορά πάνω σε όλα τα objects με target το ίδιο object κάθε φορά ώστε να πέτυχουμε την περιστροφή γύρω από τον εαυτό τους. Η ταχύτητα εδώ είναι ιδιό για όλους τους πλανήτες και ίση με 10, δίνεται η δυνατότητα να αλλάξουμε αυτή την ταχύτητα. (*bonus ερώτημα*)

iii)

Για την FirstPersonCamera αρχικά φτιάξαμε ένα κενό object που το ονομάσαμε FPS, μέσα στο κενό object φτιάξαμε ένα capsule object που είναι ο παίχτης μας, του έχουμε βάλει πράσινο χρώμα ώστε να ξεχωρίζουμε την θέση του. Επίσης βάζουμε κάτω από το κενό object την mainCamera μας και την τοποθετούμε στην κορυφή της κάψουλας μας με τέτοιο τρόπο ώστε να μην βλέπουμε το πράσινο αυτής. Κάνουμε reset τα transforms της κάμερας και της κάψουλας ώστε να είναι όλα στο 0 και πλέον μπορούμε και μετακινούμε το κενό μας object FPS, έχοντας κολλημένα πάνω του την κάψουλα και την κάμερα μας.

Για την λειτουργία της FirstPersonCamera έχουμε φτιάξει τα script **Mouse_Look.cs** και **PlayerMovement.cs**.

Το **Mouse_Look.cs** είναι υπεύθυνο για την χρήση του ποντικιού και την κατεύθυνση την οποία βλέπει ο παίχτης κάθε φορά που κουνά το ποντίκι. Το sensitivity του ποντικιού έχει οριστεί στο 100, όμως αυτό

μπορεί ευκολά να αλλάξει από το inspector αφού έχουμε ορίσει public μεταβλητή το sensitivity το ποντικιού στο script.

Το script **PlayerMovement.cs** είναι υπεύθυνο για την κίνηση του παίχτη στον τρισδιάστατο χώρο. Έχει προστεθεί και η λειτουργία Hover ώστε ο παίχτης να μπορεί να κινηθεί σε υψηλότερο ή χαμηλότερο επίπεδο από αυτόν τον πλανητών .

Έτσι μπορεί να βλέπει από οποία γωνιά , θέση , και ύψος το ηλιακό σύστημα μας.

Με τα πλήκτρο <w> ο παίχτης κινείται προς τα εμπρός.

Με τα πλήκτρο <s> ο παίχτης κινείται προς τα πίσω.

Με τα πλήκτρο <a> ο παίχτης κινείται προς τα αριστερά.

Με τα πλήκτρο <d> ο παίχτης κινείται προς τα δεξιά.

Με τα πλήκτρο <tab> ο παίχτης κινείται προς τα πάνω.

Με τα πλήκτρο <left ctrl> ο παίχτης κινείται προς τα κάτω.

Επίσης η ταχύτητα του παίχτη έχει οριστεί 20 αλλά μπορεί να αλλάξει από τον inspector.

Αν ο παίχτης κοιτά προς τα πάνω πατώντας το w ο παίχτης θα κινηθεί προς τα μπροστά και προς τα πάνω. Εμείς όμως φτιάξαμε την λειτουργία με το tab ώστε ο χρήστης να μπορεί να αλλάξει ύψος πιο άμεσα. Έτσι υπάρχει μια άμεση σχέση της θέσης του ποντικιού (της κατεύθυνση που κοιτά ο παίκτης) και της κίνησης του παίχτη όπως είναι δηλαδή μια first person camera.

v)

Για να εισάγουμε τον μετεωρίτη μας και να τον χρησιμοποιήσουμε στο script το οποίο χρειαζόμαστε τόσο για την δημιουργία του την εκτόξευση του αλλά και την σύγκρουση του με τους πλανήτες και τον ήλιο.

Φτιάχνουμε μια σφαίρα ακτίνας 1 της εφαρμόζουμε την υφή meteor.jpg . Στην συνέχεια επιλέγουμε τον μετεωρίτη μας από το

Hierarchy και το κάνουμε drop στο πεδίο assets ώστε να το έχουμε σαν αντικείμενο αλλά όχι στην σκηνή μας. Έτσι σβήνουμε τον μετεωρίτη από το Hierarchy.

Για την δημιουργία αλλά και την εκτόξευση του μετεωρίτη έχει φτιαχτεί το **spawn.cs** . Το ίδιο script κάνει scale τυχαία τον μετεωρίτη μας στο διάστημα [1.0,5.0].

Εκεί ορίζουμε την θέση που θέλουμε το αντικείμενο μας να δημιουργείται (μετεωρίτη) ως **SpawnPos** , το αντικείμενο που θέλουμε να δημιουργήσουμε ως **spawnee**, και πιάνου αντικειμένου οι συντεταγμένες θα γίνουν **scale** ως **scaling** .

Την δημιουργία του μετεωρίτη μας την κάνουμε με την εντολή Instantiate.

Το ίδιο script έχει αναλάβει να εκτοξεύει τον μετεωρίτη μας με την συνάρτηση **InstantiateProjectile** ορίζουμε για την λειτουργία αυτή την κάμερα (**cam**) από την οποία θα εκτοξεύονται οι μετεωρίτες μας και την ταχύτητα με την οποία εκτοξεύονται (**firing_speed**).

Για την σύγκρουση του μετεωρίτη με τους πλανήτες και τον ήλιο, έχουμε φτιάξει δυο script.

Το **Impact.cs** που το εφαρμόζουμε αποκλειστικά στον μετεωρίτη μας και μας δίνει την δυνατότητα όταν συγκρούετε με έναν πλανήτη να εξαφανίζεται ο ίδιος ο μετεωρίτης

Αν δεν συγκρουσθεί υστέρα από 7 δευτερόλεπτα καταστρέφεται. Την καταστροφή του μετεωρίτη μετρά από 7 δευτερόλεπτα την κατασκευάσαμε έτσι ώστε να μην υπάρχουν πολλοί πλανήτες ταυτόχρονα στο παιχνίδι μας και έχουμε μείωση της απόδοσης.

Το **Impact_planets.cs** το δημιουργήσαμε έτσι ώστε όταν ο πλανήτης συγκρουσθεί με τον μετεωρίτη να εξαφανίζεται και ο ίδιος. Αυτό το script το έχουμε εφαρμόσει σε όλους τους πλανήτες **εκτός** από τον ήλιο που δεν θέλουμε να καταστρέφεται από την σύγκρουση του μετεωρίτη.

Για την υλοποίηση και την σωστή λειτουργία των παραπάνω scripts έχουμε βάλει τα tag “meteor” στον μετεωρίτη μας και το tag “planet” στον ήλιο και τους υπολοίπους πλανήτες μας.