

2Η ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ UNITY ΓΡΑΦΙΚΑ

ΑΛΕΞΑΝΔΡΟΣ-ΔΗΜΗΤΡΙΟΣ ΤΖΙΜΟΓΙΑΝΝΗΣ, Α.Μ. 4179
ΓΕΩΡΓΙΟΣ ΤΡΙΑΝΤΟΣ, Α.Μ. 4184

Το project υλοποιήθηκε σε Windows 10 Version 1909 x64 σε Unity 2020.1.17f1.

Γενικά, τα scripts μας βρίσκονται στο directory Assets/Scripts και τα Materials στο Assets/Materials.

Για το ερώτημα (i):

Φτιάξαμε την εφαρμογή με όνομα Συγκρουόμενα, τρέχει με ανάλυση 1024x768 και δείχνει τον αρχικό κύβο. Το χρώμα του κύβου δημιουργείται τυχαία απ' το script RandomColor.cs που βρίσκεται στο Assets/Scripts και μέσω του Cube Material, που βρίσκεται στο Assets/Materials, κάναμε τον κύβο διαφανή βάζοντας στο rendering mode την επιλογή transparent. Το background color είναι λευκό, το οποίο το βάλαμε αλλάζοντας στη Main Camera (που είναι "παιδί" του Player) την επιλογή του Clear Flags (βάλαμε Solid Color) και στην επιλογή Background βάλαμε το λευκό χρώμα. Ο κύβος εκτείνεται απ' το (0, 0, 0) έως το (100, 100, 100) βάζοντας στον Cube position (50, 50, 50) και scale (100, 100, 100).

Για το ερώτημα (ii) :

Κάθε φορά που ο χρήστης πατάει το space εμφανίζεται ένα μικρό αντικείμενο (κύβος, κύλινδρος ή σφαίρα τυχαία κάθε φορά) που κινείται προς τυχαία κατεύθυνση με τυχαίο χρώμα και ενεργοποιημένο το collider.

Συγκεκριμένα, φτιάξαμε ένα GameObject στο οποίο προσθέσαμε στα Components το script μας LoadObj και μέσω αυτού, κάθε φορά που πατάμε space κάνουμε spawn ένα μικρό στοιχειώδες αντικείμενο. Με την Random.Range υλοποιήσαμε την τυχαία επιλογή αντικειμένου κάθε φορά, το μέγεθος των μικρών αντικειμένων και την αρχική τους κατεύθυνση. Για το μέγεθος του cylinder, βάλαμε το y του διανύσματος για το transform να γίνει το μισό του τυχαίου d ώστε οι κύλινδροι να έχουν το ίδιο ύψος και διάμετρο βάσης. Ενεργοποιούμε το Rigidbody κάθε μικρού αντικειμένου και για την κατεύθυνση χρησιμοποιούμε την εντολή velocity και πολλαπλασιάζουμε με το speed για να αυξηθεί η ταχύτητα (χωρίς τον πολλαπλασιασμό η ταχύτητα είναι πολύ μικρή). Τέλος, βάλαμε 6 πηγές φωτισμού, κάθε μία να χτυπάει κάθετα σε κάθε πλευρά του κύβου με intensity 0.6.

Για το ερώτημα (iii):

Μέσα στον κύβο υπάρχει η σφαίρα SPH με κόκκινο χρώμα και κέντρο το σημείο (50, 50, 50). Η σφαίρα κινείται στους άξονες x και y με τα βέλη (x+ με right arrow, x- με left arrow, y+ με up arrow, y- με down arrow) και στον άξονα z (z+ με NumpadPlus, z- με NumpadMinus ή και ταυτόχρονα z+ με Equals, που είναι και συν, αν πατηθεί με Shift (δίπλα από Backspace), και z- με Minus (δίπλα απ' το Equals)) και συγκρούεται με τα μικρά στοιχειώδη αντικείμενα, αφού ενεργοποιήσαμε το Sphere Collider απ' το Unity.

Για το ερώτημα (iv):

Υλοποιήσαμε τη first person camera στο script FPS.cs (το οποίο το κάναμε add component στον Player), με την οποία ο χρήστης βλέπει τη σκηνή από οποιαδήποτε γωνία και θέση με την κίνηση του mouse και τα πλήκτρα WASDQE (κίνηση στον z με WS, στον x με AD και στον y με QE).

Για την υλοποίηση της κίνησης του mouse χρησιμοποιήσαμε κώδικα που βρήκαμε στο παρακάτω link: <https://www.youtube.com/watch?v=DxrncPIQ0mM&feature=youtu.be>

Για το ερώτημα (v):

Όταν ο χρήστης πατάει το t, ενεργοποιείται και όταν το ξαναπατάει απενεργοποιείται το texture της σφαίρας. Η υλοποίηση έγινε στο Script LoadTexture.cs. Συγκεκριμένα, έχουμε δημιουργήσει 2 διαφορετικά materials για τη σφαίρα στο Unity, το SphereMaterial που περιέχει το αρχικό κόκκινο χρώμα που πρέπει να έχει η σφαίρα και το TexturedMaterial, στο οποίο βάλαμε στο Shader: Legacy Shaders/Diffuse και επιλέξαμε για texture το texture-sphere.jpg, το οποίο βρίσκεται στο Assets/Textures. Προσπαθήσαμε να το κάνουμε με τον τρόπο που αναφέρατε ώστε το script να βρίσκει το path που υπάρχει το εκτελέσιμο και να φορτώνει από εκεί το texture (ο κώδικας που γράψαμε έβρισκε το path του εκτελέσιμου σε οποιοδήποτε pc), αλλά μας χτυπούσε το συγκεκριμένο error:

UnauthorizedAccessException: Access to the path 'C:\Users\...\Unity\Συγκρούμενα' is denied.

Ο κώδικας αυτός υπάρχει σαν σχόλια στο Script LoadTexture.cs.

Για το μπόνους (iii):

Απ' την υλοποίηση μας στο ερώτημα (ii) τα αντικείμενά μας έχουν ήδη Rigidbody και Collider, επομένως γίνεται σύγκρουση και αναπήδηση μεταξύ τους.

Για το μπόνους (v):

Δημιουργήσαμε ένα empty GameObject που το ονομάσαμε Player και βάλαμε 2 "παιδιά" στον Player, τη MainCamera και ένα GameObject Capsule. Ο παίχτης "κουβαλάει" την κάμερα και συγκρούεται με τα μικρά αντικείμενα και τη σφαίρα SPH.

ΠΑΡΑΤΗΡΗΣΕΙΣ:

- Εκτός απ' τον κύβο, δημιουργήσαμε και 6 πλευρές ακριβώς στη θέση που βρίσκεται ο αρχικός κύβος (δηλαδή 6 κύβους με scale z=0) και απενεργοποιήσαμε το meshRenderer, ώστε και οι 6 πλευρές να είναι αόρατες. Οι 3 πλευρές που έχουν κοινό σημείο το (0, 0, 0), δηλαδή το σημείο που εμφανίζονται αρχικά τα μικρά στοιχειώδη αντικείμενα, που είναι στη δικιά μας περίπτωση οι Cube (1), Cube (4), Cube (6), εμποδίζουν την είσοδο των μικρών αντικειμένων, αφού το Box Collider της κάθε πλευράς είναι ενεργοποιημένο. Υλοποιήσαμε από ένα Script για κάθε μία από αυτές τις

3 πλευρές (MoveCube1.cs, MoveCube4.cs, MoveCube6.cs), όπου με το κάθε Script, κάθε φορά που ο χρήστης πατάει το space για να εμφανιστεί ένα νέο μικρό αντικείμενο, η κάθε μία από τις 3 πλευρές μετακινείται για 5 εκατοστά του δευτερολέπτου (0.05 sec) 10 μονάδες προς την κατάλληλη κατεύθυνση στο σύστημα και επιστρέφει στην αρχική της θέση. Αυτό έχει ως αποτέλεσμα να επιτρέπει στα αντικείμενα να μπαίνουν στον κύβο και κατευθείαν επανέρχεται η κάθε μία απ' τις 3 αυτές πλευρές στη θέση της και το πρόγραμμα λειτουργεί κανονικά.

- Έχουμε δημιουργήσει στο Assets/Materials ένα New Physic Material, το οποίο το έχουμε τροποποιήσει ώστε τα αντικείμενα που συγκρούονται με τις αόρατες πλευρές του κύβου, τη σφαίρα SPH και την κάψουλα-κάμερα, να αναπηδάνε με ίδια ταχύτητα με αυτή που συγκρούονται.
- Βάλαμε στη σφαίρα SPH, στο component Rigidbody μεγάλο mass (10000 για την ακρίβεια), έτσι ώστε όταν τα μικρά στοιχειώδη αντικείμενα συγκρούονται με τη σφαίρα, η τελευταία να μένει σταθερή, καθώς πριν αυξήσουμε το mass, τα αντικείμενα συγκρούονταν με τη σφαίρα και η σφαίρα αναπηδούσε και αυτή και, ως αποτέλεσμα, δε μπορούσε ο χρήστης να ελέγξει την κίνησή της από το πληκτρολόγιο.
- Όσον αφορά την κάμερα, κάθε φορά που τρέχουμε την εφαρμογή, το ποντίκι εξαφανίζεται και κεντράρει στο κέντρο της οθόνης.
- Η εφαρμογή τερματίζει, αν ο χρήστης πατήσει το πλήκτρο Esc.