Εργασία 1

Παιχνίδια και Τεχνητή Νοημοσύνη

Αλέξανδρος Γαϊτάνης, ΑΜ: 63

ΠΜΣ στην Τεχνητή Νοημοσύνη

Χειμερινό εξάμηνο 2021 - 2022

Τμήμα Πληροφορικής

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Όλα τα projects βρίσκονται μέσα στον εξής φάκελο:

https://drive.google.com/drive/folders/1LNLOw1yYsBhpRlCmyCRqeU8iT1fnzCaQ?usp=sharing

1. Σωματίδια μέσα σε κύβο

Το project βρίσκεται στον εξής φάκελο:

https://drive.google.com/drive/folders/1ELaGYOoN2LA9hRt7qDboxwdOeAF8Rpwx?usp=sharing

Για να δουλέψει σωστά το 1ο ερώτημα πρέπει τα objects *Attractor1* και *Attractor2* να είναι disabled.

Ο κώδικας βρίσκεται μέσα στον φάκελο *Assets* > *Scripts*. Το script *ParticleGenerator* παράγει τα σωματίδια σε ένα σημείο με τυχαίες ταχύτητες. Οι ταχύτητες παίρνουν τιμές από ένα range το οποίο είναι ορισμένο στο object *ParticleGenerator*. Το script *ParticleDestroyer* είναι υπεύθυνο για την καταστροφή τους μετά από ένα χρονικό διάστημα. Αυτό είναι συνδεδεμένο με το prefab *Particle*. Με αυτό το prefab είναι συνδεμένο και το script *ParticleController* το οποίο είναι το script που ελέγχει την κίνηση του κάθε σωματιδίου.

Σε κάθε frame καλείται η *Update()* του *ParticleController* και υπολογίζεται η καινούρια θέση και η ταχύτητα του σωματιδίου. Ουσιαστικά ο υπολογισμός αυτός γίνεται μέσα στην *CalcNewState()* χρησιμοποιώντας Euler και εκεί εφαρμόζονται όλες οι δυνάμεις που περιγράφονται στην εκφώνηση.

Έπειτα με την *CheckForCollision()* γίνεται ο έλεγχος για σύγκρουση. Αν το σωματίδιο έχει βγει εκτός του κύβου τότε διορθώνονται η θέση και η ταχύτητα του. Η διόρθωση γίνεται μέσα σε ένα while-loop γιατί στην περίπτωση που έχει μεγάλη ταχύτητα μπορεί να ξαναβρεθεί εκτός κύβου. Η *CheckForCollisionCore()* βρίσκει ποιο είναι το πιο κοντινό επίπεδο στο οποίο ανιχνεύθηκε collision γιατί το σωματίδιο μπορεί να βρεθεί από την άλλη πλευρά του επιπέδου για παραπάνω από ένα επίπεδα του κύβου. Τέλος η διόρθωση γίνεται με την *CheckForCollisionWithPlane()* θεωρώντας ευθύγραμμη κίνηση με σταθερή ταχύτητα από το παλιό σημείο στο νέο. Το σημείο τομής της ευθείας αυτής με το επίπεδο βρίσκεται με την *FindIntersectionLineWithPlane()*. Η κάθετη συνιστώσα της ταχύτητας αντιστρέφεται και μειώνεται σε μέτρο και μετά υπολογίζεται το νέο σημείο ξανά με την *CalcNewState()*.

2. Σωματίδια μέσα σε κύβο με ελκυστές

Το project είναι το ίδιο με αυτό που χρησιμοποιήθηκε στο 1ο ερώτημα. Για να δουλέψει σωστά το 2ο ερώτημα πρέπει τα objects *Attractor1* και *Attractor2* να είναι enabled.

Η μόνη διαφορά με το προηγούμενο ερώτημα είναι ότι στην *CalcNewState()* προστίθενται και οι δυνάμεις από κάθε σωματίδιο προς τους ελκυστές.

3. Σωματίδια μέσα σε σφαίρα

Το project βρίσκεται στον εξής φάκελο:

https://drive.google.com/drive/folders/170AGNRBvYALfrXyaqTAfi91LRnHp3yTn?usp=sharing

Ο κώδικας βρίσκεται μέσα στον φάκελο *Assets* > *Scripts*. Το script *ParticleGenerator* παράγει έναν συγκεκριμένο αριθμό σωματιδίων σε ένα σημείο. Το script *ParticleController* ελέγχει την κίνηση του κάθε σωματιδίου.

Ο υπολογισμός της νέας θέσης και της ταχύτητας του σωματιδίου γίνεται πάλι με Euler, στην *CalcNewState()*, όπου εφαρμόζονται οι δυνάμεις από κάθε σωματίδιο προς όλα τα άλλα. Η διαδικάσια διόρθωσης της θέσης και της ταχύτητας για το collision με τη σφαίρα είναι ίδια με το 1ο και το 2ο ερώτημα. Η διαφορά είναι ότι η εύρεση του σημείου τομής της ευθείας με τη σφαίρα γίνεται με την *FindIntersectionLineWithSphere()* και ότι η κάθετη συνιστώσα της ταχύτητας έχει κατεύθυνση από το σημείο τομής προς το κέντρο της σφαίρας.

4. Bird flocking

Το project βρίσκεται στον εξής φάκελο:

https://drive.google.com/drive/folders/1ycmfrV-OX-xIZ_gM1n_Zn_2hvUZTdPvb?usp=sharing

Στην σκηνή χρησιμοποιήθηκαν τα εξής:

- Νησί: https://123free3dmodels.com/small-tropical-island-9619
- Ουρανός: https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353
- Θάλασσα (συγκεκριμένα το prefab *WaterProNighttime*): https://assetstore.unity.com/packages/essentials/asset-packs/standard-assets-for-unity-2018-4-32351.
- Πουλιά: https://assetstore.unity.com/packages/3d/characters/animals/birds/seagull-larus-canus-109558
- Ήχος του πουλιού: https://quicksounds.com/uploads/tracks/1236253129_610739684_1548623274.mp3
- Ήχος της θάλασσας: https://bigsoundbank.com/detail-1046-small-waves-facing-the-ocean.html
- Ένα directional light για το φεγγάρι
- Ένα point light μέσα στο σπίτι
- Πάτωμα για το σπίτι: 3D Object του Unity στο οποίο έβαλα ίδιο texture με το σπίτι

Ο κώδικας βρίσκεται μέσα στον φάκελο *Assets > Scripts*. Το script *ParticleGenerator* παράγει έναν συγκεκριμένο αριθμό πουλιών σε τυχαία σημεία και με τυχαίες ταχύτητες σε μια αρχική περιοχή στην σκηνή. Το script *ParticleController* ελέγχει την κίνηση του κάθε πουλιού.

Στην *Update()* του *ParticleController()* υπολογίζεται η θέση, η ταχύτητα, η επιτάχυνση και ο προσανατολισμός του πουλιού. Η *Update()* ξεκινάει επίσης και την αναπαραγωγή του ήχου του. Ανάλογα με την τιμή της επιτάχυνσης του πουλιού ενεργοποιείται και το κατάλληλο animation για το χτύπημα των φτερών για επιβράδυνση, επιτάχυνση ή απλώς άνοιγμα των φτερών αν η επιτάχυνση είναι πολύ μικρή.

Η κίνηση σε σμήνος προκύπτει από το άθροισμα 3 δυνάμεων για separation, cohesion και alignment πάνω στο πουλί. Το separation επιτυγχάνεται εφαρμόζοντας δυνάμεις από τα άλλα πουλιά που είναι κοντά του προς αυτό, ζυγισμένες ως προς την απόσταση. Το cohesion επιτυγχάνεται εφαρμόζοντας μια δύναμη από το πουλί προς το κέντρο των πουλιών της γειτονίας του. Το alignment επιτυγχάνεται εφαρμόζοντας μια δύναμη με κατεύθυνση την μέση ταχύτητα των πουλιών της γειτονίας του. Τέλος, εξασφαλίζεται ότι τα πουλιά κινούνται μέσα σε ένα ορθογώνιο παραλληλεπίπεδο πάνω από το νησί εφαρμόζοντας δυνάμεις αντίστροφου τετραγώνου από τις πλευρές του προς τα πουλιά.