

Πανεπιστήμιο Πατρών
Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Μιχαήλ Μαναγούδης 1059398

Γραφικά και Εικονική Πραγματικότητα

Απαλλακτική Εργασία : **Special Effects**

Ημερομηνία Υποβολής : 21/02/2021

Περιγραφή:

Το ζητούμενο της εργασίας αυτής είναι η απόδοση μίας σκηνής (με εφέ) μίας πολύ γνωστής ταινίας δράσης με βάση τις μεθόδους που διδάχθηκαν στο εργαστήριο του μαθήματος (φυσικά έχοντας κατανοήσει την βασική θεωρία από τις διαλέξεις).

Παρακάτω απαντώνται τα επιμέρους ερωτήματα της εργασίας με επεξηγήσεις και παραδείγματα εκτέλεσης της.

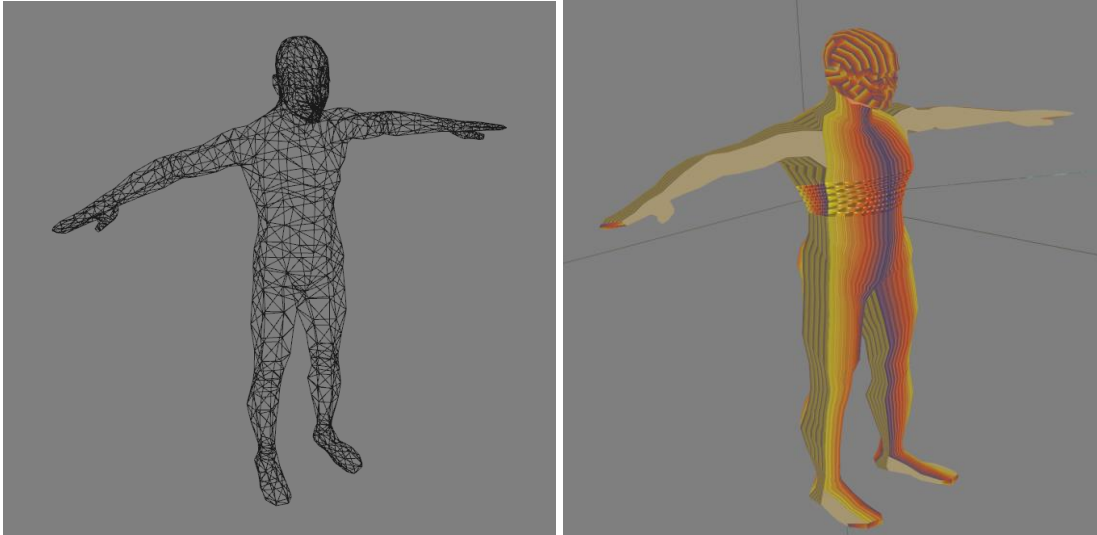
Διευκρινήσεις :

Αρκετό μέρος του κώδικα είναι αποτέλεσμα συγχώνευσης των αρχείων που χρησιμοποιήθηκαν στα εργαστήρια.

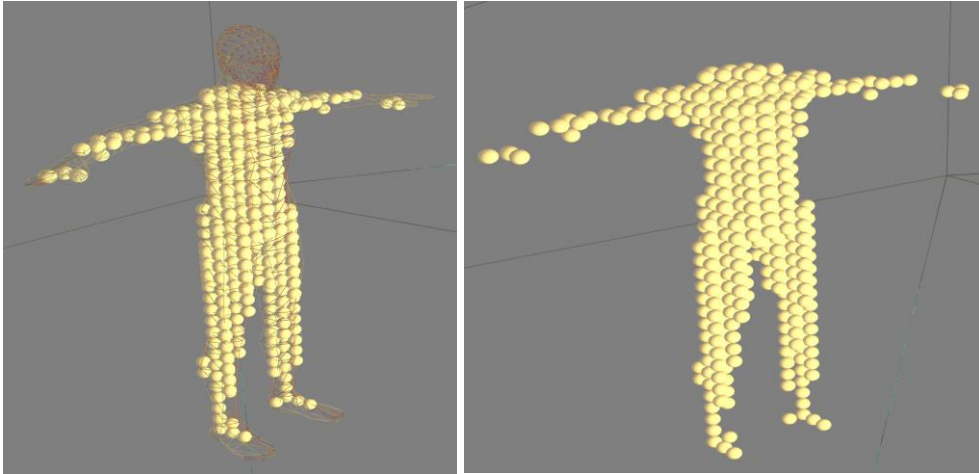
Επιπλέον ο κώδικας της εργασίας είναι στον μεγαλύτερό του βαθμό παραμετροποιήσιμος ώστε να λειτουργεί σε διάφορες περιπτώσεις μοντέλων και των υπόλοιπων παραμέτρων που ορίζονται μέσα του.

- **Μέρος Α :**

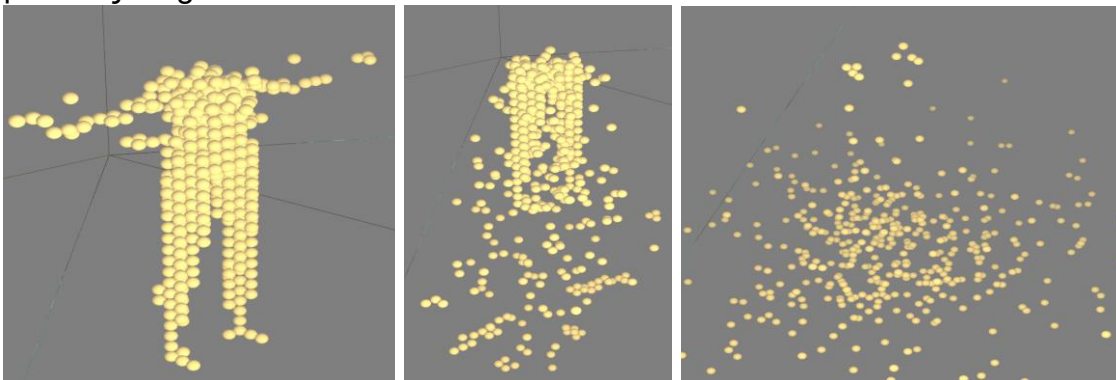
- Αρχικά φορτώνεται ένα μοντέλο ανθρώπου στην σκηνή με σχετικά μικρό αριθμό τριγώνων (λιγότερα από 5.000). Αυτό επιτυγχάνεται εύκολα δημιουργώντας ένα αντικείμενο της κλάσης Drawable δίνοντάς του το μονοπάτι του αρχείου obj που περιέχει το μοντέλο. Στην συνέχεια προστίθεται ένα texture στο μοντέλο για να γίνει πιο αληθοφανές.



- Για να γεμίσει το κούφιο μοντέλο με μπάλες ορίζονται η επιθυμητή μέγιστη ποσότητα, η ακτίνα και ένας πίνακας που θα περιέχει τις τελικές μπάλες. Έπειτα κατασκευάζεται ένα BBox που θα περικλείει το μοντέλο ώστε μέσα σε αυτόν τον χώρο να τοποθετηθούν αρχικά οι μπάλες και να διατηρηθούν μόνο όσες βρίσκονται μέσα στο ίδιο το μοντέλο. Για τον έλεγχο αν οι μπάλες βρίσκονται μέσα στο μοντέλο χρησιμοποιείται ο αλγόριθμος Möller–Trumbore, ένας αλγόριθμος ray tracing, που εξετάζει αν μία ακτίνα από το κέντρο της σφαίρας προς το άπειρο τέμνει σε μονό αριθμό το μοντέλο (οπότε η μπάλα βρίσκεται μέσα). Τέλος η κύρια συνάρτηση (fill_human) ελέγχει σε κάθε θέση μέσα στο BBox αν αυτή είναι εσωτερική του μοντέλου και σε αυτήν την περίπτωση δημιουργεί ένα στιγμιότυπο της κλάσης Sphere και το τοποθετεί στον αντίστοιχο πίνακα. Σε περίπτωση που οι αρχική ποσότητα για τις μπάλες είναι μεγαλύτερη από όσες μπορούν να χωρέσουν στο μοντέλο (με βάση τα χαρακτηριστικά της κάθε μίας) τοποθετούνται μόνο όσες χωράνε και ενημερώνεται η μεταβλητή που μετρά το πλήθος τους.

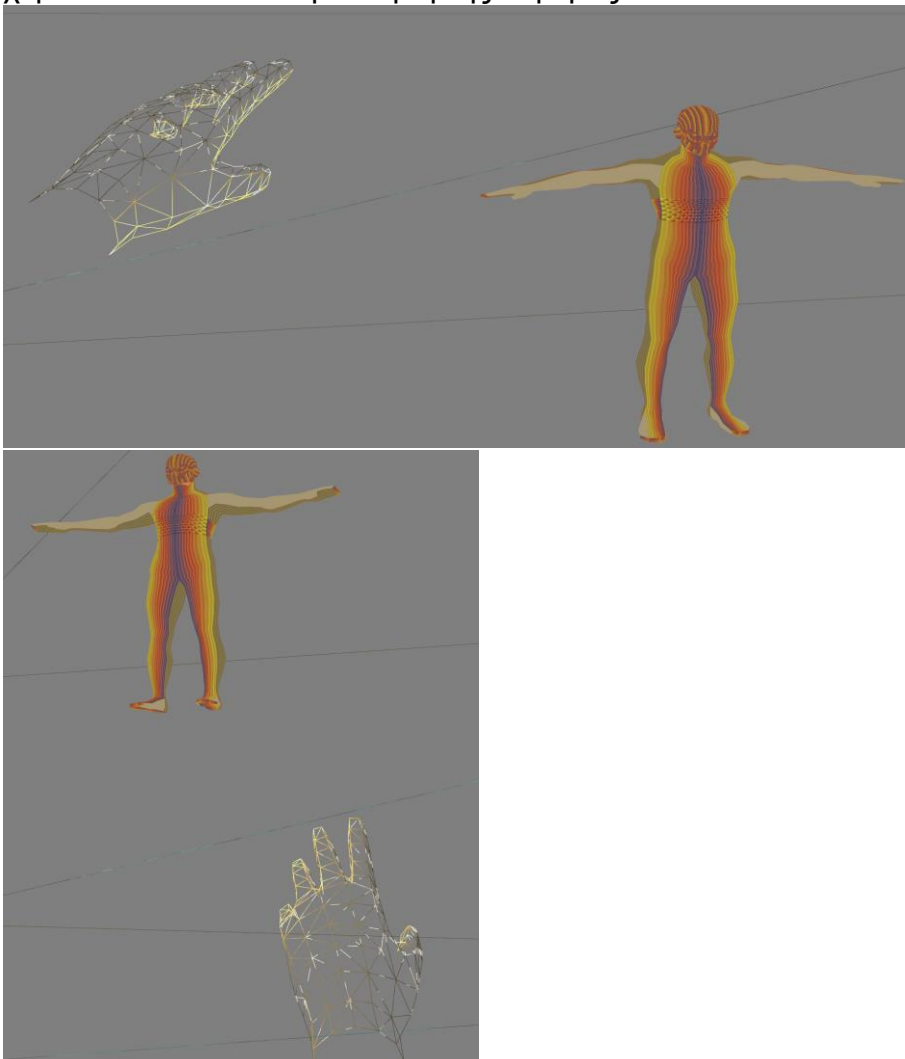


- iii. Με κάποιο πλήκτρο (V key) από το πληκτρολόγιο ή το ποντίκι δίνεται η επιλογή στο χρήστη να εξαφανίσει το μοντέλο (δηλαδή απλά να σταματήσει να αποτυπώνεται στην οθόνη) και από εκείνη την χρονική στιγμή και μετά οι μπάλες να είναι ελεύθερες να πέσουν στο πάτωμα με ρεαλιστικό τρόπο. Μέσα στην main loop ελέγχεται αν ο χρήστης εξαφάνισε το μοντέλο και τότε προστίθεται στα χαρακτηριστικά των σφαιρών η δύναμη της βαρύτητας. Για να επιτευχθούν ρεαλιστικές συνθήκες στην πτώση των σφαιρών υπάρχουν 2 συναρτήσεις που ελέγχουν τις συγκρούσεις μεταξύ μπάλας και δαπέδου (το οποίο δημιουργείται σαν έννοια του χώρου-δωματίου) και μεταξύ μπάλας με άλλη μπάλα. Σε αυτές τις συναρτήσεις σύμφωνα με την αρχή διατήρησης της ορμής, της ενέργειας και των εξισώσεων ελαστικής κρούσης υπολογίζονται κατάλληλα οι νέες θέσεις και ταχύτητες των σωμάτων (ή του σώματος) μετά την σύγκρουση. Για ακόμα ρεαλιστικότερες συνθήκες βάζουμε μία απόσβεση στις συγκρούσεις για να αποφευχθεί η τελείως ελαστική κρούση (καλύτερή αίσθηση του δαπέδου σαν χώμα) και ελέγχεται αν η ταχύτητα κάθε σφαίρας είναι μικρότερη από μία τιμή 'σφάλματος', όπου και θεωρείται ότι είναι κινητοποιημένη μηδενίζοντας την ταχύτητά της από κει και πέρα. Επιπλέον αμέσως μετά την εξαφάνιση του μοντέλου, επειδή ο αλγόριθμος που τοποθετεί τις μπάλες μέσα στο μοντέλο είναι έτσι ορισμένος που οι μπάλες είναι ακριβώς η μία πάνω στην άλλη, προσδίδεται στιγμιαία σε κάθε μπάλα μία ελάχιστη και τυχαία κατά μέτρο ταχύτητα. Να σημειωθεί ότι για τον υπολογισμό της επόμενης θέσης χρησιμοποιείται η μέθοδος `rungeKuta4th`.



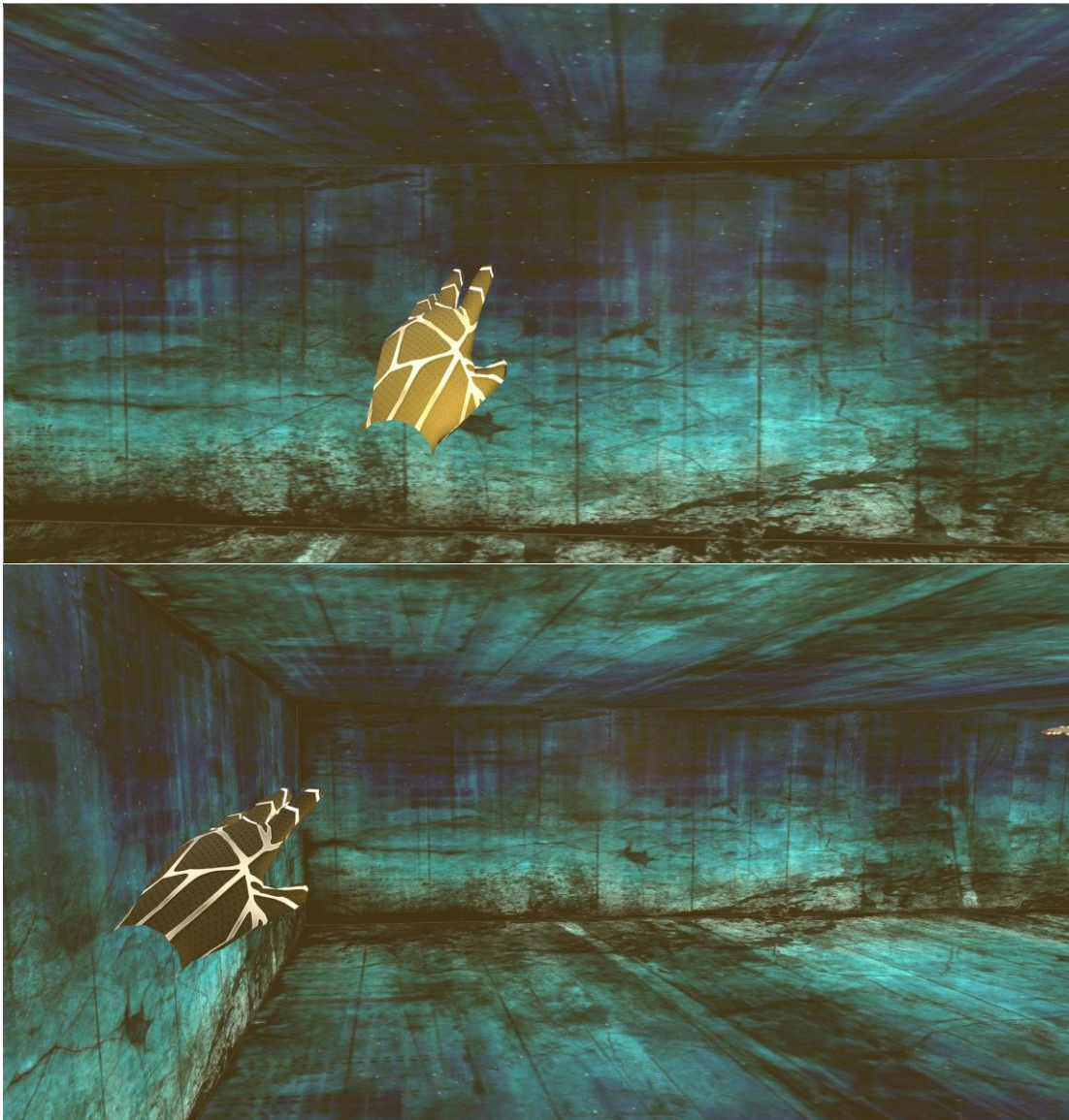
- **Μέρος Β :**

- i. Όμοια με το ερώτημα Αί φορτώνεται ένα μοντέλο χεριού. Για να δείχνει στο ίδιο σημείο που δείχνει και το ποντίκι και ταυτόχρονα να ακολουθεί την κάμερα θα χρησιμοποιηθούν οι πίνακες μετασχηματισμού (Μεταφορά, Περιστροφή, Κλιμάκωση). Συγκεκριμένα αρχικά κλιμακώνεται το μοντέλο για να έχει το κατάλληλο μέγεθος μπροστά στην κάμερα, στην συνέχεια περιστρέφεται αρχικά για να αποκτήσει την σωστή ‘πόζα’ και μετά ανάλογα με τις γωνίες περιστροφής της κάμερας (horizontal, vertical angle) Τέλος μεταφέρεται σε τέτοιο σημείο ώστε να φαίνεται μπροστά και αριστερά στην κάμερα (αριστερό χέρι). Επειδή θεωρείται ότι το περιεχόμενο της εικόνας θα βρίσκεται κυρίως μπροστά στην κάμερα εισάγεται στον πίνακα περιστροφής του χεριού και η horizontal γωνία της κάμερας και να αποδοθεί η αίσθηση ότι όταν ο χρήστης κοιτάει δεξιά ή αριστερά στρέφει στην πραγματικότητα μόνο το κεφάλι του και δεν γυρνάει ολόκληρο τον κορμό του. Εναλλακτικά με μία απλή αλλαγή στον πίνακα περιστροφής (μέσω του διανίσματος up) θα μπορούσε να υλοποιηθεί και το άλλο σενάριο. Παρακάτω φαίνονται οι διαφορετικές οπτικές του χεριού όταν αλλάζει η κατακόρυφη ή οριζόντια γωνία της κάμερας αλλά συγχρόνως και το χέρι που ακολουθεί την κίνηση της κάμερας.

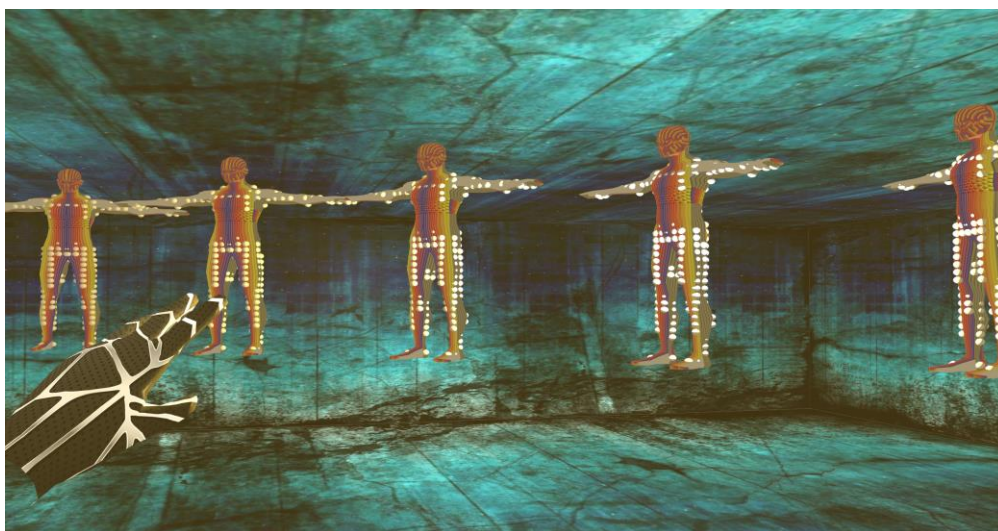


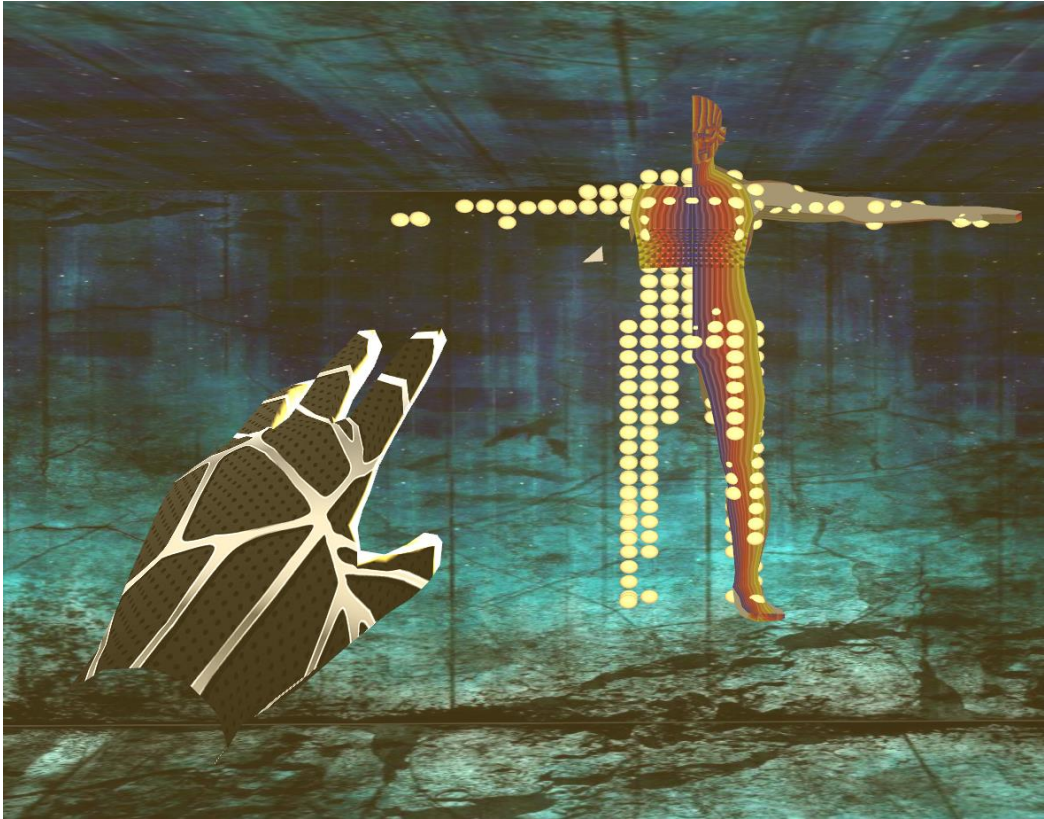
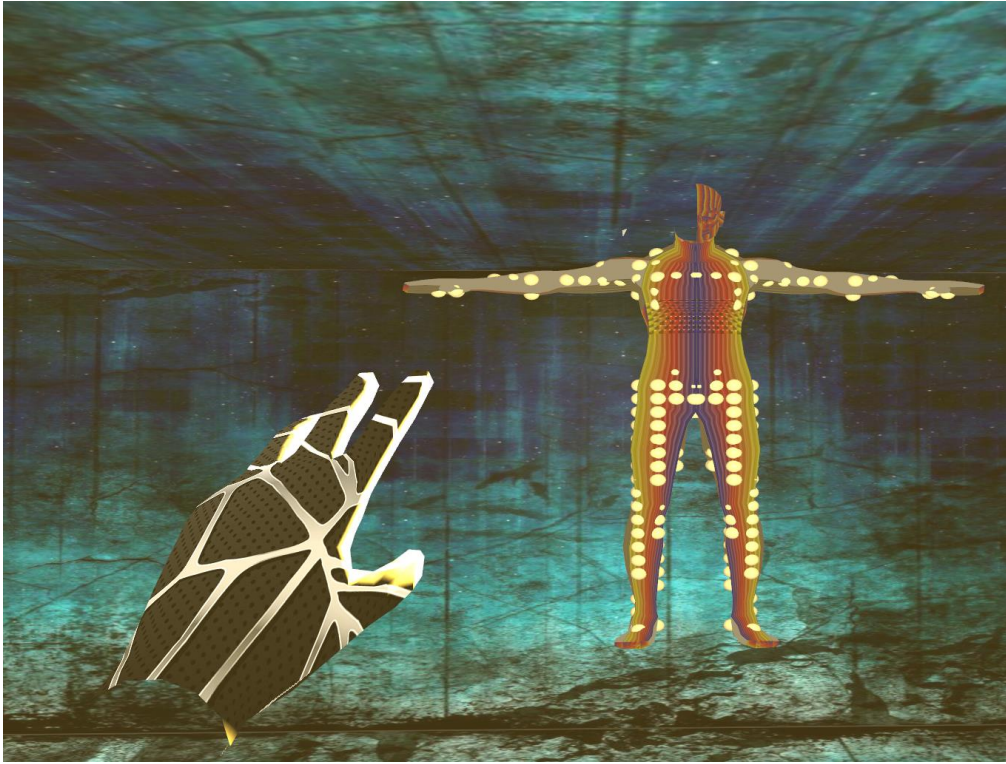
- ii. -Το κομμάτι του σπασίματος του χεριού και της κίνησης του μοντέλου στο τελευταίο ερώτημα (που και τα 2 αφορούν σκελετική δομή) εξηγούνται στο τέλος της αναφοράς.-
Για να φαίνεται ότι το χέρι φοράει κάποιο γάντι αρκεί να προσθεθεί ένα κατάλληλο texture που θα έχει την 'υφή' ενός σιδερένιου γαντιού (αντίστοιχα όπως στην σκηνή της ταινίας)
Επιπλέον σε αυτό το σημείο προστείνεται και ένα αντίστοιχο texture στο χώρο-δωμάτιο για να κάνει καλύτερη την αίσθηση του χώρου.

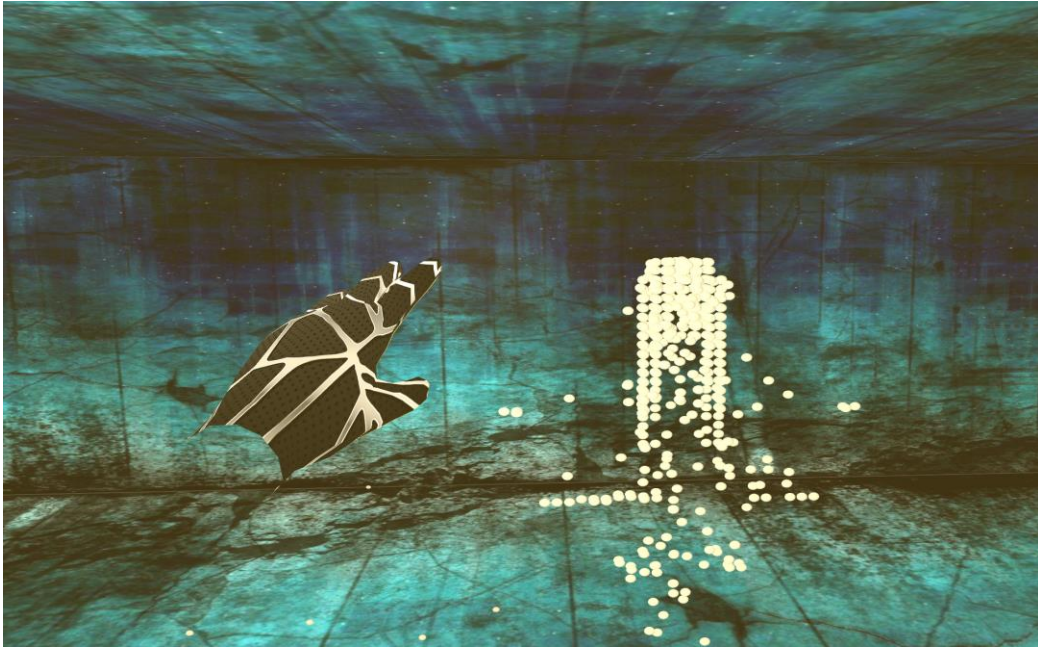
Από αυτό το σημείο και μετά αλλάζει αρκετά η δομή του αντικειμένου του μοντέλου. Πλέον κατασκευάζεται μία ενιαία δομή (struct) με βασικά χαρακτηριστικά το κούφιο μοντέλο, τον πίνακα των σφαιρών, τον πίνακα μετασχηματισμού μοντέλου και κάποιες άλλες χρήσιμες παραμέτρους που θα κάνουν ευκολότερη και ταχύτερη την οργάνωση των παρακάτω ερωτημάτων.



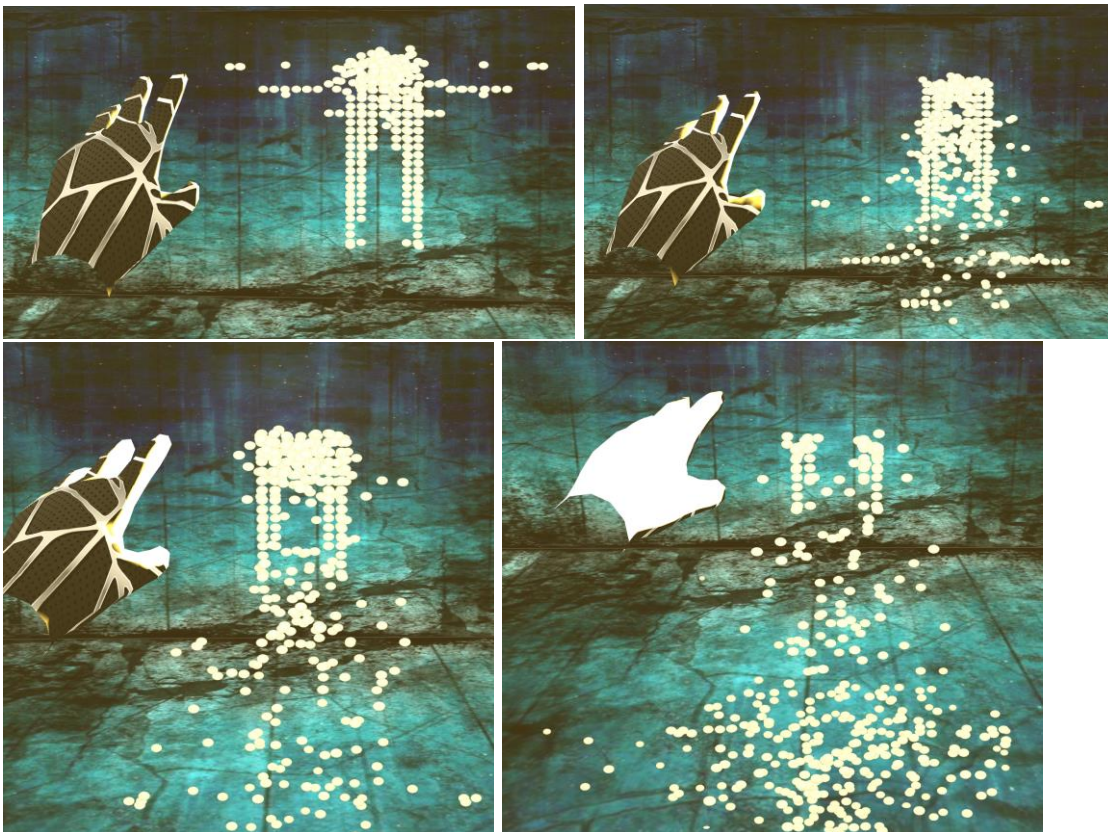
- iii. Έχοντας παραμετροποιήσει έτσι τον κώδικα αρκεί μόνο να τεθεί στην σταθερά `h_h` το πλήθος των μοντέλων που επιθυμεί ο χρήστης (φυσικά όσο περισσότερα τόσο πιο αργή θα είναι η εφαρμογή στην εκτέλεση καθώς οι σφαίρες είναι αρκετές για να επιβαρύνουν το πρόγραμμα για κάθε μοντέλο – επιπλέον για μεγαλύτερο αριθμό μοντέλων μπορεί να χρειαστεί ένα μικρό `offset` στον πίνακα μεταφοράς για να χωρέσουν και άλλα μοντέλα).
- Σημαδεύοντας με το ποντίκι κάποιο μοντέλο (με χρήση του αλγορίθμου `ray intersect` που αναφέρθηκε και προηγουμένως αν η κατεύθυνση που δείχνει το ποντίκι, δηλαδή η κατεύθυνση της κάμερας δείχνει σε κάποιο μοντέλο) με αριστερό ‘κλικ’ του ποντικιού το μοντέλο αρχίζει τρίγωνο τρίγωνο (όχι από πάνω προς τα κάτω αλλά με τυχαία σειρά) να ξεκολλάει και να το παίρνει ο άνεμος. Σε αυτό το σημείο χρησιμεύουν οι βοηθητικές μεταβλητές της νέας δομής (που αναφέρθηκε στο προηγούμενο ερώτημα) του μοντέλου για να ξέρει το πρόγραμμα σε ποια από τις 3 δυνατές καταστάσεις (`show_human`) βρίσκεται το κάθε μοντέλο (ολόκληρο, υπο καταστροφή και κατεστραμμένο), ώστε να εκτελείται ο κατάλληλος κώδικας κάθε φορά. Η μέθοδος για το ‘`thanos snap effect`’ (δηλαδή να ξεκολλάνε ένα ένα τα τρίγωνα και να τα παίρνει ο άνεμος) είναι η εξής: από το `vector indices` του μοντέλου (που είναι οι θέσεις των σημείων για κάθε τρίγωνο) διαγράφονται σε κάθε καρέ τα 3 τελευταία στοιχεία (δηλαδή ένα τρίγωνο τη φορά) τα οποία αποθηκεύονται προσωρινά για να μετακινηθούν στο χώρο μέσω του πίνακα μετασχηματισμού σε μία πιο απομακρυσμένη θέση (στο επόμενο καρέ) ώστε να γίνει πιο αληθοφανές το εφέ του ανέμου. Σαν `bonus` δυνατότητα υπάρχει η επιλογή ο χρήστης να επιλέξει το μοντέλο με δεξί κλικ, όπου σε αυτή την περίπτωση το μοντέλο εξαφανίζεται ακαριαία, χωρίς το `thanos effect`, μοντελοποιώντας έτσι κάποιο πιο δυνατό ‘όπλο’ συγκριτικά με την προηγούμενη ενέργεια.



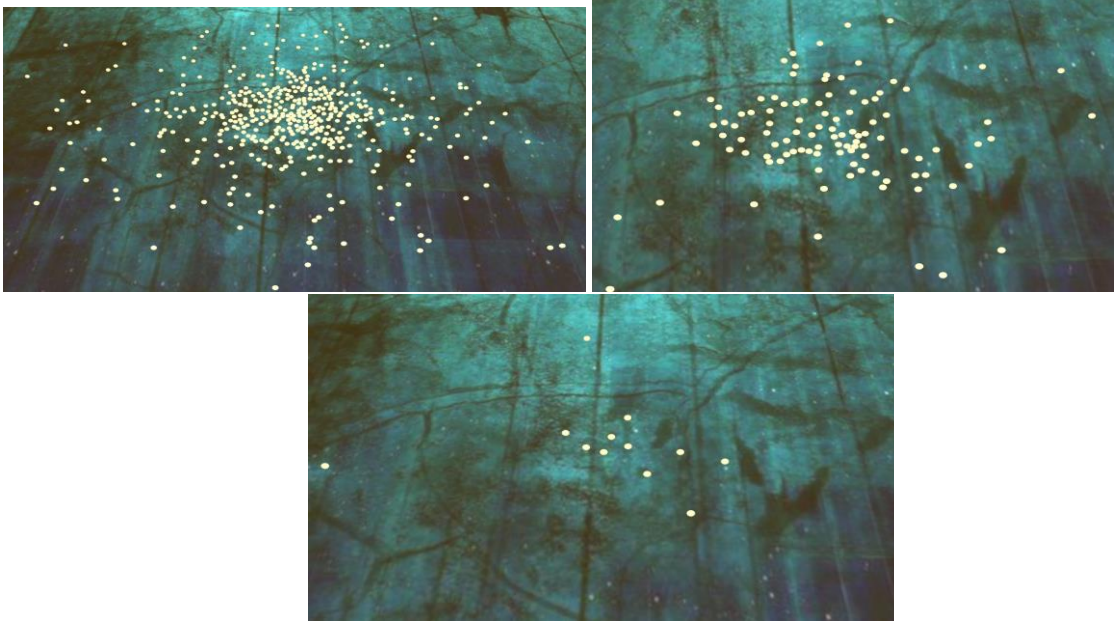




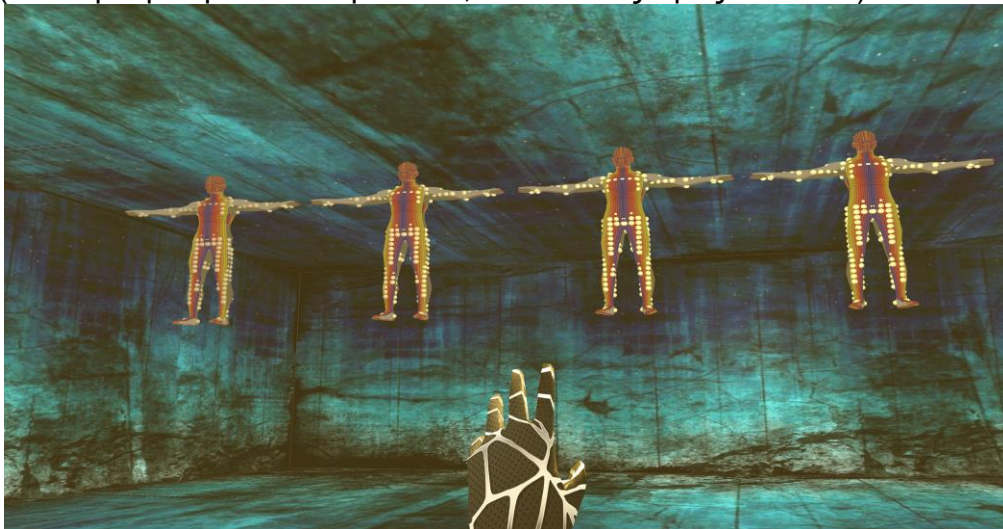
- iv. Με έναν επιπλέον πίνακα στην δομή του μοντέλου (struct) το πρόγραμμα είναι σε θέση να ξέρει κάθε φορά για κάθε μπάλα αν αυτή είναι η στιγμή για να αφεθεί ελεύθερη στον αέρα. Έτσι υπάρχει μία μεταβλητή ύψους που σε κάθε καρέ μειώνεται κατά μία τιμή έτσι ώστε όποια μπάλα (ακίνητοποιημένη μέχρι τώρα) βρίσκεται ψηλότερα από αυτό το ύψος να αφήνεται ελεύθερη να πέσει. Παρακάτω φαίνονται τα στάδια που σιγά σιγά οι μπάλες ελευθερώνονται από τις πιο ψηλές προς τις πιο χαμηλές.



Επιπλέον για καλύτερη προσωμοίωση οι κάθε μπάλα από την στιγμή που ελευθερώνεται έχει έναν συγκεκριμένο χρόνο ζωής (ttl) που θα εμφανίζεται στην οθόνη με αποτέλεσμα να δημιουργείται ένα εφέ εξαφάνισης και στις μπάλες εκτός από τα τρίγωνα



- V. Εισάγεται μία νέα μεταβλητή (game mode) η οποία παίρνει τιμή ανάλογα με τις 2 δυνατές καταστάσεις του παιχνιδιού (α) να μην έχει εξαφανιστεί κανένα μοντέλο και β) να έχει εξαφανιστεί έστω και ένα που σημαίνει ότι το παιχνίδι βρίσκεται σε κατάσταση επιβίωσης-κυνηγητού) Επίσης ελέγχεται στην 2^η περίπτωση αν ο χρήστης βρεθεί σε κοντινή απόσταση με κάποιο μοντέλο 'ζωντανό' (κατάσταση ολόκληρο από το B iii) τότε να χάνει και να τερματίζει το παιχνίδι. Τέλος εισήχθησαν κάποια όρια μετακίνησης ώστε να μην μπορεί ο χρήστης να βγει εκτός από τα όρια του χώρου-δωματίου και επιπλέον να έχει έναν περιορισμό-δυσκολία στην μετακίνησή του για να κάνει το παιχνίδι πιο απαιτητικό και με περισσότερη δράση. Συγκεκριμένα δεν μπορεί κοιτάζοντας πάνω να ανέβει στον κατακόρυφο άξονα πάρα μόνο με τα ειδικά πλήκτρα Q,E (ενώ πριν μπορούσε και με τα W,S κοιτώντας προς τα πάνω).



Game Over!

- Βιβλιογραφία :

1. https://en.wikipedia.org/wiki/M%C3%B6ller%E2%80%93Trumbore_intersection_algorithm
2. <https://stackoverflow.com/>
3. <https://learnopengl.com/>
4. <https://www.turbosquid.com/>
5. <https://www.khronos.org/registry/OpenGL-Refpages/gl4/html/glUniform.xhtml>
6. https://www.glfw.org/docs/3.3/input_guide.html#input_key