



**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**  
Τμήμα Ηλεκτρονικών Μηχανικών  
& Μηχανικών Η/Υ

**ΠΛΗ 418: Γραφική**  
**Project 2014: Flappy Bird 3D**



*Πέγιος Φώτιος 2010030001*

**2013/14**

## 1. Αντικείμενα

Αρχικά, πρέπει να επισημάνω πως όλα τα αντικείμενα δημιουργήθηκαν με vertices, δηλαδή δεν έχει εισαχθεί κάποιο 3D αντικείμενο στο παιχνίδι. Έχουν σχεδιαστεί τα εξής αντικείμενα:

**Σωλήνας:** Με την `InitBuffers` δημιουργείται το αντικείμενο Σωλήνας, ανάμεσα από το οποίο πρέπει να πετάει το πουλί.

**Πουλί:** Το πουλί αποτελείται από τρία αντικείμενα. Αποτελείται από τα `Main Body`, `Left Wing` και `Right Wing`. Δημιουργήθηκαν τρία διαφορετικά καθώς με κάθε flap πρέπει να γίνεται rotate στα φτερά ώστε να κινούνται προς τα κάτω ενώ το `Main Body` να παραμένει ως έχει. Όλα αυτά δημιουργούνται καλώντας τις συναρτήσεις `initBirdBuffers`, `initLeftWingBirdBuffers`, `initRightWingBirdBuffers`.

**Επίπεδο:** Με τη συνάρτηση `initGroundBuffers`, ορίζουμε ένα παραλληλεπίπεδο με το οποίο απεικονίζουμε τέσσερα επίπεδα με διαφορετικά textures σε αυτά. Απεικονίζουμε το έδαφος, τα βουνά στο Background της πίστας, τον ουρανό με τα σύννεφα και τέλος τα βουνά που φαίνονται δεξιά σε περίπτωση που αλλάξουμε την κάμερα σε broadcasting.

## 2. Πίστα

Η πίστα έχει υλοποιηθεί χρησιμοποιώντας τρεις πίνακες, `mapX`, `mapY` και `mapZ` οι οποίοι έχουν μέγεθος 60 (δηλαδή σε κάθε πίστα εμφανίζονται maximum 60 σωλήνες) και περιέχουν αντίστοιχα τις θέσεις στις οποίες εμφανίζονται στον τρισδιάστατο χώρο. Αλλάζοντας πίστα αλλάζουν οι τιμές των πινάκων `mapX` και `mapY`, ενώ ο `mapZ` παραμένει ίδιος καθώς η απόσταση των σωλήνων θέλουμε να παραμένει η ίδια έτσι ώστε να προλαβαίνει το πουλί να περάσει ανάμεσα από το κενό.

## 3. Χειρισμός Πλήκτρων

Τα πλήκτρα που ελέγχουμε σε κάθε tick αν έχουν πατηθεί είναι τα `w,a,d` για το χειρισμό του πουλιού πάνω, αριστερά και δεξιά αντίστοιχα, το `Left Ctrl` για την έναρξη του παιχνιδιού όταν ξεκινάει από την αρχή, αλλά και για προσπάθεια ξανά μόλις ο χρήστης χάσει.

## 4. Textures και Lighting

Σύμφωνα με τα lessons της `webgl` που υπάρχουν στο internet, αλλαχτήκαν οι shaders που υπήρχαν στο αρχικό `sandbox`, με τους κατάλληλους και επίσης προστέθηκαν ορισμένες συναρτήσεις για τη σωστή λειτουργία και εμφάνιση των textures πάνω στις εικόνες. Για το φωτισμό (lighting) προσθέσαμε επίσης στην `Html` κάποια `Input` για να μπορεί ο χρήστης να επιλέγει το direction του φωτισμού μέσα στο παιχνίδι αλλά και το χρώμα αυτού, εμφανίζοντας σκιές για περισσότερο ρεαλισμό.

## 5. Κίνηση Πουλιού / Smoothing

Ο χρήστης καθώς παίζει το παιχνίδι νομίζει πως το πουλί πηγαίνει προς τους σωλήνες, αλλά στην πραγματικότητα το πουλί παραμένει ακίνητο καθ' όλη τη διάρκεια του παιχνιδιού (οι μόνες κινήσεις που κάνει rotate δεξιά και αριστερά όταν στρίβει) και όλη η πίστα (σωλήνες, επίπεδα) έρχονται προς αυτό. Αυτό βέβαιο είναι σχετικό (όπως έλεγε κι ο Αϊνστάιν), γι' αυτό το λόγο ο χρήστης δεν μπορεί να καταλάβει τι πραγματικά γίνεται.

Το πουλί λοιπόν κινείται δεξιά και αριστερά με γραμμικό τρόπο, αλλά επειδή υπάρχει βαρύτητα απογειώνεται και προσγειώνεται δεχόμενο τη δύναμή της. Έτσι λοιπόν ξεκινώντας το παιχνίδι το πουλί αρχίζει να εκτελεί ελεύθερη πτώση προς το έδαφος, αλλά με τη στιγμή που πατήσουμε w για να πετάξει, του δίνουμε μια αρχική ταχύτητα η οποία πάλι μετά από λίγο επηρεάζεται από την επιτάχυνση της βαρύτητας και ξεκινάει πάλι να πέφτει. Αυτό γίνεται κάθε φορά που κάνουμε flap. Με αυτόν τον τρόπο υλοποιήσαμε το smoothing χρησιμοποιώντας τις κινηματικές εξισώσεις.

Σε περίπτωση flap του πουλιού, τα φτερά πέφτουν κάτω στιγμιαία εκτός αν πατηθεί το w αρκετή ώρα άρα μένουν κάτω όσο πατάμε το w.

## 6. Εμφάνιση αντικειμένων

Στη συνάρτηση drawscene εμφανίζουμε τα αντικείμενά μας. Τα αντικείμενα όλα έρχονται προς το πουλί, καθώς σε κάθε tick όπως αναφέραμε και πριν, αυξάνουμε το z στο οποίο έπρεπε να εμφανιστούν έτσι ώστε την επόμενη φορά να έρθουν πιο κοντά στην «κάμερα».

Στην drawscene επίσης γίνεται και το collision detection που αναφέρεται μετά όπως και το smoothing στην κίνηση του πουλιού.

Επίσης, υπάρχουν όρια στην κίνηση του πουλιού. Δηλαδή το πουλί μπορεί να πετάξει μέχρι ένα σημείο. Αν φτάσει σε αυτό και του πούμε πήγαينه πιο ψηλά, αυτό θα παραμείνει εκεί αγνοώντας την εντολή μας. Το ίδιο συμβαίνει για αριστερά και δεξιά, ενώ αν το αφήσουμε να πέσει κάτω, τότε χάνει και πρέπει να ξεκινήσει ξανά από την αρχή.

Επίσης πρέπει να προσθέσουμε πως έχουμε προσθέσει δεύτερη «κάμερα» που επιλέγεται από το μενού παρακάτω κάνοντας στο viewport δύο rotate.

## 6. Collision Detection και Score

Καθώς λοιπόν το πουλί κινείται στο χώρο πρέπει ελέγχουμε αν πέσει πάνω σε κάποιον σωλήνα ή περάσει ανάμεσά του. Ο έλεγχος αυτός λοιπόν πραγματοποιείται με μερικές εμφωλευμένες if. Ελέγχουμε αρχικά αν ο σωλήνας έχει έρθει στο z που βρίσκεται το πουλί. Μετά βλέπουμε αν βρίσκεται μέσα στο εύρος x που βρίσκεται και το πουλί. Τέλος, εφόσον ισχύουν αυτά, το πουλί πρέπει να δούμε αν το πουλί χτυπήσει ή κερδίσει

πόντο ελέγχοντας το εύρος του y του σωλήνα σε σχέση με του πουλιού. Σε περίπτωση collision ή αύξησης του score ακούγονται ανάλογοι ήχοι.

Επειδή το συγκεκριμένο παιχνίδι σε αντίθεση με το γνωστό flappy bird είναι 3D, μπορεί το πουλί να αποφύγει τους σωλήνες χωρίς να χάσει απλά δεν θα κερδίσει πόντους. Δηλαδή, σκοπός είναι να περάσει ανάμεσα από όλο και περισσότερους σωλήνες. Ακόμη κρατάμε σε έναν πίνακα δύο θέσεων τα highscore για κάθε πίστα.

## 7. Personal Touch

Τα επιπλέον στοιχεία που προστέθηκαν στο παιχνίδι είναι:

- Menu επιλογών στην html
- Επιλογή πίστας ανάμεσα σε δύο Level
- Δυνατότητα εναλλαγής soundtrack στο παιχνίδι
- Επιλογή για παύση (pause) στη διάρκεια του παιχνιδιού
- Επιλογή για απενεργοποίηση των ήχων του παιχνιδιού (Mute)
- Δυνατότητα εναλλαγής κάμερας (First Person ή Broadcasting)
- Οδηγίες για το παιχνίδι
- Εμφάνιση σκόρ και πλήθος σωληνών που έχει προσπεράσει το πουλί
- Εμφάνιση αριθμού Level που παίζεται εκείνη τη στιγμή και το highscore σε αυτήν
- Εμφάνιση των highscore σε κάθε level.



Καλή Διασκέδαση!