

Αναφορά 2^{ου} Project Γραφικής

Bowling using Unity3D

Φωτάκης Τζανής - 2014030054

Download Link:

<https://drive.google.com/open?id=0BwzmJoFAcvK9SG9OSFhTaDdZVzQ>

ΞΕΚΙΝΩΝΤΑΣ

Το παιχνίδι αρχίζει με τον παίκτη να βρίσκεται στην είσοδο ενός Bowling. Ανά πάσα στιγμή με το πάτημα του Esc το παιχνίδι τερματίζεται και η εφαρμογή κλείνει. Ο παίκτης μας για να προηγηθεί στον χώρο μπορεί κάνοντας χρήση των πλήκτρων W, A, S, D για κίνηση εμπρός, αριστερά, πίσω και δεξιά αντιστοίχως. Επίσης για την στρέψη του βλέμματός του μπορεί να χρησιμοποιηθεί το ποντίκι. Για να ξεκινήσει κάποιος να παίζει Bowling σε κάποιο διάδρομο πρέπει πρώτα να θέσει τους όρους του παιχνιδιού του, δηλαδή τον αριθμό των ανθρώπινων παικτών και τον αριθμό των τεχνητών παικτών. Συνολικά ένα παιχνίδι μπορεί να έχει 5 παίκτες (όριο για να μπορούν να εμφανίζονται τα σκορ που εξηγούνται παρακάτω), δηλαδή 0-5 ανθρώπινους παίκτες και 0-5 τεχνητούς παίκτες των οποίων το μέγιστο άθροισμα είναι 5. Για να οριστούν οι παραπάνω όροι αρκεί να πλησιάσει ο παίκτης την reception και με τα up-down βέλη να ορίσει τον αριθμό των ανθρώπινων παικτών, επιβεβαιώνοντας/αποθηκεύοντας την επιλογή με το Enter και αντίστοιχα για τον αριθμό των τεχνητών παικτών. Το σχετικό μήνυμα εμφανίζεται το οποίο ανακοινώνει τον αριθμό του διαδρόμου στον οποίο ξεκινά το παιχνίδι. Αμέσως μετά ο παίκτης μας μπορεί να λάβει θέση για βολή σημαδεύοντας με την θέση της μπάλας σε σχέση με τον διάδρομο και εκτοξεύοντας την χρησιμοποιώντας το Space. Το σκορ του παιχνιδιού που παίζεται σε κάθε διάδρομο εμφανίζεται στην οθόνη που βρίσκεται ακριβώς από πάνω του.

3D-Modeling

Ένα μεγάλο μέρος της υλοποίησης του project ήταν η δημιουργία και η εύρεση των κατάλληλων 3D μοντέλων τα οποία αποτελούν τον χώρο και τα αντικείμενα του παιχνιδιού. Όλος ο χώρος του bowling είναι σχεδιασμένος στο πρόγραμμα 3DS-Max (διάδρομοι, μηχανήματα ανάκτησης της μπάλας, κτλ), και αντικείμενα όπως οι καναπέδες, τα τραπέζια, το γραφείο, οι πόρτες, οι οθόνες, ο υπολογιστής είναι κατεβασμένα από το archive3d.net ενώ τα φωτιστικά και ο γραμματέας προέρχονται από το Unity Asset Store.

Sound Effects & Music

Ήχοι αλληλεπίδρασης της μπάλας με το έδαφος και τις κορίνες προέρχονται από:

www.freesoundeffects.com/free-sounds/bowling-10102/

Η μουσική του παιχνιδιού προέρχεται από: www.mixcloud.com

Scripting

- **AINavigation.cs:** Υπεύθυνο για την κίνηση του τεχνητού παίκτη στον χώρο, το σημάδεμα, την εκτόξευση και την ανάκτηση της μπάλας. Ουσιαστικά, η λογική του τεχνητού παίκτη.
- **AIPlayer.cs:** Υποκλάση της κλάσης **Player**, υπεύθυνη για την αρχικοποίηση και ενεργοποίηση των λειτουργιών σε κάθε γύρο του τεχνητού παίκτη.
- **AllayBorderController.cs:** Υπεύθυνο για να δέχεται collisions ένας αόρατος κύβος από τον παίκτη αλλά όχι από την μπάλα. Ο κύβος αυτός λειτουργεί ως όριο για το πόσο μπορεί ο παίκτης να κινηθεί πάνω στον διάδρομο.
- **AllayController.cs:** Υπεύθυνο για την βασική λογική του παιχνιδιού, προτεραιότητα παικτών, γύροι, βολές, αρχικοποίηση και λήξη παιχνιδιού, ανανέωση των σκορ στην οθόνη.
- **BallController.cs:** Υπεύθυνο για την συμπεριφορά της μπάλας, την κατάστασή της, την θέση της, την ενεργοποίηση ηχητικών εφέ κατά την επαφή της με τον διάδρομο αλλά και με τις κορίνες και την εναλλαγή και μετακίνηση των χρησιμοποιούμενων καμερών.
- **ColorChanger.cs:** Υπεύθυνο για τον χρωματισμό και την ένταση των κύριων φωτιστικών δίνοντας την αίσθηση φωτορυθμικού. Χρησιμοποιείται η φασματική ανάλυση της μουσικής που παίζει εκείνη την στιγμή και ανάλογα την ένταση των χαμηλών, των μεσαίων και των υψηλών συχνοτήτων της μουσικής επηρεάζονται οι παράμετροι κόκκινο, πράσινο και μπλε αντίστοιχα των φωτιστικών.
- **FrameScore.cs:** Κλάση η οποία αποθηκεύει τα σκορ για κάθε frame και υπολογίζει μερικό και ολικό σκορ
- **HumanPlayer.cs:** Υποκλάση της κλάσης **Player**, υπεύθυνη για την αρχικοποίηση και ενεργοποίηση των λειτουργιών σε κάθε γύρο του ανθρώπινου παίκτη.
- **MusicPlayer.cs:** Υπεύθυνο για την μουσική του χώρου, κάθε εκκίνηση του παιχνιδιού ξεκινάει το προαναφερθέν mix σε ένα random σημείο δίνοντας την αίσθηση ότι μόλις βρέθηκε ο παίκτης σε ένα ήδη εν λειτουργία χώρο.
- **PinController.cs:** Αναγνωρίζει αν η κορίνα είναι πεσμένη ή όχι.
- **PinSetController.cs:** Υπεύθυνο για τον υπολογισμό του σκορ βολής (ελέγχοντας τον αριθμό των πεσμένων κορίνων) και την διαγραφή των πεσμένων κορίνων.

- `Player.cs`: Διατηρεί το αναλυτικό σκορ ενός παίκτη και υπολογίζει το συνολικό του με βάση τον αριθμό των ολοκληρωμένων γύρων.
- `ReceptionController.cs`: Υπεύθυνο για την αρχικοποίηση της λογικής του παιχνιδιού, τον ορισμό των παραμέτρων παρτίδας (αριθμό ανθρώπινων/τεχνητών παικτών) και την επιλογή του διαθέσιμου διαδρόμου.