

Καφαδάρης Γιώργος – 2457 Νάκος Ιωάννης – 2502

2/12/16

1η ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ

CUBECEPTION

README

1. Βιβλιοθήκες

<GL/glut.h> <GL/glu.h> <GL/gl.h> : Για να χρησιμοποιήσουμε τις βιβλιοθήκες opengl και glut.

<time.h>: Για να χρησιμοποιήσουμε τη συνάρτηση rand().

<string> <string.h><stdlib.h>: Για να συνενώσουμε 2 string και για να μετατρέψουμε το int σε string.

<math.h>: Για να χρησιμοποιήσουμε τις συναρτήσεις sin και cos.

<stdio.h>: Για να τυπώνουμε.

<iostream><stdio.h>: Για τη χρησιμοποίηση του cout και cin. Καθώς και printf.

2. Βασικές δηλώσεις

main:

int dimension: Το μέγεθος του του map (dimension*dimension*dimension).

int remcubes: Οι διαθέσιμοι κύβοι.

int points: Οι πόντοι του παίχτη.

int lives: Οι ζωες που του παίχτη.

char *cubeLayout**: Ο πίνακας που προσωμοιώνουμε τους κύβους.

int playerPosx, playerPosy, playerPosz: Οι συντεταγμένες του παίχτη.

enum color[]: Τα χρώματα των κουτιών.

Int debug: ενεργοποιεί/απενεργοποιεί το debug Mode.

int lights: ενεργοποιεί/απενεργοποιεί τον φωτισμό.

GLfloat viewDirx, viewDiry,viewDirz: το view Direction του παίχτη.

Camera:

struct SF3dVector,SF2dVector,SI3dVector: Το F υποδειλώνει float, το I integer. Το 2 υποδειλώνει 2 παραμέτρους και το 3 υποδειλώνει 3 παραμέτρους. Χρησιμοποιούνται για την αποτελεσματικότερη ομαδοποίηση μεταβλητών σε όλο το μήκος του κώδικα.

RotatedX/Y/Z: η περιστροφή της κάμερας γύρω από τους αντίστοιχους άξονες.

ViewDirChanged: γίνεται ίση με 1 μετά από κάθε περιστροφή γύρω από το X,Y άξονα.

3. Βασικές συναρτήσεις του κώδικα

Main:

kickbox(): Μετακινεί τον κύβο που βρίσκεται μπροστά στον παίχτη(καλείται με το δεξί κλικ) κατά 1 προς την κατεύθυνση που κοιτάει (**viwDirx,vieDirz**). Ελέγχει αν ο εν λόγω κύβος πρόκειται να βγεί εκτός πλέγματος και αν ναι στην επόμενη κλήση της τον διαγράφει. Προσδιορίζει τον άξονα(X,Z) στον οποίο θα μετακινηθεί ο κύβος με βάση τον προσανατολισμό του παίχτη.Για να το καταφέρει αυτό υπολογίζει ποιά είναι η καταλληλότερη θέση μετακίνησης του κύβου καλώντας την **calcNextBoxPos()**.Για να μετακινήσει τον κύβο κατά μια θέση υπολογίζει την πρώτη διαθέσιμη θέση, στον άξονα μετακίνησης(X,Z) (**FirstEmptyBox**), και ξεκινώντας από αυτή ανταλλάζει το περιεχόμενο μιας διαθέσιμης θέσης με αυτό της προηγούμενης μέχρι να φτάσει 1 θέση πριν τον παίχτη(**playerPos-1**).

drawText(): Προσδιορίζει το κείμενο που πρόκειται να τυπωθεί στην οθόνη και ανανεώνει τους πόντους, τις ζωές και τους κύβους που έχουν απομείνει στον παίχτη.

drawBox(): Ζωγραφίζει έναν κύβο τυχαίου χρώματος(B,G,R,Y), καθώς και το κύβο στη μέση του map χρώματος magenta, στις συντεταγμένες x,y,z μεγέθους 1.

drawNet(): Σχεδιάζει το πλέγμα των κύβων στο 0 επίπεδο. Διαβάζει απο τον πίνακα **cubeLayout[y][x][z]**, στον οποίο περιέχετε το map του παιχνιδιού, το χρώμα κάθε κύβου προς σχεδίαση και καλεί την **drawBox()** για να τον ζωγραφίζει.

addBox(): Προσθέτει έναν κύβο(καλείται με το αριστερό κλικ) στο ίδιο επίπεδο (**playerPosy**) με τον παίχτη στην κατεύθυνση που κοιτάει ο παίχτης. Η καταλληλότερη θέση τοποθέσης του καινούριου κύβου υπολογίζεται με τη χρήση της **calcNextBoxPos()**. Δεν θα προσθέσει κύβο εκτός πλέγματος και αν το απόθεμα κύβων του παίχτη (**remcubes**) είναι μηδενικό. Με κατάλληλο έλεγχο ,σε περίπτωση που ο παίχτης έχει ήδη κύβο μπροστά του και στο ίδιο επίπεδο με αυτόν, ο καινούριος κύβος προστίθεται πάνω στον υπάρχον.

KeyDown(): Είναι υπεύθυνη για την αντιστοίχιση των πλήκτρων του πληκτρολογίου με τις κατάλληλες συναρτήσεις του προγράμματος.

calculateE(): Καλείται απο την **KeyDown()** με το πάτημα του κουμπιου 'e' και εξαφινίζει όλα τα κουτια που βρίσκονται στον άξονα των y στην κατεύθυνση που κοιτάει ο παίχτης. Συγκεκριμένα, υπολογίζει, με την χρήση της **calcNextBoxPos()**, σε πιο κύβο είναι προσανοτισλισμένος ο παίχτης και έπειτα διαγράφει επαναληπτικά τους απαρέτιτους κύβους, ξεκινώντας απο το 0 επίπεδο μέχρι τον τελευταίο κύβο.

calculateQ(): Καλείται απο την **KeyDown()** με το πάτημα του κουμπιου 'q' και σβήνει το κουτί που βρίσκεται μπροστά στον παίχτη στο ίδιο επίπεδο(Y) με αυτόν.

calculateR(): Χρησιμοποιείται ώστε να «πέσουν κάτω» όλα τα κουτιά που βρίσκονται στον «αέρα». Δηλαδή αυτά που δεν έχουν κανένα κουτί από κάτω τους. Συγκεκριμένα με τη χρήση επανάληψης ελέγχει κάθε καταχώρηση του πίνακα(**cubeLayout**) και οποία αντιστοιχεί σε κύβο ο οποίος δεν εφάπτεται με κάποιον άλλο κύβο απο την κάτω πλευρά μετακινήτε ένα επίπεδο πιο κάτω, μαζί με τυχόν κύβους οι οποίοι βρίσκονται στην ίδια θέση αλλά σε μεγαλύτερο επίπεδο.

reset(): Σε περίπτωση που ο παίχτης βρεθεί στο επίπεδο 0 χάνει μία ζωή και μετακινείται στη θέση

$(\text{dimension}/2|1|\text{dimension}/2)$ του map.

CalcNextBoxPos(): Αθροίζει τις συντεταγμένες της κάμερας με τις συντεταγμένες της κατεύθυνσης θέασης επί το βήμα και επιστρέφει το άθροισμα. Αυτό προσδιορίζει την θέση ακριβώς μπροστά του παίχτη σε απόσταση 1 από την τρέχουσα θέση του.

moveInY(): Η συνάρτηση αυτή υλοποιεί την πτώση καθώς και την ανάβαση του παίχτη από/σε κύβο. Συγκρίνει την τρέχουσα θέση του παίχτη με μια θέση αυξημένη κατά 1 επίπεδο καθώς και με μια μειωμένη κατά 1 επίπεδο. Αν η θέση από κάτω του παίχτη είναι κενή ('E') τότε πραγματοποιείτε πτώση. Αν η θέση στο ίδιο επίπεδο με τον παίκτη περιέχει κύβο, αλλά μόνο έναν, τότε πραγματοποιείτε ανάβαση. Σε κάθε άλλη περίπτωση ο παίχτης απωθείται προς τα πίσω.

colorpicker(): Επιστρέφει τυχαία ένα χρώμα(B,G,Y,R) με το οποίο θα χρωματιστεί ο κύβος.

OutOfBoundsForwards/Backwards(): Περιορίζει την κίνηση του παίκτη στα πλαίσια του πλέγματος. Με την χρήση της **CalcNextBoxPos()** ελέγχει αν η επόμενη θέση(μπροστά του/πίσω του παίχτη) είναι εκτός ορίων. Αν ναι τότε δεν επιτρέπει την μπροστά/πίσω μετακίνηση.

drawOutline(): Ζωγραφίζει το περίγραμμα των κύβων ώστε να ξεχωρίζουν μεταξύ τους.

initLights(): Τοποθετεί 4 spotlights ένα σε κάθε γωνία του πλέγματος.

InitCubeLayout(): Η συνάρτηση αυτή αρχικοποιεί τις διαστάσεις του πίνακα *****cubeLayout**, του δίνει τις αρχικές τιμές με την κλήση της **colorPisker()** και τέλος ορίζει όσα κελιά δεν είναι του 0 επιπέδου ίσα με 'E' empty.

Display(): Η συνάρτηση αυτή είναι υπεύθυνη για την εμφάνιση των στοιχείων στην οθόνη. Αυτή η συνάρτηση καλεί την **glutDisplayfunc()**. Καθορίζει τον φωτισμό της σκηνής, καλεί την **DrawNet()** και τυπώνει το σκορ,ζωές,και πόντους του παίχτη στην οθόνη.

Camera:

Ccamera::SetPlayerPos(GLfloat x, GLfloat y, GLfloat z): καλείται απο την main και αρχικοποιεί το Position F3dVetor στο X,Y,Z.

Ccamera::GetPlayerPos: επιστρέφει το F3dVetor Position στην main το οποίο μετατρέπεται σε integer και αντιστοιχεί στην θέση του παίχτη(**PlayerPos**).

Ccamera::GetPlayerViewDir(): Μετατρέπει να περιστοφική κίνηση γύρο απο τους άξονες X,Y σε μοίρες.

Ccamera::Move(SF3dVector Direction): Υλοποιεί την κάθετη κίνηση,προς τα πάνω ή προς τα κάτω (άξονας Y). Προσθέτει το Direction στο Position.

Ccamera::RotateY/RotateX(GLfloat Angle): Προσθέτει στην μεταβλητή **RotatedY** το Angle. Η περιστροφή πραγματοποιείται στην Render με χρήση **glRotatef()**.Η **RotateX** είναι ενεργή **μόνο** στο Debug Mode.

Ccamera::StrafeRight(GLfloat Distance): Πραγματοποιεί κίνηση δεξιά ή αριστερά με την κάμερα να κοιτάει ευθεία. Ενεργή **μόνο** στο Debug Mode.

COMPILING:

Για την επιτυχή μεταγλώττιση του προγράμματος αρκεί να δωθεί η εντολή **make** στο τερματικό και έπειτα τρέχει με το όνομα **Cubeseption.out** . Το πρόγραμμα λειτουργεί όπως ζητήθηκε στην εκφώνηση της άσκησης **EKTOS** debugging mode. Το debugging mode προσφέρει απλώς επιπλέον δυνατότητες στον χρήστη η οποίες κατα την κανονική εκτέλεση δεν είναι απαραίτητες!