

14/1/2016

Ράντογλου – Μελάς Σπυρίδων AM:2342

Μιγκοτζίδης Παναγιώτης AM:2999

ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΗ ΑΣΚΗΣΗ 2 – README

1.Βιβλιοθήκες που κάνουμε include στο αρχείο:

`<iostream>` `<fstream>` Όστε να μπορούμε να διαβάσουμε το file.maz .

`<GL/glut.h>` `<GL/gl.h>` `<GL/glu.h>` Όστε να χρησιμοποιήσουμε τις βιβλιοθήκες OpenGL και Glut.

`<time.h>` Όστε να χρησιμοποιήσουμε την συνάρτηση rand().

`<math.h>` Όστε να χρησιμοποιήσουμε τις συναρτήσεις sin() και cos().

`<string>``<string.h>``<stdlib.h>` Όστε να μετατρέψουμε int σε string και να συνενώσουμε 2 strings.

`<vector>` Όστε να χρησιμοποιήσουμε την δομή vector για την καταχώρηση των τιμών του αρχείου file.maz.

`<soil.h>` Για να φορτώσουμε τα textures.

2.Βασικές μεταβλητές που χρησιμοποιούμε στον κώδικα:

vector<vector<vector<string>> maze : Είναι ένα τρισδιάστατο vector το οποίο κρατάει τις τιμές string E,G,B,R,T1,T2,T3 οι οποίες καθορίζουν τον τύπο του κύβου για κάθε επίπεδο. Το vector είναι στην μορφή γ,χ,z δηλαδή π.χ. το maze[0][1][2] είναι ο τύπος για το κύβο στο επίπεδο 0 με x=1 και z=2.

int dimension : Η διάσταση του κάθε επιπέδου.

int levels: Το πλήθος των επιπέδων.

int totalHammers: Το πλήθος των σφυριών.

int hammers: Το πλήθος των σφυριών που απομένουν στον χρήστη.

int score: Το score του παίχτη.

int moves: Οι κινήσεις του παίχτη.

enum colors : Κάνει enumerate τους τύπους των χρωμάτων για ευκολία.

double prob: Την χρησιμοποιούμε για να καθορίσουμε στην αρχή την τυχαία θέση του παίχτη.

float eyex,eyez,eyey: Κρατάν τις συντεταγμένες για τα τρία πρώτα ορίσματα της gluLookAt() δηλαδή για το που βρίσκεται ο παίχτης.

int Posx,Posy,Posz: Κρατάν τις συντεταγμένες στις οποίες βρίσκεται ο παίχτης.

float centerx,ceentery,centerz: Κρατάν τις συντεταγμένες για την δεύτερη τριάδα ορισμάτων της gluLookAt() δηλαδή για το που κοιτάει ο παίχτης.

GLUquadric* obj: Ο κύλινδρος.

int viewState: Εάν είναι 0 σημαίνει ότι ο παίχτης κοιτάει από έξω τον λαβύρινθο. Αν είναι 1 σημαίνει ότι ο παίχτης βρίσκεται μέσα και αν είναι 2 σημαίνει ότι το παιχνίδι έχει τελειώσει.

GLuint tex_2d,GLuint tex_2d1,GLuint tex_2d2,GLuint tex_floor:

Οι παραπάνω μεταβλητές κρατάν τις τιμές των pixel για κάθε texture και για το πάτωμα(το αρχείο περιλαμβάνεται στο turnin).

int transparency: Είναι 0 εάν το λαβύρινθος δεν είναι ημιδιαφανής και 1 εάν είναι.

float lx,lz,ly: Είναι τα διανύσματα κατεύθυνσης στους άξονες x,y,z αντίστοιχα.

3.Συναρτήσεις που χρησιμοποιώ στον κώδικα:

Όλες οι παρακάτω συναρτήσεις είναι τύπου void.

loadGLTextures(): Διαβάζει τα textures με την χρήση της SOIL και τα αποθηκεύει στις αντίστοιχες μεταβλητές. Εάν υπάρχει πρόβλημα κατά το διάβασμα του αρχείου εμφανίζει το κατάλληλο μήνυμα λάθους.

readFile(): Διαβάζει το αρχείο file.maz και καταχωρεί τις τιμές του στο διάλυσμα maze. Επίσης επιλέγει τυχαία την αρχική θέση του παίχτη με την χρήση του reservoir sampling.

(https://en.wikipedia.org/wiki/Reservoir_sampling)

calculateScore(): Υπολογίζει το σκορ του παίχτη με βάση τον τύπο που δίνεται από την εκφώνηση και το καταχωρεί στην μεταβλητή σκορ.

renderBitmapString(int x,int y,int z,char *string): Την χρησιμοποιούμε για να τυπώνουμε τους χαρακτήρες στην οθόνη.

drawCube(int x,int y,int z,int color): Σχεδιάζει έναν κύβο στις συντεταγμένες x,y,z με μέγεθος 5 και χρώμα color. Επίσης κάνει έλεγχο αν ο κύβος πρέπει να είναι ημιδιαφανές. Μετά την σχεδίαση, σχεδιάζει και έναν περίγραμμα γύρω-γύρω από τον κύβο ώστε να είναι ευδιάκριτοι. Για την σχεδίασή τους έχουμε χρησιμοποιήσει την εντολή glQuads για κάθε πλευρά και ύστερα τις εντολές vertex3f και normal3f.

drawTextureCube(int x,int y,int z,int textureType): Σχεδιάζει έναν κύβο με την αντίστοιχη διαδικασία με την διαφορά ότι ο κύβος είναι textured.

drawOutline(int x,int y,int z): Σχεδιάζει το κατάλληλο περίγραμμα γύρω από τον κύβο. Καλείται μέσα στην drawCube.

drawCylinder(int x,int y,int z): Σχεδιάζει τον κύλινδρο και τον κάνει translate στην θέση που βρίσκεται ο παίχτης.

changeSize(int w,int h): Χρειάζεται για την αυξομείωση του παραθύρου.

renderScene(): Αυτή η συνάρτηση καλείται στην glutDisplayFunc. Αρχικά ανάλογα με την τιμή του viewState καθορίζει που θα κοιτάει η κάμερα. Επίσης καθορίζει τον φωτισμό στην σκηνή. Ύστερα σχεδιάζει το πάτωμα και τους κύβους. Αν το παιχνίδι έχει τελειώσει τυπώνει το σκορ.

mouseButton(int button,int state ,int x,int y),mouseMove(int x,int y):

Αυτές οι δύο συναρτήσεις χρησιμεύουν ώστε να αλλάζει η κατεύθυνση που κοιτάει ο παίχτης όταν πατάει το αριστερό κλικ και στρίβει την κάμερα προς οποιαδήποτε κατεύθυνση. Η mousebutton ελέγχει εάν έχει πατηθεί το αριστερό κλικ και εάν ναι κρατάει της συντεταγμένες και η mouse move εάν είναι πατημένο το κλικ θα αυξήσει ή θα μειώσει τις μεταβλητές lx,ly και lz οι οποίες βρίσκονται μέσα στην gluLookAt με αποτέλεσμα να αλλάξει το σημείο όπου κοιτάει ο παίχτης.

processNormalKeys(unsigned char key,int x,int x): Υλοποιεί όλες τις λειτουργίες του πληκτρολογίου. Εάν ο χρήστης κοιτάει από πάνω το maze τότε όσο πατάει "R" μετακινείται γύρω από το maze κάνοντας κυκλική κίνηση καθώς οι τιμές των eyex eyez αλλάζουν ώστε να περιστρέφεται γύρω από το σημείο που βρίσκεται στην μέση του λαβυρίνθου. Αν ο παίχτης κοιτάει εκτός και πατήσει n τότε μεταφέρεται στην θέση του λαβυρίνθου που βρισκόταν πριν δει το maze από έξω. Αλλιώς γίνεται το αντίθετο δηλαδή κοιτάει από πάνω το maze. Αν ο χρήστης πατήσει w τότε αρχικά υπολογίζουμε την απόσταση που θα διανύσει (eyex -3*lx) (eyez -3*lz) και εάν δεν «πέφτει» πάνω σε κάποιο κυβάκι ή εάν δεν βγαίνει εκτός ορίων του maze κάνοντας αυτή την απόσταση τότε προχωράει. Για να κάνουμε αυτό τον έλεγχο κοιτάμε το vector maze και ελέγχουμε ένα το κυβάκι που θα πάει είναι τύπου «E» ή «W». Η αντίστοιχη διαδικασία γίνεται και όταν ο παίχτης πατήσει το πλήκτρο s για να μετακινηθεί πίσω. Η κίνηση είναι συνεχής δηλαδή ο χρήστης πατώντας μία φορά το «w» ή το «s» θα μετακινηθεί λίγο και όχι ανά κυβάκι. Επίσης κάθε φορά που γίνεται μια κίνηση γίνεται έλεγχος για το αν ο παίχτης έχει κυβάκι από κάτω του ή κενό και αν έχει κενό τότε μετακινείται προς τα κάτω όσα επίπεδα χρειάζεται. Όλα αυτά γίνονται αλλάζοντας της μεταβλητές eyex,eyez,eyez οι οποίες καθορίζουν την θέση της κάμερας. Όταν ο χρήστης πατήσει το πλήκτρο "h", εάν έχει κάποιο σφυρί στην διάθεσή του, τότε θα διαγραφτεί το

κυβάκι(θα γίνει «Ε») που κοιτάει αρκεί να βρίσκεται σε απόσταση ένα. Αυτό γίνεται παίρνοντας τα διανύσματα lx, ly, lz και υπολογίζουμε προς ποια κατεύθυνση κοιτάει ο παίχτης. Αν ο χρήστης πατήσει το πλήκτρο “space” τότε αν είναι στο τελευταίο επίπεδο το παιχνίδι τελειώνει και υπολογίζεται το σκορ, αλλιώς γίνεται έλεγχος για το αν το κυβάκι που βρίσκεται από πάνω του είναι «Ε» και αν είναι ανεβαίνει ένα επίπεδο. Αν ο παίχτης πατήσει «x» το παιχνίδι τελειώνει επίσης και το score γίνεται 0. Επίσης το παιχνίδι τελειώνει αν το παίχτης πατήσει Ε και βρίσκεται στο τελευταίο επίπεδο.

Bonus 1 : Σε κάθε επίπεδο υπάρχουν δύο μαύροι κύβοι. Όταν ο παίχτης πατήσει “w” ή “s” και «πέσει» πάνω σε έναν από αυτούς, τότε γίνεται μια αναζήτηση στο επίπεδο που βρίσκεται ο παίχτης μέχρι να βρεθούν οι συντεταγμένες του άλλου μαύρου κύβου. Όταν βρεθούν τότε ο παίχτης «μεταφέρεται» εκεί καθώς αλλάζει η θέση της κάμερας. Στο αρχείο file.maz γύρω από τα μαύρα κουτιά υπάρχει κενό έτσι ώστε να είναι ορατό το τι γίνεται στην περίπτωση της «μεταφοράς».

Bonus 3 : Όταν ο παίχτης κοιτάει το επίπεδο από έξω, πατώντας «a» όλοι οι κύβοι γίνονται ημιδιαφανείς. Αυτό γίνεται με την χρήση του glColor4f όπου το τελευταίο όρισμα είναι μικρότερο του 1. Αν το ξαναπατήσει αποκτούν ξανά το αρχικό τους χρώμα.

Σημείωση : Το πρόγραμμα γίνεται compile με την εντολή : g++ -o maze maze.cpp -ISOIL -IGLU -lglut -IGL . Το πρόγραμμα τρέχει στους υπολογιστές opti7020XX του τμήματος.