STRUCTURED PROGRAMMING SIMULATOR
ΕΡΓΑΣΙΑ ΣΤΟΝ ΔΟΜΗΜΕΝΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2022-2023

ΜΑΘΗΜΑ : ΔΟΜΗΜΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ(004)

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ ΑΠΘ

ΣΤΕΦΑΝΟΣ ΧΑΤΖΗΚΥΡΙΑΚΟΥ
chatzikys@ece.auth.gr

1. ΠΕΡΙΕΧΌΜΕΝΑ

2. ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ

Το θέμα που επιλέχθηκε ήταν το παιχνίδι με τίτλο

ΠΕΡΙΠΕΤΕΙΑ ΣΕ ΔΩΜΑΤΙΟ ΔΙΑΦΥΓΗΣ

Το πρόγραμμα υλοποιήθηκε ακολουθώντας όλες τις προδιαγραφές που δόθηκαν.

θέμα παιχνιδιού:

Έτος 2222.

Μετά από 200 χρόνια βαθιάς κρυογονικής ύπνωσης ξυπνάτε σε μια διαφορετική πραγματικότητα. Βρίσκεστε κλειδωμένος σε ένα δωμάτιο διαφυγής (escape room). Όμως τα δωμάτια διαφυγής έχουν εξελιχθεί σε επικίνδυνα παιχνίδια πραγματικότητας και ο νικητής κερδίζει €100.000.000. Το δωμάτιο διαφυγής που έχετε μπει είναι ένας υποθαλάσσιος λαβύρινθος. Ο λαβύρινθος έχει διάφορα μονοπάτια με κλειδωμένες πόρτες σε διάφορα σημεία. Η κάθε πόρτα ανοίγει με ένα κωδικό και έχετε 3 προσπάθειες να την ανοίξετε, αλλιώς μπλοκάρει. Πριν από κάθε πόρτα υπάρχει μία οθόνη που εμφανίζει ένα πρόβλημα, το οποίο θα πρέπει να λύσετε για να σας αποκαλυφθεί ο κωδικός ή τμήμα του κωδικού, ώστε να μπορέσετε να προχωρήσετε. Προτείνεται τα προβλήματα που εμφανίζονται να είναι ερωτήματα για τη γλώσσα C, είτε ως κώδικας, είτε ερωτήσεις πολλαπλών επιλογών, χωρίς να αποκλείεται ο συνδυασμός τους με ερωτήματα άλλων πεδίων. Για να νικήσετε θα πρέπει να βρείτε την έξοδο. Ένα πρόβλημα όμως είναι ότι ο λαβύρινθος παίρνει νερά από τη θάλασσα και η στάθμη του νερού ανεβαίνει, επομένως έχετε περιορισμένο χρόνο για να διαφύγετε…

Το παιχνίδι έχει σχετικά μικρή διάρκεια όπως ζητήθηκε.

3. ΟΔΗΓΙΕΣ ΜΕΤΑΓΛΩΤΤΙΣΗΣ ΚΑΙ ΕΓΚΑΤΑΣΤΑΣΗΣ.

Αυτό το αρχείο είναι μέρος από ένα συμπιεσμένο αρχείο το οποίο περιέχει μέσα τον πηγαίο κώδικα μαζί με όλα τα επιπλέον αρχεία που απαιτούνται για τη μεταγλώττιση του, το εκτελέσιμο μαζί με τα αντίστοιχα αρχεία του, αυτό το έγγραφο τεκμηρίωσης και το έγγραφο άδειας δικαιωμάτων ανοιχτού κώδικα.

Στον φάκελο ***game*** βρίσκεται ο πηγαίος κώδικας όπως και ένα έτοιμο εκτελέσιμο αρχείο του παιχνιδιού στον φάκελο ***exec***.

Αν κάποιος όμως επιθυμεί να κάνει τη μεταγλώττιση μπορεί να το κάνει με μεγάλη ευκολία χρησιμοποιώντας το ***compile.bat***, το οποίο περιέχει όλες τις εντολές για τη μεταγλώττιση του προγράμματος. Επίσης υπάρχει αντίγραφο της βιβλιοθήκης ***raylib.h*** .

Τα μόνα αρχεία που χρειάζονται για την εκτέλεση του προγράμματος, είναι το εκτελέσιμο αρχείο όπως και ο φάκελος ***assets***, που περιέχει τα γραφικά.

4. ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ.

Όπως αναφέρεται και στην αρχική οθόνη του παιχνιδιού τα πλήκτρα χειρισμού είναι τα εξής ,

Τα πλήκτρα ***βελών*** για την κίνηση του παίκτη μέσα στο χάρτη

Το πλήκτρο ***S*** για την έναρξη του παιχνιδιού

Τα πλήκτρα ***A B C D*** για την επιλογή της σωστής απάντησης

Το πλήκτρο ***R*** σε περίπτωση που ο παίκτης εγκλωβιστεί λόγω κλειδωμένων πορτών.

Και το πλήκτρο ***M*** για την σίγαση της μουσικής.

Επιπλέον λειτουργεί και το πλήκτρο ***ESC*** για την έξοδο του χρήστη και από τα μενού ερωτήσεων αλλά και την έξοδο του χρήστη από το πρόγραμμα.

5. ΠΑΡΑΡΤΗΜΑ

5.1. ΚΑΤΑΓΡΑΦΗ ΟΛΩΝ ΤΩΝ ΣΥΝΑΡΤΗΣΕΩΝ

Πριν αρχίσουμε να αναλύουμε κάθε αρχείο θα ασχοληθούμε με κάποιες σημαντικές δομές από το αρχείο ***game.h***.

5.1.1. Σημαντικές δομές από το αρχείο game.h

```
typedef enum AppStates
{Loading, Menu, Game, Questions, Final, GameEnd, Quit};
```

Αυτό είναι ένα από τα πιο βασικά στοιχεία του προγράμματος μας επιτρέπει να εναλλάσσουμε τις λειτουργίες του προγράμματος πολύ εύκολα και να είναι πιο συμμαζεμένος ο κώδικας

```
struct player
{
Vector2 p;
enum direction dir;
int h;
int w;
Color color;
Rectangle player_rect;
};
```

Αυτή είναι επίσης μια πολύ σημαντική δομή η οποία περιγράφει τις συντεταγμένες του παίκτη , την κατεύθυνση που κοιτάει ο παίκτης, το μέγεθος του, και το χρώμα του(Σημείωση: το χρώμα του παίκτη όπως και το τετράγωνο του παίκτη είναι 2 στοιχεία τα οποία χρησιμοποιήθηκαν κατά την ανάπτυξη του προγράμματος και δεν εμφανίζονται πλέον στο τελικό πρόγραμμα έχουν μείνει όμως μέσα για διάφορους λόγους)

5.1.2. Συναρτήσεις στο αρχείο movement.h

```
void initPlayer(struct player* ptr, Vector2 spawn)
```

Αυτή η συνάρτηση αρχικοποιεί την θέση του παίκτη μέσα στο χάρτη, την κατεύθυνση που κοιτάει, το χρώμα του τετραγώνου του παίκτη όπως και το ίδιο το τετράγωνο. Αξίζει να αναφερθεί ότι το λεγόμενο τετράγωνο δεν εμφανίζεται στο τελικό πρόγραμμα απλά επειδή τα γραφικά τα οποία χρησιμοποιούνται είναι ακριβώς πάνω από το από το τετράγωνο οπότε δεν υπάρχει λόγος να εμφανίζεται. Αυτή η συνάρτηση παίρνει ως παραμέτρους έναν δείκτη σε δομή ***struct player*** όπως και το διάνυσμα ***Vector2 spawn***

```
void UpdatePlayerPos(struct player* chara, Rectangle* rec)
```

Ίσως και η πιο χρήσιμη αλλά συγχρόνως περίπλοκη συνάρτηση σε όλο το πρόγραμμα . Αυτή η συνάρτηση ανανεώνει την θέση του παίκτη μέσα στον χάρτη όπως και την κατεύθυνση που κοιτάει ώστε να αλλάξουν τα γραφικά αντίστοιχα πιο αναλυτικά έχει σαν παραμέτρους έναν δείκτη σε δομή ***struct player*** , όπως και ένα τετράγωνο το οποίο αντιπροσωπεύει το μοντέλο του παίκτη. αρχικά αυτή η συνάρτηση ελέγχει άμα ο χρήστης πιέζει πλήκτρα κατευθύνσεως, αλλάζει τη θέση του παίκτη και μετά καλεί μια συνάντηση η οποία ελέγχει άμα ο παίκτης είναι κάπου που δεν θα έπρεπε να είναι, δηλαδή ελέγχει για ***collision***, τη λεγόμενη συνάρτηση θα την δούμε σε λίγο.

```
void CheckBounds(struct player* chara)
```

Αυτή η συνάρτηση αν και σε αυτό το στάδιο του προγράμματος δεν έχει πολλή χρήση, ήταν από τις πρώτες συναρτήσεις που γραφτήκαν, η οποία ελέγχει και οριοθετεί τη θέση του παίκτη μέσα στο παράθυρο εκτέλεσης και έχει παραμείνει μέσα στο πρόγραμμα γιατί δεν το επηρεάζει αρνητικά.

```
void UpdatePlayerDir(Rectangle* rec, struct player* chara)
```

Επίσης μια πολύ μικρή συνάρτηση η οποία αυτό που κάνει είναι να ελέγχει αν υπαρχει αλλαγή στην κατεύθυνση του παίκτη και να ανανεώνει το πλάτος του τετραγώνου του μοντέλου ώστε να φαίνεται στη σωστή κατεύθυνση. και εδώ πρέπει να πούμε ότι αν μια εικόνα έχει αρνητικό πλάτος αυτό απλά κάνει ένα ***flip*** την εικόνα.

5.1.3. Συναρτήσεις στο αρχείο map.h

Σημαντικό είναι να αναφέρουμε ότι ο χάρτης του παιχνιδιού είναι μια δισδιάστατη συστοιχία τύπου ***int*** στην οποία κάθε στοιχείο της να αντιστοιχεί σε στοιχείο ***enum Block***. Επίσης να αναφερθεί εδώ ότι κάθε στοιχείο μέσα στο πρόγραμμα όπως και ο παίκτης απεικονίζονται σαν ένα τετράγωνο διαστάσεων ***blocksize****(το οποίο έχουμε ορίσει ότι είναι ίσο με* ***32****)*

```
void UpdateMap(int map[][(int)Width / blocksize])
```

Αυτή η συνάρτηση καλείται ούτε να ανανεώσει τον χάρτη καλώντας την ***DrawBlock***

```
void DrawBlock(int i, int j, enum Block block)
```

Αυτή είναι η βασική συνάρτηση για την εμφάνιση του χάρτη καλείται από διάφορες συναρτήσεις και λαμβάνει ως παραμέτρους συντεταγμένες ***int i*** , ***int j*** και τον τύπο του ***block*** που πρέπει να εμφανίσει. Επίσης αυτή η συνάρτηση ορίζει την αρχική θέση του παίκτη στο επίπεδο μέσω μιας μεταβλητής τύπου διανύσματος ***Vector2 spawnpoint***

```
int CollisionDetect(int map[][(int)Width / blocksize], struct
player* chara, enum direction dir)
```

Ακόμα μια αρκετά περίπλοκη συνάρτηση της οποίας ο σκοπός είναι να ελέγχει κάθε φορά την θέση που πάει ο παίκτης και να τον αποτρέπει να περνάει μέσα από τοίχους ή να βγαίνει εκτός ορίων . Ελέγχει έχοντας υπόψη την θέση του παίκτη αλλά και το πού θέλει να πάει το επόμενο ***block*** αποτρέποντας τον παίκτη σε περίπτωση που πάει να μπει μέσα σε κάποιον τοίχο. Επίσης αυτή η συνάρτηση αλλάζει οι τρέχουσα λειτουργία του προγράμματος δηλαδή για παράδειγμα αν ο παίκτης πάει σε μια πόρτα να του εμφανίσει τον γρίφο.

```
void LockDoor() // void ClearDoor()
```

Τέλος θα μιλήσουμε για αυτές τις 2 πολύ μικρές αλλά χρήσιμες συναρτήσεις, οι οποίες αντιστοίχως είτε κλειδώνουν μια πόρτα ή την ανοίγουν οριστικά. και δεν έχουν παραμέτρους γιατί κάθε φορά που ο χρήστης έχει πρόσβαση σε μια πόρτα οι συντεταγμένες της αποθηκεύονται προσωρινά μέσα σε μια μεταβλητή, με την οποία μετά μπορούμε οι πόρτες να κλειδώσουν ή να ανοίξουν αυτόματα.

5.1.4. Συναρτήσεις στο αρχείο questions.h

Όπως έχει προαναφερθεί οι ερωτήσεις αποθηκεύονται σε μια συστοιχία τύπου δομής ***struct question***, συμπεριλαμβάνοντας μέσα όλες τις απαραίτητες πληροφορίες που χρειάζεται μια εικονική πόρτα

```
void DrawQuestions()
```

Αυτή η συνάρτηση απεικονίζει στην οθόνη μια ερώτηση ,δέχεται είσοδο από το χρήστη και ελέγχει την ορθή απάντηση των ερωτήσεων. Σε περίπτωση που απαντηθούν σωστά οι ερωτήσεις ανοίγει την πόρτα και αυξάνει κατά ένα τη μεταβλητή των ερωτήσεων η οποία καθορίζει ποια ερώτηση θα εμφανιστεί στο χρήστη. Σε περίπτωση εσφαλμένης απάντησης αυξάνει κατά ένα τον αριθμό δοκιμών στην συγκεκριμένη ερώτηση αν γίνουν 3 εσφαλμένες προσπάθειες κλειδώνει πόρτα και δεν μπορεί να ανοίξει για κανένα λόγο.

```
void DrawFinalQuestion(Texture2D codesnip)
```

Αυτή είναι και η τελευταία μεγάλη συνάρτηση που έχει γράφει, ο σκοπός της είναι να εμφανίζει στον παίκτη τον τελευταίο γρίφο, ο οποίος απαιτεί τη χρήση πληκτρολογίου, και ελέγχει την απάντηση ώστε να καθορίσει αν ο παίκτης έχει χάσει ή εάν έχει κερδίσει. Επίσης σε περίπτωση που ο παίκτης έχει απαντήσει σωστά σε όλες τις ερωτήσεις ακόμα και σε αυτές που δεν ήταν στο δρόμο του και δεν έχει κάνει ούτε ένα λάθος τον κατευθύνει στο λεγόμενο secret ending.Παραμέτρους έχει μόνο την εικόνα που χρειάζεται για τον τελευταίο γρίφο.

```
bool IsAnyKeyPressed()
```

Η τελευταία συνάρτηση η οποία είναι κομμάτι της προηγούμενης ελέγχει αν ο χρήστης έχει εισάγει σωστό χαρακτήρα στο πεδίο εισαγωγής ώστε να μην ανιχνεύονται λάθος πατήματα.

5.1.5. Δομή της main.c

```
InitWindow(screenWidth, screenHeight, "Structured Programming Simulator");
```

Αυτή η συνάρτηση αρχικοποιεί το παράθυρο εκτέλεσης του προγράμματος.

```
while (!WindowShouldClose() && !exitWindow)
```

Βασικό ***loop*** προγράμματος.

```
switch (AppState)
```

Αρχικό ***switch*** ελέγχου προγράμματος.

```
BeginDrawing();
```

Αρχή τοποθέτησης γραφικών στην οθόνη μετά από το οποίο ακολουθεί ένα ακόμα switch το οποίο ανάλογα με το ***AppStates***, ορίζει τι γραφικά θα εμφανίζονται στην οθόνη.

```
EndDrawing();
```

Τερματισμός τοποθέτησης γραφικών στην οθόνη.

```
CloseWindow();
```

Κλείσιμο παραθύρου

5.2. ΠΗΓΕΣ

https://www.sanfoundry.com/c-interview-questions-answers/

https://www.raylib.com/ (Βασική Βιβλιοθήκη)

https://boscaceoil.net/ (Μουσική Σχεδίαση)

https://0x72.itch.io/pixeldudesmaker (Δημιουργία Μοντέλου Χαρακτήρα)

(Η εκμάθηση της βιβλιοθήκης ***raylib*** έγινε με χρήση των παραδειγμάτων ***www.raylib.com/examples.html*** σε κάποια κομμάτια του κώδικα ενδέχεται να υπάρχουν παρόμοιες δομές.)

5.3. ΆΔΕΙΑ ΔΙΚΑΙΩΜΑΤΩΝ ΓΙΑ ΑΝΟΙΚΤΟ ΕΛΕΥΘΕΡΟ ΚΩΔΙΚΑ