



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

**University of Western Macedonia**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**Department of Electrical and Computer Engineering**

**Ομάδα Game Development**

**Παιχνίδι τρίλιζας στην c**

**Tic tac toe in c**

**ΧΡΗΣΤΟΣ ΤΥΠΟΥ**

## Σκοπός του παιχνιδιού

Ο στόχος του παιχνιδιού είναι να δημιουργήσουμε ένα παιχνίδι τρίλιζας στο οποίο μπορείς να παίξεις ενάντια σε έναν υπολογιστή ή ενάντια με έναν φίλο σου. Για να νικήσεις θα πρέπει να συμπληρώσεις μια τριάδα κάθετα, οριζόντια και διαγώνια. Αν δεν συμβεί κάτι από τα παραπάνω δηλαδή δεν συμπληρώσει κανείς τριάδα τότε το παιχνίδι έληξε με ισοπαλία.

Ο στόχος μας είναι, τελικά, να κατανοήσουμε τα βασικά στοιχεία της γλώσσας προγραμματισμού C και να εισέλθουμε στον τομέα της ανάπτυξης παιχνιδιών, το οποίο αποτελεί και το κύριο αντικείμενο μελέτης της ομάδας μας.

## Ανάλυση κώδικα στην c

```
//δήλωση βιβλιοθηκών
#include <stdio.h>
#include <stdlib.h>

//δήλωση σταθερών παραμέτρων
#define N 3
#define K 3
#define P 24
#define L 2

//δήλωση υπογραφών συναρτήσεων
void paixnidil(int array[N][K], char* paixtis1, char* paixtis2);
void paixnidi2(int array[N][K], char* paixtis1, char* paixtis2);
void ypologisths_play(int array[N][K], char* ypologisths, int i, int antipalos);
void search(int array[N][K], int ypologisths, int pt[L][P]);
void paixtis_play(int array[N][K], char* paixtis, int i);
void print_board(int array[N][K],char* paixtis1, char* paixtis2);
void check_who_has_win(int array[N][K], int* paixtis1, int* paixtis2);

//βασική συνάρτηση
int main(){
    //δήλωση και αρχικοποίηση με μηδέν (0) βασικού πίνακα
    int array[N][K]={{0,0,0},{0,0,0},{0,0,0}};

    //δήλωση μεταβλητών
    int x, y, w, z;
```

```
int i, j;
char* paixtis1;
char* paixtis2;

//κάνουμε εκκαθάριση το Terminal από προηγούμενες εγγραφές
system("cls");

//εδώ ρωτάμε τον χρήστη αν θέλει να παίξει ενάντια με υπολογιστή ή ενάντια με κάποιον φίλο του
do{
    printf("Επίλεξε αν θέλεις να παίξεις ενάντια στον υπολογιστή πάτα το πλήκτρο 1 ή με κάποιον φίλο σου πάτα το
πλήκτρο 2\n");
    scanf("%d",&i);
}while(i!=2 && i!=1);

//εδώ ρωτάμε τον πρώτο παίχτη με τι σύμβολο θέλει να παίξει
do{
    printf("Με πιο συμβολο θέλει να παίξει ο παίχτης 1 αν θέλετε να παίξετε με το X σύμβολο πατήστε 1 ή αν θέλετε
να παίξετε με το σύμβολο O πατήστε 2\n");
    scanf("%d",&j);
}while(j!=1 && j!=2);

//ανάλογα το τι έχει επιλέξει ο χρήστης κάνουμε την ανάλογη ανάθεση στις παρακάτω μεταβλητές
if(j==1){
    paixtis1="X";
    paixtis2="O";
}else{
    paixtis2="X";
    paixtis1="O";
}
```

```
//εδώ εκτυπώνουμε τον πίνακα τον αρχικό πίνακα
print_board(array,paixtis1,paixtis2);

//εδώ ανάλογα τι έχει επιλέξει ο χρήστης δηλαδή αν θέλει να παίξει με φίλο του ή με τον υπολογιστή τότε εκκινούμε
εκείνο το παιχνίδι
if(i==2){
    //χρήστης ενάντια με φίλο του
    paixnidi1(array, paixtis1, paixtis2);
}else{
    //χρήστης ενάντια υπολογιστή
    paixnidi2(array, paixtis1, paixtis2);
}

return 0;
}

//συνάρτηση για χρήστη ενάντια με φίλο του
void paixnidi1(int array[N][K], char* paixtis1, char* paixtis2){
    int i=0,j=0,k=0;

    //δημιουργία ενός άπειρου βρόγχου
    while(1){
        //εκκίνηση της συνάρτησης για να παίξει ο πρώτος χρήστης
        paixtis_play(array,paixtis1,1);
        //ελέγχουμε αν ο πρώτος παίκτης έχει συμπλήρωση τριάδα
        check_who_has_win(array,&i,&j);
        //εδώ εκτυπώνουμε τον πίνακα με την νέα του κατάσταση
        print_board(array,paixtis1,paixtis2);
    }
}
```

```

//αν ο πρώτος χρήστης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει i=1 and j=0 και εκτυπώνει το ανάλογο
μήνυμα και σταματάει την εκτέλεση του παιχνιδιού
if(i==1 && j==0){
    printf("the one who won is the player with the symbol %s",paixtis1);
    exit(0);
}

//απλά αυξάνουμε έναν counter
k=k+1;
//αν ο παραπάνω counter γίνει ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι
είναι ισοπαλία
if(k==9){
    printf("the game is draw");
    exit(0);
}

//εκκίνηση της συνάρτησης για να παίξει το δεύτερος χρήστης
paixtis_play(array,paixtis2,2);
//ελέγχουμε αν ο δεύτερος χρήστης έχει συμπληρώσει τριάδα
check_who_has_win(array,&i,&j);
//εκτυπώνουμε τον πίνακα με την νέα του κατάσταση
print_board(array,paixtis1,paixtis2);
//αν ο δεύτερος παίχτης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει j=1 and i=0 και εκτυπώνει το ανάλογο
μήνυμα και σταματάει η εκτέλεση του παιχνιδιού
if(j==1 && i==0){
    printf("the one who won is the player with the symbol %s",paixtis2);
    exit(0);
}

```

```

//εδώ αυξάνουμε έναν counter
k=k+1;
//αν ο counter γίνει ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι είναι
ισοπαλία
if(k==9){
    printf("the game is draw");
    exit(0);
}
}

//συνάρτηση για χρήστη ενάντια με υπολογιστή
void paixnidi2(int array[N][K], char* paixtis1, char* paixtis2){
    int i=0,j=0,k=0;

    //δημιουργία άπειρου βρόγχου
    while(1){
        //αν το σύμβολο του παίχτη είναι χ τότε παίζει πρώτος αλλιώς παίζει πρώτος ο υπολογιστής
        if(paixtis1=="X"){
            //καλούμε την συνάρτηση του παίχτη
            paixtis_play(array,paixtis1,1);
            //εδώ ελέγχουμε αν ο πρώτος παίχτης έχει συμπληρώσει την τριάδα που αναζητάει το παιχνίδι τρίλιζα
            check_who_has_win(array,&i,&j);
            //εδώ εκτυπώνουμε τον καινούργιο πίνακα
            print_board(array,paixtis1,paixtis2);

            //αν ο πρώτος χρήστης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει i=1 and j=0 και εκτυπώνει το
            ανάλογο μήνυμα και σταματάει την εκτέλεση του παιχνιδιού
        }
    }
}

```

```

if(i==1 && j==0){
    printf("the one who won is the player with the symbol %s",paixtis1);
    exit(0);
}

//εδώ αυξάνουμε έναν counter
k=k+1;

//αν ο counter γίνεται ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι
είναι ισοπαλία
if(k==9){
    printf("the game is draw");
    exit(0);
}

//καλούμε την συνάρτηση του υπολογιστή
ypologists_play(array,paixtis2,2,1);
//εδώ ελέγχουμε αν ο υπολογιστής παίχτης έχει συμπληρώσει την τριάδα που αναζητάει το παιχνίδι τρίλιζα
check_who_has_win(array,&i,&j);
//εδώ εκτυπώνουμε τον καινούργιο πίνακα
print_board(array,paixtis1,paixtis2);

//αν ο υπολογιστής χρήστης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει i=0 and j=1 και εκτυπώνει το
ανάλογο μήνυμα και σταματάει την εκτέλεση του παιχνιδιού
if(j==1 && i==0){
    printf("the one who won is the player with the symbol %s",paixtis2);
    exit(0);
}

```

```
//εδώ αυξάνουμε έναν counter
k=k+1;

//αν ο counter γίνει ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι
είναι ισοπαλία
if(k==9){
    printf("the game is draw");
    exit(0);
}
else{
    //καλούμε την συνάρτηση του υπολογιστή
    ypologists_play(array,paixtis1,1,2);
    //εδώ ελέγχουμε αν ο υπολογιστής παίχτης έχει συμπληρώσει την τριάδα που αναζητάει το παιχνίδι τρίλιζα
    check_who_has_win(array,&i,&j);
    //εδώ εκτυπώνουμε τον καινούργιο πίνακα
    print_board(array,paixtis1,paixtis2);

    //αν ο υπολογιστής χρήστης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει i=1 and j=0 και εκτυπώνει το
    ανάλογο μήνυμα και σταματάει την εκτέλεση του παιχνιδιού
    if(i==1 && j==0){
        printf("the one who won is the player with the symbol %s",paixtis1);
        exit(0);
    }

    //εδώ αυξάνουμε έναν counter
    k=k+1;

    //αν ο counter γίνει ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι
    είναι ισοπαλία
```

```
if(k==9){
    printf("the game is draw");
    exit(0);
}

//καλούμε την συνάρτηση του παιχτη
paixtis_play(array,paixtis2,2);
//εδώ ελέγχουμε αν ο δεύτερος παιχτης έχει συμπληρώσει την τριάδα που αναζητάει το παιχνίδι τρίλιζα
check_who_has_win(array,&i,&j);

//εδώ εκτυπώνουμε τον καινούργιο πίνακα
print_board(array,paixtis1,paixtis2);

//αν ο δεύτερος χρήστης έχει συμπληρώσει τριάδα τότε η συνάρτηση επιστρέφει i=0 and j=1 και εκτυπώνει το
ανάλογο μήνυμα και σταματάει την εκτέλεση του παιχνιδιού
if(j==1 && i==0){
    printf("the one who won is the player with symbol %s",paixtis2);
    exit(0);
}

//εδώ αυξάνουμε έναν counter
k=k+1;

//αν ο counter γίνει ίσος με 9 και δεν έχει ενεργοποιηθεί η προηγούμενη εντολή ελέγχου τότε το παιχνίδι
είναι τσοπαλία
if(k==9){
    printf("the game is draw");
    exit(0);
}
```

```
        }
    }
}

//η συνάρτηση του παιχτη
void paixtis_play(int array[N][K], char* paixtis, int i){
    int x,y;
    int k;

    //εδώ ζητάμε να βάλει ο χρήστης έναν αριθμό από το διάστημα [1,9] και κάνουμε αντιστοιχία σε θέσεις του πίνακα αν η θέση που επέλεξε είναι πιασμένη τότε του ξανά ζητάμε να επιλέξει μια άλλη θέση
    do{
        printf("player %d input one number where you want to input your symbol %s\n", i, paixtis);
        scanf("%d",&k);
        if(k==1){
            x=0;
            y=0;
        }else if(k==2){
            x=1;
            y=0;
        }else if(k==3){
            x=2;
            y=0;
        }else if(k==4){
            x=0;
            y=1;
        }else if(k==5){
            x=1;
            y=1;
        }
    }while(k>5 || k<1);
}
```

```

}else if(k==6){
    x=2;
    y=1;
}else if(k==7){
    x=0;
    y=2;
}else if(k==8){
    x=1;
    y=2;
}else if(k==9){
    x=2;
    y=2;
}else{
    x=0;
    y=0;
}
}while(array[y][x]!=0 || k<1 || k>9);

//αν η θέση είναι έγκυρη τότε τοποθετούμε στον πίνακα το key του παίχτη
array[y][x]=i;
}

//η συνάρτηση του υπολογιστή
void ypologisths_play(int array[N][K], char* ypologisths, int u, int antipalos){
    int arr[L][P];
    int arr_antipalou[L][P];
    int j=0,x,y,k=0,i,o,h=0;

    //εδώ αρχικοποιούμε τους πίνακες με -1 και του αντιπάλου αλλά και του υπολογιστή

```

```

for (i=0; i<L; i++) {
    for(o=0; o<P; o++){
        arr[i][o]=-1;
        arr_antipalou[i][o]=-1;
    }
}

//σε αυτήν την συνάρτηση ψάχνουμε αν ο υπολογιστής είναι ένα βήμα πριν κάνει τρίλιζα και μας επιστρέφει τις πιθανές
θέσεις
search(array,u,arr);

//εδώ βλέπουμε αν η προηγούμενη συνάρτηση έχει βρει καμία θέση όπου μπορεί να συμπληρώσει τρίλιζα
for(i=0; i<L; i++){
    for(o=0; o<P; o++){
        if(arr[i][o]==-1){
            //αυξάνουμε έναν counter
            j++;
        }
    }
}

//αν ο παραπάνω counter γίνει ταυτόσημος με P τότε εκτελούμε την παρακάτω συνθήκη
if(j==P){
    //ελέγχουμε αν ο αντίπαλος πάει να κάνει τρίλιζα
    search(array,antipalos,arr_antipalou);

    //ελέγχουμε αν ο αντίπαλος είναι σε θέση να κάνει τρίλιζα
    for(i=0; i<P; i++){
        if(array[arr_antipalou[0][i]][arr_antipalou[1][i]]==0 && arr_antipalou[0][i]>-1 && arr_antipalou[1][i]>-1){
    }
}

```

```
        array[arr_antipalou[0][i]][arr_antipalou[1][i]]=u;
        //για να σταματήσουμε την δομή επανάληψης αν βρεθεί μια θέση μη κενή
        i=P;
    }else{
        //κάνουμε αύξηση έναν counter
        h++;
    }
}

//αν ο counter γίνει ίσος με P τότε λειτουργεί η παρακάτω συνθήκη
if(h==P){
    //εδώ προσπαθούμε μέσω έτοιμων θέσεων να κάνουμε έξυπνο τον υπολογιστή και αν είναι κενή η θέση κάνουμε και
    ανάθεση της θέσης στον υπολογιστή ανάλογα με το key του υπολογιστή
    if(array[1][1]==0){
        array[1][1]=u;
    }else if(array[0][0]==0){
        array[0][0]=u;
    }else if(array[2][0]==0){
        array[2][0]=u;
    }else if(array[2][2]==0){
        array[2][2]=u;
    }else if(array[0][2]==0){
        array[0][2]=u;
    }else if(array[1][0]==0){
        array[1][0]=u;
    }else if(array[0][1]==0){
        array[0][1]=u;
    }else if(array[1][2]==0){
        array[1][2]=u;
    }
}
```

```

    }else{
        array[2][1]=u;
    }
}
}else{
    //εδώ αν ο υπολογιστής πάει να κάνει τρίλιζα τότε παίρνουμε τις θέσεις αυτές και ελέγχουμε αν είναι κενές και στην
    πρώτη κενή θέση κάνουμε ανάθεση του key του υπολογιστή
    for(i=0; i<P; i++){
        if(array[arr[0][i]][arr[1][i]]==0){
            array[arr[0][i]][arr[1][i]]=u;
            //stop loop εάν βρούμε μια κενή θέση
            i=P;
        }else{
            //αυξάνουμε έναν counter
            k++;
        }
    }

    //αν ο παραπάνω counter γίνει
    if(k==P){
        h=0;

        //ελέγχουμε αν ο αντίπαλος πάει να κάνει τρίλιζα
        search(array,antipalos,arr_antipalou);

        //ελέγχουμε αν ο αντίπαλος είναι σε θέση να κάνει τρίλιζα
        for(i=0; i<P; i++){
            if(array[arr_antipalou[0][i]][arr_antipalou[1][i]]==0 && arr_antipalou[0][i]>-1 && arr_antipalou[1][i]>-1){

```

```
        array[arr_antipalou[0][i]][arr_antipalou[1][i]]=u;
        //για να σταματήσουμε την δομή επανάληψης αν βρεθεί μια θέση μη κενή
        i=P;
    }else{
        //αυξάνουμε έναν counter
        h++;
    }
}

//αν ο παραπάνω counter γίνει
if(h==P){
    //εδώ προσπαθούμε μέσω έτοιμων θέσεων να κάνουμε έξυπνο τον υπολογιστή και αν είναι κενή η θέση κάνουμε
    και ανάθεση της θέσης στον υπολογιστή ανάλογα με το key του υπολογιστή
    if(array[1][1]==0){
        array[1][1]=u;
    }else if(array[0][0]==0){
        array[0][0]=u;
    }else if(array[2][0]==0){
        array[2][0]=u;
    }else if(array[2][2]==0){
        array[2][2]=u;
    }else if(array[0][2]==0){
        array[0][2]=u;
    }else if(array[1][0]==0){
        array[1][0]=u;
    }else if(array[0][1]==0){
        array[0][1]=u;
    }else if(array[1][2]==0){
        array[1][2]=u;
```

```
        }else{
            array[2][1]=u;
        }
    }
}

//η συνάρτηση για εύρεση αν ο υπολογιστής ή ο αντίπαλος είναι σε θέση να κάνει τρίλιζα κάποιος
void search(int array[N][K], int ypologisths, int pt[L][P]){
    int i=0;

    //σε όλες τις συνθήκες ελέγχου ελέγχουμε κατά ζεύγη αν υπάρχει πιθανότητα τρίλιζας και επιστρέφουμε αυτές τις
    πιθανές τιμές
    if(array[0][0]==array[1][1] && array[0][0]==ypologisths){
        pt[1][i]=2;
        pt[0][i]=2;
        i++;
    }

    if(array[0][0]==array[2][2] && array[0][0]==ypologisths){
        pt[1][i]=1;
        pt[0][i]=1;
        i++;
    }

    if(array[1][1]==array[2][2] && array[2][2]==ypologisths){
        pt[1][i]=0;
    }
}
```

```
pt[0][i]=0;
i++;
}

if(array[0][2]==array[1][1] && array[1][1]==ypologisths){
    pt[1][i]=0;
    pt[0][i]=2;
    i++;
}

if(array[1][1]==array[2][0] && array[1][1]==ypologisths){
    pt[1][i]=2;
    pt[0][i]=0;
    i++;
}

if(array[0][2]==array[2][0] && array[0][2]==ypologisths){
    pt[1][i]=1;
    pt[0][i]=1;
    i++;
}

for(int j=0; j<N; j++){
    if(array[j][0]==array[j][1] && array[j][0]==ypologisths){
        pt[1][i]=2;
        pt[0][i]=j;
        i++;
    }
}
```

```
if(array[j][1]==array[j][2] && array[j][1]==ypologisths){
    pt[1][i]=0;
    pt[0][i]=j;
    i++;
}

if(array[j][0]==array[j][2] && array[j][0]==ypologisths){
    pt[1][i]=1;
    pt[0][i]=j;
    i++;
}
}

for(int j=0; j<N; j++){
    if(array[0][j]==array[1][j] && array[0][j]==ypologisths){
        pt[1][i]=j;
        pt[0][i]=2;
        i++;
    }

    if(array[1][j]==array[2][j] && array[1][j]==ypologisths){
        pt[1][i]=j;
        pt[0][i]=0;
        i++;
    }

    if(array[0][j]==array[2][j] && array[0][j]==ypologisths){
        pt[1][i]=j;
        pt[0][i]=1;
    }
}
```

```
        i++;
    }
}
}

//εδώ εκτυπώνουμε τον πίνακα ανάλογα τα key των παιχτών για να εκτυπώσουμε τα καταλληλά σύμβολα
void print_board(int array[N][K],char* paixtis1, char* paixtis2){
    int i, j;

    system("cls");

    printf("\n");

    for(i=0; i<3; i++){
        for(j=0; j<3; j++){
            if(array[i][j]==1){
                printf(paixtis1);
            }else if(array[i][j]==2){
                printf(paixtis2);
            }else{
                //για όποιες θέσεις είναι κενές εκτυπώνουμε ποιες θέσεις είναι ελεύθερες
                if(i==0 && j==0){
                    printf("□");
                }else if(i==0 && j==1){
                    printf("□");
                }else if(i==0 && j==2){
                    printf("□");
                }else if(i==1 && j==0){
                    printf("4□");
                }
```

```

        }else if(i==1 && j==1){
            printf("█");
        }else if(i==1 && j==2){
            printf("█");
        }else if(i==2 && j==0){
            printf("█");
        }else if(i==2 && j==1){
            printf("█");
        }else{
            printf("█");
        }
        if(j<2){
            printf(" | ");
        }
    }
    printf("\n-----\n");
}

//εδώ ελέγχουμε αν συμπλήρωσε κανείς τρίλιζα και ανάλογα τον παίχτη επιστρέφουμε τον κατάλληλο συνδυασμό από της
μεταβλητές i and j
void check_who_has_win(int array[N][K], int *paixtis1, int *paixtis2){
    //διαγώνια
    if(array[0][0]==array[1][1] && array[0][0]==array[2][2]){
        if(array[0][0]==1{
            *paixtis1=1;
            *paixtis2=0;
        }
    }
}

```

```
}

if(array[0][0]==2){
    *paixtis1=0;
    *paixtis2=1;
}
}

if(array[0][2]==array[1][1] && array[0][2]==array[2][0]){
    if(array[0][2]==1){
        *paixtis1=1;
        *paixtis2=0;
    }

    if(array[0][2]==2){
        *paixtis1=0;
        *paixtis2=1;
    }
}

//οριζόντια
for(int i=0; i<N; i++){
    if(array[i][0]==array[i][1] && array[i][0]==array[i][2]){
        if(array[i][i]==1){
            *paixtis1=1;
            *paixtis2=0;
        }

        if(array[i][i]==2){
```

```
        *paixtis1=0;
        *paixtis2=1;
    }
}

//κάθετα
if(array[0][0]==array[1][0] && array[0][0]==array[2][0]){
    if(array[0][0]==1){
        *paixtis1=1;
        *paixtis2=0;
    }

    if(array[0][0]==2){
        *paixtis1=0;
        *paixtis2=1;
    }
}

if(array[0][1]==array[1][1] && array[0][1]==array[2][1]){
    if(array[0][1]==1){
        *paixtis1=1;
        *paixtis2=0;
    }

    if(array[0][1]==2){
        *paixtis1=0;
        *paixtis2=1;
    }
}
```

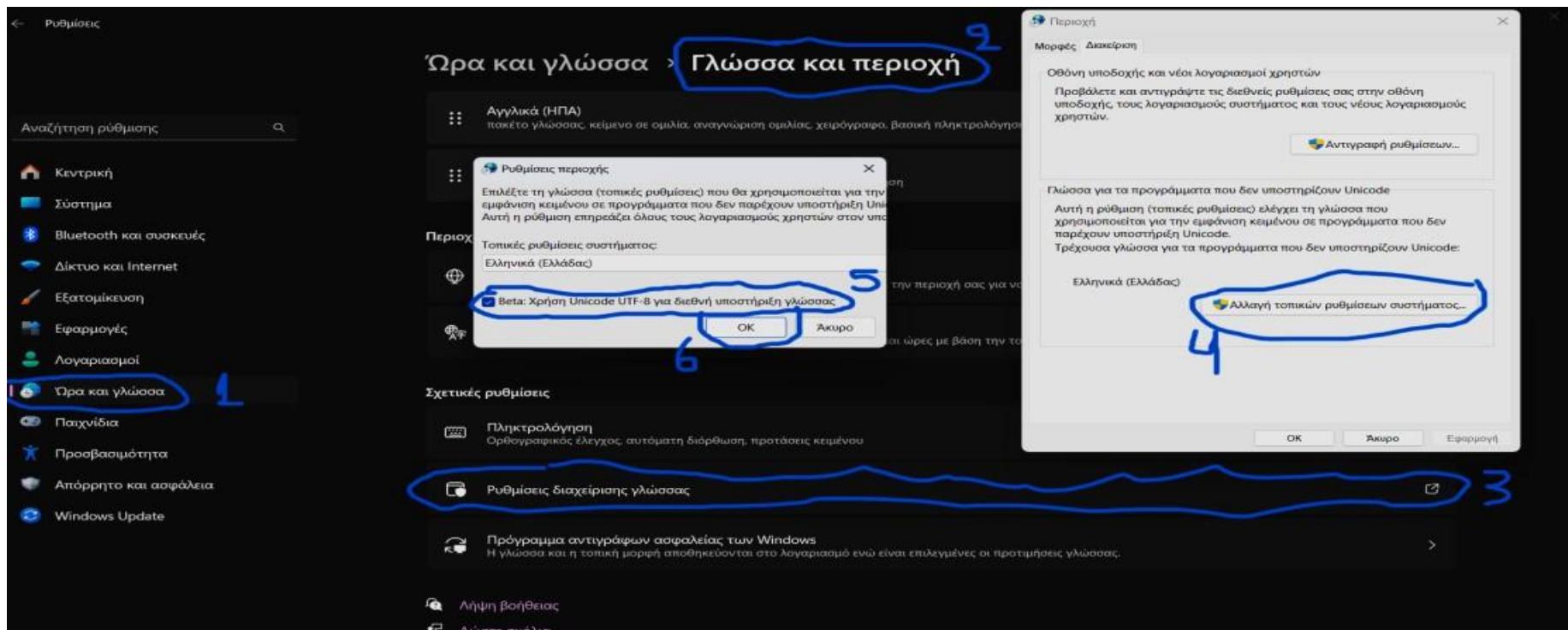
```
}

if(array[0][2]==array[1][2] && array[0][2]==array[2][2]){
    if(array[0][2]==1){
        *paixtis1=1;
        *paixtis2=0;
    }

    if(array[0][2]==2){
        *paixtis1=0;
        *paixtis2=1;
    }
}
```

## Τρόπος εκτύπωσης emoji στο Terminal

Θα πρέπει να ακολουθήσετε τις παρακάτω οδηγίες:



Μόλις κάνετε όλα τα παραπάνω θα σας ζητήσει να κάνετε επανεκκίνηση τον υπολογιστή για να εφαρμοστούν οι αλλαγές (θα πρέπει να την κάνετε)

## Βιβλιογραφία - Πηγές

1. Το μάθημα δομημένος προγραμματισμός

<https://eclass.uowm.gr/courses/ICTE258/>

2. Για το παραπάνω παιχνίδι χρησιμοποιήσαμε το πρόγραμμα visual studio code και για γλώσσα προγραμματισμού την c για να κατεβάσετε το visual studio code μπορείτε να το βρείτε δωρεάν από το Microsoft store και για να ενεργοποιήσετε την γλώσσα και τον compiler θα πρέπει να

ακολουθήσετε το παρακάτω YouTube video:

<https://www.youtube.com/watch?v=Ubfgi4NoTP>