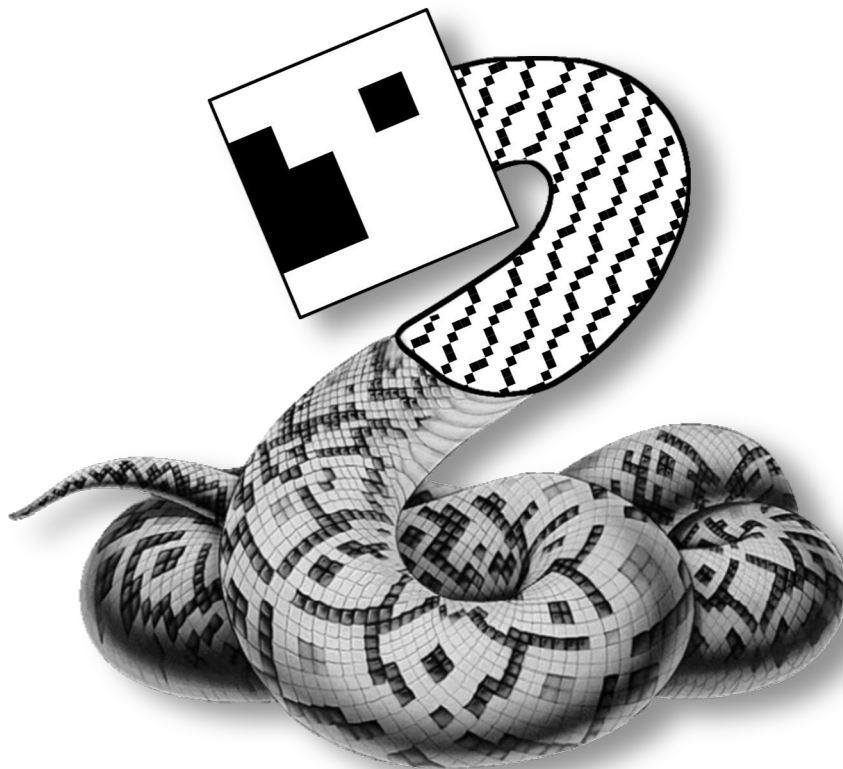


ANNEXOS DE CODI DEL TREBALL:

"Construcció d'una videoconsola i
programació dels 25 "Snakes" que conté"



2n Batxillerat C
Curs acadèmic: 2017/2018

ÍNDEX

main.c	pàg 1
funcions/setup.h	pàg 6
sprites/imatgeC.c	pàg 10
sprites/LletresNumeros.c	pàg 11
sprites/HUDsEspecials.c	pàg 16
sprites/Serps.c	pàg 19
sprites/Rates.c	pàg 23
funcions/funcions_motor.h	pàg 24
funcions/funcions_menu.h	pàg 28
funcions/funcions_joc.h	pàg 32
funcions/funcions_setup.h	pàg 53
modes/modes_control.h	pàg 55
modes/modes_limit.h	pàg 56
modes/modes_serp.h	pàg 57
modes/modes_rata.h	pàg 59
modes/modes_grafic.h	pàg 60
modes/modes_objectiu.h	pàg 61
modes/modes_altres.h	pàg 63

main.c

```
//#include <stdio.h>
//#include <stdlib.h>
//#include <string.h>
#include "samd21_platform.c"

#include "funcions/aleatorietat.h"

#include "funcions/setup.h"

#include "sprites/imatgeC.c"
#include "sprites/LletresNumeros.c"
#include "sprites/HUDsEspecials.c"
#include "sprites/Serps.c"
#include "sprites/Rates.c"

#include "funcions/funcions_motor.h"
#include "funcions/funcions_menu.h"
#include "funcions/funcions_joc.h"

#include "funcions/funcions_setup.h"

int mode_00_CLASSIC(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(0) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

#include "modes/modes_control.h"
#include "modes/modes_limits.h"
#include "modes/modes_serp.h"
#include "modes/modes_rata.h"
#include "modes/modes_grafic.h"
#include "modes/modes_objectiu.h"
#include "modes/modes_altres.h"

int main(){
    inicia_generador_aleatoris();

    int ample_finestra = 128;    int alcada_finestra = 64;
    bp = alloc_bitmap_memory(ample_finestra, alcada_finestra);
    graphics_init(ample_finestra, alcada_finestra, titol_finestra, factor_zoom);

    /*U32 i = 0;
    bool pin_state = true;
    while(i < 256){
        //port_pin_set_output_level(LED_0_PIN, pin_state);
        //pin_state = !pin_state;
        //delay_ms(1000/64.f);
        //
        if(port_pin_get_input_level(PIN_PA02)){
            port_pin_set_output_level(LED_0_PIN, HIGH);
        }
        else{
            port_pin_set_output_level(LED_0_PIN, LOW);
        }

        graphics_render(bp);

        bp.pixels[i] = 1;
        i = (i+1)%(128*64);

        ControllerState controller_state = get_controller_state();
    }*/
```

```

inicialitzar_finestres();
for (int i=0; i<240; i++){p.m6.posicionsPINTAR[i] = 'n';}

inicialitzar_TOTS_menus();
Menu menu_a_mostrar = menu_inici;
int op = 0;
int op_pri = 0; //principal
int op_esp = 0; //especifica

printf("\nCOMENÇO\n\n");
while(!user_requested_exit()){
    int he_triat = 0;
    int falsa_alarma = 0;
    int mode = -1;
    while(he_triat == 0){
        /* //NOTE: visualize refresh rate using LED
        {
            port_pin_set_output_level(LED_0_PIN, pin_state);
            pin_state = !pin_state;
            //delay_ms(1000/64.f);
        } */

        controller = get_controller_state();
        op = pintar_menus(&menu_a_mostrar, op);
        if (controller.A.pressed && controller.A.half_transition_count){
            falsa_alarma = 0;
            if(menu_a_mostrar.titol == menu_inici.titol){ //Inici
                if (op == 0){mode = 0;}
                if(op==1){
                    menu_a_mostrar = menu_principal;
                    op = 0;
                    falsa_alarma = 1;
                }
            }
            else if(menu_a_mostrar.titol == menu_principal.titol){
                op_pri = op;
                if(op==0){menu_a_mostrar = menu_control;}
                else if(op==1){menu_a_mostrar = menu_limits;}
                else if(op==2){menu_a_mostrar = menu_serp;}
                else if(op==3){menu_a_mostrar = menu_rata;}
                else if(op==4){menu_a_mostrar = menu_grafic;}
                else if(op==5){menu_a_mostrar = menu_objectiu;}
                else if(op==6){menu_a_mostrar = menu_altres;}
                op = 0;
                falsa_alarma = 1;
            }
            else if(menu_a_mostrar.titol == menu_control.titol){ //Control
                if (op == 0){mode = 1;}
                else if (op == 1){mode = 2;}
                else if (op == 2){mode = 3;}
            }
            else if(menu_a_mostrar.titol == menu_limits.titol){ //Limits
                if (op == 0){mode = 4;}
                else if (op == 1){mode = 5;}
                else if (op == 2){menu_a_mostrar = menu_editor; op = 0; falsa_alarma = 1;}
            }
            else if(menu_a_mostrar.titol == menu_editor.titol){ //Menu especial: Editor
                if (op == 0){mode = 6;}
                else if (op == 1){mode = 66;}
            }
            else if(menu_a_mostrar.titol == menu_serp.titol){ //Serp
                if (op == 0){mode = 7;}
                else if (op == 1){mode = 8;}
                else if (op == 2){mode = 9;}
                else if (op == 3){mode = 10;}
                else if (op == 4){mode = 11;}
                else if (op == 5){mode = 12;}
                else if (op == 6){mode = 13;}
            }
            else if(menu_a_mostrar.titol == menu_rata.titol){ //Rata
                if (op == 0){mode = 14;}
                else if (op == 1){mode = 15;}
                else if (op == 2){mode = 16;}
            }
        }
    }
}

```

```

else if(menu_a_mostrar.titol == menu_grafic.titol){ //Gràfic
    if (op == 0){mode = 17;}
    else if (op == 1){mode = 18;}
}
else if(menu_a_mostrar.titol == menu_objectiu.titol){ //Objectiu
    if (op == 0){mode = 19;}
    else if (op == 1){mode = 20;}
    else if (op == 2){mode = 21;}
    else if (op == 3){mode = 22;}
    else if (op == 4){mode = 23;}
    else if (op == 5){mode = 24;}
}
else if(menu_a_mostrar.titol == menu_altres.titol){ //Altres
    if (op == 0){mode = 25;}
    else if (op == 1){falsa_alarma = 1;}
}
if(falsa_alarma != 1){he_triat = 1; op_esp = op;}
}
else if(controller.B.pressed && controller.B.half_transition_count){
    if (menu_a_mostrar.titol == menu_inici.titol){}
    else if(menu_a_mostrar.titol == menu_principal.titol){menu_a_mostrar = menu_inici;
op = 1;}

    else if(menu_a_mostrar.titol == menu_editor.titol){menu_a_mostrar = menu_limits; op
= 2;}

    else{menu_a_mostrar = menu_principal; op = op_pri;}
}
graphics_render(bp);
++numero_de_fotograma;
if(user_requested_exit()){break;} // ANTIGA GUARRADA SUPREMA
}

int again = -1;
while (again != 0){
    iniciar_modes_i_serp(mode);
    segons = time(NULL);
    int fi = 0;
    while(fi == 0)
    {
        controller = get_controller_state();
        if (controller.start.half_transition_count && controller.start.pressed){
            if (p.m0.pausat == 0){p.m0.pausat = 1; indexA = 0; indexB = -2;}
            else{p.m0.pausat = 0; segons += t_pausa;}
        }
        if(p.m0.pausat == 0){
            if(mode == 0){ fi = mode_00_CLASSIC();}
            else if(mode == 1){fi = mode_01_INVERTIT();}
            else if(mode == 2){fi = mode_02_EGO();}
            else if(mode == 3){fi = mode_03 ESTRATEGIC();}
            else if(mode == 4){fi = mode_04_VORES();}
            else if(mode == 5){fi = mode_05_FANTASMA();}
            else if(mode == 6){ mode_06_PINTAR();}
            else if(mode == 66){fi= mode_66_BARRERES();}
            else if(mode == 7){fi = mode_07_REGULABLE();}
            else if(mode == 8){fi = mode_08_MANUAL();}
            else if(mode == 9){fi = mode_09_INCRE();}
            else if(mode == 10){fi = mode_10_CAPICUA();}
            else if(mode == 11){fi = mode_11_REMOT();}
            else if(mode == 12){fi = mode_12_DIAGONAL();}
            else if(mode == 13){fi = mode_13_SMOOTH();}
            else if(mode == 14){fi = mode_14_PLUS();}
            else if(mode == 15){fi = mode_15_DIRECCIONALS();}
            else if(mode == 16){fi = mode_16_RATESMOBILS();}
            else if(mode == 17){fi = mode_17_SURVIVAL();}
            else if(mode == 18){fi = mode_18_GALLINETA();}
            else if(mode == 19){fi = mode_19_EASY();}
            else if(mode == 20){fi = mode_20_SelNat();}
            else if(mode == 21){fi = mode_21_BASQUET();}
            else if(mode == 22){fi = mode_22_LOOPS();}
            else if(mode == 23){fi = mode_23_1D();}
            else if(mode == 24){fi = mode_24_IA();}
            else if(mode == 25){fi = mode_25_RETRO();}
            else{fi = 7;}
        }
    }
    else{

```

```

        t_pausa = time(NULL) - segons - t;
        pintarhoTOT(mode);
        for(int i = 0; i<5; i++){
            if (i==indexA){ps(&finestra_joc, lletresPAUSA+i, i2(posicionsPAUSA[i].x,
posicionsPAUSA[i].y+2));}
            else if (i==indexB){ps(&finestra_joc, lletresPAUSA+i,
i2(posicionsPAUSA[i].x, posicionsPAUSA[i].y-2));}
            else{ps(&finestra_joc, lletresPAUSA+i, posicionsPAUSA[i]);}
        }
        if(numero_de_fotograma % 10 == 0){
            if (indexB < 5){indexA+=1; indexB = indexA-2;}
            else{indexA=0; indexB = -2;}
        }
        ++numero_de_fotograma;
        if(controller.B.pressed){fi = 1;}
    }
    if(user_requested_exit()){break;} // ANTIGA GUARRADA SUPREMA
    graphics_render(bp);
}

if(fi == 1){printf("M'he pirat de la pausa");}
//else if(fi == 2){printf("He clicat la creueta");}
else if(fi == 3){printf("Màxima puntuació!(kindof)");}
else if(fi == 4){printf("Estic fora del ring");}
else if(fi == 5){printf("He xocat amb mi mateix");}
//else if(fi == 6){printf("El temps s'ha acabat");}
else if(fi == 7){printf("No me'l sé aquest mode");}
else if(fi == 8){printf("Mort especifica del mode");}
else{printf("Que parezca un accidente");}

printf(" i per tant\n\nACABO\n");
if (mode == 6){
    printf("Has modificat %i caselles i n'has deixat %i sense
tocar\n",p.m6.parets+p.m6.NORATAs, 240-p.m6.parets-p.m6.NORATAs);
    printf("TEMPS: %li segons\n",t);
    printf("mod/reg: %f percent\n", ((p.m6.parets+p.m6.NORATAs)/(float)240)*100);
    printf("FOTO: %i fotogrames\n\n",numero_de_fotograma);
}
else if (mode == 19){
    printf("PUNTUACIÓ: T'han faltat %i caselles -> tens %i punts de 1200
possibles\n",240-s.length, (s.length)*5);
    printf("TEMPS: %li segons\n",t);
    printf("full/total: %f percent\n", (s.length/(float)240)*100);
    printf("FOTO: %i fotogrames\n\n",numero_de_fotograma);
}
else{
    printf("PUNTUACIÓ: %i rates -> %i punts\n",p.m0.puntuacio, p.m0.puntuacio*5);
    printf("TEMPS: %li segons\n",t);
    printf("PUNTUACIO/TEMPS: %f percent\n", (p.m0.puntuacio/(float)t)*100);
    printf("FOTO: %i fotogrames\n\n",numero_de_fotograma);
}

if(mode == 66 && p.m6.acabemdecomencar==2){ //GUARRADA NECESSARIA PEL FLAIX 66
    ferforma(&finestra_general,i2(0,0),128,64,blanc);
    for(int i = 0; i<10; i++){
        ++numero_de_fotograma;
        graphics_render(bp);
    }
}

int num;
    if(mode == 9 && 10000/p.m9.increment<5){num = 10000/p.m9.increment;}
    else if(mode == 13){num = 1;}
    else if(mode == 19){num = 2;}
    else if(mode == 25){num = 3;}
    else if(mode == 23){num = 4;}
    else{num = 5;}
    fi = 0;
    while(fi == 0)
    {
        if(numero_de_fotograma % num == 0){
            pintarhoTOT(mode);
            if(mode == 66 &&
p.m6.acabemdecomencar==2){ferforma(&finestra_general,i2(0,0),128,64,blanc);}

```

```

        else if(mode == 17){pintarhoTOT(177);}
        fi = moureSERP_FINAL();
    }
    if (mode == 18){pintarhoTOT(mode);}
    ++numero_de_fotograma;
    graphics_render(bp);
    controller = get_controller_state();
    if((controller.A.pressed && controller.A.half_transition_count)||
        (controller.B.pressed && controller.B.half_transition_count)||
        (controller.start.pressed && controller.start.half_transition_count)){fi = 1;}
    if(user_requested_exit()){break;} // ANTIGA GUARRADA SUPREMA
}

Menu menu_resultats = {
    .titol = "results",
    .opcions = 5,
};
char* noms_resultats[5];
int xifres_resultats[3];
if(mode == 6){
    noms_resultats[0]="mod:"; noms_resultats[1]="reg:"; noms_resultats[2]="div:";
    noms_resultats[3]="play"; noms_resultats[4]="exit";
    xifres_resultats[0]=p.m6.parets+p.m6.NORATAs;
xifres_resultats[1]=240-xifres_resultats[0];
    xifres_resultats[2]=(xifres_resultats[0]/(float)240)*100;
}
else if(mode == 66 && p.m6.acabemdecomencar==2){
    noms_resultats[0]="invalid"; noms_resultats[1]="starting";
noms_resultats[2]="position";
    noms_resultats[3]="edit"; noms_resultats[4]="exit";
}
else if(mode == 19){
    noms_resultats[0]="full:"; noms_resultats[1]="empty:"; noms_resultats[2]="div:";
    noms_resultats[3]="again"; noms_resultats[4]="exit";
    xifres_resultats[0]=s.length; xifres_resultats[1]=p.m0.puntuacio;
xifres_resultats[2]=(s.length/(float)240)*100;
}
else{
    noms_resultats[0]="rats:"; noms_resultats[1]="time:"; noms_resultats[2]="div:";
    noms_resultats[3]="again"; noms_resultats[4]="exit";
    xifres_resultats[0]=p.m0.puntuacio; xifres_resultats[1]=(int)t;
    if(t != 0){xifres_resultats[2]=(p.m0.puntuacio/(float)t)*100;}
else{xifres_resultats[2]=0;}
}

inicialitzar_menu(&menu_resultats, noms_resultats, xifres_resultats, 'x');
op = 0;
he_triat = 0;
while(he_triat == 0){
    controller = get_controller_state();
    if(mode != 66 || p.m6.acabemdecomencar!=2){op = pintar_menu_XIFRES(&menu_resultats,
op, 'x');}

    else{op = pintar_menu_XIFRES(&menu_resultats, op, 'o');}

    if (controller.A.pressed && controller.A.half_transition_count){
        if(op==0){again = 1; he_triat = 1;
            if(mode == 6){mode = 66; op_esp = 1;}
            else if(mode == 66 && p.m6.acabemdecomencar==2){mode = 6; op_esp = 0;}
        }
        else if(op==1){again = 0; he_triat = 1;}}
    graphics_render(bp);
    ++numero_de_fotograma;
    if(user_requested_exit()){break;} // ANTIGA GUARRADA SUPREMA
}
op = op_esp;

    if(user_requested_exit()){break;}
}
}

graphics_end();
return 0;
}

```

funcions/setup.h

```
long int t;
long int t_pausa;
char *titol_finestra = "Snakest";
int factor_zoom = 6;
time_t segons = 0;

Bitmap bp = {};
Pixel negre = {0}; Pixel blanc = {1};
//Pixel vermell = {255, 0, 0}; Pixel blau = {0, 0, 255}; Pixel croma = {0,255,0};
int numero_de_fotograma = 0;

ControllerState controller;

typedef struct I2{
    int8_t x;
    int8_t y;
    //Byte x;
    //Byte y;
}I2;

static I2 i2(int x, int y){
    I2 v = {x, y};
    return v;
}

I2 posicionsXY(int index){
    int i = 0;
    for (int x = 0; x<20; x++){
        for (int y = 0; y<12; y++){
            if(index == i){return i2(14 + x*5, 2 + y*5);}
            i++;
        }
    }
    return i2(109,57);
}
//int posicionsXY_invers(I2 index){return ((index.x-14)/5)*12+((index.y-2)/5);}

typedef struct Serp{
    I2 posicions[240*5];
    int length;
    int creixer;
    char dir;
    char dirR;
    char dircua;
    I2 cap;
    int lloc;} Serp;
Serp s;

typedef struct Mode0{
    int HS0;
    int puntuacio;
    I2 R;
    int pausat;}Mode0;

static I2 taula_general[240];

typedef struct Model{
    int HS1;}Model;

typedef struct Mode2{
    int HS2;
    int puc_girar;
    int index_dir;
    char direccions[5];}Mode2;

typedef struct Mode3{
    int HS3;
    I2 mira;
    char dir_mira;
    I2 ordre;
    int distanciaX;
    int distanciaY;
```



```

    int ordre_donada;
    char direccio_a_moure_serp;}Mode3;

typedef struct Mode4{
    int HS4;}Mode4;

typedef struct Mode5{
    int HS5;
    int NUMposicionsvisibles;
    I2 *posicionsvisibles;
}Mode5;

typedef struct Mode6{
    int HS6;
    int acabemdecomencar;
    char colorBN;
    char posicionsPINTAR[240];
    int parets;
    int NORATAs;
    int cossosreals;
    I2 origen;
    char dirOri;}Mode6;

typedef struct Mode7{
    int HS7;
    int acabemdecomencar;}Mode7;

typedef struct Mode8{
    int HS8;}Mode8;

typedef struct Mode9{
    int HS9;
    int increment;}Mode9;

typedef struct Mode10{
    int HS10;
    int mode;
    I2 *posicionsaux;
    char diraux;}Mode10;

typedef struct Mode11{
    int HS11;
    I2 P;
    int pitfora;
    char direcciopit;}Mode11;

typedef struct Mode12{
    int HS12;
    char dir1;
    char dir2;
    I2 *posicions12;
    int index12;
    I2 cap12[2];}Mode12;

typedef struct Mode13{
    int HS13;
    int NUMcasellesocupades;
    I2 *casellesocupades;
}Mode13;

typedef struct rosegador{
    I2 origen;
    int durada;
    int quanhaaparegut;
    int caracter;}rosegador;
typedef struct Mode14{
    int HS14;
    rosegador rates[5];}Mode14;

typedef struct Mode15{
    int HS15;
    char dirRATA;
    int ladirecciodelarataescorrecta;}Mode15;

```

```

typedef struct Model6{
    int HS16;}Model6;

typedef struct Model7{
    int HS17;
    int sumes[5];}Model7;

typedef struct Model8{
    int HS18;
    int difX;
    int difY;
    int distancia;
    int flash;}Model8;

typedef struct Model9{
    int HS19;}Model9;

typedef struct Mode20{
    int HS20;}Mode20;

typedef struct Mode21{
    int HS21;
    I2 forat;}Mode21;

typedef struct Mode22{
    int HS22;
    I2 *posicions22;
    int index22;}Mode22;

typedef struct Mode23{
    int HS23;}Mode23;

typedef struct Mode24{
    int HS24;
    char dirRATA;
    I2 prota;
    I2 capAUX;}Mode24;

typedef struct Mode25{
    int HS25;}Mode25;

typedef struct Partida{
    Mode0 m0; Model1 m1; Mode2 m2; Mode3 m3; Mode4 m4; Mode5 m5;
    Mode6 m6; Mode7 m7; Mode8 m8; Mode9 m9; Model10 m10;
    Model11 m11; Model12 m12; Model13 m13; Model14 m14; Model15 m15;
    Model16 m16; Model17 m17; Model18 m18; Model19 m19; Mode20 m20;
    Mode21 m21; Mode22 m22; Mode23 m23; Mode24 m24; Mode25 m25;
}Partida;

Partida p = {
    .m0 = {.HS0 = 0},
    .m1 = {.HS1 = 0},
    .m2 = {.HS2 = 0, .direccions = "NESO"},
    .m3 = {.HS3 = 0},
    .m4 = {.HS4 = 0},
    .m5 = {.HS5 = 0, .posicionsvisibles = taula_general},
    .m6 = {.HS6 = 0, .dirOri = 'E', .origen = {19,2}},
    .m7 = {.HS7 = 0},
    .m8 = {.HS8 = 0},
    .m9 = {.HS9 = 0},
    .m10 = {.HS10 = 0, .mode = 0, .posicionsaux = taula_general},
    .m11 = {.HS11 = 0},
    .m12 = {.HS12 = 0, .posicions12 = taula_general},
    .m13 = {.HS13 = 0, .casellesocupades = taula_general},
    .m14 = {.HS14 = 0},
    .m15 = {.HS15 = 0},
    .m16 = {.HS16 = 0},
    .m17 = {.HS17 = 0, .sumes = {-10,-5,0,5,10}},
    .m18 = {.HS18 = 0},
    .m19 = {.HS19 = 240},
    .m20 = {.HS20 = 0},
    .m21 = {.HS21 = 0},
    .m22 = {.HS22 = 0, .posicions22 = taula_general},
    .m23 = {.HS23 = 0},

```

```
.m24 = {.HS24 = 0},  
.m25 = {.HS25 = 0},  
};  
  
typedef struct Finestra{  
    I2 origen;  
    I2 mida;  
    Bitmap bitmap;  
}Finestra;  
  
Finestra finestra_general = {}; Finestra finestra_record = {};  
Finestra finestra_puntuacio = {}; Finestra finestra_joc = {};
```

sprites/imatgeC.c

```
typedef struct imatgeC {  
    unsigned int  width;  
    unsigned int  height;  
    unsigned char *pixel_data; // [width * height * bytes_per_pixel + 1]  
}imatgeC;
```

sprites/LletresNumeros.c

```
unsigned char lletra_a_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9";
imatgeC lletra_a = {
    8, 10, lletra_a_data
};

unsigned char lletra_b_data[] =
    "\x66\x6A\x99\x9A\x6A\xA6\x9A\xA9\x66\x66\x99\x9A\x6A\xA6\x9A\xA9\x66\x66\x99\x9A";
imatgeC lletra_b = {
    8, 10, lletra_b_data
};

unsigned char lletra_c_data[] =
    "\x66\x69\x99\x6A\xA9\xAA\x6A\xA9\xAA\x6A\xA9\xAA\x66\x69\x99";
imatgeC lletra_c = {
    6, 10, lletra_c_data
};

unsigned char lletra_d_data[] =
    "\x66\xAA\x99\xAA\x6A\x6A\x9A\x9A\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x9A";
imatgeC lletra_d = {
    8, 10, lletra_d_data
};

unsigned char lletra_e_data[] =
    "\x66\x69\x99\x6A\xA9\xAA\x66\x69\x99\x6A\xA9\xAA\x66\x69\x99";
imatgeC lletra_e = {
    6, 10, lletra_e_data
};

unsigned char lletra_f_data[] =
    "\x66\x69\x99\x6A\xA9\xAA\x66\x69\x99\x6A\xA9\xAA\x6A\xA9\xAA";
imatgeC lletra_f = {
    6, 10, lletra_f_data
};

unsigned char lletra_g_data[] =
    "\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x6A\xAA\x9A\xAA\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC lletra_g = {
    8, 10, lletra_g_data
};

unsigned char lletra_h_data[] =
    "\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9";
imatgeC lletra_h = {
    8, 10, lletra_h_data
};

unsigned char lletra_i_data[] =
    "\x69\x69\x69\x69\x69";
imatgeC lletra_i = {
    2, 10, lletra_i_data
};

unsigned char lletra_j_data[] =
    "\x66\x69\x99\xAA\x6A\xA9\xAA\x6A\xA9\x6A\x69\xA9\x66\x69\x9A";
imatgeC lletra_j = {
    6, 10, lletra_j_data
};

unsigned char lletra_k_data[] =
    "\x6A\xA6\x9A\x99\x6A\x66\x99\x9A\x66\x6A\x99\xAA\x66\x6A\x99\x9A\x6A\x66\x9A\x99";
imatgeC lletra_k = {
    8, 10, lletra_k_data
};

unsigned char lletra_l_data[] =
    "\x6A\xA9\xAA\x6A\xA9\xAA\x6A\xA9\xAA\x6A\xA9\xAA\x66\x69\x99";
imatgeC lletra_l = {
    6, 10, lletra_l_data
};
```

```

unsigned char lletra_m_data[] =
    "\x66\xA6\x66\x66\x66\x99\xA6\x69\xA6\x69\x9A\x66\x9A\x66\xAA\xA6\x6A\xA6\xAA\x66\xAA\x60";
imatgeC lletra_m = {
    9, 10, lletra_m_data
};

unsigned char lletra_n_data[] =
    "\x6A\xA6\x9A\xA9\x66\xA6\x99\xA9\x6A\x66\x99\xA9\x6A\x66\x9A\x99\x6A\xA6\x9A\xA9";
imatgeC lletra_n = {
    8, 10, lletra_n_data
};

unsigned char lletra_o_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC lletra_o = {
    8, 10, lletra_o_data
};

unsigned char lletra_p_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x6A\xAA\x9A\xAA";
imatgeC lletra_p = {
    8, 10, lletra_p_data
};

unsigned char lletra_q_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x6A\x66\x9A\x9A\x66\x66\x99\xA9";
imatgeC lletra_q = {
    8, 10, lletra_q_data
};

unsigned char lletra_r_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x9A\x6A\x66\x9A\x99";
imatgeC lletra_r = {
    8, 10, lletra_r_data
};

unsigned char lletra_s_data[] =
    "\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99";
imatgeC lletra_s = {
    8, 10, lletra_s_data
};

unsigned char lletra_t_data[] =
    "\x99\x99\x66\x66\xA9\xAA\xAA\x6A\xA9\xAA\xAA\x6A\xA9\xAA\xAA\x6A\xA9\xAA\xAA\x6A";
imatgeC lletra_t = {
    8, 10, lletra_t_data
};

unsigned char lletra_u_data[] =
    "\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC lletra_u = {
    8, 10, lletra_u_data
};

unsigned char lletra_v_data[] =
    "\x9A\xA9\x9A\xAA\x99\xAA\xA9\x9A\xA9\x99\xA9\x99\x99\x9A\x9A\x9A\xA9\x9A\xAA\x9A\xA0";
imatgeC lletra_v = {
    9, 10, lletra_v_data
};

unsigned char lletra_w_data[] =
    "\x40\x0\x44\x0\x44\x0\x4\x40\x4\x40\x0\x44\x0\x44\x4\x4\x44\x44\x44\x4\x44\x0\x40";
imatgeC lletra_w = {
    9, 10, lletra_w_data
};

unsigned char lletra_x_data[] =
    "\x9A\xA9\x99\xA9\x99\x99\x9A\x99\x9A\xA9\x9A\xAA\x9A\xAA\x99\xAA\x99\x9A\x99\x99\x9A\x90";
imatgeC lletra_x = {
    9, 10, lletra_x_data
};

unsigned char lletra_y_data[] =
    "\x9A\xA9\x99\xA9\x99\x99\x9A\x99\x9A\xA9\x9A\xAA\x9A\xAA\x99\xAA\xA9\xA9\x9A\xAA\x9A\xA0";

```

```

imatgeC lletra_y = {
    9, 10, lletra_y_data
};

unsigned char lletra_z_data[] =
    "\x44\x44\x11\x11\x0\x4\x0\x11\x0\x44\x1\x10\x4\x40\x11\x0\x44\x44\x11\x11";
imatgeC lletra_z = {
    8, 10, lletra_z_data
};

unsigned char espai_data[] =
    "\xAA\xAA\xA0";
imatgeC espai = {
    1, 10, espai_data
};

unsigned char punt_data[] =
    "\xAA\xAA\xAA\xAA\x69";
imatgeC punt = {
    2, 10, punt_data
};

unsigned char dospunts_data[] =
    "\xAA\x69\xAA\x69\xAA";
imatgeC dospunts = {
    2, 10, dospunts_data
};

unsigned char percent_data[] =
    "\x6A\xA6\x9A\x99\xAA\x66\xA9\x9A\xAA\x6A\xA9\xAA\xA6\x6A\x99\xAA\x66\xA6\x9A\xA9";
imatgeC percent = {
    8, 10, percent_data
};

unsigned char zero_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC zero = {
    8, 10, zero_data
};

unsigned char un_data[] =
    "\x66\x99\xA6\xA9\xA6\xA9\xA6\xA9\xA6\xA9";
imatgeC un = {
    4, 10, un_data
};

unsigned char dos_data[] =
    "\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x66\x66\x99\x99";
imatgeC dos = {
    8, 10, dos_data
};

unsigned char tres_data[] =
    "\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99";
imatgeC tres = {
    8, 10, tres_data
};

unsigned char quatre_data[] =
    "\x6A\xA6\x9A\xA9\x6A\xA6\x9A\xA9\x66\x66\x99\x99\xAA\xA6\xAA\xA9\xAA\xA6\xAA\xA9";
imatgeC quatre = {
    8, 10, quatre_data
};

unsigned char cinc_data[] =
    "\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99";
imatgeC cinc = {
    8, 10, cinc_data
};

unsigned char sis_data[] =
    "\x66\x66\x99\x99\x6A\xAA\x9A\xAA\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC sis = {
    8, 10, sis_data
};

```

```

};

unsigned char set_data[] =
    "\x66\x66\x99\x99\xAA\xA6\x69\xA6\x9A\xA9\xAA\xA6\xAA\xA9\xAA\xA6\xAA\xA9\xAA";
imatgeC set = {
    8, 10, set_data
};

unsigned char huit_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99";
imatgeC huit = {
    8, 10, huit_data
};

unsigned char nou_data[] =
    "\x66\x66\x99\x99\x6A\xA6\x9A\xA9\x66\x66\x99\x99\xAA\xA6\xAA\xA9\x66\x66\x99\x99";
imatgeC nou = {
    8, 10, nou_data
};

unsigned char p_zero_data[] =
    "\x0\x0\x0\x5\x0\x50\x5\x0\x50\x5\x0\x50\x0\x0\x0";
imatgeC p_zero = {
    6, 10, p_zero_data
};

unsigned char p_un_data[] =
    "\x0\x50\x5\x50\x55\x5\x50\x55\x5\x50\x55\x5\x50\x55\x5";
imatgeC p_un = {
    6, 10, p_un_data
};

unsigned char p_dos_data[] =
    "\x0\x0\x0\x55\x5\x50\x0\x0\x0\x5\x50\x55\x0\x0\x0";
imatgeC p_dos = {
    6, 10, p_dos_data
};

unsigned char p_tres_data[] =
    "\x0\x0\x0\x55\x5\x50\x0\x0\x0\x55\x5\x50\x0\x0\x0";
imatgeC p_tres = {
    6, 10, p_tres_data
};

unsigned char p_quatre_data[] =
    "\x5\x0\x50\x5\x50\x0\x0\x0\x55\x5\x50\x55\x5\x50";
imatgeC p_quatre = {
    6, 10, p_quatre_data
};

unsigned char p_cinc_data[] =
    "\x0\x0\x0\x5\x50\x55\x0\x0\x0\x55\x5\x50\x0\x0\x0";
imatgeC p_cinc = {
    6, 10, p_cinc_data
};

unsigned char p_sis_data[] =
    "\x0\x0\x0\x5\x50\x55\x0\x0\x0\x5\x0\x50\x0\x0\x0";
imatgeC p_sis = {
    6, 10, p_sis_data
};

unsigned char p_set_data[] =
    "\x0\x0\x0\x55\x5\x50\x55\x5\x50\x55\x5\x50\x55\x5\x50";
imatgeC p_set = {
    6, 10, p_set_data
};

unsigned char p_vuit_data[] =
    "\x0\x0\x0\x5\x50\x0\x0\x0\x5\x0\x50\x0\x0\x0";
imatgeC p_vuit = {
    6, 10, p_vuit_data
};

```



```

unsigned char p_nou_data[] =
    "\x0\x0\x0\x5\x0\x50\x0\x0\x0\x55\x5\x50\x0\x0\x0";
imatgeC p_nou = {
    6, 10, p_nou_data
};

unsigned char punts_data[] =
    "\x80\x0\x0\xA\xA0\xAA\x0\x28\x0\xAA\xA\xA0\x0\x0\x2";
imatgeC punts = {
    6, 10, punts_data
};

unsigned char record_r_data[] =
    "\x0\x0\x0\x5\x0\x50\x0\x0\x0\x50\x1\x4\x0\x50";
imatgeC record_r = {
    6, 10, record_r_data
};

unsigned char record_data[] =
    "\xA\x0\xA0\xA\x0\xA0\x0\x0\xA\x0\xA0\xA\x0\xA0";
imatgeC record = {
    6, 10, record_data
};

unsigned char punts_m_data[] =
    "\x2A\x80\xA0\x0\x0\x0\xA\x0\xA0\xA\x0\xA0\xA\x0\xA0";
imatgeC punts_m = {
    6, 10, punts_m_data
};

```

sprites/HUDsEspeciales.c

```
unsigned char SNAKEST_data[] =

"\x96\x59\x65\xAA\x5A\xAA\x96\xA9\x65\x96\x5A\xA5\xAA\xAA\xAA\x96\x59\x6A\x96\x59\x65\xAA\x59\x65\x96\x65\x96\x59\x69\x96\xAA\x65\xA6\x59\x65\x96\x99\x6A\xA5\xAA\x65\x96\x5A\x65\x96\x59\x69\x96\x59\x65\x96\x69\x66\xAA\x59\xA5\x96\x59\x66\x96\x6A\x99\x6A\x59\x65\x9A\x59\x65\x96\x69\x65\x96\x59\x96\x65\xAA\x59\x6A\x96\xA9\x65\x96\x5A\xA5\xAA\x56\x6A\x96\x59\x6A\x96\x59\x65\xAA\x59\x65\x96\x65\xAA\xAA\xA9\x96\x5A\x65\xA6\x5A\xA9\x96\x99\x69\x95\xAA\x65\xAA\xAA\x65\xAA\xAA\xAA\xAA\x59\xAA\x59\xAA\xAA\xA9\x65\x9A\x59\xA5\x9A\xA9\x66\x96\x69\x66\xAA\x59\xAA\xAA\x59\xAA\xAA\xAA\xAA\x96\xAA\x96\x65\x59\x65\xAA\x59\x65\x96\xA9\x65\x96\x5A\xA5\x96\x5A\xAA\x96\x59\x6A\x96\x59\x65\xAA\xAA\x65\xAA\x65\x96\x59\x69\x96\x99\x65\xA6\x59\x65\x96\x99\x65\xAA\xAA\x65\x96\x5A\x65\x96\x59\x6A\xAA\x59\xAA\x59\x65\x96\x69\x66\x96\x59\xA5\x96\x59\x66\x96\x59\xAA\xAA\x59\x65\x9A\x59\x65\x96\x6A\xAA\x96\xAA\x96\x59\x65\xAA\x5A\xA5\x96\xA9\x65\x96\x5A\xA5\x96\x5A\xAA\x96\x59\x6A\x96\x59\x65\xAA\xAA\x65\xAA\xAA\xAA\x99\x69\x96\xAA\x65\xA6\x5A\xA9\x96\x99\x69\x96\xAA\x65\xAA\xAA\xAA\x99\x6A\xAA\x59\xAA\xAA\xAA\x96\x69\x66\xAA\x59\xA5\x9A\xA9\x66\x96\x69\x65\xAA\x59\xAA\xAA\xAA\xAA\x96\x6A\xAA\x96\xAA\x96\x59\x65\xAA\x5A\xAA\x96\xA9\x6A\xAA\x5A\xA5\xAA\x59\x6A\x96\x59\x6A\x96\x59\x65\xAA\xAA\x65\xAA\x65\x96\x59\x69\x96\xAA\x65\xA6\x5A\xA9\x96\x99\x6A\x96\x5A\x65\x96\x5A\x65\x96\x59\x6A\xAA\x59\xAA\x59\x65\x96\x69\x66\xAA\x59\xA5\x9A\xA9\x66\x96\x6A\xA5\x9A\x59\x65\x9A\x59\x65\x96\x6A\xAA\x55\xAA\xA9\x65\x59\x65\xAA\x5A\xAA\x96\xA9\x6A\xAA\x5A\xA5\xAA\xA9\x6A\x96\x59\x6A\x96\x59\x65\xAA\xAA\x96\xAA";
imatgeC SNAKEST = {
    100, 16, SNAKEST_data
};

unsigned char pausa_P_data[] =

"\x80\x0\x2\xA8\x14\x51\x42\xA1\x14\x51\x4A\x85\x14\x51\x2\x5\x14\x51\x40\x45\x0\x11\x41\x44\x0\x51\x1\x45\x14\x50\x11\x45\x14\x50\x51\x45\x14\x40\x51\x45\x14\x4\x50\x0\x2\x14\x4A\xAA\xA8\x14\x2A\xAA\xA1\x14\xAA\xAA\x85\x12\xAA\xAA\x5\xA\xAA\xAA\x0\xAA\xAA\xA0";
imatgeC pausa_P = {
    15, 18, pausa_P_data
};

unsigned char pausa_A_data[] =

"\x80\x0\x0\x8\x14\x51\x45\x1\x14\x51\x45\x5\x14\x51\x44\x5\x14\x51\x40\x45\x0\x11\x41\x44\x0\x51\x1\x45\x14\x50\x11\x45\x14\x50\x51\x45\x14\x40\x51\x45\x14\x4\x50\x1\x14\x14\x4A\x85\x10\x14\x2A\x5\x1\x14\xA8\x45\x5\x12\xA1\x44\x5\xA\x81\x42\x0\xAA\x80\x20";
imatgeC pausa_A = {
    15, 18, pausa_A_data
};

unsigned char pausa_U_data[] =

"\x80\x2A\xA0\x8\x14\x2A\x5\x1\x14\xA8\x45\x5\x12\xA1\x44\x5\xA\x81\x40\x45\x2A\x11\x41\x44\xA8\x51\x1\x42\xA0\x50\x11\x4A\x84\x50\x51\x2A\x14\x40\x50\xA8\x14\x4\x52\xA1\x14\x14\x40\x5\x10\x14\x51\x45\x1\x14\x14\x51\x45\x5\x14\x51\x44\x5\x14\x51\x42\x0\x0\x0\x20";
imatgeC pausa_U = {
    15, 18, pausa_U_data
};

unsigned char pausa_S_data[] =

"\x80\x0\x0\x8\x14\x51\x45\x1\x14\x51\x45\x5\x14\x51\x44\x5\x14\x51\x40\x45\x0\x0\x21\x44\x0\x0\x81\x45\x14\x50\x11\x45\x14\x50\x51\x45\x14\x40\x51\x45\x14\x20\x0\x1\x14\x80\x0\x5\x10\x14\x51\x45\x1\x14\x51\x45\x5\x14\x51\x44\x5\x14\x51\x42\x0\x0\x0\x20";
imatgeC pausa_S = {
    15, 18, pausa_S_data
};

unsigned char pausa_E_data[] =

"\x80\x0\x0\x8\x14\x51\x45\x1\x14\x51\x45\x5\x14\x51\x44\x5\x14\x51\x40\x45\x0\x0\x21\x44\x0\x0\x81\x45\x14\x50\x11\x45\x14\x50\x51\x45\x14\x40\x51\x45\x14\x4\x50\x0\x2\x14\x40\x0\x8\x14\x51\x45\x1\x14\x4\x51\x45\x5\x14\x51\x44\x5\x14\x51\x42\x0\x0\x0\x20";
imatgeC pausa_E = {
    15, 18, pausa_E_data
};

imatgeC lletresPAUSA[] = {
    {15, 18, pausa_P_data},
    {15, 18, pausa_A_data},
    {15, 18, pausa_U_data},
```

```

        {15, 18, pausa_S_data},
        {15, 18, pausa_E_data},
    };
I2_posicionsPAUSA[] = {{28,24},{43,24},{58,24},{73,24},{88,24}};
int indexA; int indexB;

unsigned char mira_data[] =
    "\x59\x5A\x9A\xA9\xA9\x59\x40";
imatgeC mira = {
    5, 5, mira_data
};

unsigned char creuA_data[] =
    "\xAA\xA6\x6A\x6A\x66\xAA\x80";
imatgeC creuA = {
    5, 5, creuA_data
};

unsigned char creuB_data[] =
    "\xAA\xAA\xAA\x6A\xAA\xAA\x80";
imatgeC creuB = {
    5, 5, creuB_data
};

unsigned char creuC_data[] =
    "\xAA\xA9\xA9\x9A\x9A\xAA\x80";
imatgeC creuC = {
    5, 5, creuC_data
};

unsigned char cercle_data[] =
    "\x15\x15\x55\x55\x55\x15\x0";
imatgeC cercle = {
    5, 5, cercle_data
};

unsigned char cercleNEG_data[] =
    "\x6A\x6A\xAA\xAA\xAA\x6A\x40";
imatgeC cercleNEG = {
    5, 5, cercleNEG_data
};

unsigned char Nord_data[] =
    "\x4\x5\x44\x44\x10\x4\x0";
imatgeC Nord = {
    5, 5, Nord_data
};

unsigned char Sud_data[] =
    "\x4\x1\x4\x44\x54\x4\x0";
imatgeC Sud = {
    5, 5, Sud_data
};

unsigned char Est_data[] =
    "\x4\x0\x45\x54\x4\x4\x0";
imatgeC Est = {
    5, 5, Est_data
};

unsigned char Oest_data[] =
    "\x4\x4\x5\x54\x40\x4\x0";
imatgeC Oest = {
    5, 5, Oest_data
};

unsigned char focus_data[] =

"\x55\x55\xAA\xAA\x95\x55\x55\x56\xAA\xAA\xAA\x55\x55\x5A\xAA\xAA\xAA\xA9\x55\x5A\xAA\xAA\xAA\xAA\x9
5\x5A\xAA\xAA\xAA\xAA\xA9\x56\xAA\xAA\xAA\xAA\x56\xAA\xAA\xAA\xAA\xAA\xA5\xAA\xAA\xAA\xAA\xAA\xA
9\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xA
A\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xAA\xA
A\xAA\xAA\xAA\xAA\xAA\xAA\x9A\xAA\xAA\xAA\xAA\x96\xAA\xAA\xAA\xAA\xAA\xA5\x6A\xAA\xAA\xAA\xAA\xAA
5\x5A\xAA\xAA\xAA\xAA\xA9\x55\xAA\xAA\xAA\xAA\xA9\x55\x5A\xAA\xAA\xAA\xA9\x55\x55\x6A\xAA\xAA\xA5\x5

```

```
5\x55\x55\xAA\xAA\x95\x55\x40";
imatgeC focus = {
    25, 25, focus_data
};

unsigned char cau_data[] =
    "\x15\x15\x55\x55\x55\x15\x0";
imatgeC cau = {
    5, 5, cau_data
};
```

sprites/Serps.c

```
unsigned char capN_data[] =
    "\x4\x5\x45\x55\x11\x55\x40";
imatgeC capN = {
    5, 5, capN_data
};

unsigned char capS_data[] =
    "\x55\x51\x15\x54\x54\x4\x0";
imatgeC capS = {
    5, 5, capS_data
};

unsigned char capE_data[] =
    "\x55\x51\x45\x41\x50\x55\x40";
imatgeC capE = {
    5, 5, capE_data
};

unsigned char capO_data[] =
    "\x55\x45\x10\x54\x15\x55\x40";
imatgeC capO = {
    5, 5, capO_data
};

unsigned char cos_data[] =
    "\x51\x51\x51\x51\x51\x51\x40";
imatgeC cos_ = {          // NOTE: He canviat el nom porque cos es comfon amb la funcio trigonometrica
    5, 5, cos_data
};

unsigned char cuaN_data[] =
    "\x15\x15\x55\x55\x51\x51\x40";
imatgeC cuaN = {
    5, 5, cuaN_data
};

unsigned char cuaS_data[] =
    "\x51\x51\x55\x55\x55\x15\x0";
imatgeC cuaS = {
    5, 5, cuaS_data
};

unsigned char cuaE_data[] =
    "\x55\x11\x51\x55\x55\x55\x0";
imatgeC cuaE = {
    5, 5, cuaE_data
};

unsigned char cuaO_data[] =
    "\x15\x55\x55\x51\x51\x15\x40";
imatgeC cuaO = {
    5, 5, cuaO_data
};

//ESPECIALETS BELOW
unsigned char capEfant_data[] =
    "\x80\x2\x55\x49\x14\x25\x40\x95\x2\x55\x48\x0\x0";
imatgeC capEfant = {
    7, 7, capEfant_data
};

unsigned char capNfant_data[] =
    "\xA0\x2A\x4\x20\x54\x5\x54\x11\x10\x55\x4A\xAA\x80";
imatgeC capNfant = {
    7, 7, capNfant_data
};

unsigned char capOfant_data[] =
    "\x0\x8\x55\x60\x51\x80\x56\x1\x58\x55\x60\x0\x80";
imatgeC capOfant = {
    7, 7, capOfant_data
};
```

```

};

unsigned char capSfant_data[] =
    "\xAA\xA8\x55\x41\x11\x5\x54\x5\x42\x4\x2A\x2\x80";
imatgeC capSfant = {
    7, 7, capSfant_data
};

unsigned char cuaEfant_data[] =
    "\x80\xA\x55\x9\x15\x21\x54\x95\x52\x55\x8\x0\x80";
imatgeC cuaEfant = {
    7, 7, cuaEfant_data
};

unsigned char cuaOfant_data[] =
    "\x80\x8\x15\x61\x55\x85\x52\x15\x18\x15\x68\x0\x80";
imatgeC cuaOfant = {
    7, 7, cuaOfant_data
};

unsigned char cuaSfant_data[] =
    "\xAA\xA8\x51\x41\x51\x5\x54\x15\x50\x15\x8\x0\x80";
imatgeC cuaSfant = {
    7, 7, cuaSfant_data
};

unsigned char cuaNfant_data[] =
    "\x80\x8\x15\x1\x55\x5\x54\x11\x50\x51\x4A\xAA\x80";
imatgeC cuaNfant = {
    7, 7, cuaNfant_data
};

unsigned char cosXfant_data[] =
    "\x80\xA\x51\x69\x15\xA1\x52\x95\x1A\x51\x68\x0\x80";
imatgeC cosXfant = {
    7, 7, cosXfant_data
};

unsigned char cosYfant_data[] =
    "\xAA\xA8\x51\x41\x15\x1\x50\x15\x10\x51\x4A\xAA\x80";
imatgeC cosYfant = {
    7, 7, cosYfant_data
};

unsigned char cosSOfant_data[] =
    "\x80\x2\x51\x49\x15\x21\x50\x95\x12\x51\x4A\xAA\x80";
imatgeC cosSOfant = {
    7, 7, cosSOfant_data
};

unsigned char cosSEfant_data[] =
    "\x0\x8\x51\x61\x15\x81\x52\x15\x18\x51\x6A\xAA\x80";
imatgeC cosSEfant = {
    7, 7, cosSEfant_data
};

unsigned char cosNOfant_data[] =
    "\xAA\xAA\x51\x49\x15\x21\x50\x95\x12\x51\x48\x0\x0";
imatgeC cosNOfant = {
    7, 7, cosNOfant_data
};

unsigned char cosNEfant_data[] =
    "\xAA\xA8\x51\x61\x15\x81\x52\x15\x18\x51\x60\x0\x80";
imatgeC cosNEfant = {
    7, 7, cosNEfant_data
};

unsigned char diagonalNE_data[] =
    "\x99\xA9\x9A\x9A\xAA\x80";
imatgeC diagonalNE = {
    5, 5, diagonalNE_data
};

```

```

unsigned char diagonalNO_data[] =
    "\x59\x99\xA9\xA9\xAA\xAA\x80";
imatgeC diagonalNO = {
    5, 5, diagonalNO_data
};

unsigned char diagonalSE_data[] =
    "\xAA\xAA\x9A\x9A\x99\x99\x40";
imatgeC diagonalSE = {
    5, 5, diagonalSE_data
};

unsigned char diagonalSO_data[] =
    "\xAA\x9A\xA9\xA9\x9A\x99\x80";
imatgeC diagonalSO = {
    5, 5, diagonalSO_data
};

unsigned char capNE_data[] =
    "\x15\x54\x55\x45\x55\x55\x0";
imatgeC capNE = {
    5, 5, capNE_data
};

unsigned char capNO_data[] =
    "\x55\x14\x54\x55\x55\x15\x40";
imatgeC capNO = {
    5, 5, capNO_data
};

unsigned char capSE_data[] =
    "\x55\x15\x55\x45\x45\x15\x40";
imatgeC capSE = {
    5, 5, capSE_data
};

unsigned char capSO_data[] =
    "\x15\x55\x54\x55\x45\x55\x0";
imatgeC capSO = {
    5, 5, capSO_data
};

unsigned char cuaNE_data[] =
    "\x95\x95\x55\x55\x55\x55\x80";
imatgeC cuaNE = {
    5, 5, cuaNE_data
};

unsigned char cuaNO_data[] =
    "\x95\x95\x55\x55\x55\x95\x40";
imatgeC cuaNO = {
    5, 5, cuaNO_data
};

unsigned char cuaSE_data[] =
    "\x55\x95\x55\x55\x55\x95\x80";
imatgeC cuaSE = {
    5, 5, cuaSE_data
};

unsigned char cuaSO_data[] =
    "\x95\x55\x55\x55\x55\x95\x80";
imatgeC cuaSO = {
    5, 5, cuaSO_data
};

unsigned char cuaamunt_data[] =
    "\x99\x65\x5A\x5A\xA5\xAA\x80";
imatgeC cuaamunt = {
    5, 5, cuaamunt_data
};

unsigned char cuaavall_data[] =
    "\xAA\xAA\x5A\x5A\x55\x99\x40";

```

```

imatgeC cuaavall = {
    5, 5, cuaavall_data
};

unsigned char cosA13_data[] =
    "\x91\x91\x11\x11\x11\x91\x80";
imatgeC cosA13 = {
    5, 5, cosA13_data
};

unsigned char cosB13_data[] =
    "\x84\x84\x44\x44\x44\x84\x80";
imatgeC cosB13 = {
    5, 5, cosB13_data
};

unsigned char puntaN_data[] =
    "\x2A\x2A\xAA\xAA\xAA\xAA\x80";
imatgeC puntaN = {
    5, 5, puntaN_data
};

unsigned char puntaS_data[] =
    "\xAA\xAA\xAA\xAA\xAA\x2A\x0";
imatgeC puntaS = {
    5, 5, puntaS_data
};

unsigned char puntaE_data[] =
    "\xAA\x2A\xAA\xAA\xAA\xAA\x0";
imatgeC puntaE = {
    5, 5, puntaE_data
};

unsigned char puntaO_data[] =
    "\x2A\xAA\xAA\xAA\xAA\x2A\x80";
imatgeC puntaO = {
    5, 5, puntaO_data
};

```


sprites/Rates.c

```
unsigned char rata_data[] =
    "\x51\x54\x54\x44\x54\x4\x0";
imatgeC rata = {
    5, 5, rata_data
};

unsigned char rataNEG_data[] =
    "\x4\x1\x1\x11\x1\x51\x40";
imatgeC rataNEG = {
    5, 5, rataNEG_data
};

unsigned char rataFALSA_data[] =
    "\x59\x59\x96\x66\x56\xA6\x80";
imatgeC rataFALSA = {
    5, 5, rataFALSA_data
};

unsigned char rataN_data[] =
    "\xA6\xA5\x66\x65\x65\x59\x40";
imatgeC rataN = {
    5, 5, rataN_data
};

unsigned char rataS_data[] =
    "\x59\x56\x56\x66\x56\xA6\x80";
imatgeC rataS = {
    5, 5, rataS_data
};

unsigned char rataE_data[] =
    "\x56\x96\x6A\x55\x66\x56\x80";
imatgeC rataE = {
    5, 5, rataE_data
};

unsigned char rataO_data[] =
    "\xA5\x66\x55\x6A\x65\xA5\x40";
imatgeC rataO = {
    5, 5, rataO_data
};
```

funcions/funcions_motor.h

```
#define invertir(x) (!(x))

/*
int ps_mode_orig(Finestra *f, imatgeC *sprite, I2 pos, int mode){
    int pintar = 1;

    Pixel* pixels_origen = (Pixel *) sprite->pixel_data;

    int ample = sprite->width;
    int abs;
    int alcada = sprite->height;
    int ord;

    int pixel = 0;
    int posicioBP;

    while (pixel < ample*alcada){
        if (mode == 0) //Surt d'una banda i entra per l'oposada
        {
            abs = f->origen.x+(((pixel % ample) + pos.x) % (f->mida.x));
            if (abs < f->origen.x){abs = f->mida.x + abs;}

            ord = f->origen.y+(((pixel / ample) + pos.y) % (f->mida.y));
            if (ord < f->origen.y){ord = f->mida.y + ord;}
        }
        else //if(mode == 1) //Desapareix per les 4 bandes
        {
            pintar = 0;
            abs = (pixel % ample) + pos.x;
            ord = (pixel / ample) + pos.y;
            if (mode != 2){
                if (abs >= f->origen.x+f->mida.x || abs < f->origen.x || ord >=
f->origen.y+f->mida.y || ord < f->origen.y)
                {}//El pixel és fora de la pantalla
                else {pintar = 1;}
            }
        }
        //else{printf("Mode no vàlid: %i",mode); break;}

        Pixel foto;
        foto = (( pixels_origen[pixel/4] & (0x3<<(6-2*(pixel%4))) ) >> (6-2*(pixel%4)));
        if(foto == 2){
            pintar = 0;
            if(mode == 2){pintar = 2;}
        }

        posicioBP = (ord*bp.width)+abs;
        if (pintar == 1){f->bitmap.pixels[posicioBP] = foto;}
        else if (pintar == 2){f->bitmap.pixels[posicioBP] = invertir(f->bitmap.pixels[posicioBP]);}

        pixel += 1;
    }

    return 0;
}
*/

int ps_mode(Finestra *f, imatgeC *sprite, I2 pos, int mode){
    BEG_TIMER(ps_mode);

    Pixel* pixels_origen = (Pixel *) sprite->pixel_data;

    int ample = sprite->width;
    int abs;
    int alcada = sprite->height;
    int ord;

    #if 1
        I2 top_left = pos;
        I2 top_left_sprite = {0, 0};
        // NOTE: Respecte a la finestra
    #endif
}
```

```

if(top_left.x < f->origen.x){
    U32 diff = f->origen.x - top_left.x;
    ample -= diff;
    top_left.x = f->origen.x;
    top_left_sprite.x = diff;
}
if(top_left.y < f->origen.y){
    U32 diff = f->origen.y - top_left.y;
    alcada -= diff;
    top_left.y = f->origen.y;
    top_left_sprite.y = diff;
}

I2 bottom_right = {pos.x + ample, pos.y + alcada}; // NOTE: Respeste a la finestra

if(bottom_right.x > f->origen.x + f->mida.x){
    ample -= bottom_right.x - (f->origen.x + f->mida.x);
    bottom_right.x = f->origen.x + f->mida.x;
}
if(bottom_right.y > f->origen.y + f->mida.y){
    alcada -= bottom_right.y - (f->origen.y + f->mida.y);
    bottom_right.y = f->origen.y + f->mida.y;
}

I2 bottom_right_sprite = {top_left_sprite.x + ample, top_left_sprite.y + alcada};

#endif

int pixel = 0;
int posicioBP;

if(mode == 2){
    // TODO: Aquesta branca encara s'ha d'arreglar
    while (pixel < ample*alcada){
        Pixel foto = (( pixels_origen[pixel/4] & (0x3<<(6-2*(pixel%4))) ) >> (6-2*(pixel%4)));
        if(foto == 2){
            abs = (pixel % ample) + pos.x;
            ord = (pixel / ample) + pos.y;
            posicioBP = (ord*bp.width)+abs;
            f->bitmap.pixels[posicioBP] = invertir(f->bitmap.pixels[posicioBP]);
        }
        pixel += 1;
    }
}
else{
    // NOTE: He deixat dues opcions per aquesta branca
#if 1
    for(int ord = 0; ord < alcada; ++ord){ // NOTE: Un 30% mes rapida que l'altra opcio,
        pero s'enten pitjor
        pixel = (ord+top_left_sprite.y)*sprite->width + top_left_sprite.x;
        posicioBP = (ord+top_left.y)*bp.width + top_left.x;

        int abs = 0;
        while(abs < ample){
            Pixel pixel_sprite = pixels_origen[pixel/4];

            switch(pixel%4){
                case 0:
                    if(!(pixel_sprite & 0x80)){
                        f->bitmap.pixels[posicioBP] = pixel_sprite>>6 & 0x1;
                    }
                    ++posicioBP; ++abs;
                    if(abs == ample) break;
                case 1:
                    if(!(pixel_sprite & 0x20)){
                        f->bitmap.pixels[posicioBP] = pixel_sprite>>4 & 0x1;
                    }
                    ++posicioBP; ++abs;
                    if(abs == ample) break;
                case 2:
                    if(!(pixel_sprite & 0x08)){
                        f->bitmap.pixels[posicioBP] = pixel_sprite>>2 & 0x1;
                    }
                    ++posicioBP; ++abs;

```

```

        if(abs == ample) break;
    case 3:
        if(!(pixel_sprite & 0x02)){
            f->bitmap.pixels[posicioBP] = pixel_sprite & 0x1;
        }

        ++posicioBP; ++abs;
        if(abs == ample) break;
        break;
    }
    pixel += 4 - (pixel%4);
}
}
#else
for(int ord = 0; ord < alcada; ++ord){
    pixel = (ord+top_left_sprite.y)*sprite->width + top_left_sprite.x;
    posicioBP = (ord+top_left.y)*bp.width + top_left.x;

    for(int abs = 0; abs < ample; ++abs){
        Pixel pixel_sprite = ( pixels_origen[pixel/4] >> (6-2*(pixel%4)) ) & 0x3;
        if(pixel_sprite != 2){
            f->bitmap.pixels[posicioBP] = pixel_sprite;
        }
        ++pixel; ++posicioBP;
    }
}
#endif
}

END_TIMER(ps_mode);

return 0;
}

int ps(Finestra *f, imatgeC *sprite, I2 pos){
    return ps_mode(f, sprite, pos, 1);
}
/*
int performa_mode_orig(Finestra *f, I2 pos, int ample, int alcada, Pixel *color, int mode){
    BEG_TIMER(ps_mode);
    int pintar = 1;
    int abs;
    int ord;

    int pixel = 0;
    int posicioBP;

    while (pixel < ample*alcada){
        if (mode == 0)
        {
            abs = f->origen.x+((pixel % ample) + pos.x) % (f->mida.x));
            if (abs < f->origen.x){abs = f->mida.x + abs;}

            ord = f->origen.y+((pixel / ample) + pos.y) % (f->mida.y));
            if (ord < f->origen.y){ord = f->mida.y + ord;}
        }
        else
        {
            pintar = 0;
            abs = (pixel % ample) + pos.x;
            ord = (pixel / ample) + pos.y;
            if (abs >= f->origen.x+f->mida.x || abs < f->origen.x || ord >= f->origen.y+f->mida.y ||
ord < f->origen.y) //El pixel és fora de la pantalla
            {}
            else {pintar = 1;}
        }
        //else{printf("Mode no vàlid: %i",mode); break;}

        if(pintar == 1)
        {
            posicioBP = (ord*bp.width)+abs;
            if (mode == 2)
            {f->bitmap.pixels[posicioBP] = invertir(f->bitmap.pixels[posicioBP]);}

```

```

        else if (mode >= 1000)
        {
            int possibilitatpercent = mode-1000;
            int tiradadedaus = aleat(0,1000);
            if (tiradadedaus > possibilitatpercent)
                {f->bitmap.pixels[posicioBP] = blanc;}
            else{f->bitmap.pixels[posicioBP] = negre;}
        }
        else{f->bitmap.pixels[posicioBP] = *color;}
    }
    pixel += 1;
}

END_TIMER(ps_mode);

return 0;
}
*/

int ferforma_mode(Finestra *f, I2 pos, int ample, int alcada, Pixel color, int mode /*= 1 */){
    BEG_TIMER(ferforma_mode);
    int posicioBP;

    I2 top_left = pos;
    if(top_left.x < f->origen.x){
        ample -= f->origen.x - top_left.x;
        top_left.x = f->origen.x;
    }
    if(top_left.y < f->origen.y){
        alcada -= f->origen.y - top_left.y;
        top_left.y = f->origen.y;
    }
    I2 bottom_right = {pos.x + ample-1, pos.y + alcada-1};
    // IMPORTANT: Si fas servir alcada, amplada, corregeix-los
    if(bottom_right.x > f->origen.x + f->mida.x) bottom_right.x = f->origen.x + f->mida.x;
    if(bottom_right.y > f->origen.y + f->mida.y) bottom_right.y = f->origen.y + f->mida.y;

    if (mode == 2){
        for(U32 ord = top_left.y; ord <= (U32)bottom_right.y; ++ord){
            posicioBP = ord * bp.width + top_left.x;
            for(U32 abs = top_left.x; abs <= (U32)bottom_right.x; ++abs){
                f->bitmap.pixels[posicioBP] = invertir(f->bitmap.pixels[posicioBP]);
                ++posicioBP;
            }
        }
    }
    else if (mode >= 1000){
        for(U32 ord = top_left.y; ord <= (U32)bottom_right.y; ++ord){
            posicioBP = ord * bp.width + top_left.x;
            for(U32 abs = top_left.x; abs <= (U32)bottom_right.x; ++abs){
                int possibilitatpercent = mode-1000;
                int tiradadedaus = aleat(0,1000);
                f->bitmap.pixels[posicioBP] = (tiradadedaus > possibilitatpercent);
                ++posicioBP;
            }
        }
    }
    else{
        for(U32 ord = top_left.y; ord <= (U32)bottom_right.y; ++ord){
            posicioBP = ord * bp.width + top_left.x;
            for(U32 abs = top_left.x; abs <= (U32)bottom_right.x; ++abs){
                f->bitmap.pixels[posicioBP] = color;
                ++posicioBP;
            }
        }
    }
    END_TIMER(ferforma_mode);
    return 0;
}

int ferforma(Finestra *f, I2 pos, int ample, int alcada, Pixel color){
    return ferforma_mode(f, pos, ample, alcada, color, 1);
}

```

funcions/funcions_menu.h

```
typedef struct Menu{
    char* titol;
    int opcions;
    char* noms[7];
    unsigned char llargades_pix[7];
    short int xifres[3];
}Menu;

Menu menu_inici = {
    .titol = "inici",
    .opcions = 2,
};
Menu menu_principal = {
    .titol = "modes",
    .opcions = 7,
};
Menu menu_control = {
    .titol = "control",
    .opcions = 3,
};

Menu menu_limits = {
    .titol = "limits",
    .opcions = 3,
};
Menu menu_editor = {
    .titol = "editor",
    .opcions = 2,
};

Menu menu_serp = {
    .titol = "snake",
    .opcions = 7,
};
Menu menu_rata = {
    .titol = "rat",
    .opcions = 3,
};
Menu menu_grafic = {
    .titol = "graphics",
    .opcions = 2,
};
Menu menu_objectiu = {
    .titol = "goal",
    .opcions = 6,
};
Menu menu_altres = {
    .titol = "other",
    .opcions = 2,
};

imatgeC lletres[41] = {
    {8, 10, lletra_a_data},
    {8, 10, lletra_b_data},
    {6, 10, lletra_c_data},
    {8, 10, lletra_d_data},
    {6, 10, lletra_e_data},
    {6, 10, lletra_f_data},
    {8, 10, lletra_g_data},
    {8, 10, lletra_h_data},
    {2, 10, lletra_i_data},
    {6, 10, lletra_j_data},
    {8, 10, lletra_k_data},
    {6, 10, lletra_l_data},
    {9, 10, lletra_m_data},
    {8, 10, lletra_n_data},
    {8, 10, lletra_o_data},
    {8, 10, lletra_p_data},
    {8, 10, lletra_q_data},
    {8, 10, lletra_r_data},
    {8, 10, lletra_s_data},
    {8, 10, lletra_t_data},
```

```

        {8, 10, lletra_u_data},
        {9, 10, lletra_v_data},
        {9, 10, lletra_w_data},
        {9, 10, lletra_x_data},
        {9, 10, lletra_y_data},
        {8, 10, lletra_z_data},
        {2, 10, punt_data},
        {2, 10, dospunts_data},
        {8, 10, percent_data},
        {1, 10, espai_data},
        {8, 10, zero_data},
        {4, 10, un_data},
        {8, 10, dos_data},
        {8, 10, tres_data},
        {8, 10, quatre_data},
        {8, 10, cinc_data},
        {8, 10, sis_data},
        {8, 10, set_data},
        {8, 10, vuit_data},
        {8, 10, nou_data}
};
char abc[41] = "abcdefghijklmnopqrstuvwxyz.:% 0123456789";
I2 posicions_op_inici[] = {{4,32},{4,44}};
I2 posicions_op[] = {{4,14},{4,26},{4,38},{4,50}};

int comptar_llargada_pix(char* paraula){
    int llargadaOPCIO = 0;
    for(int c = 0; c < (int)strlen(paraula); c+=1){
        for(int ll = 0; ll < 41; ll+=1){
            if (paraula[c] == abc[ll]){
                llargadaOPCIO += lletres[ll].width+2;
            }
        }
    }
    return llargadaOPCIO;
}

int pintar_paraula(char* paraula, I2 pos){
    int x = pos.x; int y = pos.y;
    for(int c = 0; c < (int)strlen(paraula); c+=1){
        for(int ll = 0; ll < 41; ll+=1){
            if (paraula[c] == abc[ll]){
                ps(&finestra_general, lletres[ll], i2(x,y));
                x += lletres[ll].width+2;
            }
        }
    }
    return x;
}

int pintar_puntuacio(char* nom, I2 pos, int puntuacioINT){
    int centenens = puntuacioINT / 100;
    int desenes = (puntuacioINT - 100*centenens) / 10;
    int unitats = puntuacioINT - 100*centenens - 10*desenes;
    char puntuacioCHAR[] = {abc[30+centenens], abc[30+desenes], abc[30+unitats]};

    pintar_paraula(nom, pos);
    pintar_paraula(puntuacioCHAR, i2(59, pos.y));

    return comptar_llargada_pix(puntuacioCHAR);
}

void inicialitzar_menu(Menu *me, char* noms_de_les_opcions[], int xifres_de_les_opcions[] /*= {}*/,
char mode /*= 'o'*/) {
    for(int i = 0; i < me->opcions; i+=1){
        me->noms[i] = noms_de_les_opcions[i];
        me->llargades_pix[i] = comptar_llargada_pix(noms_de_les_opcions[i]);
        if(mode == 'x'){me->xifres[i] = xifres_de_les_opcions[i];}
    }
}

void inicialitzar_TOTS_menus(){
    char* noms_inici[] = {"original snake", "modes"};
    char* noms_principal[] = {"control", "limits", "snake", "rat", "graphics", "goal", "other"};

```

```

char* noms_control[] = {"inverted","egocentric","strategic"};

char* noms_limits[] = {"edges","ghost","editor"};
char* noms_editor[] = {"edit","play"};

char* noms_serp[] =
{"adjustable","manual","accelerating","capicua","spitting","diagonal","smooth"};
char* noms_rata[] = {"too many","directional","mobile"};
char* noms_grafic[] = {"survival","blind hen"};
char* noms_objectiu[] = {"everything","hunger","basket","loops","ld","ai"};
char* noms_altres[] = {"retro","multiplayer"};

inicialitzar_menu(&menu_inici, noms_inici, 0, 'o');
inicialitzar_menu(&menu_principal, noms_principal, 0, 'o');
inicialitzar_menu(&menu_control, noms_control, 0, 'o');

inicialitzar_menu(&menu_limits, noms_limits, 0, 'o');
inicialitzar_menu(&menu_editor, noms_editor, 0, 'o');

inicialitzar_menu(&menu_serp, noms_serp, 0, 'o');
inicialitzar_menu(&menu_rata, noms_rata, 0, 'o');
inicialitzar_menu(&menu_grafic, noms_grafic, 0, 'o');
inicialitzar_menu(&menu_objectiu, noms_objectiu, 0, 'o');
inicialitzar_menu(&menu_altres, noms_altres, 0, 'o');
}

int pintar_menus(Menu *menu, int op){
ferforma(&finestra_general,i2(0,0),128,64,negre);
int x; int y;
if (menu->titol == menu_inici.titol){
x = 64-50; y=7;
ps(&finestra_general, &SNAKEST,i2(x,y));
x = 5; y = 7+16+10;
}
else{
int llargadaTITOL = comptar_llargada_pix(menu->titol);
x = 64-llargadaTITOL/2; y = 2;
pintar_paraula(menu->titol,i2(x,y));
x = 5; y = 15;
}
int o_inicial; int o_final;
if(op/4 == 0){o_inicial = 0; o_final = 4;}
else{o_inicial = 4; o_final = menu->opcions;}
for (int o = o_inicial; o<o_final && o<menu->opcions; o+=1){
pintar_paraula(menu->noms[o],i2(x,y));
y+=12;}
if (controller.up.pressed && controller.up.half_transition_count && op>0){op-=1;}
if (controller.down.pressed && controller.down.half_transition_count &&
op<menu->opcions-1){op+=1;}
if(numero_de_fotograma % 20 < 10){
if (menu->titol == menu_inici.titol){ferforma_mode(&finestra_general,
posicions_op_inici[op], menu->llargades_pix[op], 12, blanc, 2);}
else{ferforma_mode(&finestra_general, posicions_op[op%4], menu->llargades_pix[op], 12,
blanc, 2);}
}

return op;
}

int pintar_menu_XIFRES(Menu *menu, int op, char mode /* = 'x' */){
ferforma(&finestra_general,i2(0,0),128,64,negre);
int llargadaTITOL = comptar_llargada_pix(menu->titol);
int x = 64-llargadaTITOL/2; int y = 2;
pintar_paraula(menu->titol,i2(x,y));
x = 5; y = 15;
int o_inicial; int o_final;
int llargada_xi;
for (int o = 0; o<3; o+=1){
if(mode == 'x'){llargada_xi = pintar_puntuacio(menu->noms[o],i2(x,y),menu->xifres[o]);}
else{pintar_paraula(menu->noms[o],i2(x,y));}
x=5; y+=12;}
}
if(mode == 'x'){ps(&finestra_general, &percent, i2(59+llargada_xi,y-12));}
pintar_paraula(menu->noms[3],i2(20,y)); pintar_paraula(menu->noms[4],i2(77,y));

```



```

    if (controller.left.pressed && controller.left.half_transition_count && op>0){op-=1;}
    if (controller.right.pressed && controller.right.half_transition_count && op<1){op+=1;}
    if(numero_de_fotograma % 20 < 10)
    {
        if(op==0){ferforma_mode(&finestra_general, i2(19,50), menu->llargades_pix[3], 12, blanc,
2);}
        else      {ferforma_mode(&finestra_general, i2(76,50), menu->llargades_pix[4], 12, blanc,
2);}
        //ferforma(finestra_general, {79+33,50}, menu->llargades_pix[4], 12, vermell);
    }
    return op;
}

```

funcions/funcions_joc.h

```
int pintarHUD_mode(int HS, int S, int mode /* = 0 */){

    if (S>=0 && HS>=0 && S<999 && HS<999){
        //Pintar vores i números
        ferforma(&finestra_general,i2(0,0),128,64,blanc);
        if(mode == 25){ferforma(&finestra_joc,i2(14,3),100,58,negre);}
        else if (mode == 13){ferforma(&finestra_joc,i2(14,2),100,60,negre);}
        else{ferforma(&finestra_general,i2(13,1),102,62,negre);}
        //else{ferforma(finestra_joc,{14,2},100,60,negre);}

        imatgeC numeros[] = {p_zero,p_un,p_dos,p_tres,p_quatre,p_cinc,p_sis,p_set,p_vuit,p_nou};

        int puntuacio = HS;
        int centenes = puntuacio / 100;
        int desenes = (puntuacio - 100*centenes) / 10;
        int unitats = puntuacio - 100*centenes -10*desenes;

        if(mode == 6){ps(&finestra_record,&record_r,i2(4,9));}
    else{ps(&finestra_record,&record,i2(4,9));}
        ps(&finestra_record,numeros+centenes,i2(4,21));
        ps(&finestra_record,numeros+desenes,i2(4,33));
        ps(&finestra_record,numeros+unitats,i2(4,45));

        puntuacio = S;
        centenes = puntuacio / 100;
        desenes = (puntuacio - 100*centenes) / 10;
        unitats = puntuacio - 100*centenes -10*desenes;

        if(mode == 6){ps(&finestra_puntuacio,&punts_m,i2(118,9));} else
    {ps(&finestra_puntuacio,&punts,i2(118,9));}
        ps(&finestra_puntuacio,numeros+centenes,i2(118,21));
        ps(&finestra_puntuacio,numeros+desenes,i2(118,33));
        ps(&finestra_puntuacio,numeros+unitats,i2(118,45));

        return 0;
    }
    else{return 1;}
}

int pintarHUD(int HS, int S){
    return pintarHUD_mode(HS, S, 0);
}

int pintarSERP(){
    for (int n = 1; n<s.length; n++){
        {ps(&finestra_joc,&cos_,s.posicions[n]);}

        if (s.dirR == 'N'){ ps(&finestra_joc,&capN,s.posicions[0]);}
        if (s.dirR == 'S'){ ps(&finestra_joc,&capS,s.posicions[0]);}
        if (s.dirR == 'E'){ ps(&finestra_joc,&capE,s.posicions[0]);}
        if (s.dirR == 'O'){ ps(&finestra_joc,&capO,s.posicions[0]);}

        if (s.dircua == 'N'){ ps(&finestra_joc,&cuaN,s.posicions[s.length]);}
        if (s.dircua == 'S'){ ps(&finestra_joc,&cuaS,s.posicions[s.length]);}
        if (s.dircua == 'E'){ ps(&finestra_joc,&cuaE,s.posicions[s.length]);}
        if (s.dircua == 'O'){ ps(&finestra_joc,&cuaO,s.posicions[s.length]);}

        return 0;
    }
}

int pintarSERP_FANTASMA(){
    if (s.dircua == 'N'){
        ps(&finestra_joc,&cuaNfant,i2(s.posicions[s.length].x-1,s.posicions[s.length].y-1));}
    if (s.dircua == 'S'){
        ps(&finestra_joc,&cuaSfant,i2(s.posicions[s.length].x-1,s.posicions[s.length].y-1));}
    if (s.dircua == 'E'){
        ps(&finestra_joc,&cuaEfant,i2(s.posicions[s.length].x-1,s.posicions[s.length].y-1));}
    if (s.dircua == 'O'){
        ps(&finestra_joc,&cuaOfant,i2(s.posicions[s.length].x-1,s.posicions[s.length].y-1));}

    for (int n = s.length-1; n>0; n--){
        if (s.posicions[n].x > s.posicions[n+1].x && s.posicions[n].y < s.posicions[n-1].y)
```

```

        {ps(&finestra_joc,&cosSOfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x > s.posicions[n-1].x && s.posicions[n].y < s.posicions[n+1].y)
        {ps(&finestra_joc,&cosSOfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x > s.posicions[n+1].x && s.posicions[n].y > s.posicions[n-1].y)
        {ps(&finestra_joc,&cosNOfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x > s.posicions[n-1].x && s.posicions[n].y > s.posicions[n+1].y)
        {ps(&finestra_joc,&cosNOfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x != s.posicions[n+1].x && s.posicions[n].y == s.posicions[n-1].y)
        {ps(&finestra_joc,&cosXfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x < s.posicions[n+1].x && s.posicions[n].y < s.posicions[n-1].y)
        {ps(&finestra_joc,&cosSEfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x < s.posicions[n-1].x && s.posicions[n].y < s.posicions[n+1].y)
        {ps(&finestra_joc,&cosSEfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x < s.posicions[n+1].x && s.posicions[n].y > s.posicions[n-1].y)
        {ps(&finestra_joc,&cosNEfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x < s.posicions[n-1].x && s.posicions[n].y > s.posicions[n+1].y)
        {ps(&finestra_joc,&cosNEfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    else if (s.posicions[n].x == s.posicions[n+1].x && s.posicions[n].y != s.posicions[n-1].y)
        {ps(&finestra_joc,&cosYfant,i2(s.posicions[n].x-1,s.posicions[n].y-1));}
    }
    /*
    if(s.posicions[s.length].x == s.posicions[s.length-1].x && s.posicions[s.length].y ==
s.posicions[s.length-1].y){
        if (s.dircua == 'N'){
ps(&finestra_joc,cuaNfant,{s.posicions[s.length].x-1,s.posicions[s.length].y-1});}
            else if (s.dircua == 'S'){
ps(&finestra_joc,cuaSfant,{s.posicions[s.length].x-1,s.posicions[s.length].y-1});}
            else if (s.dircua == 'E'){
ps(&finestra_joc,cuaEfant,{s.posicions[s.length].x-1,s.posicions[s.length].y-1});}
            else if (s.dircua == 'O'){
ps(&finestra_joc,cuaOfant,{s.posicions[s.length].x-1,s.posicions[s.length].y-1});}
        }
    */
    if (s.dirR == 'N'){ ps(&finestra_joc,&capNfant,i2(s.posicions[0].x-1,s.posicions[0].y-1));}
    if (s.dirR == 'S'){ ps(&finestra_joc,&capSfant,i2(s.posicions[0].x-1,s.posicions[0].y-1));}
    if (s.dirR == 'E'){ ps(&finestra_joc,&capEfant,i2(s.posicions[0].x-1,s.posicions[0].y-1));}
    if (s.dirR == 'O'){ ps(&finestra_joc,&capOfant,i2(s.posicions[0].x-1,s.posicions[0].y-1));}

    return 0;
}

int pintarSERP_CAPICUA(){
    for (int n = 1; n<s.length; n++){
        {ps(&finestra_joc,&cos_,s.posicions[n]);}

        if(p.m10.mode == 0){
            if (s.dirR == 'N'){ ps(&finestra_joc,&capN,s.posicions[0]);}
            if (s.dirR == 'S'){ ps(&finestra_joc,&capS,s.posicions[0]);}
            if (s.dirR == 'E'){ ps(&finestra_joc,&capE,s.posicions[0]);}
            if (s.dirR == 'O'){ ps(&finestra_joc,&capO,s.posicions[0]);}

            if (s.dircua == 'N'){ ps(&finestra_joc,&cuaN,s.posicions[s.length]);}
            if (s.dircua == 'S'){ ps(&finestra_joc,&cuaS,s.posicions[s.length]);}
            if (s.dircua == 'E'){ ps(&finestra_joc,&cuaE,s.posicions[s.length]);}
            if (s.dircua == 'O'){ ps(&finestra_joc,&cuaO,s.posicions[s.length]);}
        }
        else{
            if (s.dirR == 'N'){ ps(&finestra_joc,&cuaN,s.posicions[0]);}
            if (s.dirR == 'S'){ ps(&finestra_joc,&cuaS,s.posicions[0]);}
            if (s.dirR == 'E'){ ps(&finestra_joc,&cuaE,s.posicions[0]);}
            if (s.dirR == 'O'){ ps(&finestra_joc,&cuaO,s.posicions[0]);}

            if (s.dircua == 'N'){ ps(&finestra_joc,&capN,s.posicions[s.length]);}
            if (s.dircua == 'S'){ ps(&finestra_joc,&capS,s.posicions[s.length]);}
            if (s.dircua == 'E'){ ps(&finestra_joc,&capE,s.posicions[s.length]);}
            if (s.dircua == 'O'){ ps(&finestra_joc,&capO,s.posicions[s.length]);}
        }

        return 0;
    }
}

int pintarSERP_DIAGONAL(){
    for (int index = 0; index < p.m12.index12; index++){p.m12.posicions12[index] = i2(0,0);}
    p.m12.index12 = 0; p.m12.cap12[0] = p.m12.cap12[1] = i2(0,0);

    for (int n = s.length; n>0; n--){

```

```

ps(&finestra_joc,&cos_,s.posicions[n]);
if (s.posicions[n].x != s.posicions[n-1].x && s.posicions[n].y != s.posicions[n-1].y)
{
    if (s.posicions[n].y>s.posicions[n-1].y && s.posicions[n].x<s.posicions[n-1].x){
        ps(&finestra_joc,&diagonalSE,i2(s.posicions[n].x,s.posicions[n].y-5));
        ps(&finestra_joc,&diagonalNO,i2(s.posicions[n].x+5,s.posicions[n].y));
        if (n>1){p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x,s.posicions[n].y-5);
p.m12.index12 += 1;
        p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x+5,s.posicions[n].y);
p.m12.index12 += 1;}}
    else{
        p.m12.cap12[0] = i2(s.posicions[n].x,s.posicions[n].y-5);
        p.m12.cap12[1] = i2(s.posicions[n].x+5,s.posicions[n].y);}}
    else if (s.posicions[n].y<s.posicions[n-1].y && s.posicions[n].x<s.posicions[n-1].x){
        ps(&finestra_joc,&diagonalSO,i2(s.posicions[n].x+5,s.posicions[n].y));
        ps(&finestra_joc,&diagonalNE,i2(s.posicions[n].x,s.posicions[n].y+5));
        if (n>1){p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x+5,s.posicions[n].y);
p.m12.index12 += 1;
        p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x,s.posicions[n].y+5);
p.m12.index12 += 1;}}
    else{
        p.m12.cap12[0] = i2(s.posicions[n].x+5,s.posicions[n].y);
        p.m12.cap12[1] = i2(s.posicions[n].x,s.posicions[n].y+5);}}
    else if (s.posicions[n].y>s.posicions[n-1].y && s.posicions[n].x>s.posicions[n-1].x){
        ps(&finestra_joc,&diagonalSO,i2(s.posicions[n].x,s.posicions[n].y+5));
        ps(&finestra_joc,&diagonalNE,i2(s.posicions[n].x-5,s.posicions[n].y));
        if (n>1){p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x,s.posicions[n].y+5);
p.m12.index12 += 1;
        p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x-5,s.posicions[n].y);
p.m12.index12 += 1;}}
    else{
        p.m12.cap12[0] = i2(s.posicions[n].x,s.posicions[n].y-5);
        p.m12.cap12[1] = i2(s.posicions[n].x-5,s.posicions[n].y);}}
    else if (s.posicions[n].y<s.posicions[n-1].y && s.posicions[n].x > s.posicions[n-1].x){
        ps(&finestra_joc,&diagonalNO,i2(s.posicions[n].x,s.posicions[n].y+5));
        ps(&finestra_joc,&diagonalSE,i2(s.posicions[n].x-5,s.posicions[n].y));
        if (n>1){p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x,s.posicions[n].y+5);
p.m12.index12 += 1;
        p.m12.posicions12[p.m12.index12] = i2(s.posicions[n].x-5,s.posicions[n].y);
p.m12.index12 += 1;}}
    else{
        p.m12.cap12[0] = i2(s.posicions[n].x,s.posicions[n].y+5);
        p.m12.cap12[1] = i2(s.posicions[n].x-5,s.posicions[n].y);}}
    }
}

if (s.dirR == 'W'){ps(&finestra_joc,&capNE,s.posicions[0]);}
if (s.dirR == 'Q'){ps(&finestra_joc,&capNO,s.posicions[0]);}
if (s.dirR == 'D'){ps(&finestra_joc,&capSE,s.posicions[0]);}
if (s.dirR == 'A'){ps(&finestra_joc,&capSO,s.posicions[0]);}
if (s.dirR == 'N'){ps(&finestra_joc,&capN,s.posicions[0]);}
if (s.dirR == 'S'){ps(&finestra_joc,&capS,s.posicions[0]);}
if (s.dirR == 'E'){ps(&finestra_joc,&capE,s.posicions[0]);}
if (s.dirR == 'O'){ps(&finestra_joc,&capO,s.posicions[0]);}

ferforma(&finestra_joc,s.posicions[s.length],5,5,negre);
if (s.posicions[s.length].y>s.posicions[s.length-1].y &&
s.posicions[s.length].x<s.posicions[s.length-1].x){
    ps(&finestra_joc,&cuaSO,s.posicions[s.length]);}
    else if (s.posicions[s.length].y<s.posicions[s.length-1].y &&
s.posicions[s.length].x<s.posicions[s.length-1].x){
    ps(&finestra_joc,&cuaNO,s.posicions[s.length]);}
    else if (s.posicions[s.length].y>s.posicions[s.length-1].y &&
s.posicions[s.length].x>s.posicions[s.length-1].x){
    ps(&finestra_joc,&cuaSE,s.posicions[s.length]);}
    else if (s.posicions[s.length].y<s.posicions[s.length-1].y && s.posicions[s.length].x >
s.posicions[s.length-1].x){
    ps(&finestra_joc,&cuaNE,s.posicions[s.length]);}
    else if (s.dircua == 'N'){ ps(&finestra_joc,&cuaN,s.posicions[s.length]);}
    else if (s.dircua == 'S'){ ps(&finestra_joc,&cuaS,s.posicions[s.length]);}
    else if (s.dircua == 'E'){ ps(&finestra_joc,&cuaE,s.posicions[s.length]);}
    else if (s.dircua == 'O'){ ps(&finestra_joc,&cuaO,s.posicions[s.length]);}

return 0;

```

```

}

int pintarSERP_SMOOTH() {

    for (int n = 1; n<s.length; n++)
    {
        if(n%5==0) {
            ps(&finestra_joc,&cosB13,s.posicions[n]);}
        else{
            ps(&finestra_joc,&cosA13,s.posicions[n]);}
        }
    ps(&finestra_joc,&cosB13,s.posicions[s.length-1]);

    if (s.dirR == 'N'){ ps(&finestra_joc,&capN,s.posicions[0]);}
    if (s.dirR == 'S'){ ps(&finestra_joc,&capS,s.posicions[0]);}
    if (s.dirR == 'E'){ ps(&finestra_joc,&capE,s.posicions[0]);}
    if (s.dirR == 'O'){ ps(&finestra_joc,&capO,s.posicions[0]);}

    ps(&finestra_joc,&cercle,s.posicions[s.length]);

    return 0;
}

/*int pintarSERP_GALLINETA_16() {
    for (int cos = 0; cos<=s.length; cos++)
    {
        p.ml8.difX = abs(p.m0.R.x - s.posicions[cos].x) / 5;
        p.ml8.difY = abs(p.m0.R.y - s.posicions[cos].y) / 5;
        p.ml8.distancia = p.ml8.difX+p.ml8.difY;
        if(p.ml8.distancia > 16) {p.ml8.distancia=16;}

        U8 rgb = 255-(255/16)*(p.ml8.distancia-1);
        Pixel color = {rgb,rgb,rgb};
        ferforma(&finestra_joc, s.posicions[cos],5,5,color);
    }
    if (s.dirR == 'N'){ ps(&finestra_joc,puntaN,s.cap);}
    else if (s.dirR == 'S'){ ps(&finestra_joc,puntaS,s.cap);}
    else if (s.dirR == 'E'){ ps(&finestra_joc,puntaE,s.cap);}
    else if (s.dirR == 'O'){ ps(&finestra_joc,puntaO,s.cap);}

    if (s.dirCua == 'N'){ ps(&finestra_joc,puntaN,s.posicions[s.length]);}
    else if (s.dirCua == 'S'){ ps(&finestra_joc,puntaS,s.posicions[s.length]);}
    else if (s.dirCua == 'E'){ ps(&finestra_joc,puntaE,s.posicions[s.length]);}
    else if (s.dirCua == 'O'){ ps(&finestra_joc,puntaO,s.posicions[s.length]);}

    return 0;
}*/

int pintarSERP_GALLINETA() {
    for (int cos = 0; cos<=s.length; cos++)
    {
        p.ml8.difX = abs(p.m0.R.x - s.posicions[cos].x) / 5;
        p.ml8.difY = abs(p.m0.R.y - s.posicions[cos].y) / 5;
        p.ml8.distancia = p.ml8.difX+p.ml8.difY;

        ferforma_mode(&finestra_joc,s.posicions[cos],5,5,negre,1000 +
        (p.ml8.distancia-1)*(p.ml8.distancia+5));
    }

    if (s.dirR == 'N'){ ps(&finestra_joc,&puntaN,s.cap);}
    if (s.dirR == 'S'){ ps(&finestra_joc,&puntaS,s.cap);}
    if (s.dirR == 'E'){ ps(&finestra_joc,&puntaE,s.cap);}
    if (s.dirR == 'O'){ ps(&finestra_joc,&puntaO,s.cap);}

    if (s.dirCua == 'N'){ ps(&finestra_joc,&puntaN,s.posicions[s.length]);}
    if (s.dirCua == 'S'){ ps(&finestra_joc,&puntaS,s.posicions[s.length]);}
    if (s.dirCua == 'E'){ ps(&finestra_joc,&puntaE,s.posicions[s.length]);}
    if (s.dirCua == 'O'){ ps(&finestra_joc,&puntaO,s.posicions[s.length]);}

    return 0;
}

int pintarSERP_1D() {

```

```

pintarSERP_DIAGONAL();
if (s.dirR == 'N')
{
    ferforma(&finestra_joc,s.posicions[0],5,5,negre);
    ferforma(&finestra_joc,i2(s.posicions[0].x+3,s.posicions[0].y+1),1,5,negre);
    ps(&finestra_joc,&capN,i2(s.posicions[0].x-2,s.posicions[0].y));
}
else if (s.dirR == 'S')
{
    ferforma(&finestra_joc,s.posicions[0],5,5,negre);
    ferforma(&finestra_joc,i2(s.posicions[0].x+3,s.posicions[0].y-1),1,5,negre);
    ps(&finestra_joc,&capS,i2(s.posicions[0].x-2,s.posicions[0].y));
}
else
{ps(&finestra_joc,&capE,s.posicions[0]);}

return 0;
}

int pintarSERP_RETRO(){
    for (int n = 0; n<=s.length; n++)
        {ferforma(&finestra_joc,s.posicions[n],2,2,blanc);}

    return 0;
}

int pintarhoTOT(int mode){
    if (mode == 0){
        if (p.m0.HS0 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m0.HS0 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m0.HS0, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 1){
        if (p.m1.HS1 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m1.HS1 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m1.HS1, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 2){
        if (p.m2.HS2 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m2.HS2 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m2.HS2, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 3){
        if (p.m3.HS3 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m3.HS3 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m3.HS3, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);
        int pinto_la_mira = 1;
        if (p.m3.mira.x == p.m0.R.x && p.m3.mira.y == p.m0.R.y){pinto_la_mira = 0;}
        else{
            for (int n = 0; n<=s.length; n+=1){
                if (s.posicions[n].x == p.m3.mira.x && s.posicions[n].y == p.m3.mira.y){
                    pinto_la_mira = 0; break;}
            }
        }
        if(p.m3.ordre_donada == 1){
            if (p.m3.ordre.x == p.m0.R.x && p.m3.ordre.y ==
p.m0.R.y){ferforma(&finestra_joc,p.m0.R,5,5,negre); ps(&finestra_joc,&rataN,p.m0.R);}
            else{ps(&finestra_joc,&creuC,p.m3.ordre);}
        }
        if(pinto_la_mira == 1){ps(&finestra_joc,&mira,p.m3.mira);}
        else{ferforma_mode(&finestra_joc,p.m3.mira,5,5,blanc,2);}
    }
    if (mode == 4){
        if (p.m4.HS4 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m4.HS4 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m4.HS4, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 5){
        if (p.m5.HS5 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m5.HS5 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m5.HS5, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP_FANTASMA(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 6){
        if (pintarHUD_mode(240-p.m6.parets-p.m6.NORATAs,p.m6.parets+p.m6.NORATAs,6) == 1){return 1;}
        //ps(finestra_general, marc, {13,1});
        for (int i = 0; i<240; i++){
            if (p.m6.posicionsPINTAR[i] == 'b'){ferforma(&finestra_joc,posicionsXY(i),5,5,blanc);}
            else if (p.m6.posicionsPINTAR[i] == 'r'){ps(&finestra_joc,&rataNEG,posicionsXY(i));}
        }
        if(p.m6.dirOri == 'N'){ps(&finestra_joc,&Nord,p.m6.origen);}
    }
}

```

```

        else if(p.m6.dirOri == 'E'){ps(&finestra_joc,&Est,p.m6.origen);}
        else if(p.m6.dirOri == 'S'){ps(&finestra_joc,&Sud,p.m6.origen);}
        else if(p.m6.dirOri == 'O'){ps(&finestra_joc,&Oest,p.m6.origen);}
        ps_mode(&finestra_joc, &cercleNEG, s.cap, 2);}
    if (mode == 66){
        if (p.m6.HS6 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m6.HS6 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m6.HS6, p.m0.puntuacio*5) == 1){return 1;}
        for (int i = 0; i<240; i++){if(p.m6.posicionsPINTAR[i] ==
'b'){ferforma(&finestra_joc,posicionsXY(i),5,5,blanc);}}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 7){
        if (p.m7.HS7 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m7.HS7 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m7.HS7, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 8){
        if (p.m8.HS8 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m8.HS8 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m8.HS8, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 9){
        if (p.m9.HS9 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m9.HS9 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m9.HS9, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 10){
        if (p.m10.HS10 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m10.HS10 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m10.HS10, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP_CAPICUA(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 11){
        if (p.m11.HS11 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m11.HS11 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m11.HS11, p.m0.puntuacio*5) == 1){return 1;}
        if (p.m11.pitfora == 1){ps(&finestra_joc,&cercle,p.m11.P);}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 12){
        if (p.m12.HS12 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m12.HS12 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m12.HS12, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP_DIAGONAL(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 13){
        if (p.m13.HS13 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m13.HS13 = p.m0.puntuacio*5;}
        if (pintarHUD_mode(p.m13.HS13, p.m0.puntuacio*5, 13) == 1){return 1;}
        //for(int i = 0; i < p.m13.NUMcasellesocupades;
i++){ps(&finestra_joc,&rata,p.m13.casellesocupades[i]);}
        pintarSERP_SMOOTH(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 14){
        if (p.m14.HS14 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m14.HS14 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m14.HS14, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP();
        for(int r=0; r<5; r+=1){
            if (p.m14.rates[r].caracter == 1){ps(&finestra_joc,&rata,p.m14.rates[r].origen);}
            else{ps(&finestra_joc,&rataFALSA,p.m14.rates[r].origen);}
        }
    }
    if (mode == 15){
        if (p.m15.HS15 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m15.HS15 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m15.HS15, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP();
        if (p.m15.dirRATA == 'N'){ps(&finestra_joc,&rataS,p.m0.R);}
        if (p.m15.dirRATA == 'S'){ps(&finestra_joc,&rataN,p.m0.R);}
        if (p.m15.dirRATA == 'E'){ps(&finestra_joc,&rataO,p.m0.R);}
        if (p.m15.dirRATA == 'O'){ps(&finestra_joc,&rataE,p.m0.R);}
    }
    if (mode == 16){
        if (p.m16.HS16 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m16.HS16 = p.m0.puntuacio*5;}
        if (pintarHUD(p.m16.HS16, p.m0.puntuacio*5) == 1){return 1;}
        pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
    if (mode == 17 || mode == 177){
        if (mode == 17) {
            if (p.m17.HS17 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m17.HS17 =
p.m0.puntuacio*5;}
            if (pintarHUD_mode(p.m17.HS17, p.m0.puntuacio*5, 17) == 1){return 1;}
            //ps(finestra_general, marc, {13,1});
            pintarSERP();
            int rataalavista = 0;
            I2 top_left = {s.cap.x+p.m17.sumes[0],s.cap.y+p.m17.sumes[0]};
            I2 bottom_right = {s.cap.x+p.m17.sumes[4],s.cap.y+p.m17.sumes[4]};
            for (int index = 0; index<240; index++){
                I2 i = posicionsXY(index);

```

```

        if (i.x < top_left.x || i.y < top_left.y || i.x > bottom_right.x || i.y >
bottom_right.y)
            {ferforma(&finestra_joc, i,5,5,blanc);}
            else{if (i.x == p.m0.R.x && i.y == p.m0.R.y) {rataalavista = 1;}}
        }
        ps(&finestra_joc,&rata,p.m0.R);
        if (rataalavista == 0){ferforma_mode(&finestra_joc,p.m0.R,5,5,negre,2);}
        ps(&finestra_joc,&focus,i2(s.cap.x+p.m17.sumes[0],s.cap.y+p.m17.sumes[0]));
    }
    else{pintarHUD(p.m17.HS17, p.m0.puntuacio*5); pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
}
if (mode == 18){
    if (p.m18.HS18 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m18.HS18 = p.m0.puntuacio*5;}
    if (pintarHUD(p.m18.HS18, p.m0.puntuacio*5) == 1){return 1;}
    pintarSERP_GALLINETA();
    if(p.m18.flash >= 1){//GUARRADA NECESSARIA FLAIX 18
        if(p.m18.flash<=5){p.m18.flash++;} else{p.m18.flash = 0;}
        ferforma(&finestra_joc,i2(14,2),100,60,blanc); ps(&finestra_joc,&rataNEG,p.m0.R);}
}
if (mode == 19){
    if (p.m19.HS19 > p.m0.puntuacio){p.m19.HS19 = p.m0.puntuacio;}
    if (pintarHUD(p.m19.HS19, p.m0.puntuacio) == 1){return 1;}
    pintarSERP();
}
if(mode == 20){
    if (p.m20.HS20 < t){p.m20.HS20 = t;} if (pintarHUD(p.m20.HS20, t) == 1){return 1;}
    pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
if (mode == 21){
    if (p.m21.HS21 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m21.HS21 = p.m0.puntuacio*5;}
    if (pintarHUD(p.m21.HS21, p.m0.puntuacio*5) == 1){return 1;}
    ps(&finestra_joc,&cercle,p.m21.forat); pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
if (mode == 22){
    if (p.m22.HS22 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m22.HS22 = p.m0.puntuacio*5;}
    if (pintarHUD(p.m22.HS22, p.m0.puntuacio*5) == 1){return 1;}
    pintarSERP(); ps(&finestra_joc,&rata,p.m0.R);}
if(mode == 23){
    if (p.m23.HS23 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m23.HS23 = p.m0.puntuacio*5;}
    if (pintarHUD(p.m23.HS23, p.m0.puntuacio*5) == 1){return 1;}
    pintarSERP_1D(); ps(&finestra_joc,&rata,p.m0.R);}
if(mode == 24){
    if (p.m24.HS24 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m24.HS24 = p.m0.puntuacio*5;}
    if (pintarHUD(p.m24.HS24, p.m0.puntuacio*5) == 1){return 1;}
    pintarSERP_FANTASMA(); ps(&finestra_joc,&rata,p.m24.prota);
ps(&finestra_joc,&rataNEG,p.m0.R);}
if (mode == 25){
    if (p.m25.HS25 < p.m0.puntuacio*5 && p.m0.puntuacio*5 < 999){p.m25.HS25 = p.m0.puntuacio*5;}
    if (pintarHUD_mode(p.m25.HS25, p.m0.puntuacio*5, 25) == 1){return 1;}
    pintarSERP_RETRO(); ferforma(&finestra_joc,p.m0.R,2,2,blanc);}

return 0;
}

int gestionarINPUT(){
    if (controller.left.pressed && s.dirR != 'E'){s.dir = 'O';}
    if (controller.right.pressed && s.dirR != 'O'){s.dir = 'E';}
    if (controller.down.pressed && s.dirR != 'N'){s.dir = 'S';}
    if (controller.up.pressed && s.dirR != 'S'){s.dir = 'N';}
    return 0;
}

int gestionarINPUT_INVERS(){
    if (controller.right.pressed && s.dirR != 'E'){s.dir = 'O';}
    if (controller.left.pressed && s.dirR != 'O'){s.dir = 'E';}
    if (controller.down.pressed && s.dirR != 'S'){s.dir = 'N';}
    if (controller.up.pressed && s.dirR != 'N'){s.dir = 'S';}
    return 0;
}

int gestionarINPUT_EGO(){
    if (controller.left.half_transition_count && controller.left.pressed && p.m2.puc_girar == 1){
        p.m2.puc_girar = 0;
        p.m2.index_dir -= 1;
        if (p.m2.index_dir < 0){p.m2.index_dir = 3;}
    }
    if (controller.right.half_transition_count && controller.right.pressed && p.m2.puc_girar == 1){
        p.m2.puc_girar = 0;
        p.m2.index_dir += 1;
    }
}

```



```

        if (p.m2.index_dir > 3){p.m2.index_dir = 0;}}
        s.dir = p.m2.direcciones[p.m2.index_dir];
        return 0;
    }

    int gestionarINPUT_STRA(){

        if (controller.left.pressed){p.m3.dir_mira='O';}
        if (controller.right.pressed){p.m3.dir_mira='E';};
        if (controller.down.pressed){p.m3.dir_mira='S';}
        if (controller.up.pressed){p.m3.dir_mira='N';}

        if (controller.A.pressed && controller.A.half_transition_count){p.m3.ordre = p.m3.mira;
p.m3.ordre_donada = 1;}
        if (controller.B.pressed && controller.B.half_transition_count){p.m3.ordre_donada = 0;}
        return 0;
    }

    int gestionarINPUT_PINTAR(){

        if (controller.left.pressed){s.dir = 'O';}
        if (controller.right.pressed){s.dir = 'E';}
        if (controller.down.pressed){s.dir = 'S';}
        if (controller.up.pressed){s.dir = 'N';}

        if(p.m6.acabemdecomencar == 0){
            if (controller.A.pressed && controller.B.pressed == true){p.m6.colorBN = 'r'; p.m6.HS6 = 0;}
            else if (controller.A.pressed){p.m6.colorBN = 'b'; p.m6.HS6 = 0;}
            else if (controller.B.pressed){p.m6.colorBN = 'n'; p.m6.HS6 = 0;}
            else{p.m6.colorBN = 'i';}
        }
        else{
            if (controller.A.half_transition_count){p.m6.acabemdecomencar = 0;}}
        return 0;
    }

    int gestionarINPUT_MANUAL(){
        s.dir = ' ';
        if (controller.left.pressed && controller.left.half_transition_count && s.dirR != 'E'){s.dir =
'O';}
        if (controller.right.pressed && controller.right.half_transition_count && s.dirR != 'O'){s.dir =
'E';}
        if (controller.down.pressed && controller.down.half_transition_count && s.dirR != 'N'){s.dir =
'S';}
        if (controller.up.pressed && controller.up.half_transition_count && s.dirR != 'S'){s.dir = 'N';}
        return 0;
    }

    int capicua(){

        for(int pos = 0; pos <=s.length; pos++){p.m10.posicionsaux[pos] = s.posicions[s.length-pos];}
        for(int pos = 0; pos <=s.length; pos++){s.posicions[pos] = p.m10.posicionsaux[pos];}
        s.cap.x = s.posicions[0].x; s.cap.y = s.posicions[0].y;

        if(p.m10.mode==0){p.m10.mode=1;} else{p.m10.mode=0;}
        p.m10.diraux = s.dircua;
        s.dircua = s.dirR;
        s.dir = p.m10.diraux;
        s.dirR = s.dir;

        return 0;
    }

    int gestionarINPUT_CAPICUA(){

        if (controller.left.pressed && s.dirR != 'E' && controller.left.half_transition_count){s.dir =
'O';}
        if (controller.right.pressed && s.dirR != 'O' && controller.right.half_transition_count){s.dir =
'E';}
        if (controller.down.pressed && s.dirR != 'N' && controller.down.half_transition_count){s.dir =
'S';}
        if (controller.up.pressed && s.dirR != 'S' && controller.up.half_transition_count){s.dir = 'N';}

        if (controller.left.pressed && controller.left.half_transition_count){

```

```

        if(s.dirR != 'E'){s.dir = 'O';}else{capicua();}}
    if (controller.right.pressed && controller.right.half_transition_count){
        if(s.dirR != 'O'){s.dir = 'E';}else{capicua();}}
    if (controller.down.pressed && controller.down.half_transition_count){
        if(s.dirR != 'N'){s.dir = 'S';}else{capicua();}}
    if (controller.up.pressed && controller.up.half_transition_count){
        if(s.dirR != 'S'){s.dir = 'N';}else{capicua();}}
    if (controller.A.pressed && controller.A.half_transition_count){capicua();}
    return 0;
}

//TODO: SEGUIR BUSCANT ELSE IFs
int gestionarINPUT_REMOT(){
    int shadefer = 0;

    if (controller.left.pressed && s.dirR != 'E'){s.dir = 'O';}
    if (controller.right.pressed && s.dirR != 'O'){s.dir = 'E';}
    if (controller.down.pressed && s.dirR != 'N'){s.dir = 'S';}
    if (controller.up.pressed && s.dirR != 'S'){s.dir = 'N';}

    if (controller.up.pressed && controller.up.half_transition_count && s.dirR != 'S'){
        if (s.dirR != 'N'){s.dir = 'N';}
        else{if (p.m11.pitfora == 0){shadefer = 1;}}}
    else if (controller.down.pressed && controller.down.half_transition_count && s.dirR != 'N'){
        if(s.dirR != 'S'){s.dir = 'S';}
        else{if (p.m11.pitfora == 0){shadefer = 1;}}}
    else if (controller.left.pressed && controller.left.half_transition_count && s.dirR != 'E'){
        if(s.dirR != 'O'){s.dir = 'O';}
        else{if (p.m11.pitfora == 0){shadefer = 1;}}}
    else if (controller.right.pressed && controller.right.half_transition_count && s.dirR != 'O'){
        if(s.dirR != 'E'){s.dir = 'E';}
        else{if (p.m11.pitfora == 0){shadefer = 1;}}}
    if (controller.A.pressed && controller.A.half_transition_count){if (p.m11.pitfora == 0){shadefer
= 1;}}

    if(shadefer == 1 && s.length >1){p.m11.pitfora = 1; p.m11.direcciopit = s.dirR; s.length -= 1;}
    return 0;
}

int gestionarINPUT_DIAGONAL(){
    if (controller.up.pressed){p.m12.dir1 = 'N';}
    else if (controller.down.pressed){p.m12.dir1 = 'S';}
    else{p.m12.dir1 = ' ';}

    if (controller.left.pressed){p.m12.dir2 = 'O';}
    else if (controller.right.pressed){p.m12.dir2 = 'E';}
    else{p.m12.dir2 = ' ';}
    return 0;
}

int gestionarINPUT_1D(){
    if (controller.down.pressed){s.dir = 'S';}
    if (controller.up.pressed){s.dir = 'N';}
    if (controller.right.pressed){s.dir = 'E';}
    return 0;
}

int gestionarINPUT_IA(){
    if (controller.left.pressed){p.m24.dirRATA = 'O';}
    if (controller.right.pressed){p.m24.dirRATA = 'E';}
    if (controller.down.pressed){p.m24.dirRATA = 'S';}
    if (controller.up.pressed){p.m24.dirRATA = 'N';}
    return 0;
}

int ferCREIXERoDECREIXER_mode(int mode /*= 0 */){
    if (mode == 0){mode = s.creixer;}

    if (mode == 1){
        s.length += 1; s.posicions[s.length] = s.posicions[s.length-1];}

    if (mode == -1){s.length -= 1;}
}

```

```

        s.creixer = 0;
        return 0;
    }

    int ferCREIXERoDECREIXER() {
        return ferCREIXERoDECREIXER_mode(0);
    }

    int ferCREIXERoDECREIXER_1D() {
        if (s.creixer == 1)
        {
            p.m0.puntuacio += 1; s.creixer = 0;
            if(s.length < 8)
            {
                s.length += 1;
                s.posicions[s.length] = i2(s.posicions[s.length-1].x-5, s.posicions[s.length-1].y);
            }
        }
        return 0;
    }

    int moureSERP_dir_mode(char dir_obligada /*= 'X'*/, int mode /*= 0 */){
        I2 posicio_al_inici;
        posicio_al_inici = s.cap;
        char direccio_a_moures; direccio_a_moures = s.dir;
        if (dir_obligada != 'X'){direccio_a_moures = dir_obligada;}
        //printf("%i ",s.lloc);
        if (direccio_a_moures == 'N'){s.cap.y -= 5; s.dirR = 'N'; s.lloc -= 1;}
        if (direccio_a_moures == 'S'){s.cap.y += 5; s.dirR = 'S'; s.lloc += 1;}
        if (direccio_a_moures == 'O'){s.cap.x -= 5; s.dirR = 'O'; s.lloc -= 12;}
        if (direccio_a_moures == 'E'){s.cap.x += 5; s.dirR = 'E'; s.lloc += 12;}
        //printf("%i\n",s.lloc);

        //Detectar sortida dels limits
        if (s.cap.x < 14 || s.cap.x > 109 || s.cap.y < 2 || s.cap.y > 57)
        {s.cap = posicio_al_inici; return 1;}
        if (mode == 66 && p.m6.posicionsPINTAR[s.lloc] == 'b')
        {s.cap = posicio_al_inici; return 1;}

        //Detectar trompada amb ell mateix
        for (int n = 1; n<s.length; n+=1)
        {
            if (s.cap.x == s.posicions[n].x && s.cap.y == s.posicions[n].y)
            {s.cap = posicio_al_inici; return 2;}
        }

        //Moure les posicions
        for (int n = s.length; n>0; n-=1)
        {s.posicions[n] = s.posicions [n-1];}
        s.posicions[0].x = s.cap.x;
        s.posicions[0].y = s.cap.y;

        //Canviar direcció de la cua
        if (s.posicions[s.length].x > s.posicions[s.length-1].x)
        {s.dircua = 'E';}
        if (s.posicions[s.length].x < s.posicions[s.length-1].x)
        {s.dircua = 'O';}
        if (s.posicions[s.length].y > s.posicions[s.length-1].y)
        {s.dircua = 'S';}
        if (s.posicions[s.length].y < s.posicions[s.length-1].y)
        {s.dircua = 'N';}

        return 0;
    }

    int moureSERP_dir(char dir_obligada){
        return moureSERP_dir_mode(dir_obligada, 0);
    }

    int moureSERP(){
        return moureSERP_dir_mode('X', 0);
    }

```

```

int moureSERP_VORES() {

    I2 posicio_al_inici = s.cap;

    //printf("%i ",s.lloc);
    if (s.dir == 'N'){s.cap.y -= 5; s.dirR = 'N'; s.lloc -= 1;}
    if (s.dir == 'S'){s.cap.y += 5; s.dirR = 'S'; s.lloc += 1;}
    if (s.dir == 'O'){s.cap.x -= 5; s.dirR = 'O'; s.lloc -= 12;}
    if (s.dir == 'E'){s.cap.x += 5; s.dirR = 'E'; s.lloc += 12;}
    //printf("%i\n",s.lloc);

    //Detectar sortida dels limits
    if (s.cap.x < 14) {s.cap.x = 109; s.lloc+=20*12;}// s.length-=1;}
    if (s.cap.x > 109) {s.cap.x = 14; s.lloc-=20*12;}// s.length-=1;}
    if (s.cap.y < 2) {s.cap.y = 57; s.lloc+=12;}// s.length-=1;}
    if (s.cap.y > 57) {s.cap.y = 2; s.lloc-=12;}// s.length-=1;}

    //Detectar trompada amb ell mateix
    for (int n = 1; n<s.length; n+=1)
    {
        if (s.cap.x == s.posicions[n].x && s.cap.y == s.posicions[n].y)
            {s.cap = posicio_al_inici; return 2;}
    }

    //Moure les posicions
    for (int n = s.length; n>0; n-=1)
    {s.posicions[n] = s.posicions [n-1];}
    s.posicions[0].x = s.cap.x;
    s.posicions[0].y = s.cap.y;

    //Canviar direcció de la cua
    if (s.posicions[s.length].x > s.posicions[s.length-1].x)
    {s.dircua = 'E';}
    if (s.posicions[s.length].x < s.posicions[s.length-1].x)
    {s.dircua = 'O';}
    if (s.posicions[s.length].y > s.posicions[s.length-1].y)
    {s.dircua = 'S';}
    if (s.posicions[s.length].y < s.posicions[s.length-1].y)
    {s.dircua = 'N';}

    if (s.posicions[s.length].x == 109 && s.posicions[s.length-1].x == 14) {s.dircua = 'O';}
    if (s.posicions[s.length].x == 14 && s.posicions[s.length-1].x == 109) {s.dircua = 'E';}
    if (s.posicions[s.length].y == 57 && s.posicions[s.length-1].y == 2) {s.dircua = 'N';}
    if (s.posicions[s.length].y == 2 && s.posicions[s.length-1].y == 57) {s.dircua = 'S';}

    return 0;
}

int moureSERP_FANTASMA_dir_mode(char dir_obligada /*= 'X' */, char mode /*= 0*/){

    I2 posicio_al_inici;
    posicio_al_inici = s.cap;
    char direccio_a_moures; direccio_a_moures = s.dir;
    if (dir_obligada != 'X'){direccio_a_moures = dir_obligada;}
    //printf("%i ",s.lloc);
    if (direccio_a_moures == 'N'){s.cap.y -= 5; s.dirR = 'N'; s.lloc -= 20;}
    if (direccio_a_moures == 'S'){s.cap.y += 5; s.dirR = 'S'; s.lloc += 20;}
    if (direccio_a_moures == 'O'){s.cap.x -= 5; s.dirR = 'O'; s.lloc -= 1;}
    if (direccio_a_moures == 'E'){s.cap.x += 5; s.dirR = 'E'; s.lloc += 1;}
    //printf("%i\n",s.lloc);

    //Detectar sortida dels limits
    if (s.cap.x < 14 || s.cap.x > 109|| s.cap.y < 2 || s.cap.y > 57)
    {return 1;}

    //Moure les posicions
    for (int n = s.length; n>0; n-=1)
    {s.posicions[n] = s.posicions [n-1];}
    s.posicions[0].x = s.cap.x;
    s.posicions[0].y = s.cap.y;

    if (s.length >=5 && mode == 0){
        for (int n = 3; n<s.length; n+=1){
            if (s.posicions[0].x == s.posicions[n].x && s.posicions[0].y == s.posicions[n].y){

```

```

        if((s.posicions[1].x == s.posicions[n-1].x && s.posicions[1].y ==
s.posicions[n-1].y)||
        (s.posicions[1].x == s.posicions[n+1].x && s.posicions[1].y ==
s.posicions[n+1].y)){return 2;}
    }
}

//Canviar direcció de la cua
if (s.posicions[s.length].x > s.posicions[s.length-1].x)
{s.dircua = 'E';}
if (s.posicions[s.length].x < s.posicions[s.length-1].x)
{s.dircua = 'O';}
if (s.posicions[s.length].y > s.posicions[s.length-1].y)
{s.dircua = 'S';}
if (s.posicions[s.length].y < s.posicions[s.length-1].y)
{s.dircua = 'N';}

return 0;
}

int moureSERP_FANTASMA(){
    return moureSERP_FANTASMA_dir_mode('X', 0);
}

//TODO: BUSCAR ELSE IFS
int moureSERP_DIAGONAL(){

    //TODO: (o no) No permetre que les direccions diagonals siguin accessibles si no s'estan pulsant
    les tecles corresponents
    if (p.m12.dir1 == 'N'){
        if (p.m12.dir2 == 'E' && s.dirR != 'A'){s.dirR = 'W';}
        else if (p.m12.dir2 == 'O' && s.dirR != 'D'){s.dirR = 'Q';}
        else{if(s.dirR != 'S'){s.dirR = 'N';}}
    }
    else if (p.m12.dir1 == 'S'){
        if (p.m12.dir2 == 'E' && s.dirR != 'Q'){s.dirR = 'D';}
        else if (p.m12.dir2 == 'O' && s.dirR != 'W'){s.dirR = 'A';}
        else{if(s.dirR != 'N'){s.dirR = 'S';}}
    }
    else{
        if (p.m12.dir2 == 'E' && s.dirR != 'O'){s.dirR = 'E';}
        else if (p.m12.dir2 == 'O' && s.dirR != 'E'){s.dirR = 'O';}
    }

    if (s.dirR == 'W'){s.cap.y -= 5; s.cap.x += 5;}
    if (s.dirR == 'Q'){s.cap.y -= 5; s.cap.x -= 5;}
    if (s.dirR == 'D'){s.cap.y += 5; s.cap.x += 5;}
    if (s.dirR == 'A'){s.cap.y += 5; s.cap.x -= 5;}
    if (s.dirR == 'N'){s.cap.y -= 5;}
    if (s.dirR == 'S'){s.cap.y += 5;}
    if (s.dirR == 'E'){s.cap.x += 5;}
    if (s.dirR == 'O'){s.cap.x -= 5;}

    if (s.cap.x < 14 || s.cap.x > 109|| s.cap.y < 2 || s.cap.y > 57)
    {return 1;}//{printf("Sóc fora del ring"); return 1;}

    //Detectar trompada amb ell mateix
    for (int n = 1; n<s.length; n+=1)
    {
        if (s.cap.x == s.posicions[n].x && s.cap.y == s.posicions[n].y)
        {return 2;}//{printf("He xocat"); return 1;}
    }
    for (int i = 0; i<p.m12.index12; i+=1)
    {
        if (s.cap.x == p.m12.posicions12[i].x && s.cap.y == p.m12.posicions12[i].y)
        {return 2;}//{printf("He xocat"); return 1;}
    }

    //Moure les posicions
    for (int n = s.length; n>0; n-=1)
    {s.posicions[n] = s.posicions [n-1];}
    s.posicions[0].x = s.cap.x;
    s.posicions[0].y = s.cap.y;
}

```

```

//Canviar direcció de la cua
if (s.posicions[s.length].x > s.posicions[s.length-1].x)
{s.dircua = 'E';}
if (s.posicions[s.length].x < s.posicions[s.length-1].x)
{s.dircua = 'O';}
if (s.posicions[s.length].y > s.posicions[s.length-1].y)
{s.dircua = 'S';}
if (s.posicions[s.length].y < s.posicions[s.length-1].y)
{s.dircua = 'N';}

return 0;
}

int moureSERP_SMOOTH() {

I2 posicio_al_inici = s.cap;
//printf("%i ",s.lloc);
if (s.dir == 'N'){s.cap.y -= 1; s.dirR = 'N'; s.lloc -= 1;}
if (s.dir == 'S'){s.cap.y += 1; s.dirR = 'S'; s.lloc += 1;}
if (s.dir == 'O'){s.cap.x -= 1; s.dirR = 'O'; s.lloc -= 12;}
if (s.dir == 'E'){s.cap.x += 1; s.dirR = 'E'; s.lloc += 12;}
//printf("%i\n",s.lloc);

//Detectar sortida dels limits
if (s.cap.x < 14 || s.cap.x > 109 || s.cap.y < 2 || s.cap.y > 57)
{s.cap = posicio_al_inici; return 1;}
//Detectar trompada amb ell mateix
for (int n = 8; n<s.length; n+=1)
{
    for(int x = -4; x < 5; x++){
        for(int y = -4; y < 5; y++){
            if (s.posicions[n].x+x == s.cap.x && s.posicions[n].y+y == s.cap.y){return 2;}
        }
    }
}

//Moure les posicions
for (int n = s.length; n>0; n-=1)
{s.posicions[n] = s.posicions [n-1];}
s.posicions[0].x = s.cap.x;
s.posicions[0].y = s.cap.y;

//Canviar direcció de la cua
if (s.posicions[s.length].x > s.posicions[s.length-1].x)
{s.dircua = 'E';}
if (s.posicions[s.length].x < s.posicions[s.length-1].x)
{s.dircua = 'O';}
if (s.posicions[s.length].y > s.posicions[s.length-1].y)
{s.dircua = 'S';}
if (s.posicions[s.length].y < s.posicions[s.length-1].y)
{s.dircua = 'N';}

return 0;
}

int moureSERP_1D() {

I2 posicio_al_inici = s.cap;
//printf("%i ",s.lloc);
if (s.dir == 'N'){s.cap.y -= 5; s.dirR = 'N'; s.lloc -= 1;}
if (s.dir == 'S'){s.cap.y += 5; s.dirR = 'S'; s.lloc += 1;}
if (s.dir == 'E'){s.dirR = 'E';}

//Detectar sortida dels limits
if (s.cap.y < 2 || s.cap.y > 57)
{s.cap = posicio_al_inici; return 1;}

//Detectar trompada amb una rata
for (int n = s.length; n>0; n-=1)
{
    if (s.posicions[n].x-5 <= p.m0.R.x && s.posicions[n].x >= p.m0.R.x && s.posicions[n].y ==
p.m0.R.y){return 2;}
    s.posicions[n].y = s.posicions[n-1].y;
}
}

```

```

    }
    s.posicions[0].x = s.cap.x;
    s.posicions[0].y = s.cap.y;

    //Canviar direcció de la cua
    if (s.posicions[s.length].x > s.posicions[s.length-1].x)
    {s.dircua = 'E';}
    if (s.posicions[s.length].x < s.posicions[s.length-1].x)
    {s.dircua = 'O';}
    if (s.posicions[s.length].y > s.posicions[s.length-1].y)
    {s.dircua = 'S';}
    if (s.posicions[s.length].y < s.posicions[s.length-1].y)
    {s.dircua = 'N';}
    return 0;
}

int moureSERP_RETRO(){

    I2 posicio_al_inici = s.cap;
    //printf("%i ",s.lloc);
    if (s.dir == 'N'){s.cap.y -= 3; s.dirR = 'N'; s.lloc -= 1;}
    if (s.dir == 'S'){s.cap.y += 3; s.dirR = 'S'; s.lloc += 1;}
    if (s.dir == 'O'){s.cap.x -= 3; s.dirR = 'O'; s.lloc -= 12;}
    if (s.dir == 'E'){s.cap.x += 3; s.dirR = 'E'; s.lloc += 12;}
    //printf("%i\n",s.lloc);

    //Detectar sortida dels limits
    if (s.cap.x < 15 || s.cap.x > 111 || s.cap.y < 3 || s.cap.y > 60)
        {s.cap = posicio_al_inici; return 1;}
    //Detectar trompada amb ell mateix
    for (int n = 1; n<s.length; n+=1)
    {
        if (s.cap.x == s.posicions[n].x && s.cap.y == s.posicions[n].y)
            {s.cap = posicio_al_inici; return 2;}
    }

    //Moure les posicions
    for (int n = s.length; n>0; n-=1)
    {s.posicions[n] = s.posicions [n-1];}
    s.posicions[0].x = s.cap.x;
    s.posicions[0].y = s.cap.y;

    //Canviar direcció de la cua
    if (s.posicions[s.length].x > s.posicions[s.length-1].x)
    {s.dircua = 'E';}
    if (s.posicions[s.length].x < s.posicions[s.length-1].x)
    {s.dircua = 'O';}
    if (s.posicions[s.length].y > s.posicions[s.length-1].y)
    {s.dircua = 'S';}
    if (s.posicions[s.length].y < s.posicions[s.length-1].y)
    {s.dircua = 'N';}

    return 0;
}

int moureSERP_FINAL(){
    for (int n = s.length; n>0; n-=1)
    {s.posicions[n] = s.posicions[n-1];}

    //Canviar direcció de la cua
    if (s.posicions[s.length].x > s.posicions[s.length-1].x)
    {s.dircua = 'E';}
    if (s.posicions[s.length].x < s.posicions[s.length-1].x)
    {s.dircua = 'O';}
    if (s.posicions[s.length].y > s.posicions[s.length-1].y)
    {s.dircua = 'S';}
    if (s.posicions[s.length].y < s.posicions[s.length-1].y)
    {s.dircua = 'N';}

    if((s.posicions[s.length].x==s.posicions[1].x &&
s.posicions[s.length].y==s.posicions[1].y) || (s.length <= 0)){return 1;}

    return 0;
}

```

```

int crearRATA_mode(int mode /*= 0*/){

    //Crear nova rata
    int posicioReal = 0;
    int posicioIrreal;

    int xinicial;
    int yinicial;
    int ycaselles;
    int xcaselles;
    int desplaament;

    if (mode == 22){
        xinicial = 19; yinicial = 7;
        xcaselles = 18; ycaselles = 10;
        desplaament = 5;
    }
    if(mode == 25){
        xinicial = 15; yinicial = 4;
        xcaselles = 33; ycaselles = 19;
        desplaament = 3;
    }
    if(mode != 22 && mode != 25){
        xinicial = 14; yinicial = 2;
        xcaselles = 20; ycaselles = 12;
        desplaament = 5;
    }
    }

    posicioIrreal = aleat(1, (xcaselles * ycaselles - (s.length+1)));

    for(int y = 0; y < ycaselles; y+=1){
        for(int x = 0; x < xcaselles; x+=1){
            int augmenta = 1;
            for(int cos = 0; cos <= s.length; ++cos)
            {
                if(s.posicions[cos].x == xinicial+x*desplaament && s.posicions[cos].y ==
yinicial+y*desplaament)
                {
                    augmenta = 0; break;
                }
            }

            if(augmenta == 1){++posicioReal;}

            if(posicioReal >= posicioIrreal)
            {
                p.m0.R = i2(xinicial+x*desplaament, yinicial+y*desplaament);
                //printf("I%i R%i X%i Y%i\n",posicioIrreal, posicioReal, p.m0.R.x, p.m0.R.y);
                return 0;
            }
        }
    }
    return 1;
}

int crearRATA(){
    return crearRATA_mode(0);
}

int recomptePOSICIONS_6(){
    int cossosirreals = 0;
    for (int i = 0; i<240; i++){
        if(p.m6.posicionsPINTAR[i] == 'r'){
            for (int n = 0; n<=s.length; n+=1){
                if (posicionsXY(i).x == s.posicions[n].x && posicionsXY(i).y == s.posicions[n].y){
                    cossosirreals += 1; break;
                }
            }
        }
    }
    p.m6.cossosreals = s.length+1 - cossosirreals;
    return 0;
}

```



```

int crearRATA_BARRERES(){

    //Crear nova rata
    int posicioReal = 0;
    int posicioIrreal;

    recomptePOSICIONS_6(); if(p.m6.cossosreals!=0){p.m6.cossosreals-=1;}
    if (p.m6.cossosreals+p.m6.parets+p.m6.NORATAs >= 20*12){return 1;}
    posicioIrreal = aleat(1, (20 * 12 - (p.m6.cossosreals+p.m6.parets+p.m6.NORATAs)));

    for(int y = 0; y < 12; y+=1){
        for(int x = 0; x < 20; x+=1){
            int augmenta = 1;
            for(int cos = 0; cos <= s.length; ++cos)
            {
                if(s.posicions[cos].x == 14+x*5 && s.posicions[cos].y == 2+y*5)
                {augmenta = 0; break;}
            }
            if(p.m6.posicionsPINTAR[x*12+y] != 'n'){augmenta = 0;}
            if(p.m0.R.x == 14+x*5 && p.m0.R.y == 2+y*5){augmenta = 0;}

            if(augmenta == 1){++posicioReal;}

            if(posicioReal >= posicioIrreal)
            {
                p.m0.R = i2(14+x*5, 2+y*5);
                //printf("I%i R%i X%i Y%i\n",posicioIrreal, posicioReal, p.m0.R.x, p.m0.R.y);
                return 0;
            }
        }
    }
    return 1;
}

int crearRATA_5i13(int mode /*= 0*/){

    //Crear nova rata
    int posicioReal = 0;
    int posicioIrreal;

    if (mode == 5){posicioIrreal = aleat(1, (20 * 12 - (p.m5.NUMposicionsvisibles)));}
    if (mode == 13){posicioIrreal = aleat(1, (20 * 12 - (p.m13.NUMcasellesocupades)));}
    if (mode == 24){posicioIrreal = aleat(1, (20 * 12 - (p.m5.NUMposicionsvisibles+1)));}

    for(int y = 0; y < 12; y+=1){
        for(int x = 0; x < 20; x+=1){
            int augmenta = 1;
            if (mode == 5 || mode == 24){
                for(int posicio = 0; posicio <= p.m5.NUMposicionsvisibles; ++posicio)
                {
                    if(p.m5.posicionsvisibles[posicio].x == 14+x*5 &&
                       p.m5.posicionsvisibles[posicio].y == 2+y*5)
                    {
                        augmenta = 0; break;
                    }
                }
            }
            if (mode == 24){if (p.m24.prota.x == 14+x*5 &&
                             p.m24.prota.y == 2+y*5)
                {
                    augmenta = 0;
                }
            }
        }
    }
    else if (mode == 13){
        for(int casella = 0; casella <= p.m13.NUMcasellesocupades; ++casella)
        {
            if(p.m13.casellesocupades[casella].x == 14+x*5 &&
               p.m13.casellesocupades[casella].y == 2+y*5)
            {
                augmenta = 0; break;
            }
        }
    }
}

```

```

        if(p.m0.R.x == 14+x*5 && p.m0.R.y == 2+y*5){augmenta = 0;}

        if(augmenta == 1){++posicioReal;}

        if(posicioReal >= posicioIrreal)
        {
            p.m0.R = i2(14+x*5, 2+y*5);
            //printf("I%i R%i X%i Y%i\n",posicioIrreal, posicioReal, p.m0.R.x, p.m0.R.y);
            return 0;
        }
    }
}
return 1;
}

int crearRATA_DIAGONAL(){

    //Crear nova rata
    int posicioReal = 0;
    int posicioIrreal;

    posicioIrreal = aleat(1, (20 * 12 - (s.length+1+p.m12.index12)));

    for(int y = 0; y < 12; y+=1){
        for(int x = 0; x < 20; x+=1){
            int augmenta = 1;
            for(int cos = 0; cos <= s.length; ++cos)
            {
                if(s.posicions[cos].x == 14+x*5 && s.posicions[cos].y == 2+y*5)
                {augmenta = 0; break;}
            }
            for(int i12 = 0; i12 < p.m12.index12; ++i12)
            {
                if(p.m12.posicions12[i12].x == 14+x*5 && p.m12.posicions12[i12].y == 2+y*5)
                {augmenta = 0; break;}
            }
            if(p.m0.R.x == 14+x*5 && p.m0.R.y == 2+y*5){augmenta = 0;}

            if(augmenta == 1){++posicioReal;}

            if(posicioReal >= posicioIrreal)
            {
                p.m0.R = i2(14+x*5, 2+y*5);
                //printf("I%i R%i X%i Y%i\n",posicioIrreal, posicioReal, p.m0.R.x, p.m0.R.y);
                return 0;
            }
        }
    }
    return 1;
}

int moureMIRA(){

    if (p.m3.dir_mira == 'N'){p.m3.mira.y -= 5;}
    if (p.m3.dir_mira == 'S'){p.m3.mira.y += 5;}
    if (p.m3.dir_mira == 'O'){p.m3.mira.x -= 5;}
    if (p.m3.dir_mira == 'E'){p.m3.mira.x += 5;}

    if (p.m3.mira.x < 14) {p.m3.mira.x = 109;}
    if (p.m3.mira.x > 109) {p.m3.mira.x = 14;}
    if (p.m3.mira.y < 2) {p.m3.mira.y = 57;}
    if (p.m3.mira.y > 57) {p.m3.mira.y = 2;}

    return 0;
}

int moureORIGEN_EDITOR(){

    if (s.cap.x == p.m6.origen.x && s.cap.y == p.m6.origen.y){
        if (s.dirR == 'N'){p.m6.origen.y -= 5;} else if (s.dirR == 'S'){p.m6.origen.y += 5;}
        else if (s.dirR == 'O'){p.m6.origen.x -= 5;} else if (s.dirR == 'E'){p.m6.origen.x += 5;}
    }

    if (p.m6.origen.x < 14){p.m6.origen.x=109; p.m6.dirOri = 'O';}

```

```

        if (p.m6.origen.x > 109){p.m6.origen.x=14; p.m6.dirOri = 'E';}
        if (p.m6.origen.y < 2){p.m6.origen.y=57; p.m6.dirOri = 'N';}
        if (p.m6.origen.y > 57){p.m6.origen.y=2; p.m6.dirOri = 'S';}

        return 0;
    }

int moureRATA_MOBILS(){
    int xy = aleat(0,3);
    for (int i = 0; i <= s.length; ++i)
    {
        if (
            ((xy == 0) && ((p.m0.R.x+5 == s.posicions[i].x && p.m0.R.y == s.posicions[i].y) ||
            (p.m0.R.x == 109))) ||
            ((xy == 1) && ((p.m0.R.x-5 == s.posicions[i].x && p.m0.R.y == s.posicions[i].y) ||
            (p.m0.R.x == 14 ))) ||
            ((xy == 2) && ((p.m0.R.y+5 == s.posicions[i].y && p.m0.R.x == s.posicions[i].x) ||
            (p.m0.R.y == 57 ))) ||
            ((xy == 3) && ((p.m0.R.y-5 == s.posicions[i].y && p.m0.R.x == s.posicions[i].x) ||
            (p.m0.R.y == 2 )))
        )
            {return 0;}
        }
        if (xy == 0){p.m0.R.x += 5;} if (xy == 1){p.m0.R.x -= 5;} if (xy == 2){p.m0.R.y += 5;} if (xy ==
        3){p.m0.R.y -= 5;}

        return 0;
    }

int moureRATA_BASQUET(){
    if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){
        if (s.dirR == 'N'){p.m0.R.y -= 5;} if (s.dirR == 'S'){p.m0.R.y += 5;}
        if (s.dirR == 'O'){p.m0.R.x -= 5;} if (s.dirR == 'E'){p.m0.R.x += 5;}
    }

    if (p.m0.R.x < 14 || p.m0.R.x > 109|| p.m0.R.y < 2 || p.m0.R.y > 57){return 1;}

    for (int n = 0; n<=s.length; n++)
    {
        if (s.posicions[n].x == p.m0.R.x && s.posicions[n].y == p.m0.R.y)
            {return 2;}
    }

    return 0;
}

int moureRATA_IA(){

    I2 posicio_al_inici = p.m24.prota;
    //printf("%i ",s.lloc);
    if (p.m24.dirRATA == 'N'){p.m24.prota.y -= 5;}
    if (p.m24.dirRATA == 'S'){p.m24.prota.y += 5;}
    if (p.m24.dirRATA == 'O'){p.m24.prota.x -= 5;}
    if (p.m24.dirRATA == 'E'){p.m24.prota.x += 5;}
    //printf("%i\n",s.lloc);

    //Detectar sortida dels limits
    if (p.m24.prota.x < 14 || p.m24.prota.x > 109 || p.m24.prota.y < 2 || p.m24.prota.y > 57)
        {p.m24.prota = posicio_al_inici; return 1;}

    //Detectar trompada amb ell mateix
    for (int n = 1; n<=s.length; n+=1)
    {
        if (p.m24.prota.x == s.posicions[n].x && p.m24.prota.y == s.posicions[n].y)
            {p.m24.prota = posicio_al_inici; return 2;}
    }

    return 0;
}

int mourePIT(){
    if (p.m11.pitfora == 0){p.m11.P = s.cap;}
    else {
        if (p.m11.P.x < 14 || p.m11.P.x > 109|| p.m11.P.y < 2 || p.m11.P.y > 57)

```

```

    {p.m11.pitfora = 0;}

    else{
        if (p.m11.direcciopit == 'E'){p.m11.P.x += 5;}
        if (p.m11.direcciopit == 'O'){p.m11.P.x -= 5;}
        if (p.m11.direcciopit == 'N'){p.m11.P.y -= 5;}
        if (p.m11.direcciopit == 'S'){p.m11.P.y += 5;}

        for (int n = 2; n<=s.length; n+=1)
        {
            if (p.m11.P.x == s.posicions[n].x && p.m11.P.y == s.posicions[n].y)
                {p.m11.pitfora = 0; return 1;} //printf("M'ha entrat un mugró a l'ull");
        }
        if (p.m11.P.x == p.m0.R.x && p.m11.P.y == p.m0.R.y){
            p.m11.pitfora = 0; ferCREIXERoDECREIXER_mode(1); s.creixer = 1; crearRATA();
            //printf("pit tocant rata\n");
        }
    }
}
return 0;
}

char donarlordre(){
    p.m3.distanciaX = p.m3.ordre.x-s.cap.x;
    p.m3.distanciaY = p.m3.ordre.y-s.cap.y;
    if(abs(p.m3.distanciaX)>abs(p.m3.distanciaY)){
        if (p.m3.distanciaX < 0 && s.dirR != 'E'){return 'O';}
        else if (p.m3.distanciaX > 0 && s.dirR != 'O'){return 'E';}
    }
    else{
        if (p.m3.distanciaY < 0){return 'N';}
        else if (p.m3.distanciaY > 0){return 'S';}
    }
    else{
        if (s.cap.y == 2){return 'S';}    else if (s.cap.y == 57){return 'N';}
        else{if(aleat(0,1) == 0){return 'N';} else{return 'S';}}}
    }
}
else{
    if (p.m3.distanciaY < 0 && s.dirR != 'S'){return 'N';}
    else if (p.m3.distanciaY > 0 && s.dirR != 'N'){return 'S';}
}
else{
    if (p.m3.distanciaX < 0){return 'O';}
    else if (p.m3.distanciaX > 0){return 'E';}
}
else{
    if (s.cap.x == 19){return 'E';}    else if (s.cap.x == 109){return 'O';}
    else{if(aleat(0,1) == 0){return 'O';} else{return 'E';}}}
}
}
return 'X';
}

int recompteCOSSOS_5(){
    int sumo;
    p.m5.posicionsvisibles[0] = s.posicions[0];
    p.m5.NUMposicionsvisibles = 1;
    for (int n = 1; n<=s.length; n+=1){
        sumo = 1;
        for (int m = n-1; m>=0; m-=1){
            if (s.posicions[n].x == s.posicions[m].x && s.posicions[n].y == s.posicions[m].y){sumo =
0; break;}
        }
        if (sumo == 1){p.m5.posicionsvisibles[p.m5.NUMposicionsvisibles] = s.posicions[n];}
        p.m5.NUMposicionsvisibles+=sumo;
    }
    return 0;
}

I2 * casellesocupades(I2 pos){
    static I2 caselles[4];
    if (pos.x-14 % 5 == 0 && pos.x-2 % 5 == 0)
    {
        caselles[0] = pos;
    }
    else if (pos.x-14 % 5 != 0 && pos.x-2 % 5 == 0)
    {
        caselles[0] = i2((pos.x-14)/5*5+14,pos.y);
    }
}

```

```

        caselles[1] = i2(((pos.x-14)/5+1)*5+14,pos.y);
    }
    else if (pos.x-14 % 5 == 0 && pos.x-2 % 5 != 0)
    {
        caselles[0] = i2(pos.x,(pos.y-2)/5*5+2);
        caselles[1] = i2(pos.x,((pos.y-2)/5+1)*5+2);
    }
    else if (pos.x-14 % 5 != 0 && pos.x-2 % 5 != 0)
    {
        caselles[0] = i2((pos.x-14)/5*5+14,(pos.y-2)/5*5+2);
        caselles[1] = i2(((pos.x-14)/5+1)*5+14,((pos.y-2)/5+1)*5+2);
        caselles[2] = i2((pos.x-14)/5*5+14,((pos.y-2)/5+1)*5+2);
        caselles[3] = i2(((pos.x-14)/5+1)*5+14,(pos.y-2)/5*5+2);
    }
    return caselles;
}

int recompteCASELLES_13(){
    int augmenta;
    I2 *casellesfuncio;
    p.m13.NUMcasellesocupades = 0;
    //int n = 0;
    for(int cos = 0; cos <= s.length; ++cos)
    {
        casellesfuncio = casellesocupades(s.posicions[cos]);
        for(int i = 0; i < 4; i++){
            augmenta = 1;
            for(int casocu = 0; casocu < p.m13.NUMcasellesocupades; casocu++)
            {
                //n++;
                if (casellesfuncio[i].x == p.m13.casellesocupades[casocu].x && casellesfuncio[i].y
== p.m13.casellesocupades[casocu].y)
                {augmenta = 0; break;}
            }
            if (augmenta == 1){
                p.m13.casellesocupades[p.m13.NUMcasellesocupades] = casellesfuncio[i];
                p.m13.NUMcasellesocupades+=augmenta;
            }
        }
    }
    //printf("%i\n",n);
    return 0;
}

int gestionarRATES_mode(int mode /*= 0 */){
    for(int r=0; r<5; r+=1){
        int mha_pillat = 0;
        if (s.posicions[0].x == p.m14.rates[r].origen.x && s.posicions[0].y ==
p.m14.rates[r].origen.y){
            if(p.m14.rates[r].caracter == 1){mha_pillat = 1;}
            else{return 1;}
        }

        if (mha_pillat == 1 || p.m14.rates[r].quanhaaparegut + p.m14.rates[r].durada*5 <=
numero_de_fotograma || mode == 1){
            if(mha_pillat == 1){s.creixer = 1; ferCREIXERoDECREIXER();}
            int surto = 0;
            while (surto == 0){
                surto = 1;
                p.m14.rates[r].origen.x = 14+aleat(0,19)*5;
                p.m14.rates[r].origen.y = 2+aleat(0,11)*5;
                for(int i = 0; i<=s.length; ++i){
                    if (s.posicions[i].x == p.m14.rates[r].origen.x && s.posicions[i].y ==
p.m14.rates[r].origen.y)
                    {surto = 0; break;}
                }
                int aux = 0;
                for(int rr=0; rr<5; rr+=1){
                    if (p.m14.rates[rr].origen.x == p.m14.rates[r].origen.x &&
p.m14.rates[rr].origen.y == p.m14.rates[r].origen.y)
                    {aux += 1;}
                }
                if(aux > 1){surto = 0;}
            }
        }
    }
}

```

```

        p.m14.rates[r].quanhaaparegut = numero_de_fotograma;
        p.m14.rates[r].durada = aleat(15,50);
        p.m14.rates[r].caracter = aleat(0,1);
    }
}
return 0;
}

int gestionarRATES(){
    return gestionarRATES_mode(0);
}

int toca_paret(I2 pos, int mode){
    int ustontodelavariablenmode = mode; //printf("%i",mode);
    if (pos.x < 14 || pos.x > 109 || pos.y < 2 || pos.y > 57){
        return 1; }

    if (p.m22.index22 != 0){
        for (int i = 0; i<p.m22.index22; i+=1){
            if (p.m22.posicions22[i].x == pos.x && p.m22.posicions22[i].y == pos.y){return 0;} } }

    for (int c = 0; c<=s.length; c+=1){
        if (s.posicions[c].x == pos.x && s.posicions[c].y == pos.y){return 0;} }

    p.m22.posicions22[p.m22.index22] = pos; p.m22.index22 += 1;
    if (toca_paret(i2(pos.x, pos.y-5),1) || //north
        toca_paret(i2(pos.x, pos.y+5),2) || //south
        toca_paret(i2(pos.x+5, pos.y ),3) || //east
        toca_paret(i2(pos.x-5, pos.y ),4)){ //west
        return 1; }
    else{return 0;}
}

/*
int trucs(){
    if (controller.A.pressed){ferCREIXERoDECREIXER();}
    if (controller.B.pressed && s.length > 1){ferCREIXERoDECREIXER();}
    if (controller.C.pressed){recompteCASELLES_13(); crearRATA_5i13(13);}
    if (controller.UP.pressed && controller.UP.half_transition_count){p.m0.R.y -= 5;}
    else if (controller.DOWN.pressed && controller.DOWN.half_transition_count){p.m0.R.y += 5;}
    else if (controller.LEFT.pressed && controller.LEFT.half_transition_count){p.m0.R.x -= 5;}
    else if (controller.RIGHT.pressed && controller.RIGHT.half_transition_count){p.m0.R.x += 5;}
    return 0;
}
*/

```

funcions/funcions_setup.h

```
void inicialitzar_finestres(){

    finestra_general.origen = i2(0, 0);
    finestra_general.mida = i2(127, 63);
    finestra_general.bitmap = bp;

    finestra_record.origen = i2(0, 0);
    finestra_record.mida = i2(14, 64);
    finestra_record.bitmap = bp;

    finestra_puntuacio.origen = i2(114,0);
    finestra_puntuacio.mida = i2(14, 64);
    finestra_puntuacio.bitmap = bp;

    finestra_joc.origen = i2(14, 2);
    finestra_joc.mida = i2(100, 60);
    finestra_joc.bitmap = bp;
}

int iniciar_modes_i_serp(int mode){

    numero_de_fotograma = 0;

    s.length = 1;
    s.creixer = 0;
    s.dir = 'E';
    s.dirR = 'E';
    s.dircua = 'O';
    s.cap = i2(19,2);
    s.lloc = 12;
    s.posicions[0] = s.cap;
    s.posicions[1] = i2(14, 2);

    memset((void *)taula_general, 0, 240 * sizeof(I2)); // NOTE: Reset de tots els bytes de
    taula_general
    // NOTE: Equivalent a:
    // for(int i=0; i < 240; ++i){
    //     taula_general[i].x = 0;
    //     taula_general[i].y = 0;
    // }

    crearRATA_mode(22);
    //mo0
    p.m0.pausat = 0;
    p.m0.puntuacio = 0;

    //mo1
    if(mode == 2){//mo2
        p.m2.index_dir = 1;}
    if(mode == 3){//mo3
        p.m3.mira = i2(109, 57);
        p.m3.ordre_donada = 0;}
    //mo4
    //mo5
    if(mode == 6){//mo6
        s.posicions[0] = s.cap = i2(109,57);
        s.lloc = 239;
        s.dir = 'O';
        p.m0.R = p.m6.origen;
        p.m6.acabemdecomencar = 1;
        p.m6.colorBN = 'i';}
    if(mode == 66){//mo66
        s.posicions[0] = s.cap = p.m6.origen;
        s.lloc = ((s.cap.x-14)/5)*12+((s.cap.y-2)/5);
        s.dir = p.m6.dirOri;
        if(s.dir == 'N'){s.posicions[1] = i2(s.posicions[0].x,s.posicions[0].y+5);}
        else if(s.dir == 'S'){s.posicions[1] = i2(s.posicions[0].x,s.posicions[0].y-5);}
        else if(s.dir == 'E'){s.posicions[1] = i2(s.posicions[0].x-5,s.posicions[0].y);}
        else if(s.dir == 'O'){s.posicions[1] = i2(s.posicions[0].x+5,s.posicions[0].y);}
        crearRATA_BARRERES();}
    if(mode == 7){//mo7
        p.m7.acabemdecomencar = 1;}
```

```

//mo8
if(mode == 9){//mo9
    p.m9.increment = 1000;}
if(mode == 10){//mo11
    p.m10.mode = 0;}
if(mode == 11){//mo11
    p.m11.pitfora = 0;}
//mo12
if(mode == 13){//mo13
    s.length = 10;
    s.cap.x = 24; s.posicions[0] = s.cap;
    for(int i=1;i<=10;i++){s.posicions[i] = i2(s.cap.x-1*i, 2);}}
if(mode == 14){//mo14
    gestionarRATES_mode(1);}
if(mode == 15){//mo15
    p.m15.dirRATA = 'S';}
//mo16
//mo17
if(mode == 18){//mo18
    p.m18.flash = 0;}
//mo19
//mo20
if(mode == 20){
    s.length = 4;
    s.posicions[4]=s.posicions[3]=s.posicions[2]=s.posicions[1];}
if(mode == 21){//mo22
    p.m21.forat = p.m0.R; crearRATA_mode(22);}
if(mode == 22){//mo21
    s.length = 6;
    s.posicions[6]=s.posicions[5]=s.posicions[4]=s.posicions[3]=s.posicions[2]=s.posicions[1];}
if(mode == 23){//mo23
    s.length = 2;
    s.cap = i2(49,32);
    s.posicions[0] = s.cap; s.posicions[1] = i2(44,32); s.posicions[2] = i2(39, 32);
    p.m0.R.x = 109;}
if(mode == 24){//mo24
    p.m24.dirRATA = 'O';
    p.m24.prota = i2(109,57);}
if(mode == 25){//mo25
    s.length = 0;
    s.cap = i2(15,4); s.posicions[0] = s.cap;
    crearRATA_mode(25);}

return 0;
}

```


modes/modes_control.h

```
int mode_01_INVERTIT(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_INVERS();
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(1) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_02_EGO(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_EGO();
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
        p.m2.puc_girar = 1;
    }
    if (pintarhoTOT(2) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_03 ESTRATEGIC(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    p.m3.dir_mira = 'X';
    gestionarINPUT_STRA();
    if (s.cap.x == p.m3.ordre.x && s.cap.y == p.m3.ordre.y){p.m3.ordre_donada = 0;}
    if (numero_de_fotograma % 15 == 0){trucs(controller, s, p);
        if(p.m3.ordre_donada == 1){p.m3.direccio_a_moure_serp = donarlordre();}
        else{p.m3.direccio_a_moure_serp = s.dir = s.dirR;}
        int m = moureSERP_dir(p.m3.direccio_a_moure_serp);
        if(m == 1){final = 4;} else if(m == 2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (numero_de_fotograma % 5 == 0){moureMIRA();}
    if (pintarhoTOT(3) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}
```

modes/modes_limit.h

```
int mode_04_VORES(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        int m = moureSERP_VORES(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(4) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_05_FANTASMA(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        ferCREIXERoDECREIXER();
        int m = moureSERP_FANTASMA(); if(m == 1){final = 4;} if(m == 2){final = 8;}
        recompteCOSSOS_5();
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; crearRATA_5i13(5);}
    }
    if (pintarhoTOT(5) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_06_PINTAR(){
    int final = 0; t = time(NULL) - segons;
    gestionarINPUT_PINTAR();
    if (numero_de_fotograma % 10 == 0){
        moureSERP_VORES();
        int n = moureORIGEN_EDITOR();
    }
    if(p.m6.colorBN != 'i'){p.m6.posicionsPINTAR[s.lloc] = p.m6.colorBN;}
    p.m6.parets = p.m6.NORATAs = 0;
    for (int i = 0; i<240; i++){
        if(p.m6.posicionsPINTAR[i] == 'b'){p.m6.parets+=1;}
        else if(p.m6.posicionsPINTAR[i] == 'r'){p.m6.NORATAs+=1;}}

    //if(controller.C.pressed){final = 6;}
    if (pintarhoTOT(6) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_66_BARRERES(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    if(p.m6.posicionsPINTAR[s.lloc] == 'b'){p.m6.acabemdecomencar = 2; return 8;}
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        I2 posicio_al_inici = s.cap;
        int m = moureSERP_dir_mode('X',66); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){
            s.creixer = 1; ferCREIXERoDECREIXER();
            if(crearRATA_BARRERES()==1){s.posicions[0] = s.cap = posicio_al_inici; final = 8;}}
    }
    if (pintarhoTOT(66) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}
```

modes/modes_serp.h

```
int mode_07_REGULABLE(){
    int final = 0; t = time(NULL) - segons;
    gestionarINPUT();
    if(p.m7.acabemdecomencar == 0){
        if (controller.A.pressed){s.creixer = 1;} if (controller.B.pressed && s.length >
1){s.creixer = -1;}}
    else{
        if (controller.A.half_transition_count){p.m7.acabemdecomencar = 0;}}
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        ferCREIXERoDECREIXER();
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){p.m0.puntuacio += 1; crearRATA();}
    }
    if (pintarhoTOT(7) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_08_MANUAL(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_MANUAL();
    if(s.dir != ' '){
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(8) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_09_INCRE(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma >= 10000) {trucs(controller, s, p);
        numero_de_fotograma = 0; p.m9.increment += 1;
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(9) == 1){final = 3;}

    numero_de_fotograma += p.m9.increment;
    return final;}

int mode_10_CAPICUA(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_CAPICUA();
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
    }
    if (pintarhoTOT(10) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_11_REMOT(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_REMOT();
    if (numero_de_fotograma % 5 == 0) {
        int n = mourePIT(); if(n == 1){final = 8;}
        if (numero_de_fotograma % 10 == 0){trucs(controller, s, p);
            int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        }
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; crearRATA();}
        ferCREIXERoDECREIXER();
    }
    if (pintarhoTOT(11) == 1){final = 3;}

    ++numero_de_fotograma;
```

```

    return final;}

int mode_12_DIAGONAL(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_DIAGONAL(); //TODO: POTSER UNIR TOTS DOS IFs?
    if (numero_de_fotograma % 7 == 0) { //trucs(controller, s, p);
        ferCREIXERoDECREIXER(); int m = moureSERP_DIAGONAL();
        if(m == 1){final = 4;} else if(m==2){final = 5;}}
    if (numero_de_fotograma % 7 == 1) {
        if ((s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y) ||
            (p.m12.cap12[0].x == p.m0.R.x && p.m12.cap12[0].y == p.m0.R.y) ||
            (p.m12.cap12[1].x == p.m0.R.x && p.m12.cap12[1].y == p.m0.R.y)){
            s.creixer = 1; crearRATA_DIAGONAL();}
    }
    if (pintarhoTOT(12) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_13_SMOOTH(){
    int final = 0; t = time(NULL) - segons;
    gestionarINPUT();
    if (numero_de_fotograma % 2 == 0) { //trucs();
        int m = moureSERP_SMOOTH(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        for(int i = 0; i < 4; i++){
            if(casellesocupades(s.cap)[i].x == p.m0.R.x && casellesocupades(s.cap)[i].y == p.m0.R.y)
            {
                recompteCASELLES_13();
                crearRATA_5i13(13);
                for(int i=0;i<5;i++){ferCREIXERoDECREIXER_mode(1);}
                p.m0.puntuacio+=1;
            }
        }
        if (pintarhoTOT(13) == 1){final = 3;}
    }

    ++numero_de_fotograma;
    return final;}

```

modes/modes_rata.h

```
int mode_14_PLUS(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        I2 posicio_al_inici = s.cap;
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (gestionarRATES()==1){s.posicions[0] = s.cap = posicio_al_inici; final = 8;};
    }
    if (pintarhoTOT(14) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_15_DIRECCIONALS(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        I2 posicio_al_inici = s.cap;
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){
            if(s.dirR == p.m15.dirRATA){
                s.creixer = 1; ferCREIXERoDECREIXER(); crearRATA();
                p.m15.ladirecciodelarataescorrecta = 0;
                while(p.m15.ladirecciodelarataescorrecta != 1){
                    p.m15.ladirecciodelarataescorrecta = 1;
                    p.m15.dirRATA = p.m2.direccions[aleat(0,3)];
                    if (p.m0.R.y == 2 && p.m15.dirRATA == 'S'){p.m15.ladirecciodelarataescorrecta =
0;}}
                                else if (p.m0.R.y == 57 && p.m15.dirRATA ==
'N'){p.m15.ladirecciodelarataescorrecta = 0;}}
                                if (p.m0.R.x == 14 && p.m15.dirRATA == 'E'){p.m15.ladirecciodelarataescorrecta =
0;}}
                                else if (p.m0.R.x == 109 && p.m15.dirRATA ==
'O'){p.m15.ladirecciodelarataescorrecta = 0;}}
                            }
                        }
                    else{s.posicions[0] = s.cap = posicio_al_inici; final = 8;}}
                }
            }
        if (pintarhoTOT(15) == 1){final = 3;}

        ++numero_de_fotograma;
        return final;}

int mode_16_RATESMOBILS(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
        moureRATA_MOBILS();
    }
    if (pintarhoTOT(16) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}
```

modes/modes_grafic.h

```
int mode_17_SURVIVAL(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA();}
        pintarhoTOT(17);
    }

    ++numero_de_fotograma;
    return final;}

int mode_18_GALLINETA(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) {trucs(controller, s, p);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearRATA(); p.m18.flash = 1;}
    }
    pintarhoTOT(18);

    ++numero_de_fotograma;
    return final;}
```

modes/modes_objectiu.h

```
int mode_19_EASY(){
    int final = 0; t = time(NULL) - segons; //p.m0.puntuacio = 239-s.length;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        ferCREIXERoDECREIXER_mode(1);
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
    }
    p.m0.puntuacio = 239-s.length; if(p.m0.puntuacio == 0){final = 8;}
    if (pintarhoTOT(19) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_20_SelNat(){
    int final = 0; t = time(NULL) - segons;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        if (numero_de_fotograma % 130 == 0){if (s.length > 1){ferCREIXERoDECREIXER_mode(-1);}
    else{final = 8;}}
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
    p.m0.puntuacio+=1; crearRATA();}
    }
    if (pintarhoTOT(20) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_21_BASQUET(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        I2 posicio_al_inici = s.cap;
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if(s.cap.x == p.m21.forat.x && s.cap.y == p.m21.forat.y){s.posicions[0] = s.cap =
    posicio_al_inici; final = 8;}
        int n = moureRATA_BASQUET();
        if(n == 1){if(s.length > 1){s.length -= 1; crearRATA_mode(22);} else{final = 8;}}
        else if(n==2){final = 8;}
        else if(p.m0.R.x == p.m21.forat.x && p.m0.R.y == p.m21.forat.y){s.creixer = 1;
    ferCREIXERoDECREIXER(); crearRATA_mode(22);}
    }
    if (pintarhoTOT(21) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_22_LOOPS(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-6;
    gestionarINPUT();
    if (numero_de_fotograma % 10 == 0) { //trucs(controller, s, p);
        I2 posicio_al_inici = s.cap;
        int m = moureSERP(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.posicions[0] = s.cap = posicio_al_inici;
    final = 8;}
        else{
            p.m22.index22 = 0;
            if (toca_paret(p.m0.R,0)==0){s.creixer = 1; ferCREIXERoDECREIXER();
    crearRATA_mode(22);} //printf("\n");
        }
    }
    if (pintarhoTOT(22) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_23_1D(){
    int final = 0; t = time(NULL) - segons;
    gestionarINPUT_1D();
    if (numero_de_fotograma % 5 == 0) { //trucs(controller, s, p);
        I2 posicio_aux = p.m0.R;
        p.m0.R.x -= 5;
```

```

        if (p.m0.R.x < 14){crearRATA(); while(s.cap.y == p.m0.R.y){crearRATA();} p.m0.R.x = 109;}
        if (numero_de_fotograma % 10 == 0){
            ferCREIXERoDECREIXER_1D();
            int m = moureSERP_1D(); if(m == 1){final = 4;} else if(m==2){p.m0.R = posicio_aux; final
= 8;}
        }
        if (s.cap.x-5 <= p.m0.R.x && s.cap.x >= p.m0.R.x && s.cap.y == p.m0.R.y){
            s.creixer = 1; crearRATA(); while(s.cap.y == p.m0.R.y){crearRATA();} p.m0.R.x = 109;}
    }
    if (pintarhoTOT(23) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

int mode_24_IA(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length-1;
    gestionarINPUT_IA();
    if (numero_de_fotograma % 5 == 0){//trucs(controller, s, p);
        if (numero_de_fotograma % 10 == 0) {
            int r = moureRATA_IA(); if(r==1){final = 4;} else if(r==2){final = 5;}
        }
        if (numero_de_fotograma % 15 == 0) {
            ferCREIXERoDECREIXER();
            p.m3.ordre = p.m24.prota;
            p.m24.capAUX = s.cap; moureSERP_FANTASMA_dir_mode(donarlordre(),1);
            recompteCOSSOS_5();
        }
        if ((p.m24.prota.x == p.m0.R.x && p.m24.prota.y == p.m0.R.y) ||
            (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y)){s.creixer = 1;
crearRATA_5i13(24);}
            else if (p.m24.prota.x == s.cap.x && p.m24.prota.y == s.cap.y){final = 8; s.cap =
s.posicions[0] = p.m24.capAUX;}
        }
        if (pintarhoTOT(24) == 1){final = 3;}

        ++numero_de_fotograma;
        return final;}

```


modes/modes_altres.h

```
int mode_25_RETRO(){
    int final = 0; t = time(NULL) - segons; p.m0.puntuacio = s.length;
    gestionarINPUT();
    if (numero_de_fotograma % 6 == 0) { //trucs(controller, s, p);
        int m = moureSERP_RETRO(); if(m == 1){final = 4;} else if(m==2){final = 5;}
        if (s.cap.x == p.m0.R.x && s.cap.y == p.m0.R.y){s.creixer = 1; ferCREIXERoDECREIXER();
crearATA_mode(25);}
    }
    if (pintarhoTOT(25) == 1){final = 3;}

    ++numero_de_fotograma;
    return final;}

```