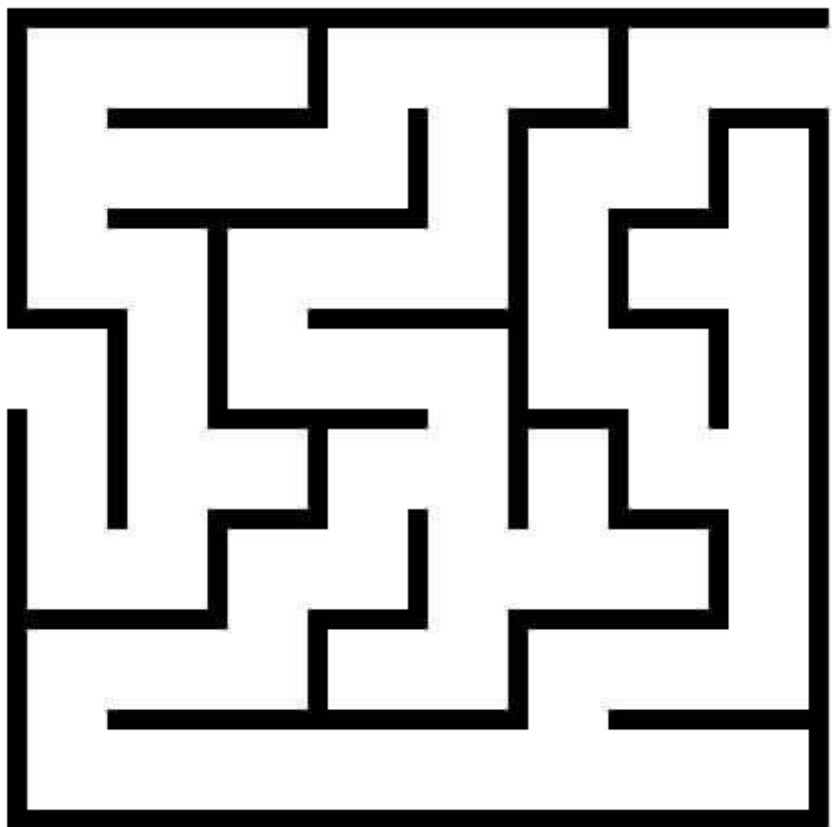


LABERINTOAREN JOKOA



Egileak: Jon Dorronsoro, Mikel Insausti eta Markel Galarraga

Sarrera:

Proiektu honetan C programazio lengoaia erabili dugu. Beste hainbat lengoaia ikusi ditugu beste ikasgaietan baina C lengoaia erabiltzea tokatu zaigu proiektu hau gauzatu ahal izateko. Esan daiteke, gure kasuan, lengoaia hau aurretik ikusitakoak baina errazagoa egin zaigula.

Hasieran nahiko trakets hasi ginen proiektuaren lehen laborategietan baina denborak aurrera egin ahala hobeto moldatu gara.

Jokoaren azalpena:

Gure jokoaren helburua laberinto bat pasatzea da denbora tarte batean, 3 minututan hain zuzen ere.. Laberintoa hainbat bidez osatuta dago eta helburua bide egokiak aukeratuz irteeraraino iristea da. Kontu handiz ibili beharko da bideak aukeratzeko momentuan, bide okerra aukeratuz gero, ez baitago irteerarik eta denbora aurrera doala, berriro etorritako bidetik itzuli beharra egongo da.

Jokoaren garapena:

Proiektuarekin hasteko, lehenbizi automata egin genuen. Ez genuen inongo arazorik eduki hau egiterakoan, gure jokoaren automata nahiko simplea da eta.

Automata egin ondoren, jokoaren kodeak egitera pasa ginen. Hasieran ez genekien ongi zer egin beharra geneukan eta nola, baina apunteen eta irakaslearen laguntzaz lortu genuen bide onetik jarraitzea.

Lehenik eta behin Egela-tik jetsitako 1.jarduerako karpeta barruan zeuden artxiiboak garatzen hasi ginen bertan emandako jarraipenak jarraituz. Bertan periferikoak.c eta jokia01.c artxiiboak osatzen hasi ginen. Ondoren, bigarren saioan, lehen laborategian egindakoa erabiliz aurrera jarraitu genuen.

Laberintoa aukeratzeko ordua iritsi zen; beraz, genituen laberinto ezberdinen artean bat aukeratu genuen, egokiena iruditu zitzaiguna. Hemen, matrize bat erabili dugu laberintoko paretak ez zeharkatu ahal izateko. Lehenik eta behin laberintoaren irudia photoshop programarekin ireki eta bertan pixelez-pixel kontatuz excel erabiliz matrizea osatu genuen. Matrizea bi karaktere ezberdinez osatu dugu; 1 zenbakia dagoen tokian laberintoko pareta dagoela esan nahi du eta, beraz, ezingo da zeharkatu eta 2 zenbakia dagoen tokitik zeharkatu ahal izango da. Gero excel programan egindako matrizea gure proiektura kopiatu genuen eta azkenean laberintoa prest egongo da.

Azkenik, falta ziren kode guztiak osatu genituen eta laberintoa pasatzeko temporizadorea jarri genuen.

Ukipen pantailaren kontrola:

Jokoan ukipen pantaila behin erabiltzen da, joko hasteko.

Teklatua kontrola:

Teklatuari dagokionez, geziak erabiltzen ditugu laberintoan zehar mugiarazteko, eta A tekla joko berrabiarazteko.

Denboragailuaren kontrola:

Jokoa hastean, denbora aktibatzen da. Jokalariak 3 minutu izango ditu laberintoa gainditzeko. Denbora pasatuz gero, joko amaitua izango da eta galdu egingo du.

Zehaztapenaren desbiderapena:

Hasierako automatan, START botoia erabiltzea pentsatua genuen pausa egoerara joateko, baina hainbat arazo izan ditugu START botoiak funtzionatu dezan eta azkenean denbora gutxi geneukanez, egoera hau kentzea erabaki dugu.

Kode zatia:

```
/*-----  
  
periferikoak.c  
  
-----*/  
#include <nds.h>  
#include <stdio.h>  
#include "definizioak.h"  
int tekla; //sakatutako tekla gordetzeko aldagaia  
int TeklaDetektatu()  
{  
    //TRUE itzultzen du teklaren bat sakatu dela detektatzen badu  
    if ((~TEKLAK_DAT & 0x03ff)!=0) return 1;  
    else return 0;  
}  
int SakatutakoTekla()  
{  
    //Sakatutako teklaren balioa itzultzen du:  
    A=0;B=1;Select=2;Start=3;Esk=4;Ezk=5;  
    //Gora=6;Behera=7;R=8;L=9;  
    if ((~TEKLAK_DAT & 0x0001)!=0) return A;
```

```

else if ((~TEKLAK_DAT & 0x0002)!=0) return B;
else if ((~TEKLAK_DAT & 0x0004)!=0) return SELECT;
else if ((~TEKLAK_DAT & 0x0008)!=0) return START;
else if ((~TEKLAK_DAT & 0x0010)!=0) return ESKUBI;
else if ((~TEKLAK_DAT & 0x0020)!=0) return EZKER;
else if ((~TEKLAK_DAT & 0x0040)!=0) return GORA;
else if ((~TEKLAK_DAT & 0x0080)!=0) return BEHERA;
else if ((~TEKLAK_DAT & 0x0100)!=0) return R;
else if ((~TEKLAK_DAT & 0x0200)!=0) return L;
}
void konfiguratuTeklatua(int TEK_konf)
{
    //Teklatuaren konfigurazioa bere S/I erregistroak aldatuz
    TEKLAK_KNT = TEKLAK_KNT | TEK_konf;
}
void konfiguratuTenporizadorea(int Latch, int TENP_konf)
{
    //Tenporizadorearen konfigurazioa bere S/I erregistroak aldatuz
    // Erregistroen aldaketa

    DENBO_DAT = Latch; // Latch-a datu erregistroan
gorde
    DENBO_KNT = DENBO_KNT | TENP_konf; // Konfiguratu kontrol-erregistroa
}

void TekEtenBaimendu()
{
    //Teklatuaren etenak baimendu
    //Lan hau burutzeko lehenengo eten guztiak galarazi behar dira eta bukaeran
baimendu
    IME=0;
    IE=IE|0X1000;
    IME=1;
}
void TekEtenGalarazi()
{
    //Teklatuaren etenak galarazi
    //Lan hau burutzeko lehenengo eten guztiak galarazi behar dira eta bukaeran
baimendu
    IME=0;
    IE=IE&~(0X1000);
    IME=1;
}
void DenbEtenBaimendu()
{
    //Denboragailu baten etenak baimendu (Timer0)
    //Horretarako lehenengo eten guztiak galarazi eta bukaeran berriro baimendu

```

```

        IME=0;
        IE=IE|0X0008;
        IME=1;
    }
    void DenbEtenGalarazi()
    {
        //Denboragailu baten etenak galarazi (Timer0)
        //Horretarako lehenengo eten guztiak galarazi eta bukaeran berriro baimendu
        IME=0;
        IE=IE&~(0X0008);
        IME=1;
    }

    void ErlojuaMartxanJarri()
    {
        DENBO_KNT=DENBO_KNT|0X0080;
    }

    void ErlojuaGelditu()

    {

        DENBO_KNT=DENBO_KNT&~(0X0080);

    }

```

```

/*-----
Hemen bideo sistemaren definizioa baino ez dugu egiten, beste EZER EZ DA IKUTU
BEHAR

Kode hau garatu da dovotoren "Simple sprite demo" adibidean eta Jaeden Ameronen
beste

adibide batean oinarrituta.

-----*/

#include <nds.h>           //nds-rako garatuta dagoen liburutegia

#include <stdio.h>         //c-ko liburutegi estandarra sarrera eta irteerako
funtzioak definitzen dituen

```

```

#include <stdlib.h>           //c-ko liburutegi estandarra memoria erreserbak eta
                              zenbaki konbertsioak egiteko

#include <unistd.h>           //Sistema eragileen arteko konpatibilitatea ziurtatzeko
                              liburutegia

#include "grafikoak.h"

#include "spriteak.h"


void hasieratuBideoa() {

    /* Bi pantailatan irudiak erakutsi ahal izateko memoria mapeatu. */

    vramSetMainBanks(VRAM_A_MAIN_BG_0x06000000,

                     VRAM_B_MAIN_BG_0x06020000,

                     VRAM_C_SUB_BG_0x06200000,

                     VRAM_E_LCD);

    vramSetBankE(VRAM_E_MAIN_SPRITE);

    vramSetBankD(VRAM_D_SUB_SPRITE);


    /* Pantaila nagusirako bideo modua ezarri */

    videoSetMode(MODE_5_2D | // 5. bideo modua ezarri

                 DISPLAY_BG2_ACTIVE | // 2. fondoa aktibatu

                 DISPLAY_BG3_ACTIVE); // 3. fondoa aktibatu

```

```

/* 2. mailako pantaila bideo modua ezarri */

videoSetModeSub(MODE_5_2D | // 5. bideo modua ezarri

    DISPLAY_BG3_ACTIVE); // 3. fondoa aktibatu
}


/* ---- ZATI HAU EZ DA ALDATU BEHAR ---- */

/* ---- FONDOEN KONFIGURAZIOA DA ---- */

/* ---- HEMENDIK HASITA ---- */


/* Fondo sistema konfiguratzeko prozedura */
void hasieratuFondoak() {

    /* Pantaila nagusiko 3 fondoaren afinitatea ezarri 16 biteko koloretarako. */

    REG_BG3CNT = BG_BMP16_256x256 |

        BG_BMP_BASE(0) | // Memoriako hasierako helbidea

        BG_PRIORITY(3); // Lehentasun baxua


    /* Pantaila nagusiko 3 fondoaren transformazio matrizeari identitate matrizea
    esleitu. */

    REG_BG3PA = 1 << 8;

    REG_BG3PB = 0;

    REG_BG3PC = 0;

    REG_BG3PD = 1 << 8;

```

```

/*****
*****/

/* Pantaila nagusiko 3 fondoaren egoera definitu. */

/* Grafikoa beste posizio batean jarri nahi izanez gero hau aldatu beharko da. */

REG_BG3X = 0;

REG_BG3Y = 0;

/*****
*****/

/* Pantaila nagusiko 3 fondoaren afinitatea ezarri 16 biteko koloretarako. */

REG_BG2CNT = BG_BMP16_128x128 |

    BG_BMP_BASE(8) | // Memoriako hasierako helbidea

    BG_PRIORITY(2); // Lehentasun baxua

/* Pantaila nagusiko 3 fondoaren transformazio matrizeari identitate matrizea
esleitu. */

REG_BG2PA = 1 << 8;

REG_BG2PB = 0;

REG_BG2PC = 0;

REG_BG2PD = 1 << 8;

/*****
*****/

/* Pantaila nagusiko 3 fondoaren egoera definitu. */

/* Grafikoa beste posizio batean jarri nahi izanez gero hau aldatu beharko da. */

REG_BG2X = -(SCREEN_WIDTH / 2 - 32) << 8;

```



```

REG_BG2Y = -32 << 8;

/*****
*****/

/* Bigarren mailako pantailako 3 fondoaren afinitatea ezarri 16 biteko koloretarako.
*/

REG_BG3CNT_SUB = BG_BMP16_256x256 |

    BG_BMP_BASE(0) | // Memoriako hasierako helbidea

    BG_PRIORITY(3); // Lehentasun baxua

/* Bigarren mailako pantailako 3 fondoaren transformazio matrizeari identitate
matrizea esleitu. */

REG_BG3PA_SUB = 1 << 8;

REG_BG3PB_SUB = 0;

REG_BG3PC_SUB = 0;

REG_BG3PD_SUB = 1 << 8;

/*****
*****/

/* Bigarren mailako pantailako 3 fondoaren egoera definitu. */

/* Grafikoa beste posizio batean jarri nahi izanez gero hau aldatu beharko da. */

REG_BG3X_SUB = 0;

REG_BG3Y_SUB = 0;

/*****
*****/

}

```

```

/* ---- HONAINO ---- */

void GrafikoakHasieratu()
{
    powerOn(POWER_ALL_2D);

    lcdMainOnBottom();

    hasieratuBideoa();

    //hasieratuFondoak();

    consoleDemoInit(); //Goiko pantaila testua erakusteko eta beheko irudiak
    erakusteko erabiliko da.
}


//*****

//Spriteak

//*****

/* Spritentzako memoria hasieratu.*/
void initSpriteMem() {

    oamInit(&oamMain, SpriteMapping_1D_32, false);

    oamInit(&oamSub, SpriteMapping_1D_32, false);

}

```

```

void HasieratuGrafikoakSpriteak()
{
    GrafikoakHasieratu();

    hasieratuFondoak();

    initSpriteMem();

    memoriaErreserbatu();

    PaletaNagusiaEzarri();

    SpriteakMemorianGorde();
}

```

```

/*-----
periferikoak.c
-----*/

#include <nds.h>
#include <stdio.h>
#include "definizioak.h"
#include "periferikoak.h"
#include "fondoak.h"
#include "spriteak.h"

int EGOERA;
int seg2=10;
int seg1=5;
int min=2;

void tekEten ()
{
    if (EGOERA==1){

```

```

        if (SakatutakoTekla()==GORA && Matrizea[y][x]!=1){
            fprintf("\x1b[5;1HTekla gora");
            y=y-2;
            ErakutsiErronboa(1,x,y);
        }
        if (SakatutakoTekla()==BEHERA && Matrizea[y+14][x]!=1){
            fprintf("\x1b[5;1HTekla behe");
            y=y+2;
            ErakutsiErronboa(1,x,y);
        }
        if (SakatutakoTekla()==EZKER && Matrizea[y][x]!=1){
            fprintf("\x1b[5;1HTekla ezkr");
            x=x-2;
            ErakutsiErronboa(1,x,y);
        }
        if (SakatutakoTekla()==ESKUBI && Matrizea[y][x+14]!=1){
            fprintf("\x1b[5;1HTekla eskb");
            x=x+2;
            ErakutsiErronboa(1,x,y);
        }
    }
}

```

```

void tenpEten()
{

```

```

    if(EGOERA == 1){
        seg2--;
        fprintf("\x1b[1;1HDenbora=%d", min);
        fprintf("\x1b[1;10H:%d", seg1);
        fprintf("\x1b[1;12H%d", seg2);
        if ((min == 0) && (seg1 == 0) && (seg2 == 0)){
            min=3;
            EGOERA=3;
            erakutsiBukaera2_Egoera();
        }
        if ((seg1==0)&&(seg2==0)){
            min--;
            seg1=5;
            seg2=10;
        }
        if (seg2==0){
            seg1--;
            seg2=10;
        }
    }
}

```

```

}

void etenZerbErrutEzarri()
{
    irqSet(IRQ_KEYS,tekEten);
    irqSet(IRQ_TIMER0,tenpEten);
}

```

```

/*-----
definizioak.h
-----*/

#include<nds.h>           //nds-rako garatuta dagoen liburutegia
#include<stdio.h>         //c-ko liburutegi estandarra sarrera eta irteerako
funtzioak definitzen dituen
#include<stdlib.h>        //c-ko liburutegi estandarra memoria erreserbak eta
zenbaki konbertsioak egiteko
#include<unistd.h>        //Sistema eragilearen arteko konpatibilitatea ziurtatzeko
liburutegia

#define IME      (*(vuint32*)0x04000208) //Interrupt Master Enable -- Eten
guztiak baimendu-galarazi
#define IE       (*(vuint32*)0x04000210) //InterruptEnable -- Etenak banaka
baimendu-galarazi
#define IF       (*(vuint32*)0x04000214) //Interrupt Flag -- Eten eskaeren
erregistroa

// Etenen tratamendurako funtzioak
#defineEtenakBaimendu() REG_IME=1 //Eten guztiak baimendu
#defineEtenakGalarazi() REG_IME=0 //Eten guztiak galarazi

//teklatuaren erregistroak
#define TEKLAK_DAT (*(vu16*)0x4000130) //teklatuaren datu erregistroa
#define TEKLAK_KNT (*(vu16*)0x4000132) //teklatuaren kontrol erregistroa

//denboragailuaren (Timer0) erregistroak
#define DENBO_KNT  (*(vuint16*)0x04000102) //kontrol erregistroa
#define DENBO_DAT  (*(vuint16*)0x04000100) //datu-erregistroa

//ukimen-pantailaren erregistroak
touchPosition PANT_DAT;

// Teklei izena eman diegu zenbakiak nahasgarriak izan daitezkeelako

```

```
#define A          0
#define B          1
#define SELECT     2
#define START      3
#define ESKUBI     4
#define EZKER      5
#define GORA       6
#define BEHERA     7
#define R          8
#define L          9
```

```
extern int EGOERA;
extern uint8 Matrizea[192][256];
extern int x,y;
extern int seg1,seg2,min;
```

Ondorioak:

Proiektua egin ondoren, esan daiteke C programazio lengoaia landu dugula eta arazoak gure artean lagunduz konpontzen ere ongi moldatu garela. Hasieran kosta egin zaigu hastea baina behin hasi eta gero nahiko arin eta azkartasunez egin dugu aurrera.

Jokoa sinplea izan arren, bere lana duela ikusi ahal izan dugu. Hasieran beldur ginen laberintoaren jokoa egiteko denboraz ea ongi ibiliko ginen, baina azkenean ez dugu arazo handirik izan denbora aldetik.

Zailtasun aldetik, gure jokoa oso sinplea da jokatzeko garaian baina ematen duena baino zailagoa da programatzerako garaian. Denboraz nahiko ongi ibili garen arren, denbora gehiago izanez gero, gauza gehiago gehitu ahalko genizkio. Hala ere, gustura geratu gara gure proiektuarekin.