Projecte curs Programació de plaques robòtiques per

Adrià Zayas

Enfonsar la flota v1.0

ÍNDEX:

1 - INTRODUCCIÓ

La idea del projecte es portar a terme el joc, per tots conegut, "Hundir la flota".

La part forta del projecte recau en el codi i no tant, potser, en la part electrònica.

He pogut dur a terme quasi tots els objectius inicials, però d'altres no s'han assolit degut a la manca de temps.

2 - DESCRIPCIÓ DEL PROJECTE

El joc tracta d'enfonsar els vaixells del rival, en aquest cas d'Arduino, mitjançant dispars en una quadrícula de 4x4. És un joc per torns i guanya el primer que enfonsi tots els vaixells rivals. Està fet per a un sol jugador (persona) contra la "intel·ligència artificial" d'Arduino però es podria modificar fàcilment per a implementar un mode de jugador contra jugador.

El programa comença demanant al jugador que introdueixi la posició de tots els seus vaixells, que en la versió actual són quatre, tres vaixells de dues caselles i un vaixell de tres caselles. Entrarem la posició de cada vaixell mitjançant una membrana de teclat (keypad) i guardarem aquesta informació en una array de Strings per a cada vaixell. En el moment de cada introducció, el programa comprovarà si les caselles introduïdes són correctes, fent una comparativa amb una array bidimensional que conté totes les caselles de la quadrícula. Si no fos correcte, en una pantalla LCD, es mostrarà un missatge d'error informant que la casella no és vàlida. Si és correcte procediría a demanar la posició del següent vaixell.

Un cop el programa disposi de les posicions dels vaixells del jugador, generarà de manera aleatòria, els seus propis vaixells.Ho farà per etapes.

Primer pas - Crearà un vaixell de dues caselles partint d'una posició escollida a l'atzar, fent servir la funció rand(), i després, també amb rand() i tenint en compte la posició relativa a la quadrícula, decidirà si crea la segona i última posició del vaixell en una direcció o a l'altre. Exemple: Si la primera posició ceada es B1, les úniques opcions posibles serien crear la segona posició en A1, B2 o C1 i això ho decidirà fent servir un algoritme de aleatorietat.

Segon pas - Igual que en el primer pas, generarà el vaixell fent servir el mateix algoritme. Una vegada creat, comprovarà que cap de les seves caselles es solapi/entri en conflicte amb les caselles ja generades del primer vaixell. Si aquest fos el cas, generaría de nou un altre vaixell fins que passes els criteris donats en el codi.

Tercer pas: El mateix que en el segon pas, però a l'hora de comprovar ho haurà de fer amb els dos vaixells ja creats.

Quart pas: Aquí generem el quart vaixell i l'únic de tres caselles. Ho farem agafant una casella inicial a l'atzar. Seguidament tindrem en compte la posició en la quadrícula i en aquest cas, només deixarem a l'atzar dues de les quatre possibilitats(X+ i Y+ o X- i Y-),ja que les altres vindran donades per la posició de la casella en la quadrícula. Exemple: Imaginem que la primera posició creada a l'atzar es B2. En aquest cas només podrem generar les 2 caselles restants en B3 i B4(eix Y+) o en C2 i D2(eix X+). Si generessin en direcció a B1 o A2, sortiríem de la quadrícula.

Un cop generades les 3 posicions del vaixell farem les comprovacions, com en els passos anteriors, per a assegurar-nos que no es solapen amb les posicions dels altres vaixells. Si no fos així, repetiríem el procés fins que es donin les condicions.

Mitjançant la pantalla LCD el joc ens informa de que els seus vaixells han estat generats i dona pas a començar amb el torn de dispar de l'usuari.

Introduirem la casella on volem disparar fent servir el teclat de membrana i per enviar la informació haurem de prémer " * ". El programa comprovarà si la casella introduïda està dins la quadrícula, si no és així, ens ho farà saber i haurem de repetir el dispar. Si és correcte comprovarà si hi ha encert o no i, mitjançant efectes de so, lumínics i en la pantalla LCD, ens

ho farà saber. També comprovarà si , en el cas de que hi hagi encert, el vaixell ha sigut enfonsat. Si fos així, també ens ho diria amb altres efectes de so i lumínics.

Un cop el jugador o l'Arduino consegueixi enfonsar tots els vaixells rivals, s'acaba el joc. Un servo amb una bandera farà aparició i ens dirà si hem guanyat o hem perdut. Seguidament el programa procedirà a posar els comptadors a 0 i a reiniciar les variables que siguin necessàries per al correcte desenvolupament del joc. Després d'uns efectes de llum i so ens demanarà de nou que introduïm la nova posició dels nostres vaixells.

Podem veure en l'esquema el funcionament del programa:

3 - MATERIAL UTILITZAT

1x Arduino Uno R3

1x Placa proves petita

1x Pantalla LCD (16, 2)

1x Teclat membrana (keypad)

1x Buzzer Passiu

4x LED

4x Resistència 220 Ω (LEDs)

1x Resistència 2KΩ (resistència fixa per a LCD)

2x Servo motor

35x Cables MM

4 - CODI

No podem posar tot el codi del que consta al programa aquí, ja que és molt extens, però sí que podem veure les parts més rellevants o importants. En la funció generaVaixellsRival() podem veure com crida a les funcions generaVaixellDe2() i generaVaixellDe3() i fa les comprovacions pertinents per decidir si la generació és bona o no.

```
void generaVaixellsRival(){
bool coincidencia=false;
bool noSolapaVaixells=true;
generaVaixellDe2(pos1Ard);
while(!noSolapaVaixells){
generaVaixellDe2(pos2Ard);
coincidencia=false;
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
if(pos1Ard[i]==pos2Ard[j]){
coincidencia=true;
break;
}
}
if(coincidencia==true)break;
}
if(coincidencia==false)noSolapaVaixells=true;
```

```
}
noSolapaVaixells=false;
while(!noSolapaVaixells){
generaVaixellDe2(pos3Ard);
coincidencia=false;
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
if(pos3Ard[i]==pos1Ard[j]||pos3Ard[i]==pos2Ard[j]){
coincidencia=true;
break;
}
}
if(coincidencia==true)break;
}
if(coincidencia==false)noSolapaVaixells=true;
}
noSolapaVaixells=false;
while(!noSolapaVaixells){
generaVaixellDe3(pos4Ard);
coincidencia=false;
for(int i=0;i<3;i++){
for(int j=0;j<2;j++){
if(pos4Ard[i]==pos3Ard[j]||pos4Ard[i]==pos2Ard[j]||pos4Ard[i]==pos1Ard[j]){
coincidencia=true;
break;
```

```
}
}
if(coincidencia==true)break;
}
if(coincidencia==false)noSolapaVaixells=true;
}
imprimirLCD("Vaixells Arduino", " generats!");
delay(2000);
imprimirLCD("Esperant dispar", " de l'usuari...");
}


void generaVaixellDe2(String pos[]){
int randomNum1=rand()%4;
int randomNum2=rand()%4;
pos[0]=bidimensionalQuadri[randomNum1][randomNum2];
if(rand()%2>0){
//-------------eix Y--------------
if(randomNum1==0)randomNum1++;
else if(randomNum1==3)randomNum1--;
else
{
if(rand()%2>0)randomNum1++;
else randomNum1--;
}
```

```
}
else
{
//-------------eix X--------------
if(randomNum2==0)randomNum2++;
else if(randomNum2==3)randomNum2--;
else
{
if(rand()%2>0)randomNum2++;
else randomNum2--;
}
}
pos[1]=bidimensionalQuadri[randomNum1][randomNum2];
}


void generaVaixellDe3(String pos[]){
int randomNum1=rand()%4;
int randomNum2=rand()%4;
pos[0]=bidimensionalQuadri[randomNum1][randomNum2];
if(rand()%2>0){
//------------------------eix Y-------------------------
if(randomNum1<2){
pos[1]=bidimensionalQuadri[randomNum1+1][randomNum2];
pos[2]=bidimensionalQuadri[randomNum1+2][randomNum2];
```

```
}
else
{
pos[1]=bidimensionalQuadri[randomNum1-1][randomNum2];
pos[2]=bidimensionalQuadri[randomNum1-2][randomNum2];
}
}
else
{
//------------------------eix X-------------------------
if(randomNum2<2){
pos[1]=bidimensionalQuadri[randomNum1][randomNum2+1];
pos[2]=bidimensionalQuadri[randomNum1][randomNum2+2];
}
else
{
pos[1]=bidimensionalQuadri[randomNum1][randomNum2-1];
pos[2]=bidimensionalQuadri[randomNum1][randomNum2-2];
}
}
}
```

5 - QUÈ HE APRÈS?

Memòria dinàmica i memòria d'emmagatzematge d'Arduino Uno R3 limitada. En aquest projecte hem vist que Arduino està molt limitat per a treballar amb programes de molt codi que contingui moltes variables globals, hem hagut de desistir en certes implementacions com podria ser fer sonar música durant el joc. També hem desistit de fer una connexió per I2C amb la pantalla LCD alliberant així quatre dels sis pins de dades.

També he hagut de no implementar algunes llibreries per el mateix motiu, la memoria.

Hem vist que en llenguatge C++ al voler fer servir la funció sizeof() en una array (que en altres llenguatges seria algo semblant com lenght()) no ens retornava la quantitat d'elements que conté sinó que retorna la quantitat de bytes que conté l'array. Això pot no ser un problema si els elements que conté ocupen tots la mateixa quantitat de bytes, però en el nostre cas hem hagut de fer servir un comptador per a saber la quantitat d'elements que conté certa array en cada moment.

He après també a incloure fitxers.h per a tenir el codi més net i intelligible. Alhora també he vist que es poden fer #includes parcials fent servir aquest mètode.

També he après que les aplicacions que pot tenir Arduino son infinites (tenint en compte les seves limitacions) i que pot ser utilitzat per a mil i un projectes.

6 - MILLORES I REPTES

- Implementar mode de dos jugadors.

- Incrementar el tamany de la quadrícula. En un moment vaig decidir fer la cuadrícula de 4x4 per facilitar el procés d'escriptura de tot el codi i perquè el keypad hem permetía introduir qualsevol casella de la quadrícula. Una vegada acabat, no seria tan complicat poder-ho implementar.

- Posar un mode difícil en la IA d'Arduino i que aquest tingui en compte els encerts. Així podria disparar a caselles annexes.