

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

ARDUINO PONG GAME

Documentatie

Facultatea: Automatica si Calculatoare

Roman Alexandru | Grupa 30238

CUPRINS

1. Obiectivul si descrierea proiectului
2. Componente folosite
3. Implementare
4. Rezultate
5. Concluzii
6. Bibliografie

1.Obiectivul acestui calculator

Jocurile sunt o categorie ale aplicatiilor software destinate destinderii si distractiei. Ele pot fii jucate de oricine detine un calculator, xbox, chiar si placute arduino. Poate este o opinie personala, dar jocurile devin maxim de frumoase si de distractive atunci cand sunt jucate cu prietenii. Din acest motv, am luat decizia sa creez un joculet de tip Pong in arduino care poate fii jucat de 2 oameni.

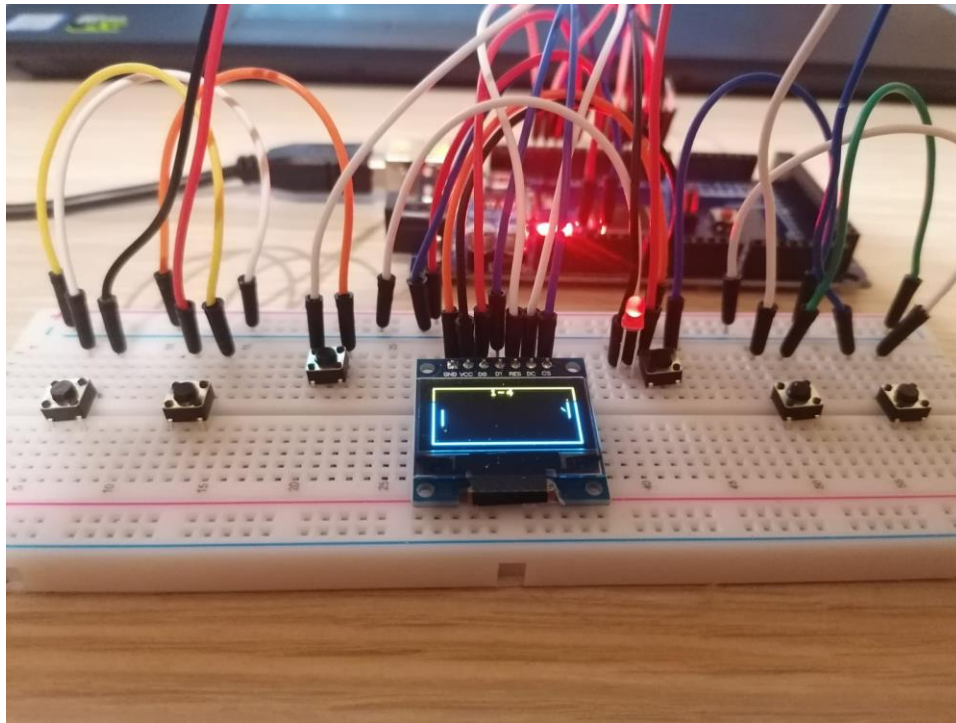
Proiectul este creat pe o placuta mega 2560.



2. Componente folosite

Pentru a rula acest proiect aveti nevoie de urmatoarele componente:

- o placuta arduino(poate fii mega 2560, sau alta)
- fire de conectare
- breadboard
- 6 butoane
- 1 led
- Lcd 0.96' (bazat pe circuit integrat 1306) dar poate fii portat si pe alte display uri
- 1 cablu de conectare a placutei la laptop



3. Implementare

Pentru inceput am adaugat librarile grafice Adafruit-GFX si Adafruit-SSD1306 care ofera o sintaxa comuna si un set de functii grafice. Acestea vor ajuta cu desenarea pixelilor si formelor geometrice care vor fii afisate pe ecran.

Urmatorul pas este conectarea butoanelor, displayului si ledurilor la placa folosind firele si breadboardul.

In functia setup cMinfiguram pinii pentru butoane si led, si folosim rezistenta pullup care ne scuteste de rezistente externe. Folosim functia display.begin() pentru a porni displayul. Afisam pe ecran logoul Adafruit care este prezent in bufferul displayului la initializare, apoi printam pe ecran cu ajutorul functiei welcome(), care scrie pe ecran "WELCOME TO ARDUINO PONG". Apoi apelam functia pressButtonTostart() pentru a afisa pe ecran "press button to enter". Dupa aceasta urmeaza un while care asteapta un input de la un buton ca si escape.

Urmeaza intrarea in loop, unde se ruleaza jocul. Aisi se afiseaza un dreptunghi care reprezinta marginea ecranului. Cu ajutorul unor variabile de tip bool detectam care butoane se apasa. Cu ajutorul unor while-uri se implementeaza functionalitatea butonului care pune pauza. Se utilizeaza mai multe while-uri deoarece 2 asteapta ca butonul sa nu mai fie apasat iar doar al treilea este while-ul care tine pauza jocului. Urmeaza un if important care verifica variabila resetBall care se activeaza la score sau end game.

```
if(ResetBall) //when game starts
{
    ballX = random(60, 68);
    ballY = 32;

    do{ballDirX = random(-1, 2);}
    while(ballDirX == 0);
    do{ballDirY = random(-1, 2);}
    while(ballDirY == 0);

    ResetBall = false;
}
```

Urmeaza o sectiune care se ocupa de mecanicile mingii. Vom folosi coordonatele mingii curente si directia in care se indreapta pentru a calcula coordonatele urmatoarei pozitii ale mingii. Aceste valori se aleg random atat pentru x cat si pentru y intre valorile {0,1} prin functia random(). Urmeaza niste verificari ale coordonatelor pozitiei urmatoare ale mingii pentru a determina daca se va apela una dintre functiile showScore() sau endGame(). Urmatoarele 3 if-uri verifica daca mingea se loveste de peretii displayului, respectiv paletele playerilor, si in functie de asta inmulteste y-ul, respectiv x-ul cu -1.

```

//-----hitting the horizontal walls
if(updateY == 2 || updateY == 62 ){
    ballDirY *= -1;
    //printNumberUtil(ballDirY, 108, 2, 1);
}
//-----hitting the paddle 1
if(updateX == paddle1X && updateY >=paddle1Y && updateY <= paddle1Y + PADDLE_SIZE ){
    ballDirX *= -1;
}
//-----hitting the paddle 2
if(updateX == paddle2X && updateY >=paddle2Y && updateY <= paddle2Y + PADDLE_SIZE ){
    ballDirX *= -1;
}

```

-important de mentionat faptul ca paleta 1 este cea din dreapta si 2 este cea din stanga.

Dupa toate aceste verificari se poate afisa mingea. Eu am ales sa afisez atat mingea cat si urmatoarea ei pozitie astfel, pe display apare o animatie foarte draguta a mingii, observandu-se de asemenea directia ei.

```

display.drawPixel(ballX, ballY, WHITE);
display.drawPixel(updateX, updateY, WHITE);

```

Urmatoarea parte se ocupa de miscarea paletelor. Aceste se misca in functie de inputurile celor 2 player, desigur, in limitele coordonatelor ecranului daca butonul de AI nu a fost apasat.

```

if(GameRunning && !AI) //paddle movement
{
    int paddle1UpperY = paddle1Y + PADDLE_SIZE, paddle2UpperY = paddle2Y + PADDLE_SIZE;
    //paddleUpdate += PADDLE_SIZE;
    display.drawFastVLine(paddle1X, paddle1Y, PADDLE_SIZE, WHITE);
    display.drawFastVLine(paddle2X, paddle2Y, PADDLE_SIZE, WHITE);

    if(paddle1UpperY < 61){
        if(P1_DOWN)
            paddle1Y++;
    }
    if(paddle1Y > 3){
        if(P1_UP)
            paddle1Y--;
    }
    if(paddle2UpperY < 61){
        if(P2_DOWN)
            paddle2Y++;
    }
    if(paddle2Y > 3){
        if(P2_UP)
            paddle2Y--;
    }

    P1_UP = false;
    P1_DOWN = false;
    P2_UP = false;
    P2_DOWN = false;
}

```

Daca defapt a fost apasat, atunci directia paletelor va fii modificata de “inteligenta artificiala”, care tot ce face este sa modifice pozitia paletelor dupa pozitia mingii atunci cand aceasta este aproape de palete. Partea de IA este implementata printr-un cod foarte simplu.

```

if( ballX > (SCREEN_WIDTH - 10) / 2){
    if((paddle1UpperY + paddle1Y)/2 > ballY){
        AI1_UP = true;
    }
    if((paddle1UpperY + paddle1Y)/2 < ballY){
        AI1_DOWN = true;
    }
}
else if( ballX < (SCREEN_WIDTH + 10) / 2){
    if((paddle2UpperY + paddle2Y)/2 > ballY){
        AI2_UP = true;
    }
    if((paddle2UpperY + paddle2Y)/2 < ballY){
        AI2_DOWN = true;
    }
}
}

```

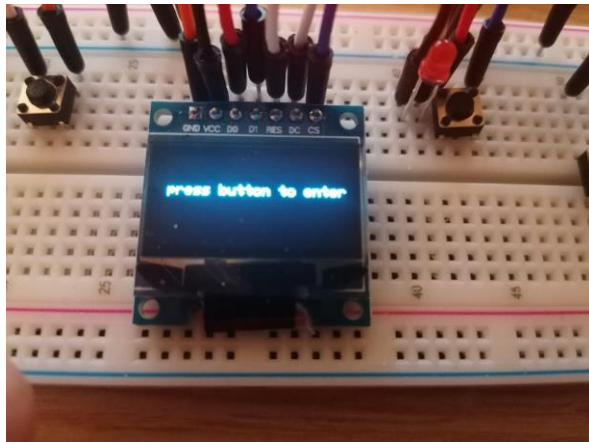
Ultimul lucru care ramane de facut este sa afisam bufferul scris si apoi sa il stergem pentru urmatoarea iteratie in loop().

Este de mentionat faptul ca in toate testele mele, efectuate pe o perioada lunga de timp, inteligenta artificiala nu a pierdut niciodata mingea.

4. Rezultate

Jocul se reseteaza la fiecare endGame si oricand poate se poate activa butonul de AI, care poate face paletetele sa joace pentru tine daca vrei sa iei o pauza cu adversarul tau. De asemenea ai si butonul de pauza in caz ca nu vrei sa te lasi in mana inteligentei artificiale.

Probabil ati observat ca nu am facut nici o posibilitate de a juca player vs AI. Ei bine, cred ca stim cu totii cine ar pierde pana la urma.. Dar nu acesta a fost motivul, ci a fost doar faptul ca proiectul l-am facut pentru mine, personal si imi place sa ma joc alaturi de prieteni, deci nu am avut folosinta de un mod de joc player vs AI.



5. Concluzii

In concluzie, mi-a facut multa placere sa lucrez la acest proiect. Displayul pe care am lucrat este unul de 0.96 inch dar detin un al doilea display de 1.8 tft rgb si intentionez sa-l portez si pe acela pentru a ma putea bucura de dimensiunea ecranului. Pot spune ca a fost un proiect care m-a facut sa imi placa mai mult atat sa folosesc arduino cat si partea de game developing.

Pentru explicatie video detaliata atat a video-ului, cat si a proiectului, puteti intra pe urmatoarele linkuri de pe canalul meu de YouTube:

6. Bibliografie

<https://learn.adafruit.com/adafruit-gfx-graphics-library>