

Micro:Bit Console

Dinu Dan Nicolae 333CC

Introducere

Prezentarea pe scurt a proiectului vostru:

- Proiectul Micro:Bit Game este o consolă de jocuri dezvoltată pe microcontrolerul care îi poartă și numele - Micro Bit. Această consolă vine echipată cu un set de butoane și un ecran LCD. Jocul principal al acestei console este un driving simulator dintr-o perspectiva bird's eye într-un format 2D.
- Scopul consolei este de a ține captivat utilizatorul într-un driving simulator bazat pe evitarea coliziunilor cu traficul auto-generat.
- Ideea acestui proiect a venit din dorința mea de a dezvolta propriul joc de condus și de a provoca abilitățile, cât și creativitatea utilizatorilor într-un scenariu de evitare a traficului.
- Microbit:Bit Game console servește ca o sursă de distracție și relaxare pentru pasionații de jocuri video. Acest proiect reprezintă, de asemenea, o oportunitate de a pune în aplicare cunoștințele dobândite în cadrul materiei de PM și de a înțelege mai bine modul de funcționare al microcontrolerului Micro Bit, împreună cu runtime-ul CODAL. Prin intermediul acestui proiect, îmi dezvolt abilitățile practice în acest domeniu, aplicând conceptele teoretice într-un mediu real.

Descriere generală

Consola este echipată cu un set de trei butoane care permit controlul direcțional (viraj la stânga, dreapta) și un buton pentru pauză/restart, precum și un ecran LCD pentru afișarea informațiilor de joc. Pentru dezvoltarea pe microcontrolerul Micro:Bit, mă voi folosi de API-ul CODAL, care va îmbunătăți performanțele consolei și va face experiența de utilizare mai rapidă și mai plăcută. Utilizarea CODAL permite o mai bună gestionare a resurselor hardware și optimizează răspunsul la comenzi, oferind astfel o interacțiune fluidă și intuitivă cu jocul.



Hardware Design

- Componente
 - Micro:Bit V2
 - BreadBoard Mini
 - Edge-connector breakout board
 - 3x Butoane

- Ecran RGB TFT_LCD 128×160.
- Fire
- 3x Rezistente 10k ohm
- 2x Baterii AAA 1.5V

Schema plăcii de dezvoltare MicroBit V2.



Schema proiectului realizată în TinkerCad.



Schema electronică a proiectului.



De asemenea, interacțiunea cu ecranul LCD va fi realizată prin SPI, iar frecvența va fi setată la 64 MHz. Conexiunea ecranului LCD cu placa de dezvoltare va fi realizată astfel:



Software Design

Link pentru vizionarea unui demo al proiectului:

<https://youtube.com/shorts/Q7knMvZmnyo?feature=share>

Din punct de vedere al software-ului, proiectul se folosește de runtime-ul DAL/CODAL. Acesta este un framework care oferă un strat consistent de abstractizare a interacțiunii dintre hardware și software. Codal încorporează și folosește concepte de POO pentru a reprezenta și gestiona componentele hardware (placa de dezvoltare, butoanele și ecranul).

Oferind o perspectivă asupra video-ului prezentat în demo, proiectul implementează 3 idei generale:

- Gestionarea NPC-urilor (Traficul pe care jucătorul trebuie să îl evite)
- Controlul jucătorului (Mașina controlată de jucător)
- Background-ul

Acesta este un scurt exemplu din workflow-ul proiectului folosind resursele puse la dispoziție de framework-ul CODAL.

```
if(!uBit->io.P9.getDigitalValue())
{
    paused=!paused;
    if(paused==true)
        pauseGame();
    else updateScreen(img,lcd);
        uBit->sleep(400);
}
```

Acesta este un exemplu al interacțiunii dintre hardware și software folosind abstractizarea oferită de framework-ul CODAL. "uBit" este o instanță a clasei "MicroBit" prin care pot referenția componentele hardware atașate la μC .

Întreaga afișare a jocului este creată prin umplerea pixelilor cu valorile dorite, pe anumite poziții (nu există imagini predefinite/incărcate, totul este "desenat" în cod).

```
void Game::drawBackground(ManagedBuffer buf) {
    uint16_t *p = (uint16_t *) &buf[0];
    uint16_t value=LCD_GREEN_16_BIT;
    while(p<(uint16_t *) &buf[buf.length()])
    {
        *p=value;
        p++;
    }
}
```

Această funcție primește ca parametru o instanță a clasei ManagedBuffer. Aceasta reprezintă transpunerea ecranului într-un tablou unidimensional. Astfel, folosind un pointer de unsigned 16 voi traversa acest tablou și voi adăuga valoarea (culoare) pe poziția dorită. Ulterior, datele vor fi afișate astfel:

```
void Game::updateScreen(ManagedBuffer buf,Adafruit_ST7735 *lcd) {
    lcd->sendData(0, 0, LCD_WIDTH, LCD_HEIGHT, buf.getBytes());
    uBit->sleep(500);
}
```

Adafruit_ST7735 *lcd: Acesta este un pointer către o instanță a clasei Adafruit_ST7735, care este o clasă pentru gestionarea unui afișaj. Aceasta conține metode pentru inițializarea afișajului, trimiterea de date către acesta și alte operații legate de afișaj.

Prin aceeași metodologie am "desenat" ecranul de start, pauza, background-ul (soseaua) și mașinile jucătorului, cât și ale NPC-urilor.

Funcționalitățile jucătorului sunt:

- schimbarea benzii la stânga/dreapta
- pauze game

În momentul schimbării benzii, mașina jucătorului este ștearsă și redesenată pe noua bandă. În același mod sunt gestionate mașinile NPC-urilor. Se aleg valori random pentru delay-ul spawn-ului și banda. Ulterior, acestea sunt șterse și redesenată cu un offset pentru a simula mișcarea acestora.

În cazul unei coliziuni programul va reseta jocul și va afișa ecranul de start.

Rezultate Obținute

Sunt multumit de stadiul curent al proiectului. Rezultatul este cel pe care mi l-am propus pentru moment, considerând că există o posibilitate de scalabilitate a acestui proiect atât din punct de vedere

hardware, cat si software pentru a face experienta una mult mai interactiva si mai captivanta.

Concluzii

Pe parcursul dezvoltarii acestui proiect am intampinat atat probleme legate de partea software, cat si de hardware. Desi hardware-ul nu este unul complex, intampinam pierderi de conexiune dintre fire. Astfel, in mod regulat conexiunile cu breadboard-ul necesitau mici ajustari.

Din punct de vedere al software-ului, research-ul cu privire la functionalitatea si utilizarea framework-ului CODAL a reprezentat o parte esentiala a invatarii si adaptarii la idea initiala a proiectului. Folosirea CODAL a adus multa claritate in manipularea hardware-ului pus la dispozitie de Micro:Bit, oferind o abordare familiara principiilor de POO. Principala provocare cu privire la dezvoltarea software-ului a fost "desenarea" jocului in sine, de la background pana la masina jucatorului si a NPC-urilor a necesitate precizie si rabdare in privinta determinarea pozitiilor acestora.

Cu toate aceste provocari, proiectul a oferit o oportunitate valorasa de invatare despre integrarea si utilizarea CODAL, interactiunea dintre hardware si software si explorarea unei noi tehnologii.

Download

[dinudan333cc_microbitconsole.zip](#)

Bibliografie/Resurse

Microbit CODAL:

<https://tech.microbit.org/software/runtime/>

<https://lancaster-university.github.io/microbit-docs/>

<https://github.com/lancaster-university/microbit-v2-samples>

Microbit Hardware:

<https://tech.microbit.org/hardware/schematic/>

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2024/aungureanu/dan_nicolae.dinu



Last update: **2024/05/26 13:05**