Space Invaders

- Autor: Ursu Dan-Andrei
- Grupa: 334CA

Introducere

Proiectul propune realizarea unei versiuni personalizate a celebrului joc de tip arcade Space Invaders si are scop de entertainment.

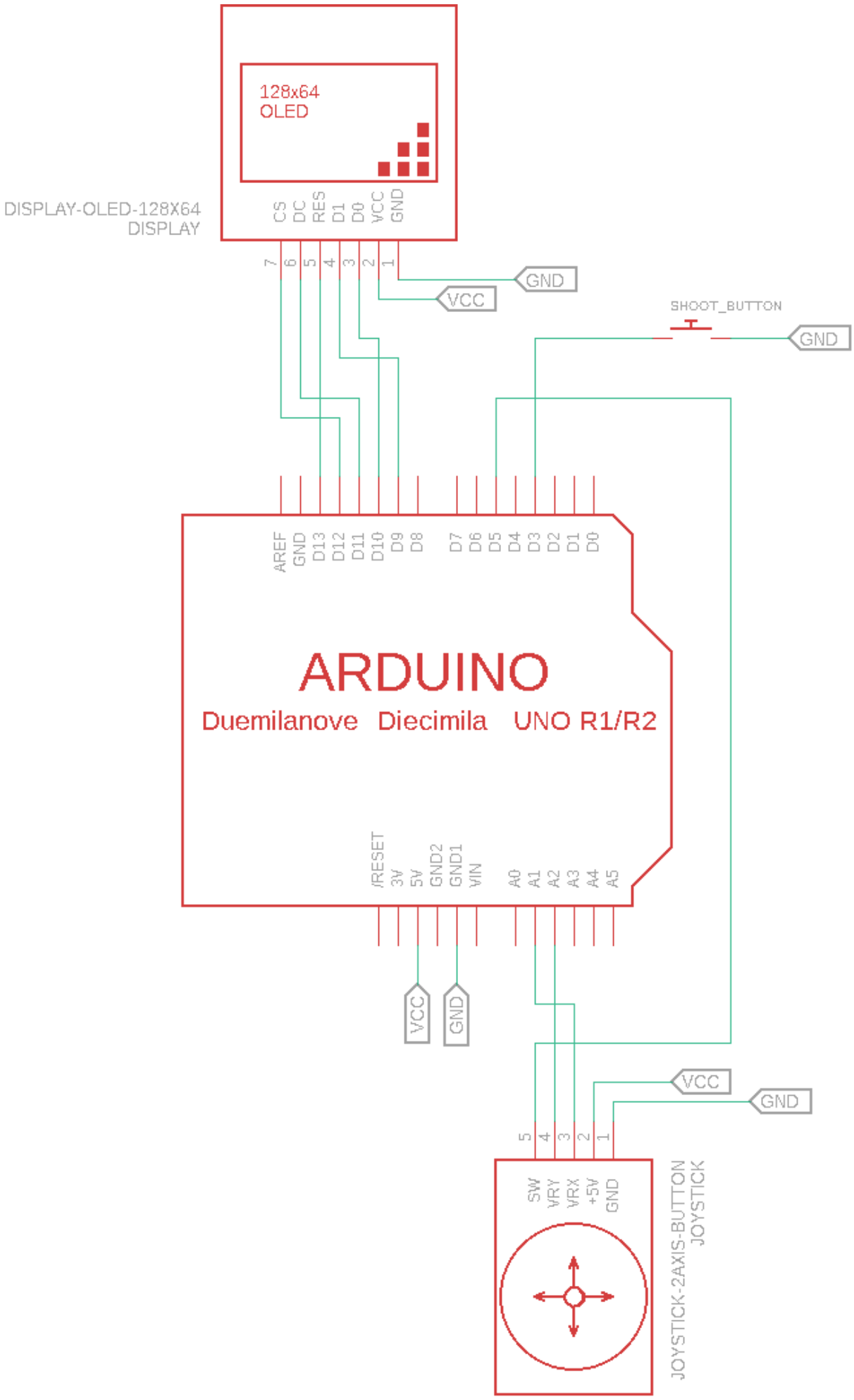
Obiectivul jocului este neutralizarea tuturor extraterestrilor prin lansarea unor proiectile inainte ca acestia sa distruga nava controlata de jucator. Se pot primi diverse power-ups pentru distrugerea unui cargo ship, precum proiectile de dimensiune marita, o viata suplimentara, extinderea razei unui scut etc.. Daca jucatorul nu mai are vieti ramase si este lovit de un extraterestru jocul se considera pierdut si se afiseaza scorul final.

Descriere generala
Schema bloc

Functionalitatea modulelor

- **Arduino Uno**: logica jocului este implementata in microprocesorul **ATmega328p**
- **Input**: 1 joystick biaxial pentru pozitionarea navei, conectat la Arduino prin 2 pini de tip analog si un pin GPIO + 1 buton pentru tragere conectat la placa printr-un pin GPIO
- **Display**: un ecran OLED ce comunica cu placa prin SPI

Hardware Design
Schema electrica


128x64
OLED
DISPLAY-OLED-128X64
DISPLAY
CS DC RES D1 D0 VCC GND
7 6 5 4 3 2 1
GND
VCC
SHOOT_BUTTON
GND
AREF GND D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
ARDUINO
Duemilanove Diecimila UNO R1/R2
/RESET 3V 5V GND2 GND1 VIN A0 A1 A2 A3 A4 A5
VCC
GND
VCC
GND
5 4 3 2 1
SW VRY VRX +5V GND
JOYSTICK-2AXIS-BUTTON
JOYSTICK

Componente

- 1x Arduino Uno
- 1x breadboard
- 1x ecran OLED SPI
- 1x button 6x6x6
- 1x joystick biaxial
- 20x fire tata-tata
- 10x fire mama-tata

Software Design
Programe utilizate pentru dezvoltare

- **Arduino IDE**: pentru incarcarea codului pe placuta si debugging
- **Sublime**: pentru realizarea implementarii codului
- **Eagle**: pentru proiectarea schemei electrice
- **draw.io**: pentru realizarea schemei bloc

Biblioteci Arduino

- Adafruit_GFX.h si Adafruit_SSD1306.h pentru transmiterea imaginilor obiectelor catre display-ul OLED
- SPI.h pentru comunicarea cu display-ul
- AxisJoystick.h si Joystick.h pentru parsarea input-ului primit de la joystick
- Button.h pentru preluarea input-ului de la shoot button, biblioteca implementeaza mecanismul de debouncing
- Vector.h pentru manipularea facila a vectorilor de obiecte

Implementare

Implementarea urmareste 5 etape principale descrise mai jos alaturi de functiile ce fac posibila realizarea acestora. Etapele 3, 4, 5 ruleaza continuu in cadrul functiei loop().

1. asigurarea comunicatiei cu perifericele hardware (*display OLED, joystick, shoot button*)
   - **initHardwareComponents()**: apeleaza functiile de initializarea specifice obiectelor responsabile de gestiunea componentelor hardware pentru stabilirea piniilor la care sunt legate perifericele, dimensiunile acestora, protocolul de comunicatie utilizat etc.
2. afisarea meniului jocului si asteptarea selectarii unei optiuni din acesta
   - **displayNewGameScreen()**
   - **waitForNewGame()**
3. initializarea si alocarea componentelor interne
   - **initVectorStorage()**: indica zona de memorie unde fiecare vector poate sa isi stocheze elementele
   - **init<*Object*>()**: stabileste pozitia si dimensiunea initiala a obiectului
4. schimbarea starii obiectelor in functie de evolutia jocului si de input-ul primit de la utilizator
   - **move<*Object*>()**: schimba poztia obiectului in functie de traiectoria pe care acesta trebuie sa o urmeze si de input-ul jucatorului
   - **maybeSpaceShipShoot()**: se verifica daca shoot button-ul a fost apasat pentru lansarea unui proiectil, caz in care se realieaza actiunea
   - **maybeAliensShipsShoot()**: fiecare nava inamica are o probabilitate de a trage ce creste invers proportional cu numarul de inamici
   - **handleProjectiles<*Object*>Collision()**: gestioneaza efectul coliziunilor asupra obiectelor
5. afisarea pe ecran

- **draw<*Object*>()**: prin intermediul unei instante a clasei Adafruit_SSD1306 se transmit ecranului OLED coordonatele pixelilor ce trebuie aprinsi pentru desenarea obiectului

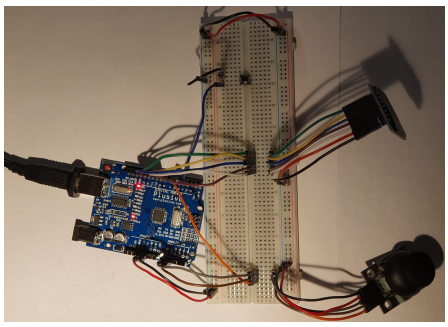
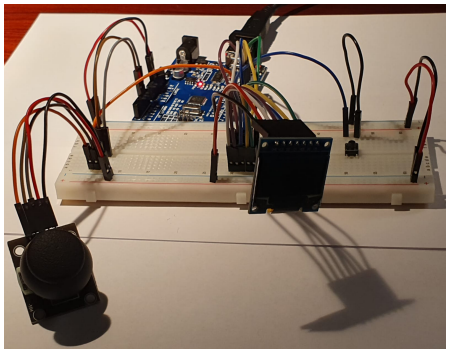
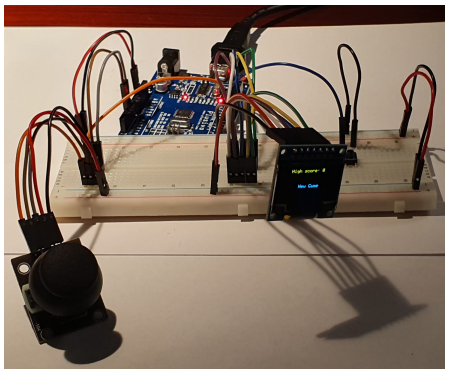
<*Object*> desemneaza instantele SpaceShip, AliensShips, CargoShip, Shields, Projectiles
Rezultate obtinute
Descreiere

Jocul este complet functional si trece toate testele pe care le-am creat pentru validarea acestuia. Dupa ce programul este incarcat pe placuta, se afiseaza un meniu de pornire si scorul maxim obtinut pana acum. Se alege optiunea new game si se incepe jocul. Initial pe ecran apare nava controlata de jucator, extraterestrii si 3 scuturi pentru protectia navei. Hitbox-ul extraterestrilor se micsoreaza cu fiecare lovitura pana ce acestia dispar de pe ecran. Un cargo ship trece ocazional prin raza de actiunea a navei cu o viteza foarte mare. Nimerirea acestuia aduce o viata suplimentara jucatorului si puncte bonus. Jocul se termina cand toti extraterestrii au fost anihilati sau cand jucatorul a ramas fara vieti.

Limitari si planuri de dezvoltare

- extraterestrii au o miscare mult prea previzibila, vreau sa realizez o traiectorie mai putin intuitiva, diferita de la extraterestru la extraterestru
- extraterestrii trag cu un singur tip de proiectile, planuiesc sa diversific armamentul acestora
- distrugerea cargo ship-ului ofera doar o viata suplimentara si puncte bonus, imi propun sa implementez si alte power-ups
- meniul de pornire ofera doar posibilitatea inceperii unui nou joc, vreau sa fac posibila si oprirea jocului curent si reluarea ulterioara a acestuia
- scorul este salvat in memorie volatila si se pierde la deconectarea alimentarii
- multe dintre limitarile descrise mai sus au fost cauzate de dimensiunea mica a memoriei RAM si frecventa redusa a microprocesorului, din acest motiv, pentru a putea extinde functionalitatiile jocului, trebuie sa achizitionez o placuta ce dispune de resurse hardware mai performante

Galerie foto

Galerie video

Demo

Link demo: https://youtu.be/7K6zVms1858

Concluzii

Desi n-am reusit sa implementez toate functionalitatiile pe care mi le-am propus initial, consider ca proiectul a fost unul reusit, la care am lucrat cu placere. Dificultatea realizarii jocului a constat in gestiunea restrictiilor impuse de memoria RAM si de capacitatea redusa de calcul a microprocesorului. Aceste limitari in care a trebuit sa ma incadrez m-au ajutat sa imi dezvolt abilitatiile de administrare a resurselor hardware limitate.

Download

- Source Code
- This page as pdf

Bibliografie si Resurse

- Arduino documentation
- Conectare display
- Adafruit_GFX.h
- Adafruit_SSD1306.h

- SPI.h
- AxisJoystick.h
- Joystick.h
- Button.h
- Vector.h
- space_invaders_original_game.gif
- Demo's soundtrack