Joc Snake

Autor

Bukkosi George-Daniel

Introducere

Am ales ca tema de proiect implementarea clasicului joc Snake pe Arduino, facand afisajul pe un LCD. Acest joc presupune controlarea unui personaj (sarpele in acest caz) astfel incat acesta sa inghita cat mai multe mancaruri de pe harta. Harta este reprezentata de display in sine si nu este marginita stanga/dreapta/sus/jos, playerul pierzand jocul in cazul in care iese din ecran. Prinderea mancarurilor de pe harta duce la cresterea in marime a sarpelui si cresterea scorului. Natural, coliziunea capului sarpelui cu o parte a corpului sau are ca rezultat tot pierderea jocului. La finalul jocului utilizatorul va putea sa isi vada punctele acumulate si sa inceapa un nou joc daca doreste prin apasarea butonului de Reset.

M-am hotarat asupra aceastei teme de proiect deoarece consider ca are o complexitate destul de buna atat din punct de vedere hardware, cat si software si reprezinta un bun punct de plecare in lucrul cu Arduino si bibliotecile sale. De asemenea, este fun de implementat!

Descriere generală

Pe post de “controller” voi utiliza un breadboard pe care voi avea patru butoane pentru directie (sus, jos, stanga, dreapta) si un buton pentru Start/Reset game. Actionarea butoanelor genereaza inputul jocului, care, prelucrat in Arduino, va afisa miscarile corespunzatoare pe display-ul ST7920 conectat ca output.

Hardware Design

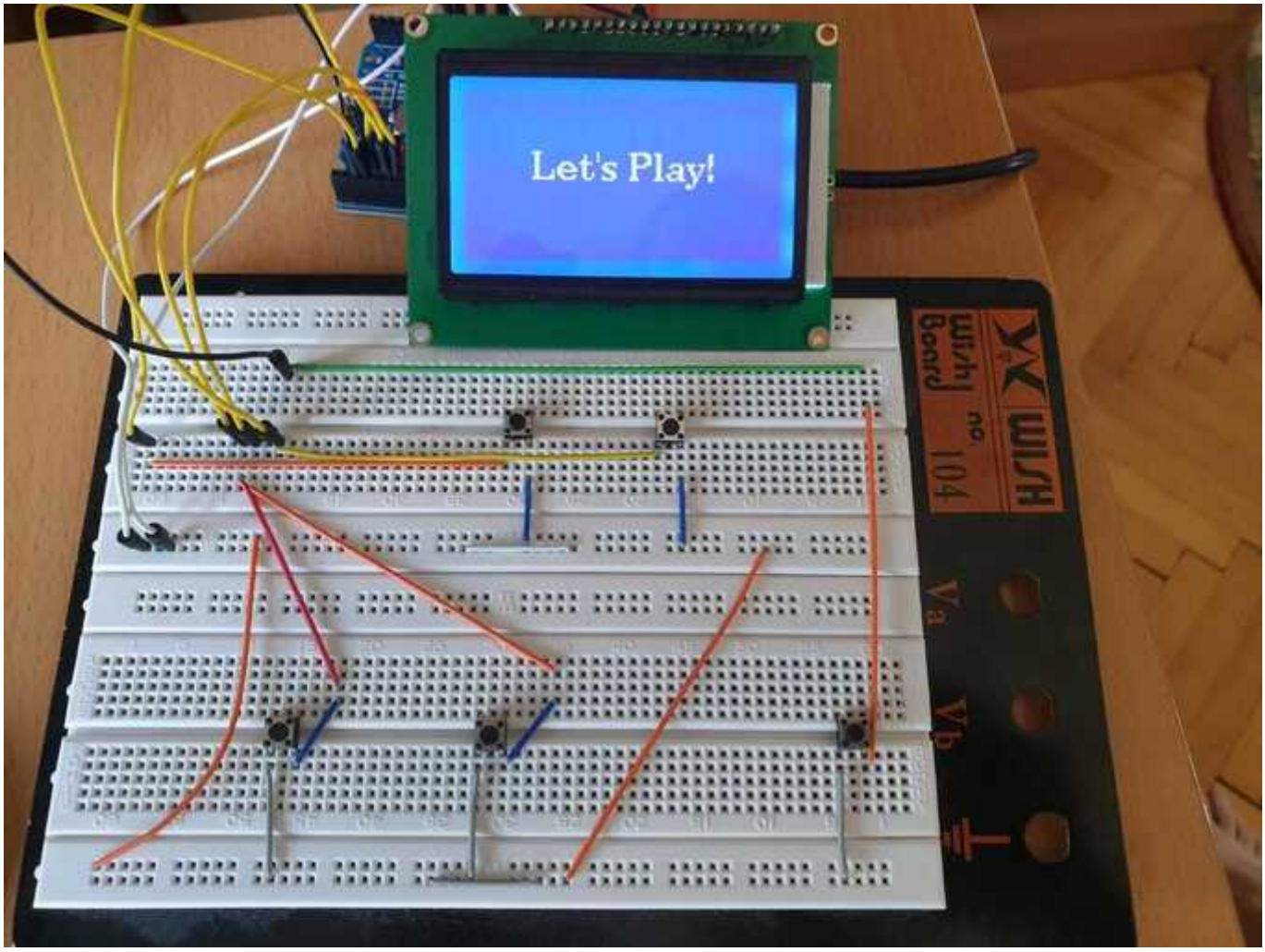
Pentru realizarea proiectului voi utiliza urmatoarele piese:

- Placa dezvoltare Arduino UNO R3

- Display ST7920
- Fire
- Rezistente
- Breadboard
- Butoane

Diagrama bloc a proiectului meu arata astfel:

Componentele conectate arata astfel:



Schema electrica este urmatoarea:

Configurarea pinilor este urmatoarea:

Butoane → Arduino:

- Buton1 → D2
- Buton2 → D3
- Buton3 → D4

- Buton4 → D5
- Buton5 → D6

LCD → Arduino:

- GND→ GND
- VCC → 5V
- RS → D10
- R/W → D11
- E → D13
- PSB → GND
- RST → D8
- BLA → 3.3V
- BLK → GND

Software Design

Biblioteci utilizate:

- U8g2lib.h
- ArduinoQueue.h
- stdlib.h

Mediul de dezvoltare utilizat a fost Arduino IDE.

Am folosit o coada (din biblioteca ArduinoQueue.h) pentru a stoca pozitiile sarpelui. Pozitia sa initiala este aleator generata. Fiecare patratel din corpul sarpelui are asociata cate o pozitie (x,y) a coltului stanga sus. La afisarea pe ecran se foloseste functia u8g2.drawBox(), care ia ca parametru coltul din stanga sus si size-ul laturii in pixeli. Pentru pozitie am creat structura Position, ce are x si y. De fiecare data cand sarpele creste, o noua pozitie este adaugata in coada.

Deoarece update-ul sarpelui vrem sa il facem independent de functia de loop() care se acceseaza de foarte multe ori pe secunda (la frecventa procesorului arduino), functia updateSnake() va fi apelata doar cand a trecut suficient timp (care este considerat variabila “speed”, deoarece modificarea sa creeaza iluzia de viteza marita a sarpelui cu cat se avanseaza in dificultate). Update-ul sarpelui la fiecare “frame” se face in functie de doua detalii: directia in care se merge si daca sarpele a mancat sau nu. Miscarea sarpelui se realizeaza prin randarea pe directia corespunzatoare a unui patratel in fata corpului deja existent si stergerea unuia de la coada. In cazul in care se intampla ca sarpele sa fi si mancat la acel frame, se randeaza doua patratele la cap.

De asemenea, la fiecare apelare a loop-ului se verifica daca exista un bait generat, iar daca nu este, se genereaza la o pozitie aleatoare pe ecran. Input-ul va fi considerat si updatat la fiecare loop(), nu la fiecare “frame”, pentru o actualizare cat mai precisa. In cazul in care nu s-a modificat directia, se va intra in loop() cu directia anterior setata.

Functia isSnakeCollision() verifica daca sarpele are o coliziune cu el insusi si intoarce un flag, care este verificat la fiecare loop().

Demo

Se poate observa cum sarpele castiga in lungime si capata viteza cu cat mananca mai mult. De asemenea, se poate vedea ca atunci cand acesta se mananca pe sine sau iese din ecran, jocul se sfarseste si se afiseaza punctajul obtinut de jucator pentru runda respectiva.

Demo video

Rezultate Obţinute

Prin realizarea acestui proiect am obtinut o “mini-consola” pe care se poate juca clasicul joc Snake. Dificultatea este una medie la primele mancaruri, sarpele miscandu-se din ce in ce mai repede pe cat se avanseaza in joc. O potentiala imbunatatire ar reprezenta-o un despawn mai rapid al momelii.

Concluzii

Am invatat sa ma documentez despre conectarea pieselor cu Arduino (in cazul de fata display-ul ST7920) si citirea inputului generat de butoanele conectate la breadboard. De asemenea, lucrul cu IDE-ul oferit de Arduino a fost o experienta placuta, iar afisarea pe display a reprezentat un mic challenge ce a necesitat de asemenea documentare.

Download

Descarca PDF

Jurnal

- **25 aprilie** – Alegerea temei de proiect si crearea paginii wiki
- **30 aprilie** – Comandarea pieselor
- **5 mai** – Lipirea firelor la display-ul ST7920 si conectarea displayului cu Arduino
- **7 mai** – Conectare butoane la Arduino si citire input
- **8 mai** – Incepere implementare software
- **14 mai** – Finalizare implementare software
- **26 mai** – Realizare video demo
- **27 mai** – Realizare schema electrica

- **31 mai** – Update final wiki si upload proiect
- **31 mai** – Prezentarea proiectului la laborator

Bibliografie si Resurse

- https://www.arduino.cc/reference/en/language/functions/random-numbers/random/ (generating a random number)
- https://www.arduino.cc/en/Tutorial/DigitalInputPullup (using the input pullup)
- https://www.youtube.com/watch?v=GsrugNJ2JXU (ST7920 tutorial)
- https://github.com/olikraus/u8g2 (u8g2 library for ST7920 display)
- https://github.com/EinarArnason/ArduinoQueue (ArduinoQueue)
- https://www.arduino.cc/reference/en/language/functions/time/millis/ (“framing” system)

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

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2021/agrigore/snake

Last update: **2021/05/31 06:21**