

Text-Based RPG

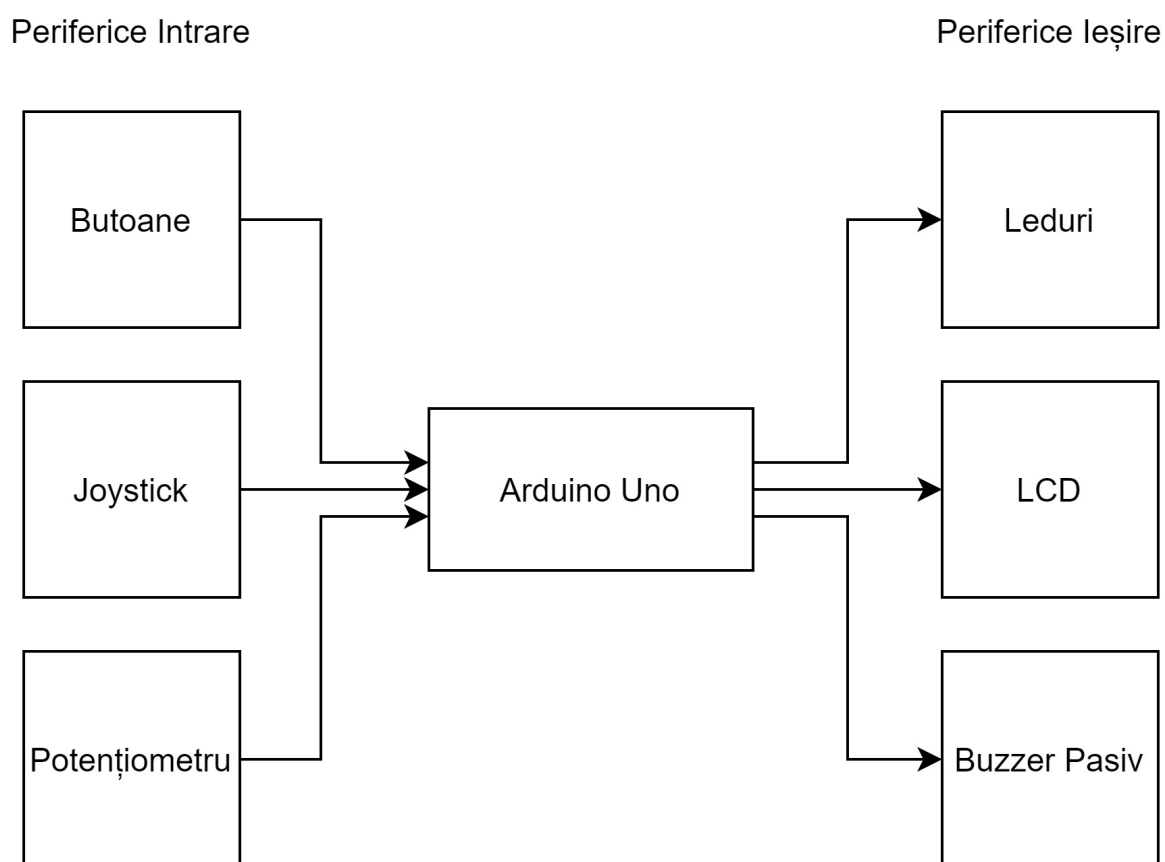
Student: Login to see contact info.

Introducere

Proiectul reprezintă implementarea unui joc RPG cu interfață text. Jocul este alcătuit din mai multe evenimente generate aleator. Jucătorul poate interacționa în mai multe feluri cu evenimentele întâmpinate, ducând la rezultate diferite.

Descriere generală

Schema bloc:



Periferice de intrare:

- Butoane - pentru acțiuni de bază, alegeri, alte funcționalități
- Un joystick - pentru text scrolling (asemănător cu mouse wheel)

- Un potențiomtru - pentru lock-picking (un minigame care poate apărea ca un eveniment aleator, necesită rotirea potențiometrului pentru a deschide un lacăt)

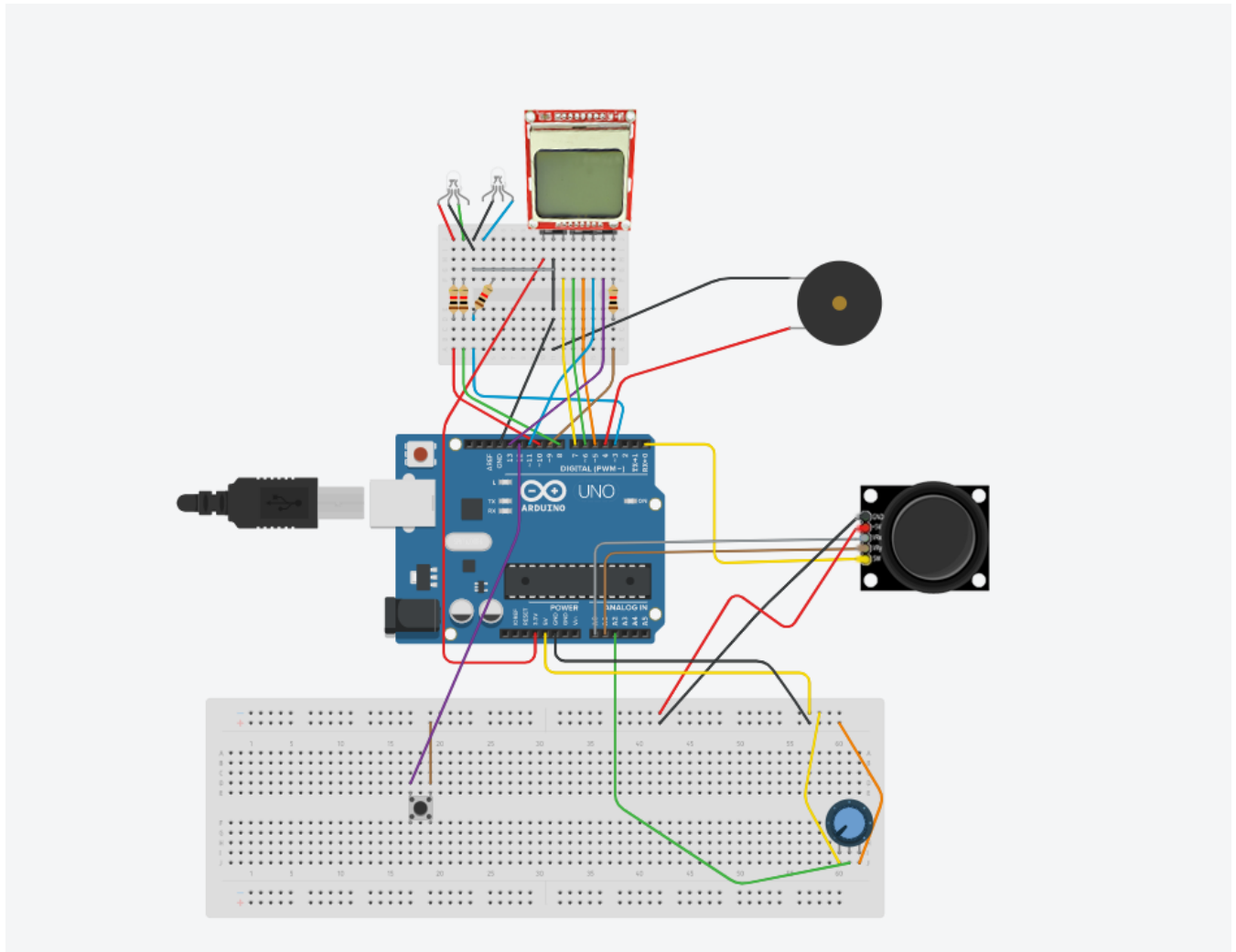
Periferice de ieșire:

- Un LED RGB care se modifică Verde→Galben→Roșu→Negru - pentru afișarea sănătății jucătorului (Health)
- Un LED RGB care se modifică Bleu→Albastru→Negru - pentru afișarea manei jucătorului (Mana)
- Un ecran LCD Nokia 5110 84×48 - pentru afișarea textului
- Un Buzzer pasiv - pentru efecte sonore (de exemplu, un efect sonor va alerta jucătorul dacă lacătul se deschide)

Hardware Design

Lista de componente:

- 1 Arduino UNO R3
- 2 LED-uri RGB
- 1 LCD Nokia 5110 84×48
- 1 Joystick
- 1 Potențiomtru
- 1 Buzzer Pasiv
- Butoane
- Rezistențe
- Fire



Software Design

IDE folosite: Arduino pentru compilare și trimitere; VSC pentru editare surse

Tutorial LCD + Biblioteca LCD_Functions.h:

- <https://learn.sparkfun.com/tutorials/graphic-lcd-hookup-guide>

Fișiere cu cod:

- Text-Based_RPG.ino - fișierul de bază, afișează logo și pornește update-urile
- Text-Based_RPG.h - apelează funcții ale claselor UI
- UIAbstract.h - UI de bază
- NSText.h - un namespace care conține un UI cu text
- NSInfo.h - un namespace care conține un UI cu informații despre caracter
- NSActions.h - un namespace care conține un UI cu meniu cu acțiuni
- NSLockpicking.h - un namespace care conține un UI cu mini-game-ul de lockpicking
- ActionManager.h - definește acțiunile și encounter-urile
- GameUnit.h - clasa unitate, care are health și mana, și funcții pentru modificarea lor
- GamePlayer.h - clasa jucător, o unitate care mai are gold și lockpicks
- GameAction.h - clasa unei acțiuni care apare în meniu, având o funcție de afișare și un id

corespunzător

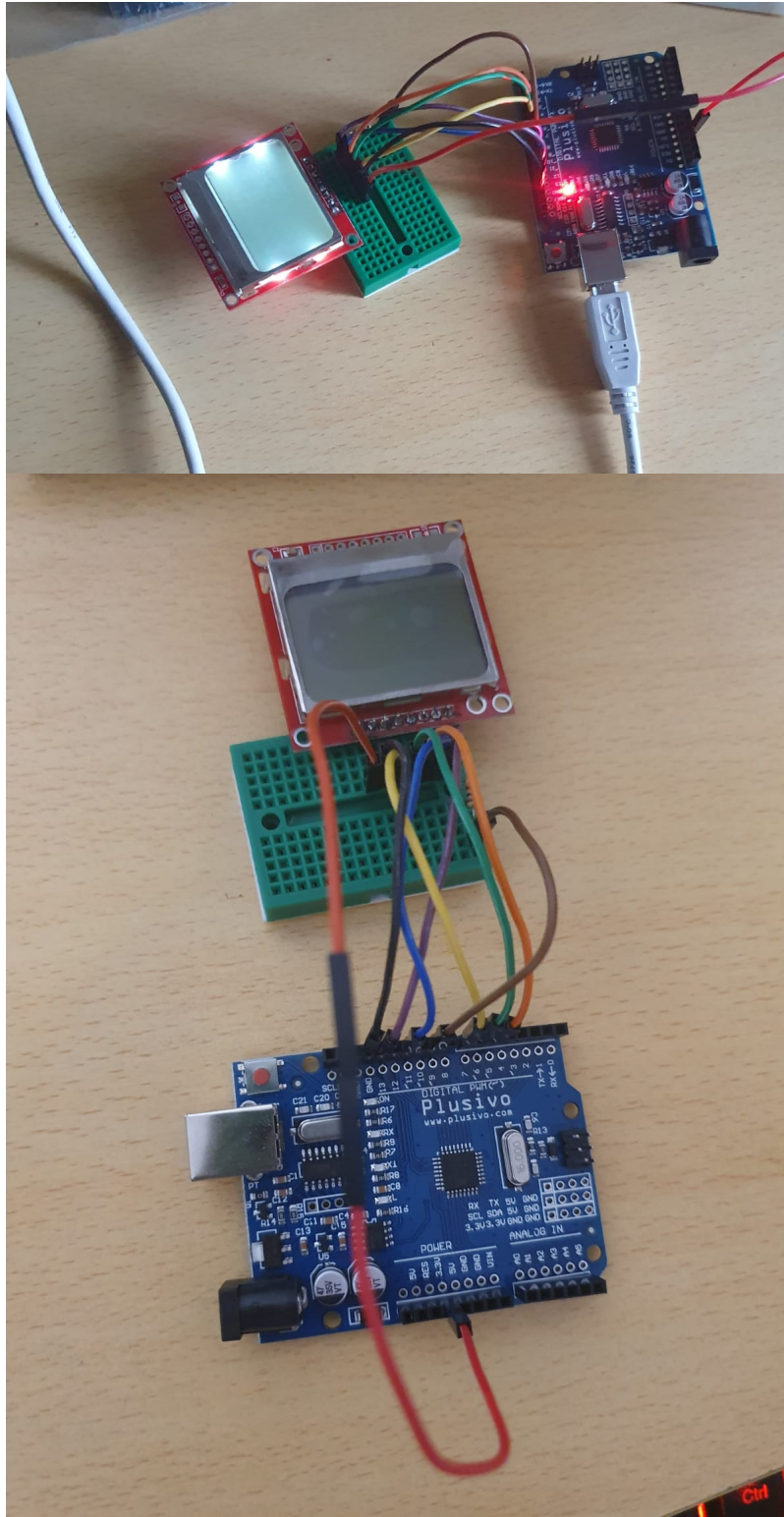
- sound_functions.h - definește mai multe funcții pentru sunete diverse
- game_data.h - definește clasa Cost și ține instanța jucătorului
- definitions.h - are define-uri pentru pini, inputuri, acțiuni și alte constante
- LCD_Manager.h - funcții pentru modificarea și afișarea unui buffer cu text pe ecran
- LCD_Functions.h - bibliotecă precizată mai sus, la care am adus mici modificări

Funcționalități

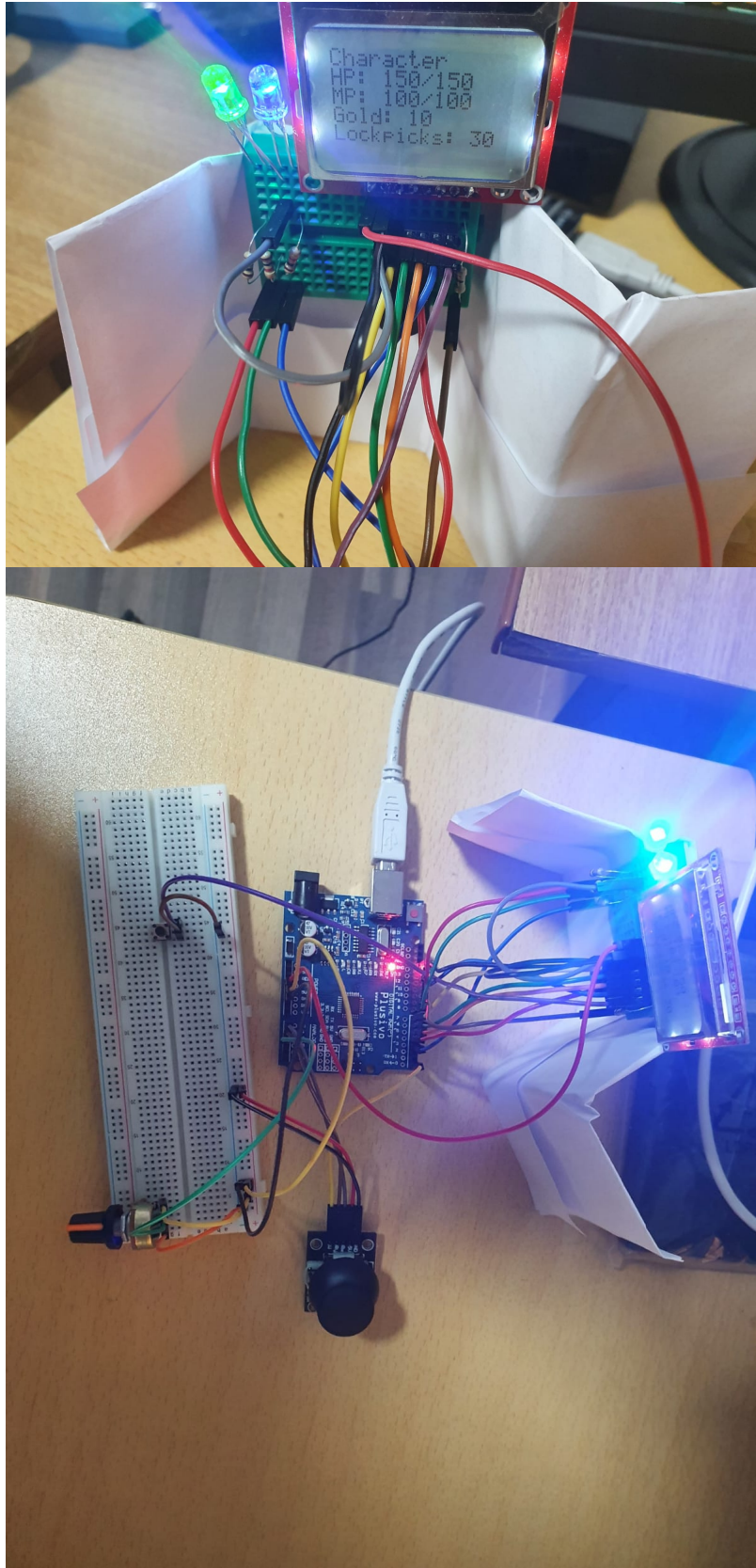
- Afișează un logo la început.
- Are 3 meniuri selectabile prin Joystick pe axa x, în mod ciclic (Aventura, Acțiuni, Caracter).
- În cadrul meniului aventură apar întâmplări aleatoare, care modifică statisticile jucătorului.
- O întâmplare poate fi mini-game-ul de lockpicking.
- În cadrul meniului acțiuni, jucătorul poate să continue aventura, să se vindece (costă mana) sau să cumpere un lockpick (costă gold)
- Prețurile acțiunilor sunt notate după semnul dolar '\$' cu H (dacă ar exista acțiuni care ar cost health), M (pentru mana), G (pentru gold) sau combinații (dacă ar exista acțiuni care ar costa mai mult de o resursă).
- Joystick-ul poate fi mutat pe axa y pentru a se face scroll la text sau pentru a selecta acțiunea dorită.
- Butonul este folosit pentru a continua aventura, a alege o acțiune sau pentru a încerca lacătul la lockpicking.
- Potențiometrul este folosit pentru a selecta căsuța în mini-game-ul de lockpicking (are valori digitalizate 0-33).
- Buzzer-ul pasiv este folosit pentru efecte sonore (bun, rău, mai bun, succes și incremental).
- Sunetul incremental (soundRange) este folosit pentru a îl ghida pe jucător către poziția corectă la lockpicking. Cu cât sunetul este mai puternic, cu atât el este mai aproape de poziția corectă.
- LED-urile RGB sunt folosite pentru a arăta sănătatea și mana jucătorului, având culorile precizate mai sus.
- LED-urile se sting în meniuri în care ele nu au rost. De exemplu în mini-game-ul de lockpicking, unde sănătatea și mana nu contează.
- PIN-ul analog A4 este folosit ca seed pentru random.
- Cheat-uri prin litere pe serială: 'l' pentru a arăta poziția corectă la lockpicking, 'u' pentru a incrementa sănătatea și mana, 'd' pentru a decrementa sănătatea și mana.

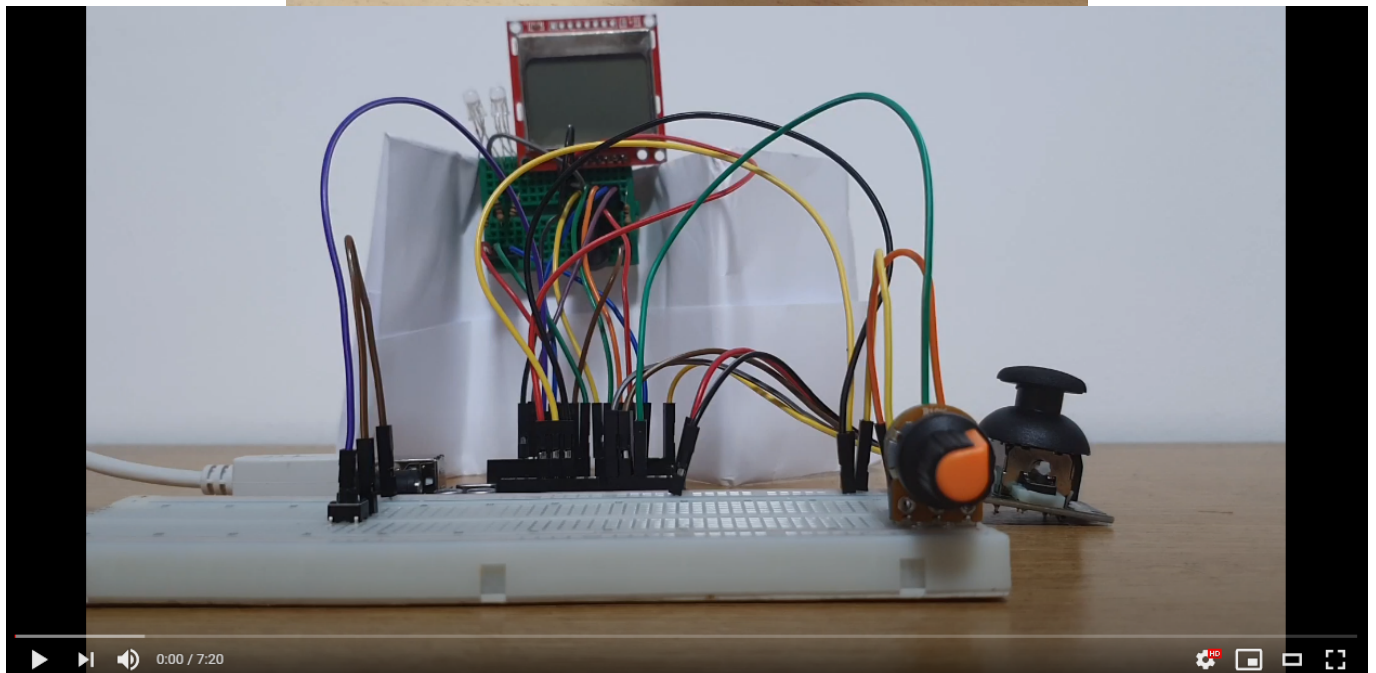
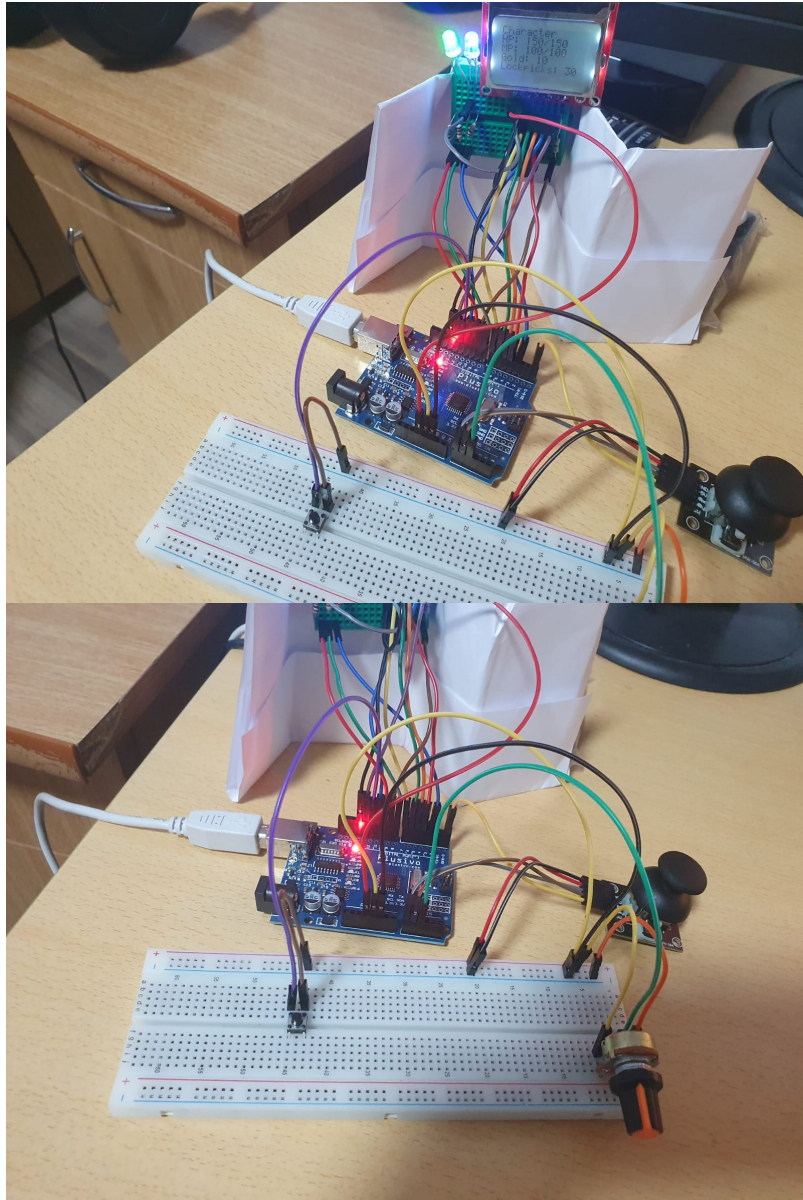
Rezultate Obținute

[Legarea plăcii cu ecranul:](#)



Jocul terminat:





Concluzii

Lucrând la acest proiect am realizat că trebuie să am foarte mare grijă la memorie. Au apărut foarte multe bug-uri în timpul implementării, cum ar fi pagini care nu apăreau în totalitate sau restartări ale plăcii. Foarte multe s-au datorat umplerii memoriei. Pentru a maximiza funcționalitățile și a minimaliza memoria utilizată am lucrat după următoarele reguli:

- 1) #define în loc de const
- 2) bufferii globali sunt reutilizați
- 3) variabilele sunt definite în blocuri
- 4) byte (unsigned char) și unsigned short în loc de int
- 5) variabilele globale și câmpurile din clasele globale folosite rar pot fi înlocuite cu funcții care le recalculează pentru a nu ocupa memorie când nu sunt utilizate (de exemplu am înlocuit UI::usesLeds cu UI::UsesLeds(), care întoarce true sau false în loc să rețină valoarea)

Download

Arhivă cu sursele: [text-based_rpg.zip](#)

Jurnal

https://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/text-based_rpg?do=revisions

Bibliografie/Resurse

LCD: <https://learn.sparkfun.com/tutorials/graphic-lcd-hookup-guide>

Joystick: <https://www.youtube.com/watch?v=B6YEQj4d5WU>

Potențiometrul: <https://www.youtube.com/watch?v=PUte1cmJ44A>

Buzzer Pasiv: <https://www.youtube.com/watch?v=dHYKRrIPNzM>

Documentația în format [PDF](#)

From:

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

Permanent link:

http://ocw.cs.pub.ro/courses/pm/prj2021/dbrigalda/text-based_rpg



Last update: **2021/06/04 07:05**