ARDUINO BULLET HELL GAME

Documentation

Съдържание:

1. Описание на проекта
2. Блокова и електрическа схема
3. Списък със съставни части
4. Описание на сорс код
5. Заключение
6. **Описание на проекта**

Проектът ми за практическия изпит на ИТ Кариера е мини-версия на ретро игра в която трябва играчът (който е точката най-долу в матрицата) да избягва патроните които играта се опитва да стреля по него с увеличаваща се скорост. Тази игра наподобява много популярните ‘bullet hell’ тип игри и затова заглавието на проекта произлиза от този игров жанр.

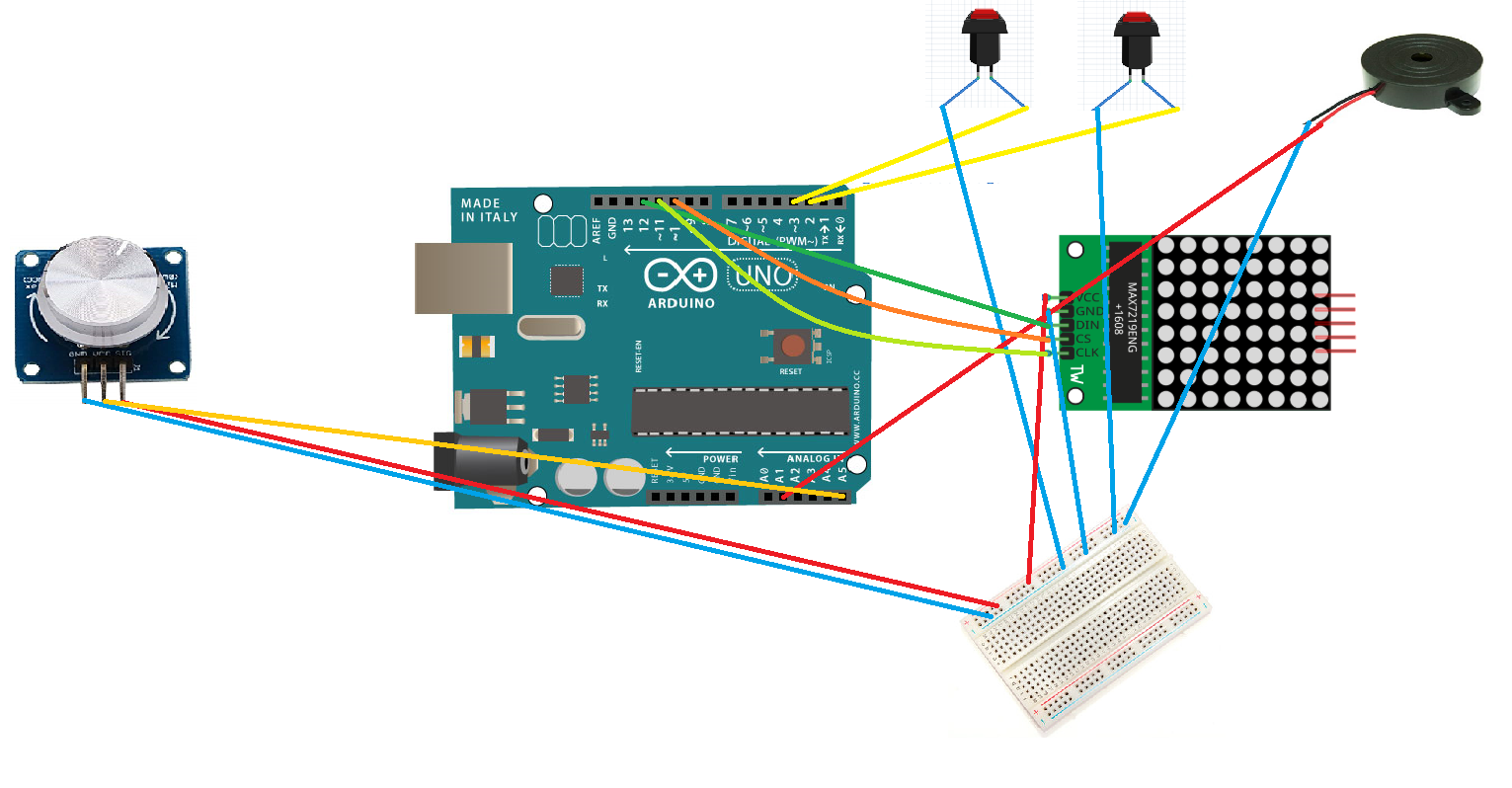
В началото на играта, играчът е оповестен за започването й чрез анимация, след което на екрана се изписва GO! *(Давай!).* Играта след това започва и патроните започват да падат сравнително бавно. Играчът точка трябва да избягва да бъде уцелен като се движи или с копчетата които са отдясно на екрана или чрез потенциометъра. Обаче колкото повече патрона играчът успява да избегне, толкова по-трудно става самото му оцеляване. Времето в играта се измерва и всеки 6 ticks, бързината на падащите патрони става по-бърза.

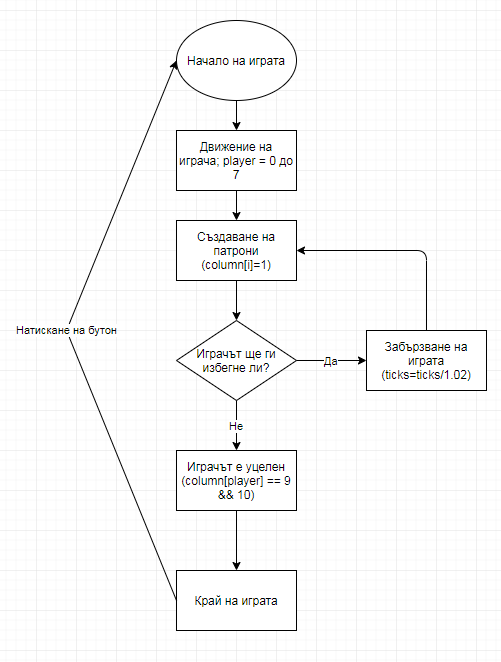
Ако играчът бъде уцелен, той експлодира с малка анимация и звук и играта приключва. Преди да бъдат разкрити точките, се представя още една анимация която се променя на базата на това колко точки е събрал по време на неговото оцеляване. След малка пауза, екранът показва точките, които за целта на измерването им, са ticks *(1 tick = 300 милисекунди = 0.3 секунди)* и тъй като матрицата не побира числа по-големи от 99, ако играча е събрал повече от 100, за всеки 100 се добавя един правоъгълник над самия му едноцифрен или двуцифрен резултат. За тези които са по-competitive, последния възможен ‘ending’ се добива на 400 точки.

По избор, играчът може свободно да започне нова игра като натисне един от двата бутона след като точките се появят на екрана след края на първата му игра.

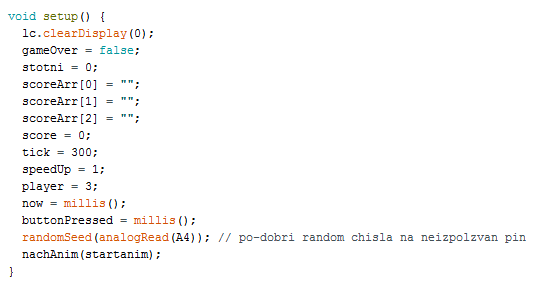
1. **Схеми – блокова и електрическа**

За този проект ми бяха нужни Ардуино Уно Р3 за управление на целия проект, бутони и потенциометър за движение на играча, зумър за издаване на различни звуци, LED 8x8 матрица MAX7219 за екран, бредборд дъска за свързване на кабели, както и USB кабел за захранване и свързване към компютър.

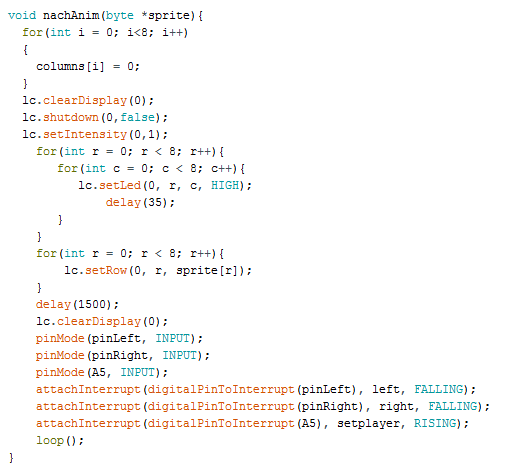


****

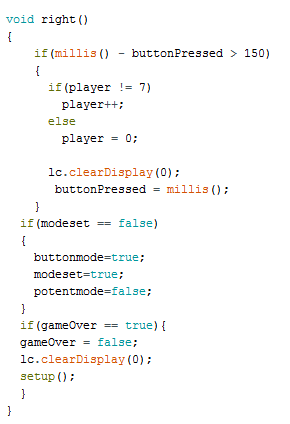
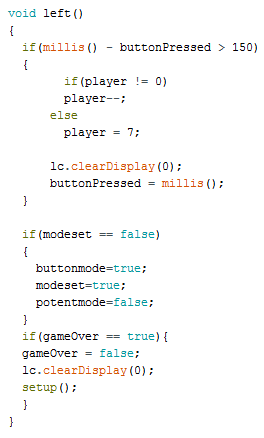
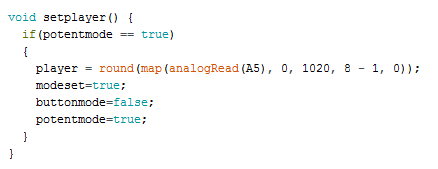
1. **Списък на съставните части**
2. 1x Arduino Uno R3
3. 1x Ротационен Потенциометър
4. 1x Зумър
5. 1x LED Матрица Макс7219 8x8
6. 2x Бутон
7. 1x Solderless Breadboard
8. ∞x Кабели (женски-мъжки и мъжки-мъжки)
9. **Описание на сорс код**

****

void setup(); започва самият проект. Слага стойност на ticks, стартово местоположение за играча, както и стартира началната анимация която след това ще започне играта. Генерират се по-добри случайни числа като се използва analogRead(); на свободен pin.

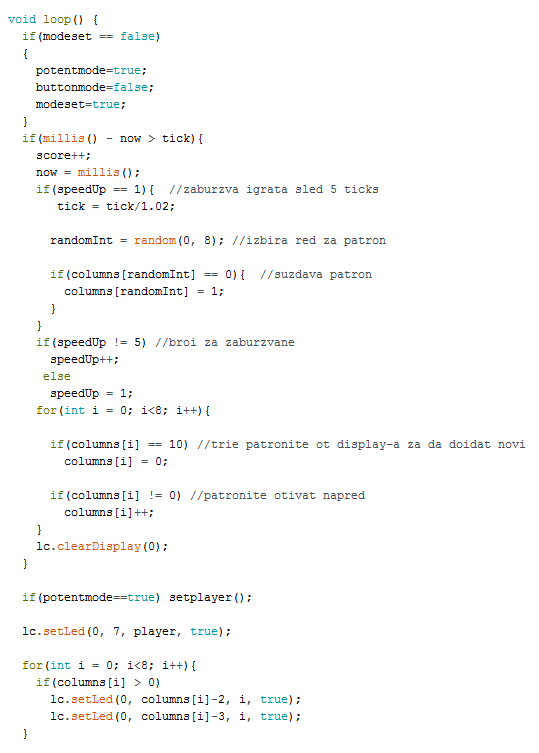


В void nachAnim(), матрицата се изкарва от режим за пестене на енергия. Започва началната анимация както и се изписва GO преди играта да започне. Настройват се бутоните и потенциометъра и започва void loop();



Тези три снимки са движението на играча чрез бутоните и потенциометъра. Бутоните имат 3 функции: преди игра ако бъдат натиснати играта ще бъде играна с тях вместо с потенциометъра. По време на игра движат играча наляво или надясно, като има проверка дали има поне 150 милисекунди между натисканията. След играта те рестартират и тя започва отново.

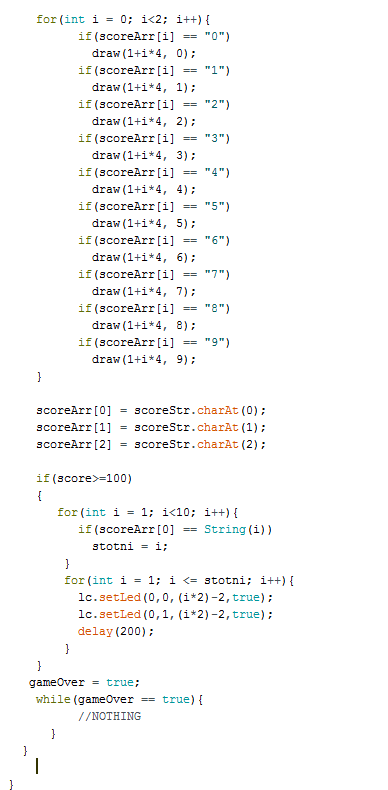
Потенциометърът се включва автоматично ако не бъде дадена команда за бутон режим в началото. Чрез map(), можем да намалим интервала който потенциометъра може да мери и да го направим да движи играча от колона на колона от 0 до 7.



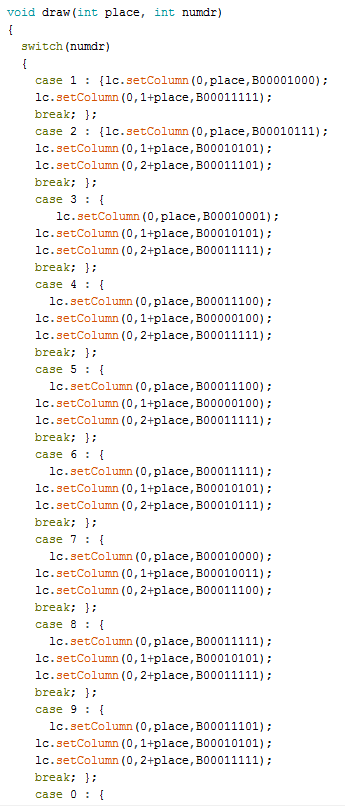
В тази част на void loop(); автоматично започва потенциометър режим ако бутон не е бил натиснат. Резултатът започва да се измерва като 1 точка на всеки tick (0.3 секунди). Играта се забързва всеки 1.5 секунди и се създава патрон в колона с случаен номер. Патронът бавно пада докато не стигне края на самата колона и не бъде изтрит. Играчът също е създаден чрез lc.setLed(0, 7, player, true);. Най-долу патроните отиват напред чрез for функция която засича всички активни патрони.



Във втората част на void loop(); се проверява дали играчът е оцелен от патрон (чрез проверка дали е под самия екран – ако той бъде уцелен ще бъде повлачен от самата програма надолу). Ако това стане, той експлодира и се чува звук. След кратка пауза започва крайната анимация която е подобна на началната, но има специална картинка която се променя ако играча събере даден брой точки. След това се изчиства дисплея, и се започва изписването на резултата. Ако резултата е трицифрен – отначало се издърпват само десетиците и единиците, ако не е, то тогава само те се изписват.



В третата част на void loop(); чертаем двуцифрения резултат като викаме функцията void draw(); (която е обяснена по-долу). Първият параметър е местоположението на числото (или започва в колона 1 или в колона 5) а вторият е самото число. След това отново се разбива самия резултат на числа и ако е над сто се проверяват стотните и се добавя правоъгълник над двуцифреното число което символизира 100 точки. Последният просто чака gameover да стане false чрез натискане на бутон.



Това е void draw();, чрез тази функция се чертаят числата чрез lc.setColumn от библиотеката LedControl. Тя чертае на 2 за единица или 3 за всичко останало колони, колона по колона докато не се изпише числото.