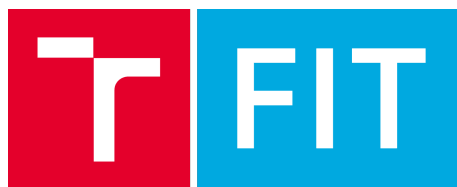


# Hra na displeji

IMP - projekt

**Autor: Tomáš Husár (xhusar11)**

Dátum: December 2023



Fakulta Informačných Technológií  
mail: [xhusar11@stud.fit.vutbr.cz](mailto:xhusar11@stud.fit.vutbr.cz)

# 1 Úvod

Cieľom bolo naprogramovať jednoduchú hru na I2C displej s ovládaním pomocou analógového joysticku. Úlohou hráča je ovládať vesmírnu loď tak, aby sa čo najdlhšie vyhýbala okolitým asteroidom. Každým úspešne obídeným asteroidom sa navyšuje výsledné skóre.

## 2 Využité periférie

- WEMOS D1 R32 w/ESP32
- Joystick
- SSD1306 I2C display
- Breadboard (vhodný k lepšiemu zapojeniu)
- USB - Micro USB kábel (napájanie a prenos dát)

## 3 Technológie

- Programovacie prostredie - Arduino IDE (verzia 2.2.2)
- Knížnice - Wire.h, U8g2lib.h

## 4 Inštalácia

V programovacom prostredí Arduino, je potrebné nainštalovať rozšírenie pre dosky esp32[1]. Pre jednoduchšie narábanie s displejom bola použitá knižnica U8g2[4].

Po inštalácii týchto nástrojov je potrebné si zvoliť konkrétnu dosku (WEMOS D1 R32), a určiť port, cez ktorý je doska pripojená. Následne stačí nahráť dáta na toto zariadenie.

## 5 Popis hry

Hra sa spúšťa hneď po zapojení do prúdu. Užívateľ (hrajúca osoba) pomocou joysticku ovláda svoje vesmírne plavidlo, a vyhýba sa asteroidom pohybmi doprava a doľava (pričom menší pohyb páčkou znamená pomalší pohyb). Za každé vesmírne teleso, ktorému sa hráč vyhne, obdrží jeden bod (ktorých maximálny dosiahnuteľný počet je 9999). Hráč začína s 3 bodmi životov, a pri zrážke s asteroidom prichádza o 1 z nich. Po krátkej časovej odozve sa dostane na počiatočné miesto na obrazovke, a pokračuje v hraní. Pri strate posledného života sa hra ukončí a zobrazí sa obrazovka signalizujúca koniec hry a dosiahnuté skóre. Po niekoľkých sekundách sa hra reštartuje a hráč môže pokračovať v hraní.

## 6 Implementácia

Pred spustením hlavného cyklu sa inicializujú hodnoty potrebné ku komunikácii jednotlivých periférií, vykreslovaniu a ovládaniu.

Na začiatku sa inicializuje premenná `time_now` pomocou funkcie `millis()`, táto premenná je ďalej využívaná k dosiahnutiu čakania v hlavnom cykle, pre dosiahnutie lepšej odozvy počas hrania.

Následne sa prečíta hodnota naklonenia joysticku, a pohne sa vesmírna loď a asteroidy (`asteroid_movement()`, `spaceship_movement()`). Funkcia `asteroid_movement()` je taktiež zodpovedná za reset asteroidu (premiestnenie asteroidu znovu na vrch obrazovky po tom čo dosiahol jej spodok). Po resetnutí asteroidu sa hráčovi pripočíta jeden bod. Rýchlosť hráča je určená podľa hodnoty prečítanej z joysticku.

Po pohnutí vesmírnych objektov nastane kontrola kolízií (`collision_check()`). Ak kolízia nastala, odoberie sa hráčovi život, a loď spolu s asteroidmi sa po krátkej odozve premiestnia na počiatočné pozície.

Po kolízií s asteroidmi sa kontroluje počet životov hráča (`game_over_check()`), ak dosiahol nulu, vykreslí sa obrazovka konca hry s dosiahnutým skóre (`game_over_print()`), a po určitej časovej odozve sa reštartuje hra (`game_restart()`).

Ak sa hráč asteroidom vyhol úspešne, vykreslí sa momentálny stav hry (`game_display()`). Vykresľuje sa počet životov (`hp`), skóre (`score`), vesmírna loď (`spaceship_bitmap[]`), asteroidy (`asteroid_bitmap[]`).

Funkcionalita pohybu joysticku bola inšpirovaná návodom Arduino Joystick PS2[2]. Výpis na displej bol inšpirovaný návodom OLED displej 0,96 palce[3].

## Reference

- [1] arduino-esp32. Arduino core for the ESP32. <https://github.com/espressif/arduino-esp32>.
- [2] Luboš M. Arduino Joystick PS2. <https://navody.drtek.cz/navody-k-produktum/arduino-joystick-ps2.html>.
- [3] Luboš M. OLED displej 0,96 palce. <https://navody.drtek.cz/navody-k-produktum/oled-displej-ssd1306.html>.
- [4] olikraus. U8g2: Library for monochrome displays, version 2. <https://github.com/olikraus/u8g2>.