

PROJEKT

WIZUALIZACJA DANYCH SENSORYCZNYCH

Gra zręcznościowa „Straszna Epidemia”

Kewin Gałuszka, 241624



Prowadzący:
dr inż. Bogdan Kreczmer

Katedra Cybernetyki i Robotyki
Wydziału Elektroniki
Politechniki Wrocławskiej

31 marca 2021

Spis treści

1	Charakterystyka tematu projektu	1
1.1	Cel projektu	1
1.2	Fabula gry	1
2	Podcele i etapy realizacji projektu	1
3	Specyfikacja finalnego produktu	1
4	Terminarz realizacji poszczególnych podcelów	2
4.1	Kamienie milowe	2
4.2	Diagram Gantta	3
5	Ewaluacja	3
5.1	Przegląd literatury i zasobów Internetu związanych z tematem	3
5.2	Projekt graficznego interfejsu użytkownika	4
5.3	Obsługa żyroskopu lub/i akcelerometru na płycie deweloperskiej	5
5.4	Opracowanie i uruchomienie transmisji danych między kontrolerem gry a komputerem	6
5.4.1	CDC – wirtualny port COM	6
5.4.2	Przesyłanie danych	6
5.4.3	Suma kontrolna CRC8	7
5.5	Logika gry – implementacja postaci	8
5.6	Obsługa poziomu życia gracza, integracja części sprzętowej z oprogramowaniem, dodanie grafik i efektów dźwiękowych	9
5.6.1	Mechanika gry i dodatkowe funkcjonalności	9
5.6.2	Rysowanie wykresów	9
5.6.3	Dodanie grafik	10
5.6.4	Dodanie efektów dźwiękowych	11
5.6.5	Zmiany w projekcie	11
5.7	Podsumowanie	12
6	Bibliografia	14

1 Charakterystyka tematu projektu

1.1 Cel projektu

Celem projektu jest stworzenie gry zręcznościowej. Jako kontroler zostanie wykorzystana płytk deweloperska STM32H411VET6 wyposażona w żyroskop i akcelerometr. Dane odczytywane z akcelerometru lub/i żyroskopu będą prezentowane na ekranie jako ruch obiektu, którym steruje użytkownik. Zadaniem gracza jest trafienie w cel za pomocą pocisku.

1.2 Fabuła gry

Postacią gracza jest lekarz – bohater, który za pomocą swojej broni i wystrzeliwanych przez nią pocisków może zestrzelić lecące w jego stronę wirusy. Celem gry jest zestrzelenie jak największej ilości wirusów i niedopuszczenie do przedostania się ich poza planszę

2 Podcele i etapy realizacji projektu

- Przegląd literatury i zasobów Internetu związanych z tematem
- Ukończenie wstępnego projektu interfejsu graficznego
- Obsługa żyroskopu lub/i akcelerometru na płytce deweloperskiej
- Opracowanie i uruchomienie transmisji danych między kontrolerem gry a komputerem
- Implementacja postaci gracza
- Dodanie celów
- Dodanie funkcjonalności strzelania
- Obsługa poziomu życia gracza
- Integracja części sprzętowej z oprogramowaniem
- Dodanie grafik oraz efektów dźwiękowych
- Wprowadzenie ostatecznych poprawek do projektu

3 Specyfikacja finalnego produktu

Oczekiwany efekt jest utworzenie gry zręcznościowej, której sterowanie będzie możliwe za pomocą zmiany położenia płytki deweloperskiej. Sterowanie gry powinno być proste w nauce, interfejs powinien być przejrzysty.

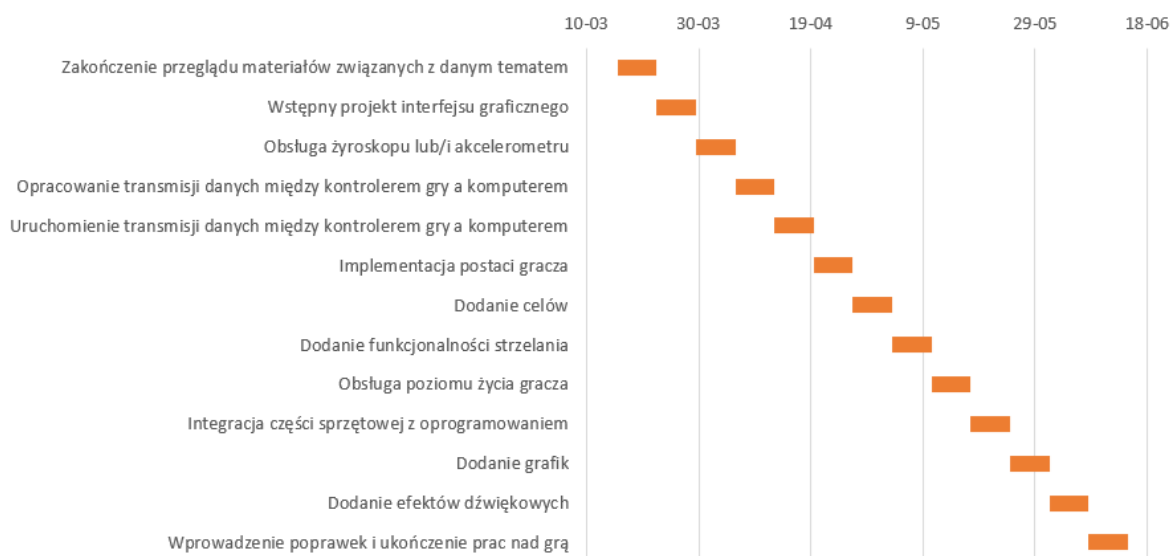
4 Terminarz realizacji poszczególnych podcelów

- 23 marca 2020 – zakończenie przeglądu materiałów związanych z danym tematem
- 30 marca 2020 – wstępny projekt interfejsu graficznego
- 6 kwietnia 2020 – obsługa żyroskopu lub/i akcelerometru
- 13 kwietnia 2020 – opracowanie transmisji danych między kontrolerem gry a komputerem
- 20 kwietnia 2020 – uruchomienie transmisji danych między kontrolerem gry a komputerem
- 27 kwietnia 2020 – implementacja postaci gracza
- 4 maja 2020 – dodanie celów
- 11 maja 2020 – dodanie funkcjonalności strzelania
- 18 maja 2020 – obsługa poziomu życia gracza
- 25 maja 2020 – integracja części sprzętowej z oprogramowaniem
- 1 czerwca 2020 – dodanie grafik
- 8 czerwca 2020 – dodanie efektów dźwiękowych
- 15 czerwca 2020 – wprowadzenie poprawek i ukończenie prac nad grą

4.1 Kamienie milowe

- Uruchomienie komunikacji między kontrolerem a komputerem
- Ukończenie pracy nad grą – sterowanie za pomocą klawiatury komputera
- Integracja gry z częścią sprzętową – obsługa kontrolera
- Dodanie grafik oraz efektów dźwiękowych

4.2 Diagram Gantta



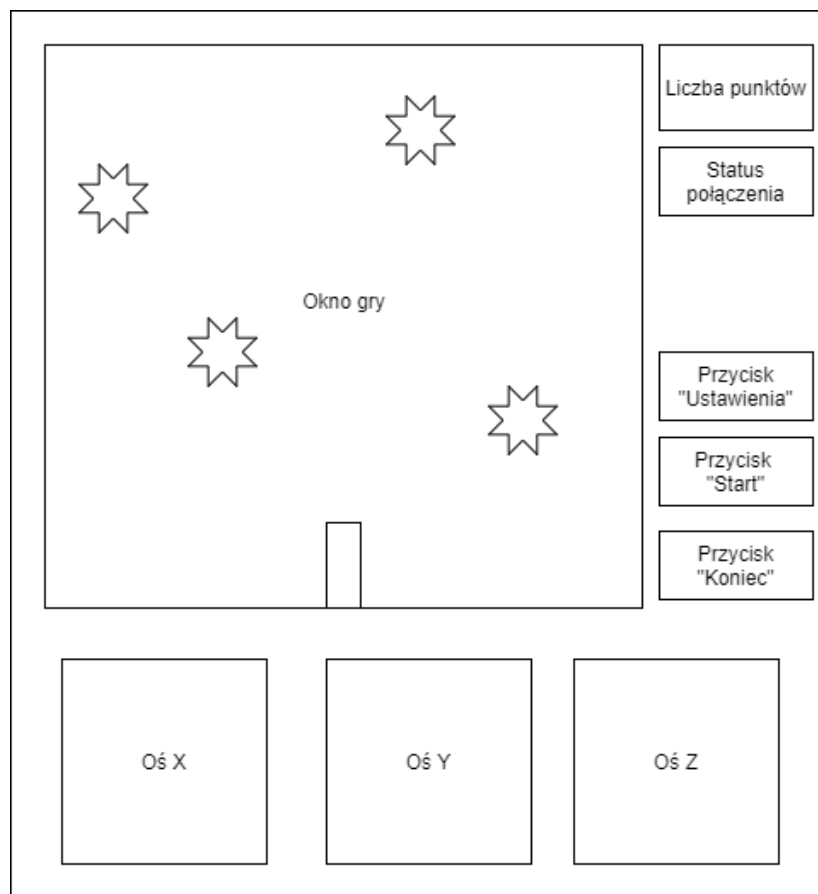
Rysunek 1: Diagram Gantta

5 Ewaluacja

5.1 Przegląd literatury i zasobów Internetu związanych z tematem

Efektem przeglądu literatury i zasobów Internetu związanych z tematem jest stworzenie pierwszej części dokumentacji projektu. Ponadto, zapoznano się z etapami budowy gry, co umożliwiło sprawne opracowanie planu realizacji projektu. Dokonano również wstępnego zapoznania się ze sposobami komunikacji kontrolera z komputerem.

5.2 Projekt graficznego interfejsu użytkownika



Rysunek 2: Projekt graficznego interfejsu użytkownika

Graficzny interfejs użytkownika, zgodnie z założeniami, powinien być intuicyjny i prosty w obsłudze (patrz rys. 2) .

Interfejs zawiera:

- Przycisk „*Start*” rozpoczynający grę
- Przycisk „*Koniec*” kończący grę
- Przycisk „*Ustawienia*” otwierający okno ustawień
- Okno „*Liczba punktów*” w którym znajdzie się aktualna liczba uzyskanych przez gracza punktów
- *Okno gry*, w którym wyświetlana będzie rozgrywka
- Widżety osi OX, OY, OZ w których wyświetlane będą wykresy odnotowanych wartości przyspieszenia poszczególnych osi odczytanych przez sensor.
- Widżet *Status połączenia*, w którym wyświetlana będzie informacja o połączeniu z płytą

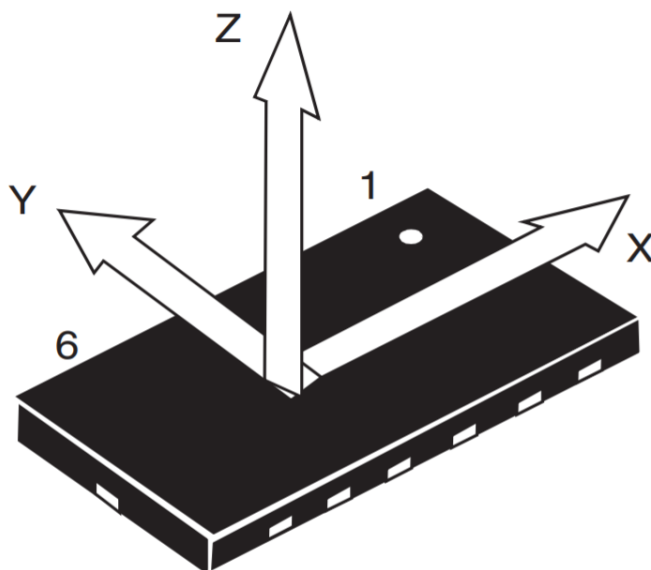
Po naciśnięciu przycisku „*Start*” rozpocznie się rozgrywka. W oknie gry wyświetlona zostanie postać gracza (lekarz) oraz wrogowie (wirusy). W trakcie gry graczowi jest udostępniona możliwość podglądu liczby uzyskanych punktów w oknie „*Liczba punktów*”. Po naciśnięciu przycisku „*Koniec*” w oknie gry zostanie wyświetlony wynik uzyskany przez gracza i gra zostanie zakończona.

Naciśnięcie przycisku „*Ustawienia*” spowoduje wyświetlenie okna, w którym będzie możliwość ustawienia podstawowych ustawień komunikacji z kontrolerem. W sekcji *Status połączenia* wyświetlany będzie aktualny stan połączenia z płytą deweloperską.

Poniżej gry umieszczono wykresy odnotowanych wartości przyspieszenia poszczególnych osi odczytanych przez sensor - widzety osi OX, OY, OZ.

5.3 Obsługa żyroskopu lub/i akcelerometru na płycie deweloperskiej

W projekcie wykorzystany zostanie akcelerometr znajdujący się na płycie deweloperskiej STM32F411 Disco. Jest nim układ LSM303DLHC, który zawiera trzyosiowy akcelerometr oraz magnetometr (patrz rys. 3).



Rysunek 3: Interpretacja osi układu LSM303DLHC

Komunikacja mikrokontrolera STM32F411 z układem akcelerometru odbywa się z użyciem magistrali I2C. Konfiguracja peryferium I2C została przedstawiona w poniższej tabeli (patrz tab. 1).

Parametr	Wartość
I2C Speed Mode	Fast Mode
I2C Clock Speed (Hz)	400000
Primary Address Length selection	7-bit

Tabela 1: Konfiguracja peryferium I2C1

5.4 Opracowanie i uruchomienie transmisji danych między kontrolerem gry a komputerem

5.4.1 CDC – wirtualny port COM

Transmisja danych pomiędzy kontrolerem gry a komputerem zrealizowana jest za pomocą interfejsu *Universal Serial Bus*. Peryferium zostało skonfigurowane zgodnie z poniższymi parametrami.

Parametr	Wartość
Mode	Device_only
Speed	Device Full Speed 12MBit/s

Tabela 2: Konfiguracja peryferium USB_OTG_FS

Communication Device Class definiuje za jakie urządzenie może podawać się kontroler USB. W tym przypadku zostanie wykorzystana klasa, która umożliwi stworzeniu wirtualnego portu COM.

5.4.2 Przesyłanie danych

Separatorem kolejnych danych są spacje. Przesył danych odbywa się według poniższego schematu

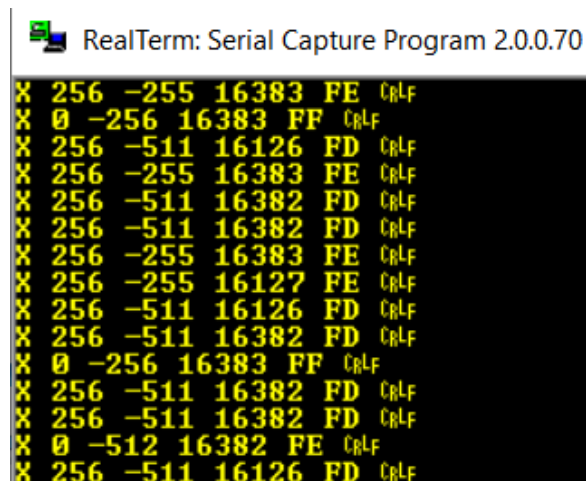
Typ danych	char	int	int	int	char	char
Pole	nagłówek	acc_ax_x	acc_ax_y	acc_ax_z	CRC	stopka
Przykład	X	500	2300	2000	8F	\r \n

Tabela 3: Ramka

Opis pól:

- *nagłówek* – sygnał początku ramki 'X')
- *acc_ax_x*, *acc_ax_y*, *acc_ax_z* – wyniki pomiaru przyspieszeń danych osi (nieprzekonwertowane na jednostkę *g*)
- *CRC8* – cykliczna suma kontrolna
- *stopka* – zakończenie ramki (sekwencja <LF><CR>)

Na poniższej grafice (patrz rys. 4) przedstawiono odbierane dane. Użyto programu RealTerm. Obserwowanym portem jest imitowany port RS232. Dla zwiększenia czytelności dodano znak końca linii w ramce danych.



Rysunek 4: Odbierane dane

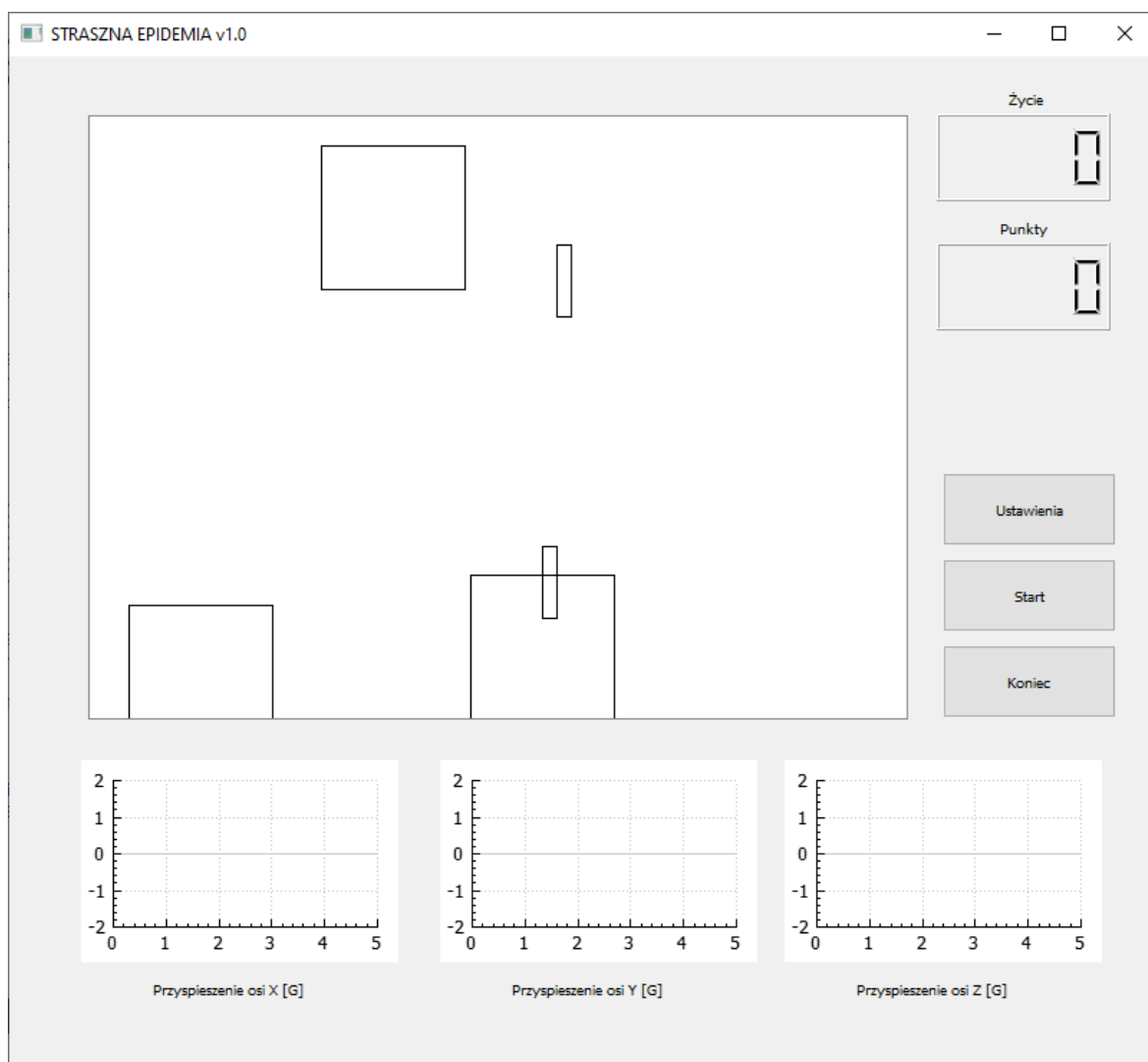
5.4.3 Suma kontrolna CRC8

Cykliczna suma kontrolna CRC8 będzie zapisywana w heksadecymalnym systemie liczbowym. Dzięki niej dodana zostanie możliwość sprawdzenia, czy w przesłanej ramce nie pojawiły się przypadkowe błędy.

5.5 Logika gry – implementacja postaci

Zaimplementowano postać gracza, postać przeciwnika (celu) oraz dodano możliwość oddawania strzału. Cele na planszy pojawiają się w losowej współrzędnej osi OX i ujemnej współrzędnej osi OY, co powoduje płynne pojawianie się przeciwnika. Gdy pocisk uderzy w przeciwnika, przeciwnik oraz pocisk znikają.

Okno gry umieszczono w odpowiednim miejscu, interfejs graficzny (patrz rys. 5) zawiera również zaprojektowane elementy, które zostaną wykorzystane przy realizacji kolejnych etapów.

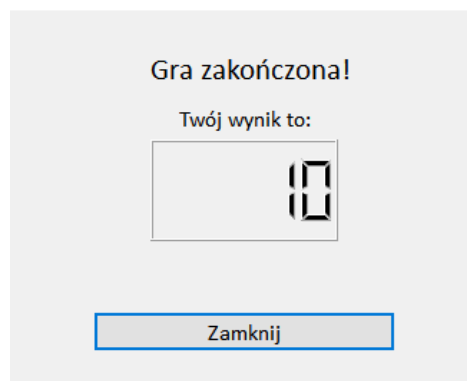


Rysunek 5: Widok okna gry

5.6 Obsługa poziomu życia gracza, integracja części sprzętowej z oprogramowaniem, dodanie grafik i efektów dźwiękowych

5.6.1 Mechanika gry i dodatkowe funkcjonalności

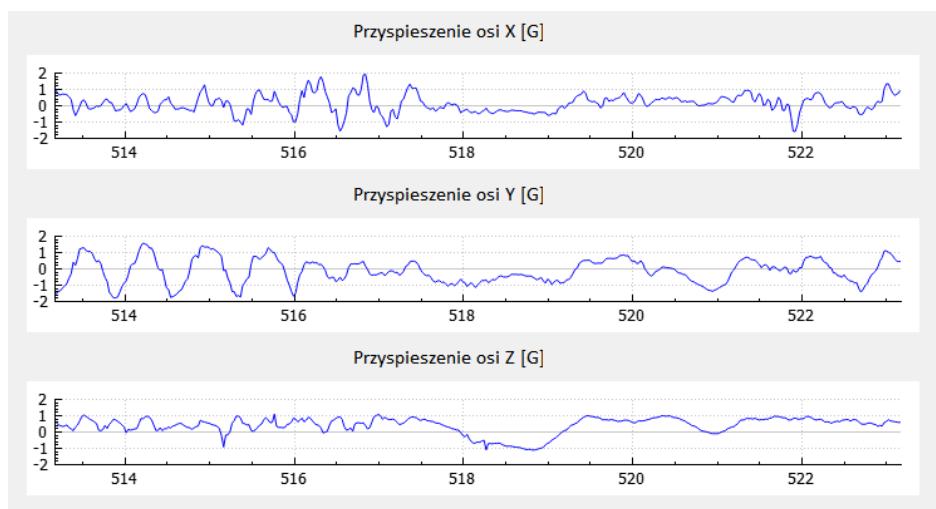
Wprowadzono możliwość zliczania punktów zdobytych przez gracza oraz punktów życia. Wyniki rozgrywki w trakcie jej trwania są wyświetlane na dwóch wyświetlaczach typu *LCD_display*. Po wciśnięciu przycisku *Start* w polu *Życie* ustalana jest początkowa wartość 5, w polu *Punkty* poprzedni wynik jest usuwany i pojawia się wartość 0. Po wciśnięciu przycisku *Pauza* istnieje możliwość wstrzymania gry. Powrót do gry możliwy jest przy wciśnięciu przycisku *Pauza*. Po wciśnięciu przycisku *Koniec* wyświetlane jest okno końcowe (patrz rys. 6), na którym przedstawiany jest ostateczny wynik uzyskany przez gracza. Niniejsze okno wyświetlane jest również, gdy liczba punktów życia spadnie do 0.



Rysunek 6: Widok okna końcowego gry

5.6.2 Rysowanie wykresów

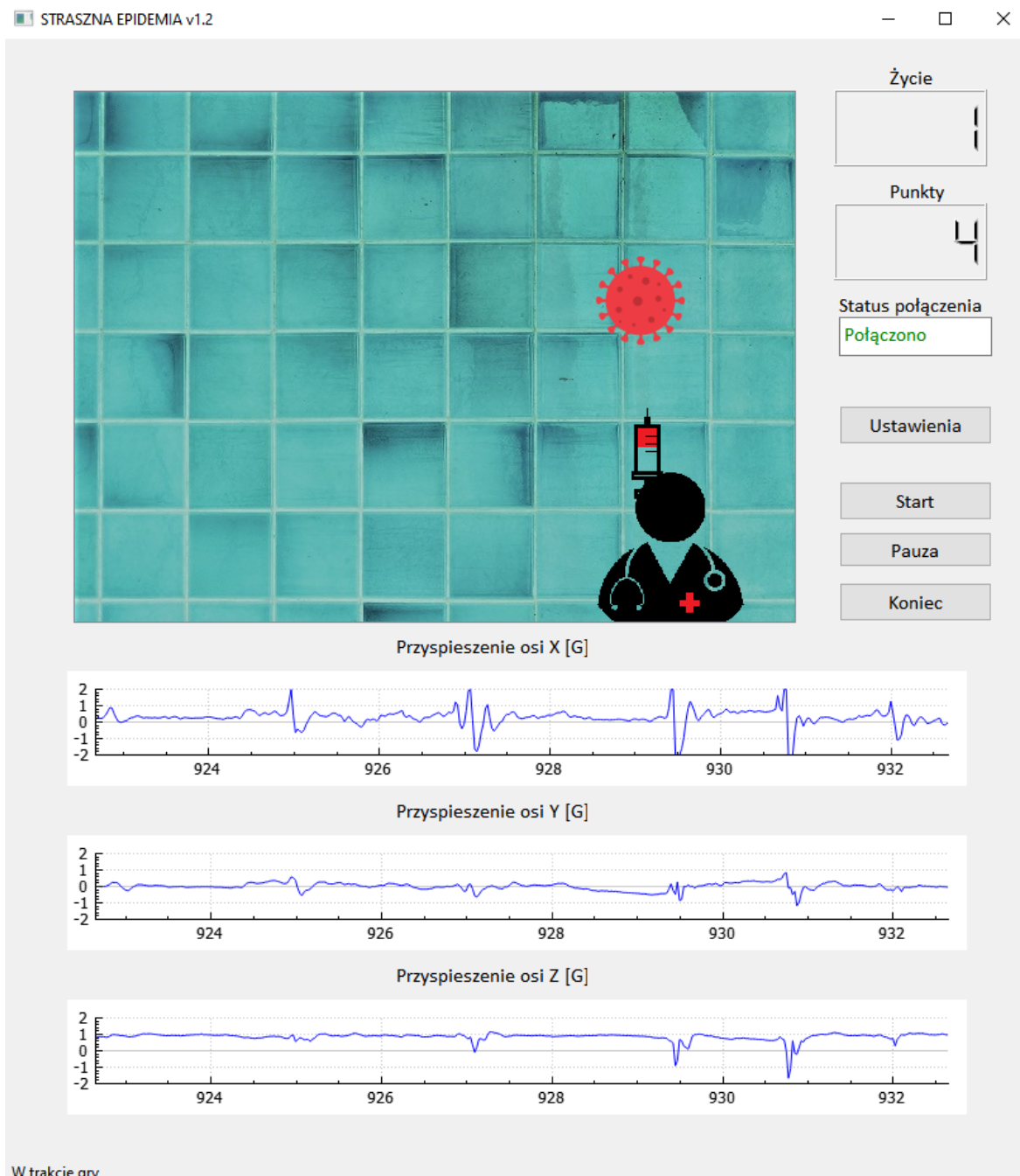
Uruchomiono funkcjonalność rysowania wykresów (patrz rys. 7). Dzięki zmianie ich ułożenia istnieje możliwość bardziej intuicyjnej obserwacji danych.



Rysunek 7: Wykresy danych odczytanych z akcelerometru

5.6.3 Dodanie grafik

Dodano grafikę gracza, pocisku (strzykawki) oraz przeciwnika (wirusa). Wszystkie pliki obrazów są w formacie *.PNG* i nie posiadają tła. Dzięki temu kształt obiektu jest dopasowany do kształtu obrazu. Dodano również grafikę tła (patrz rys. 8).



Rysunek 8: Widok okna głównego gry

5.6.4 Dodanie efektów dźwiękowych

Dodano efekty dźwiękowe podczas oddania strzału. Dodatkowo, dodana została muzyka w tle, która uprzyjemnia rozgrywkę.

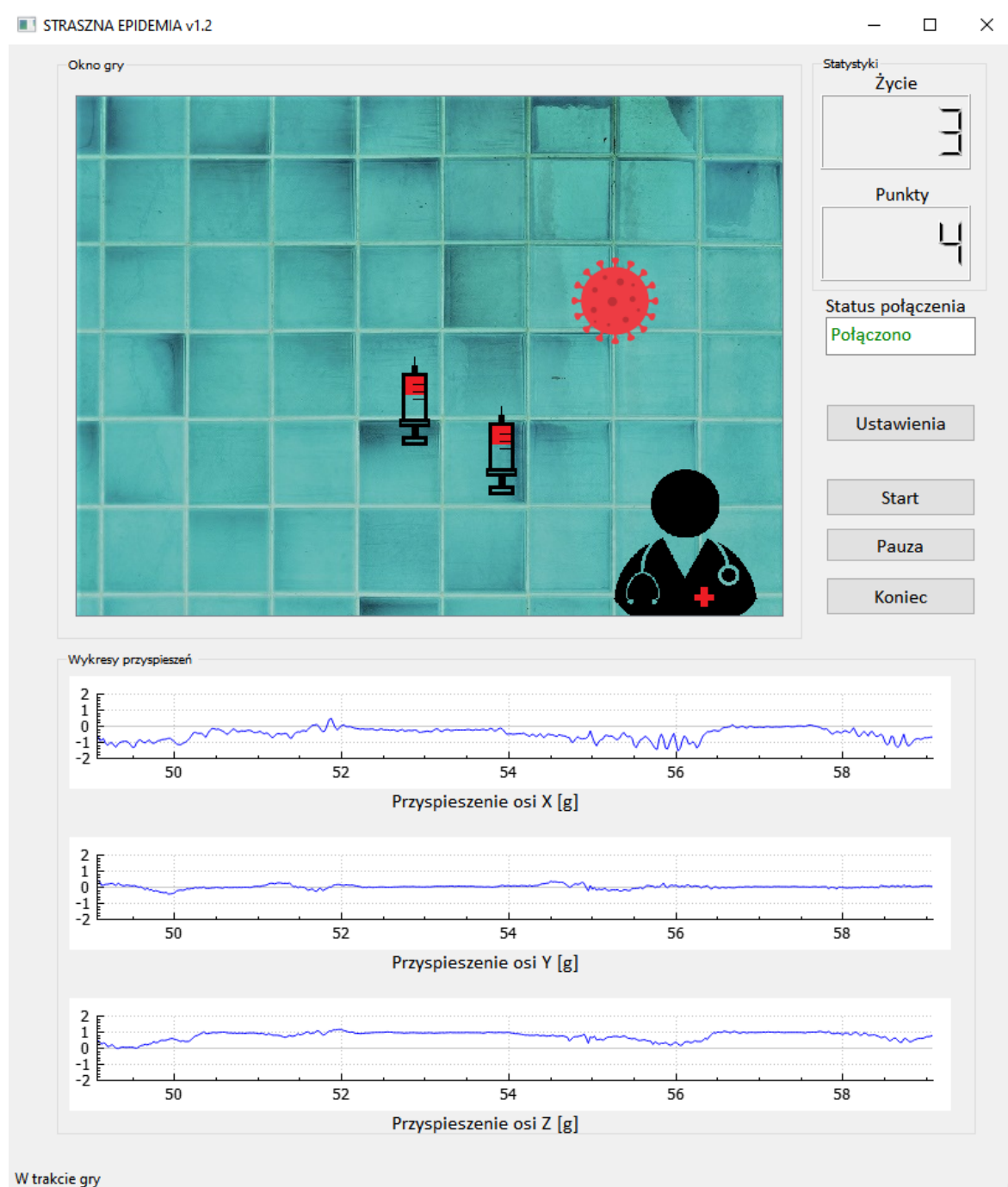
5.6.5 Zmiany w projekcie

Wprowadzono następujące modyfikacje względem poprzedniej wersji:

- zmieniono układ położenia wykresów przyspieszenia danych osi
- powiększono opisy przycisków i napisy znajdujące się w oknie celem poprawienia ich widoczności
- zmieniono mechanikę rozgrywki - dodano punkty życia
- dodano dodatkowy przycisk *Pauza* pozwalający na wstrzymanie i powrót do rozgrywki bez utraty punktów
- dodano dodatkowe okno wyświetlające się po zakończeniu rozgrywki
- dodano widżet wyświetlający aktualny stan połączenia *Połączono* lub *Rozłączono*
- dodano funkcjonalność wyświetlania na belce aktualnego statusu gry *W trakcie gry*, *Rozgrywka wstrzymana* oraz *Koniec gry*

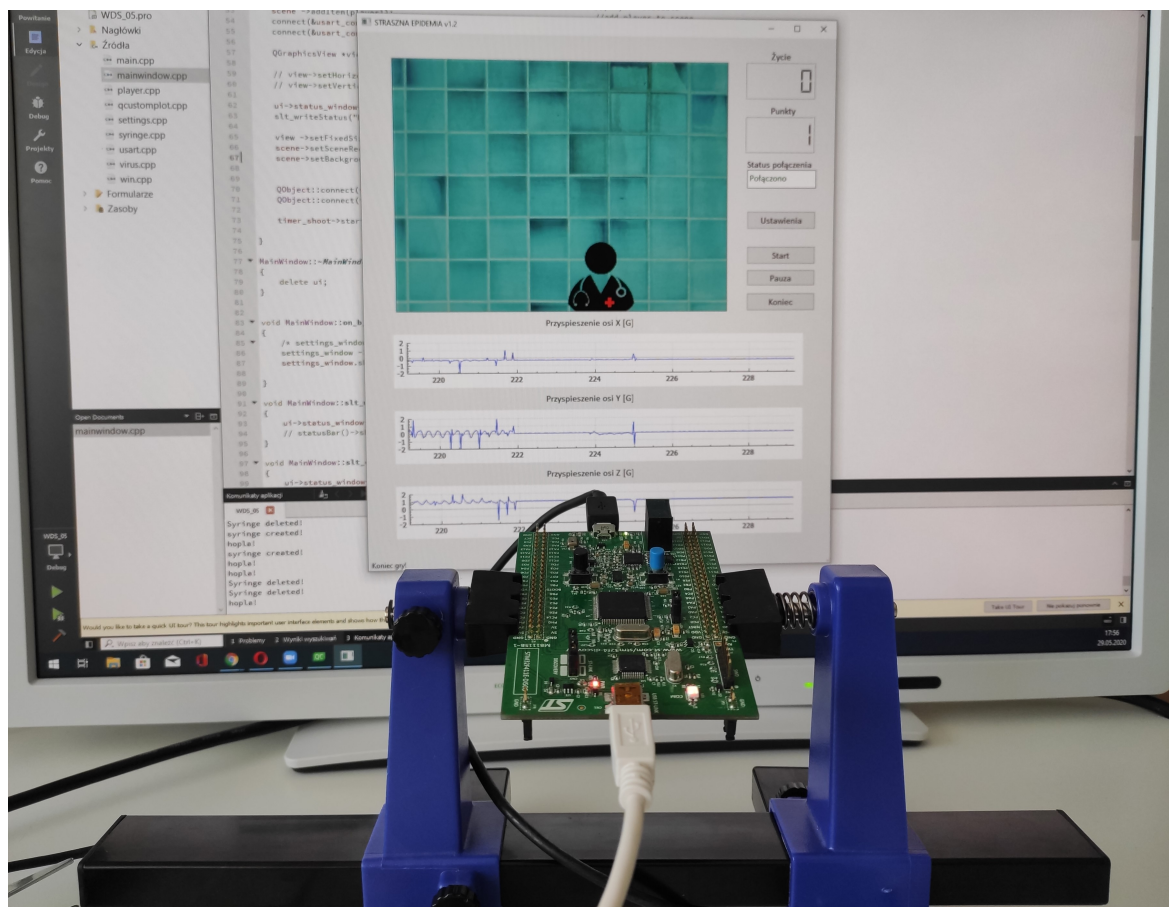
5.7 Podsumowanie

Projekt wykonano zgodnie z założeniami projektowymi. Wprowadzone zmiany są efektem pracy nad polepszeniem jakości rozgrywki, poprawienia widoczności i ułatwieniem dostępu do poszczególnych sekcji (patrz rys. 9).



Rysunek 9: Widok okna głównego ukończonej gry

Użytkownik ma możliwość korzystania uchwytu do kontrolera (patrz rys. 10) , który pozwala na dokładniejsze sterowanie postacią.



Rysunek 10: Wygląd kontrolera umieszczonego w uchwycie

6 Bibliografia

A. Kurczyk. *Mikrokontrolery STM32 dla początkujących*. Wydawnictwo BTC, Legionowo, 2019

QT for begginers Praca zbiorowa, stan na 22.03.2020

https://wiki.qt.io/Qt_for_Beginners

Kurs QT – Forbot Mateusz Patyk, stan na 22.03.2020

<https://forbot.pl/blog/kurs-qt-1-pierwsza-aplikacja-w-praktyce-id35549>

User manual Discovery kit with STM32F411VE MCU, stan na 22.03.2020

https://www.st.com/resource/en/user_manual/dm00148985.pdf

Kurs STM32 – Forbot Bartek Kurosz, stan na 20.04.2020

<https://forbot.pl/blog/kurs-stm32-f4-11-komunikacja-przez-usb-id13477>