

PROJEKT
WIZUALIZACJA DANYCH SENSORYCZNYCH

Płytką odbijającą piłkę

Marcel Konieczny, 252966

Prowadzący:
dr inż. Bogdan Kreczmer



Katedra Cybernetyki i Robotyki
Wydziału Elektroniki, Fotoniki i
Mikrosystemów
Politechniki Wrocławskiej

Spis treści

1	Charakterystyka tematu projektu	1
2	Podcele i etapy realizacji projektu	1
3	Specyfikacja finalnego produktu	1
4	Terminarz realizacji poszczególnych podcelów	2
5	Kamienie milowe projektu	3
6	Projekt graficznego interfejsu użytkownika	4
7	Projekt elektroniczny mikrokontrolera	7
8	Oprogramowanie mikrokontrolera	8
9	Aplikacja okienkowa	9
9.1	Menubar	9
9.2	StackWidget	10
10	Zastosowane algorytmy	11
11	Prezentacja działania aplikacji	12
12	Podsumowanie	12

1 Charakterystyka tematu projektu

Celem projektu jest wizualizacja płytki odbijającej piłkę, płytka będzie się poruszała w 4 kierunkach. Płytką będzie sterowana na podstawie odczytów z akcelerometru podłączonego do mikrokontrolera STM32. Aby poruszyć płytka w aplikacji w prawą lub lewą stronę będzie konieczne przechylenie akcelerometru w odpowiednią stronę. Poruszenie akcelerometrem do tyłu będzie powodowało, że podczas odbicia kula będzie zwolniona, poruszenie akcelerometrem do przodu spowoduje przyśpieszenie kuli. Płytką którą użytkownik będzie mógł sterować będzie znajdować się na spodzie interfejsu, na górze interfejsu będzie znajdować się płytka sterowana przez komputer (będzie sterowana na podstawie algorytmu).

2 Podcele i etapy realizacji projektu

Lista podcelów:

- Przegląd literatury i zasobów Internetu związanych z tematem projektu
- Projekt graficzny aplikacji
- Projekt układu elektronicznego (schemat ideowy)
- Utworzenie prototypu układu na płytce stykowej
- Oprogramowanie komunikacji komputer-STM32 za pomocą UART
- Utworzenie szaty graficznej aplikacji w QtCreatorze
- Poruszanie w aplikacji płytka na podstawie odczytów z akcelerometru,
- Zaimplementowanie odbijania piłki od płytki
- Zaimplementowanie poruszania drugiej płytki przez komputer
- Dopracowanie i testy zaimplementowanego algorytmu odbicia
- Oprogramowanie wizualizacji odczytów prędkości piłki na wykresie
- Testy gotowej aplikacji

3 Specyfikacja finalnego produktu

Efekt końcowy projektu będzie zawierał następujące funkcjonalności:

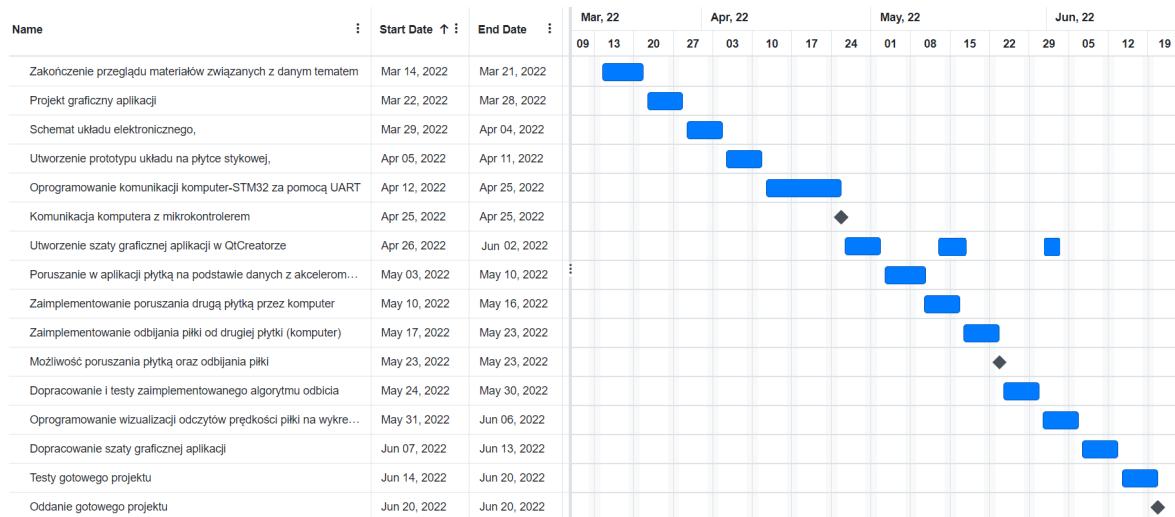
- Aplikacja przedstawiająca płytka oraz piłkę
- Wizualizacja poruszania płytki w aplikacji za pomocą akcelerometru
- Odbijanie piłki przez płytki
- Wizualizacja danych odczytanych z akcelerometru na wykresie
- Wizualizacja danych na temat prędkości piłki na wykresie

4 Terminarz realizacji poszczególnych podcelów

- 21 marca 2022 – zakończenie przeglądu materiałów związanych z danym tematem
- 28 marca 2022 – projekt graficzny aplikacji
 - Stworzenie projektu wyglądu aplikacji na kartce papieru,
 - Uwzględnienie umiejscowienia przycisków, wykresów, obszaru poruszania płytki i piłki,
- 4 kwietnia 2022 – schemat układu elektronicznego
- 11 kwietnia 2022 – utworzenie prototypu układu na płytce stykowej,
- 25 kwietnia 2022 – oprogramowanie komunikacji komputer-STM32 za pomocą UART
 - Konfiguracja poszczególnych peryferiów STM32 w celu komunikacji UART,
 - Wyświetlanie danych z akcelerometru na terminalu komputera,
- 2 maja 2022 – utworzenie szaty graficznej aplikacji w QtCreatorze
 - utworzenie oraz umiejscowienie widgetów na podstawie projektu graficznego,
 - wyświetlanie piłki oraz płytka,
- 9 maja 2022 – poruszanie w aplikacji płytką na podstawie odczytów z akcelerometru,
 - utworzenie funkcji za pomocą której będzie możliwe poruszanie płytka w odpowiednich kierunkach na podstawie odczytów z akcelerometru,
- 16 maja 2022 – zaimplementowanie odbijania piłki od płytki,
 - zaimplementowanie funkcji odpowiedzialnej za odbijanie piłki od płytka,
 - rozpoczęcie implementacji funkcji odpowiedzialnej za zwalnianie oraz przyspieszanie piłki,
- 23 maja 2022 – zaimplementowanie poruszania drugiej płytki przez komputer
 - zaimplementowanie funkcji odpowiedzialnej za poruszanie się drugiej płytki na podstawie ruchów piłki (komputer oblicza z wyprzedzeniem gdzie w przyszłości będzie znajdująć się piłka i w to miejsce porusza płytka znajdującą się na górze aplikacji),
- 30 maja 2022 – dopracowanie i testy zaimplementowanego algorytmu odbicia,
 - skończenie implementacji funkcji odpowiedzialnej za zwalnianie oraz przyspieszanie piłki,
 - przetestowanie różnych wariantów odbicia piłki od płytki, ściany boczne,

- 6 czerwca 2022 – oprogramowanie wizualizacji odczytów prędkości piłki na wykresie
 - Oprogramowanie widgetów odpowiedzialnych za tworzenie wykresów na podstawie prędkości piłki,
- 13 czerwca 2022 – dopracowanie szaty graficznej aplikacji,
- 20 czerwca 2022 – testy gotowego projektu.

Wykres Gantta wygląda następująco



Rysunek 1: Wykres Gantta

5 Kamienie milowe projektu

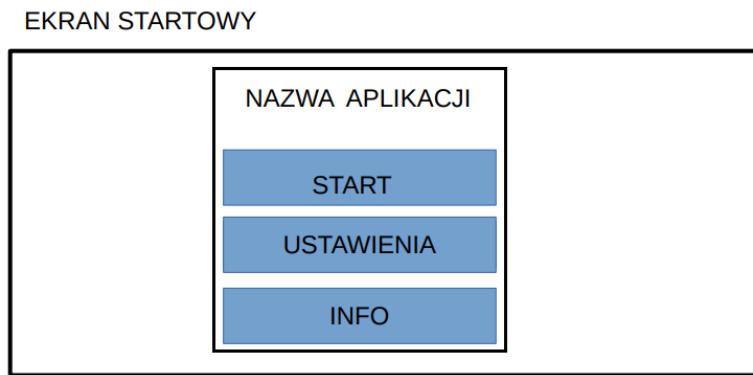
W trakcie tworzenia projektu wystąpią następujące kamienie milowe:

- 27 kwiecień – Komunikacja komputera z mikrokontrolerem
- 23 maj – Możliwość poruszania płytą oraz odbijania piłki
- 20 czerwca 2022 – Oddanie gotowego projektu

6 Projekt graficznego interfejsu użytkownika

Włączając aplikację pojawi się okno (rys. 2) zawierające 3 przyciski :

- przycisk **START** - po wciśnięciu tego przycisku na ekranie pojawi się okno gry,
- przycisk **USTAWIENIA** – po wciśnięciu tego przycisku na ekranie pojawi się okno ustawienia,
- przycisk **INFO** – po wciśnięciu tego przycisku na ekranie pojawią się informacje na temat autora aplikacji, daty wydania oraz wersji aplikacji.



Rysunek 2: Projekt ekranu startowego

Widget gry (rys. 3) będzie posiadać następujące funkcjonalności:

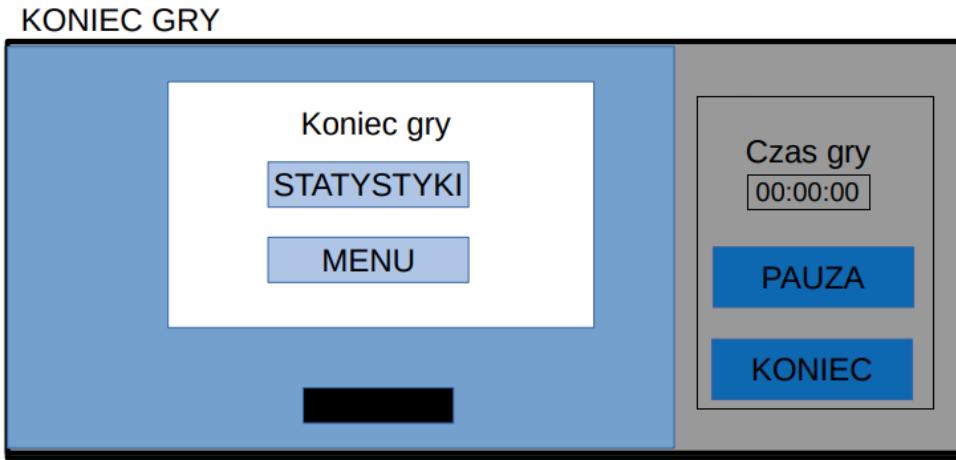
- powierzchnia gry – miejsce gdzie będą znajdować się 2 płytki oraz piłka,
- czas gry – będzie pokazywać jak długo trwa gra,
- przycisk PAUZA – po wciśnięciu gra się zatrzyma, po ponownym wciśnięciu gra będzie działać,
- przycisk KONIEC – po wciśnięciu użytkownik wróci do ekranu startowego.



Rysunek 3: Projekt widget'u gry

Podczas końca gry na ekranie pojawi się dialog (rys. 4) zawierający:

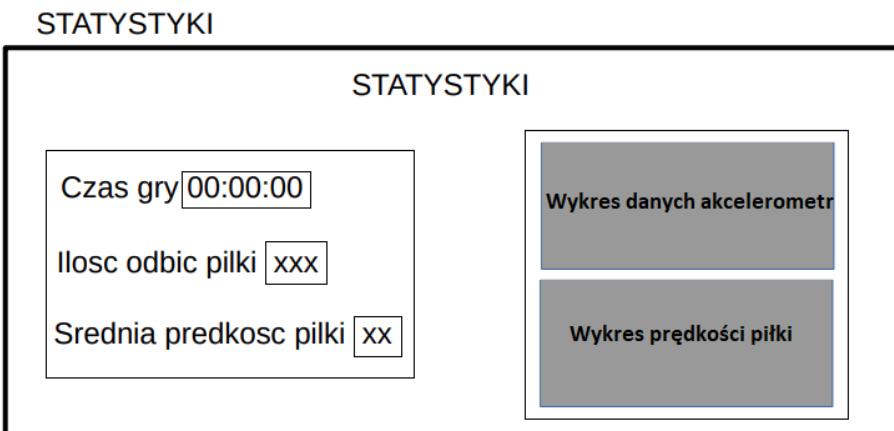
- przycisk STATYSTYKI – przycisk otwierający okno statystyk na temat gry,
- przycisk MENU – przycisk umożliwiający powrót do okna początkowego.



Rysunek 4: Projekt końca gry

Widget statystyk (rys. 5) będzie posiadał następujące funkcjonalności:

- informację na temat czasu gry,
- informację na temat ilości odbić piłki w czasie gry,
- informację na temat średniej prędkości piłki,
- wykres przedstawiający informację na temat danych z akcelerometru,
- wykres przedstawiający informację na temat prędkości piłki w czasie gry,
- przycisk MENU – przycisk umożliwiający powrót do okna początkowego.

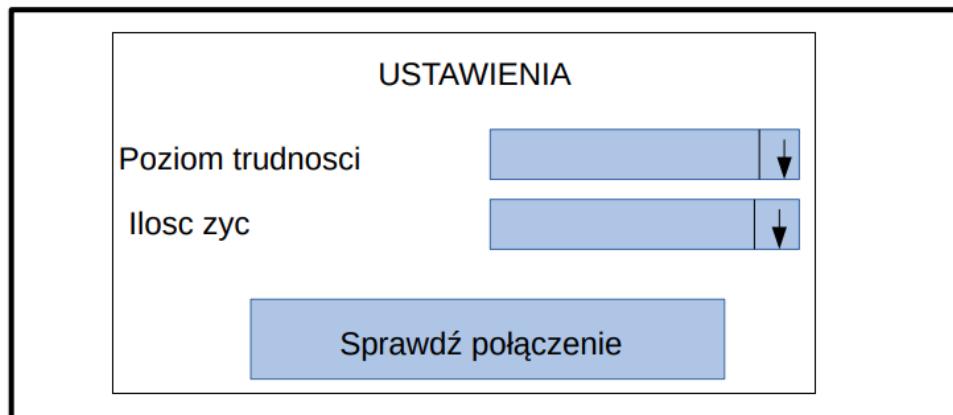


Rysunek 5: Projekt widget'u statystyk

Widget ustawień (rys. 6) będzie posiadać następujące funkcjonalności:

- Rozwijana zakładka **Poziom trudności** – będzie można wybrać poziom trudności gry komputera,
- Rozwijana zakładka **Ilość życia** – będzie można wybrać ilość życia w grze,
- przycisk **Sprawdź połaczenie** – po wciśnięciu komputer sprawdzi poprawność połączenia z mikrokontrolerem, jeśli połączenie będzie poprawne przycisk zmieni kolor na zielony,
- przycisk **MENU** – przycisk umożliwiający powrót do okna początkowego.

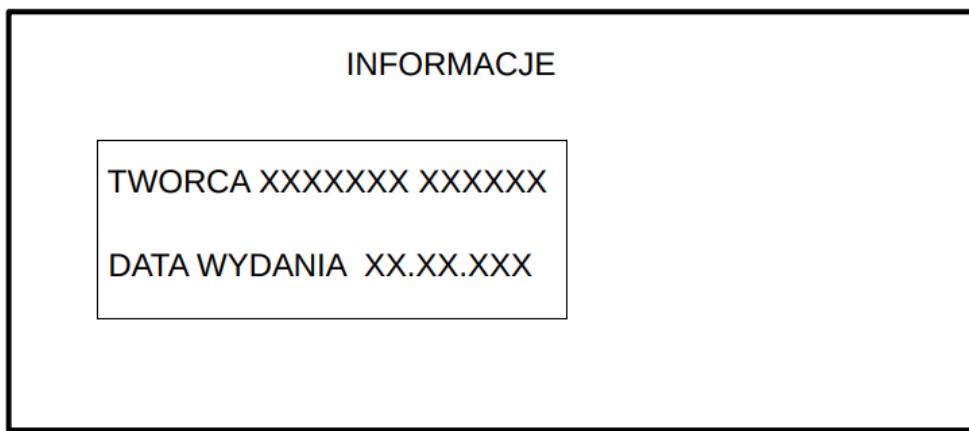
USTAWIENIA



Rysunek 6: Projekt widget'u ustawień

Widget info (rys. 7) będzie wyglądać następująco:

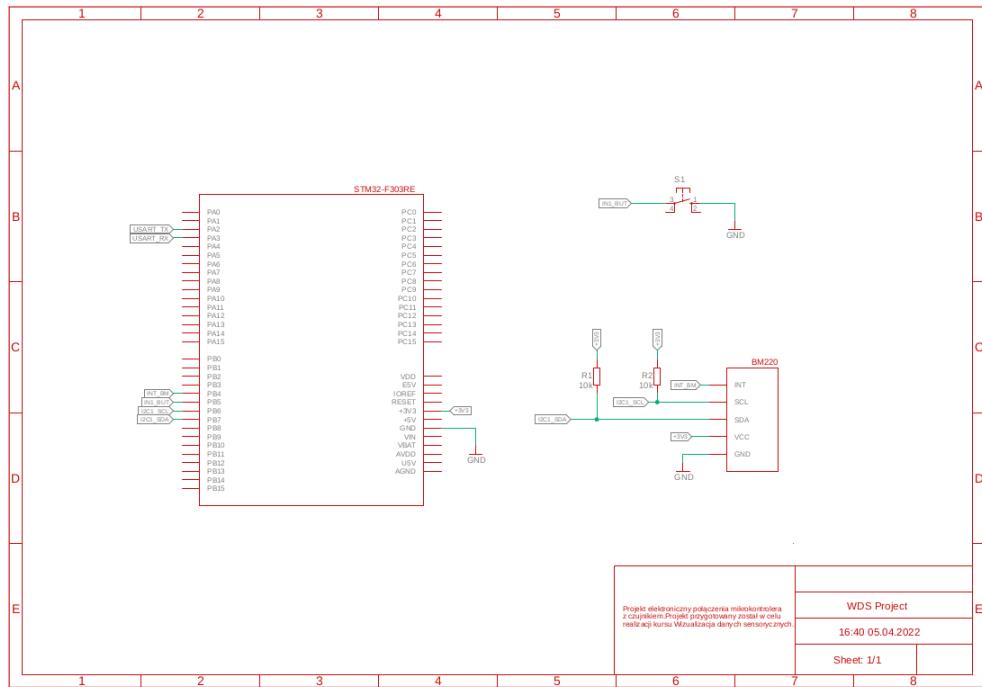
INFO



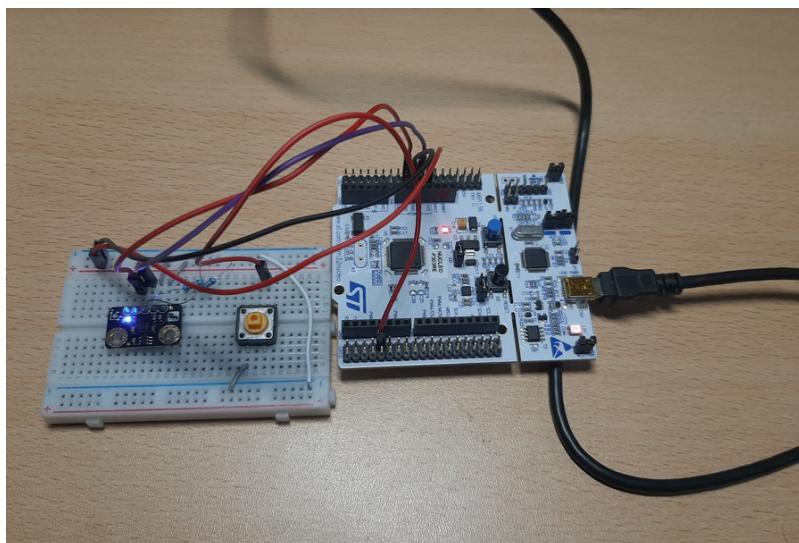
Rysunek 7: Projekt widget'u info

7 Projekt elektroniczny mikrokontrolera

W celu wykonania projektu wykonany został schemat elektroniczny układu (rys. 8). Na podstawie schematu elektronicznego wykonany został zaprojektowany układ (rys. 9).



Rysunek 8: Schemat elektroniczny



Rysunek 9: Układ elektroniczny

8 Oprogramowanie mikrokontrolera

W celu odczytu danych z akcelerometru została zaimplementowana biblioteka do obsługi akcelerometru za pomocą której mikrokontroler STM32 jest w stanie odczytywać dane na temat położenia czujnika.

Następnie zostało zaimplementowane wysyłanie danych za pomocą peryferium UART. Zdecydowałem że przesyłane dane będą w formacie:

”X (p.o. x) (p.o. y) (p.o. z) CRC8”

Znaczenie poszczególnych elementów przesyłanych danych:

- X – znak początkowy sygnalizujący że następuje przesył danych,
- (p.o. x) – dane odczytane z akcelerometru względem osi x,
- CRC8 – suma kontrolna zapisywana w systemie szesnastkowym.

Suma kontrolna CRC8 jest obliczana na podstawie przesyłanych danych.
Przysłane dane wyglądają następująco:

```
"X 4 2 14 E2"  
"X 3 1 17 26"  
"X 5 1 12 89"  
"X 6 1 16 C9"  
"X 5 3 14 D1"  
"X 4 2 14 E2"  
"X 3 2 15 14"  
"X 3 1 13 C3"  
"X 3 2 16 B7"  
"X 3 1 14 85"  
"X 4 0 13 DB"  
"X 5 0 17 82"  
"X 4 1 14 12"  
"X 5 0 17 82"  
"X 4 0 14 9D"  
"X 3 1 14 85"  
"X 3 0 15 6B"  
"X 3 1 14 85"  
"X 3 1 14 85"  
"X 4 0 14 9D"  
"X 3 1 15 E4"  
"X 4 0 15 FC"  
"X 4 1 15 73"  
"X 2 1 14 39"
```

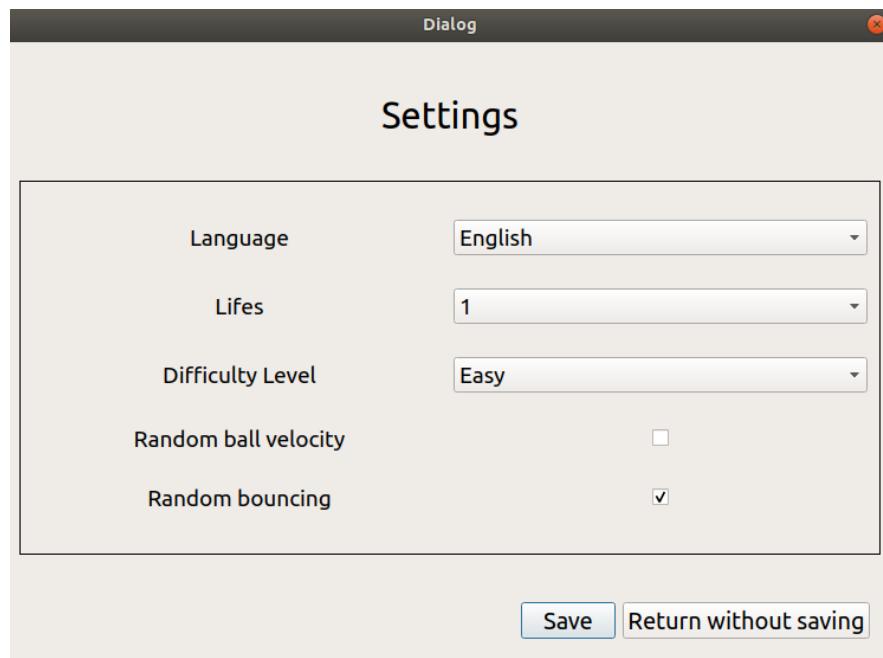
Rysunek 10: Otrzymywane dane

9 Aplikacja okienkowa

Okno główne aplikacji składa się z `stackWidget`'a oraz `menubar`'a.

9.1 Menubar

W menubar stworzona została zakładka **Settings** oraz **Connecitivity**. W zakładce **settings** wywoływany jest dialog ustawień za pomocą którego wybiera się parametry gry (rys. 11).

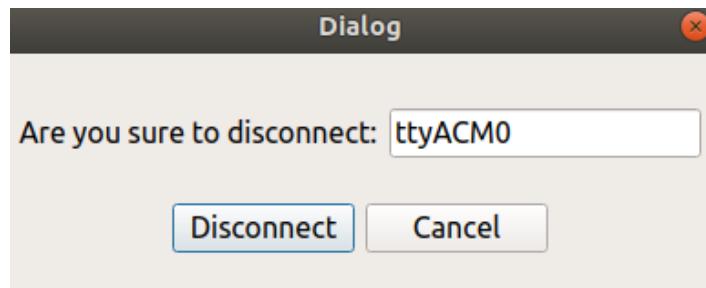


Rysunek 11: Dialog ustawień

Zakładka Connecitivity składa się z dialogu Connect (rys. 12) oraz Disconnect (rys. 13). Po otwarciu dialogu connect należy uruchomić wyszukiwanie podłączonych urządzeń, jeśli zostało wykryte urządzenie zdolne wysyłać dane z akcelerometru, można się z nim połączyć za pomocą przycisku connect lub zamknąć dialog. Za pomocą dialogu disconnect można rozłączyć się z aktualnie połączonym urządzeniem, jeśli aplikacja nie jest połączona z urządzeniem, rozłączenie nie jest możliwe.



Rysunek 12: Dialog connect



Rysunek 13: Dialog disconnect

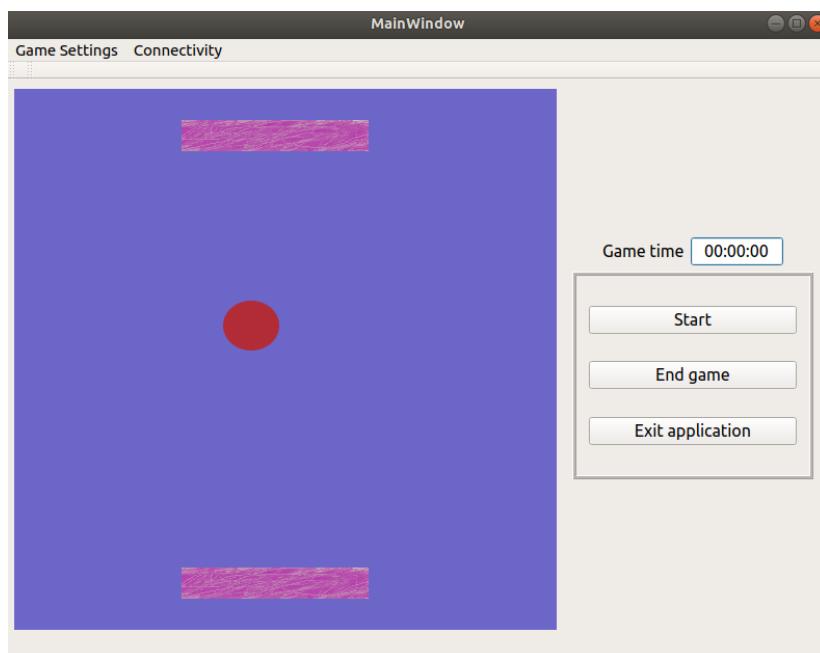
9.2 StackWidget

StackWidget składa się z widget'a gry (rys. 14) oraz widgeta statystyk końcowych (rys. 15).

Widget gry jest reprezentowany przez klasę `mainWidget`. Dodatkowo klasa ta składa się z widget'u zawierającego panel boczny oraz widget'u pola gry. Pole gry symuluje przebieg gry poprzez rysowanie płytka oraz piłki. Jedna z płytka rysowana jest na podstawie odczytów z akcelerometru, druga sterowana jest na podstawie pozycji piłki. Panel boczny składa się z miejsca gdzie na bieżąco przedstawiany jest aktualny czas gry oraz z trzech przycisków:

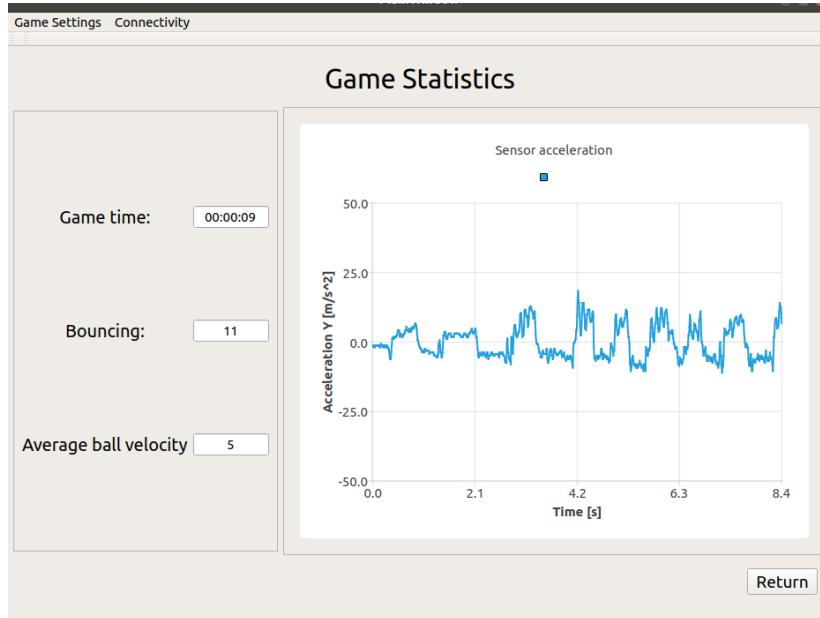
- **Pauze** – Przycisk odpowiedzialny za zatrzymywanie oraz wznawianie rozgrywki,
- **End Game** – przycisk odpowiedzialny za kończenie rozgrywki,
- **Exit application** – przycisk odpowiedzialny za zamknięcie aplikacji.

Dodatkowo w czasie końca gry nie jest możliwe wcisnięcie przycisków **Pauze** oraz **End Game**.



Rysunek 14: Widget gry

Widget statystyk końcowych (rys. 15), który jest reprezentowany przez klasę `gameStatisticsWidget`. Klasa ta modeluje statystyki końcowe takie jak: czas trwania gry, ilość odbić piłki od płytaków oraz średnia prędkość piłki. Dodatkowo po lewej stronie wyświetlanym jest wykres przedstawiający wartości przyśpieszenia względem osi y w trakcie poszczególnych sekund gry.



Rysunek 15: Widget statystyk końcowych

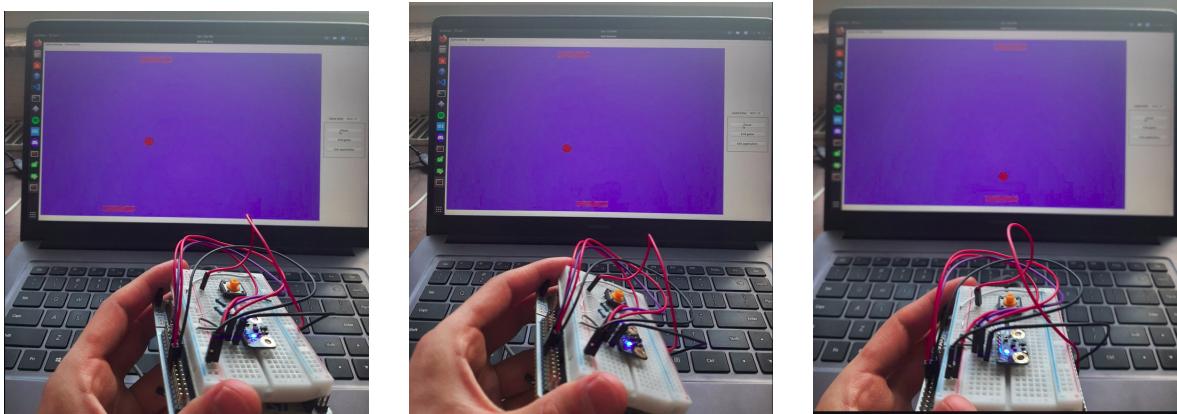
10 Zastosowane algorytmy

W procesie wykonania gry zostały zaimplementowane następujące algorytmy:

- **Algorytm poruszania się piłki** – W celu poruszania się piłki utworzono pole `currAng` w klasie `imageParameters` zawierające informację na temat aktualnego kąta reprezentujący miejsce w którym piłka się porusza oraz pole `ballVel` informujące z jaką prędkością piłka się porusza. Na podstawie tych informacji podczas wystąpienia `timeout'a` (okres 10 mil. sec.) mnożone są odpowiednie wektory i wyznaczana jest nowa pozycja piłki.
- **Algorytm odbijania się piłki** – Podczas każdego przesunięcia piłki sprawdzane jest czy nastąpi dotknięcie piłki z ścianą lub płytaką. Jeśli następuje dotknięcie, przeliczane jest `currAng` odpowiedzialne za kierunek w którym piłka się aktualnie porusza.
- **Algorytm poruszania się płytaków komputera** – Płytki sterowane przez komputer poruszają się w kierunku miejsca w którym znajduje się aktualnie piłka, w czasie każdego `timeout'a` pobierana jest aktualna pozycja piłki, pozycja ta jest miejscem, gdzie płytka ma się przemieścić. W zależności od poziomu trudności, płytka porusza się z różnymi prędkościami (wraz z wzrostem trudności szybciej się porusza). Dodatkowo wraz z zmianą szerokości okna aplikacji płytka zmienia swoją prędkość, ponieważ wraz z wzrostem szerokości płytka może się poruszać po większej ilości pikseli.

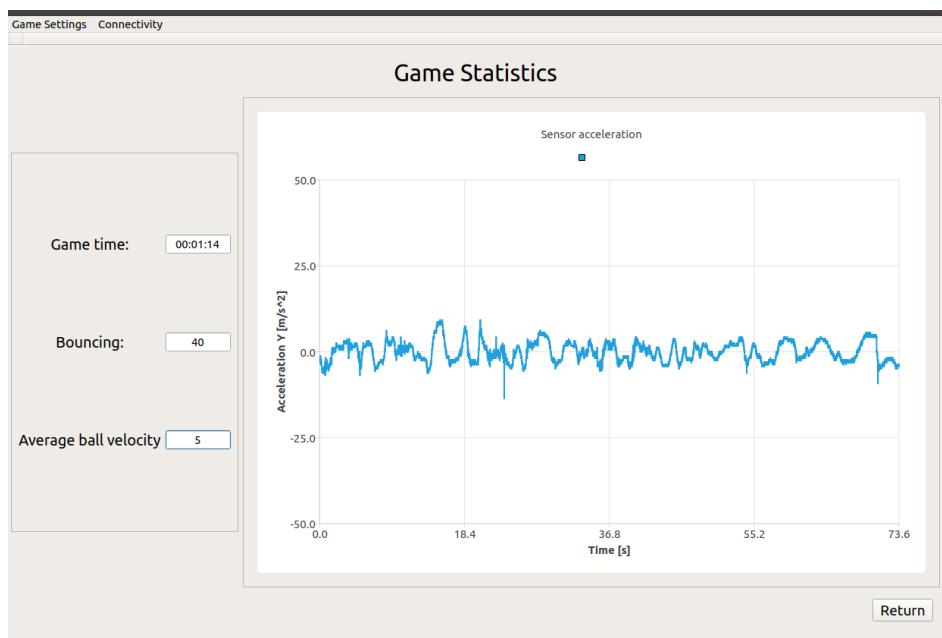
11 Prezentacja działania aplikacji

Działanie aplikacji wygląda następująco (rys. 16).



Rysunek 16: Prezentacja gry

Statystyki wybranej gry (rys. 17).



Rysunek 17: Statystyki gry

12 Podsumowanie

Projekt został wykonany w wyznaczonym terminie, w czasie realizacji nie wystąpiły znaczące opóźnienia. Względem założeń początkowych zmieniono sposób sterowania z przesuwania układu elektronicznego do przechylania układu względem odpowiedniej osi. Taka zmiana była spowodowana problemami z sterowaniem płytka podczas przesuwania układu. Został także zmieniony projekt interfejsu: panel ustawień wyświetla się jako dialog (nie jako widget) oraz zrezygnowano z widget'u INFO.