

# AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU

Wydział Nauk Inżynieryjnych  
Katedra Informatyki

## DOKUMENTACJA PROJEKTOWA PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

### **Gra logiczna**

Autor:  
Maciej Śmierciak  
Michał Jonak  
Konrad Szczurek

Prowadzący:  
mgr inż. Dawid Kotlarski

Nowy Sącz 2022

# Spis treści

<b>1. Ogólne określenie wymagań</b>	<b>3</b>
1.1. Gra logiczna . . . . .	3
1.1.1. Tryb Graficzny . . . . .	5
1.1.2. Tryb Tekstowy . . . . .	7
<b>2. Określenie wymagań szczegółowych</b>	<b>9</b>
2.1. Założenia główne . . . . .	9
2.1.1. Utrzymanie modułowości projektu . . . . .	9
2.1.2. Łatwość implementacji . . . . .	9
2.1.3. Prostota w testowaniu i ewentualnym debuggingu . . . . .	9
2.1.4. Użycie technologii Bluetooth . . . . .	9
2.1.5. Użycie żyroskopu i czujnika oświetlenia . . . . .	10
2.1.6. Użycie latarki . . . . .	10
<b>3. Projektowanie</b>	<b>11</b>
3.1. Środowisko - Xamarin . . . . .	11
<b>4. Implementacja</b>	<b>12</b>
<b>5. Testowanie</b>	<b>13</b>
<b>6. Podręcznik użytkownika</b>	<b>14</b>
<b>Literatura</b>	<b>15</b>
<b>Spis rysunków</b>	<b>15</b>
<b>Spis tabel</b>	<b>16</b>
<b>Spis listingów</b>	<b>17</b>

# 1. Ogólne określenie wymagań

## 1.1. Gra logiczna

Projektem jest gra logiczna możliwa do zagrania tylko w trybie kooperacji. Gra będzie opierać się na stosunkowo łatwych zagadkach, które będzie można rozwiązać tylko współpracując.

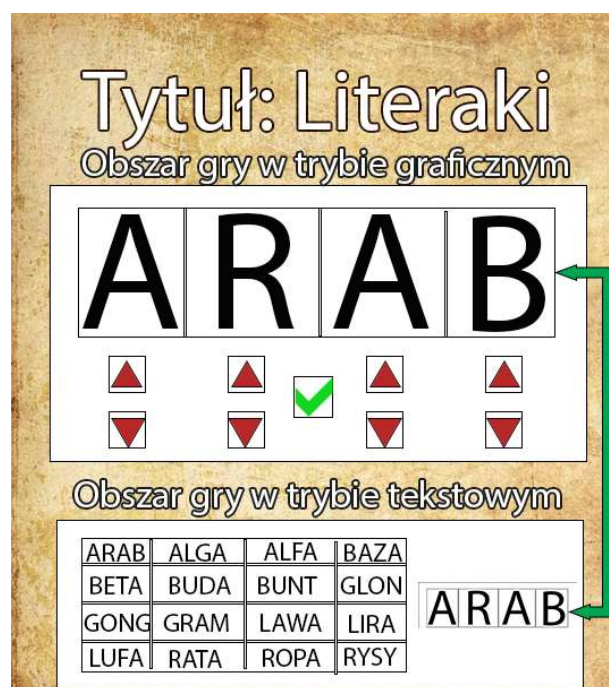
Ogólnym konceptem jest podzielenie gry na 2 główne części:

- Tryb graficzny
- Tryb tekstowy



Rys. 1.1. Labirynt

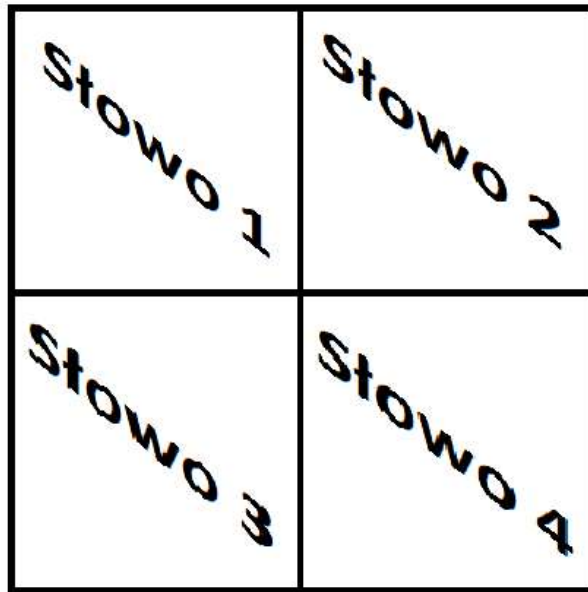
Ja możemy zauważyć na rysunku będziemy mieli określoną liczbę żyć na rozwiązanie określonej liczby zagadek w określonym czasie. W tym trybie zagadki będą polegały na wyjściu z labiryntu.



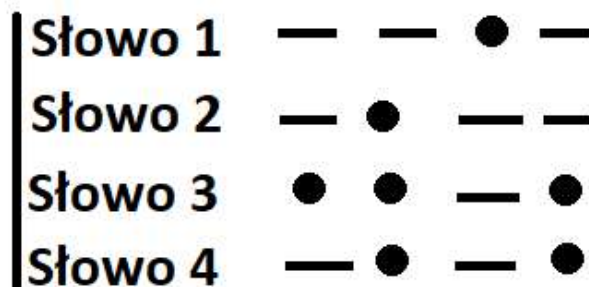
Rys. 1.2. Literaki

W tym trybie gracze mają za zadanie ułożyć czteroliterowy wyraz, który nakłada się z wyrazem w bazie. Dostęp do bazy wyrazów ma gracz w trybie tekstowym. Gracz obsługujący tryb graficzny za pomocą strzałek zmienia litery na danej pozycji. Ze względu na to, że nie wszystkie litery są na wszystkich polach możliwość będzie tylko jedna. Błędna kombinacja oznacza utratę jednego z żyć. Podobnie jak w trybie labiryntu po utracie 3 żyć gracze przegrywają.

Trzeci tryb gry, który zostanie zaimplementowany do gry będzie oparty na latarce zawartej w telefonie. Telefon osoby obsługującej tryb graficzny włączy i wyłączy latarkę określoną liczbę razy tworząc przy tym "kod morse" opisany w trybie tekstowym. Gracz obsługujący tryb graficzny będzie miał za zadanie zapamiętać stosunkowo krótki kod i podać go osobie będącej w trybie graficznym. Następnie osoba zarządzająca trybem graficznym musi dopasować go do jednego ze słów po czym podaje słowo kluczowe współnikowi. Wybór złego słowa pozbawia nas jednego życia i losuje nowy sygnał.



Rys. 1.3. Kod - Tryb graficzny

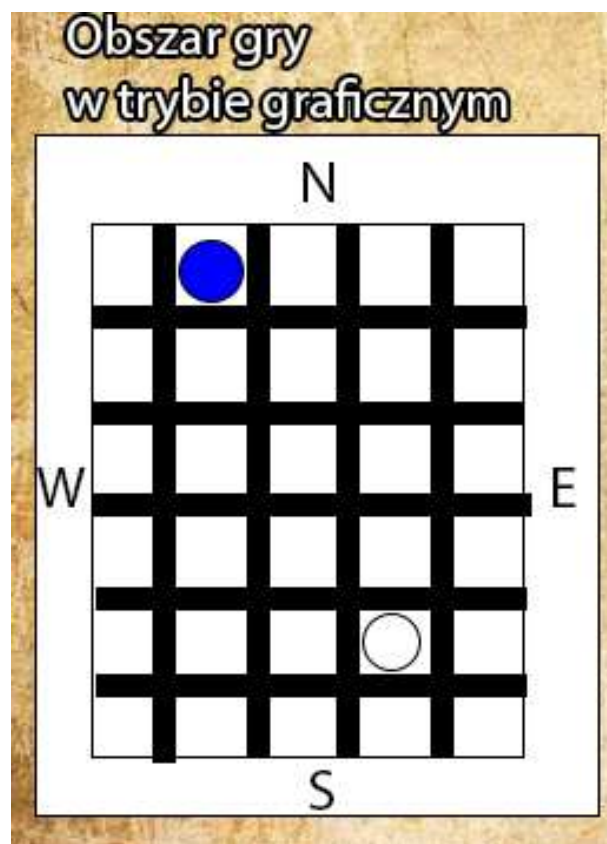


Rys. 1.4. Kod - Tryb tekstowy

### 1.1.1. Tryb Graficzny

Będzie opierał się na rozwiązywaniu zagadek. Gracz sam nie będzie w stanie rozwiązać zagadki, ponieważ podpowiedzi czy też cała solucja danej zagadki będą zawarte w trybie tekstowym.

W tym trybie będziemy widzieć plansze rozgrywki i będziemy mogli sterować naszą postacią.



Rys. 1.5. Labirynt

W trybie graficznym, jak widać na powyższym rysunku, widzimy naszą postać, niebieską kulę, i nasz cel, białą kulę. Natomiast nie widzimy drogi do mety i w tym celu musimy komunikować się z partnerem.

Za każdym razem jak wykonamy zły ruch czyli wejdziemy w ścianę nasza kulka będzie wracać na początek trasy a my tracimy jedno z naszych żyć. Po utracie wszystkich żyć kończymy rozgrywkę.

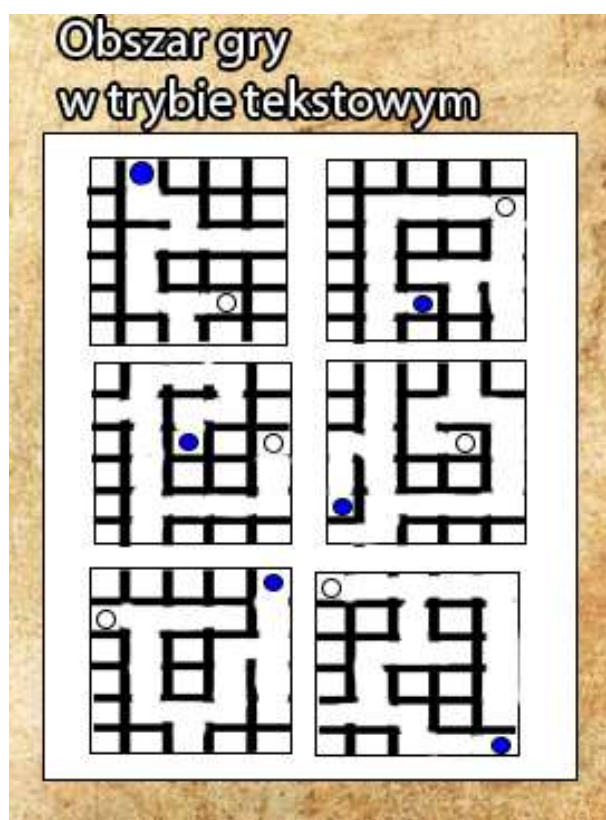
Innym trybem gry będą literaki polegające na układaniu słów. Gracz w tym trybie będzie za pomocą strzałek zmieniał litery na określonych pozycjach. Błędna kombinacja prowadzi do utraty życia i jest równoznaczna z wejściem w ścianę w trybie labiryntu.

Tryb graficzny trzeciej gry będzie oparty na prostej tabeli z paroma różnymi opcjami do wyboru. W zależności od otrzymanych instrukcji od naszego współnika będziemy musieli wybrać jedną z nich. Telefon gracza obsługującego ten tryb będzie za pomocą latarki wyświetlał jeden z dwóch rodzajów sygnału. Będzie to kod oparty o kod morsa ale ze zmienionymi słowami. Dla przykładu jeżeli latarka włączy się raz na długo a potem 3 razy mrugnie będzie to oznaczało jeden sygnał długi (oznaczenie w trybie tekstowym - kreska) i 3 krótkie (w trybie tekstowym - kropka).

### 1.1.2. Tryb Tekstowy

Będzie opierał się na znajdowaniu odpowiedzi czy też solucji do aktualnie wykonywanej zagadki przez osobę obsługującą tryb graficzny.

Naszym zadaniem będzie współpraca z osobą, która steruje postacią w trybie graficznym w celu jak najefektywniejszego ukończenia zagadek przed końcem ustalonego czasu.



Rys. 1.6. Labirynt

Jak można zobaczyć na powyższym rysunku w trybie tekstowym będziemy widzieć dostępne mapy rozgrywki. Zadaniem gracza w trybie tekstowym będzie ta-

kie poprowadzenie partnera w trybie graficznym, żeby niebieska kula dotarła do mety(białej kuli) unikając wchodzenia w ściany.

W trybie gry "Literaki" gracz w tym trybie będzie miał dostępną bazę ze słowami i będzie musiał dzięki komunikacji z graczem operującym interfejsem graficznym pomóc mu ułożyć pasujące słowo. Gracz w trybie graficznym będzie miał opcję ułożenia tylko jednego słowa z bazy.

W trzecim trybie gry będziemy musieli za pomocą komunikacji z naszym partnerem rozszyfrować o jaki kod chodzi w danym momencie. W trybie tekstowym będziemy otrzymywać informacje dotyczące "wyglądu kodu". W rzeczywistości otrzymamy informację ile było długich sygnałów a ile krótkich i w jakiej kolejności. Po otrzymaniu takowych informacji gracz będzie stawał przed wyborem jednego ze słów, które w tym przypadku będzie oznaczone przed chwilą otrzymanym kodem. Jeżeli żadne słowo się nie będzie zgadzało to najprawdopodobniej otrzymaliśmy złe podpowiedzi od naszego partnera. Jeżeli jednak znajdujemy pasujący kod to podajemy słowo, które ten kod opisuje do operatora trybu graficznego.



## 2. Określenie wymagań szczegółowych

### 2.1. Założenia główne

- Utrzymanie modułowości projektu:
- Łatwość implementacji:
- Prostota w testowaniu i ewentualnym debuggingu
- Użycie technologii Bluetooth
- Użycie żyroskopu i czujnika oświetlenia
- Użycie latarki

#### 2.1.1. Utrzymanie modułowości projektu

Pozwoli to na pracę nad wieloma "poziomymi" jednocześnie co przełoży się na lepsze rozłożenie pracy pomiędzy członków grupy.

#### 2.1.2. Łatwość implementacji

Pozwoli to na testowanie każdego modułu osobno. Dzięki temu rozwiązaniu będziemy mogli lepiej wyeliminować błędy. A co za tym idzie lepiej dopracować nasz projekt.

#### 2.1.3. Prostota w testowaniu i ewentualnym debuggingu

Chcemy dążyć do jak najłatwiejszego i jednocześnie najbardziej efektywnego sposobu testowania aplikacji. Pozwoli nam to zaoszczędzić cenny czas, który będziemy mogli poświęcić na lepsze dopracowanie szczegółów.

#### 2.1.4. Użycie technologii Bluetooth

Jedna z zagadek w naszej grze będzie oparta o technologię bluetooth. Zadaniem gracza będzie wyłączenie i włączenie bluetooth odpowiednią ilość razy.

Bluetooth to standard bezprzewodowej komunikacji krótkiego zasięgu pomiędzy różnymi urządzeniami elektronicznymi. Jest opisany w specyfikacji IEEE 802.15.1. Jego specyfikacja techniczna obejmuje trzy klasy mocy nadawczej ERP 1-3 o zasięgu 100,

10 oraz 1 metra w otwartej przestrzeni. Najczęściej spotykaną klasą jest klasa druga. Standard korzysta z fal radiowych w paśmie częstotliwości ISM 2,4 GHz.

Urządzenie umożliwiające wykorzystanie tego standardu to adapter Bluetooth.

#### **2.1.5. Użycie żyroskopu i czujnika oświetlenia**

Żyroskop jak i czujnik oświetlenia docelowo mają posłużyć do rozwiązywania zagadek. Dla przykładu niektóre zagadki mogą być wykonane tylko w nocy czy też po prostu przy słabym oświetleniu.

Żyroskop to urządzenie służące do pomiaru lub utrzymywania położenia kąowego. Żyroskop w smartfonie to bowiem mikroskopijne płytki, które w odpowiednim momencie wibrują przekazując informacje do czujników. Dzięki temu możliwe jest ustalenie położenia smartfona oraz to czy i wokół której osi został obrócony.

Czujnik oświetlenia mierzy intensywność barwy białej, czerwonej, zielonej i niebieskiej w otoczeniu. Używany jest do zarządzania kolorystyką ekranu oraz zmiany intensywności jego podświetlenia. Dzięki temu wyświetlany obraz jest lepiej dostosowany do warunków w jakich przebywamy. Pozwala też zaoszczędzić nieco energii i dzięki temu wydłużyć czas pracy na baterii.

#### **2.1.6. Użycie latarki**

Jedna z zagadek będzie oparta na wysyłaniu sygnałów w formie kodu morza za pomocą latarki. Latarka będzie uruchamiana na określony odstęp czasu po czym zostanie wyłączona i włączona ponownie jeżeli ostatni sygnał nie został pokazany. Kod będzie oparty na 2 sygnałach: krótkim i długim.

## 3. Projektowanie

### 3.1. Środowisko - Xamarin

Xamarin to platforma typu open source do tworzenia nowoczesnych i wydajnych aplikacji dla systemów iOS, Android i Windows za pomocą platformy .NET. Xamarin to warstwa abstrakcji, która zarządza komunikacją udostępnionego kodu z bazowym kodem platformy. Środowisko Xamarin działa w środowisku zarządzanym, które zapewnia wygody, takie jak alokacja pamięci i odzyskiwanie pamięci.

Xamarin umożliwia pisanie w C w Visual Studio co jest poniekąd wygodniejsze od innych rozwiązań. Ponadto połączyliśmy to również ze środowiskiem git co umożliwia nam dzielenie się kodem, zapisywanie go w "chmurze", czy też nawet wracanie do poprzednich wersji kodu.

Microsoft Visual Studio to zintegrowane środowisko programistyczne, służące do tworzenia, edytowania i debugowania kodu. IDE (z ang. Integrated Development Environment), czyli zintegrowane środowisko programistyczne to software oferujący szereg funkcji przydatnych podczas tworzenia oprogramowania dla systemów Windows, Android, iOS, rozwiązań internetowych oraz opartych o chmurę. Poza standardowym edytorem oraz debugerem, które zapewnia większość aplikacji IDE, program Microsoft Visual Studio zawiera jeszcze kompilatory, narzędzia do uzupełniania kodu i wiele innych funkcji usprawniających pracę.

Git to najczęściej używany nowoczesny system kontroli wersji. Za pomocą usługi Git możesz śledzić zmiany kodu wprowadzane w czasie i przywrócić określone wersje. Ponadto pozwala kontrolować dostęp do danych, wspiera zarządzanie wieloma repozytoriami. Jest to rozwiązanie, które pozwala zaoszczędzić sporo czasu i nerwów. Kolejnym plusem jest to, że jest on w miarę łatwy w użyciu i jest przejrzysty.

## 4. Implementacja

## 5. Testowanie

## 6. Podręcznik użytkownika

## Spis rysunków

1.1. Labirynt . . . . .	3
1.2. Literaki . . . . .	4
1.3. Kod - Tryb graficzny . . . . .	5
1.4. Kod - Tryb tekstowy . . . . .	5
1.5. Labirynt . . . . .	6
1.6. Labirynt . . . . .	7

## **Spis tabel**



## **Spis listingów**