

Gra Mission Impossible

Raport końcowy

Joanna Musialik
Patryk Nikonowicz

marzec 2024

Spis treści

1	Temat projektu	2
1.1	Podstawowe funkcjonalności	2
1.2	Kluczowe elementy implementacji	2
2	Zrzuty ekranu rozgrywki	3
3	Wykorzystywane środowisko i technologie	7
3.1	Podział na klasy	8
3.1.1	Diagram klas	8
3.1.2	Opis klas	9
4	Zespół	10
4.1	Członkowie	10
4.2	Ogólny podział ról	11
5	Kluczowe linki	11

1 Temat projektu

Aplikacja jest realizacją jednego z proponowanych tematów o poziomie trudności Trudny. Pomysł na rozgrywkę jest następujący: gracz jest potworem znajdującym się w lesie podczas pościgu zorganizowanego przez ludzi. Z powodu swoich nadnaturalnych zdolności potwór ma większą szybkość niż ludzie. Ludzie w lesie są jednak zdeterminowani, by złapać i zabić potwora - dlatego jak tylko go zauważą, zaczynają go gonić. Dodatkowo mają oni ze sobą krótkofalówki, za pomocą których komunikują się z centrum zarządzania. Centrum zarządzania informuje ludzi o przeszkodach znajdujących się w pobliżu oraz o położeniu potwora, a także może im dawać indywidualne polecenia. Gracz jest natomiast samotny - tylko on decyduje o tym, jak omijać przeszkody i skutecznie uciekać przed ludźmi. Celem rozgrywki jest przeżycie jak najdłużej. Rozgrzywka kończy się w momencie, gdy jeden z ludzi złapie potwora.

1.1 Podstawowe funkcjonalności

W ramach projektu zostaną zaimplementowane następujące funkcjonalności:

- gracz może rozpocząć rozgrywkę oraz po jej rozpoczęciu w dowolnym momencie ją przerwać bądź wstrzymać,
- gracz wybiera poziom trudności rozgrywki,
- gracz ma możliwość sprawdzenia wyniku swojej rozgrywki po jej odbyciu,
- gracz może zapoznać się z poprzednimi wynikami swoich rozgrywek z poziomu menu,
- gracz odbywa rozgrywkę poprzez poruszanie się po mapie w cztery strony (prawo, lewo, przód, tył) i unikanie przeszkód oraz ludzi.

1.2 Kluczowe elementy implementacji

Poniżej zawarto najważniejsze założenia podczas implementacji projektu:

- ludzie zaimplementowani jako agenci:
 - przesyłają informacje o położeniu przeszkód i potwora do centrum dowodzenia,
 - informacja o położeniu przeszkody bądź potwora zdobywana jest wtedy, gdy dany obiekt znajdzie się w zasięgu wzroku agenta (okrąg wokół postaci o określonym promieniu),
 - jeżeli potwór znajdzie się w zasięgu wzroku agenta, agent zaczyna za nim podążać do momentu, aż potwór opuści jego zasięg wzroku,
- centrum zarządzania agentami:
 - informuje o położeniu potwora i o przeszkodach w pobliżu,

- może wydawać pojedyncze polecenia np. poruszanie się w określonym kierunku do momentu zauważenia potwora,
- informacje o położeniu potwora i przeszkód zdobywa od agentów,
- potwór oraz ludzie mogą poruszać się w lewo, prawo, dół i górę,
- różnica pomiędzy szybkością potwora a ludzi, ilość przeszkód, zasięg wzroku ludzi, zasięg, w którym potwór jest złapany, przelicznik punktów oraz wielkość mapy zmieniają się w zależności od poziomu trudności rozgrywki,
- przeszkody są statyczne w ramach danej rozgrywki - czyli nie zmieniają swojego położenia w trakcie rozgrywki,
- zarówno ludzie jak i potwór muszą omijać przeszkody,
- rozgrywka kończy się w momencie, gdy potwór znajdzie się w zasięgu złapania człowieka,
- wynik danej rozgrywki jest zapisywany lokalnie na komputerze gracza - te wyniki można później obejrzeć w tabeli wyników,
- punkty za rozgrywkę są liczone na podstawie czasu trwania rozgrywki oraz poziomu trudności rozgrywki.

2 Zrzuty ekranu rozgrywki

Na rysunkach 1, 2 i 3 zamieszczono zrzuty ekranu z aplikacji.



(a) Główne menu gry



(b) Ekran wyboru trudności rozgrywki

Rysunek 1: Zrzuty ekranu



(a) Ranking rozgrywek

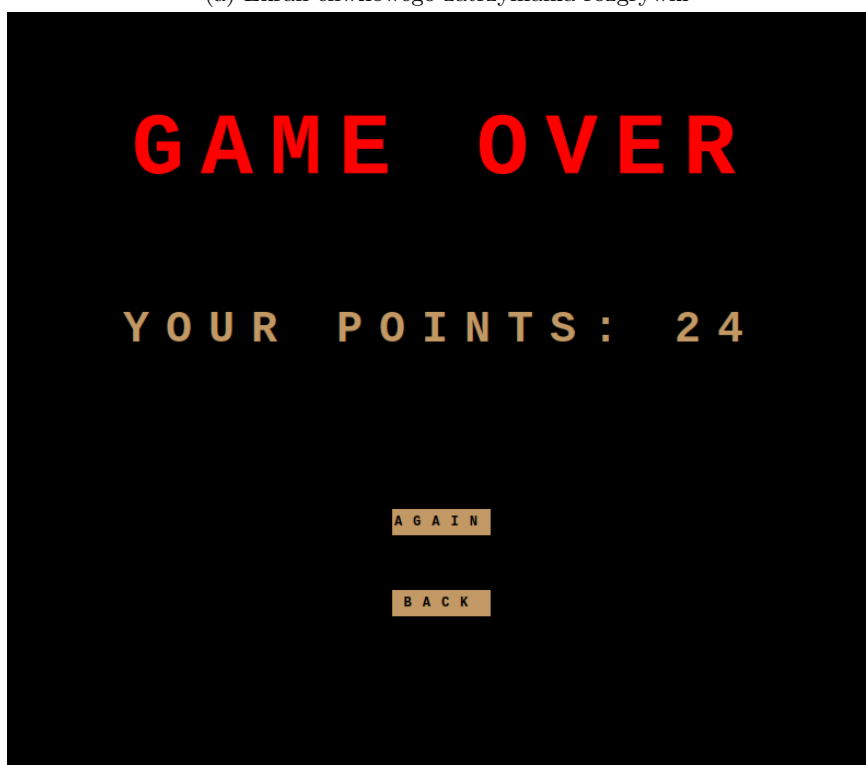


(b) Przykładowy ekran rozgrywki

Rysunek 2: Zrzuty ekranu (kontynuacja)



(a) Ekran chwilowego zatrzymania rozgrywki



(b) Ekran końca gry

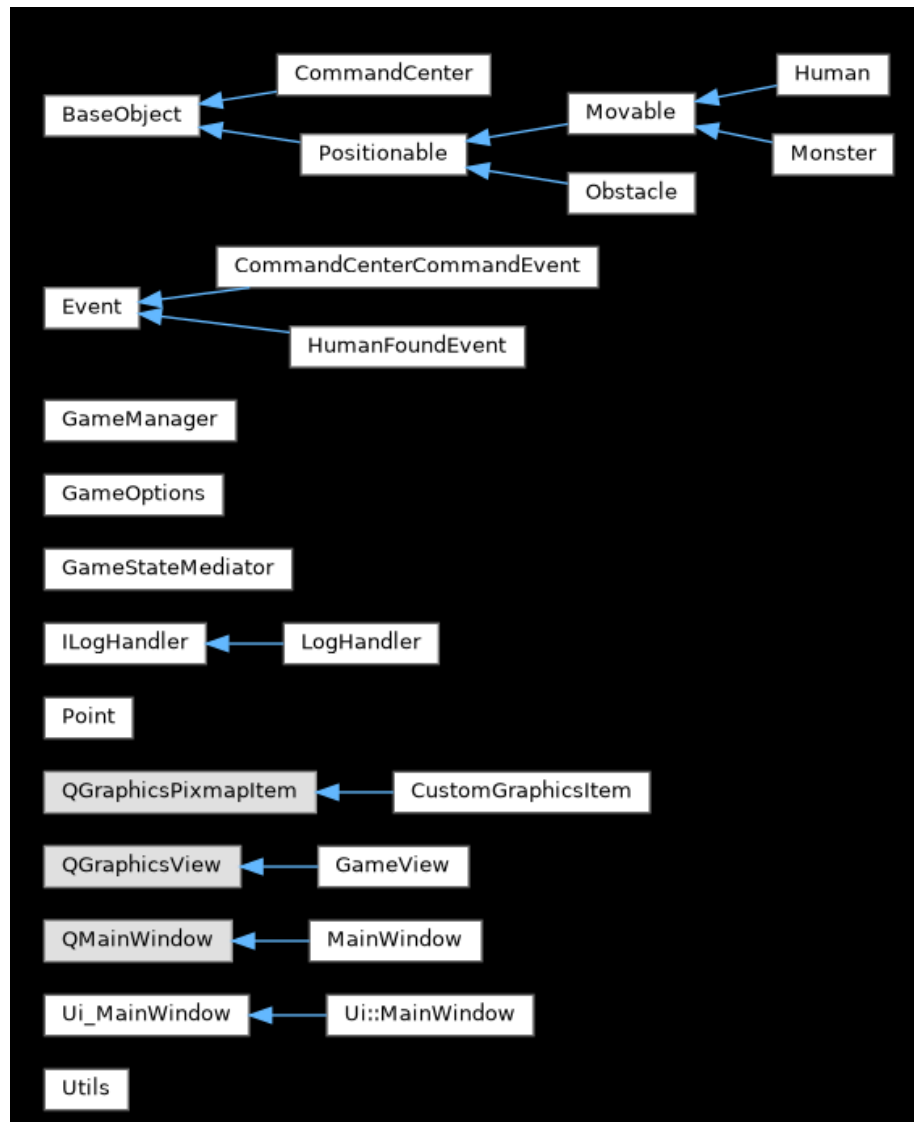
Rysunek 3: Zrzuty ekranu (kontynuacja)

3 Wykorzystywane środowisko i technologie

- system operacyjny: Ubuntu 22
- kompilator: GCC
- edytor kodu: QT Creator,
- dodatkowe biblioteki: QT 6

3.1 Podział na klasy

3.1.1 Diagram klas



Rysunek 4: Diagram klas wygenerowany przez Doxygen

3.1.2 Opis klas

Klasy logiki biznesowej:

- GameStateMediator – klasa implementująca wzorzec mediatora, umożliwiająca komunikację pomiędzy ludźmi a centrum zarządzania,
- GameManager – klasa obsługująca logikę rozgrywki,
- LogHandler – klasa obsługująca logi, to jest zapisywanie i odczytywanie wyników gier z pliku .log,
- Utils – klasa zawierająca statyczne metody publiczne niezwiązane jednoznacznie z jedną klasą.

Interfejsy:

- ILogHandler – abstrakcja wprowadzona, by umożliwić łatwą podmianę implementacji obsługi logów.

Enumy globalne:

- GameMode – enumerator używany, by informować o tym, jaki poziom rozgrywki został wybrany, używany po stronie UI,
- GameDifficulty – enumerator używany, by informować o tym, jaki poziom rozgrywki został wybrany, używany po stronie logiki biznesowej.

Obiekty gry:

- Point – klasa reprezentująca punkt na mapie o współrzędnych x, y,
- GameOptions – opcje gry, takie jak prędkości, ilość ludzi na mapie, zasięg wzroku czy zasięg złapania,
- BaseObject – bazowy obiekt, po którym dziedziczą wszystkie obiekty rozgrywki po stronie logiki biznesowej,
- Movable – bazowy obiekt reprezentujący wszystko, co może się ruszać,
- Positionable – bazowy obiekt reprezentujący wszystko, co ma pozycję na mapie,
- Human – obiekt reprezentujący wroga (człowieka),
- Monster – obiekt reprezentujący gracza (potwora),
- Obstacle – obiekt reprezentujący przeszkodę,
- CommandCenter – obiekt reprezentujący centrum zarządzania agentami.

Klasy związane z UI:

- CustomGraphicsItem – obiekt wyrenderowany na mapie,

- GameView – widok rozgrywki gry,
- MainWindow – główne okno aplikacji.

Zdarzenia:

- Event – bazowe zdarzenie,
- CommandCenterCommandEvent – zdarzenie wydawania przez centrum zarządzania rozkazu,
- HumanFoundEvent – zdarzenie zauważenia potwora przez ludzi.

4 Zespół

4.1 Członkowie

- Joanna Musialik – nr indeksu 304079
- Patryk Nikonowicz – nr indeksu 305918

4.2 Ogólny podział ról

Tabela 1: Podział ról

Implementowana funkcjonalność	Osoba odpowiedzialna
Elementy UI i ich obsługa	Joanna Musialik
Renderowanie mapy	Joanna Musialik
Kontrola poruszania się postaci	Joanna Musialik
Zapisywanie wyników rozgrywki lokalnie	Joanna Musialik
Ustawienie projektu np. cmake	Patryk Nikonowicz
Logika centrum dowodzenia	Patryk Nikonowicz
Obsługa zdarzeń	Patryk Nikonowicz
Obliczanie pozycji obiektów	Patryk Nikonowicz

5 Kluczowe linki

- Repozytorium na GitHubie:
<https://github.com/nikonowiczp/mission-impossible>

Spis tabel

1	Podział ról	11
---	-----------------------	----

Spis rysunków

1	Zrzuty ekranu	4
2	Zrzuty ekranu (kontynuacja)	5
3	Zrzuty ekranu (kontynuacja)	6
4	Diagram klas wygenerowany przez Doxygen	8