

SPIS TREŚCI

Opis projektu	2
Zasady gry	2
Cel projektu	2
Użyte narzędzia	2
Założenia funkcjonalne	2
Założenia нефunkcjonalne	3
Struktura projektu	3
Omówienie struktury oraz poruszanie się po dokumentacji	3
Link do strony z dokumentacją	3
Poruszanie się po dokumentacji	3
Opis głównych Klas w projekcie	5
Instrukcje instalacji.....	6
Instrukcje użytkowania	6
Instrukcje dotyczące uruchomienia.....	6
Opis interfejsu	6
Zasady gry.....	7
Uruchomienie i interakcja	7

PROJEKT GRY WŁASNEJ „TIC-TAC-TOE”

MACIEJ PŁUCIENNIK

OPIS PROJEKTU

ZASADY GRY

- https://pl.wikipedia.org/wiki/Kółko_i_krzyżyk

CEL PROJEKTU

Stworzenie własnej implementacji działającej gry w „Kółko i Krzyżyk” w języku C++.

UŻYTE NARZĘDZIA

Środowisko programistyczne CLion wraz z kompilatorem mingw.

ZAŁOŻENIA FUNKCJONALNE

- Stworzenie implementacji gry "Kółko i Krzyżyk" w języku C++.
- Generowanie planszy do gry.
- Możliwość sprawdzania, czy dany gracz ma szansę wygrać.
- Inicjalizacja graczy i możliwość gry z komputerem.
- Interfejs użytkownika umożliwiający interakcję, wybór rozmiaru planszy, liczby graczy i botów oraz symboli dla każdego z nich.
- Wyświetlanie współrzędnych pól na planszy dla łatwiejszej nawigacji.

ZAŁOŻENIA NIEFUNKCJONALNE

- Wykorzystanie środowiska programistycznego CLion z kompilatorem MinGW.
- Dokumentacja projektu, w tym struktura projektu, instrukcje użytkowania, opis klas i ich hierarchii, diagramy klas oraz lista plików nagłówkowych.
- Hostowana strona z dokumentacją projektu, umożliwiającą poruszanie się po dokumentacji.
- Wykorzystanie graficznych elementów, takich jak diagramy klas, strzałki i symbole, w celu wizualizacji relacji między klasami.
- Brak konieczności instalacji gry, wymagane jest jedynie środowisko programistyczne obsługujące język C++ i wbudowany kompilator.
- Prosty interfejs graficzny umożliwiający łatwe poruszanie się po grze i wprowadzanie danych przez użytkownika.

STRUKTURA PROJEKTU

OMÓWIENIE STRUKTURY ORAZ PORUSZANIE SIĘ PO DOKUMENTACJI

LINK DO STRONY Z DOKUMENTACJĄ

<https://xerochan.github.io/Tic/>

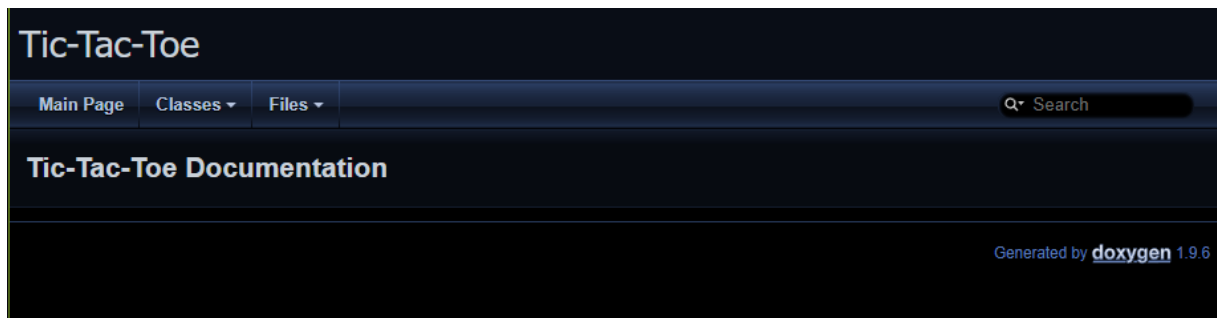
PORUSZANIE SIĘ PO DOKUMENTACJI

LEGENDA

- Niebieska strzałka służy do wizualizacji publicznej relacji dziedziczenia między dwiema klasami.
- Ciemnozielona strzałka służy do chronionego dziedziczenia.
- Ciemnoczerwona strzałka służy do dziedziczenia prywatnego.
- Fioletowa przerywana strzałka jest używana, jeśli klasa jest zawarta lub używana przez inną klasę. Strzałka jest oznaczona zmienną (zmiennymi), przez które wskazana klasa lub struktura jest dostępna.
- Żółta przerywana strzałka oznacza relację między instancją szablonu a klasą szablonu, z której została utworzona. Strzałka jest oznaczona parametrami szablonu instancji.

Sama legenda jest dostępna pod linkiem „legend” na stronie dokumentacji.

SPOSÓB PORUSZANIA SIĘ PO STRONIE

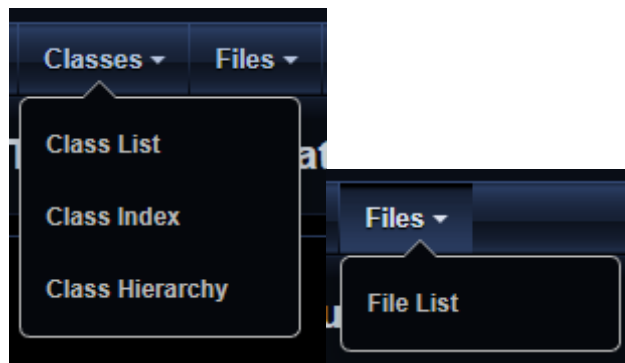


Rysunek 1 Strona główna

Po otwarciu linku do hostowanej strony z dokumentacją ukaże nam się powyższa strona główna.

Widzimy na niej tytuł projektu, zakładki Main Page, Classes, Files oraz pasek wyszukiwania „Search”.

Poniżej przedstawione są rozwinięte zakładki Classes i Files oraz ich krótki opis.



- Classes
 - ⇒ Class List – pokazuje listę klas
 - ⇒ Class Index – działa jak Class List, w ramach słownikowego spisu treści
 - ⇒ Class Hierarchy – pokazuje nam diagram klas, klasy są hiperłączami po których kliknięciu możemy przejść do detali
- Files – zawiera listę plików nagłówkowych projektu

Po otwarciu danej klasy zobaczymy przykładowo:

Game Class Reference

Public Member Functions | Protected Attributes | List of all members

Inheritance diagram for Game:

```
graph BT; Screen --> Game
```

Collaboration diagram for Game:

```
graph TD; Game -.-> Board
```

Public Member Functions

Game (const Game &other)=delete
Game & operator= (const Game &other)=delete
void play ()
void InitPlayers ()
int InitBoard ()
virtual void ViewBoard ()=0

Protected Attributes

Board * board

The documentation for this class was generated from the following files:

- **Game.h**
- **Game.cpp**

Widzimy diagramy danej klasy uwzględniając jej relacje z innymi klasami.

W zależności od klasy pod diagramami zobaczymy funkcję danej klasy z podziałem na zakres dostępu oraz atrybuty z takim samym podziałem. Gdy klasa jest klasą, która dziedziczy po innej klasie zobaczymy także listę funkcji i atrybutów, które zostały odziedziczone.

Na dole strony widzimy pliki, na podstawie których została wygenerowana dokumentacja danej klasy.

OPIS GŁÓWNYCH KLAS W PROJEKCIE

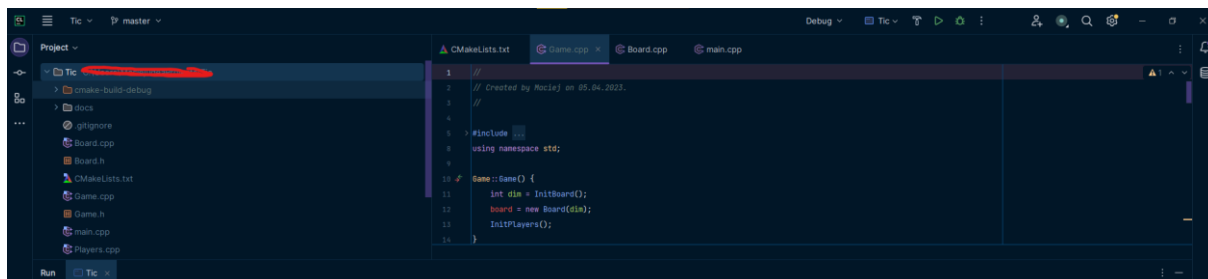
Głównymi klasami w moim projekcie są Game i Board. Klasa Board służy za generację planszy do gry oraz posiada metodę sprawdzania czy dany gracz ma szansę wygrać grę. Natomiast klasa Game jest odpowiedzialna za inicjalizację planszy poprzez wskaźnik na klasę Board, inicjalizuje ona także graczy, którzy będą korzystali z programu. Dostępna jest także funkcja gry z komputerem.

INSTRUKCJE INSTALACJI

Gra nie wymaga żadnej instalacji, potrzebne jest natomiast środowisko programistyczne obsługujące język C++ oraz kompilator, który jest zazwyczaj wbudowany w to środowisko.

INSTRUKCJE UŻYTKOWANIA

INSTRUKCJE DOTYCZĄCE URUCHOMIENIA



Rysunek 2 Ekran środowiska

W przypadku użycia środowiska CLion wystarczy utworzyć nowy projekt oraz do niego przekopiować pliki gry. CLion domyślnie generuje plik main.cpp, który będzie zaczytywany do wygenerowanego pliku .exe. Oszczędza to czas spędzony na konfiguracji samego środowiska.

OPIS INTERFEJSU

```
Enter board dimension: 5
Enter number of players who going to play:
1
Enter number of bots to play against:
1
Player 1 chooses symbol: 0
Bot 1 chooses symbol: X
Screen view:
  1  2  3  4  5
+---+---+---+---+
1 |  |  |  |  |  |
+---+---+---+---+
2 |  |  |  |  |  |
+---+---+---+---+
3 |  |  |  |  |  |
+---+---+---+---+
4 |  |  |  |  |  |
+---+---+---+---+
5 |  |  |  |  |  |
+---+---+---+---+
0's turn
Enter the field as row[space]column from range (1-5):
```

Rysunek 3 Proces inicjalizacji gry

ZASADY GRY

https://pl.wikipedia.org/wiki/Kółko_i_krzyżyk

URUCHOMIENIE I INTERAKCJA

Po uruchomieniu i skompilowaniu projektu jesteśmy pytani o to (w kolejności od góry, linia po linii) jak duża ma być nasza plansza (tutaj 5x5), ile graczy będzie grało, ile botów będzie grać (komputer losujący pola), oraz jakie symbole wybiera każdy z graczy/botów. Interfejs zawiera także współrzędne pól do prostszej, bardziej intuicyjnej nawigacji.