

Manual

Zadanie, które program ma realizować

Program, napisany w języku C++ ma implementować strategiczną grę planszową „Pentago” (<https://pl.wikipedia.org/wiki/Pentago>). Jest to gra dla dwóch osób na planszy o wymiarach 6x6. Interfejs użytkownika ma być ograniczony do trybu tekstowego w konsoli Windowsa.

Lista opcji do wyboru z krótkim opisem każdej z nich

Program przyjmuje opcjonalny parametr wejściowy, który jest ścieżką do pliku z planszą. Składnia:
> pentago.exe [ŚcieżkaDoPliku]

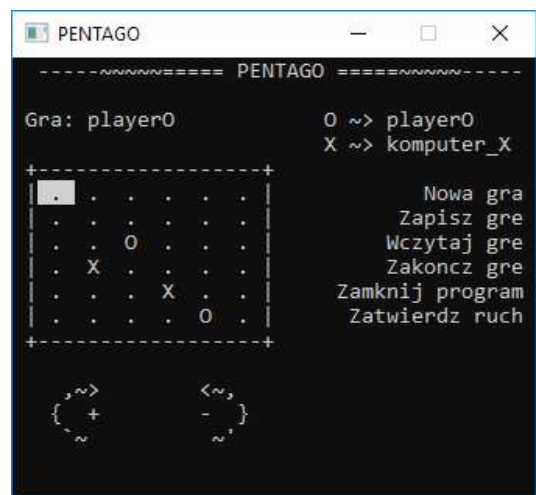
Wówczas, w przypadku pomyślnego załadowania planszy, pojawia się plansza z grą, przyciski wyboru obrotów i menu. Z programu można korzystać za pomocą klawiatury, jak i myszy. Interakcja z użytkownikiem poprzez wpisywanie tekstu została ograniczona do minimum, w grę można zagrać używając jedynie przycisków: „Enter”, „Tab” oraz strzałek albo samej myszy.

Opcje w menu:

- „Nowa gra” – umożliwia skonfigurowanie parametrów gry i rozpoczęcie rozgrywki. Jeśli trwa już gra, jej zakończenie należy potwierdzić. Użytkownik proszony jest o wybór rodzajów graczy:
 - Człowieka, wówczas wyświetla się pole do wprowadzenia imienia gracza (dozwolone są znaki alfanumeryczne oraz ‘_’, ‘.’ i ‘ '), a liczba dowolnych dostępnych znaków dla domyślnego układu interfejsu wynosi 15. Długość napisu oraz wpisywane znaki kontrolowane są w trakcie jego wprowadzania. Zabroniony znak nie zostanie wpisany. Po przekroczeniu dozwolonej długości kolejne znaki są ignorowane, a użytkownik informowany jest o przekroczeniu.
 - Komputera, wówczas można wybrać poziom trudności:
 - „Banalny”,
 - „Latwy”,
 - „Sredni”,
 - „Trudny”.

Wybierając graczy, należy mieć na uwadze, że gracz ze znakiem ‘O’ jako pierwszy wykonuje ruch. Po ustawieniu wszystkich potrzebnych opcji, rozpoczyna się gra.

- „Zapisz gre” -- jeśli trwa gra, można jej stan zapisać do pliku tekstowego, np. w celu późniejszego wczytania. Użytkownik proszony jest o podanie nazwy pliku, dozwolone znaki – jak dla imienia. Jeśli plik o podanej nazwie już istnieje oraz w przypadku innych błędów, wyświetla się komunikat.
- „Wczytaj gre” -- analogicznie, można wczytać wcześniej zapisaną grę. Należy najpierw potwierdzić przerwanie bieżącej rozgrywki. W przypadku niepowodzenia pojawia się



komunikat. Program nie gwarantuje poprawnego działania, gdy plik ze stanem gry został edytowany ręcznie przez użytkownika, nawet, jeśli uda się go załadować.

- „Zakończ grę” -- po potwierdzeniu kończy grę.
- „Zamknij program” -- po potwierdzeniu końca gry (jeśli trwa gra), opuszcza program.
- „Zatwierdź ruch” -- zatwierdza obecnie wybrany ruch. Tej akcji odpowiada wciśnięcie przycisku „Enter”, gdy nie jest wybrana żadna inna pozycja w menu.

Gra polega na naprzemiennych ruchach każdego z dwóch graczy. Każdy ruch składa się z postawienia swojego znaku na jednym z niezajętych pól planszy. Po zatwierdzeniu wyboru pola, należy wybrać ćwiartkę do obrotu o 90° i kierunek obrotu. Ten ruch również należy zatwierdzić (przycisk „Enter” lub „Zatwierdź ruch” z menu). W przypadku korzystania z klawiatury, przycisk „Tab” przełącza pomiędzy wyborem ruchów a menu, a przypadku wykonywania obrotu – również pomiędzy wyborem ćwiartki i kierunku obrotu. Po każdym obrocie sprawdzany jest warunek wygranej (5 takich samych znaków w rzędzie lub na przekątnej).

Nietypowe zachowania programu

- W przypadku próby załadowania interfejsu innego niż dostarczony z programem: jeśli nawet uda się ją wczytać, program może nie działać prawidłowo. Autorka nie przewidziała możliwości ładowania planszy znacząco innych niż dostarczona.
- W przypadku załadowania pliku ze stanem gry, który został zmodyfikowany: jak wyżej.
- W przypadku szybkiego wpisywania tekstu do pól tekstowych oraz szybkich akcji myszy: W celu uniknięcia zarejestrowania przez program dwa lub więcej razy tej samej akcji, dodany zostało oczekiwanie 200 ms przed wyczyszczeniem bufora wejścia. Przez ten nie są rejestrowane żadne akcje. Jest to szczególnie odczuwalne przy wpisywaniu testu (imiona i nazwy plików). Problem wielokrotnych akcji dotyczył głównie akcji klawiatury, niekiedy myszy.
- Program obsługuje jedynie akcje związane klawiszami „Tab”, „Enter” oraz ze strzałkami. Pomijając te trzy klawisze, żadne powiązania przycisków z akcjami, np. „Esc” z powrotem czy spacji z zatwierdzeniem, nie zostały zaimplementowane. Co więcej, ponieważ program przechwytuje wszystkie dane wejściowe do bieżącego okna konsoli, na oknie nie będą działać żadne z domyślnych skrótów klawiszowych dla okien (np. „Alt+F4”) ani prawy przycisk myszy (pomijając pasek tytułowy). Za programu można zatem wyjść albo dzięki przyciskowi na pasku tytułowym albo poprzez menu.
- Nie działają strzałki prawo/lewo przy edycji tekstu (imię, nazwa pliku).
- Nie można zmienić rozmiaru okna z programem podczas jego działania. Opcja została wyłączona, ponieważ program działa poprawnie tylko wtedy, gdy okno jest nie mniejsze niż rozmiary planszy.
- Zdarza się, że program po uruchomieniu źle wyświetla interfejs. Wymuszenie odświeżenia konsoli (np. klikając w okno programu) najczęściej rozwiązuje problem.
- Może się zdarzyć, że program przestanie reagować na akcje klawiszowe, np. po wpisywaniu znaków, kiedy nie jest to oczekiwane. „Odblokować” można poprzez akcję myszą.
- Ustawiając dwóch graczy – komputerów, blokuje się działanie programu aż do skończenia gry, ~kilka minut. Czas ten można poświęcić na obserwowanie rozgrywki.

Niezgodności z założeniami przekazanymi w treści zadania

Nie stwierdzono niezgodności, które wynikałyby wprost z pliku z treścią zadania:

<https://troja.uksw.edu.pl/zasoby/zadania-semestralne-2019.pdf> (ostatnio sprawdzane 04.06.2019)

Składnia danych wejściowych i wyjściowych

Struktura plików, typy danych przechowywane w poszczególnych polach pliku

Autorka programu nie przewidziała możliwości tworzenia przez użytkownika własnych plików wejściowych dla programu lub ich modyfikacji, jednak dopóki będą zachowane poniższe reguły dla danych zawartych w plikach, program powinien działać prawidłowo (ale nie ma na to gwarancji). Wszystkie (2) rodzaje plików zawierają dane tekstowe, które wczytywane są do zmiennych znakowych (char, char[][] lub string). Oczekuje się, że dane te będą z zakresu obsługiwanego przez program (alfanumeryczne, '_', ' ', '.').

Program jako argument przyjmuje nazwę pliku tekstowego z interfejsem. W przypadku niepodania argumentu, interfejs wczytywany jest z dostarczonego z programem pliku „board.txt”, którego oczekuje w lokalizacji, z której uruchamiany jest program. Plik ten zawiera stałe elementy interfejsu gry w formacie tekstowym, dokładnie taki, jaki pojawi się na ekranie. Program podczas startu wyszukuje pierwszego wystąpienia charakterystycznych elementów interfejsu – znaczników, których współrzędne służą do wyświetlania zmieniających się elementów (np. zmiana zaznaczonego obrotu). Te elementy to:

- ". ";
- ",~>",
- "Nowa gra",
- "O ~>",
- "X ~>",
- "Gra:",
- "ruch".

Wpływ zmiany położenia tych elementów na grę nie był testowany, aczkolwiek ich niewielkie przesunięcia i dodatek elementów, które są różne od znaczników lub leżą poniżej ich nie powinien wpłynąć znacząco na zachowanie programu. W przypadku niewykrycia któregoś ze znaczników, program nie uruchomi się.

Stan gry można zachować w pliku testowym. Nie jest on przeznaczony do edycji przez użytkownika. Plik taki składa się z następujących pól, z których każde umieszczone jest w osobnej linii w podanej kolejności:

- imię gracza grającego symbolem 'O',
- rodzaj gracza 'O' ('C' - człowiek, 'K' - komputer),
- imię gracza grającego symbolem 'X',
- rodzaj gracza 'X' ('C' - człowiek, 'K' - komputer),
- poziom trudności dla gry przeciwko komputerowi: 'R' (losowo), 'A' (łatwo), 'B' (średnio) lub 'C' (trudno); jeśli oboje graczy to ludzie, pole to nie występuje; nie jest przewidziane zapisywanie stanu gry komputer vs komputer, gdyż wówczas nie można zapisać stanu gry,
- symbol gracza, do którego należy bieżący ruch ('O' albo 'X'),
- rodzaj ruchu ('pole' albo 'rotacja'),
- wartości pól na planszy, każdy rząd w osobnej linii, wszystkie rzędy po kolei, znaki kodowane są symbolami 'O', 'X' i '.' dla pustego pola; pomiędzy polami w rzędzie nie powinno być innych znaków.

Spacje poprzedzające i kończące każdy wiersz są odrzucane. Imiona wczytywane są do miejsca, na które pozwala szerokość pola do jego wyświetlenia – dalsza część jest ignorowana. Znaki

składające się na imię nie są sprawdzane. Poprawność ustawień graczy i miejsca w grze jest rygorystycznie sprawdzana. Stan planszy jest sprawdzany pod kątem braku symboli innych niż 'O', 'X' i '.' oraz zbyt małej ich liczby w wierszu. Poza tym, poprawność stanu planszy nie jest sprawdzana merytorycznie. Do projektu dołączono przykładowy plik „savedgame.txt” z zapisanym stanem gry.

Struktura katalogów niezbędna dla poprawnego działania programu

Wszystkie pliki niezbędne do działania programu – plik wykonywalny („pentago.exe”) oraz plik tekstowy z interfejsem („board.txt”) nie muszą być w tym samym folderze.

Program przyjmuje swój katalog roboczy jako domyślną lokalizację pliku z interfejsem („board.txt”) oraz plików ze stanami gry. Jeśli plik „board.txt” nie znajduje się w lokalizacji, z której uruchamiany jest program, należy podać jego lokalizację jako argument wejściowy.

Opis kodu

Lista plików z kodem źródłowym wchodzących w skład programu

AI.cpp	AI.h	Klasa obsługująca ruchy komputera
BoardSel.cpp	BoardSel.h	Klasa zaznaczeń na planszy, sprawdzanie wygranej.
Game.cpp	Game.h	Klasa gry, obsługa stanu gry.
Game_io.cpp	Game.h	Klasa gry, wczytywanie/zapisywanie do pliku.
Game_menu.cpp	Game.h	Klasa gry, obsługa menu.
helpers.cpp	helpers.h	Funkcje pomocnicze, głównie przeciążone operatory dla COORD, klasa listy dynamicznej.
MenuSel.cpp	MenuSel.h	Klasa głównego menu, klasa bazowa dla MenuSelInput.
MenuSelInput.cpp	MenuSelInput.h	Klasa menu z pobieraniem tekstu.
QSel.cpp	QSel.h	Klasa obsługująca zdarzenia dla zaznaczania ćwiartek.
RotSel.cpp	RotSel.h	Klasa obsługująca zdarzenia dla przycisków rotacji.
Screen.cpp	Screen.h	Klasa obsługująca większość komunikacji z konsolą, wczytywanie interfejsu.
Selection.cpp	Selection.h	Klasa bazowa dla zaznaczenia.
	Player.h	Klasa reprezentująca dane gracza.
Inne pliki dostarczane z projektem		
board.txt		Plik z interfejsem, niezbędny do uruchomienia gry.
savedgame.txt		Przykładowy plik z zapisaną grą.

Diagram klas

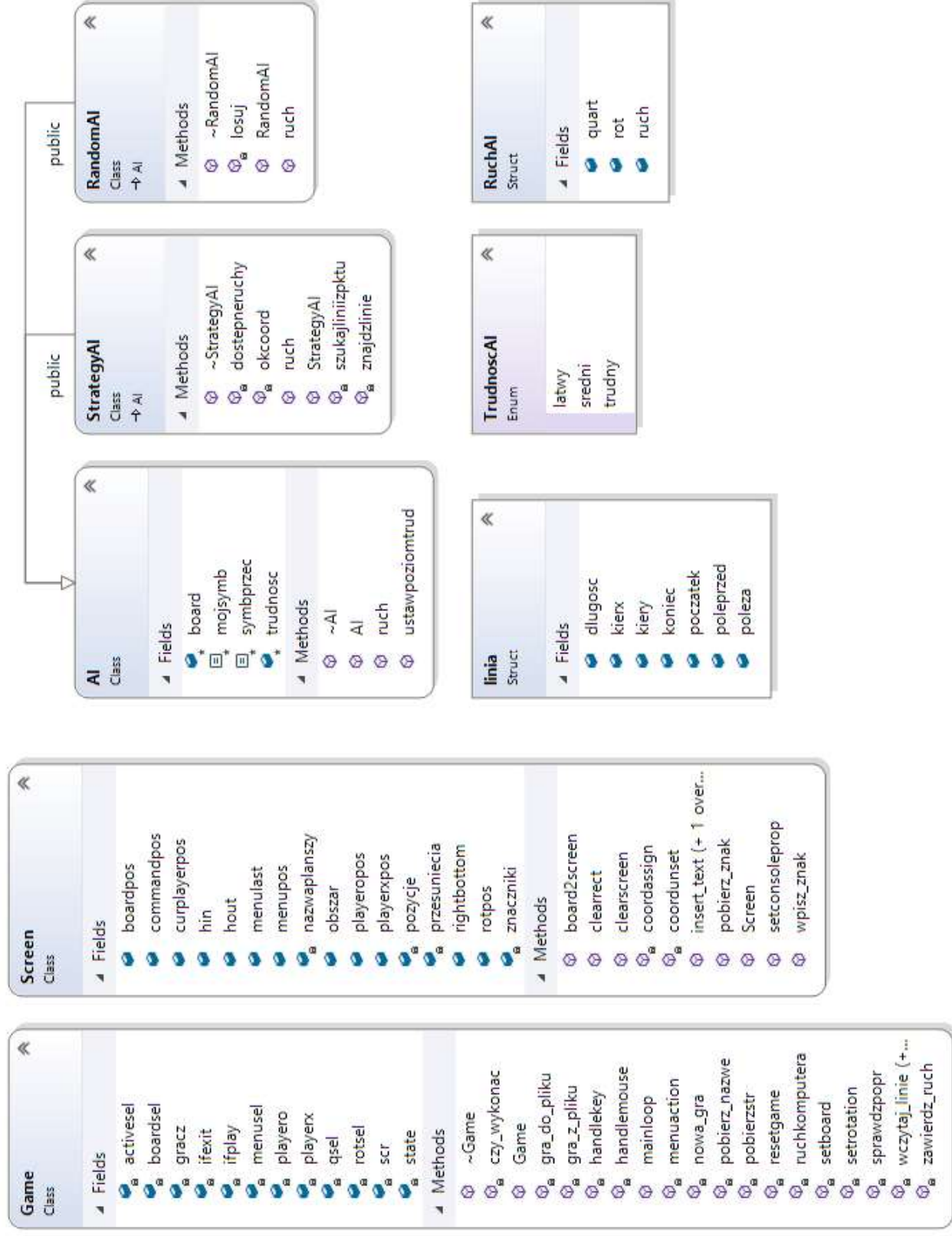


Diagram klas

