**Projekt zaliczeniowy**

**JZO**

**Bartłomiej Płóciennik**

1. **Opis gry**

* Gracz (oznaczony „P”) pojawia się w lewym górnym rogu planszy. Jego przeciwnik (oznaczony „E”) z kolei rozpoczyna grę w prawym dolnym rogu tej samej planszy.
* Zadaniem gracza jest zebranie większej ilości punktów niż przeciwnik, przez którego nie może zostać złapany. Gracz musi przy tym pamiętać, że posiada ograniczenia w zakresie możliwych do wykonania ruchów (100).
* Za każdą zdobytą monetę gracz zdobywa 1pkt, a przeciwnik 10pkt
* Przeciwnik wykonuje ruchy odwrotne do gracza   
  (gracz porusza się w dół -> przeciwnik w górę itd.)
* Na planszy znajdują się cały czas cztery monety (oznaczone „$”). Jedna moneta odpowiada jednej ćwiartce planszy, na której toczy się gra. Po zebraniu monety z danej ćwiartki w jej miejsce generuje się nowa (na tej samej ćwiartce).

1. **Założenia funkcjonalne**
   * Sterowanie graczem z pomocą klawiatury (poprzez standardowe cin lub getch())
   * Odświeżanie stanu planszy po wczytaniu kolejnego ruchu gracza
   * Oznaczenie gracza jako „P”, przeciwnika jako „E”, monety jako „$”
   * Generowanie się monet w losowych miejscach danej ćwiartki planszy
   * Rozgrywka toczy się na planszy 15x15
   * Ramka planszy wykonana jest z „#”, a pola dostępne dla gracza z „.”
   * Przechowywanie i pokazywanie współrzędnych oraz wyniku gracza  
     i przeciwnika
   * Gracz może wykonać ograniczoną liczbę ruchów (100)
   * Gracz rozpoczyna rozgrywkę na pozycji (1,1)
   * Pozycja przeciwnika jest cały czas lustrzanym odbiciem pozycji gracza
   * Zdobycie monety przez gracza zwiększa jego wynik o 1 pkt, zaś zdobycie monety przez przeciwnika zwiększa jego wynik o 10 pkt.
   * Zakończenie rozgrywki w przypadku „złapania” przez przeciwnika
   * Ukazanie końcowego wyniku rozgrywki po wyczerpaniu się puli dostępnych ruchów
2. **Założenia niefunkcjonalne i jakościowe**
   * Realizacja w środowisku CodeBlocks
   * Podział struktury kodu na klasy o osobnych plikach .h i .cpp
   * Łatwość w odczytaniu elementów na planszy
3. **Diagram klas i struktura projektu**

**Obraz zawierający diagram

Opis wygenerowany automatycznie**

* Klasa Object jest zaprzyjaźniona z klasą Silnik
* Klasa Tablica jest zaprzyjaźniona z klasą Silnik
* Klasa Keyboard jest zaprzyjaźniona z klasą Silnik
* Klasa KeyboardGetch dziedziczy po klasie Keyboard
* Klasa Gra dziedziczy po klasie Silnik
* Klasa GraCMD dziedziczy po klasie Gra

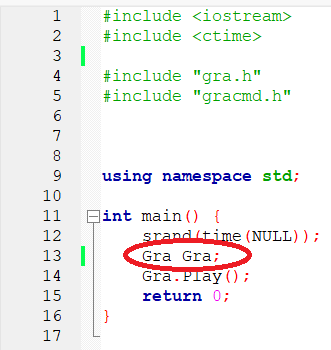
**Obraz zawierający tekst

Opis wygenerowany automatycznie**

1. **Instrukcja obsługi**

W celu uruchomieniu programu należy:

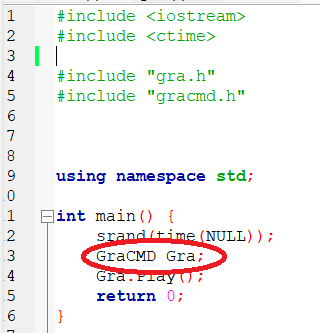
* Uruchomić plik „project.cbp” z pomocą środowiska CodeBlocks
* Skompilować program, upewniając się wcześniej, że plik main.cpp jest w jednej z dwóch następujących postaci (zmianie ulega jedynie element zaznaczony na czerwono):



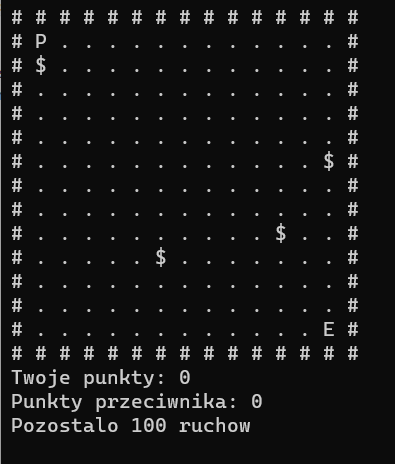
W tym wypadku po udanej kompilacji zamiast planszy z grą widoczny będzie jedynie uproszczony widok gry. Całość jest jednak w pełni funkcjonalna:

Obraz zawierający tekst

Opis wygenerowany automatycznie



W tym wypadku po udanej kompilacji widoczna będzie w pełni funkcjonalna plansza z grą.



Od tego momentu można uruchamiać program z pomocą pliku wykonywalnego „project.exe”, który został utworzony w wyniku kompilacji.

* Sterowanie postacią odbywa się z pomocą klawiszy:
  1. W/w – góra
  2. S/s – dół
  3. A/a – lewo
  4. D/d – prawo
* Sposób odczytywania klawiszy sterujących można zmienić w pliku silnik.h (zmianie ulega jedynie element zaznaczony na czerwono):

Obraz zawierający tekst

Opis wygenerowany automatycznie

W tej sytuacji sterowanie odbywa się „na żywo”. Wciśnięcie danego przycisku jest zatwierdzane automatycznie, więc gracz poruszy się od razu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

W tym wypadku każdy wpisany ruch trzeba zatwierdzać klawiszem „Enter”. Można jednak wprowadzić dłuższą ścieżkę, a następnie ją zatwierdzić. Przykładowy można wprowadzić „ssaaw” i dopiero to zatwierdzić przyciskiem „Enter”, a postać gracza wykona wszystkie (oczywiście możliwe do wykonania) ruchy z tej kolejki.