

DOKUMENTACJA PROJEKTU

POPULARNA GRA "STATKI"

Celem projektu jest napisanie gry konsolowej lub okienkowej w technice Programowania obiektowego (ang. object-oriented programming, OOP). Zasady gry:

- I. Na jednej planszy gracz zaznacza koordynację swoich statków, a na drugiej zaznacza nietrafione strzały w statki przeciwnika oraz oznaczenia trafionych okrętów.
- II. Statki mogą być narysowane w pionie i poziomie, ale nie mogą stykać się ze sobą bokami ani rogami i nie mogą być „zakręcone” czyli przypominać litery L.
- III. Gra przebiega naprzemiennie: najpierw jeden z graczy oddaje strzał, jeśli uda się mu trafić, może kontynuować „strzelanie” do momentu, aż chybi
- IV. Przeciwnik „komputer” swoje strzały będzie generował randomowo
- V. Zatopienie statku ma miejsce, kiedy wszystkie kratki symbolizujące ten okręt zostaną „trafione” przez drugiego gracza.
- VI. O oddaniu nietrafionego strzału informujemy słowem „pudło”. Natomiast przy strzale trafionym mówimy „trafiony” lub „trafiony-zatopiony”.
- VII. Na planszach 10 na 10 przyjmuje się, że każdy z graczy dysponuje flotą złożoną z:
 - jednego czteromasztowca o wielkości czterech kratek
 - dwóch trójmasztowców o wielkości trzech kratek
 - trzech dwumasztowców o wielkości dwóch kratek
 - czterech jednomasztowców o wielkości jednej kratki.

OPIS KLAS

- Klasa Plansza – będzie odpowiedzialna plansze 10x10 czyli podstawową strukturę gry. W niej zawarte będą funkcje, które odpowiadają za :
 - pokazywanie planszy z trafieniami i chybieniami
 - czy umiejscowienie statków jest prawidłowe, czy się ze sobą nie stykają, a także czy nie wychodzą za rozmiar planszy
 - jeśli umiejscowienie będzie zgodne z kryteriami to będzie podmieniać znak z planszy na znak statku
 - liczbę trafień po których zakończy się gra
- Klasa Statek – będzie miała takie zmienne jak współrzędne x i y, statek będzie reprezentowany w grze jako dynamiczna tablica tworzona w konstruktorze. W klasie będzie funkcja składowa, która będzie odpowiedzialna za sprawdzenie czy statek jest „trafiony” i określa kiedy będzie „zatopiony”, a także funkcja, która sprawdzi czy strzał jest „trafiony” czy „pudło”
- Klasa Gracz – będzie prostą klasą podstawową, z atrybutami typu nazwa, imię, zmiennymi typu bool, które będą niosły informacje czy gracz jest automatyczny czy jest komputerem,
- Klasa Komputer i Człowiek – klasy będą odpowiedzialne za kolejne ruchy obu graczy oraz wybranie położenia statków przed rozpoczęciem gry, cała mechanika gry, losowanie liczb pseudolosowych aby zapewnić grywalność z komputerem tzw. „sztuczną inteligencją”

SPIS FUNKCJI

- Funkcja menu – odpowiedzialna za wstępne menu i start gry
- Użycie biblioteki <random>, aby stworzyć generator liczb losowych
- Postaram się stworzyć grę w formie okienkowej aplikacji, po wstępnym przeszukaniu i zdobyciu wiedzy myślę o użyciu do tego celu biblioteki QT i środowiska QT Creator
- Klasa Statek
 - wez_x() – popiera wartość zmiennej x z prywatnych danych składowych klasy
 - wez_y() - popiera wartość zmiennej y z prywatnych danych składowych klasy
 - wez_x() – ustawia wartość zmiennej x z prywatnych danych składowych klasy
 - wez_y() - ustawia wartość zmiennej y z prywatnych danych składowych klasy
 - Sprawdz_trafienie() – funkcja sprawdza czy strzał trafia w statek, jeśli tak to sprawdza czy statek tonie czy nie, pętla for przebiega po rozmiarze statku szuka literki S, jak statek jeśli jest jeszcze taka literka to oznacza to, że statek jeszcze nie zatonął, w przeciwnym wypadku ustawia atrybut na zatonięty.
- Klasa Plansza
 - Rysuj_plansze() – rysuje plansze w konsoli
 - Rysuj_plansze_trafien() – rysuje w konsoli drugą plansze
 - Ustaw_Statek() – sprawdza czy ustawienia statków przez graczy jest prawidłowe
 - Sprawdz_trafienie()- funkcja wywołuje funkcję Sprawdz_trafienie() z obiektu klasy statek, sprawdzi co zwraca i ustawi na planszy znak „trafienia” bądź „zatonięcia”
- Klasa Gracz
 - bool jaki_gracz() -zwraca rodzaj gracza
 - ustaw_nazwa() – ustawia wartości atrybutów
 - ustaw_numer() - ustawia wartości atrybutów
 - ustaw_rodzaj_gracza() - ustawia wartości atrybutów
 - wez_numer() – zwraca wartość atrybutu
 - wez_nazwe() - zwraca wartość atrybutu
- Klasa Człowiek i Komputer
 - Nastepnych_Ruch(Plansza &b)- następny ruch gracza, kontrola błędów
 - Nastepnych_Ruch(Plansza &b) w klasie komputer- generator liczb losowych, strzelanie komputera, algorytm odpowiedzialny za losowanie liczb w bliskim otoczeniu „trafienia”
 - Ustaw Statki()- ustawia statki na początku gry, losowe ustawienie

HIERARCHIA



