

# Projekt POCity

Makymilian Czudziak

## Instrukcja dla użytkownika

### Uruchamianie gry

Projekt został napisany w Visual Studio na systemie Windows 10. Dlatego aby uruchomić program, należy go skompilować wewnątrz tego właśnie IDE, otwierając w VS plik "POCity.csproj". Po kompilacji i uruchomieniu otworzy się okno terminala i aby gra była poprawnie wyświetlana to czcionką (terminala) powinno być "MS Gothic", "NSimSun" lub "NSimSun-ExtB". Polecana jest pierwsza pozycja.

### Cel gry

Celem gracza jest zbudowanie miasta wartego jak największą ilość punktów za punktowane budynki.

Budynki punktowane to:

- Mieszkalny - 5 pkt
- Komercyjny - 4 pkt
- Przemysłowy - 3 pkt

Budynki punktowane to budynki strefowe, których potrzeby zostały spełnione, tym samym operujące. W przeciwnym przypadku nie są punktowane.

# Opis planszy i budynków

Gra rozgrywa się na mapie 20 na 20 pól. Na każdym z pól można wybudować nieruchomość.

Nieruchomości dzielą się następująco:

- Drogi:
  - Autostrady
  - Aleje
  - Zwykłe
- Budynki:
  - Strefowe:
    - Mieszkalne
    - Komercyjne
    - Przemysłowe
  - Specjalne:
    - Podstawowe:
      - Elektrownie
      - Wieże ciśnień
    - Publiczne:
      - Szpital
      - Posterunek Policji
      - Straz Pożarna
    - Inne:
      - Park
      - Szkoła

Budynki można stawiać tylko na pustych polach oraz przy drogach, czyli

w sąsiedztwie bezpośrednim z nieruchomością typu "Droga".

Rola wszystkich Specjalnych budynków sprowadza się do zaspokajania potrzeb budynków strefowych. Każdy budynek specjalny posiada promień, wewnątrz którego budynki strefowe mają spełnioną daną potrzebę.

np. Budynek Komercyjny potrzebuje prądu. Jeśli jest w odległości 7 od Wieży Ciśnienie to zaspokojoną potrzebę posiadania wody.

Promienie obsługi kolejnych budynków specjalnych:

- Podstawowe:
  - Elektrownie - 8
  - Wieże ciśnienie - 7
- Publiczne:
  - Szpital - 10
  - Posterunek Policji - 10
  - Straż Pożarna - 10
- Inne:
  - Park - 4
  - Szkoła - 6

Potrzeby budynków:

- Specjalny:
  - Prąd
  - Woda

- Przemysłowy:
  - To samo co Specjalny
  - Policja
  - Straż pożarna
  - Szpital
- Komercyjny:
  - To samo co Przemysłowy
  - Park
- Mieszkalny
  - To samo co Komercyjny
  - Szkola

Jeśli budynek specjalny nie ma zaspokojonych potrzeb to jego promień obsługi wynosi 0 (jest nieoperatywny).

Nie ma możliwości burzenia budynków. Dlatego należy je stawiać w przemysłany sposób.

## Obsługa gry

Gra posiada interfejs całkowicie tekstowy. Natomiast cały czas po uruchomieniu gry wyświetlana jest cała plansza, złożona z budynków, które się na niej znajdują.

Użytkownik komunikuje się z grą komendami. Po komendzie "build" i "info" należy wpisać współrzędne(x y) na mapie, gdzie chce się budować lub zapytać o informacje.

- build - umożliwia budowę budynku

- Highway - budowa autostrady
  - Avenue - budowa alei
  - PlainRoad - budowa zwykłej drogi
  - Residential - budowa strefy mieszkalnej
  - Commercial - budowa strefy komercyjnej
  - Industrial - budowa strefy przemysłowej
  - PowerPlant - budowa elektrowni
  - Watertower - budowa wieży z wodą
  - FireDept - budowa straży pożarnej
  - PoliceStation - budowa policji
  - Hospital - budowa szpitala
  - Park - budowa parku
  - School - budowa szkoły
- info - wypisuje poniżej/obok mapy informacje o zadanym budynku
  - points - podlicza i wypisuje liczbę punktów jaką jest warte miasto
  - draw - czysci i rysuje ponownie interfejs
  - save - zapisuje grę
  - load - wczytuje grę
  - end - kończy grę, zlicza i wypisuje punkty.

## Przykłady komend:

*build 10 2 Avenue*

*build 11 2 PowerPlant*

*info 11 2*

# Opis implementacji

Projekt został napisany w Visual Studio na systemie Windows 10.

Dlatego też polecane jest by kod przeglądać tym samym IDE (Uruchamiając plik "POCity.csproj").

## Schemat UML

Znajduje się w tym samym folderze, plik o nazwie Diagram\_UML.png

## Logika klas

Dlatego, że klasa Program jedyne co robi to tak naprawdę uruchamia interfejs tekstowy to posiada jedną linijkę.

Dlatego też najważniejsze klasy i ich najważniejsze funkcje to:

- TextInterface - klasa skupiająca cały intersejs, czyli przyjmowanie wejścia od użytkownika i reagowanie na nie w postaci przekazywania informacji do mapy i wypisywania informacji zwracanych przez mapę.
  - Run() - porusza interfejs do pracy, to właśnie jest wywoływane przez Main()
  - WriteMap() - wypisuje mapę w ładnej ramce ze wszystkimi budynkami
  - DrawNews() - wypisuje informacje o zadanym budynku
  - SaveGame() - zapisuje grę do pliku

- LoadGame() - wczytuje grę z pliku
- Build() - buduje budynek o zadanej nazwie
- Map - klasa posiadające tablicę dwuwymiarową obiektów Property na których przeprowadza operacje, jednocześnie na tyle duża, że rozbiłem ją na dwa pliki
  - public string RaiseBuilding(int x, int y, string building\_name) - buduje budynek na zadanej pozycji typu opisanego w przekazywanym stringu, sprawdza, czy może budować, następnie znanajduje jego sąsiadów i wymienia się z nimi informacją co może im dać i co może od nich wziąć(co zależy od promienia)
  - CountPoints() - pyta budynki ile są warte punktów, te punktowane sprawdzają, czy są operatywne i informują o tym
- Property - klasa abstrakcyjna, po której dziedziczą wszystkie budynki
  - int x, int y, string my\_name - każdy budynek wie czym jest i na jakiej pozycji się znajduje
  - string ToString() - wie jak powinien wyglądać na mapie po wypisaniu
  - void GetToKnow(Type new\_neighbour) - funkcja (wywoływana przez mapę) zapoznająca go z jego nowym sąsiadem, ale również gdy sam poznaje nowych sąsiadów, bo jest nowy w okolicy
  - string IsWorking() - każdy budynek umie określić, czy jest operatywny i dostarcza tego co jest w stanie
  - int WorthPoints() każdy budynek wie ile jest wart

punktów, oczywiście tylko budynki punktowane zwracają  
wartość niezerową