

SPECYFIKACJA FUNKCJONALNA

Projekt zespołowy

Łukasz Knigawka

Mateusz Smoliński

19 grudnia 2018

Spis treści

1	Wstęp teoretyczny	2
2	Funkcje programu	2
3	Opis interakcji użytkownika z programem	2
4	Sytuacje wyjątkowe	2
5	Zarys testów akceptacyjnych	3

1 Wstęp teoretyczny

todo

2 Funkcje programu

Program posiada trzy główne funkcje:

1. rysowanie planszy na podstawie danych z pliku,
2. modyfikowanie planszy za pomocą prostych narzędzi z przybornika,
3. pokazywanie list mieszkańców oraz obiektów na wybranym przez użytkownika obszarze.

Użytkownik otrzymuje powyższe główne funkcje dzięki poniższym funkcjom pomocniczym:

- wczytanie danych z pliku tekstowego,
- wczytanie grafiki tła,
- dodawanie elementów konturu planszy,
- dodawanie punktów kluczowych dzielących kontur na obszary,
- dodawanie obiektów o typach zadeklarowanych w pliku tekstowym,
- wypisywanie list z liczbą mieszkańców na danym obszarze,
- wypisywanie list obiektów dla danego obszaru,
- możliwość pogrupowania obiektów z tej listy,
- wyświetlenie pomocy informującej użytkownika o wszystkich funkcjach programu oraz o tym, jak należy ich używać,
- obsługa możliwych błędów oraz informowanie o ich rodzajach.

3 Opis interakcji użytkownika z programem

Każdy wie jak obsługiwać mikrofalówkę

4 Sytuacje wyjątkowe

Jednym z najważniejszych aspektów programu jest obsługa sytuacji wyjątkowych i błędów. Mogą być one związane z niepoprawnym podaniem danych przez użytkownika, podaniem danych nieobsługiwanych przez program lub próbą podjęcia niewłaściwych czynności. Poniżej znajduje się lista sytuacji wyjątkowych przewidzianych w tym projekcie, wraz ze sposobem, w jaki program reaguje na te zdarzenia.

1. Sytuacje związane z tekstowym plikiem wejściowym:
 - gdy użytkownik wskaże na nieistniejący plik, program wyświetli okno dialogowe z informacją o nieodnalezieniu żadanego pliku,

- gdy plik będzie pusty lub nie będzie złożony z czterech części oddzielonych liniami rozpoczynającymi się znakiem #, program wyświetli komunikat o niewłaściwej strukturze pliku wejściowego,
- gdy plik wejściowy będzie zawierać linię przechowującą zbyt mało argumentów, lub gdy jeden z tych argumentów będzie niepoprawnie podany (np. argument, który miał być liczbą będzie zawierać literę, lub zawiera ujemną wartość), program wypisze komunikat o błędzie odczytywania linii wraz z podaniem numeru danej linii,
- gdy jeden z parametrów x, y będzie wykraczać poza ustalony maksymalny rozmiar mapy, program wyświetli okno dialogowe z informacją przekroczeniu limitu rozmiaru w danej linii,
- gdy w deklaracji konturów wystąpią przynajmniej dwie przecinające się linie, program wypisze komunikat o przecinaniu się linii konturowych,
- gdy przy deklaracji punktów kluczowych przynajmniej jeden z nich znajduje się poza granicami wyznaczonymi wcześniej jako kontury planszy, program wyświetli okno dialogowe z informacją o niewłaściwym punkcie kluczowym wraz z podaniem numeru danej linii,
- gdy w deklaracji obiektów nie będą znajdować się trzy obowiązkowe typy obiektów – dom, szkoła oraz niedźwiedź, program wypisze komunikat o braku obowiązkowych deklaracji obiektów,
- gdy w linii zawierającej deklarację obiektu nie zostanie rozpoznany typ parametru obiektu, zostanie wyświetlone okno dialogowe o nierozpoznaniu typu parametru wraz z podaniem linii, w której wystąpił problem,
- gdy w ostatniej części pliku, zawierającej listę obiektów znajdzie się obiekt o niezadeklarowanym wcześniej typie, program wyświetli komunikat o nieznanym typie obiektu w danej linii.

2. tego jeszcze nie ma :)

- gdy nie uda się otworzyć pliku wejściowego, pojawi się okno dialogowe z komunikatem „*Could not open file!*”,
- gdy plik wejściowy okaże się pusty, pojawi się okno dialogowe z komunikatem „*Could not read an empty file!*”.

3. Sytuacje związane z niemożnością odnalezienia satysfakcjonującego wyniku programu:

- w przypadku gdy nie uda się znaleźć korzystnego sposobu wymiany waluty, w wynikowe pole tekstowe zostanie wpisane: „*Currency exchange: Could not find any beneficial exchange!*”,
- w przypadku gdy nie uda się odnaleźć arbitrażu, w wynikowe pole tekstowe zostanie wpisane „*Arbitrage finding: Could not find any arbitrage!*”.

5 Zarys testów akceptacyjnych

Test akceptacyjne zostaną przeprowadzone bez użycia zewnętrznych narzędzi. Ich celem będzie uruchomienie i sprawdzenie poprawności działania programu dla różnych danych wejściowych oraz różnych sposobów interakcji z programem.

Pierwsze testy akceptacyjne będą polegać na uruchomieniu programu z różnymi, poprawnie przygotowanymi zestawami danych. W szczególności przeprowadzone zostaną testy z zastosowaniem plików:

- z niewielką ilością punktów granicznych (mniej niż 5 elementów konturu),
- z dużą ilością punktów granicznych (więcej niż 20 elementów konturu),
- bez deklaracji dodatkowych typów obiektów,
- z deklaracjami dodatkowych typów obiektów,
- o minimalnym i maksymalnym rozmiarze planszy,
- graficznych o minimalnym i maksymalnym rozmiarze planszy.

Przeprowadzone zostaną także testy mające na celu sprawdzenie odporności programu na błędne dane wejściowe. W tej części program zostanie sprawdzony pod kątem plików:

- z błędnie zapisanymi danymi, omówionymi w sytuacjach wyjątkowych,
- zawierających kontury z przecinającymi się liniami,
- o rozmiarze planszy przekraczającym górny limit
- niezawierających żadnych punktów,
- próbujących umieścić punkty kluczowe lub obiekty poza granicami planszy,
- graficznych o zbyt dużym rozmiarze.

W trakcie trwania programu sprawdzone zostaną także próby wpisania niepoprawnych danych przy dodawaniu obiektów oraz prób ręcznego dodania obiektów i punktów kluczowych poza granicami planszy.