

GRAFIKA KOMPUTEROWA 1 PROJEKT LABORATORYJNY 1

SPECYFIKACJA

EDYTOR WIELOKĄTÓW

TYMON FELSKI, C1

1. Instrukcja obsługi programu

Dodawanie wielokąta

Wielokąty można dodawać klikając **LPM**. Na ekranie będą pojawiać się kolejne wierzchołki. Rysowanie zakończy się, jeżeli łamana zostanie zamknięta (postawimy ostatni wierzchołek dostatecznie blisko początkowego). Rysowanie można anulować wciskając **Escape**.

Przesuwanie wielokąta

Wielokąt można przesunąć wciskając **Shift + LPM** lub **ŚPM**, gdy kursor znajduje się w środku tego wielokąta. Jeżeli kilka wielokątów się pokrywa, o wyborze decydować będzie kolejność ich rysowania.

Przesuwanie wierzchołka

Wierzchołek w narysowanym wielokącie można przesunąć przytrzymując na nim **LPM**.

Usuwanie wierzchołka

Wierzchołek w narysowanym wielokącie można usunąć wciskając na nim **PPM** i wybierając opcję **Delete** w menu kontekstowym. Wierzchołki można usuwać, jeżeli ich liczba w wielokącie jest większa od 3, w przeciwnym wypadku zostanie wyświetlony komunikat o błędzie.

Dzielenie krawędzi

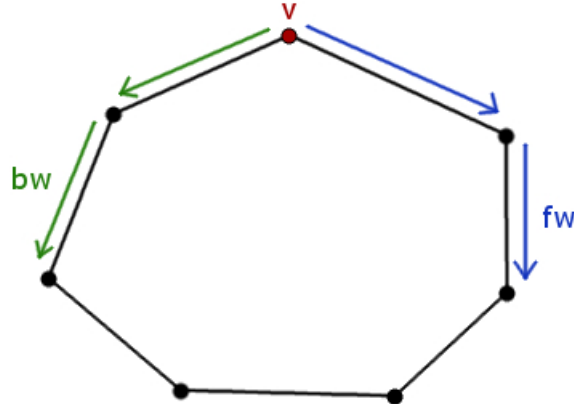
Krawędź w narysowanym wielokącie można podzielić na dwie, dodając w środku wierzchołek. W tym celu należy wcisnąć na niej **PPM** i wybrać opcję **Split** w menu kontekstowym. Ograniczenie podzielonej krawędzi zostanie wówczas usunięte (jeżeli było nadane).

Dodawanie ograniczeń krawędzi

Krawędzie mogą mieć nadawane ograniczenia. Aby to zrobić, należy wcisnąć **PPM** na konkretnej krawędzi i wybrać jedną z trzech opcji: **Horizontal** (aby zablokować krawędź w poziomie), **Vertical** (aby zablokować krawędź w pionie) lub **Length** (aby nadać krawędzi odpowiednią długość). Wybierając opcję **None** usuniemy istniejące ograniczenie. Jeżeli dodanie ograniczenia nie jest możliwe (krawędź jest zbyt pionowa/pozioma, a ustalenie długości wpłynęło by negatywnie na inne ograniczenia lub nie podano liczby z zakresu od 1 do 999), zostanie wyświetlony komunikat o błędzie.

2. Opis algorytmu nadawania ograniczeń dla krawędzi

Algorytm bazuje na pomysłach propagowania zmiany w danej krawędzi do następnej. Wówczas kolejna krawędź jest przesuwana w taki sposób, żeby zachowywała swoje ograniczenie i nie psuła ograniczenia poprzednich krawędzi. Aplikowanie ograniczeń odbywa się jednocześnie w dwie strony od wierzchołka, którym ruszamy (lub od wierzchołka początkowego krawędzi, jeżeli dodajemy jej ograniczenie). Obrazuje to rysunek poniżej:



Algorytm wykorzystuje w tym celu dwa iteratory: w przód (forward, ozn. **fw**) oraz w tył (backward, ozn. **bw**). Jest to możliwe dzięki zastosowaniu list dwukierunkowych na wierzchołki i krawędzie w implementacji klasy Polygon (reprezentującej wielokąt).

Iterator **fw** próbuje dopasować wierzchołki końcowe krawędzi do początkowych (stosuje ograniczenie “w przód”), natomiast iterator **bw** dopasowuje wierzchołki początkowe krawędzi do końcowych (stosuje ograniczenie “w tył”).

Aplikowanie ograniczeń krawędzi zostaje przerwane, jeżeli iteratory się spotkają lub będą na incydentnych krawędziach. Po wykonaniu pełnej pętli następuje sprawdzenie, czy wszystkie ograniczenia są zachowane. Jeżeli nie, zapamiętane na początku położenie wierzchołków zostaje przywrócone. W programie objawia się to “zablokowanym” wielokątem, tzn. danego ruchu nie można wykonać, bo według algorytmu spowodowałby zepsucie ograniczeń.

Pseudokod

1. zapamiętaj położenie wierzchołków w wielokącie
2. **fw** = krawędź, której wierzchołek początkowy == **v**
3. **bw** = krawędź, której wierzchołek końcowy == **v**
4. while (true)
5. zastosuj ograniczenie krawędzi w iteratorze **fw** w przód
6. zastosuj ograniczenie krawędzi w iteratorze **bw** w tył
7. break, jeżeli krawędź w **fw** == krawędź w **bw**
8. **fw** = **fw.Next** lub pierwszy element listy krawędzi,
 jeżeli doszliśmy do końca
9. break, jeżeli krawędź w **fw** == krawędź w **bw**
10. **bw** = **bw.Prev** lub ostatni element listy krawędzi,
 jeżeli doszliśmy do początku
11. sprawdź ograniczenia i przywróć położenie wierzchołków,
 jeżeli jakieś nie jest zachowane