Ograniczenia krawędzi pionowej, poziomej i o stałem długości reprezentują odpowiednio klasy: 'VerticalConstraint', 'HorizontalConstraint' i 'FixedLengthConstraint', każdy z obiektów tych klas ma referencję do krawędzi której ograniczenie dotyczy oraz każda krawędź ma referencję do ograniczenia jakie zostało na nią nałożone.

Analogicznie dla ograniczeń ciągłości - każda ograniczenie ma referencję do wierzchołka którego dotyczy oraz każdy wierzchołek ma referencję do nałożonego na niego ograniczenia.

Klasy reprezentujące ograniczenia posiadają metodę 'MaintainConstraint' której zadaniem jest sprawdzenie czy ograniczenie zostało zachowane po zmianie pozycji wierzchołków oraz ewentualne dokonanie poprawy

Każdy z wierzchołków ma dodatkowy parametr określający jego stopień swobody czyli informację dla metody 'MaintainConstraint' o tym w jakiej płaszczyźnie może zostać poruszony, na tej podstawie metoda ta może ustalić nowe pozycje wierzchołków.

Za zapewnienie spójności i integralności wielokąta w trakcie jego edycji odpowiedzialna jest klasa 'PolygonIntegrityGuard' która implementuje wzorzec obserwatora, i nasłuchuje na zdarzenia takie jak: zmiana pozycji wierzchołka, dodanie nowego wierzchołka, dodanie nowego ograniczenia, usunięcie wierzchołka, usunięcie krawędzi, dodanie segmentu beziera i inne. W odpowiedzi na te zdarzenia klasa poprawia wielokąt tak by ograniczenia zostały zachowane.

Przykład 1 - użytkownik poruszył wierzchołkiem wielokąta lub punkt kontrolny:

1. Wierzchołek/punkt kontrolny zgłasza zdarzenie 'PositionChanged'

2. Klasa 'PolygonIntegrityGuard' odbiera zdarzenie z informacją o tym który obiekt został poruszony i o jaki wektor

3. Przed podjęciem próby poprawy wielokąta tworzony jest backup z pozycjami wszystkich wierzchołków i punktów kontrolnych aby w przypadku niepowodzenia można było powrócić do stanu sprzed wykonania ruchu

4. Stopnie swobody we wszystkich wierzchołkach/punktach kontrolnych ustawiane są na najwyższe, oprócz wierzchołka/punktu kontrolnego który został poruszony przez użytkownika - jego nie możemy poruszyć

5. Jeśli poruszony obiekt jest wierzchołkiem:

5.1 Próbujemy poprawić wielokąt w lewą stronę wywołując metodę 'CorrectLeft'

5.2 'CorrectLeft' wywołuje metody 'MaintainConstraint' na na kolejnych krawędziach

5.3 'CorrectLeft' przechodzi po krawędziach w lewą stronę do czasu aż napotka na krawędź bez ograniczenia, wróci do początkowego wierzchołka lub nie uda się utrzymać ograniczenia

5.4 'CorrectLeft" zwraca:

0 - gdy napotkano na krawędź której nie można poprawić i zachować ograniczenia

1 - gdy napotkano na luźną krawędź

2 - gdy udało się przejść po całym wielokącie i zachować wszystkie ograniczenia

5.5 Jeśli 'CorrectLeft' zwróciła 0 lub jest konieczna jest jeszcze poprawa w prawą stronę, odpowiedzialna jest za to metoda 'CorrectRight' która działa dokładnie tak samo jak 'CorrectLeft' tylko przechodzi po wielokącie w przeciwną stronę.

5.6 Jeśli metodom 'CorrectLeft' i 'CorrectRight' nie udało się poprawić całego wielokąta konieczne jest przywrócenie backupu i poruszenie całego wielokąta

5.7 na koniec pozostało poprawić wszystkie ograniczenia na ciągłość – metoda 'CorrectContinuity' przechodzi po wszystkich wierzchołka i wywołuje metody 'MaintainConstraint' na wierzchołkach które te ograniczenia posiadają. Zauważmy że w przypadku gdy poruszony został wierzchołek zawsze jest możliwa poprawa ciągłości, bo punkty kontrolne będą swobodne i ich pozycje mogą zostać zmienione dowolnie bez konieczności poruszania wierzchołków wielokąta

6. Jeśli poruszony obiekt jest punktem kontrolnym:

6.1 w pierwszej kolejności musimy poprawić ciągłość w wierzchołku któremu towarzyszy poruszony punkt kontrolny oraz znaleźć wierzchołek wielokąta który został poruszony w wyniku operacji naprawy ciągłości. Odpowiedzialna za to jest metoda 'FindMoved' która znajduje wierzchołek w którym trzeba poprawić ciągłość, poprawią ją i zwraca poruszony wierzchołek wielokąta

6.2 Dalej przeskakujemy do punktu 5 i wykonujemy te same kroki.

Opis działania metod utrzymujących ograniczenia 'MaintainConstraint'

1. HorizontalConstraint:

1.1 dostajemy informację o tym który koniec krawędzi został poruszony we wcześniejszej iteracji pętli 'CorrectLeft' lub 'CorrectRight' (a) i znajdujemy drugi koniec (b)

1.2 Sprawdzamy czy ograniczenie jest zachowane, jeśli tak to ustawiamy wierzchołkowi b stopień swobody na X i wracamy

1.3 Sprawdzamy stopień swobody wierzchołka a, jeśli jest on ruchomy we współrzędnej Y to możemy go wyrównać w tej współrzędnej do wierzchołka b, tym samym nie ruszając wierzchołka b. Ustawiamy wierzchołkowi b stopień swobody na X i wracamy

1.4 Sprawdzamy czy możemy poruszyć b we współrzędnej Y, jeśli tak to wyrównujemy go do a, ustawiamy stopień swobody na X i wracamy

1.5 Jeśli nie udało się zachować ograniczenia zwracamy informację o niepowodzeniu

2. VerticalConstraint: analogicznie jak w HorizontalConstraint

3. FixedLenghtConstraint:

3.1 dostajemy informację o tym który koniec krawędzi został poruszony we wcześniejszej iteracji pętli 'CorrectLeft' lub 'CorrectRight' (a) i znajdujemy drugi koniec (b)

3.2 W pierwszej kolejności sprawdzamy czy krawędź która będzie sprawdzana w nestępnej iteracji ma również ograniczenie na stałą długość, jeśli tak to możemy spróbować ustawić wierzchołek b w takim punkcie aby ograniczenia dla obydwu krawędzi były spełnione

3.3 Sprawdzamy czy ograniczenie jest zachowane, jeśli tak to wracamy

3.4 Sprawdzamy czy a jest ruchomy w X, jeśli tak to sprawdzamy czy okrąg o środku w b

przecina współrzędną Y wierzchołka a, jeśli tak to przesuwamy a po współrzędnej X by znalazł się na okręgu i kończymy;

3.5 Analogicznie sprawdzamy ruchomość a w Y

3.6 Jeśli nie udało się poprawić wierzchołka a to musimy poprawić b, jeśli jest ruchomy we wszystkich współrzędnych to znajdujemy punkt p jak najmniej oddalony od b i znajdujący się na okręgu o środku w punkcie a. Umieszczamy b w punkcie p i wracamy

w każdym z przypadków przed powrotem ustawiamy stopnie swobody a i b na najniższe - nie możemy ich poruszać w przyszłości ani w X ani w Y.

4. G1ContinuityConstraint

4.1 bazując na wierzchołku którego ograniczenie dotyczy, znajdujemy towarzyszące mu punkty kontrolne lub punkt kontrolny i wierzchołek, oznaczamy je: v1 - wierzchołek którego dotyczy ograniczenie G1, v0 - wierzchołek/punkt kontrolny po lewej stronie v1, v2 - wierzchołek/punkt kontrolny po prawej stronie v0

4.2 Wybieramy wierzchołek/punkt kontrolny - v0 lub v2 którego pozycję zmienimy:

4.2.1 jeśli obydwa są punktami kontrolnymi, to wybieramy dowolny, który ma stopień swobody w X i Y

4.2.2 jeśli jeden z nich jest wierzchołkiem a drugi punktem kontrolnym to wybieramy punkt kontrolny, chyba że nie ma stopnia swobody w x i y

4.3 Obliczamy w jakim miejscu musi znaleźć się wierzchołek który wybraliśmy(powiedzmy v2) aby był współliniowy z v0 i v1 oraz jego odległość od v1 była taka sama jak przed dokonaniem poprawy.

4.4 Sprawdzamy czy v2 leży na krawędzi która jest ograniczona poziomo lub pionowo, jeśli tak to możemy ustawić stopnie swobody v1 i v2 tak aby można je było poprawić w przyszłości i wracamy

5 C1ContinuityConstraint:

Analogicznie jak G1, musimy tylko uwzględnić współczynnik odległości v0-v1 i v1-v2, oraz musimy ustawić stopnie swobody na najniższe, ponieważ relacja odległości musi zostać zachowana.