

Andrzej Fałkowski 119315  
Dawid Kwiatkowski 119374  
Paweł Pieniążek 119423

# Wirtualny rzeźbiarz cz.4

*Projekt realizowany w ramach przedmiotu Rzeczywistość Wirtualna*

## Założenia ogólne

Projekt polegał na stworzeniu aplikacji, umożliwiającej tworzenie rzeźb z wykorzystaniem ramienia pomiarowego urządzenia MicroScribe i eksport ich do formatu FBX. Ruchy końcówką wskaźnika są odwzorowane jako ruch narzędzia w wirtualnym świecie i powodują rzeźbienie. Użytkownik będzie miał do wyboru kilka narzędzi rzeźbiarskich (dłuta, noże i inne). W pracy wykorzystane zostały trzy sprzęty rzeczywistości wirtualnej: *MicroScribe* do rzeźbienia oraz *CyberStick2* i *SpaceNavigator* jako dwa alternatywne źródła nawigacji. Program został napisany w C# z wykorzystaniem technologii Microsoft XNA, oraz Windows Forms.

## Współpraca z urządzeniami rzeczywistości wirtualnej

**MicroScribe** - współpraca z urządzeniem MicroScribe okazała się problematyczna. Do komunikacji z komputerem wymagany był port COM, niedostępny na większości dostępnych komputerów laboratoryjnych, w związku z czym konieczne okazało się korzystanie z przejściówki na port USB (niedostępnej w zestawie). Problematyczna okazała się także instalacja sterowników urządzenia - na co najmniej jednym urządzeniu okazało się to niemożliwe. Samo operowanie z urządzenia ocenić można jako wysoce nieintuicyjne, ograniczenia ruchu w poszczególnych osiach obrotu ramienia uniemożliwiało częstokroć odwzorowanie pożądanego położenia narzędzia w programie. Brak jakiegokolwiek sprzężenia zwrotnego zapobiegał uzyskaniu wrażenia rzeźbienia w rzeczywistości wirtualnej. Jedyną zaletą urządzenia było stosunkowo bezoporowe operowanie wskaźnikiem. Mieszanie należy ocenić przeniesienie części interaktywnej wskaźnika na zewnętrzne pedały operowane nogą - z jednej strony pozwalało to na jednoczesne korzystanie z drugiego manipulatora, z drugiej - z punktu widzenia użytkownika bardziej efektywne byłoby przeniesienie przycisków na końcówkę wskaźnika MicroScribe pozwalająca na obsługę całego urządzenia za pomocą zaledwie jednej kończyny.

**CyberStick2** - komunikacja z urządzeniem przy pomocy DirectInput, wykorzystanie biblioteki SlimDX - DirectX dla C#. Urządzenie jest rozkalibrowane

**SpaceNavigator (urządzenie prywatne)** - alternatywne źródło nawigacji w przestrzeni (6 możliwości ruchu - 3 osie obrotu i 3 przesuwania), stabilne, wygodne w użyciu.

## Interfejs

### Opcje podstawowe

- klawiatura - skróty klawiszowe, wpisywanie wymiarów, nazw plików itp.
- mysz - wybieranie opcji (zapisywanie, wczytywanie obiektów, oraz cofanie i powtarzanie edycji), dobieranie parametrów suwakami (kął nachylenia, ostrosłupa imitującego dłuto)

### Sterowanie kamerą (nawigacja w świecie 3D):

- joystick - obrót w 2 osiach, plus w trzeciej na przycisk, przesuwanie grzybkim, plus przyciskami zoom
- manipulator - przesuwanie i obrót, każde w 3 osiach
- klawiatura - jw.

### Rzeźbienie (modelowanie):

- ramię pomiarowe - z urządzenia można odczytać pozycję czubka rysika, oraz wektor jednostkowy opisujący kierunek wskazywany przez rysik.
- mysz - rzeźbienie z perspektywy kamery. Wykorzystanie pozycji kamery do określenia pozycji narzędzia, jego kierunek wyznaczany jest na podstawie pozycji kamery oraz pozycji myszy

## Podstawowa funkcjonalność

### Narzędzia rzeźbiarskie:

- nóż (laser) - generuje idealnie proste cięcia, każdy ruch odcina fragment rzeźby, jeśli ruch nie spowodował odcięcia, to jest anulowany (nie da się wykroić sześciennego kształtu w sześciacie.
  - kliknięcie i przytrzymanie LPM - aproksymacja "punktów przegięcia" cięcia w miejscach, znaczącej zmiany kierunku - kolejne płaszczyzny cięcia, aktualnie płaszczyzny cięcia powstają co każde 20 pomiarów, niezależnie od kierunku.
- dłuto o modyfikowalnej wielkości i mocy, polega na stworzeniu obiektu będącego różnicą obiektu oryginalnego i ostrosłupa imitującego dłuto.

### Dodatkowo funkcjonalności:

- możliwość cofania edycji (ogromnie istotna funkcja odróżniająca rzeźbiarza wirtualnego od prawdziwego)
  - wszystkie modyfikacje trójkątów zapamiętywane jako usunięcie lub dodanie
  - tablica przechowująca zmiany podczas edycji, umożliwiająca cofanie wykonanych cięć oraz powtarzanie ich
  - wierzchołki nie są zapamiętywane, gdyż nie da się ich łatwo usuwać, ze względu na to, że trójkąty posiadają tylko indeksy wierzchołków
- zapis i odczyt pliku - format FBX, styl ASCII.
- Statystyka ilości wierzchołków, trójkątów oraz wielokątów rzeźby
- Rysowanie linii tnącej podczas cięć.
- Wykorzystanie ruchów nawigacyjnych do generowania rzeźbienia - rysik trzymany nieruchomo połączony z obrotem rzeźby w świecie również powoduje wycinanie

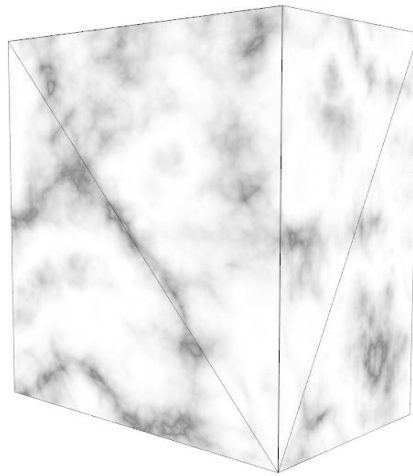
## Problemy geometryczno-matematyczne i ich rozwiązania

### Tekstura wolumetryczna 3D - szum Perlina

Kolor piksela zależy jest od jego pozycji w świecie 3D. Shader posiada funkcję, która na podstawie pozycji i wcześniej wygenerowanej tablicy z szumem, określa jego kolor.

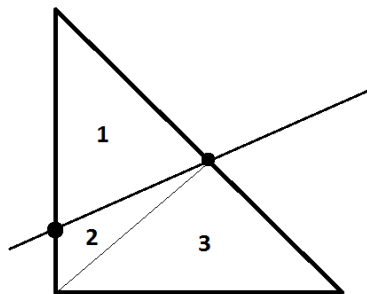
$$\sin(x + \text{sum}(1 / f(|\text{noise}|)))$$

Dzięki temu uzyskujemy imitację marmuru opisaną wzorem matematycznym. Ta wygenerowana tablica 256 liczb od 0 - 256 pozwala na zachowanie tej samej struktury marmuru przez cały etap rzeźbienia.



### Problem dokładności

Podczas cięcia należy znaleźć wszystkie trójkąty, które przecina płaszczyzna, potem wyznaczyć punkt przecięcia i wygenerować tam wierzchołki. Od razu tworzone są odpowiednie trójkąty (stary trójkąt dzielony jest na 3 nowe trójkąty):



- problem, który występuje w szczególnych przypadkach, czyli gdy cięcie przechodzi przez wierzchołek lub przez krawędź (praktycznie nieosiągalny ze względu na operowanie w dziedzinie liczb rzeczywistych), trójkąt początkowy jest wtedy dzielony na mniej trójkątów.

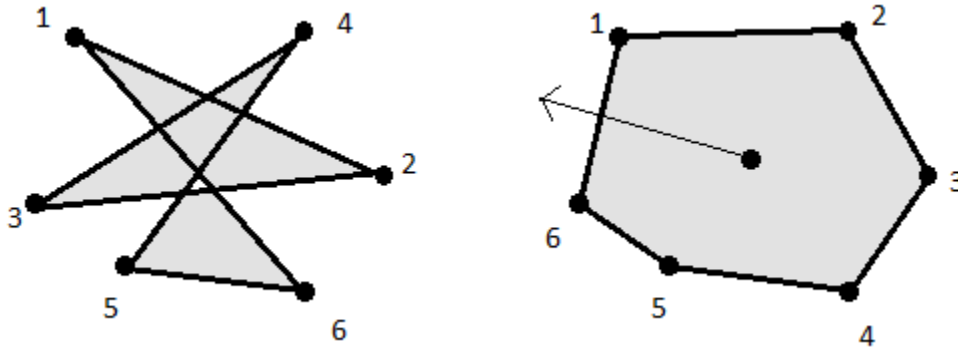
Dodatkowo zapisywane są miejsca przecięć dla danej krawędzi w celu uniknięcia

- problemów z dokładnością - punkt przecięcia P1 odcinka AB z płaszczyzną X, oraz punkt P2, jako punkt przecięcia odcinka BA z płaszczyzną X, mogą nieznacznie się różnić ze względu na niedokładność double
  - $1/3 * 2 = 0.666$
  - $2 * 1/3 = 0.667$
- problemu nadmiarowych wierzchołków - dzielenie 2 sąsiadujących trójkątów generowałoby 2 punkty zamiast 1.

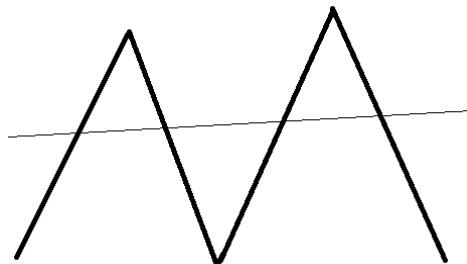
## Tworzenie nowych wielokątów

Po przecięciu bryły należy stworzyć wielokąt, lub wielokąty na płaszczyźnie cięcia.

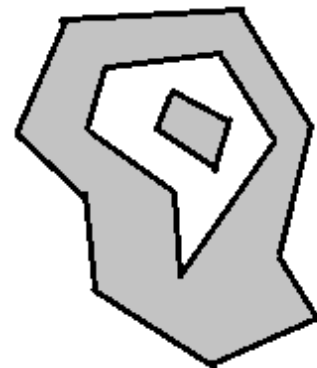
- wersja 1 - sortowanie wierzchołków w przestrzeni - wyznaczenie punktu centralnego, oraz kątów dla wierzchołków, wykorzystując ustalony wektor. Wszystko to odbywa się na płaszczyźnie cięcia - nie ma możliwości stworzenia dwóch wielokątów na wspólnej płaszczyźnie!



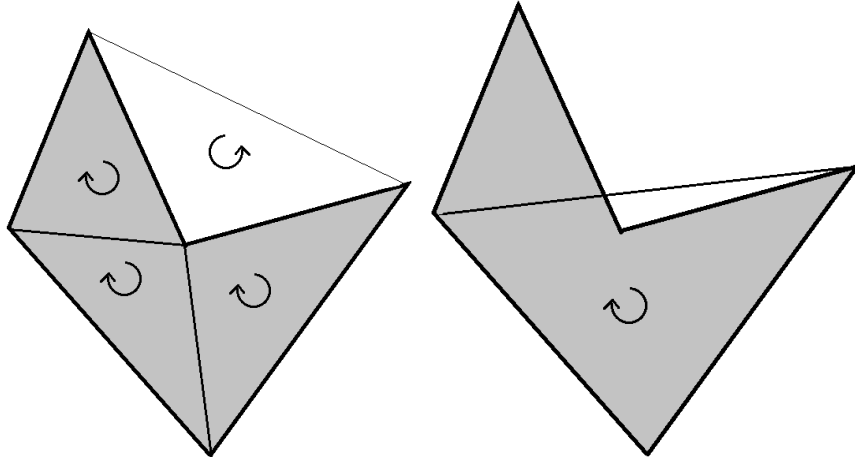
- wersja 2 - zapisywanie krawędzi dzielących trójkąt podczas cięcia i z nich tworzenie list krawędziowych (łamanych), które definiują wielokąty - w tej wersji nie występuje problem wielu wielokątów, ponieważ może powstać kilka list krawędziowych



- dodatkowo - należy sprawdzać, czy lista krawędziowa nie znajduje się przestrzennie wewnątrz innej listy, bo wtedy może być to wielokąt z dziurą.
  - powstaje problem gdy wielokąt jest z dziurą, ale ta dziura ma wyspę

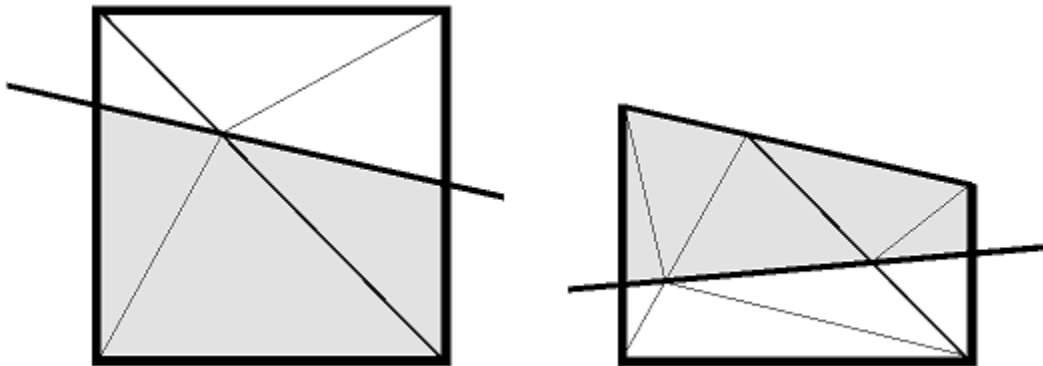


- Samo tworzenie wielokąta na podstawie listy krawędziowej nie jest proste. Dla wielokątów wypukłych można dodać punkt centralny wielokąta i każdą krawędź łączyć z tym punktem tworząc trójkąt, jednak problem pojawia się przy wielokątach wklęsłych. wielokąty wklęsłe
  - sprawdzanie kierunku skrętności trójkąta, ale istotne jest miejsce od którego się zacznie, trzeba zacząć od najmniejszych trójkątów, żeby rozwiązać ten problem został wykorzystany algorytm obcinania uszu (Ear-Clipping)

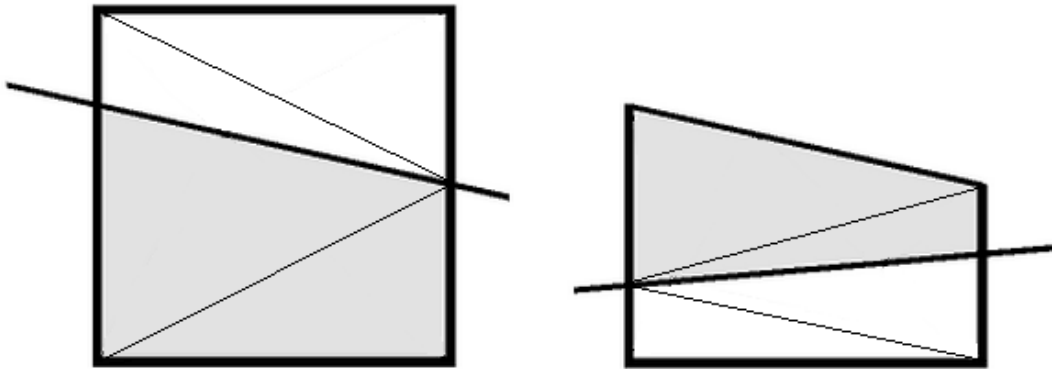


### Przechowywanie bryły w pamięci

Najprościej przechowywać trójkąty i wierzchołki. Każdy trójkąt to 3 indeksy na tablicę wierzchołków. Powoduje to pewien problem, z każdym cięciem tworzy się coraz więcej trójkątów.



Rozwiązaniem optymalizacyjnym jest reprezentacja bryły jako wielokąty (zaznaczenie krawędzi, które nie powodują zgięć (sąsiadujące trójkąty są na jednej płaszczyźnie) - przy cięciu nie generują nowych wierzchołków)



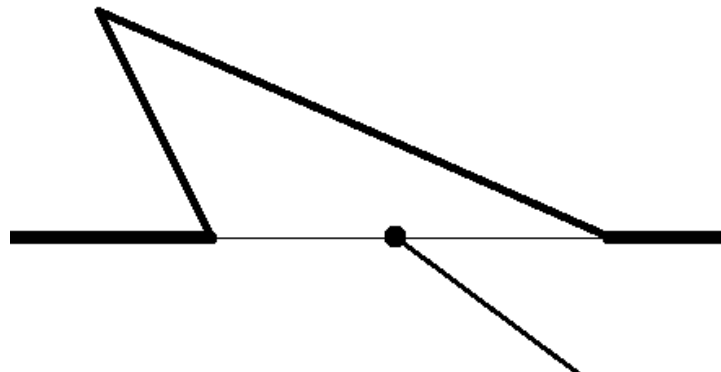
Zostało to rozwiązane w sposób następujący: cięcie przez krawędź “niewidoczną” (czyli tą wewnątrz wielokąta), powoduje stworzenie wirtualnego wierzchołka o numerze (-X), który używany jest podczas łączenia krawędzi w etapie tworzenia wielokąta, ale nie generuje on nowych wierzchołków. Dodatkowo oprócz krawędzi niewidocznych, przechowywana jest też informacja o tym do którego wielokąta należy trójkąt.

### Zapis do formatu FBX

Jest to bardzo zaawansowany format danych, który może przechować bardzo wiele informacji o obiekcie 3D. Wspiera on kości, tekstury, animacje, oświetlenie i wiele innych. W ramach projektu większość z tych rzeczy jest niepotrzebna, nie było więc konieczności tworzenia generatora kompletnych plików FBX. Rozwiązane to zostało w prosty sposób. Na początku został stworzony najprostszy możliwy plik FBX w programie graficznym. Na scenę był wstawiony sześciąt. Następnie został napisany parser, który umiał poruszać się po drzewiastej strukturze pliku i znajdował miejsca zapisu wierzchołków i trójkątów. W tym momencie działa już odczyt. Zapis polegał na podmianie danego fragmentu na nowy z bieżącymi danymi wierzchołków i trójkątów. Otworzyć można dowolny plik FBX. Zapis następuje do pliku sculpture.fbx w folderze z programem.

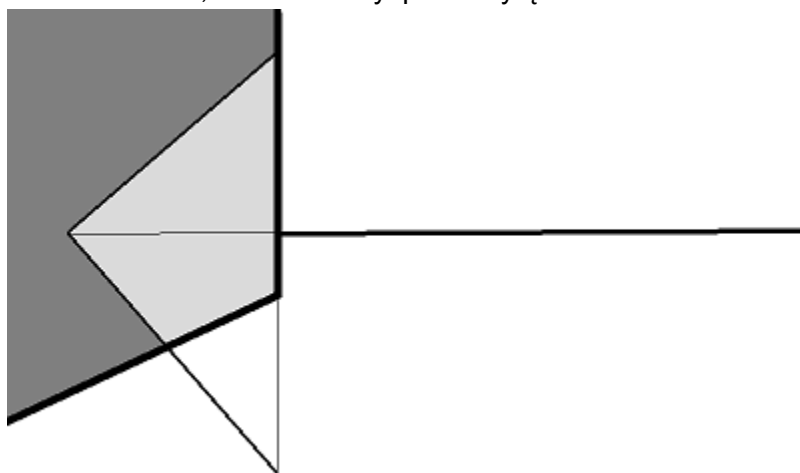
### Rzeźbienie dłutem z wykorzystaniem techniki śledzenia promieni

Rzeźbienie czyli symulowanie uderzeń dłuta, w rzeźbę, polega na wycięciu w bryle kształtu ostrosłupa (piramidki) o pozycji i kierunku zadanych przez użytkownika. Istnieje możliwość określenia kąta ostrości ostrosłupa.



Zasada działania - od punktu w nieskończoności do czubka dłuta, wzdłuż każdej krawędzi ostrosłupa rzeźbiącego rozchodzą się promienie. Znajdowane są trójkąty które przecinają daną półprostą, sortowane są po odległości od wierzchołka dłuta i tworzone są krawędzie pomiędzy kolejnymi wierzchołkami stworzonymi w punktach przecięcia. 1 i 2, 3 i 4, te krawędzie są później wykorzystane do stworzenia wielokątów głównych, generowanych przez rzeźbienie (wielokąty które są na płaszczyźnie trójkątów ostrosłupa). Tworzenie wielokątów na podstawie krawędzi zostało już opisane wcześniej.

- korzystamy tu z faktu, że w dalekim oddaleniu, na pewno promień będzie poza bryłą, każde nieparzyste przejście przez bryłę rozpoczyna krawędź, a parzyste ją kończy. Jeśli czubek ostrosłupa był parzysty, to na nim kończy się ostatnia krawędź. W przeciwnym wypadku wiadomo, że czubek był poza bryłą.



Technika promieniowa działa również, gdy uderzenia przechodzą na wylot rzeźby:



## Możliwości rozwoju

### Opcje dodatkowe:

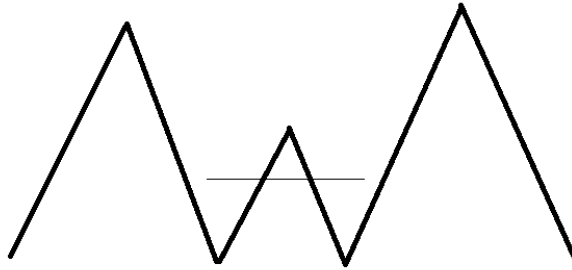
- okienko do wpisywania z klawiatury wymiarów początkowej bryły
- możliwość wybierania różnych brył początkowych o zadanych wymiarach: kuli, ostrosłupa, itp.

### Zapis:

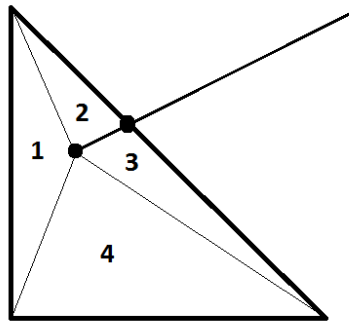
- zapisywane są tylko informacje o wierzchołkach i trójkątach, tracona jest informacja o wielokątach (trójkątach będących na jednakowej płaszczyźnie)

### Cięcie:

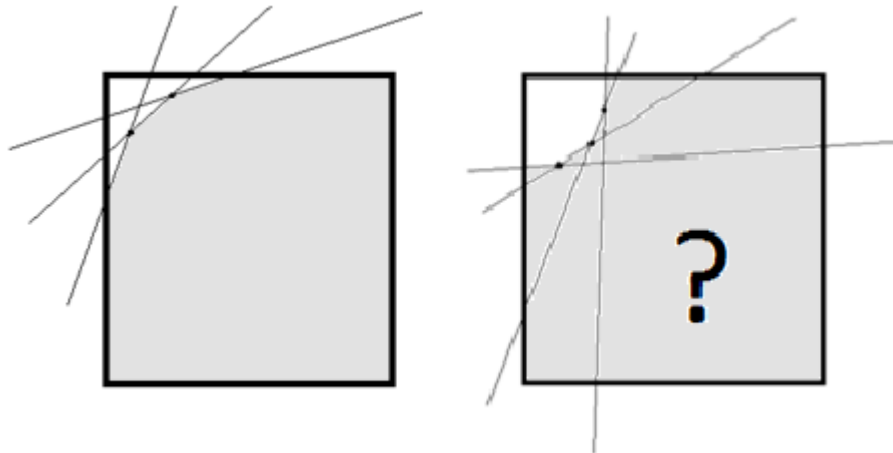
- prawdziwa aproksymacja “punktów przegięcia” cięcia w miejscach, znaczącej zmiany kierunku, a nie na sztywno co 20 pomiarów
- cięcie o ograniczonym zasięgu
  - równanie znajdujące krawędzie przecinające się z płaszczyzną zostaje uzupełnione o dodatkową nierówność informującą o zakresie



- umożliwienie wycinania brył wypukłych (problem opisany wcześniej)
  - dodanie dodatkowych wierzchołków w miejscach krańcowych płaszczyzn ograniczonych zakresem (dodanie również trójkątów)



- jeśli udałooby się rozwiązać poprzedni problem cięcia o ograniczonym zasięgu, to możnaby go tutaj wykorzystać





- robiąc cięcie, można by tworzyć odwrotny ostrosłup, co umożliwiłoby wycinanie fragmentów. Wierzchołek ostrosłupa znajdowałby się w miejscu kamery - rozwiązanie nadaje się tylko, gdy kamera się nie porusza, a sterowanie odbywa się przy pomocy myszy

#### **Rzeźbienie:**

- Różne kształty dłuta (ostrosłup o podstawie dowolnego wielokąta, możliwość definiowania własnych)
- Inne kształty, nie bazujące na ostrosłupie, np. "namiot", jako imitacja dłuta płaskiego

#### **Anty-rzeźbienie:**

- zasada podobna, tyle że zamiast wycinać kształt piramidy, można doklejać taki kształt

#### **Szlifowanie**

- wyznaczenie wierzchołka, który będzie poddawany obróbce
  - nowa pozycja wierzchołka, będzie obliczana na podstawie uśrednionej pozycji sąsiadów

### **Efekty działania programu - przykładowe rzeźby**

