

# Wykorzystanie efektów cząsteczkowych w grach

Maciej Stefańczyk

Wydział Elektroniki i Technik Informatycznych  
Politechnika Warszawska  
mstefanc@stud.elka.pw.edu.pl

**Streszczenie** Efekty cząsteczkowe są wykorzystywane do generowania najprzeróżniejszych efektów trudnych do zasymulowania przy pomocy innych technik. Niniejszy artykuł stara się w sposób krótki i treściwy zapoznać czytelnika z najważniejszymi metodami ich generowania i poprawy jakości uzyskanych rezultatów. Pokazuje także, jak wszechstronne i uniwersalne są systemy cząsteczkowe. Wszystkie metody poparte są przykładami ukazującymi zasadność ich stosowania. *abstract environment.*

**Key words:** Efekty cząsteczkowe, miękkie cząsteczki, megacząsteczki, symulacja płynów, generowanie ognia, chmury wolumetryczne

## 1 Podstawowe informacje

### 1.1 Motywacja i miejsca stosowania cząsteczek

Na początku kwestia najważniejsza – po co w ogóle cząsteczki i dlaczego właśnie one? Okazuje się, że jest cała gama efektów, które są trudne do zasymulowania i realistycznego wygenerowania (szczególnie w czasie rzeczywistym – a tak przecież działają gry) klasycznymi metodami. Efekty te to chociażby ogień, wybuchy, sierść czy inne, płynne, rozmyte bądź w jakikolwiek sposób niejednolite i losowe w swej naturze. Patrząc na kilka płonących pochodni spodziewamy się, że każdy płomień będzie zachowywał się odmiennie, a nie jest fizycznie możliwe przygotowanie animacji kilkunastu bądź kilkuset rodzajów płomieni pochodni – a to przecież tylko jeden z setek, bądź nawet tysięcy obiektów występujących w grze.

Drugą ważną kwestią jest miejsce stosowania efektów i systemów cząsteczkowych. Odpowiedź jest najprostsza, jaka tylko może być, mianowicie wszędzie. Praktycznie w każdej grze – od prostych gier platformowych, poprzez symulatory lotu, gry przygodowe, wyścigowe, sportowe możemy spotkać się z elementami generowanymi przy użyciu systemów cząsteczkowych. Weźmy na przykład kilka popularnych gier z różnych kategorii. Na początek *World of Warcraft* – gra komputerowa z gatunku MMORPG<sup>1</sup> wyprodukowana przez amerykańską firmę

<sup>1</sup> z ang. Massively Multiplayer Online Role Playing Game – rodzaj gier komputerowych RPG, w których duża liczba graczy może grać ze sobą w wirtualnym świecie; podobnie jak w innych rodzajach RPG gracz wciela się w postać i kieruje jej działaniami.

Blizzard Entertainment (rys. 1). W tej grze praktycznie w każdym momencie na ekranie widać efekty uzyskane przy pomocy cząsteczek. Wszystkie błyski spowodowane przez czary, aury unoszące się wokół bohaterów i potworów, ogień, dym, błyskawice to rezultat zastosowania opisywanych metod.



**Rysunek 1.** Zrzut ekranu z gry *World of Warcraft*

Drugim przykładem niech będzie którakolwiek z gier wyścigowych, np. *Need for Speed: Shift* – gra z serii wydawanej przez Electronic Arts (rys. 2). W grach tego typu realizm rogowki uzyskuje się poprzez dodanie efektów typu dymu spod kół, spalin bądź nawet ognia z rury wydechowej oraz śladów opon, które zostają na asfalcie. Wszystkie z nich – jak łatwo się domyślić – to systemy cząsteczkowe.

Nawet w prostych grach platformowych i zręcznościowych dla podniesienia przyjemności płynącej z gry (bądź czasami dla utrudnienia rozgrywki i rozproszenia uwagi) stosuje się dużo rozbłysków, wybuchów i innych tego typu efektów. Przykładem niech będzie gra *The Legends of Arkanoid* (rys. 3) – prosta gra polegająca na zbijaniu klocków, gdzie wykorzystane efekty widać w każdym momencie – podczas lotu piłeczki bądź po zbitiu klocka.

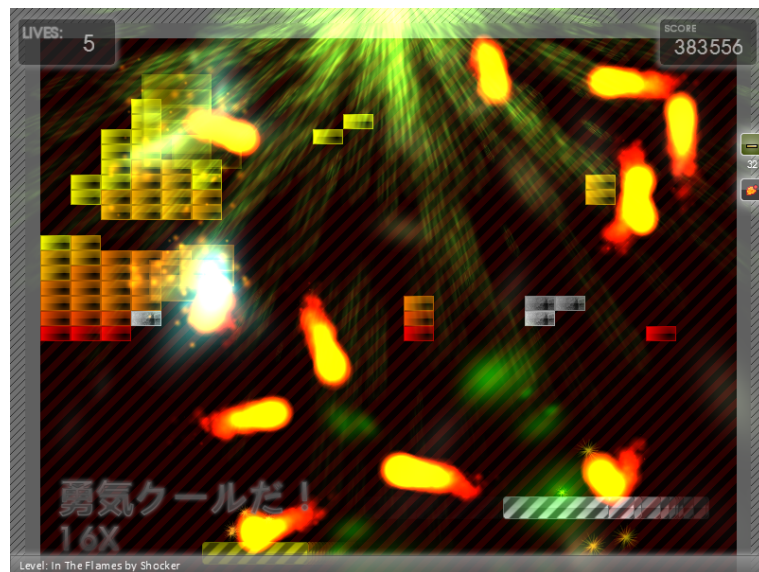
## 1.2 Definicje

Skoro już wiadomo gdzie, dlaczego i po co stosuje się efekty cząsteczkowe to czas wyjaśnić, czym i jak je generować.

Podstawowym elementem jest sama cząsteczka - pojedynczy obiekt odwzorowujący wygląd i zachowanie jednej bądź wielu cząstek obecnych w rzeczywistym



Rysunek 2. Zrzut ekranu z gry *Need for Speed: Shift*



Rysunek 3. Zrzut ekranu z gry *The Legend of Arkanoid*

efekcie (cząstki kurzu, krople wody, iskry). Ich charakterystykę można podzielić na kilka grup:

- własności fizyczne (pozycja, rozmiar, prędkość),
- własności graficzne (sposób reprezentacji na ekranie),
- zmienne związane z symulacją (wiek i czas życia cząstki)

Praktycznie wszystkie własności związane bezpośrednio z samą cząsteczką dotyczą tylko jej aktualnego stanu, w mniejszym stopniu związane są ze sposobem jej dalszego zachowania.

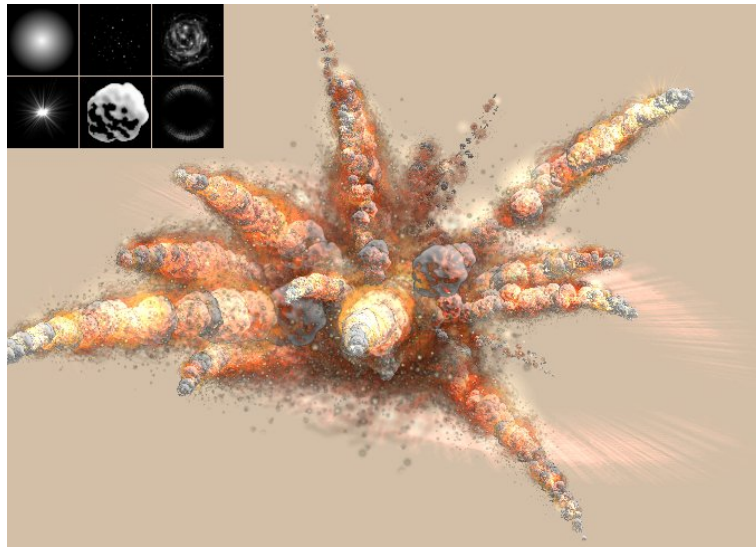
Informacje o całej grupie cząstek zawiera system cząsteczkowy. Jest to główny element zarządzający całym efektem, określający reguły generowania nowych cząsteczek (częstotliwość ich emisji, kierunek, wszystkie początkowe wartości), interakcje cząstek z elementami sceny, może zajmować się też symulacją fizyki dla samych cząsteczek. Wiele systemów ma też określoną maksymalną ilość jednocześnie wyświetlanych obiektów (ma to związek z wydajnością), a gdy tych obiektów robi się zbyt wiele najstarsze z nich są po prostu usuwane. Widać to wyraźnie na przykład w niektórych grach wyścigowych, gdzie po pewnym czasie ślady opon z asfaltu znikają. Najprościej można to zaobserwować kręcąc "bączki" w miejscu. Bardzo ważne jest więc odpowiednie dobranie wszystkich parametrów systemu w taki sposób, aby nie obniżać wydajności a jednocześnie nie doprowadzać do sytuacji takich jak wcześniej opisana, które mogą obniżać przyjemność z gry.

## 2 Przegląd technik tworzenia efektów cząsteczkowych

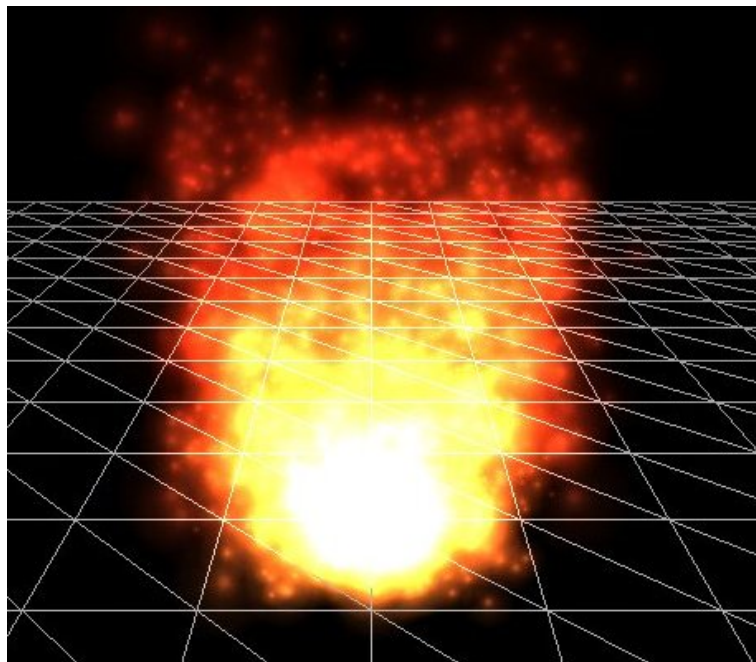
W tym rozdziale przybliżone zostanie kilka najpopularniejszych technik, wraz z ich zaletami i wadami, przykładami zastosowania w konkretnych produkcjach oraz możliwościami ich ulepszenia.

### 2.1 Najprostsze efekty

Najprostsze efekty można uzyskać już przy zastosowaniu samych kolorowych punktów (przy odpowiednich parametrach całego systemu można uzyskać dobre efekty np. Fajerwerków). Trochę bardziej złożone efekty mogą wymagać wykorzystania billboardów – otekstuiowanych czworokątów, które zwrócone są zawsze przodem w stronę obserwatora. Na rysunku 4 widać przykładowy efekt wybuchu oraz tekstury wykorzystane do jego stworzenia. W zależności od wieku cząstka przyjmuje odpowiednią teksturę oraz kolor, a po pewnym czasie znika. Przy pomocy praktycznie takich samych tekstur uzyskać można dość realistyczny efekt płomieni przedstawiony na rysunku 5. Zmianie ulegają parametry związane z kolorem, a także tryb mieszania kolorów – w przypadku przedstawionego ognia każda następna cząstka jest rysowana poprzez dodawanie swojego koloru do koloru piksela już narysowanego, przez co w środku płomienia kolor nasycza się aż do białego (efekt dość dobrze odwzorowuje rzeczywisty wygląd ognia).



**Rysunek 4.** Efekt wybuchu stworzony przy użyciu kilku tekstur



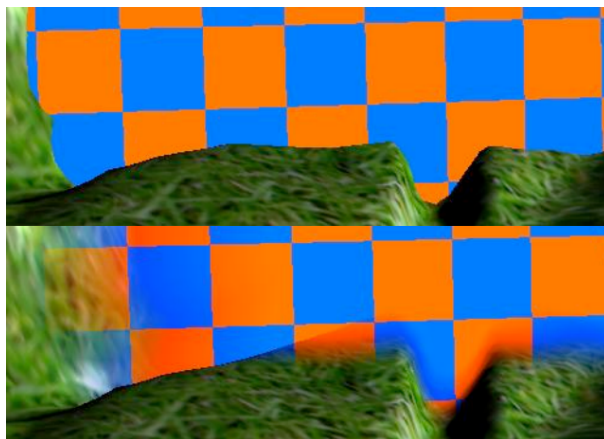
**Rysunek 5.** Ogień stworzony z tych samych tekstur co wybuch

Niewątpliwą zaletą przedstawionej metody jest prostota jej implementacji a także wykorzystania: wystarczy kilka tekstur i już przy pomocy samej manipulacji parametrami systemu możemy wygenerować całą gamę najróżniejszych efektów. Dodatkowo przy efektach składających się z małej liczby jednocześnie aktywnych cząstek (np. wybuchy) system taki jest bardzo wydajny – renderowane jest jedynie kilkanaście do kilkudziesięciu prostokątów.

Metoda ta ma jednak też szereg wad: przy większej liczbie cząstek wydajność zaczyna spadać – szczególnie jeśli chcemy zasymulować efekty takie jak kurz bądź dym (dla dobrego efektu dla skomplikowanej sceny należałoby generować wiele tysięcy cząstek). Dodatkowo sama dwuwymiarowa natura stosowanych billboardów powoduje kolejne problemy – niewłaściwe, ostre przecięcia cząstek ze sceną oraz brak prawidłowego oświetlenia (zarówno samych cząstek jak i rzucanych przez nie cieni).

## 2.2 Miękkie cząsteczki

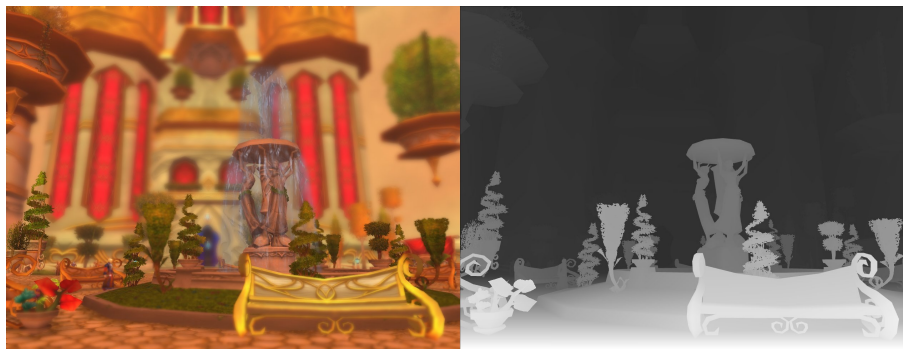
Miękkie cząsteczki są rozwiązaniem jednego z wymienionych problemów – z ich pomocą można w bardzo prosty i szybki sposób zmiękczyć krawędzie cząstek w miejscach przecięcia z obiektami już narysowanymi na scenie. W ich przypadku, tak samo jak wcześniej, same cząstki reprezentowane są przez billboardy, jednak podczas ich renderowania wykorzystywana jest dodatkowa informacja o głębi sceny w aktualnie rozpatrywanym punkcie. Jeśli obiekty są blisko cząstki to zwiększana jest jej przezroczystość, przez co płynnie wtapia się ona w otoczenie.



**Rysunek 6.** Dla uwypuklenia efektu działania miękkich cząsteczek pokazany jest nieprzezroczysty billboard przecinający się z terenem. Na górze efekt bez zmiękczenia, niżej widać płynne zwiększanie przezroczystości i miękkie wtopienie w otoczenie.



Krótkiego wyjaśnienia może wymagać jeszcze sama mapa głębi. Czym ona jest i jak powstaje? Jest to dodatkowy bufor, o wielkości równej rozmiarowi generowanego obrazu (czyli najczęściej rozdzielczości ekranu) przechowujący w każdym punkcie odległość od obserwatora do znajdującego się tam obiektu. Wypełniana jest w momencie rysowania obiektów, a więc jeśli chcemy ją wykorzystać do zmiękczenia cząsteczek, to musimy je rysować na końcu, już po narysowaniu innych obiektów.



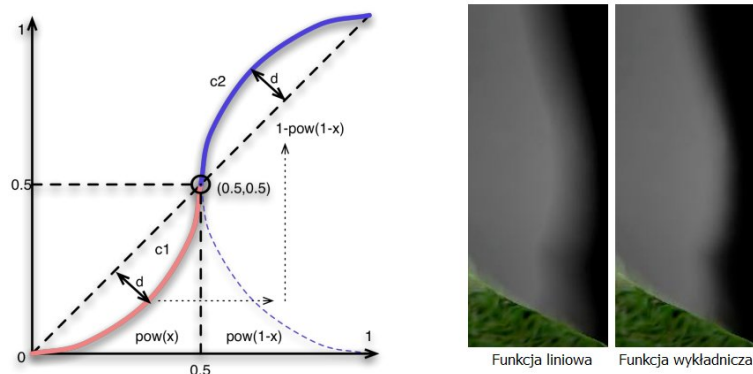
**Rysunek 7.** Wygenerowana scena wraz z jej mapą głębi. Im kolor jaśniejszy tym obiekt znajduje się bliżej obserwatora.

Z technicznego punktu widzenia do opisu metody generowania miękkich krawędzi wystarczy nam (oprócz mapy głębi) jedna funkcja opisująca szybkość zmian przezroczystości oraz pewna stała wartość oznaczająca graniczną odległość, powyżej której na przezroczystość nie będą miały wpływu obiekty znajdujące się na scenie. Wspomniana funkcja może być nawet najprostszej postaci, a więc liniowa, jednak wtedy efekt nie jest do końca satysfakcjonujący – w miejscu, gdzie cząstka znajduje się w odległości granicznej od innych obiektów występuje zauważalna zmiana jasności. Z tego powodu częściej stosuje się funkcje wykładnicze, bądź złożone z funkcji wykładniczych.

Chociaż opisywana metoda nie rozwiązuje wszystkich problemów (w szczególności cząstki nadal są dwuwymiarowe, więc nie ma mowy o efektach objętościowych), to technika ta jest często wykorzystywana ze względu na niewielki wymagany nakład sił w jej implementacji (jest to raczej rozszerzenie metody podstawowej niż coś całkowicie nowego) i dość dobre efekty.

### 2.3 Megacząsteczki

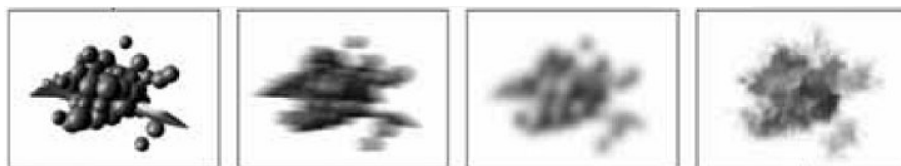
Drugą z opisywanych zaawansowanych metod generowania cząstek są megacząsteczki. Tutaj zmianie ulega już sposób reprezentacji cząstek oraz sam proces ich generowania. Najpierw na podstawie kilku bądź kilkunastu dużych brył określany jest ogólny kształt efektu oraz obliczane jest jego oświetlenie, a następnie na



**Rysunek 8.** Technika zmiękczenia cząstek. Po lewej: przykładowa funkcja zmiany przezroczystości w zależności od odległości. Po prawej: porównanie liniowej i wykładniczej funkcji wygaszania.

wygenerowanym obrazie dokonywane są przekształcenia określające jego ostateczny wygląd.

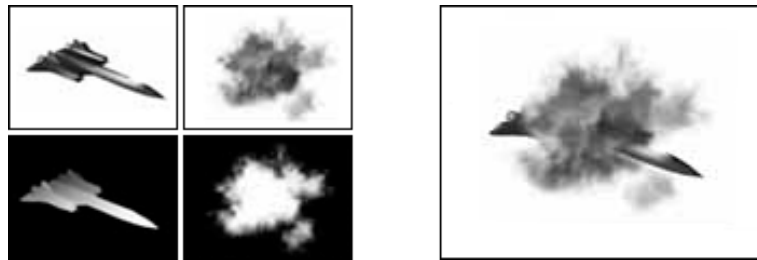
Rozdzielenie tych dwóch faz ma kolosalną zaletę: symulacja ruchu i rozprzestrzeniania wykonywana jest na bardzo małej liczbie obiektów, przez co jest bardzo szybka. Po wyrenderowaniu obrazu kul do osobnego bufora wykonywane jest na nim kilka różnych operacji, min. rozmycie, zaszumienie, przemieszczanie pikseli w różne strony. Dzięki temu w czasie niezależnym od ilości cząstek a jedynie od wielkości renderowanego obrazu otrzymujemy np. ładny i w pełni oświetlony dym. Następnie bufor ten jest mieszany z dotychczasową sceną przy wykorzystaniu informacji o głębi sceny i przezroczystości samego efektu. Widać to w uproszczeniu na przedstawionych obrazkach.



**Rysunek 9.** Technika generowania dymu przy pomocy megacząsteczek – określenie kształtu i postprocessing.

Megacząsteczki umożliwiają tworzenie efektów niemalże w pełni objętościowych – stworzony w ten sposób np. dym jest poprawnie oświetlony, a poruszanie się w jego wnętrzu nie sprawia wrażenia przebijania kolejnych warstw papieru (tak jak przy przechodzeniu przez dym złożony z billboardów). Niestety – jak wszystkie metody – i ta ma kilka wad. Po pierwsze jest to prawie pełne 3D





**Rysunek 10.** Mieszanie wygenerowanego dymu z docelową sceną.

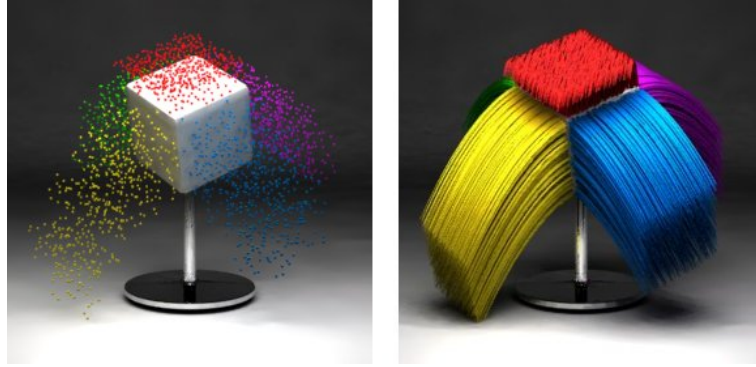
– a prawie robi wielką różnicę. Etap postprocessingu nadal wykonywany jest na obrazie dwuwymiarowym, przez co przy zbyt dużym rozciąganiu obrazu i skomplikowanych scenach może dojść mimo wszystko do nienaturalnych przecięć z innymi obiektami. Przy reprezentacji wielu cząstek przy pomocy jednego obiektu, takiego jak kula, trudno jest symulować zachowania fizyczne takie jak np. odbicia cząstek od podłoża, dlatego metoda ta sprawdza się głównie przy tworzeniu dymu bądź chmur, które nie wymagają takiej symulacji. Metoda ta wymaga także odpowiedniej kolejności generowania sceny – najpierw należy wyrenderować obiekty statyczne, żeby później móc wykorzystać informację o głębi do mieszania efektów. Z podobnych powodów trudne może okazać się też nakładanie kilku takich efektów w jednym miejscu.

## 2.4 Statyczne cząsteczki – włókna

W dalszej części omówione zostaną pokrótce jeszcze dwie metody stosowane przy tworzeniu efektów cząsteczkowych. Pierwszą z nich są statyczne cząsteczki. W tej metodzie zamiast śledzić samą cząstkę zapamiętujemy całą jej trajektorię, która ostatecznie tworzy potrzebny nam obiekt. Jest to metoda często zachłanna obliczeniowo, więc wykorzystywana jest raczej podczas przygotowywania zasobów do gier, generowania postaci bądź terenu, dużo rzadziej w czasie rzeczywistym.

Metoda stosowana może być np. do generowania realistycznych pól pełnych trawy (rozmieszczenie w sposób losowy, dodatkowo każde źdźbło może być wygenerowane losowo). Jest to metoda alternatywna do wykorzystywania fraktali, może też być stosowana równolegle z nią. Tak stworzona trawa może być jednak dość dużym obciążeniem dla systemu, w którym wyświetlić trzeba jeszcze setki innych obiektów, dlatego często statyczne cząstki wykorzystywane są do stworzenia grafik imitujących trawę, które już w dwuwymiarowej postaci wstawiane są na scenę.

Innym miejscem, gdzie zauważyć można miękkie cząsteczki może być chociażby sierść, a jedną z najbardziej znanych postaci z filmów i gier posiadających taką sierść jest niebieski stwór z produkcji *Potwory i spółka*.



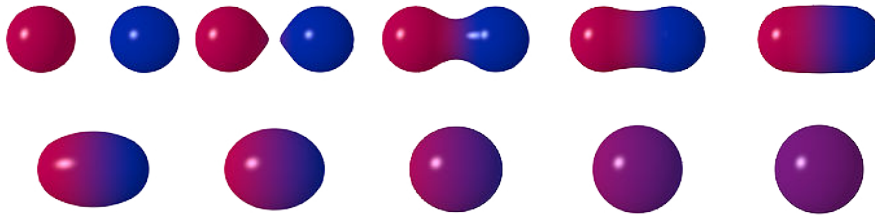
**Rysunek 11.** Po lewej stronie cząsteczki wygenerowane klasycznymi metodami. Po prawej te same obiekty emitowane przez system statycznych cząsteczek.

## 2.5 Metakule

Ostatnią opisywaną metodą jest reprezentacja graficzna cząstek przy pomocy metapowierzchni (głównie metakuli). Są to powierzchnie, zdefiniowane przy pomocy funkcji matematycznych, które płynnie się ze sobą łączą i mieszają, nadając przez to efektom bardziej organiczny wygląd. Każdą cząstkę opisujemy (w przypadku kuli) za pomocą współrzędnych jej środka, a cały efekt ma dodatkowo określoną pewną wartość progu, która określa podstawowy promień kuli.

Po zmianie stanu systemu i uaktualnieniu położenia wszystkich kul ze wzoru (1) wyliczane są punkty w przestrzeni, które są pokryte przez kule. Jeśli nierówność jest spełniona, to punkt jest uznawany za leżący wewnątrz kuli (bądź kilku połączonych).

$$\sum_{i=1}^n metakula_i(x, y, z) \leq p \quad (1)$$



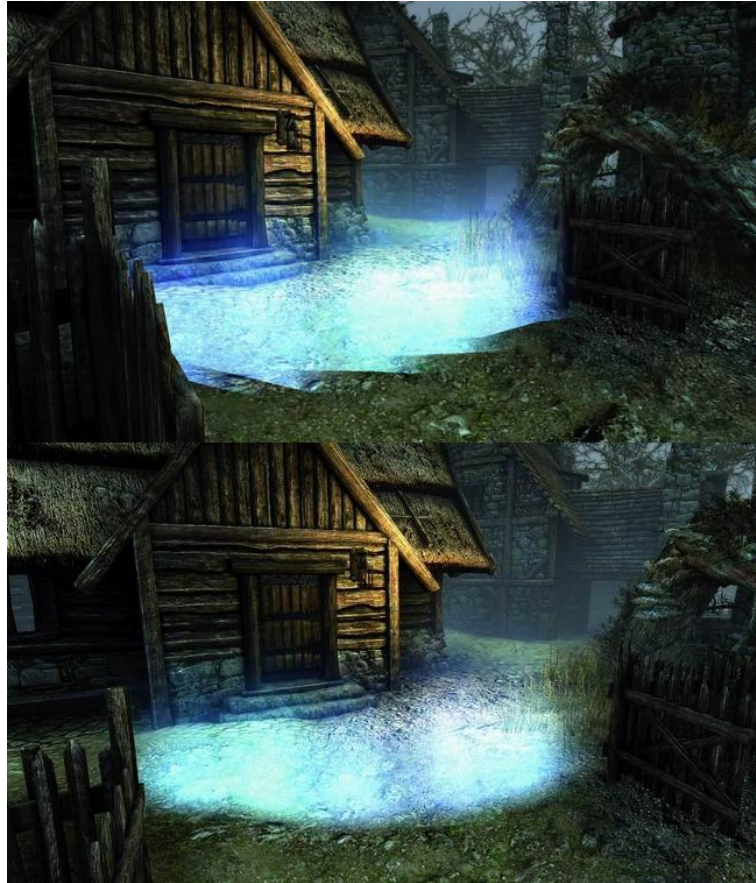
**Rysunek 12.** Poszczególne etapy łączenia się dwóch metakul. Wyraźnie widać ich przyciąganie się w momencie zbliżania. Kolor jest wyliczany na podstawie odległości danego punktu od środka kul. W ostatniej fazie wszystkie punkty są równo oddalone od obu środków (które się pokrywają), tak więc kolor jest średnią z kolorów wyjściowych.

Metakule wykorzystać można do tworzenia chociażby strumieni wody (takich jak fontanny) bądź nawet do generowania żywych stworzeń (niektóre ze stworzeń w grze Spore tworzone były poprzez generowanie odpowiedniej ilości metakulek, a ich kształt definiowany był poprzez zmianę parametrów systemu generującego). W ten sposób uzyskane efekty mają wszystkie cechy, których oczekujemy patrząc na różne żywe organizmy – każdy z nich jest niepowtarzalny (losowość systemu generującego), wszystkie wykazują pewne cechy wspólne (zasady generowania cząstek oraz ich trajektorii) a także wszystkie wyglądają w sposób zgodny z intuicją dotyczącą żywych organizmów (brak ostrych kątów i połączeń pomiędzy różnymi segmentami stworzenia).

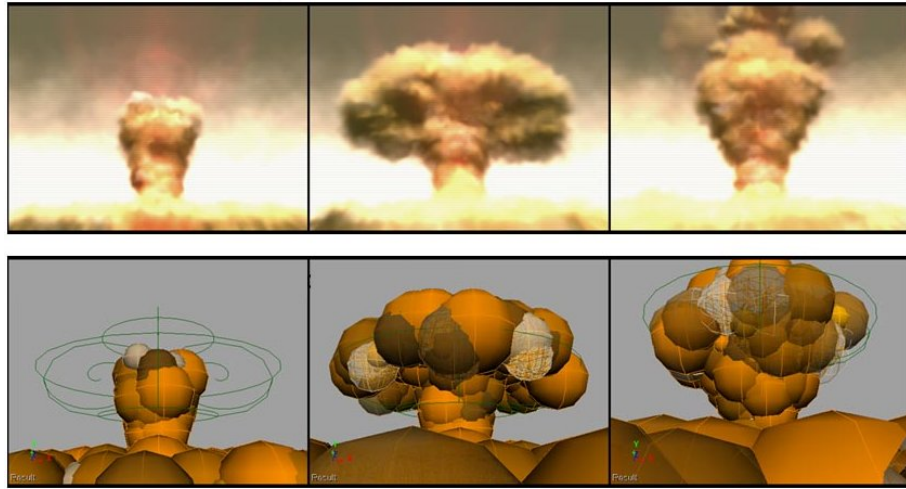
## Literatura

1. Particle System, Wikipedia, [http://en.wikipedia.org/wiki/Particle\\_system](http://en.wikipedia.org/wiki/Particle_system)
2. Lorach T.: Soft Particles, NVidia (2007)
3. Torque 3D homepage <http://www.torquepowered.com>
4. Bahnassi H., Bahnassi W.: Volumetric Clouds and Mega Particles, [http://www.inframez.com/events\\_volclouds\\_slide01.htm](http://www.inframez.com/events_volclouds_slide01.htm)
5. Metaball, Wikipedia, <http://pl.wikipedia.org/wiki/Metaball>

**Załącznik: Dodatkowe zrzuty ekranów z gier**



**Rysunek 13.** Miękkie cząsteczki: Na górze obraz z zastosowaniem klasycznej metody, na dole po włączeniu miękkich cząsteczek. Zrzut z gry *Two Worlds*.



**Rysunek 14.** Megacząsteczki: Dwa etapy działania systemu: na górze symulacja i oświetlenie na kilku dużych kulach, niżej ta sama scena po zastosowaniu postprocesingu.



**Rysunek 15.** Statyczne cząsteczki: Sierść jest jednym z obiektów, które można generować i symulować przy użyciu statycznych cząsteczek.



**Rysunek 16.** Metakule zastosowane do symulacji wody. Poszczególne krople łączą się ze sobą, tworząc jednolity strumień, odosobnione formują kule.