



Uczenie maszynowe w animacjach

Wykład nr 4

Techniki animacji wykorzystywane w grafice komputerowej

Szymon Datko

szymon.datko@pwr.edu.pl

Wydział Informatyki i Telekomunikacji,
Politechnika Wroclawska

semestr letni 2021/2022

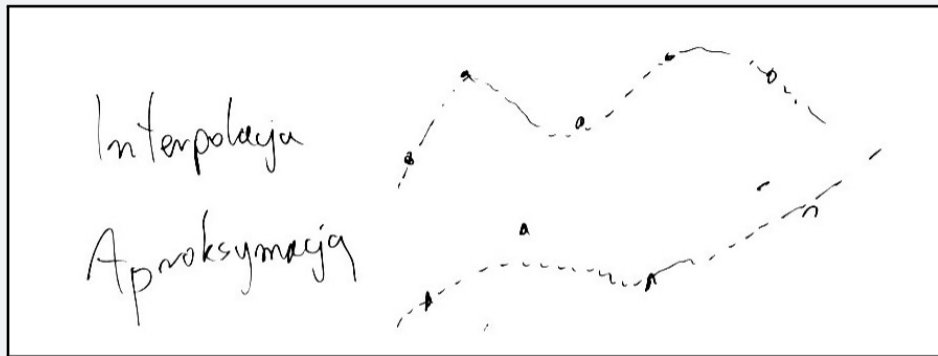
Co może być przedmiotem animacji?

- ▶ Zasadniczo animować możemy dowolną rzecz/cechę/wielkość,
 - w istocie jest to zmienianie wartości różnych jej parametrów,
 - najczęściej dziedziną tych zmian jest po prostu czas t .
- ▶ Zwykle mamy informację o początkowej i końcowej wartości,
 - ewentualnie także kilka kluczowych wartościach pomiędzy nimi,
 - w razie potrzeby możemy wyznaczyć dodatkowe wartości pośrednie, stosując procesy interpolacji lub aproksymacji,
 - odpowiednio mała różnica pomiędzy wartościami w kolejnych klatkach obrazu pozwala uzyskać wrażenie płynnego przejścia.

Klatki kluczowe


- Jedna z podstawowych technik animacji, mająca historyczne korzenie.
- Zbiór kluczowych informacji o elementach scenerii i ich ustawieniach.
 - na przykład ich położeniu początkowym i końcowym,
 - zmianach w strukturze, kolorze, składzie, itd.,
- Pomiędzy klatkami kluczowymi generowane później są **klatki pośrednie**.
 - w nich następuje interpolowane przejście między wartościami,
 - dokładny sposób tego przejścia można swobodnie dostosowywać.
- Dzięki temu nie musimy animować ruchu w każdej jednej klatce obrazu!

Interpolacja a aproksymacja



- ▶ Lokalna kontrola kształtu.
- ▶ Globalna kontrola kształtu.

Interpolacja liniowa



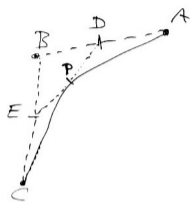
A diagram illustrating linear interpolation. It shows a line segment connecting two points, A and B. Point B is at the bottom left, and point A is at the top right. A point P is marked on the segment between A and B. An arrow points from B towards A, passing through P.

$$\vec{P} = \vec{A} + t * (\vec{B} - \vec{A})$$

$t \in [0; 1]$

W GLSL:
 $P = \text{mix}(A, B, t)$

Interpolacja kwadratowa



$$\begin{aligned} \vec{D} &= A + t * (\vec{B} - \vec{A}) \\ \vec{E} &= \vec{B} + t * (\vec{C} - \vec{B}) \\ \vec{P} &= \vec{D} + t * (\vec{E} - \vec{D}) \end{aligned}$$

$\leftrightarrow f(A, B, C, t)$

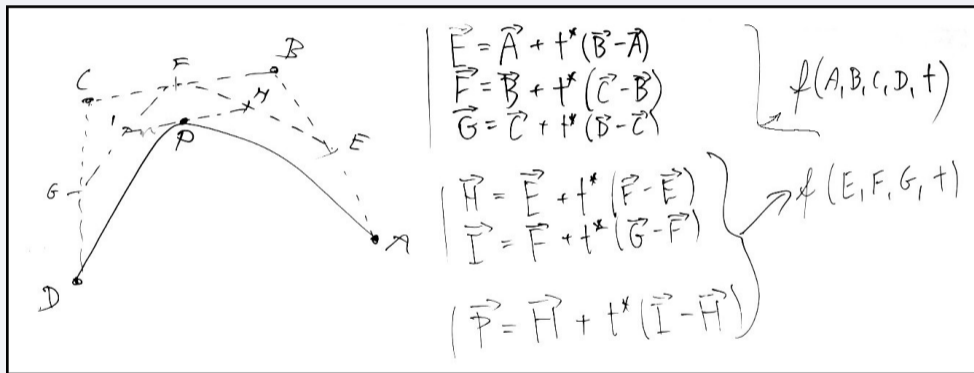
$$\vec{P} = \vec{A} + t * (\vec{B} - \vec{A}) + t * [\vec{B} + t * (\vec{C} - \vec{B})$$

$$- \vec{A} - t * (\vec{B} - \vec{A})]$$

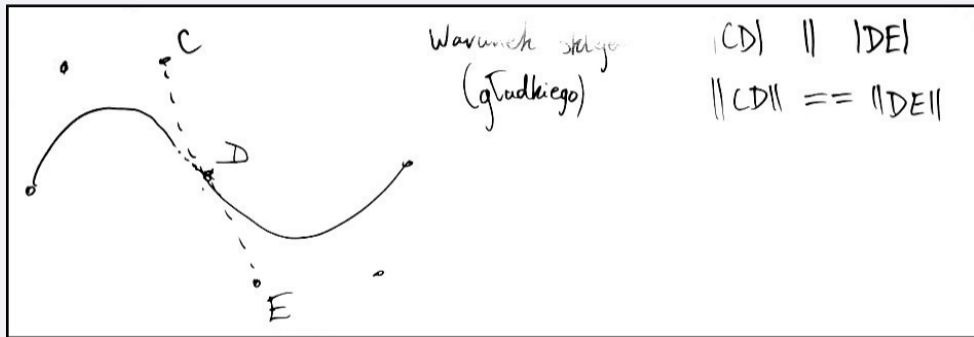
$$\vec{P} = \vec{A} + t * \vec{B} - t * \vec{A} + t * \vec{B} + t^2 * \vec{C} - t^2 * \vec{B} - t * \vec{A}$$

$$\vec{P} = \vec{A} + 2t * (\vec{B} - \vec{A}) + t^2 * (\vec{C} - 2\vec{B} + \vec{A})$$

Interpolacja wyższych rzędów



Tworzenie krzywej skleianej



Dynamika ciał w animacjach

▶ Ciała sztywne, ang. *rigid body*:

- nie uwzględnia się odkształceń na skutek działania sił,
- stosunkowo proste i mało kosztowne obliczeniowo,
- liniowe przemieszczenie wszystkich wierzchołków i ich obroty,
- wystarczające przybliżenie w wielu przypadkach.

▶ Ciała miękkie, ang. *soft body*:

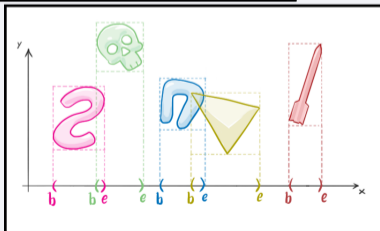
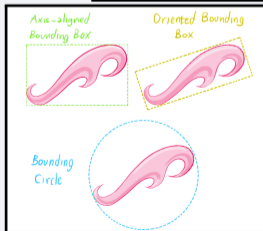
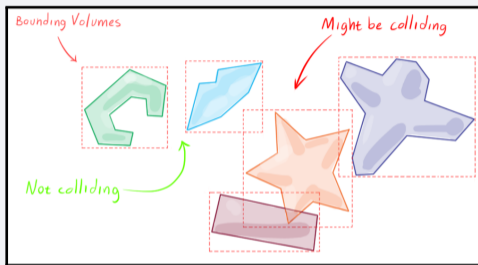
- dodatkowo uwzględnia się odkształcenia obiektu,
- zmiana rozkładu masy – wzajemnego położenia wierzchołków,
- specyficzny przypadek – obiekty nieściśliwe: **tkaniny** (ang. *cloth*).

A Cloth Simulation essentially models the object as a bunch of balls (vertices) connected to each other by fixed-length rods. Thus, for computational purposes, the simulation can treat the internal distances between points as a constant, and only has to calculate interaction with other objects.

A Soft Body Simulation, on the other hand, models the object as a bunch of balls (vertices) connected to each other by springs. So now, not only does Blender have to compute the interaction of the object with other objects, but also has to compute its own internal reaction and the spread of force applied at one point across all of those little springs, which is much more complex.

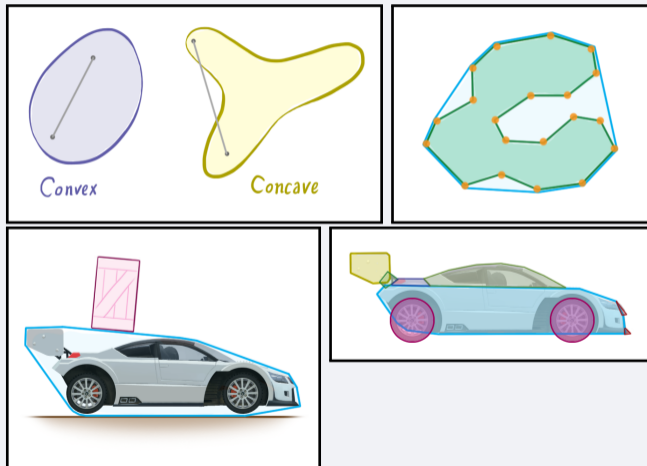
Kolizje obiektów

- Faza zgrubna – ograniczenie przestrzeni/obiektów do rozpatrzenia.



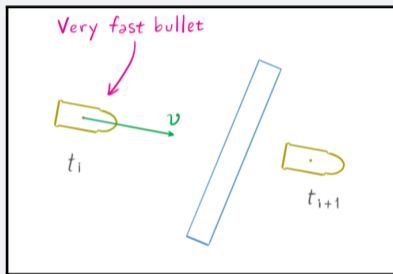
Problem obiektów niewypukłych

- ▶ Algorytm quickhull i inne oraz podział obiektu na mniejsze segmenty.



Problem szybko poruszających się obiektów

- ▶ Zagęścić liczbę kroków czasowych w których sprawdzamy kolizje.
- ▶ Analizować trajektorię przebyte przez obiekty w pobliżu.



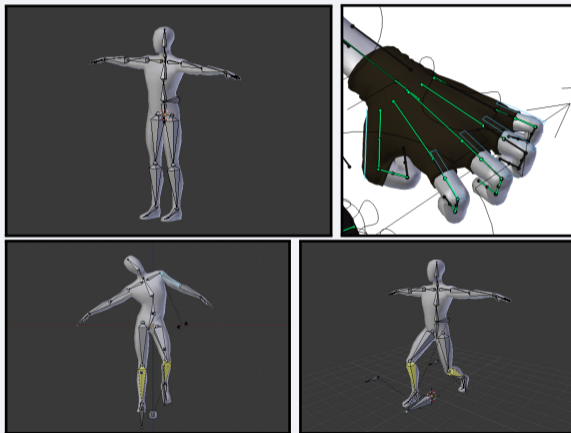
- Przykład na filmie: https://youtu.be/ogWQs_7DU0Y?t=842 od 14:02.

Uwaga! Podane w pierwszej chwili rozwiązanie problemu jest błędne. Później w materiale Pan to koryguje.

- Jeszcze inne omówienie: <https://blender.stackexchange.com/questions/146210/active-rigid-body-objects-fall-through-other-passive-rigid-body-objects>.

Modele szkieletowe

- ▶ Znane także jako angielskie **rigging** (dosłownie: olinowanie statku).
- ▶ Łączenie hierarchiczne elementów i ograniczenia w ruch poszczególnych.

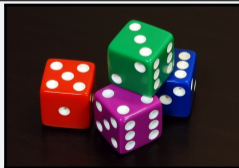
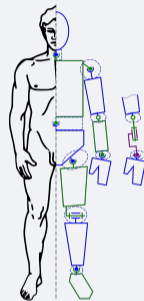
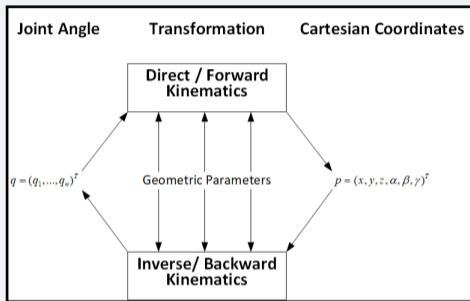


Źródło: https://en.wikipedia.org/wiki/Skeletal_animation

<https://cgi.tutsplus.com/tutorials/building-a-basic-low-poly-character-rig-in-blender--cg-16955>

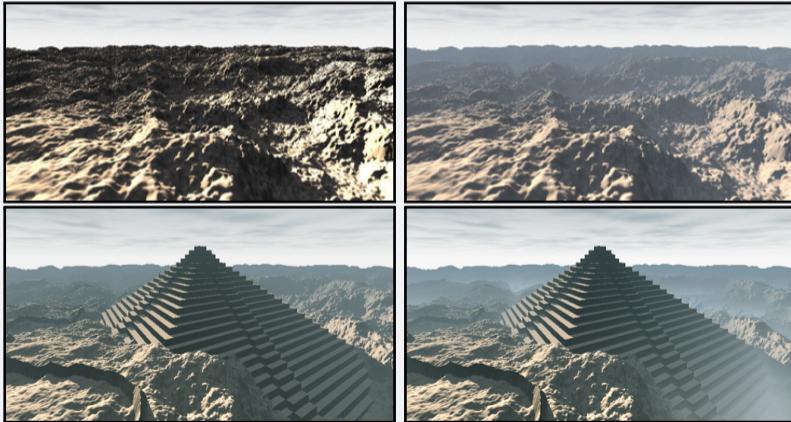
Kinematyka odwrotna

- ▶ Znamy cel i ograniczenia ruchu, poszukujemy odpowiedniej ścieżki.



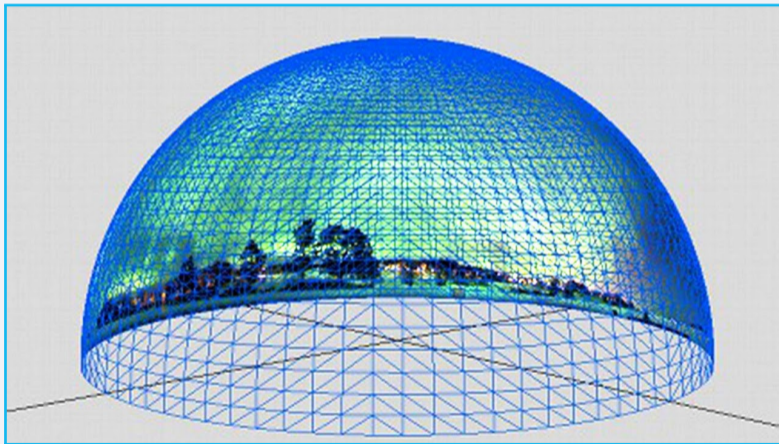
Mgła

- ▶ Rozpraszanie światła na cząsteczkach – im dalej od kamery, tym mocniej.
- ▶ W naturze zjawisko to częściej występuje w zagłębieniach terenu.



Skybox / Skydome

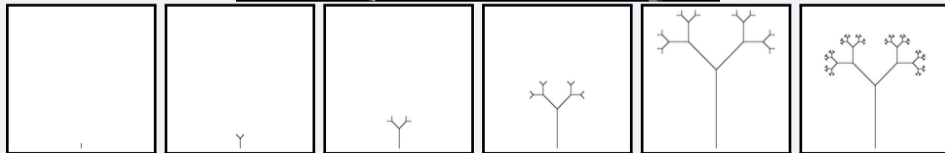
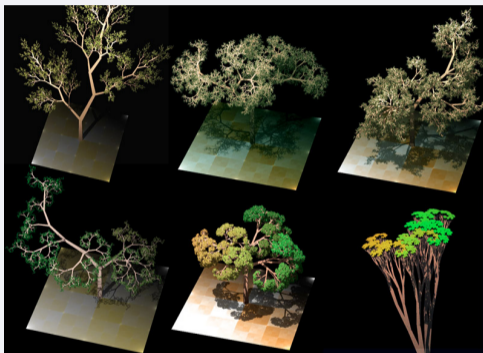
"Jaki kształt ma niebo w grze wideo?" – TVGRYpl, 2018.



<https://www.youtube.com/watch?v=-e9BIzXsL0E>

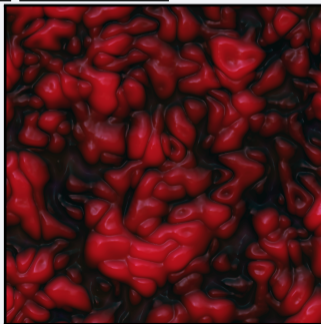
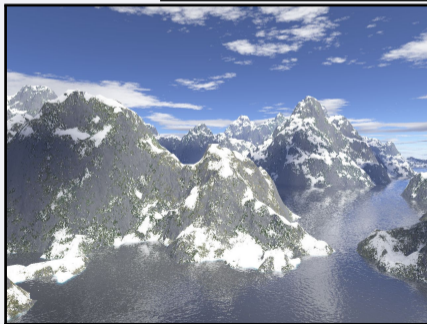
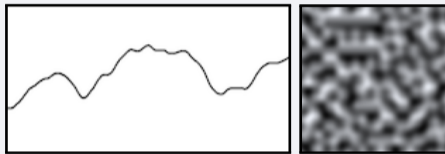
Budowanie roślinności

- ▶ L-systemy / Systemy Lindenmayera – uzyskujemy twory fraktalne.



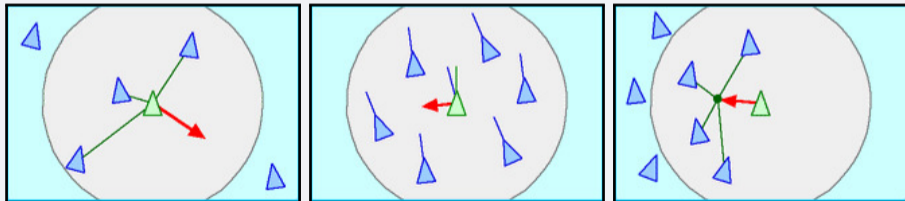
Proceduralne modelowanie terenu

- ▶ Wykorzystuje się funkcje szumu, np. szum Perlina.



Zachowanie zwierząt

- ▶ Algorytm stada – boids / flocking simulation – Craig Reynolds:
 - Rozdzielność.
 - Spójność.
 - Wyrównywanie.



Uzupełnienie informacji

- ▶ „The Nature of Code” – Daniel Shiffman,

– <https://natureofcode.com/book/>.

THE NATURE OF CODE

DANIEL SHIFFMAN

- WELCOME
- ACKNOWLEDGMENTS
- DEDICATION
- PREFACE
- INTRODUCTION
- 1. VECTORS
- 2. FORCES
- 3. OSCILLATION
- 4. PARTICLE SYSTEMS
- 5. PHYSICS LIBRARIES
- 6. AUTONOMOUS AGENTS
- 7. CELLULAR AUTOMATA
- 8. FRACTALS
- 9. THE EVOLUTION OF CODE
- 10. NEURAL NETWORKS
- FURTHER READING
- INDEX

Welcome

Hello! By browsing the table of contents on your left, you can read the entire text of this book online for free, licensed under a [Creative Commons Attribution-NonCommercial 3.0 Unported License](#).

[Start reading the introduction now!](#)

If you like this book, please consider supporting it via the links below:



[Buy from CreateSpace](#) | [Amazon.com](#) | [Buy the e-book bundle](#)

Please submit corrections to the book [on my Nature of Code GitHub repo](#). Bug reports for the site and online purchasing system can be reported on [GitHub](#) as well.

Kilka słów o animowaniu ruchów ludzi

"Jak twórcy gier oszukują w animacjach" – TVGRYpl, 2018.



<https://www.youtube.com/watch?v=0Sf3WGQUQsA>

Subtelna złożoność codziennych czynności

"Animacje, których twórcy gier wolą unikać" – TVGRYpl, 2019.



<https://www.youtube.com/watch?v=ctk6eCS3IQc>

To wszystko na dziś.

Do zobaczenia!