

# Animacja szkieletowa z wykorzystaniem kinematyki odwrotnej w silniku Unity

Łukasz Biańczak

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Techniki implementacji kinematyki odwrotnej</b>	<b>2</b>

## 1 Wstęp

\*Wskazówki dla mnie\*

- Omówienie podstaw animacji szkieletowej w grach komputerowych
- Omówienie wybranych metod kinematyki odwrotnej
- Zapoznanie się z silnikiem Unity, w szczególności komponentami odpowiedzialnymi za implementację animacji szkieletowej oraz kinematykę odwrotną
- Implementacja aplikacji demonstracyjnej w silniku Unity, wykorzystującej kinematykę odwrotną
- Porównanie animacji opartej o kinematykę odwrotną z animacją realizowaną w trybie wypalonym

### Opis tematu

W klasycznej animacji szkieletowej kości reprezentowane są za pomocą pewnej struktury drzewiastej obiektów. Wykonywanie sekwencji animacji polega zwykle na aktualizacji parametrów tych obiektów (położenia i orientacji kości) w porządku od korzenia do liści. W przypadku kinematyki odwrotnej, jak sama nazwa wskazuje, porządek aktualizacji jest odwrotny tj. od danego liścia do określonego węzła

nadrzędnego. Metody te pozwalają na proceduralne tworzenie sekwencji animacji umożliwiających wierniejsze odzwierciedlenie interakcji animowanego obiektu z otoczeniem bez konieczności ręcznego "wypalania" sekwencji animacji (ang. baked animation). Na przykład: podnoszenie różnych przedmiotów, obsługa urządzeń typu dźwignie, koła, dopasowanie ustawienia kończyn postaci do nierówności terenu itp. Znalazienie, właściwych wartości parametrów kości dla sekwencji animacji generowanych na podstawie kinematyki odwrotnej nie zawsze jest zadaniem prostym i czasem wymaga użycia zaawansowanych metod optymalizacyjnych. Celem pracy jest zapoznanie się z podstawowymi algorytmami wykorzystywanymi w kinematyce odwrotnej, jak również funkcjonalnościami oferowanymi przez silnik Unity w zakresie ww. technik.

## 2 Techniki implementacji kinematyki odwrotnej

Istnieje wiele algorytmów służących do implementacji kinematyki odwrotnej różniące się zastosowaniem i złożonością. Można wyróżnić prostsze implementacje, działające tylko na hierarchiach kości składających się z kilku obiektów. Istnieją też algorytmy, które pozwalają na wykorzystanie dowolnej długości łańcuchów, ale są bardziej złożone matematycznie. Często jest wtedy stosowane podejście heurystyczne, obliczające jedynie aproksymacje właściwego umiejscowienia kości w przestrzeni trójwymiarowej.

$$\sum_{n=1}^{\infty} 2^n$$