

Podążanie za obiektem w przestrzeni 2D (sheadovas/poradniki/goto/podazanie-za-obiektem-w-przestrzeni-2d/)

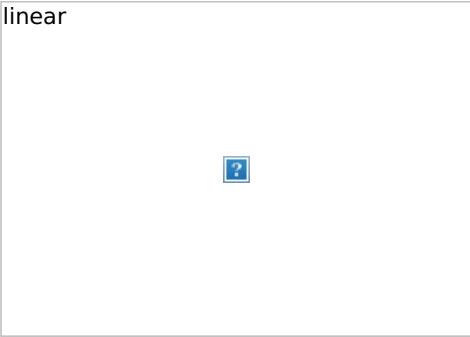
Wrz 27, 2015 / goto (sheadovas/category/poradniki/goto/)

Podążanie za obiektem (np. kursorem), od punktu do punktu, inaczej mówiąc: interpolacja liniowa.

Ostatnio zauważyłem, że wiele osób ma problem z dość prostym do rozwiązania problemem: ruchu obiektu do określonego punktu (np. do kursora myszy), całość może też być wykorzystana w np. podążaniu za obiektem. W tym krótkim wpisie pokażę Wam jak takie coś wykonać na przykładzie programu w SFML, gdzie koło podąża za kursorem myszy (demo jest na samym końcu wpisu)

Trochę matematyki...

Krótki wstęp teoretyczny (mam nadzieję, że tak prostych rzeczy nie muszę specjalnie długo tłumaczyć).
Niewiele osób zdaje sobie sprawę, że płynne poruszanie obiektu z punktu do punktu to nic innego jak wektor, dokładnie ten sam, który poznaje się w liceum.



(<https://i2.wp.com/www.shead.ayz.pl/wp-content/uploads/2015/09/linear.png>)

Co zrobić aby uzyskać wektor przemieszczenia od punktu *A* do punktu *B*? Dokładnie! Musimy od współrzędnych punktu *B* odjąć współrzędne punktu *A*.

```
1 | w = [B.x - A.x, B.y- A.y]
```

Aby uzyskać płynne poruszanie się od punktu *A* do punktu *B*, to należy przemieścić obiekt o *w**delta_time (delta_time to przyrost klatek na sekundę). Jednak w tej postaci przesunięcie nie będzie stałe tzn. im mniejsza odległość od celu, tym mniejsza prędkość.

Jeżeli zależy nam na zlikwidowaniu tego zjawiska to musimy otrzymany wektor znormalizować, tzn sprawić że jego wartości x/y będą znajdowały się w przedziale [-1,1]. Normalizacja odbywa się przez podzielenie składowych wektora przez jego początkową długość.

```
1 | v = w / |w|
```

Demo

Demo napisane w SFML przedstawiające jak napisać kod, w którym obiekt porusza się w stronę kursora.

C++

```
1  /*
2     Demo przedstawiające poruszanie obiektu za kursorem.
3
4     Author: Szymon (sheadovas) Siarkiewicz
5     Website: sheadovas/
6
7     License: CC0 (public domain)
8  */
9
10 #include <SFMLGraphics.hpp>
11 #include <cmath>
12
13
14 /*
15     Interpolacja liniowa
16
17     from    pozycja od której ma zostać wykonane przesunięcie
18     to      pozycja do której ma zostać przesunięty obiekt
19     t       opcjonalny argument dodatkowo zmieniający wartość wektora
20
21     Funkcja zwraca Vector o jaki powinien zostać przesunięty obiekt
22  */
23 sf::Vector2f lerp(sf::Vector2f from, sf::Vector2f to, float t=1)
24 {
25     // jeżeli pozycja początkowa i końcowa są takie same, to nie ma przesunięcia!
26     if(from == to) return sf::Vector2f(0,0);
27
28     // policzenie wektora przesunięcia i jego długości
29     sf::Vector2f offset = sf::Vector2f(to.x - from.x, to.y - from.y);
30     float length = sqrtf(offset.x * offset.x + offset.y*offset.y);
31
32     // normalizacja wektora
33     offset = sf::Vector2f(offset.x / length, offset.y / length);
```

@filipekczek7 pokazał jak zrobić poruszanie się obiektu do punktu tutaj (<http://pastebin.com/51YhFVez>).

Code ON!