

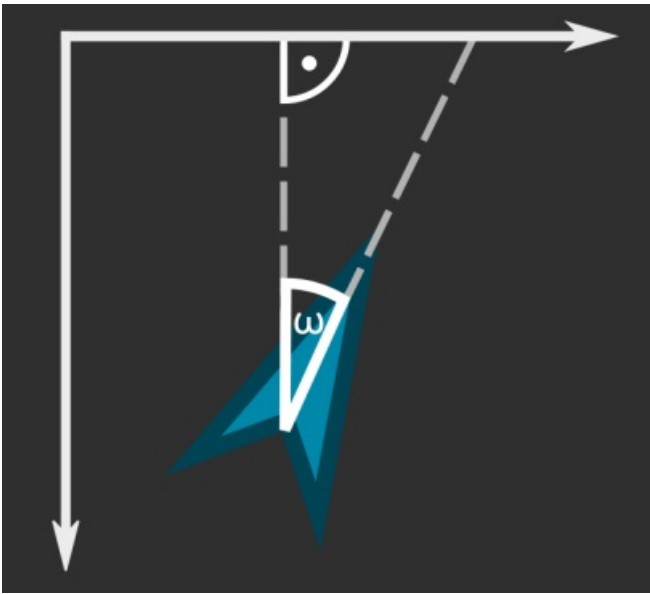
Poruszanie obiektem na podstawie rotacji (sheadovas/poradniki/goto/poruszanie-obiektem-na-podstawie-rotacji/)

Lis 11, 2015 / goto (sheadovas/category/poradniki/goto/)

Jak wyliczyć wektor przemieszczenia posiadając jedynie jego rotację?
Ten wpis jest poniekąd uzupełnieniem teorii matematycznej z *Piszemy grę w SFML'u::Lekcja 2*.

Opis i część matematyczna problemu

Naszym celem jest obliczenie 2-wymiarowego wektora, który pozwoli nam na obliczenie wektora przemieszczenia jedynie na podstawie jego rotacji podanego w stopniach.



Jak widzimy nie jest to tradycyjny układ kartezjański do jakiego jesteśmy przyzwyczajeni, lecz jego odbicie względem osi OY, czyli mamy dokładnie ten układ wg którego jest (zazwyczaj) liczona pozycja w komputerach.

Jesteśmy w stanie zauważyć, że w zasadzie pracujemy na trójkącie, w którym znamy 2 z jego 3 kątów: mamy kąt prosty oraz kąt ω , który jest kątem obrotu względem pozycji startowej. Rysunek pokazuje, że wektor przemieszczenia to przeciwprostokątna trójkąta.

Wykorzystując podstawową wiedzę matematyczną jesteśmy w stanie wyprowadzić wzór na wektor leżący na tej prostej posługując się jedynie kątami i funkcjami trygonometrycznymi. Uzyskamy następujący wzór dla układu kartezjańskiego:

$$u = (\sin\omega, \cos\omega)$$

Jednak pamiętamy, że nasz układ jest odbiciem układu kartezjańskiego względem osi OY, czyli rozwiązaniem naszego problemu jest równanie:

$$u' = (\sin\omega, -\cos\omega)$$

Poruszanie się

Skoro wiemy już jak wyznaczyć taki wektor, to większość roboty mamy już z głowy (jeżeli nawet nie całą), dlatego przykładowy kod (napisany w c++ i SFML) pozostawiam do samodzielnej analizy (nie ma tam niczego, czego nie omówiłem choćby we wcześniejszych wpisach, jeżeli nie wiesz do czego służy delta, to zapraszam tutaj (sheadovas/poradniki/goto/podazanie-za-obiektem-w-przestrzeni-2d/)).

DemoC++

```
1  #include <SFMLGraphics.hpp>
2  #include <cmath>
3
4  #define PI 3.14159265
5  #define SPEED 300
6  #define SENSITIVITY 300
7
8
9  sf::Keyboard::Key rotLeftButton = sf::Keyboard::Left;
10 sf::Keyboard::Key rotRightButton = sf::Keyboard::Right;
11 sf::Keyboard::Key moveButton = sf::Keyboard::Up;
12
13
14 /**
15     Zamiana stopni na radiany
16 */
17 inline float toRad(float degrees)
18 {
19     return (degrees*PI) / 180.f;
20 }
21
22
23 /**
24     Pobranie wektora przemieszczenia na podstawie kąta podanego w stopniach
25 */
26 sf::Vector2f getMovementVector(float rotation)
27 {
28     sf::Vector2f v;
29     float rad = toRad(rotation);
30
31     v.x = std::sin(rad);
32     v.y = -std::cos(rad);
33
```

Code ON!