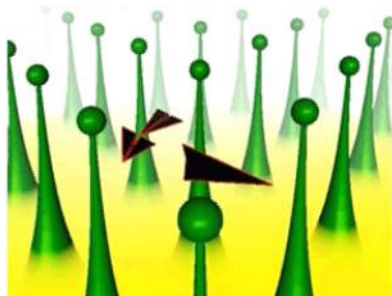
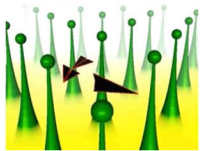


ТЕМА №14

# Ориентация



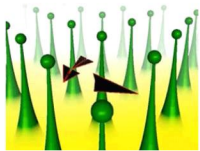


# Съдържание

## Тема 14: Ориентация

- Ойлерови ъгли
- Астрономически координати
- Динамика на полета
- Клониране

# Ориентация в 3D



# Ориентация в 3D

## Под ориентация разбираме

- Завъртяността на обект в 3D  
( т.е. не къде сме, а накъде сме)

## Цел на ориентацията на обект

- Еднозначното ѝ представяне в 3D
- Удобно за потребителя
- Предсказуем резултат

## **Някои методи за ориентация**

- Ойлерови ъгли
- Астрономически координати
- Динамика на полета

## **Избор на метод**

- Според наличните изисквания за ориентация в 3D

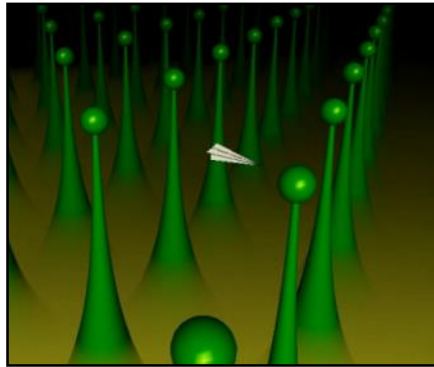
# Реализация на различните методи

- Чрез умножение на матрици
- Подробности в теми 19 и 26



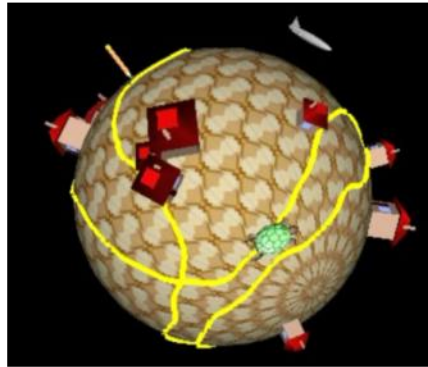
# Примери с контрол на ориентацията

- Слалом на непозната планета
- Нервна костенурка върху планета
- Полет между най-големите градове



“Slalom”

<http://youtu.be/cCxduRWjoRM>



“Turtle Land”

<http://youtu.be/HhIUgQKwc1o>

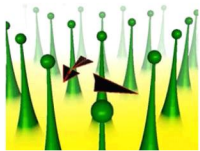


“Around The Word”

<http://youtu.be/Acx9Pa14dkk>

**Ойлерови ъгли**





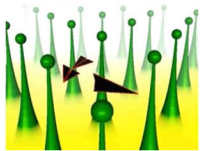
# Ойлерови ъгли

## Ойлерова теорема за ротацията

- Всяка ориентация може да се представи чрез три параметъра
- Следствие: това може да са три ъгла (ротации), но не в произволен ред

## Ойлеровите ъгли

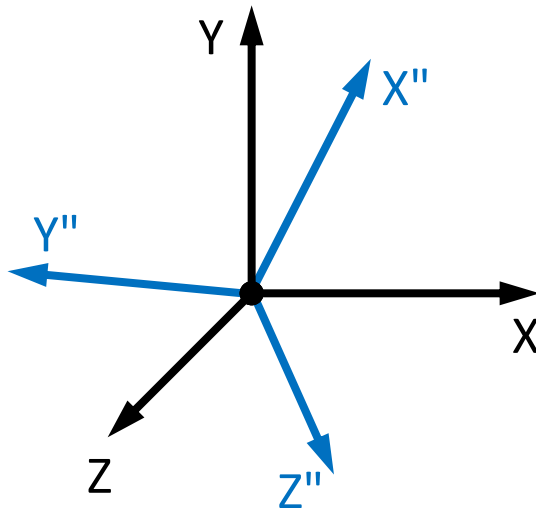
- 12 различни комплекта от по три ъгла



# Пример с комплект $ZXZ$

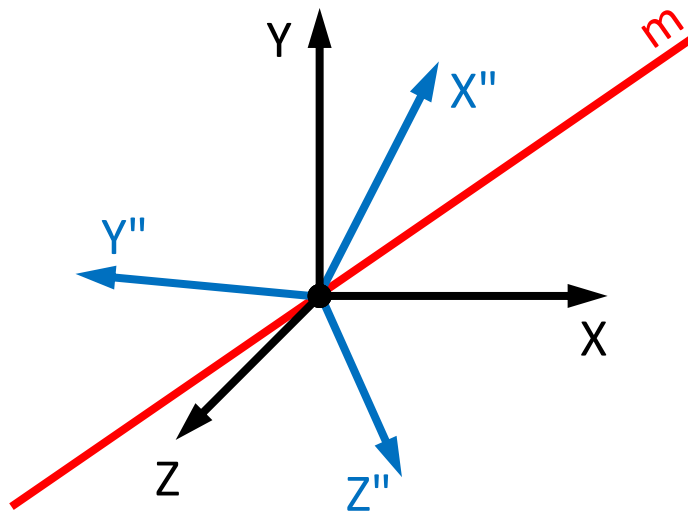
## Начално и крайно положение

- Две координатни системи  $XYZ \rightarrow X''Y''Z''$
- Общо начало (т.е. без трансляция)



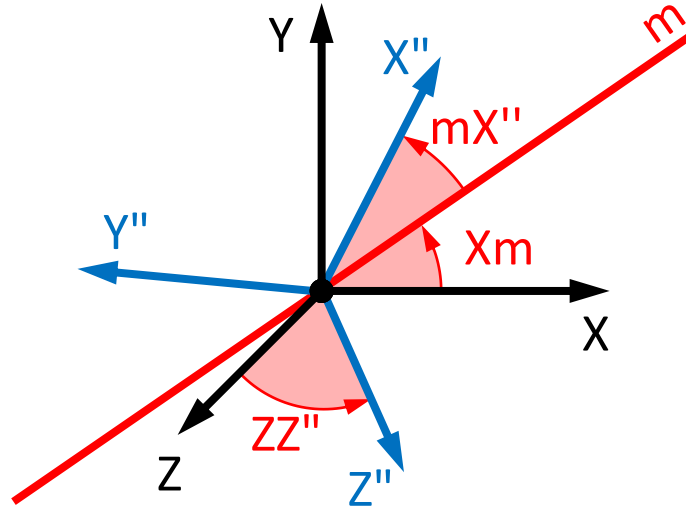
# Обща права

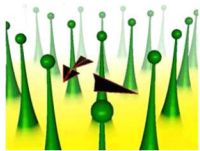
- Обща права  $t$  на равнините  $X''Y''$  и  $X'Y'$
- Правата  $t$  е перпендикулярна на  $Z''$  и  $Z$
- Правата  $t$  ще е ориентир



# Да видим ойлеровите ъгли ZXZ

- Ъгъл №1 –  $Xm$
- Ъгъл №2 –  $ZZ''$
- Ъгъл №3 –  $mX''$





# Построение

## Построение чрез комплект $ZXZ$

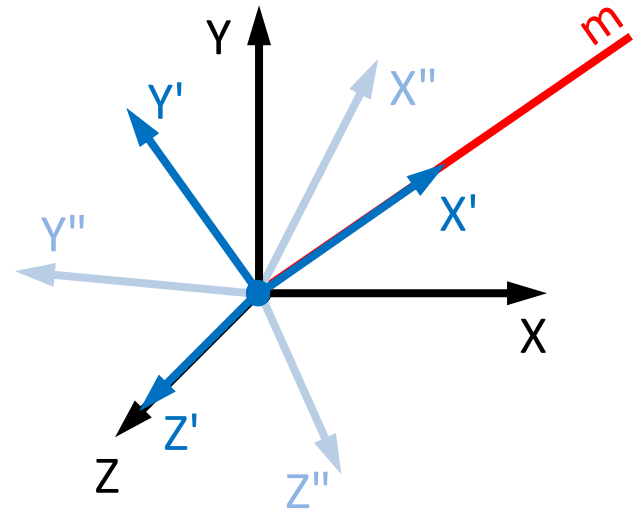
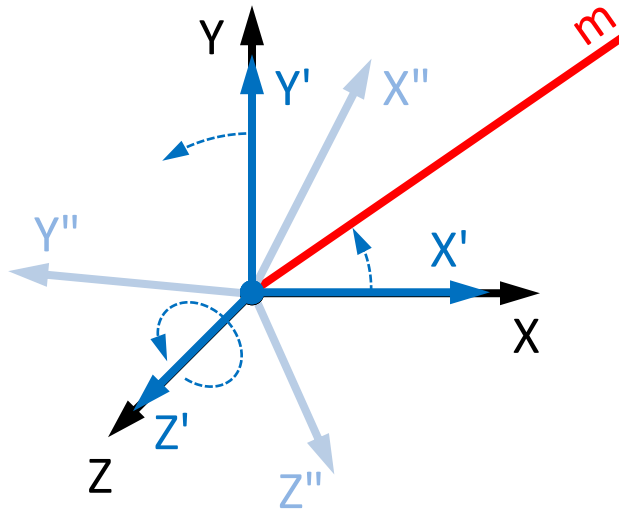
- Въртене около локалната ос  $Z$
- Въртене около локалната ос  $X$
- Въртене около локалната ос  $Z$
- Редът на ротациите е фиксиран

## Локалните оси кръщаваме $X', Y', Z'$

- Първоначално съвпадат с  $X, Y, Z$
- Накрая ще съвпадат с  $X'', Y'', Z''$

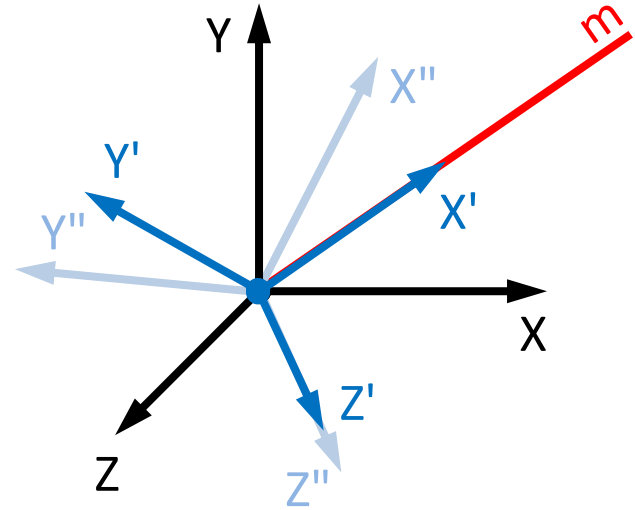
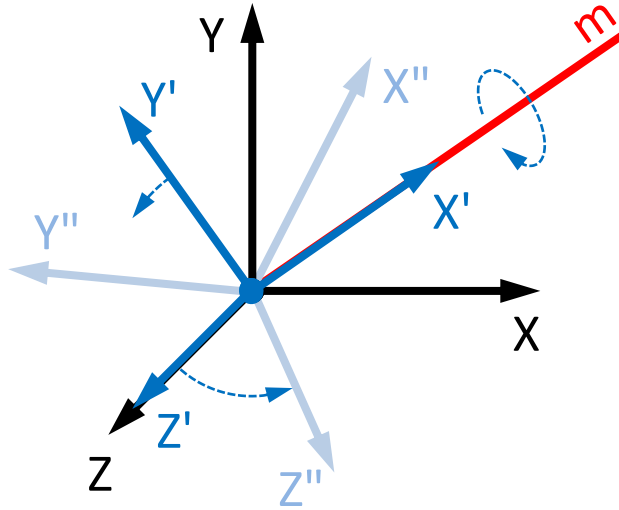
## Ъгъл №1 ( $Xm$ )

- Въртим около  $Z'$  (съвпада със  $Z$ )
- Целта е  $X'$  да отиде от  $X$  в  $m$
- Така координатна ос става обща права



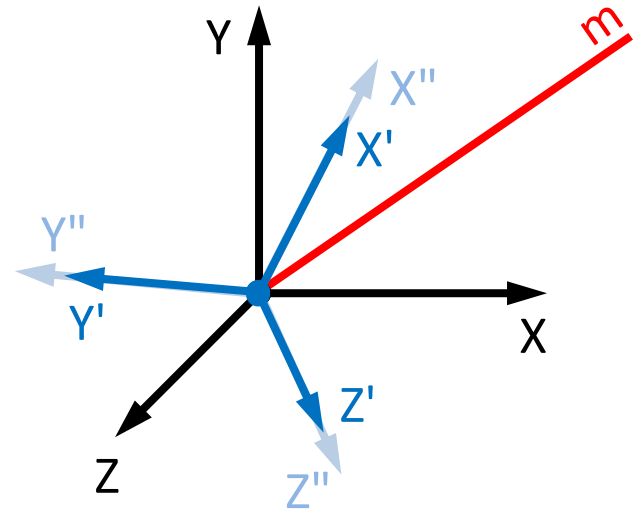
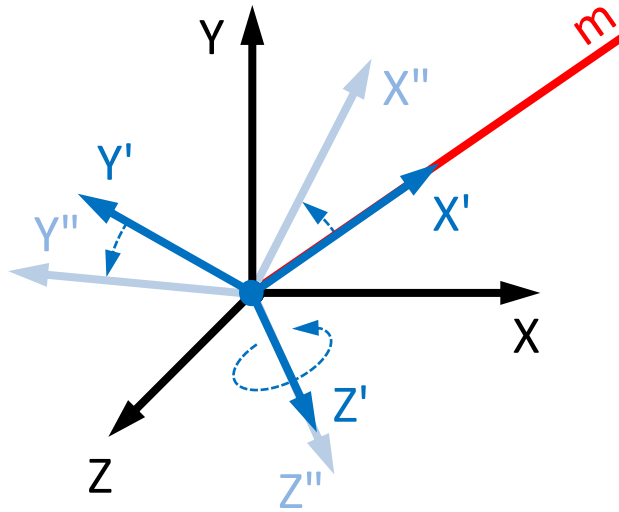
## Ъгъл №2 ( $ZZ''$ )

- Въртим около  $X'$  (съвпада с  $m$ )
- Целта е  $Z'$  да отиде в  $Z''$
- Така двете равнини се слепват

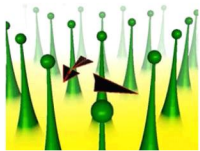


## Ъгъл №3 ( $mX''$ )

- Въртим отново около  $Z'$
- Целта е  $X'$  да отиде от  $m$  в  $X''$
- $Y'$  няма къде да ходи освен в  $Y''$  (Защо?)



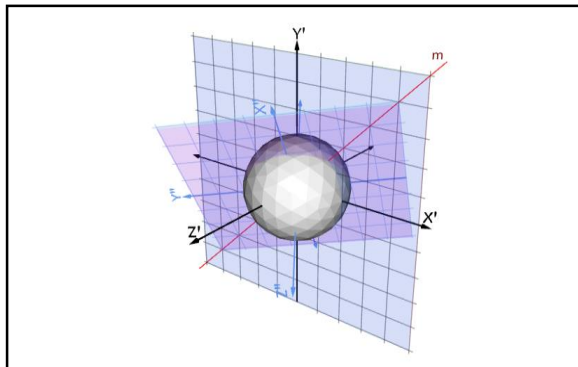




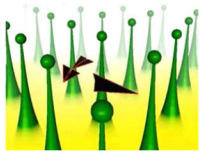
# Демонстрация

## Стъпки с ойлерови ъгли

- Обща права
- Обща равнина
- Общо пространство



# **Астрономически координати**



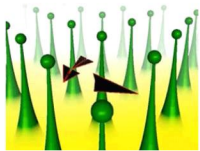
# Основни идеи

## Астрономически координати

- Определяне на положението на обекти по небето (звезди, планети, спътници, гарги)
- Различни модели (хоризонтален, екваториален, еклиптичен и др)

## В компютърната графика

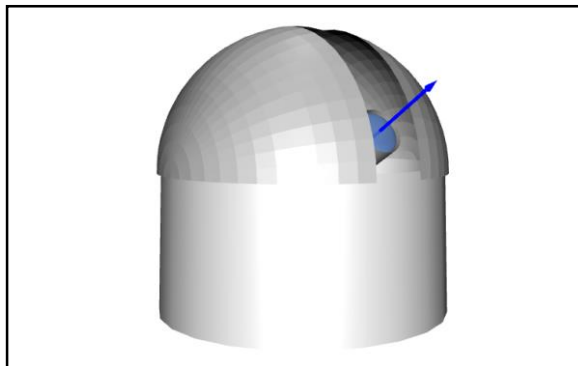
- Най-често се среща хоризонталният метод или някоя негова модификация

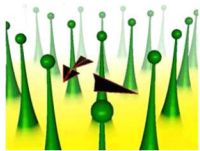


# Хоризонтален метод

## Ориентация на телескоп

- Хоризонтално въртене наляво-надясно
- Вертикално въртене нагоре-надолу





# Нещо липсва

## При ориентацията на телескоп

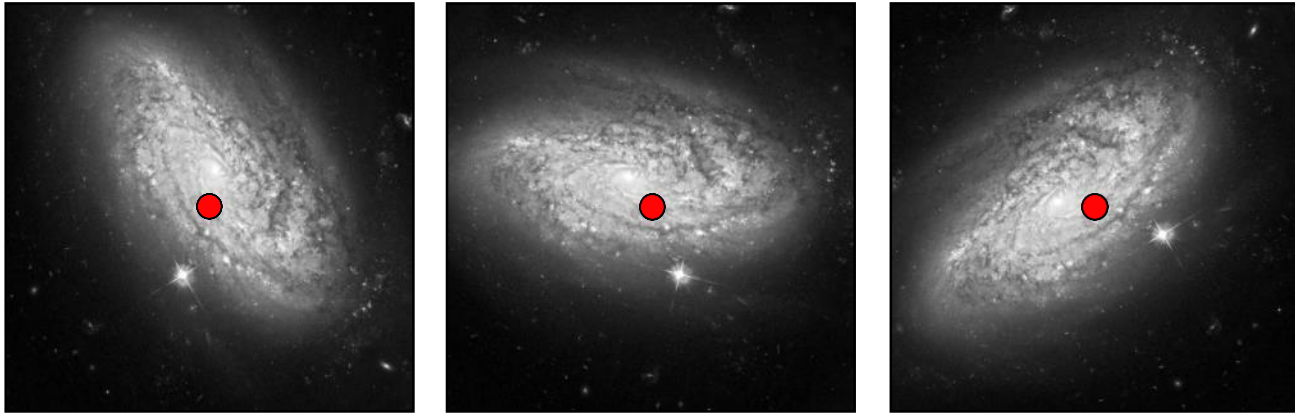
- Може да се завърти на всички посоки  
(това, че не може надолу, е дизайнерско решение)
- Ориентацията се определя от два ъгъла
- Ойлер обаче твърди, че ни трябва три

## На кого да вярваме

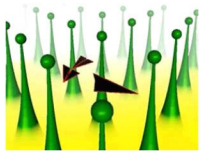
- На очите си или на Ойлер?
- Къде е разминаването?

# Липсващото число

- Без него ориентацията не е еднозначна
- Галактика NGC 3021 при една и съща ориентация на телескопа



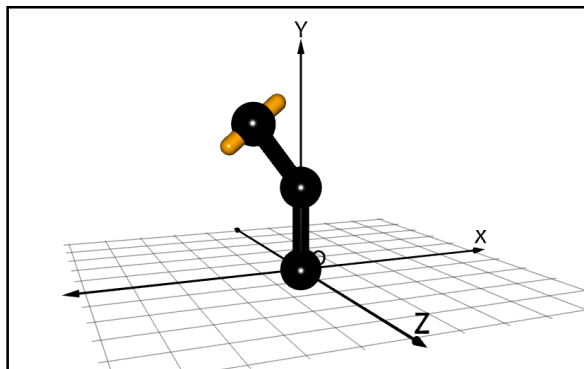
- Липсва фиксиране на въртенето на сцената около централната точка

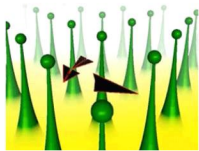


# Приложение

## Виртуален механизъм

- Вертикален ъгъл
- Хоризонтален ъгъл
- Ъгъл на въртене около собствената ос





# Ориентация с вектор

## Преимущества

- Вектор за посока, ъгъл за завъртяност
- Интуитивна дефиниция на посока

## Недостатък

- Ойлер пак е недоволен от нас
- Вектор и ъгъл са ... четири числа



$$\begin{pmatrix} \alpha \\ x \\ y \\ z \end{pmatrix}$$

## Кое число е излишното?

- Определено не е ъгълът  $\alpha$   
(без него се губи еднозначността)
- Не е и някоя от координатите  $x$ ,  $y$  или  $z$  на вектора  
(той трябва да е тримерен)

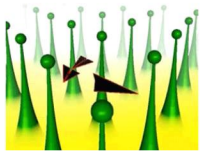
## Може ли някой да обясни парадокса

- Хем има излишно число, хем всички са жизнено важни

## Бонус-упражнение за 5 т.

- Само за първия верен отговор във форума на курса

# Динамика на полета



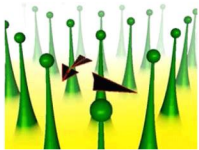
# Роли на ориентацията

## Статична роля

- Определяне на завъртността на обект
- Подходящи са ойлеровите ъгли или астрономическите координати

## Динамична роля

- Завиване при движение в 3D
- Заимстване на модели от авиониката



# Модел на ориентацията

## Координатна система

- Декартова, локална
- Движи се и се върти заедно с обекта

## Въртене

- Около локалните координатни оси
- Сложното въртене се композира от няколко по-прости ротации

# Използване в компютърната графика

- Клониране  
(подробности в тема 14)
- Геометрично създаване на фрактали  
(подробности в тема 22)
- Сложни системи от свързани елементи  
(подробности в тема 25)
- Движения на части и на цели обекти  
(подробности в тема 27)

## Преимущества

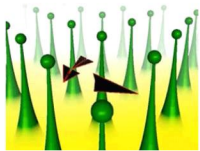
- Движението не зависи от пространственото положение и ориентация

## Недостатъци

- По-лесна ориентация, ако се „вживеем“ в обекта
- Завой наляво на екрана може да изглежда надясно

# Основни характеристики

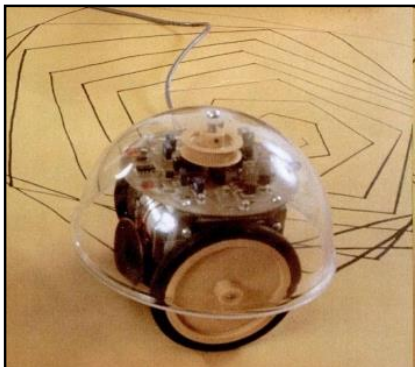
- Всичко се измерва спрямо  $AZ^{\text{ЪТ}}$
- Няма глобална координатна система
- Няма точка  $(0,0,0)$   
(т.е. има, това съм  $AZ$ , където и да съм)



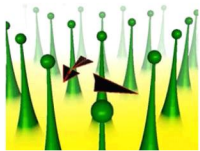
# История отпреди 40 г.

## Роботи, оставящи следи по пода

- Контролират се с програма
- Наричали са се „костенурки“ заради фòрмата и скоростта на пълзене







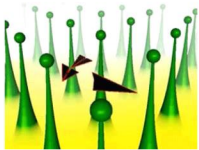
# Управление на роботите

## Команди

- Движение напред и назад
- Завой наляво и надясно

## Използване на роботите

- За образователни цели
- Обучение по математика и информатика



# Костенуркова графика

## Наименование

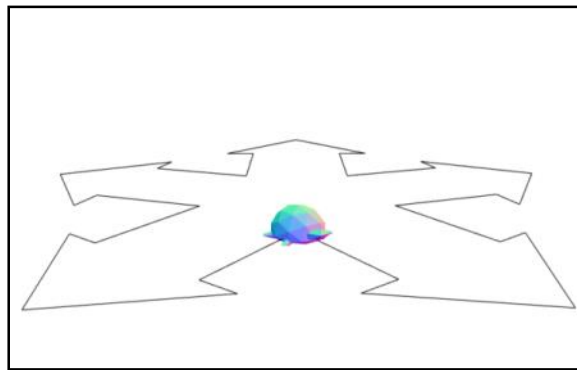
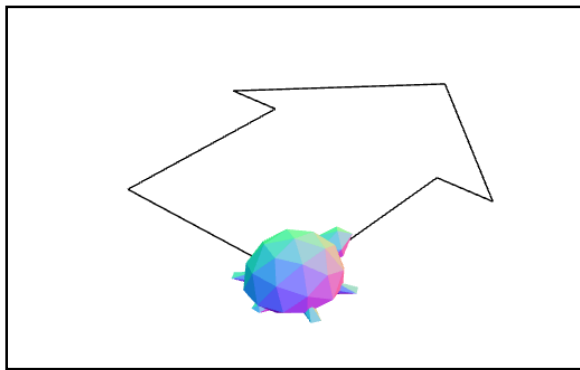
- На английски *turtle graphics*

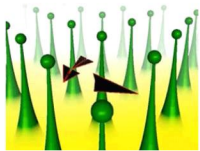
## Език за програмиране Лого

- Създаден преди 40-50 години
- Досега над 300 версии и диалекти
- Имат костенуркова графика

# Независимост от ориентацията

- Елементарна къща
- Петокъщии без основа

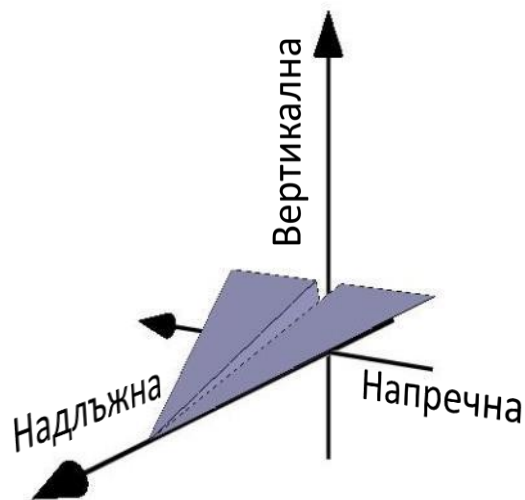




# Ориентация в 3D

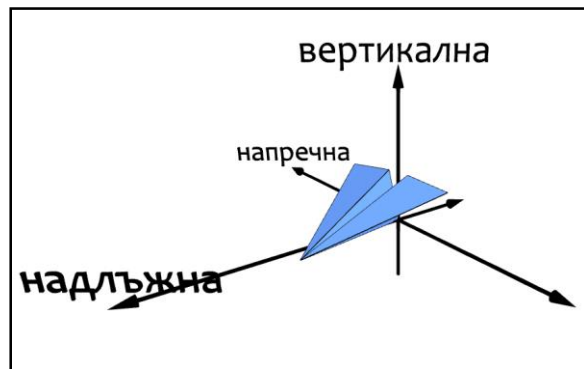
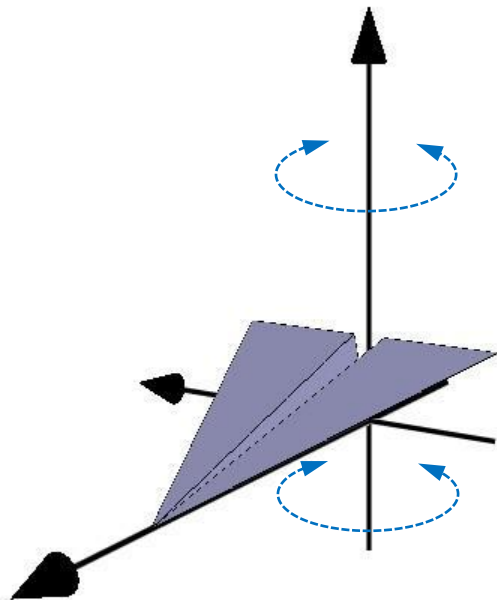
## Оси на локална координатна система

– Надлъжна, вертикална, напречна



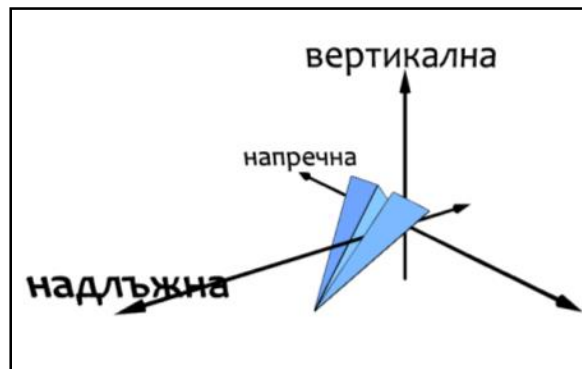
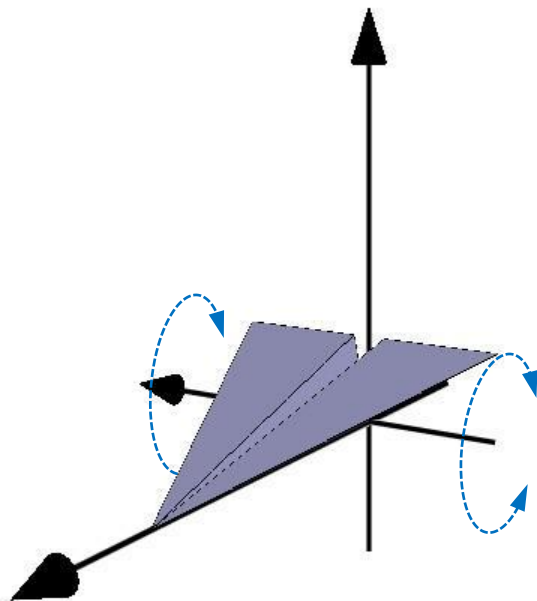
# Въртене 1

- Въртене около вертикалната ос
- Отклонение от курса – *завой*, (англ. *yaw*)



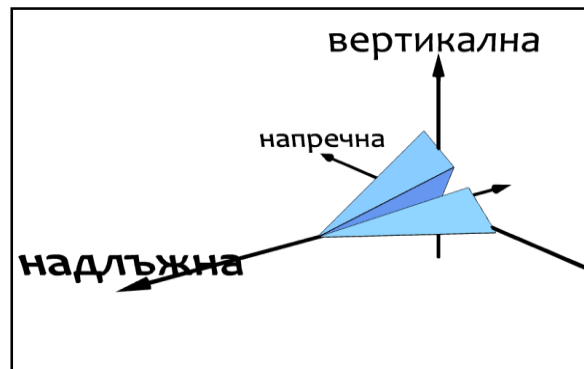
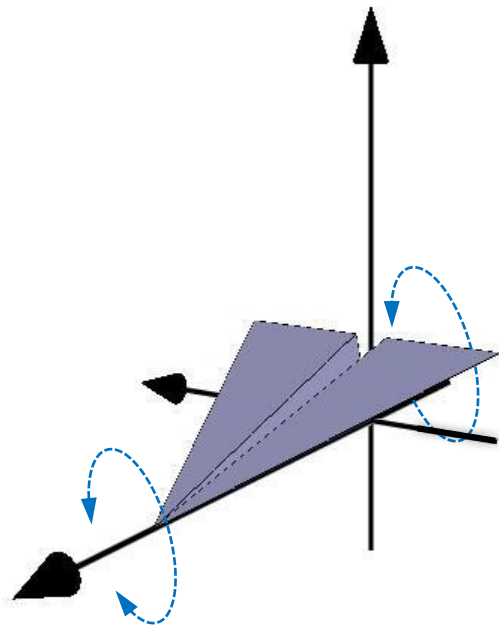
## Въртене 2

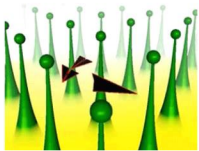
- Въртене около напречната ос
- Наклон на носа – *танграж* (англ. *pitch*)



# Въртене 3

- Въртене около надлъжната ос
- Наклон на крилата – *крен* (англ. *roll*)

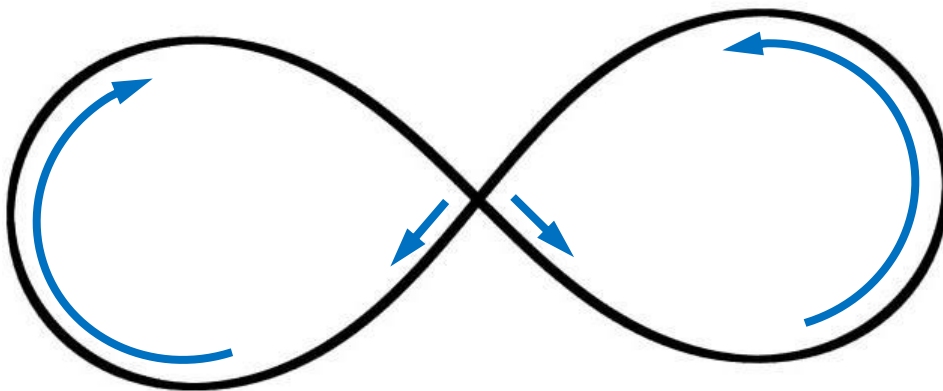




# Пример със самолетче

## Прави осморки във въздуха

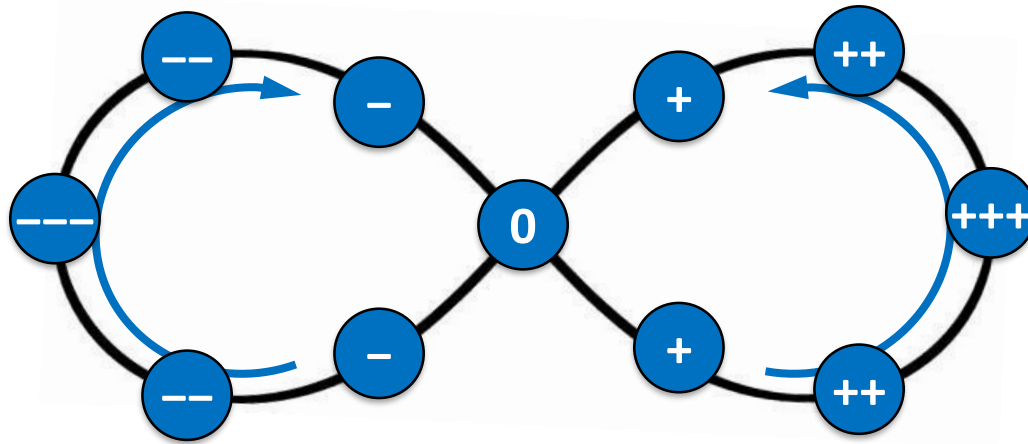
- При движение по едната примка завива наляво, а по другата – надясно
- Плавен преход между двете примки





# Анализ на ъгъла на завиване

- Положителен, ако е наляво
- Отрицателен, ако е надясно
- Ъгълът е периодична функция



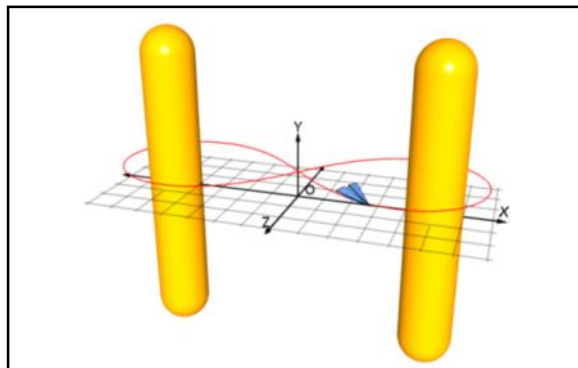
# Уравнение на движението

- Правим малки стъпки и малки завои
- Ние сме в точка  $p_i$ , движим се със стъпка  $\vec{v}_i$ , ъгълът на завиване е  $\alpha_i$  и пресмятаме в обратен ред:

$$\alpha_i = k \sin mt$$

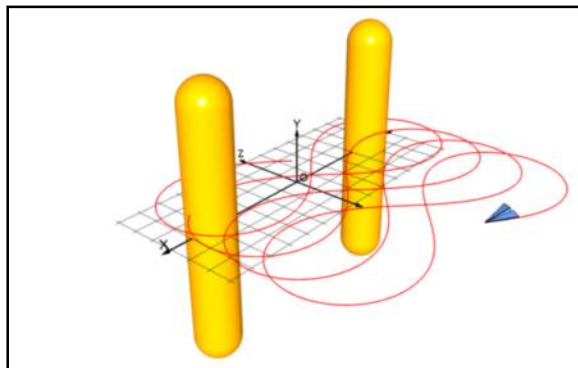
$$\vec{v}_i = \text{rot}(\vec{v}_{i-1}, \alpha_i)$$

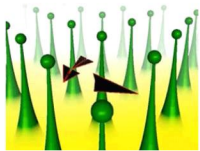
$$p_i = p_{i-1} + \vec{v}_i$$



## Параметри $k$ и $t$

- Избрани така, че кривата да се затвори
- Ако не се затвори се получава лошо

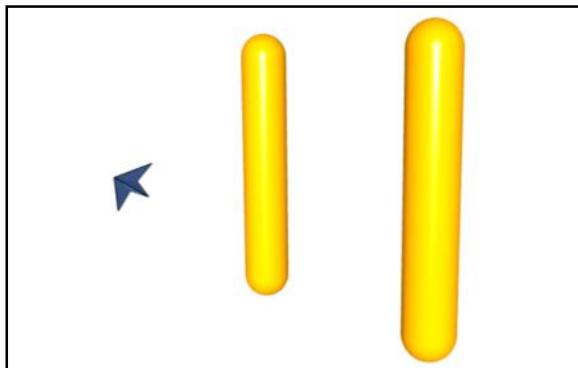




# Допълнителен ефект

## По-естествен полет

- При завой самолетът да се накланя
- Тъжно: наклоним ли самолета, променяме и равнината на траекторията



## Решение с две координатни системи

- Първата е за навигация  
(дава координатите на самолетчето)
- Втората е за ориентация  
(дава наклона на самолетчето)

## Решение с една координатна система

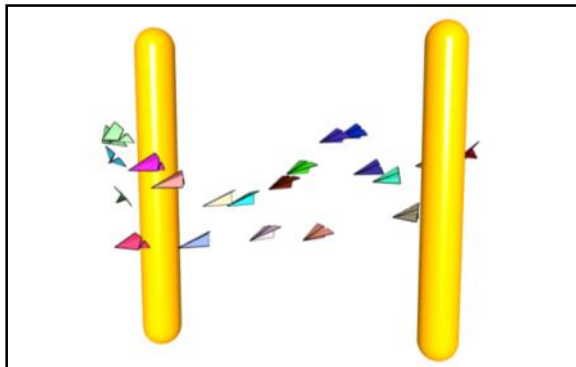
- Проблемът е да се съчетаят  
(промяната в едната влияе негативно на другата)
- И все пак да пробваме

– Споделяне на една и съща координатна система

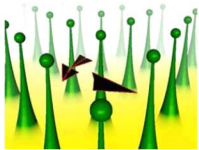
1. Правим завой
2. Стъпка напред
3. Създаваме кадър

1. Правим завой
2. Стъпка напред
3. Наклон встрани
4. Създаваме кадър
5. Обратен наклон  
до хоризонтално положение

– Да го видим



**Клониране**



# Клониране

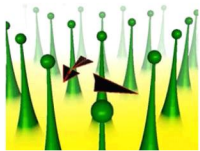
## Основна идея

- Имаме невидим обект-самолет
- Клонираме образ на друг обект там, където е невидимия обект
- Използваме неговите координати и ориентация

## Каква е полза?

- Спестяват се много сметки

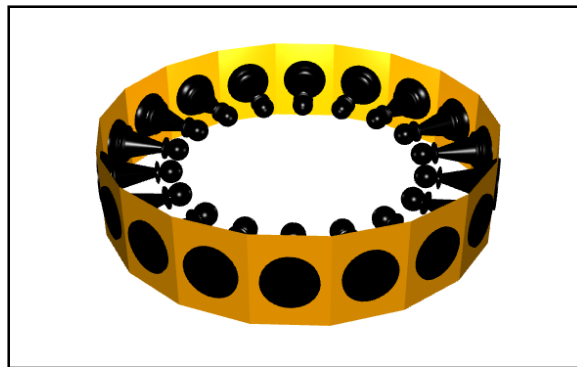
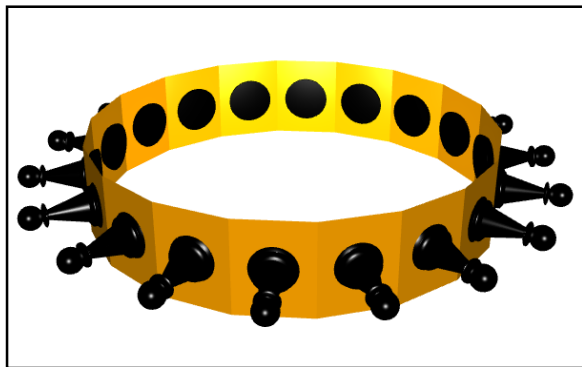




# Примери без сметки

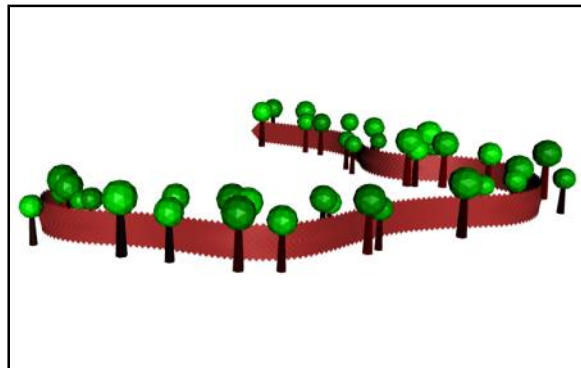
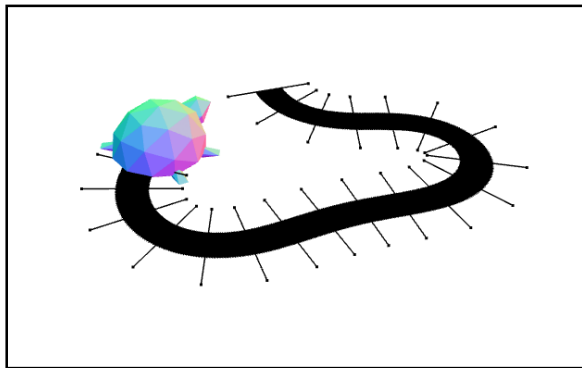
## Пешки разположени в кръг

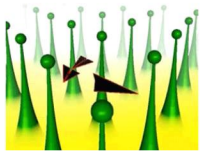
- Садистичен вариант
- Мазохистичен вариант



# Последен пример

- Метод на стоножката
- Ограда с дръвчета по нормалния вектор към нея (той е напречната ос от авиониката)





# Повече информация

[**PARE**] стр. 42-45, 54-58, 102-106

[**VINC**] стр. 69-72

## А също и:

- **Astronomical Coordinate Systems**  
<http://spider.seds.org/spider/ScholarX/coords.html>
- **Maths - Euler Angles**  
<http://www.euclideanspace.com/maths/geometry/rotations/euler/index.htm>
- **Roll, Pitch, and Yaw | How things fly**  
<http://howthingsfly.si.edu/flight-dynamics/roll-pitch-and-yaw>

**Край**