

ТЕМА №11

Анимация





Съдържание

Тема 11: Анимация

- Принципи на анимацията
- Линейно движение
- Движение от точка до точка

Принципи на анимация



Етимология на анимация

Етимология

- От латински *animo* – давам живот

Общ корен с други думи

- Аниматор (на филми, на гости)
- Анималист (художник на животни)
- Аниме (японска анимация)
- Анимизъм (всичко е живо и има душа)
- Реанимация (в болницата)



Измамата

Анимацията е двупосочна измама

- Аниматорът мами зрителя (целенасочено)
- Зрителят се оставя да е мамен (със задоволство)

Следствие

- С проектите трябва да ме измамите...
... по начина, по който ви уча да мамите



Предпоставка

Биологическа предпоставка

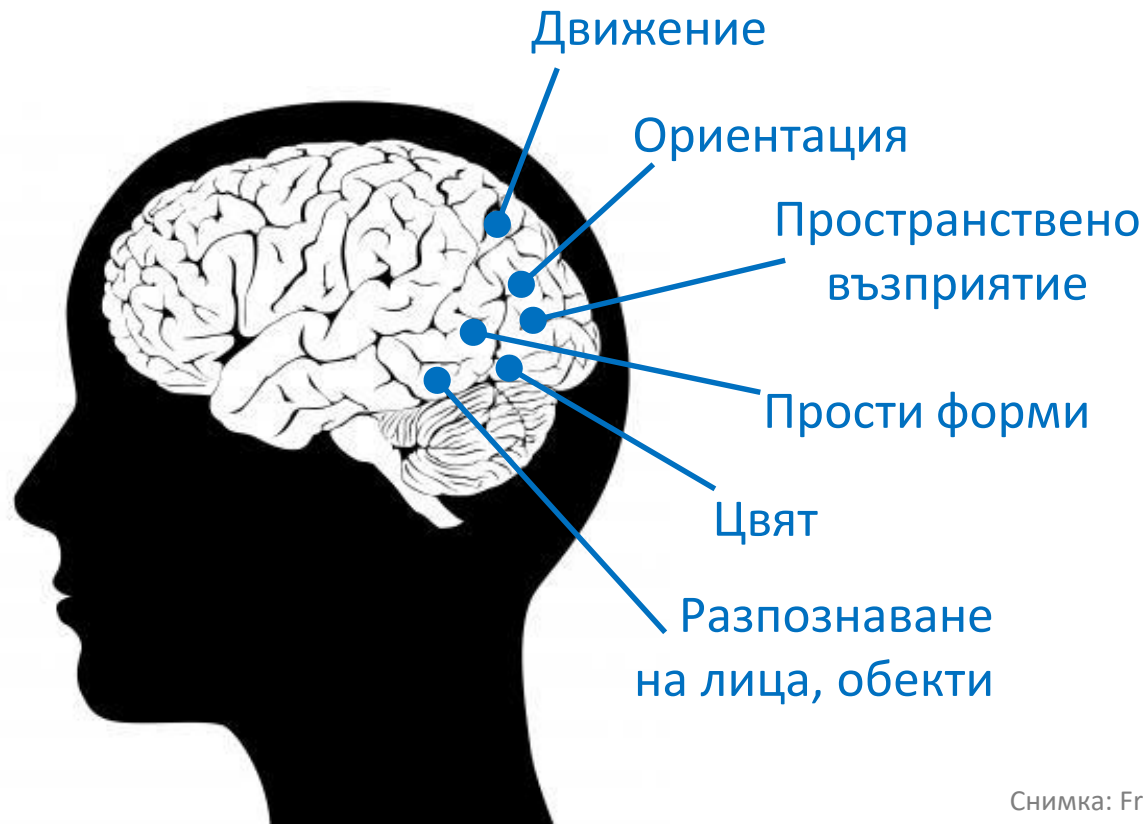
- Човек гледа с очите, но вижда с мозъка
- Сензорите в очите, нервните пътища и мозъкът имат ограничен капацитет

Човек не вижда

- Прекалено бавните движения
- Прекалено бързите движения
- Невидимите движения или докато мижа



Човешкият мозък*



* Не в реален мащаб

Снимка: FreeDigitalPhotos.net



Виждане

Картина в мозъчната кора

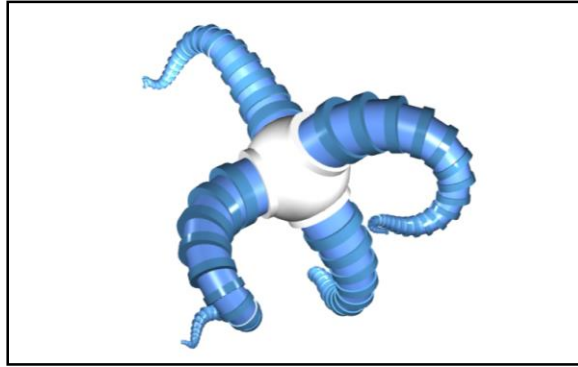
- Задържа се за около $1/15$ от секундата

Ако в мозъка постъпват

- По-малко от 15 образа в секунда – виждат се като различни образи
- Повече от 15 образа в секунда – виждат се като непрекъснато движение

Да опитаме

- Една и съща анимация с различни fps



Акинетопсия

- Невъзможност да се вижда движение
(травма във визуалния кортекс)

Veränderungsblindheit

- Свойство на човешкото зрение да не забелязва някои движения





Пак за анимацията

Реализация

- Създава отделни кадри
- Ако има движение, то е дискретно
(в смисъл обратно на непрекъснато, а в другия си смисъл е извън темите на курса)

Възприема се като движение

- Ако кадрите се сменят бързо
- Ако кадрите се променят бавно

Брой кадри в секунда

- FPS (*Frame per second*)
- Има много стандарти – 24, 25, 30, 60, ...
- Исторически първият е 24 fps – от специфичната скорост на филмовите ленти – 18 инча в минута (а пък ширината на релсовият път идва ... от осевото разстояние при колесниците на Древен Рим)

Нисък FPS

- Щади обема на визуални данни, който се генерира
- Удобен за бавни и спокойни сцени

Висок FPS

- Изисква сериозна производителност
- Удобен за спортни предавания, екшъни, компютърни игри, 3D филми



Реализация

Техники в анимацията

- Според обектите: дескриптивна, релативна, трансформационна
- Могат да се ползват комбинирано

Дескриптивна анимация

- Обектите са описани с техните свойства
- Промяната им създава анимация
(Движението в 3D в чрез промяна на центъра и главните оси)
- Най-лесна за използване

Примери

- „Завърти се в тази посока“

Релативна анимация

- Задава се относителната промяна на свойствата, а не абсолютната промяна
- Движение чрез вектори, тангенти, ...
- По-трудна, ползва диференциална геометрия

Примери

- „Завърти се наляво“

Трансформационна анимация

- Промяната се реализира с умножение с матрици
- Най-мощна, почти всичко се прави с матрици – движение, въртене, мащабиране, проекция, ...

Примери

- „Умножи с $\begin{pmatrix} 1 & 3 & 0 & -1 \\ 0 & -2 & 1 & 1 \\ 2 & 1 & 1 & 0 \\ -1 & -1 & 3 & 1 \end{pmatrix}$ “



Анимационен цикъл

Отчайващо опростен псевдокод

Създаване на обекти

Цикъл по брой кадри

{

Начало на нов кадър

Промяна на обекти

Рисуване на обекти

Показване на готов кадър

}

Линейно движение



Линейно движение

Видове

- Праволинейно – неправолинейно
- Равномерно – неравномерно
- Еднопосочно – двупосочно

Реализации

- Чрез вектор на скоростта, точка на целта и уравнение на траекторията



Движение чрез вектор

Линейно движение с вектор

- Векторът указва посоката
- Векторът указва скоростта
- Движението е праволинейно и равномерно
(при константен вектор)

Реализация

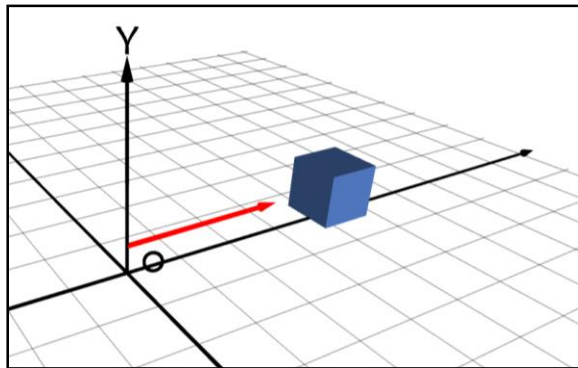
$$P_i = P_{i-1} + \vec{v}$$



Физическа основа

Пример с движение с вектор

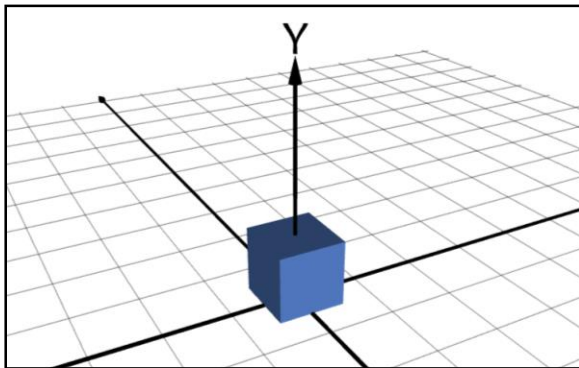
– Брой повторения (време t) и вектор (скорост \vec{v})



for (1 ... t) $P = P + \vec{v}$;

Същото движение и разстояние

- Но с по-голяма скорост



Особености

- Времето е измерено с брой кадри
- Видимата скорост зависи от хардуера
- Има максимална скорост

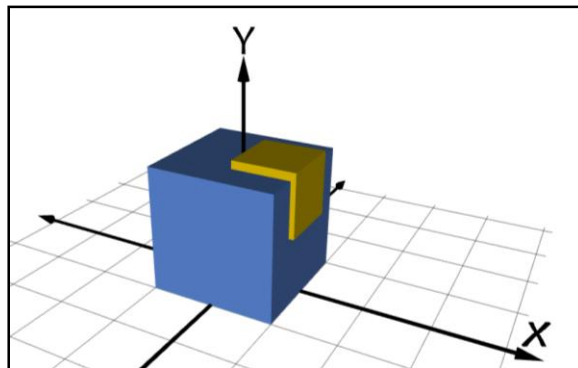


Две движения

Две кубчета, искаме следното:

- Горното се плъзга встрани (скорост \vec{v}_h)
- После пада надолу (скорост \vec{v}_v)

```
for (1 ... t)
{
     $P = P + \vec{v}_h$ ;
     $P = P + \vec{v}_v$ ;
}
```





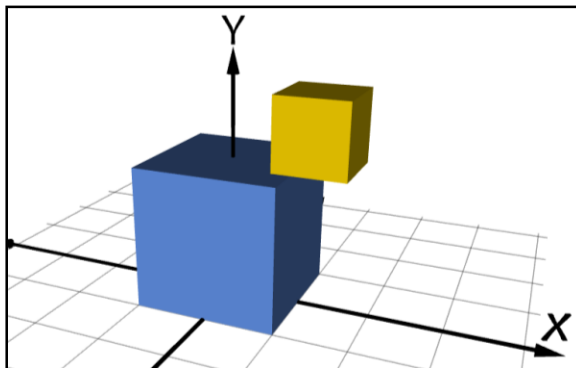
Защо толкова зле?

Поради движенията

- Те са отделни
- А трябва да са едно след друго

for $(1 \dots t_h)$
 $P = P + \overrightarrow{v_h};$

for $(1 \dots t_v)$
 $P = P + \overrightarrow{v_v};$

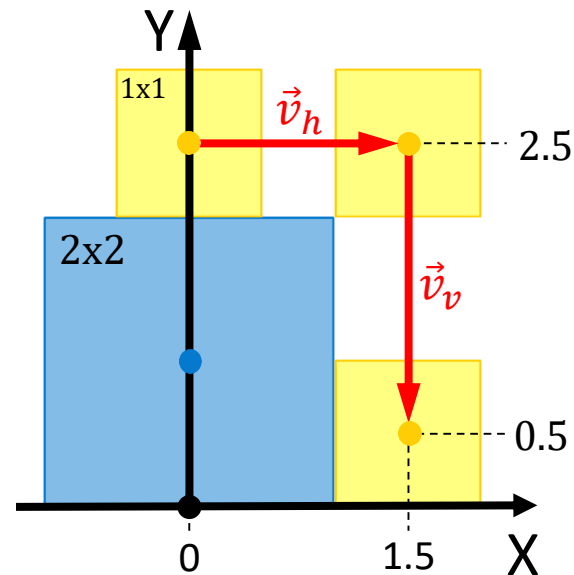
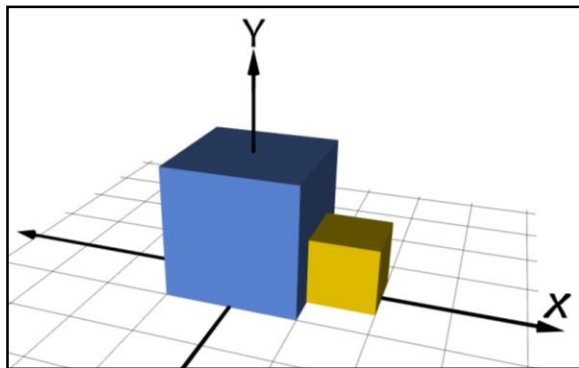


О, пак дефект

- Не пада, а пропада
- Да посмятаме

$$S_h = 1.5, t_h = 100, v_h = 0.015$$

$$S_v = 2.0, t_v = 50, v_v = 0.04$$





Поуките

Поука 1

- За повечето анимации трябва да изчисляваме различните параметри

Поука 2

- Времето (броят кадри) е цяло число, т.е. другите параметри трябва да са съгласувани с това

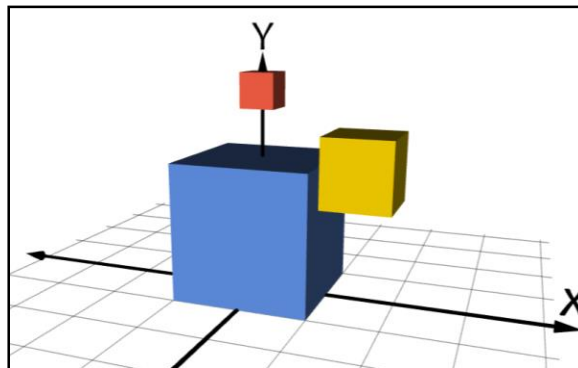


Три движения

Искаме три кубчета едно над друго

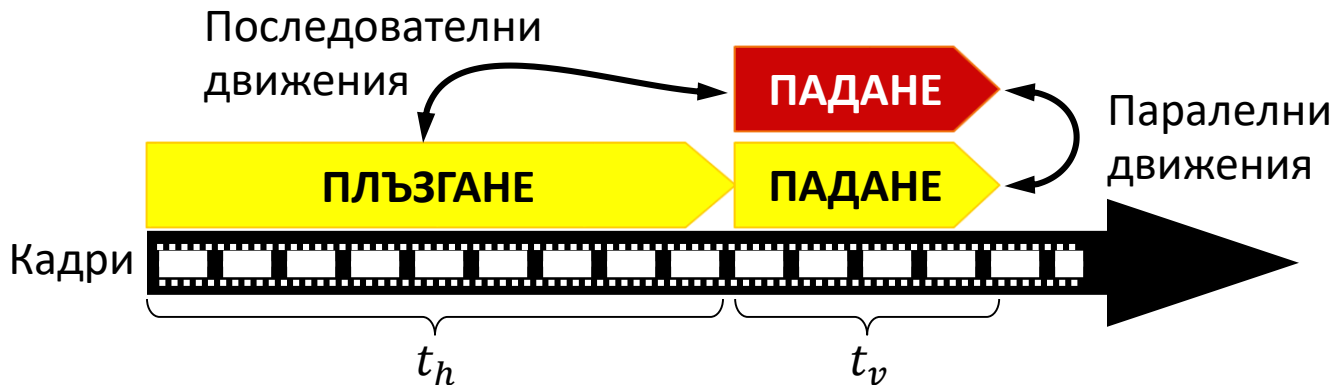
- Средното P се плъзга (\vec{v}_h) и пада (\vec{v}_v)
- Горното Q просто пада (\vec{w}_v)

```
for (1 ...  $t_h$ )  
     $P = P + \vec{v}_h$ ;  
for (1 ...  $t_v$ )  
{  
     $P = P + \vec{v}_v$ ;  
     $Q = Q + \vec{w}_v$ ;  
}
```



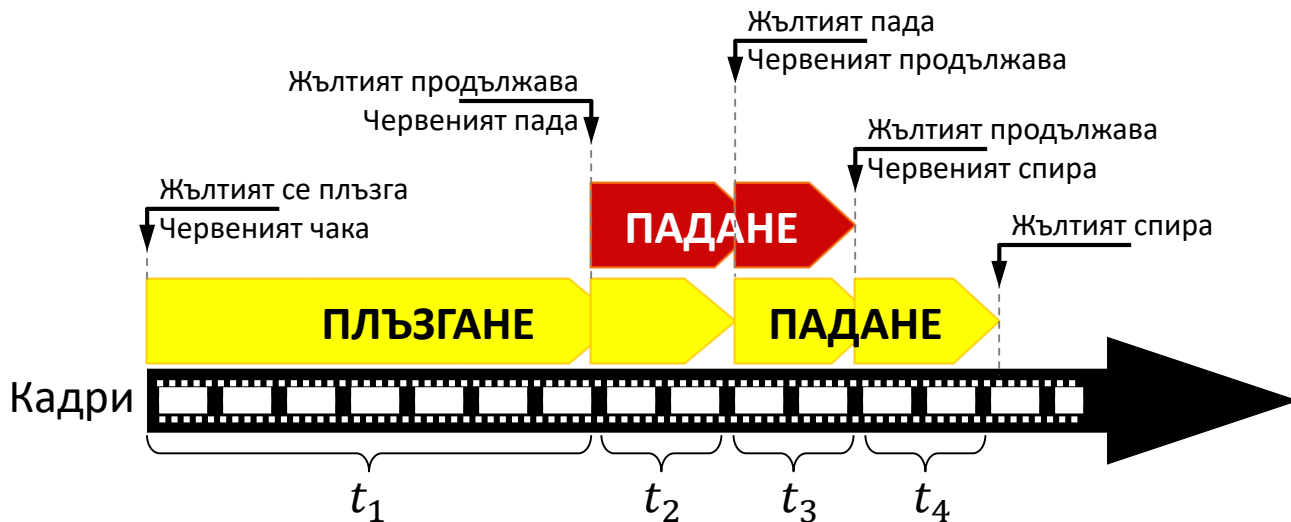
Проблем с новия куб

- Увисва във въздуха
- Допустимо само за детски анимации



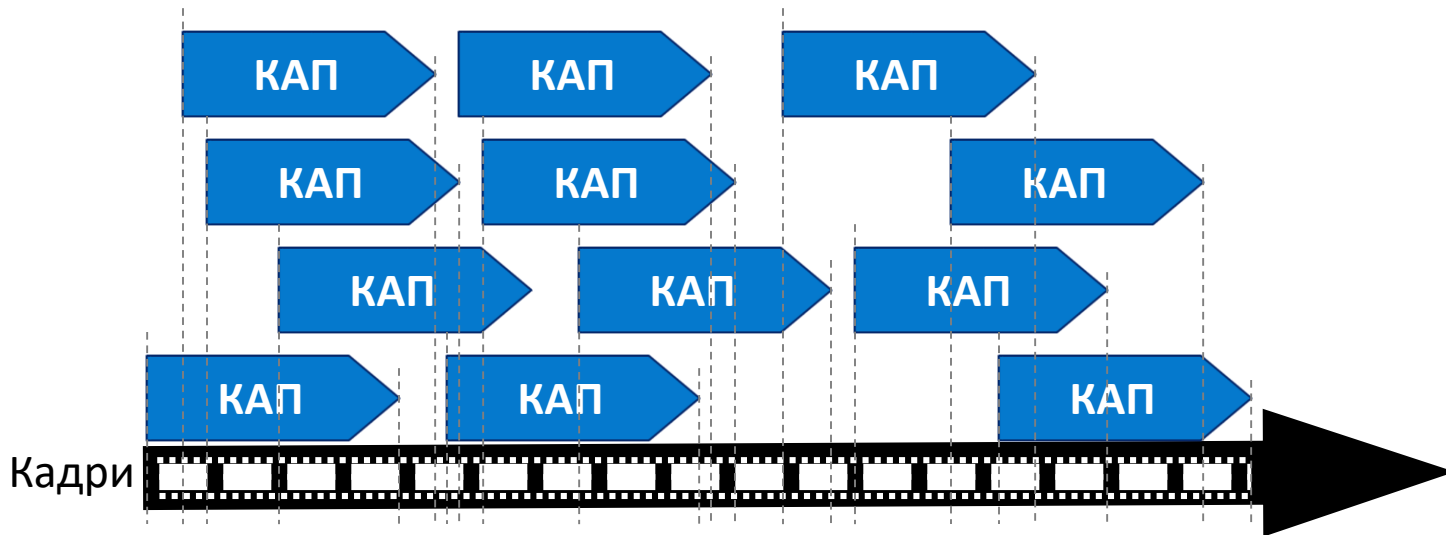
Наивна реализация

- Падането се изнася по-рано
- Движения вече се преплитат
- Разделят се на фрагменти, които не се преплитат



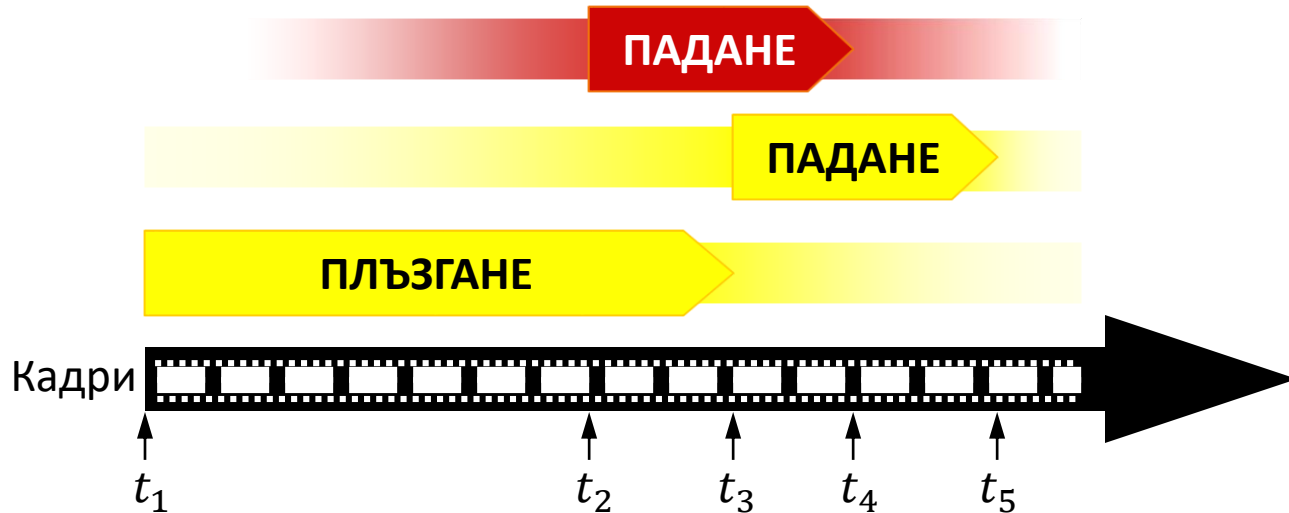
Защо наивна?

- Найлонова торба с китайска бира
- Спукана на 4 места, капят по 3 капки
- Цели 21 интервала



Решение

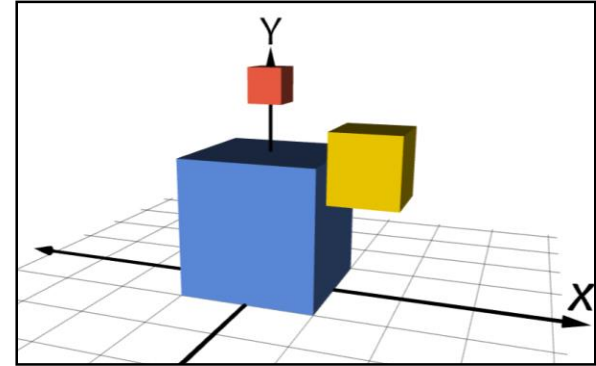
- Движенията дремят през цялото време
- Събуждат се само при нужда
- Независими са едно от друго



Да го пробваме

- Предварително сме изчислили точните моменти $t_1 \dots t_5$

```
for ( $t_1 \dots t_5$ )  
{  
  if ( $t_1 \dots t_3$ )  $P = P + v_h$ ;  
  if ( $t_3 \dots t_5$ )  $P = P + v_v$ ;  
  if ( $t_2 \dots t_4$ )  $Q = Q + w_v$ ;  
}
```



**Движение от точка
до точка**



От точка до точка

Движение от точка до точка

- Най-често срещано движение
- Примитивна форма на движение по траектория

Реализация

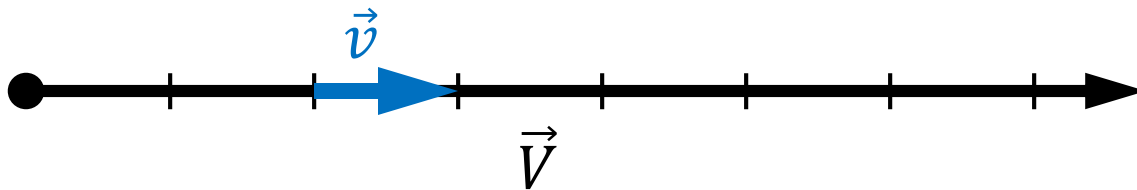
- Чрез вектор на скоростта
- Чрез линейна комбинация



Вектор на скоростта

Пресмятане на вектора

- Разглеждаме отсечката като вектор \vec{V}
- Определяме желания брой стъпки (кадри) n
- Векторът на скоростта е $\vec{v} = \frac{1}{n} \vec{V}$



Преимущества

- Бързи и лесни сметки
- Удобно за праволинейно равномерно движение

Недостатъци

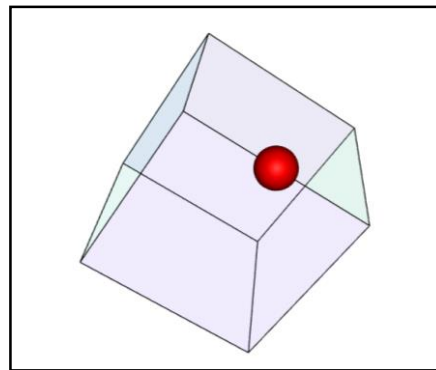
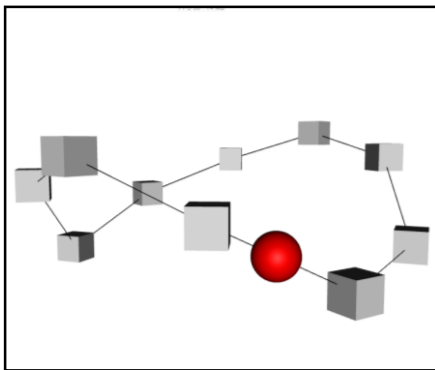
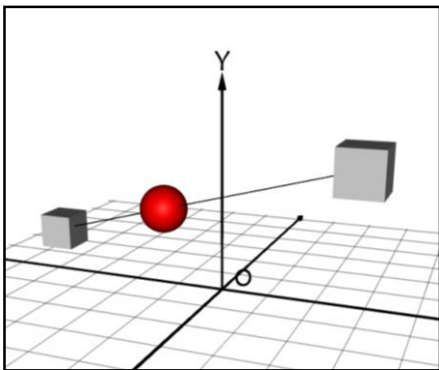
- Неудобно за неравномерно движение
- Неудобно при движеща се целева точка



Примери

Примери на движение с вектор

- Между случайни точки, по пръстен от отсечки и по ръбовете на куб





По-сложен пример

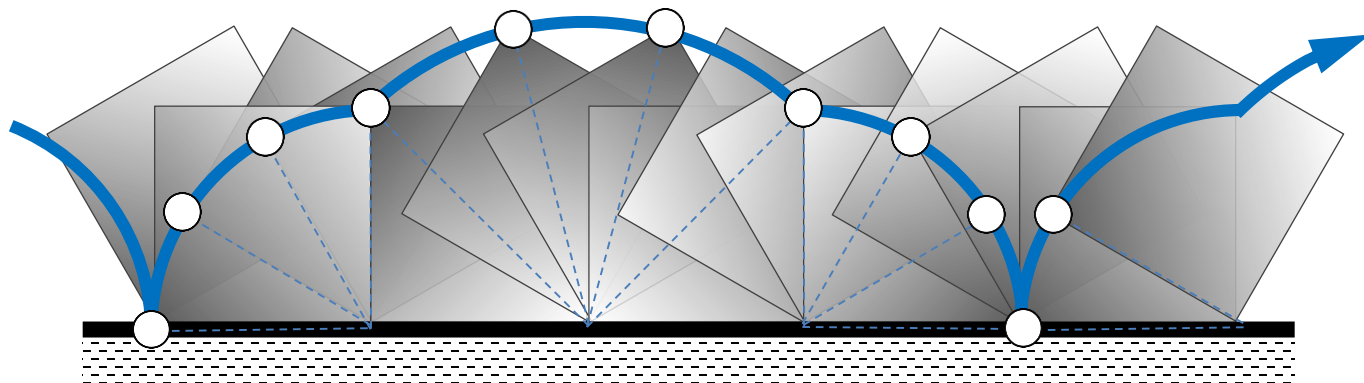
Търкалящо се кубче

- Кубче пада на лента
- Почва да се търкаля по нея
- Стига до края ѝ и пада

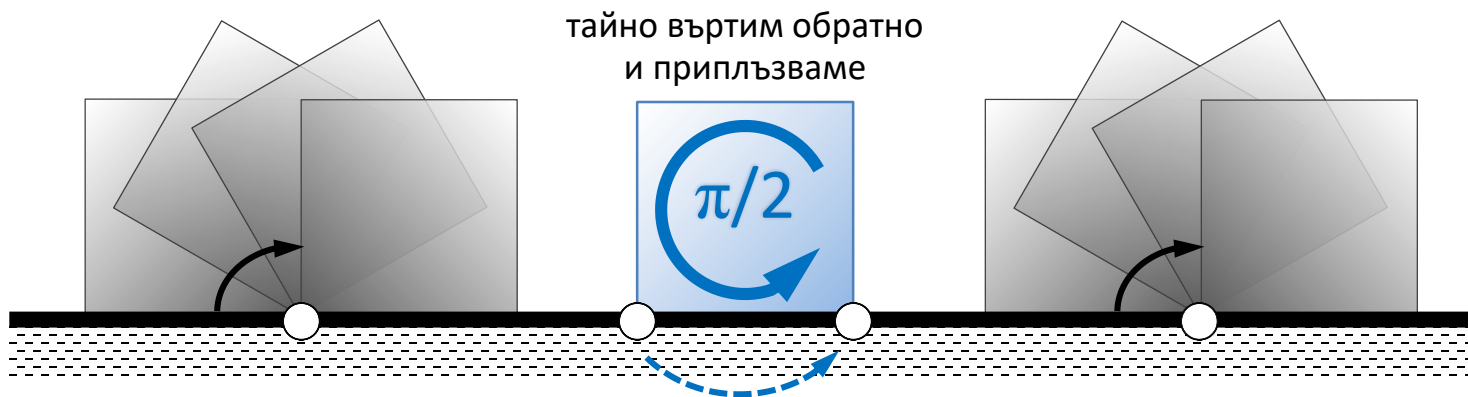
Основен проблем

- Имаме свойство за завъртяност на куб около един от ръбовете, но как да завъртим около другите?

– Решение 1: Смятаме неприятна траектория

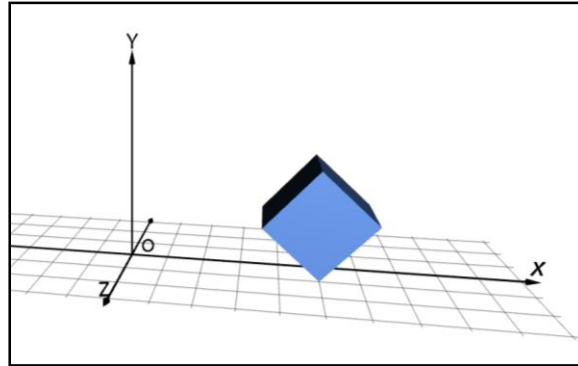


– Решение 2: Не смятаме, а мамим



Да го видим на живо

- Поука: не всеки кадър трябва да се показва, налагат се скрити промени





Линейна комбинация

Движение с линейна комбинация

- Началната и крайната точка
- Параметър $k \in [0,1]$ за координатите

Преимущества

- Ако променяме k неравномерно и крайното движение е неравномерно
- Може да променяме в реално време началната и крайната точка



Свързани пространства

Две (или повече) пространства

- С различна размерност
- С различни координатни системи
(напр. декартова и полярна)
- Движение в едното се проектира в другото

Движение с линейна комбинация

- Най-елементарното им приложение



За нашия пример

Пространство Π_1

- Времето тече паралелно с кадрите
- Движението в това пространство е на времето – равномерно и линейно $t_i = t_{i-1} + 1$

Пространство Π_2

- По естетически съображения искаме движението да е n пъти по-бързо $T = nt$

Пространство Π_3

- Едномерно (т.е. имаме само линия)
- Координатите на точка са (k)
- Движението е на k в $[0,1]$: $k = \frac{1}{2} + \frac{1}{2} \sin T$

Пространство Π_4

- Декартово, координатите са (x, y, z)
- Използваме движението на k от Π_3
- Движението в Π_4 е линейна комбинация от крайните точки A и B : $p = (1 - k)A + kB$

Понеже това е елементарен пример

- Можехме да минем без явни свързани пространства, а направо с

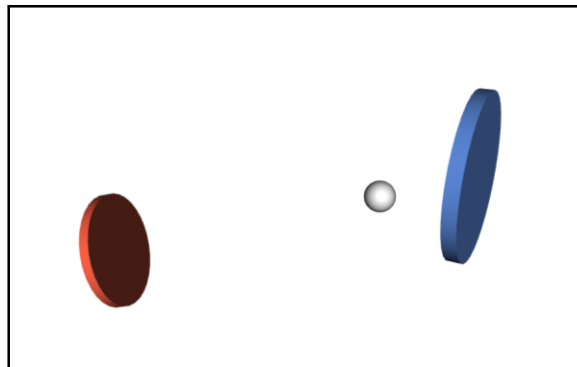
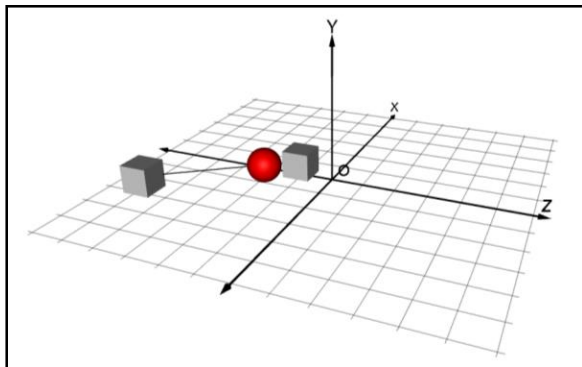
$$p(t) = \frac{1}{2}[(1 - \sin nt)A + (1 + \sin nt)B]$$



Реализация

Примерни решения

- На оригиналната задача
- Вариант на тенис на въздух

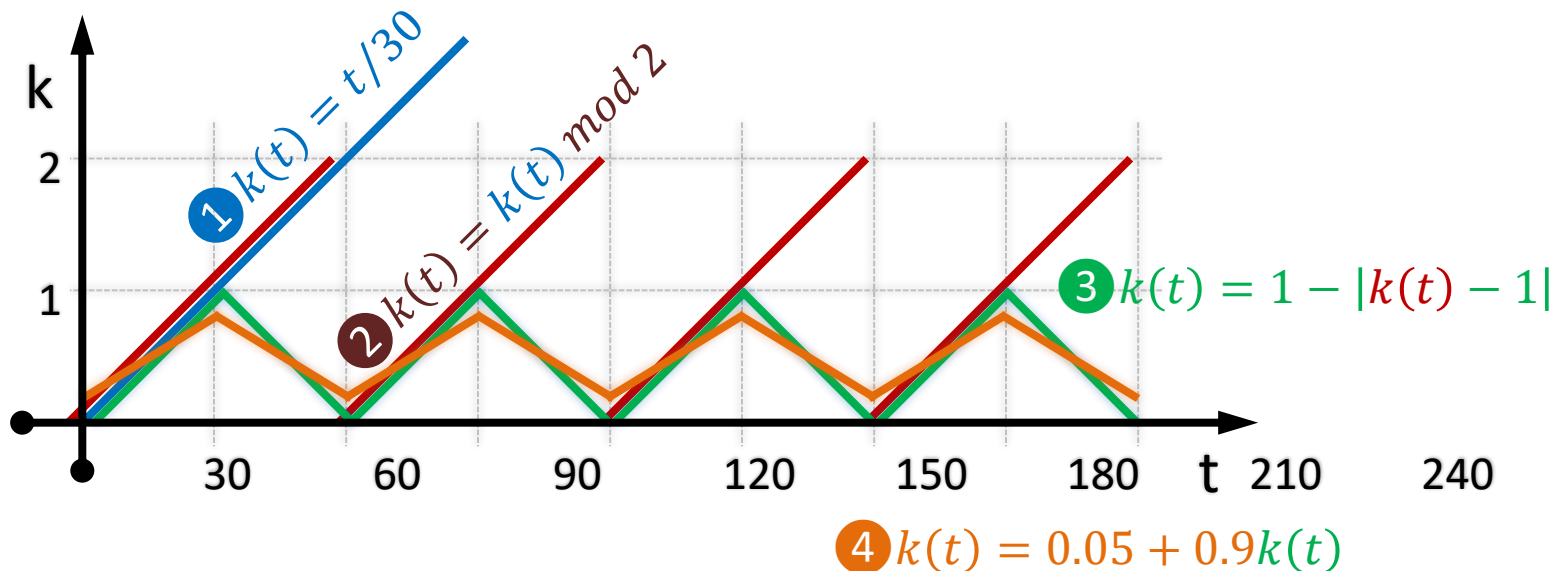




А без $\sin(x)$ или $\cos(x)$?

Равномерно движение напред-назад

– Бленуваме за $k(t) \in [0.05, 0.95]$, без $\sin x$



Въпроси?



Повече информация

[AGO1]	стр. 67-68
[LENG]	стр. 341-343
[PARE]	стр. xv-xx, 1-29
[BAGL]	стр. 142-154

Край