

Физика в играх

Лекция 1

Курс 2021

План

- Процедурная анимация
- Механика движения
- Прямая кинематика
- Обратная кинематика

Что такое процедурная анимация?

Вид компьютерной анимации, который автоматически генерирует анимацию в режиме реального времени согласно установленным правилам, законам и ограничениям

Хорошо для физических симуляций: вода, частицы, дым, системы твердых тел с шарнирами и другими соединениями.

Законы физики (в анимации)

- Любое существо, висящее в пространстве, останется висеть, пока не осознает ситуацию.
- Любое движущееся тело продолжает двигаться, пока не столкнётся с чем-то твёрдым.
- Тело, проходящее через твёрдое вещество, оставляет в нём след, соответствующий его очертаниям.
- Все принципы гравитации сводятся на нет страхом.
- Одни предметы могут проходить через твёрдые поверхности, на которых нарисованы или наклеены изображения туннелей и дыр, а другие не могут.

Постановка задачи

- $x(t) \rightarrow x(t + \Delta t)$
- Нужны законы
- Нужны начальные и/или граничные условия

Ньютон + Эйлер = Простейшая модель

$$F = ma$$

F – сила, приложенная к телу

m – масса

a – ускорение

$$a = \frac{dv}{dt}$$

t – время

v – скорость

1) Известно начальное положение x_0

2) Вычислим ускорения

3) Обновим положения и скорости

$$x_i = x_{i-1} + v\Delta t$$

$$v_i = v_{i-1} + a\Delta t$$

Метод Стёрмера — Верлэ

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t + \frac{\vec{a}(t)\Delta t^2}{2} + \frac{\vec{b}(t)\Delta t^3}{6} + O(\Delta t^4),$$

$$\vec{x}(t - \Delta t) = \vec{x}(t) - \vec{v}(t)\Delta t + \frac{\vec{a}(t)\Delta t^2}{2} - \frac{\vec{b}(t)\Delta t^3}{6} + O(\Delta t^4),$$

$$\vec{x}(t + \Delta t) = 2\vec{x}(t) - \vec{x}(t - \Delta t) + \vec{a}(t)\Delta t^2 + O(\Delta t^4).$$

b is not simulated. Instead, the skeletal animation system forces the polygonal mesh of the character to orientate itself such that the knee appears to bend naturally. Since rotation around the length axis does not comprise the essential motion of the body, this works out okay and actually optimizes space.

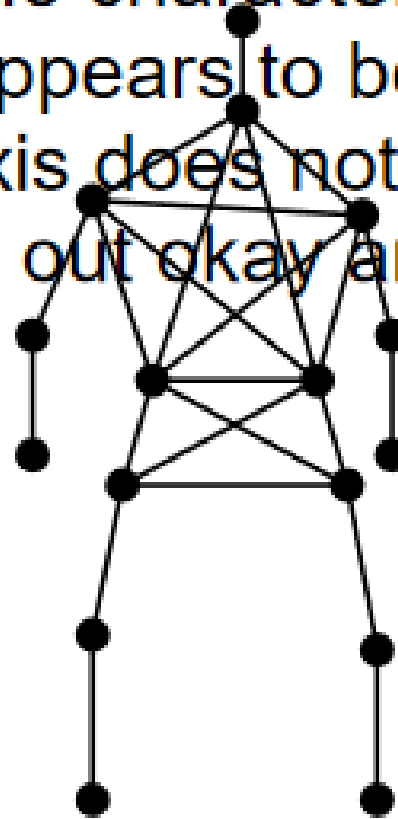


Figure 9. The particle/stick configuration used in Hitman for representing the human anatomy.

Ragdoll



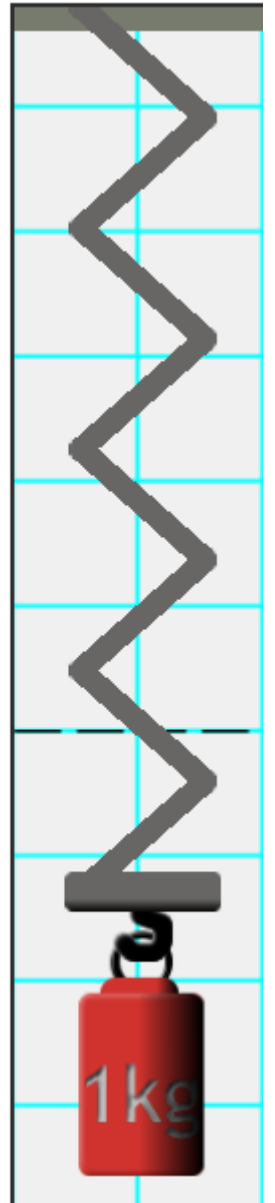
Пружина

k – жесткость

L_0 – начальная длина пружины

L – текущая длина пружины

$$F = k(L - L_0)$$



Пружина + масса

- Набор точек ненулевой массы, соединенных пружинами

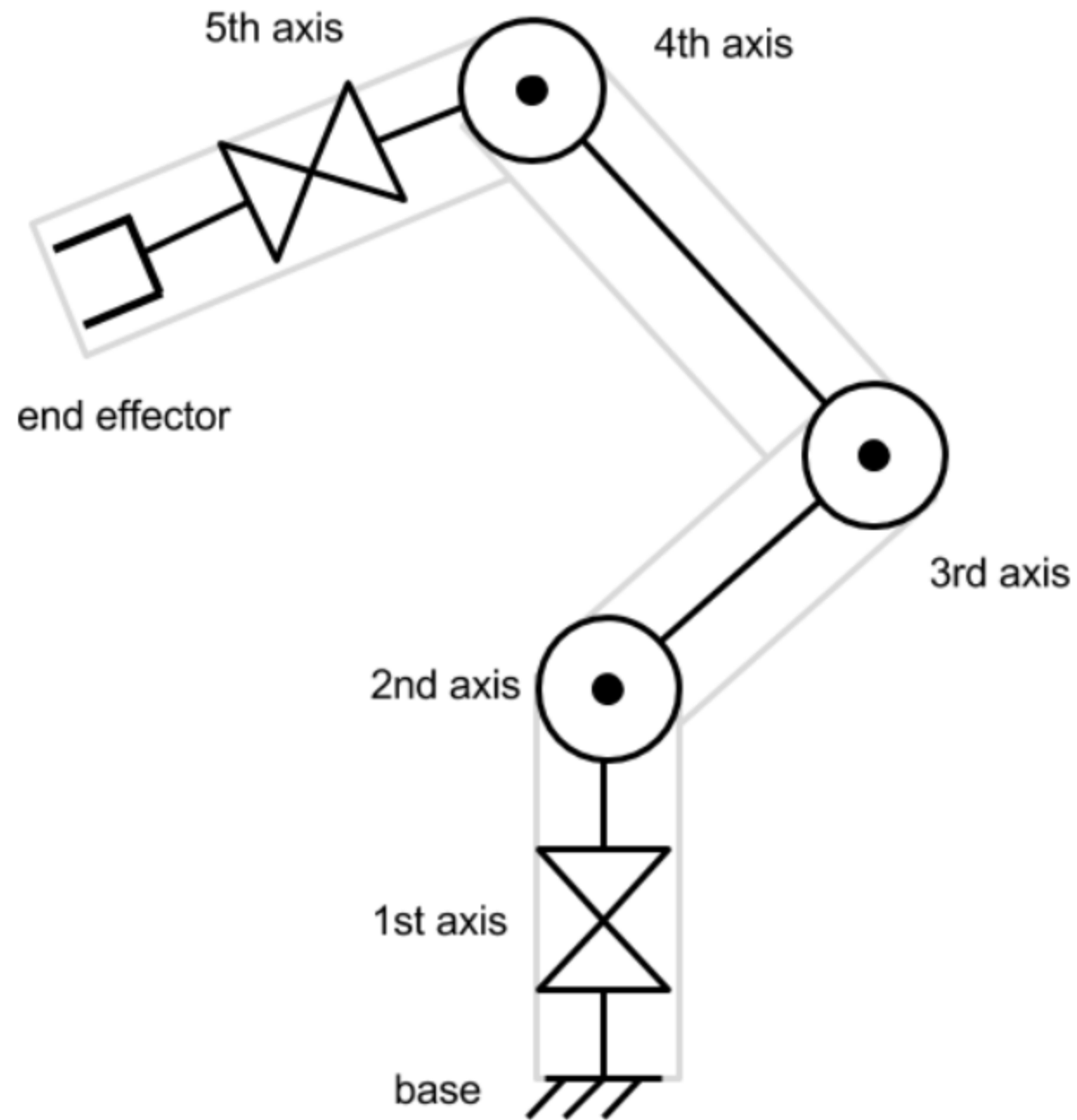
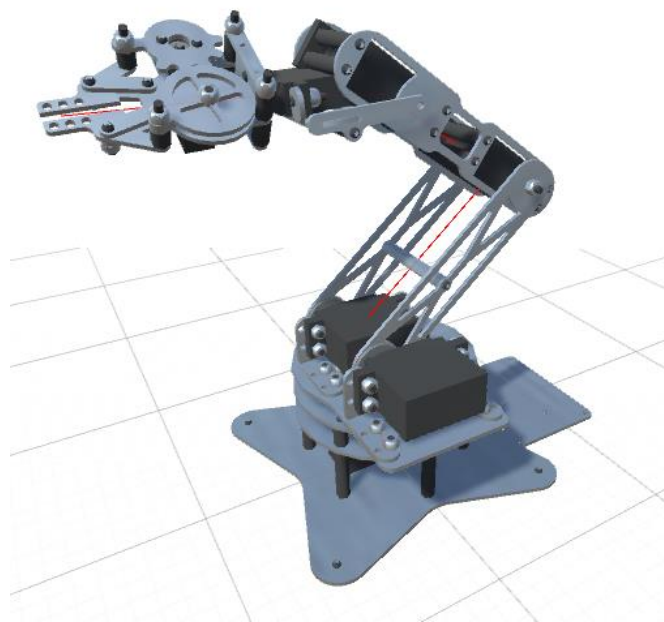
$$F_i = -F_j = k \frac{x_j - x_i}{|x_j - x_i|} (|x_j - x_i| - L_0)$$

$$F_i = -F_j = k_d (\mathbf{v}_j - \mathbf{v}_i) \frac{x_j - x_i}{|x_j - x_i|}$$

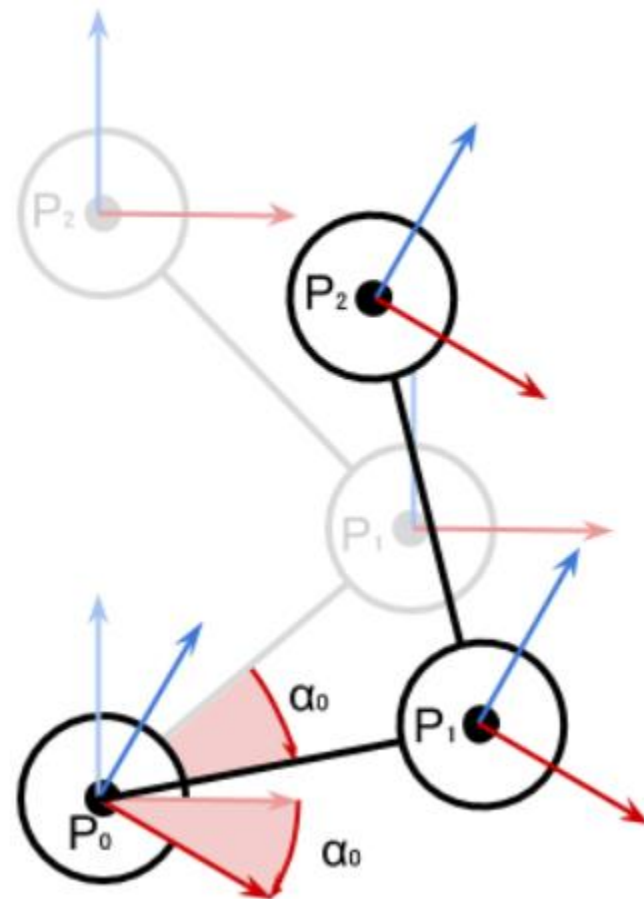
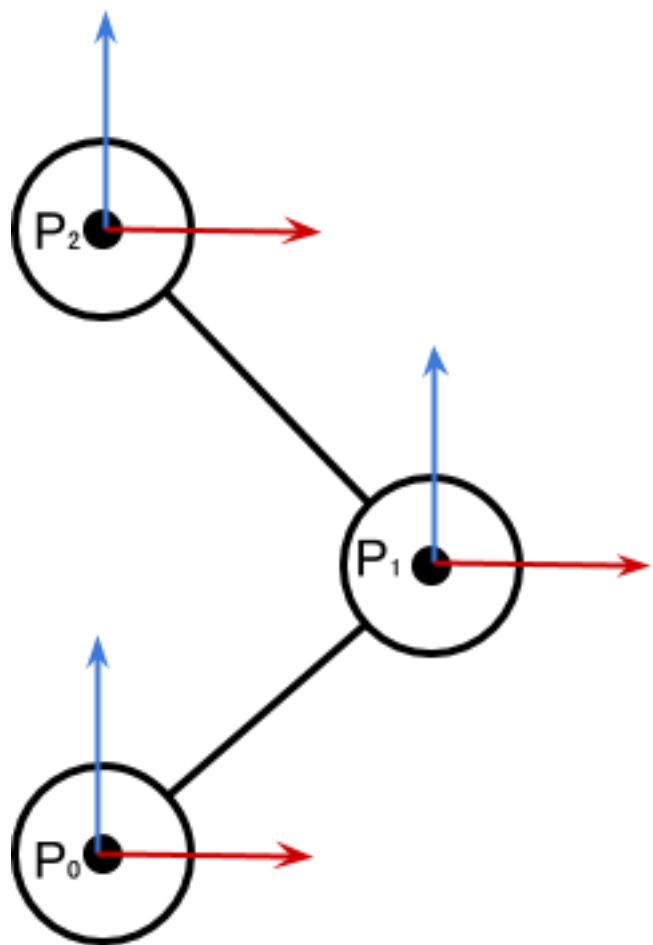
Прямая и обратная кинематика

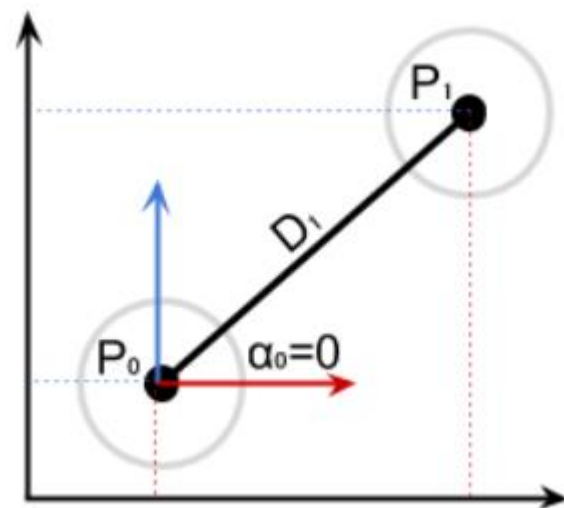
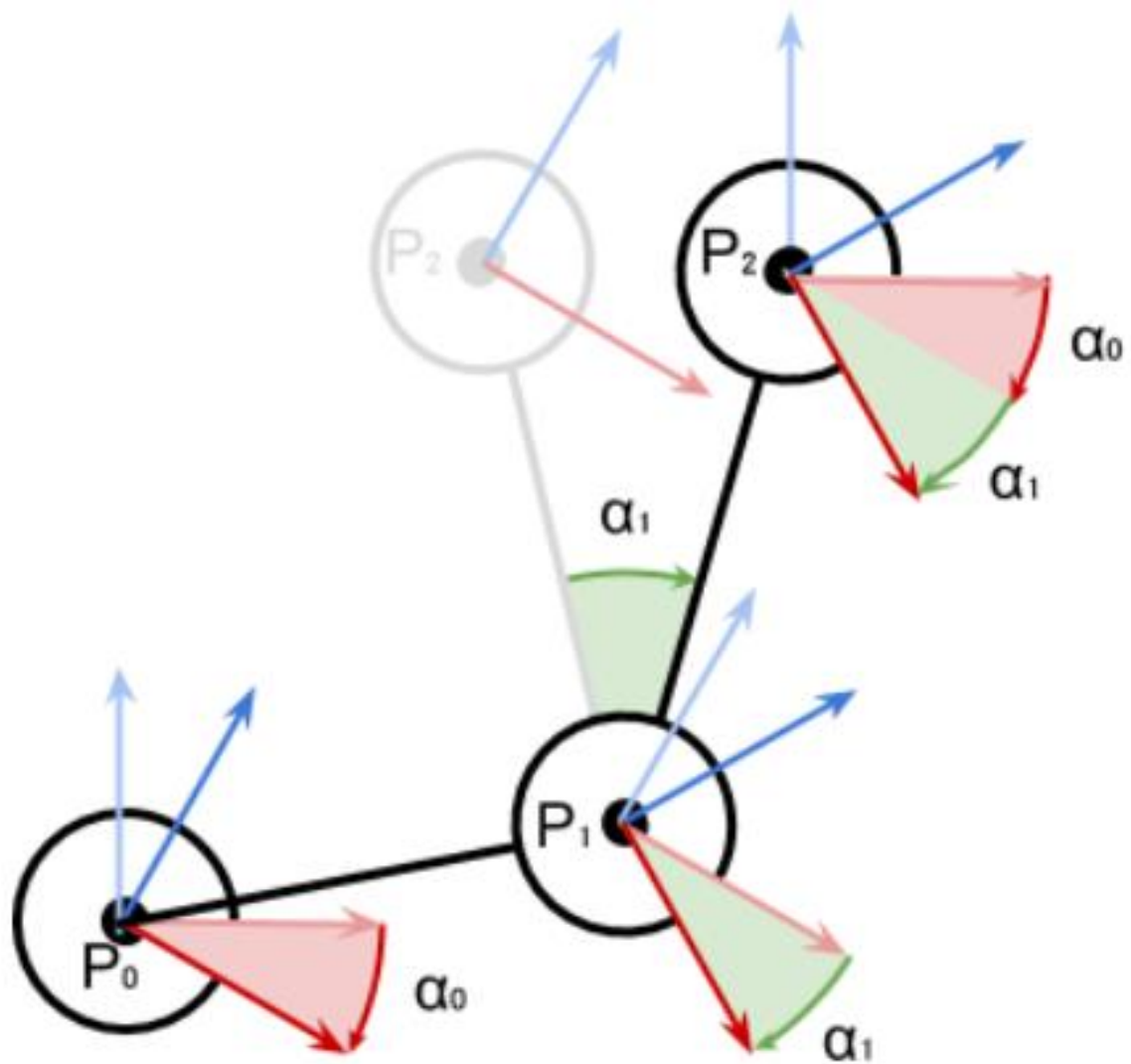
- Прямая: зная силы, найдем положения, скорости и ускорения
- Обратная: Зная траектории (позиции, скорости и ускорения) найдем силы

Прямая



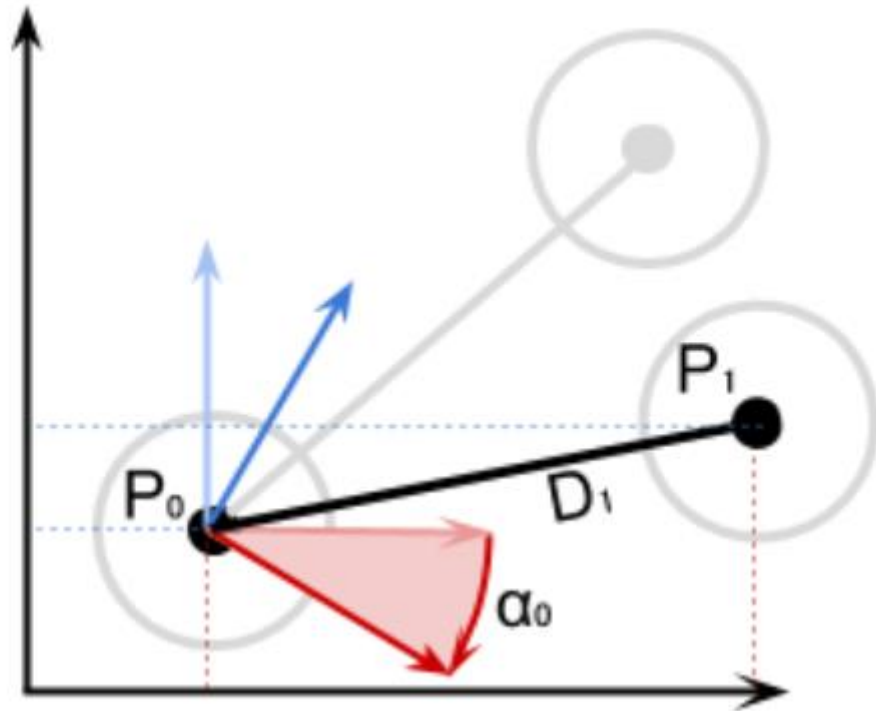
Начальное положение в 2D





$$P_1 = P_0 + D_1$$

Выполняем поворот



$$P_1 = P_0 + \text{rotate}(D_1, P_0, \alpha_0)$$

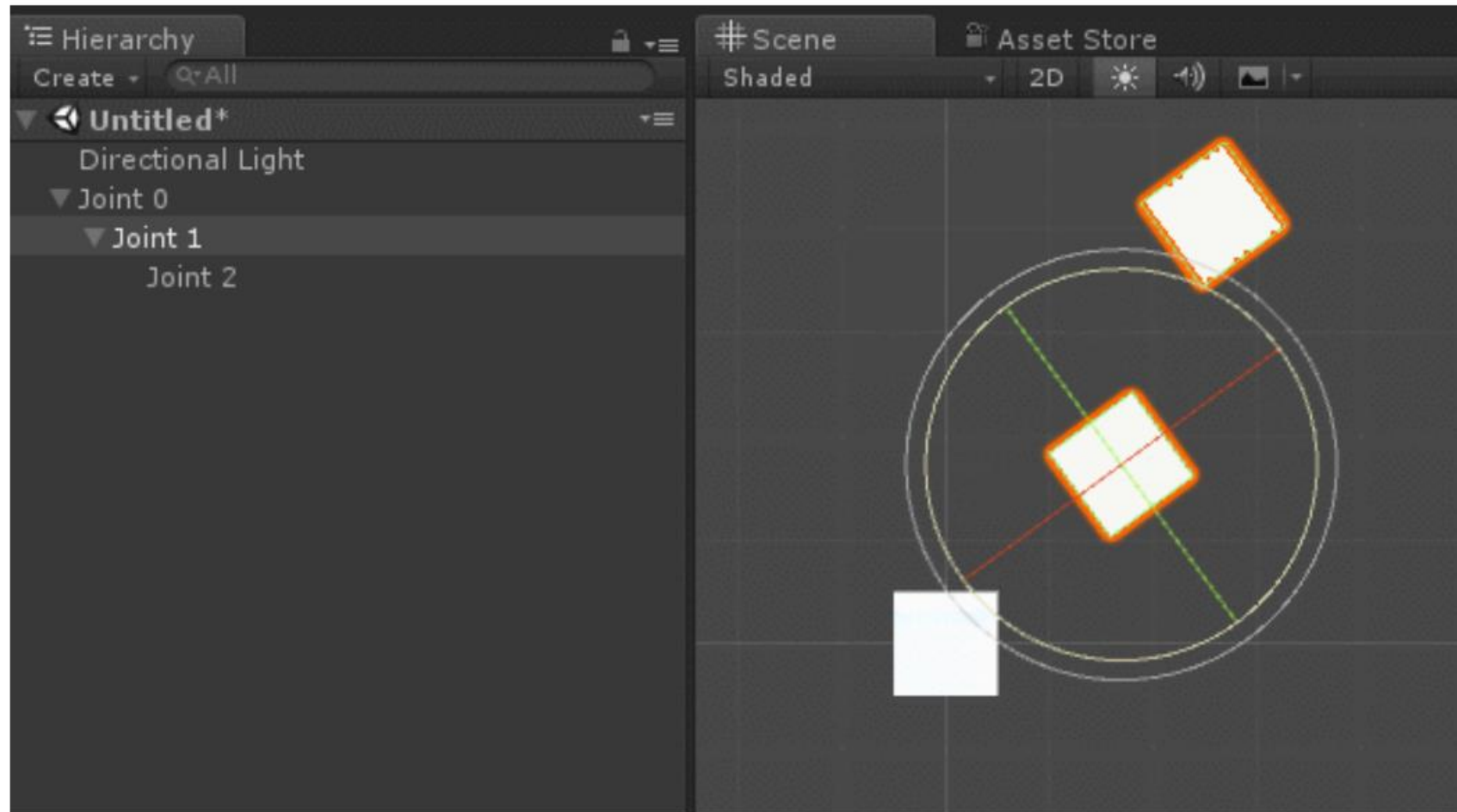
$$P_2 = P_1 + \text{rotate}(D_2, P_1, \alpha_0 + \alpha_1)$$

$$P_i = P_{i-1} + \text{rotate}\left(D_i, P_{i-1}, \sum_{k=0}^{i-1} \alpha_k\right)$$

$$P_1^x = P_0^x + D_1^x \cos \alpha_0$$

$$P_1^y = P_0^y + D_1^y \sin \alpha_0$$

Как там в Unity?



Обратный процесс

$$D = \|T - P_{effector}\|$$

Формулы нет, но D должно быть равно 0

Обратный процесс

$$D = \|T - P_{effector}\|$$

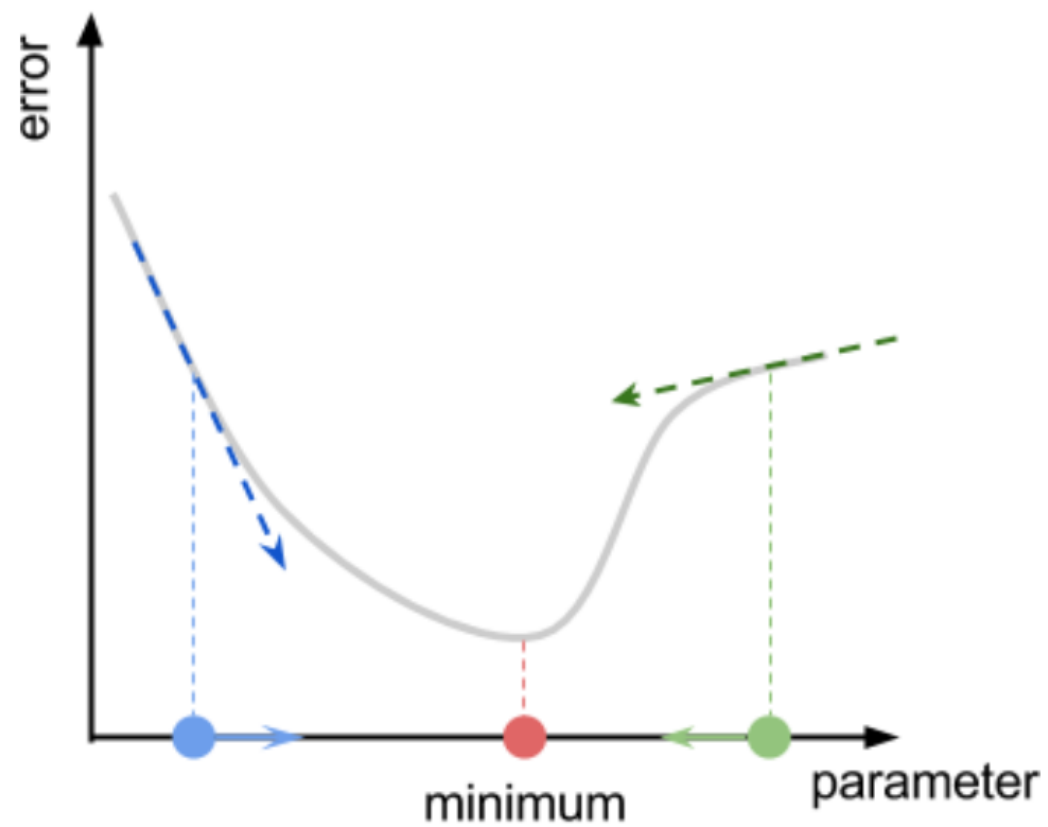
Формулы нет, но D должно быть равно 0

В смысле нет формулы?

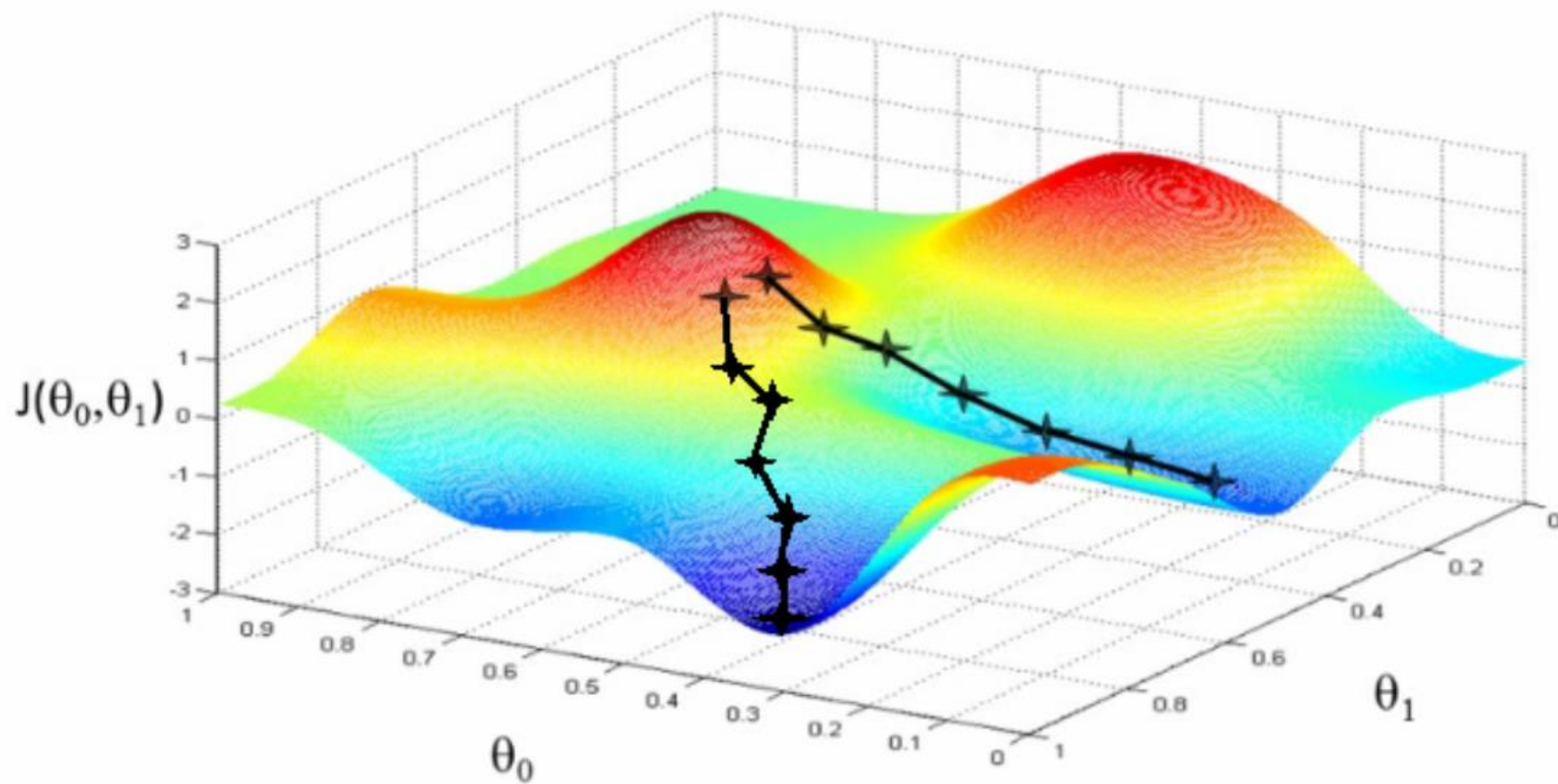
Обратные задачи

- Килограмм яблок стоит 40 рублей, а килограмм груш – 70. Мы купили 3 килограмма яблок и 2 килограмма груш. Сколько мы потратили? $120+140=260$
- Килограмм яблок стоит 40 рублей, а килограмм груш – 70. Мы купили фруктов на 260 рублей, яблок купили 3 кг. Сколько мы купили груш?
- Килограмм яблок стоит 40 рублей, а килограмм груш – 70. Мы купили фруктов на 260 рублей. Сколько мы купили яблок?

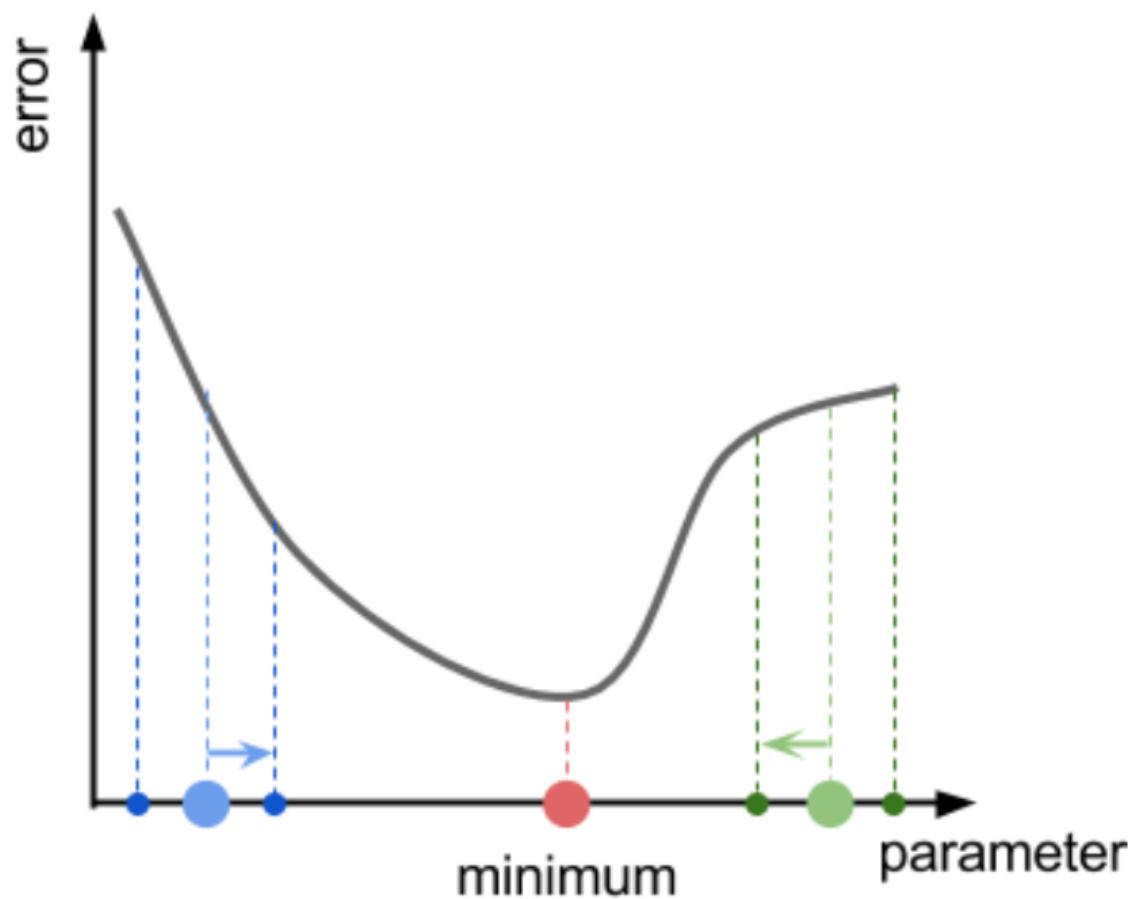
Методы оптимизации



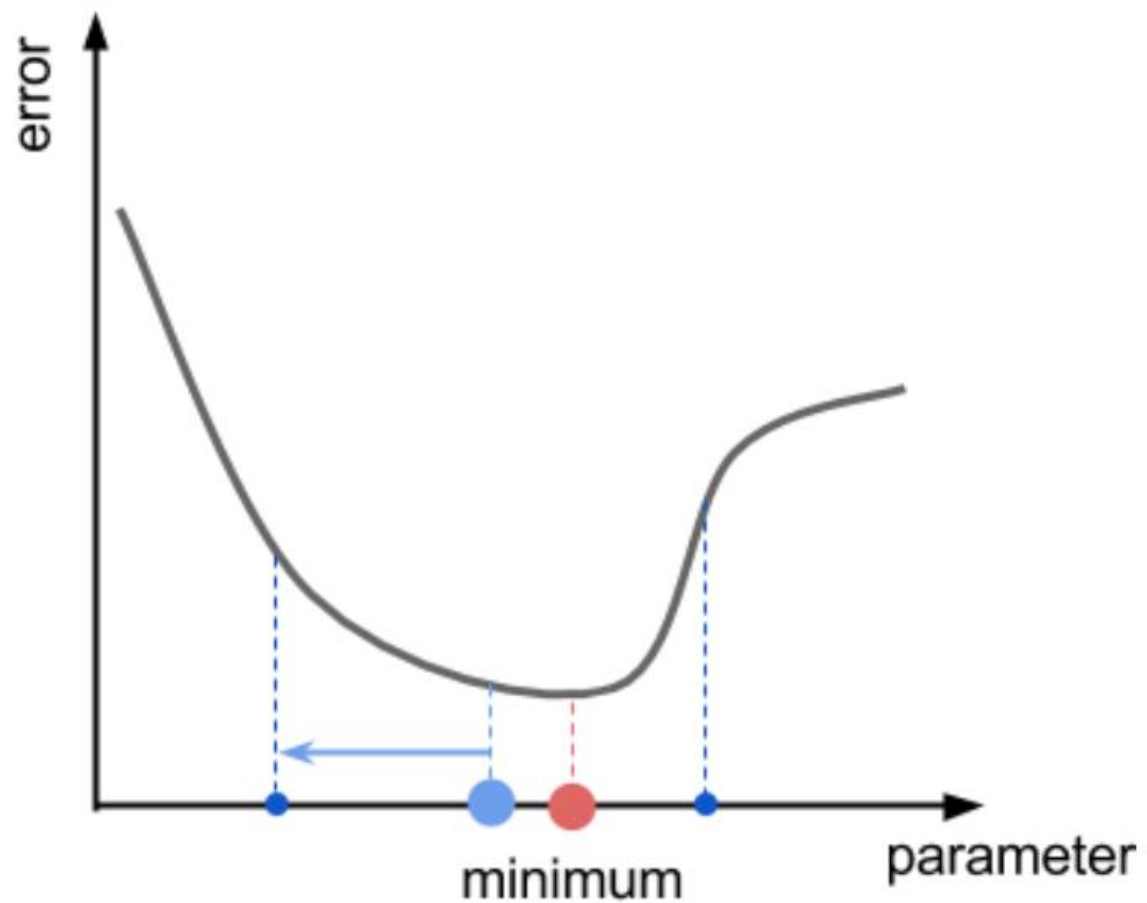
Параметров может быть много



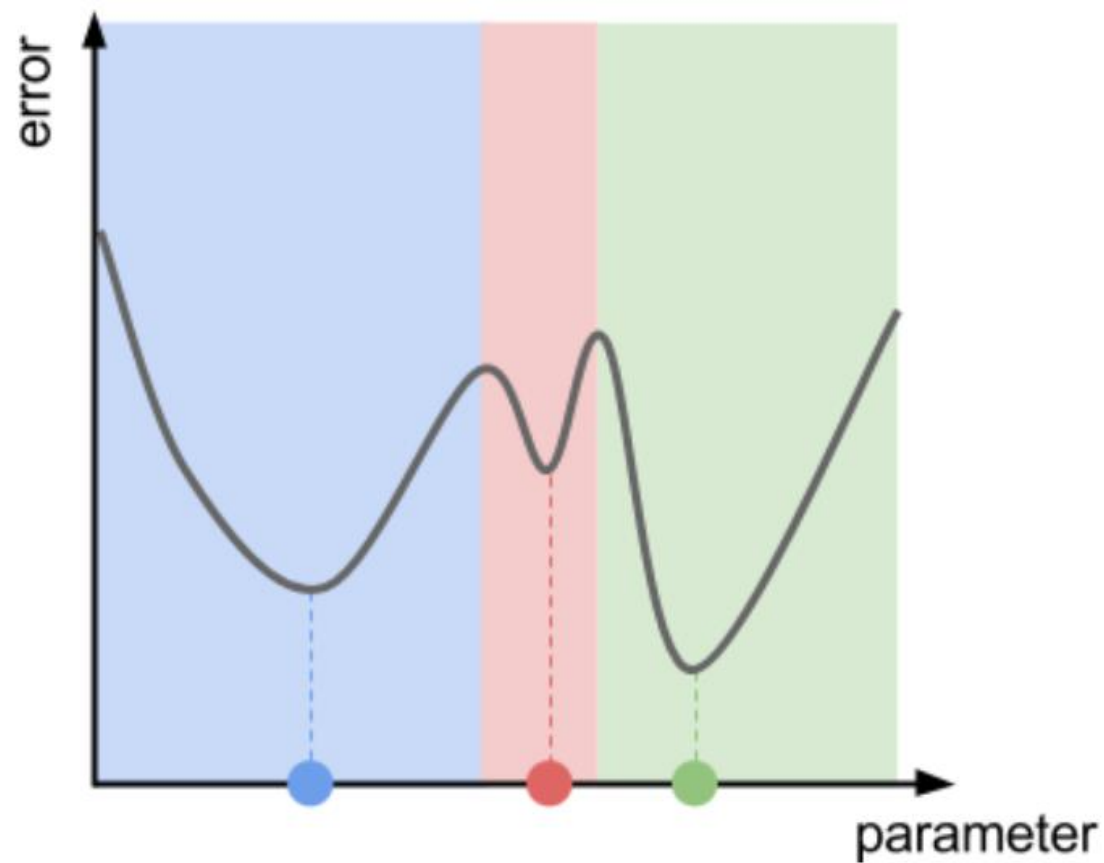
Но формул может не быть вообще



Подбор шага важен



Локальные минимумы могут быть проблемой



Производная и градиент

$$f'(p) = \lim_{\Delta x \rightarrow 0} \frac{f(p + \Delta x) - f(p)}{\Delta x}$$

$$\nabla f(p) = \frac{f(p + \Delta x) - f(p)}{\Delta x}$$

$$p_{i+1} = p_i - L \nabla f(p_i)$$

Но было несколько переменных!

$$f'_{\alpha_0}(\alpha_0, \alpha_1, \alpha_2) = \lim_{\Delta x \rightarrow 0} \frac{f(\alpha_0 + \Delta x, \alpha_1, \alpha_2) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta x}$$

$$f'_{\alpha_1}(\alpha_0, \alpha_1, \alpha_2) = \lim_{\Delta y \rightarrow 0} \frac{f(\alpha_0, \alpha_1 + \Delta y, \alpha_2) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta y}$$

$$f'_{\alpha_2}(\alpha_0, \alpha_1, \alpha_2) = \lim_{\Delta z \rightarrow 0} \frac{f(\alpha_0, \alpha_1, \alpha_2 + \Delta z) - f(\alpha_0, \alpha_1, \alpha_2)}{\Delta z}$$

Градиент в случае трех неизвестных

$$\nabla f_{\alpha_0} (\alpha_0, \alpha_1, \alpha_2) = \frac{f (\alpha_0 + \Delta x, \alpha_1, \alpha_2) - f (\alpha_0, \alpha_1, \alpha_2)}{\Delta x}$$

$$\nabla f_{\alpha_1} (\alpha_0, \alpha_1, \alpha_2) = \frac{f (\alpha_0, \alpha_1 + \Delta y, \alpha_2) - f (\alpha_0, \alpha_1, \alpha_2)}{\Delta y}$$

$$\nabla f_{\alpha_2} (\alpha_0, \alpha_1, \alpha_2) = \frac{f (\alpha_0, \alpha_1, \alpha_2 + \Delta z) - f (\alpha_0, \alpha_1, \alpha_2)}{\Delta z}$$

$$\nabla f (\alpha_0, \alpha_1, \alpha_2) = [\nabla f_{\alpha_0} (\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_1} (\alpha_0, \alpha_1, \alpha_2), \nabla f_{\alpha_2} (\alpha_0, \alpha_1, \alpha_2)]$$

Обновляем координаты

$$\alpha_0 \leftarrow \alpha_0 - L \nabla f_{\alpha_0} (\alpha_0, \alpha_1, \alpha_2)$$

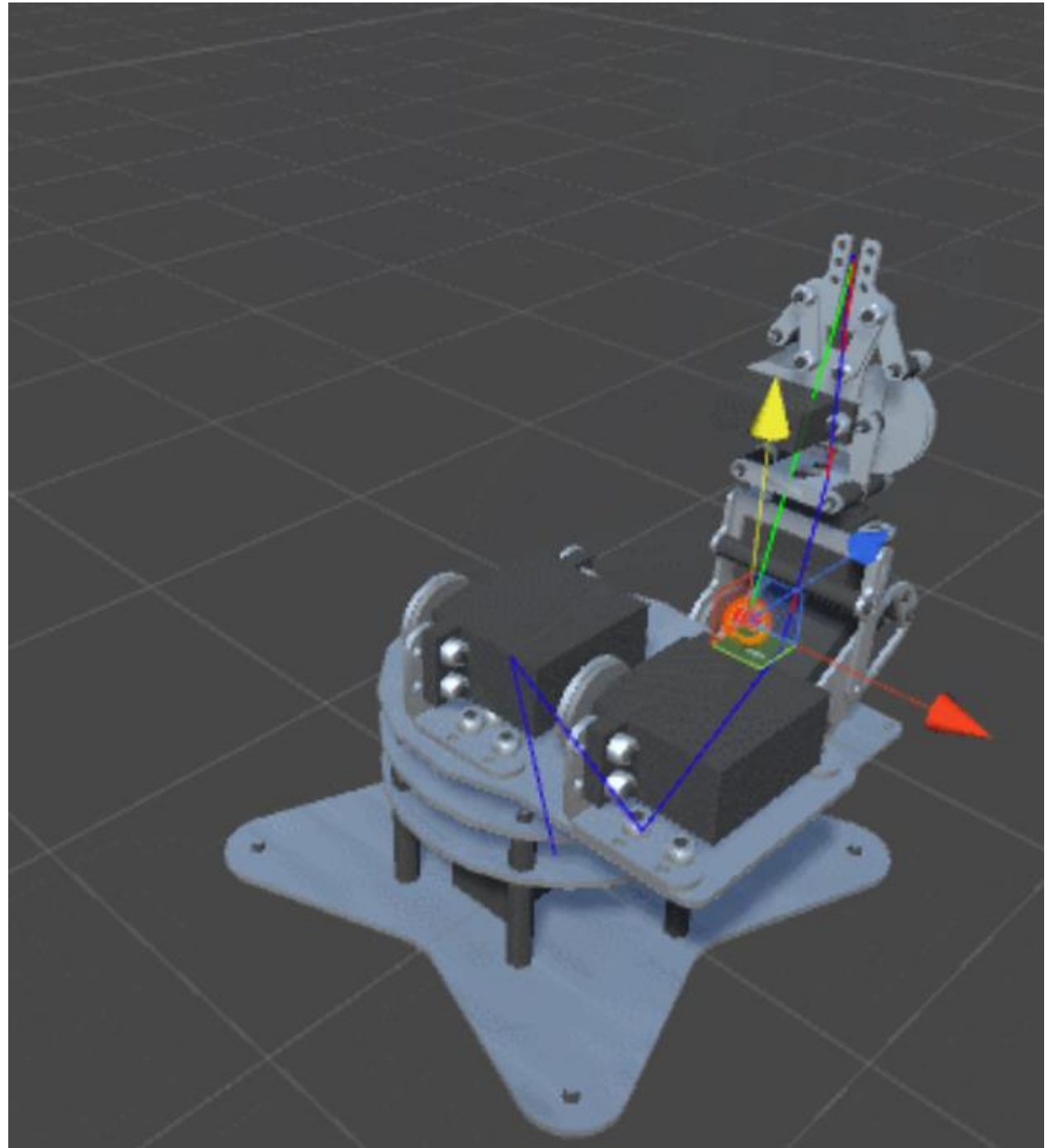
$$\alpha_1 \leftarrow \alpha_1 - L \nabla f_{\alpha_1} (\alpha_0, \alpha_1, \alpha_2)$$

$$\alpha_2 \leftarrow \alpha_2 - L \nabla f_{\alpha_2} (\alpha_0, \alpha_1, \alpha_2)$$

Что мы не учли?

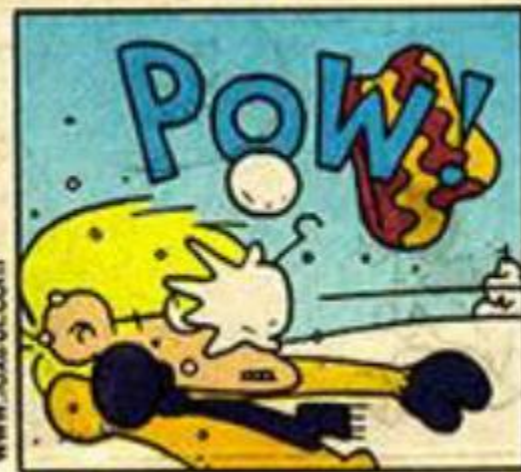
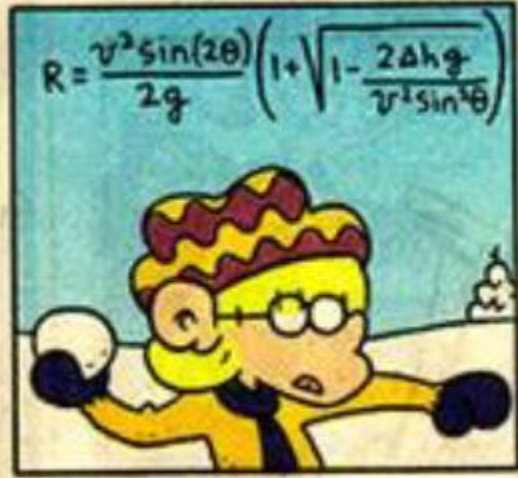
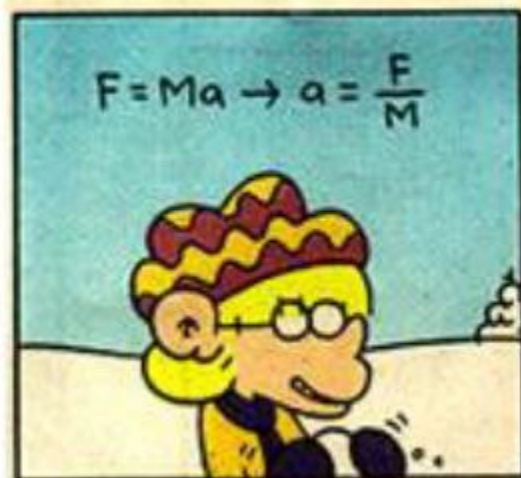
Что мы не учли?

- Границы «крутимости»
- Естественность положения
- Остановку алгоритма



Что можно сделать?

- Автоподбор шага
- Более хитрые функции ошибок



ССЫЛКИ

- <https://www.alanzucconi.com/2017/04/06/implementing-forward-kinematics/>
- <https://www.rockpapershotgun.com/how-grow-home-uses-maths-to-generate-a-personality>
- R. Featherstone, Rigid Body Dynamics Algorithms. Springer, 2008.
- <https://lear.inrialpes.fr/people/ronfard.old/cours/animation/Cours4.pdf>