

# Particles v.2 (бонусное)

Для тех, кто ещё не совсем забыл  
школьную физику

# Нужно добавить вращение

- В методе Update обновляете ещё и угловые переменные:

```
class Ball: public MaterialPoint
{
public:
    float radius;

    float angularAcceleration;
    float angularVelocity;
    float angle;
```

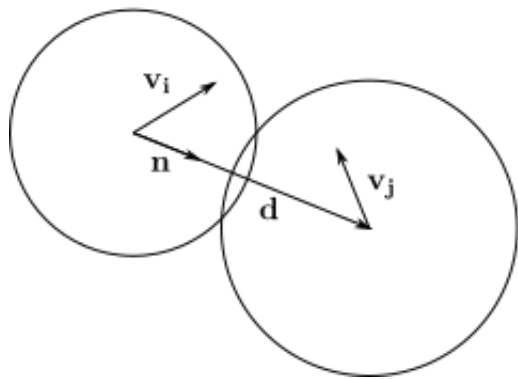
- Можно рассмотреть два случая взаимодействия: 1) шары ударяются без проскальзывания, 2) либо с проскальзыванием, когда задан некий конечный коэффициент трения  $\mu$ .

# Добавляем вращение

- Второй случай получается путем модификации первого, поэтому следует начать с него
- Круто было бы самим вывести соответствующие формулы. На следующем слайде кратко описаны идеи для вывода.
- Через слайд представлены формулы, с которыми можно свериться (p.s. знаки мог где-нибудь перепутать)

Основная идея упражнения кроется в 5 строчках формул ... у нас есть интуитивное представление того, как шары должны сталкиваться, поэтому ошибки в формулах будут проявляться уже после нескольких столкновений

# Что было, когда вращения и трения не было



1.  $|\mathbf{d}| \leq r_i + r_j$

2.  $(\mathbf{v}_j - \mathbf{v}_i) \cdot \mathbf{d} \leq 0$

3.  $\Delta \mathbf{p} = \left( 2 \left( \frac{\mathbf{v}_j - \mathbf{v}_i}{\frac{1}{m_i} + \frac{1}{m_j}} \right) \cdot \mathbf{n} \right) \mathbf{n}$

4.  $\mathbf{v}'_j = \mathbf{v}_j + \frac{\Delta \mathbf{p}}{m_j} \quad \mathbf{v}'_i = \mathbf{v}_i - \frac{\Delta \mathbf{p}}{m_i}$

Неупругое столкновение: 2 в формуле для  $\Delta \vec{p}$  заменить на  $(1 + bounceFactor)$ ,  
 $bounceFactor \in [0 \dots 1]$

Фактически, мы сказали, что у нас есть обменный импульс  $\Delta \vec{p}$ , которым обмениваются шары при ударе, что обеспечивает выполнение ЗСИ. Далее выписав ЗСЭ, предполагая, что обменный импульс направлен вдоль вектора  $\vec{n}$ , соединяющего центры шаров, получили формулу №3.

# А теперь появилось вращение

- В точка касания шаров проведем ось  $\vec{\tau}$ , перпендикулярную  $\vec{n}$ . Для выполнения законов сохранения у нас теперь будет два обменных импульса:  $\Delta\vec{p}_n = \lambda_n \vec{n}$  и  $\Delta\vec{p}_\tau = \lambda_\tau \vec{\tau}$ ,  $\Delta\vec{p} = \Delta\vec{p}_n + \Delta\vec{p}_\tau$
- Осталось записать два уравнения: 1) после удара в точка касания нет проскальзывания, т.е. проекции скоростей шаров в точке касания на ось  $\vec{\tau}$  совпадают. Отсюда находим  $\lambda_\tau$ . 2) Вдоль оси  $\vec{n}$  шары фактически обмениваются импульсами. Находим  $\lambda_n$ .

# Конечные формулы

- В первом уравнении слагаемые  $\vec{n} \cdot \vec{\tau}$  равны 0, т.к. эти векторы перпендикулярны.
- Пусть сталкиваются шары 1 и 2:

$$1/m = 1/m_1 + 1/m_2 \quad \lambda_n = \frac{(\vec{v}_2 - \vec{v}_1) \vec{n}}{1/m} (1 + bounceFactor)$$

$$\lambda_\tau = \frac{(\vec{v}_2 - \vec{v}_1) \vec{\tau} - \omega_2 r_2 - \omega_1 r_1}{1/m + r_1^2/J_1 + r_2^2/J_2}$$

P.S.  $\lambda_n$  такое же, как в случае без вращения.

# Конечные формулы

- Формула для корректировки скорости остается такой же, как в случае без вращения с учётом того, что теперь

$$\Delta \vec{p} = \lambda_n \vec{n} + \lambda_\tau \vec{\tau}$$

- Для угловой скорости

$$\omega'_1 = \omega_1 + \frac{\lambda_\tau r_1 (\vec{\tau} \times \vec{n})}{J_1}, \omega'_2 = \omega_2 + \frac{\lambda_\tau r_2 (\vec{\tau} \times \vec{n})}{J_2}$$

P.S. Мы считаем, что результат векторного произведения в 2D - скаляр

# Замечания

- В случае столкновений, когда шары не проскальзывают энергия не должна сохраняться (может только убывать), т.к. мы задаем фактически бесконечный коэф. трения.
- Проверить корректность знаков в формулах можно путем «мысленных» экспериментов, например, приводя в контакт шары, вращающиеся навстречу друг другу с одинаковой по модулю скоростью и т.п.
- Как проверить, что ф-ла верная, хоть даже и выглядит всё правдоподобно? При выводе ф-лы мы пользовались тем, что в точке контакта нет проскальзывания. Проверьте после всех корректировок скоростей и угловых скоростей, выполняется ли это условие.
- Когда вы вызываете у `sf::Sprite` метод `.rotate(45)`, то он повернётся по часовой стрелке на 45 градусов, а не против часовой. В этот момент может начаться путаница с левой и правой системами координат. Лучше не выдумывать, а вызывать метод `rotate(-angle)`, чтобы система координат осталась правой.



# Переход от бесконечного коэф. трения к конечному

- Дополнительная корректировка для  $\Delta\vec{p}_{\vec{\tau}}$  :

$$m\Delta v_{\vec{n}} = \Delta\vec{p}_{\vec{n}} = \int_0^T \vec{N} dt$$

$$\Delta\vec{p}_{\vec{\tau}} = \int_0^T \vec{f}_{\text{трения}} dt, \left| \vec{f}_{\text{трения}} \right| \leq \mu \left| \vec{N} \right|$$

$$\rightarrow \left| \Delta\vec{p}_{\vec{\tau}} \right| \leq \mu \left| \Delta\vec{p}_{\vec{n}} \right| \Leftrightarrow \left| \lambda_{\tau} \right| \leq \mu \left| \lambda_n \right|$$

P.S. знак у  $\lambda_{\tau}$  не должен измениться после этой корректировки.

# Столкновение со стенкой

- Вывести формулы, зная ответ для столкновения двух шаров очень просто: достаточно положить  $m_2$  и  $J_2$  равными бесконечности и правильно посчитать вектора  $\vec{n}$  и  $\vec{\tau}$

# Внешние силы

Можно реализовать «таскание» шаров, приложив к ним внешние силы:

- Лево́й кнопко́й мыши вы выбираете шар, и пока она зажата прикладываете силу от центра шара (O) к курсору мыши (M) пропорциональную длине отрезка OM.
- При отпускании левой кнопки мыши внешняя сила исчезает.
- Силу для отладки можно визуализировать отрезком на экране

# Успехов

