Simulazione dinamica

Stefano Lacaprara

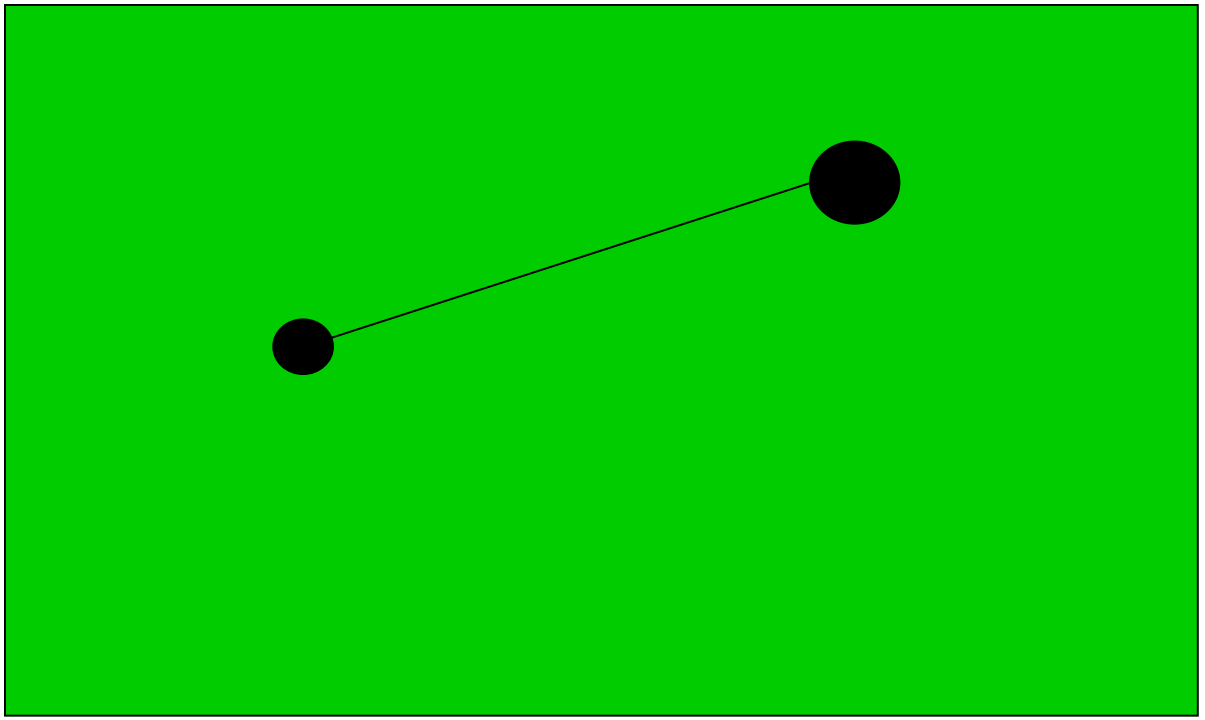
Simulazione di un sistema dinamico

- Consideriamo un sistema dinamico semplice, e proviamo a fare una simulazione del comportamento dinamico

- Idea: integrare le leggi del moto per un sistema a 2 corpi sottoposti a vincoli esterni.

Sistema

- 2 palle in una scatola, soggette a
  - forza di gravita’
  - una molla fra loro
  - rimbalzi elastici sulle pareti



Parametri

- grandezza della scatola
- massa delle palle
- costante elastica della molla
- forza di gravita'

- Per risolvere le equazioni del moto useremo una tecnica di integrazione numerica, discretizzando il tempo.
  - Dovremo scegliere bene il parametro Δt

Eq. moto

- La fisica e' semplice. Devo calcolare la risultante delle forze su ogni palla
- NB: F e a sono ***vettori***

$$\vec{F} = m * \vec{a}$$

Per la cinematica, faccio integrazione numerica

$$(t) = \frac{dv(t)}{dt} \rightarrow a(t+\Delta t) = \frac{v(t+\Delta t) - v(t)}{\Delta t}$$

Cinematica

- Per la cinematica, faccio integrazione numerica

$$a(t)=\frac{dv(t)}{dt}\rightarrow a(t+\Delta t)=\frac{v(t+\Delta t)-v(t)}{\Delta t}$$

$$v(t)=\frac{dx(t)}{dt}\rightarrow v(t+\Delta t)=\frac{x(t+\Delta t)-x(t)}{\Delta t}$$

Quindi

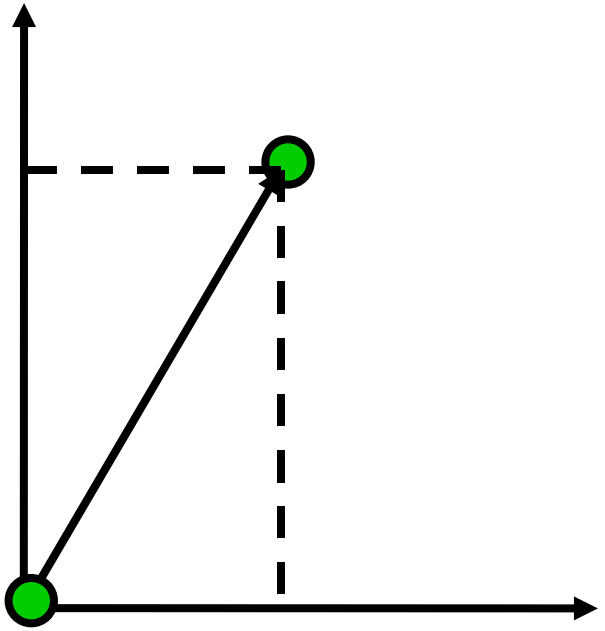
$$a(t) = \sum_{altri} F_i / m$$

$$v(t + \Delta t) = v(t) + \Delta t * a(t)$$

$$x(t + \Delta t) = x(t) + \Delta t * v(t)$$

Forze e accelerazioni

- a quali accelerazioni e' soggetta una palla:
- gravita = -g su Y
- molla = -k/m*(dist-(dist a riposo)) su (x,y)
- rimbalzi
  - semplicemente dico che invertono la componente della v corrispondente



Cosa ci serve

- Dinamica su piano: vettore 2 dimensioni
  - somma e differenza vettoriale, prodotto per scalare
- Masse:
  - conoscono posizione, velocita' (vect 2d), massa
  - sanno calcolare forza cui sono sottoposte
    - gravita' (facile)
    - molla: serve posizione altra palla
    - rimbalzi (invertire una componente velocita')
- Scatola
  - dimensioni
  - contiene le due palle
  - evolve

Vettore 2d

```
class Vec2d {
  public:
    Vec2d(float x, float y) ...
    float x() const { return theX; }
    ...
  private:
    float theX, theY;
};

Vec2d operator+(const Vec2d& a, const Vec2d& b)
{ ... }

... operator*(const float s, const Vec2d&v) { ... }

....
```

Palla

```
class Palla {
  public:
    Palla(float m, Vec2d pos. Vec2d vel) { ... }
    Vec2d pos() const ;
    ...
    void evolvi(Vec2d F); // evolvi data una forza
esterna
    void controllaBordi(float bordi ...); // controlla di
non essere arrivato al bordo

  private:
    float mass;
    Vec2d vel;
    Vec2d pos;
};
```

Scatola

```
class Scatola {
  public:
    Scatola () ...
    evolvi(); // calcola le forze su ciascuna delle due
palle e fai evolvere le palle
    display(); // mostra la scatola con palle e molla

  private:
    Palla p1,p2;
    float dimX, dimY;
};
```

Parametri

- Ci servono alcuni parametri globali
  - Usiamo namespace per proteggerli
  - accelerazione gravita'
  - k molla
  - lunghezza a riposo molla
  - Delta T discretizzazione tempo

```
namespace pars {
  const float g=9.81; // gravity acc
  const float k=1.2; // spring k
  const float l_0=5.; // spring rest lenght
  const float dt=0.1; // intervallo di tempo
}
```

Main

```
Scatola s;

for (int i=0; i<iterMax; ++i){
  s.evolvi();
  s.display();
}
```

Per visualizzare

- Uso istogramma 2d (TH2F) di root
  - Riempio isto con pos di ciascuno delle due palle, con altezza pari alla massa (cosi' vengono di colore diverso)
- Per molla uso TLine(x1,x2,y1,y2);
- Devo cancellare l'istogramma precedente ad ogni iterazione
  - Il modo piu' semplice e' creare un nuovo istogramma ad ogni step e distruggerlo dopo aver fatto DrawCopy()
- Devo lavorare su un'unica TCanvas
  - La creo io fuori del loop
  - ad ogni step chiamo Tcanvas::Update() per aggiornare la canvas

```
void Rimbalzo(int iterMax=10) {
  Scatola s;
  TCanvas * canvas = new TCanvas();
  for (int i=0; i<iterMax; ++i){
    s.evolvi();
    s.display();
    canvas->Update();
  }
  return;
}
```

```
// mostra lo stato
void display() {
  TH2F histo("fff","Tavolo",40,0,dimX,40,0,dimY);

  histo.Fill(p1().pos().x(), p1().pos().y(), p1().mass());
  histo.Fill(p2().pos().x(), p2().pos().y(), p2().mass());
  histo.DrawCopy("zcol");

  TLine theSpring(p1().pos().x(), p1().pos().y(),
                  p2().pos().x(), p2().pos().y());
  theSpring.DrawClone();
}
```