

## So You Want to Write a Physically Based Motion System

---



## Assignment #3

---

- Which is something you may wish to do since it is Assignment #3

## Assignment #3

- Write a system that will simulate a single shot of a billiards game.



## Assignment #3

- Program may be:
  - Interactive – show the animation in window on the screen
  - Batch – create an app that will produce a set of input files for a renderer.



## Dynamical System

---

- Initial value problem
  - Initial positions -- given
  - Initial velocities -- 0 (balls at rest)
  - Forces
    - Impulse force (striking of cue ball)
    - Friction forces.
    - Impulse forces (due to collisions)
      - Ball / ball
      - Ball / cushion



## Assignment #3

---

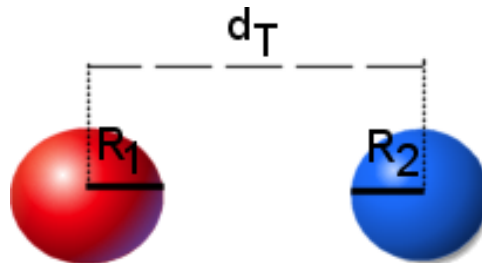
- In the file, you will be given
  - Position of the balls (3 of them + cue)
  - Impulse forces of striking the cue
    - Given as initial momentums applied to the cue ball
  - Friction & Coefficient of restitution
- Other givens (suggestions)
  - Size of table
    - 4' x 8' (1.298m x 2.438m)
    - Cushion height = 0.10m
  - Masses
    - Mass of ball = 0.5kg
  - Sizes
    - Radius of balls = 0.08m

## Assignment #3

- Simplifying assumptions
  - All balls have the same mass
  - Balls will always remain in contact with the table
    - Ignore vertical motion of balls
  - The table has no pockets
  - Collisions between balls will be totally elastic
    - All momentum is conserve

## Assignment #3

- Collision detection between spheres
  - Find distance between the 2 centers.
  - They collide if this distance is less than the sum of their two radii.

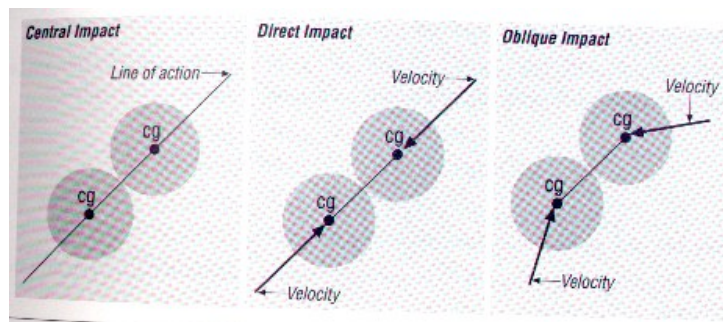


## Assignment #3

- Collision response between spheres
  - Collisions between balls will be totally elastic
    - All momentum is conserved

$$(m_1v_1 + m_2v_2)_{before} = (m_1v_1 + m_2v_2)_{after}$$

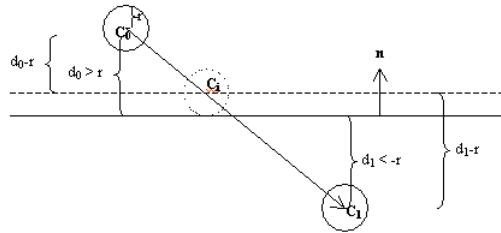
## Assignment #3



Normal is force along the line of action

## Assignment #3

- Collision between ball and cushion



## Assignment #3

- Collision between ball and cushion

$$d = (\mathbf{C} - \mathbf{p}_0) \cdot \mathbf{n}$$

where

$\mathbf{p}_0$  = any point on the plane

$\mathbf{n}$  = unit normal to the plane



## Assignment #3

---

- Cushions are immovable
  - Assume large mass

$$(m_1 v_1)_{before} = e(m_1 v_1)_{after}$$

- But in opposite direction



## Assignment #3

---

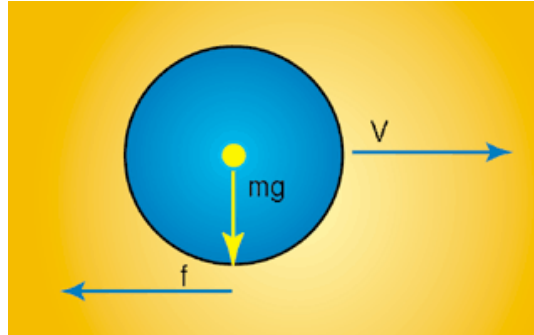
- Inertia of a sphere

$$I_{body} = \frac{2}{5} m r^2 \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

- Simple scaling in each direction

## Assignment #3

- Let's talk friction



$$F_s = u_s F_N = u_s mg$$

## Assignment #3

- Friction results in both a translational and rotational force

$$\alpha = \frac{r \times f}{I}$$

$$\alpha = \frac{r \times f}{\frac{2}{5} m R^2} = \frac{5(r \times f)}{2 m R^2}$$



## Sliding friction.

---

- Angular force causes ball to roll.
  - Eventually velocity of the point contacting the table equals the velocity of the center of mass.
  - At that point the ball stops sliding and starts rolling.
  - This happens when:

$$v = r\omega$$

- Or when

$$\Delta t = \frac{2v}{7\mu_s g}$$



## Rolling friction

---

- When ball starts to roll, the rolling friction kicks in.
- Force applied that stops the roll.
- Same idea as sliding friction
  - Different friction constant

$$F_r = \mu_r F_N = \mu_s mg$$



## Framework

---

- Step 1
  - Calculate Forces,  $F(t)$ ,  $\tau(t)$  – only friction here!
- Step 2
  - Integrate position/rotation
    - $s(t + \Delta t) = s(t) + v(t)\Delta t$
    - $q(t + \Delta t) = q(t) + 0.5 (\omega(t)q(t)) \Delta t$  /\* normalize to avoid problems \*/
    - $R(t + \Delta t) = \text{quatToRot}(q(t + \Delta t))$
    - $r(t + \Delta t) = s(t + \Delta t) + r_{\text{body}}R(t + \Delta t)$
  - Update Momentum (integrate forces)
    - $M(t + \Delta t) = M(t) + F(t) \Delta t$
    - $L(t + \Delta t) = L(t) + \tau(t) \Delta t$



## Framework

---

- Step 2.5
  - Collision detection / determination / response.
- Step 3
  - Calculate velocities (for next step)
    - $v(t + \Delta t) = M(t + \Delta t)/m + \text{impulse velocity}$
    - $I^{-1}(t + \Delta t) = R(t + \Delta t)I_{\text{body}}^{-1}R(t + \Delta t)^T$
    - $\omega(t + \Delta t) = I^{-1}(t + \Delta t)L(t + \Delta t) + \text{impulse velocity}$
  - Go to step 1
- Questions?



## Assignment #3

---

- Infile – Describes position of balls and initial forces
  - Very simple formatted text file
    - Just a suggestion -- can use your own format.
  - All values will be given in metric units.
    - First 4 lines – ball positions
      - R G B x y z
        - (R, G, B) is color of the ball
        - (x,y,z) is position of ball
        - 1<sup>st</sup> ball is assumed to be the cue ball (the ball to which force is to be applied)



## Assignment #3

---

- Infile – Describes position of balls and initial forces
  - Very simple formatted text file
    - Line 5
      - X Y Z x y z
        - (X, Y, Z) = linear momentum to apply to cue
        - (x,y,z) = initial rotational momentum
          - = vector gives axis of rotation
          - = magnitude gives angle



## Assignment #3

---

- Infile – Describes position of balls and initial forces
  - Very simple formatted text file
    - Line 6
      - $u_s e$ 
        - $u_s$  = friction constant
        - $e$  = Coefficient of restitution (for cushion collisions)



## Questions?

---



## Assignments

---

- Grading
  - Each assignment is worth 20 points:
    - 5 points – for something that compiles
    - 15 points – for something that runs incorrectly
    - 20 points – for something that runs correctly
    - Bonus points for extras



## Bells and Whistles

---

- For 20 points
  - Only consider linear motion w/friction
  - Euler integration
- Extras (*max 10 points / assignment*)
  - Add pockets (5 points)
  - Consider rotational motion (10 points)
    - Basic rotation (5 points)
    - Account for rotation resultant from friction (5 points)
  - Implement Runge-Kutta Integration
    - 2<sup>nd</sup> order / midpoint (5 points)
    - 4<sup>th</sup> order (5 points)



## Reference

---

- Gamasutra Article
  - [http://www.gamasutra.com/features/20000516/lander\\_01.htm](http://www.gamasutra.com/features/20000516/lander_01.htm)



## Due dates

---

- Assignment Due
  - Monday, January 26<sup>th</sup>
- Submission
  - Via mycourses
  - Please include documentation on
    - how to run your app
    - How to build your app
    - Renderer used if batch
    - Platform (sun vs. Windows vs Mac)
  - Please include sample input file.



## Questions?

---

- Next time:
  - Dynamics applications: Particle Systems.
  - Questions?