0616034 游茜雯

- Introduction/Motivation

  用 Visual Studio 和 C++實作：物體的碰撞偵測和處理、spring damper system、distance constraint、最後要讓物體動起來（integrator）。

- Fundamentals
  - 碰撞偵測：
    - 是否碰撞
    - 碰撞後的 penetration
    - 碰撞後的 normal
  - 碰撞後處理：
    - 碰撞後兩物體分開
    - 摩擦力
  - spring damper system：
    - 彈簧力
    - 阻尼力
  - distance constraint：
    - 兩物體間需要小於一定距離
  - integrator
    - Euler’s method
    - RK4

- Implementation
  - 碰撞偵測：

    由穿刺發生與否判斷是否碰撞。

penetration：overlap，根據碰撞發生的雙方形狀而有算法的不同。

normal：由 body0 指向 body1 的單位向量，與 overlap 同向。

- 碰撞後處理：

  計算衝量，並根據雙方的質量倒數分配要加減的速度。

  摩擦力亦同上。

- spring damper system：

  參照上課講義。

- distance constraint：

  參照助教 PPT 給的連結。

- integrator

  參照上課講義。

Result and Discussion

- Problem unsolved
  - spring damper system

    按照上課講義給定 force，但不知道為什麼跑一兩秒後速度就會變 nan，然後 objects 就會消失。

    spring force、damper force 都是按照講義計算，猜想應該是 integrator 中速度不能直接加上 deltaTime*force*InvMass，但將這部份去掉，並在 Euler's method 加上一個二階項，這樣則是跑了久一些，但最後結果一樣。

- the difference between Explicit Euler and RK4

  將 deltaTime 設成比較大的數字時（Ex. 10.0f/1000.0f），就能大概看出用 Explicit Euler method 計算出來的 position 讓 distance constraint 物體的移動看起來比較不像自然移動。

但當 deltaTime 設成比較小的數字時（Ex. 0.5f/1000.0f），用 RK4 的方式會讓畫面不太順暢，但用 Explicit Euler method 就不會有這個情況。

- Conclusion

在矩形與圓形發生碰撞時，如果圓心的 x 座標或 y 座標剛好在矩形兩 x 邊或兩 y 邊的範圍內，則 normal 的設定只會是垂直或水平方向。因此在這樣情況之下的矩形與圓形，就算看起來應該要沿著下方者的上邊落下，也只會靜止於下方者之上。

在位置與時間關係並非線性的情況下，RK4 計算出的位置通常會比 Euler's method 要來得準確，但同時運算量也是 Euler's method 的四倍多。同時，deltaTime 越小也會越精準，但相同時間內的計算量也會增加很多。