Assignment 1 Report

A022029 邱政凱

這次的作業要求我們用 Particle System 去模擬一個足球網的物理。其實以前完全沒有寫過物理模擬的程式，原本以為會很麻煩，不過助教已經把整個 Assignment 大致上的架構給我們規畫好了，用 OpenGL 去 Render 的步驟也不需要我們親自寫，所以這次作業基本上就是專心在處理物理模擬的部分。

這次的 Particle System 的架構大致上是由主程式呼叫 Mass Spring System 裡的 Integration 函式，並且 Compute Force、Handle Collision，然後依照不同的 ODE 處理方式來更新系統中的 Particles 和 Balls 的位置、速度。

實作的部分，首先是積分器，Explicit Euler 的寫法非常直覺單純，就只是把速度跟加速度取出來乘上 delta Time 加到位置和速度上。我們不用再去把受力換算成加速度，原因是因為 Ball 和 CParticle 對於加速度和受力的表示是同樣的，也就是說在前面 compute force 的部分就處理過了，所以直接把加速度取出來用就好。不過 Runge-Kutta 倒是折磨了我好一陣子，只是看公式會覺得乍看之下是很好理解的公式，不過實際上實作卻有點麻煩，每一次積分都需要預先計算四次的物理計算，再把四次算出來的加速度和速度依權重重新加回一開始的位置和速度上。(我一開始還想說為什麼在 Integration 函數中呼叫 RungeKutta 前面不用 ComputeForce，原來是因為要 Compute 不只一次)。

接著是連接彈簧，其實就很單純地依照三種彈簧不同的連接方式把 Particle 和相鄰或隔兩個的 Particle 接在一起就好了。比較需要注意的地方像是足球網邊界狀況的處理，因為如果是直接用 GetParticleID()從頭到尾接下去的話會遇到一排的最後一個接到下一排的第一個的狀況，所以要作一些限制。

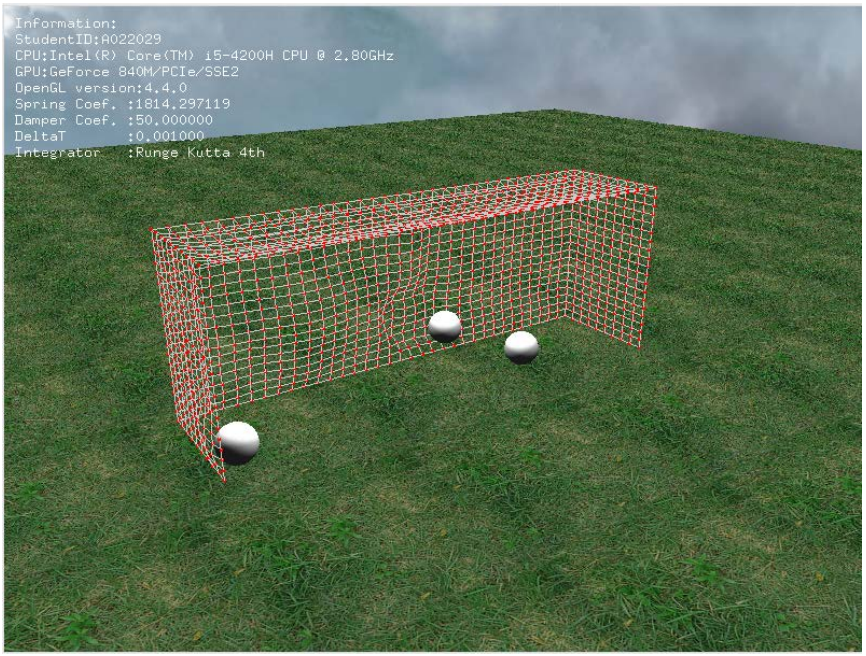
計算 Internal Force 則就是把每個彈簧取出來，依照講義上的公式計算其中一個物體的受力，並乘上-1 給另外一個物體就好了。

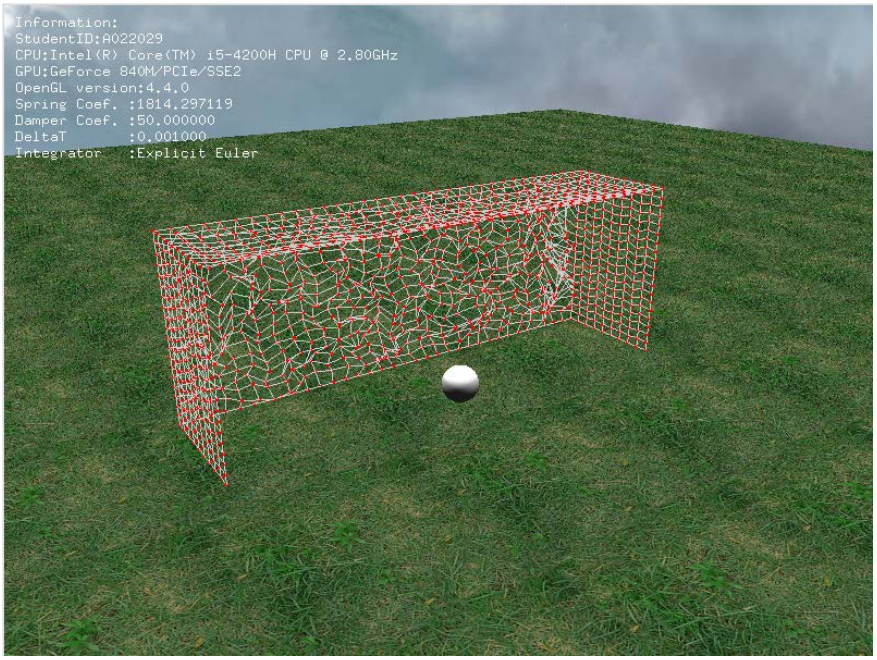
碰撞偵測大部分都也是照著講義上的公式來做就好了，除了球跟球碰撞和球跟點碰撞，我一開始一直只把碰撞條件設為兩者距離小於球的半徑，而沒有加上兩者必須是靠近中的狀態，結果造成球打到網子上後一直不會彈回來。

整體而言，實作上最需要注意的部分就是小心看懂公式並且理解 Vector3d 的運作方式，然後小心的用 code 把公式打出來，並考慮各種狀況。

至於一些值得一提的結果比較，首先就是 Explicit Euler 跟 RungeKutta 的結果比較。RungeKutta 的誤差是 h^5，而 Explicit Euler 是 h^2，很明顯的，RungeKutta 的結果會比 Explicit Euler 來的自然，這點從模擬開始時還沒發射足球前，足球網上面的網子受重力垂吊下來的時擺盪的動畫就有差異了，Explicit Euler 的 Particle 會晃動得比較大，動作也比較死板，然而 RungeKutta 的 Particle 動作就比較滑順一些，動作也不會太大。另外更關鍵的一點是，藉由調整一些參數(主要是 Spring Coef 和 Damper Coef)，可以發現 Explicit Euler 在很多狀況下

都會在模擬開始之後沒多久進入 unstable 的狀態，然後整個畫面就炸掉，然而 RungeKutta 在很多 Explicit Euler 會爆掉的參數設定下卻能正常地進行模擬(儘管一些極端的參數還是會讓它爆掉)，顯示 RungeKutta 能夠比較穩定的解決微分方程的問題，算出來的答案曲線也比較平滑(儘管計算時間是 Explicit Euler 的數倍)。





上圖： RungeKutta 把 Spring Coef 調到 1800 後用球去撞球網。
下圖: Explicit Euler 把 Spring Coef 調到 1800 後用球去撞球網

另外就是一些參數的觀察，像是把 Spring Coef 調低的話整個球網看起來就

會軟趴趴的，球一丟上去感覺網子好像就要破掉似的。Damper Coef 也對球網的行為有很大的影響，低的 Damper Force 會讓整個球網每次被打到時都有很大的震動幅度，而很高的 Damper Force 則會很容易讓整個系統馬上陷入 Unstable 的狀態(這點跟講義上的敘述吻合)。至於 deltaTime 的影響則比較難觀察到，不過還是可以稍微觀察到小一點的 deltaTime 的球網擺動的動作會滑順一點。還有就是各種碰撞的 epsilon 的取值也很關鍵，不只是對地板的碰撞，球跟點之間的碰撞偵測依據我的觀察不同的 epsilon 值會讓模擬結果整個截然不同。像是球跟球網的碰撞，如果沒有一個適當的 epsilon 值的話，很可能球會穿過球網呢。

結論而言，這次的作業讓我稍微了解了 Particle System 的運作基礎。儘管裡面使用到的數學和物理知識都是早就已經知道很久的，但是實際上要用程式讓一個虛擬的系統正常的運作起來，實在是要耗費超乎我想像的功夫，但卻也覺得非常的有趣和充滿挑戰。