

컴퓨터애니메이션실습(CAL)

HW11_Particle System Dynamics



Self-scoring table

	P1	P2	E1	Total
Score	1/1	1/1	1/1	3/3

2018707068 김경환

KwangWoon University

Code Analysis:

먼저 이번에는 external force fields인 Physical law의 영향을 받는 Point masses의 모음을 동적으로 계산하고 rendering하는 Particle System Dynamics를 수행한다.

해당 과제에서는 external force fields로 중력만을 부여하였고, 4개의 벽을 구성하여 이에 대한 Collision Handling을 수행하였다.

그리고 Particle을 rendering할 때 시간 t 에 대한 position과 velocity를 계산해야하는데, 이는 상미분방정식을 사용하여 적분을 수행해 구해낸다.

$F = ma$ 에 대한 1차 미분방정식으로만 표현된 식을 구해보면 Position의 미분은 Velocity, Velocity의 미분 Acceleration은 force / mass로 이루어진 두 개의 식을 구할 수 있다.

여기서 가장 간단한 Euler's method는 위 식에 따라 현재의 position과 velocity를 사용하여 time step 다음의 position, velocity를 계산해낸다. 그리고 Modified Euler's method는 먼저 현재의 velocity를 사용하여 다음 time step의 velocity를 구하고, 이를 반영하여 다음 time step의 position을 계산한다. 마지막으로 exercise에서 수행한 Midpoint method는 먼저 Euler step을 계산한 다음 그것을 통해 f 의 midpoint를 계산하고 이를 사용하여 다음 time step의 position을 계산한다.

위 세가지 method에서 Euler's method는 Sub steps의 개수에 의존적이므로 Sub steps의 개수가 적을 때는 만족스럽지 않은 모습을 보인다. 하지만 Modified Euler's method와 Midpoint method는 그에 비해 Sub steps의 개수에 덜 영향을 받아 Sub steps의 개수가 적어도 상대적으로 만족스러운 모습을 보인다.

이제 구현에 대한 내용으로는 먼저 모든 Particle의 위치를 매 실행마다 같은 random 값이 되도록 random의 seed를 같은 값으로 주고 수행한다.

그리고 해당 System은 전부 xy평면에서 이루어지므로 position과 velocity 모두 z값은 0으로 두고 수행한다. 또한 color는 음수가 되지 않도록 random한 rgb값에 (1, 1, 1)을 더해준 뒤 2로 나누고 normalize해준다. xy평면에 대해 벽이 이루어지므로 normal vector는 이에 맞게 설정한 뒤 collisionHandling을 수행할 때 바뀌는 window의 aspect에 따라 벽의 position을 설정해준다.

벽에 대한 collision은 벽의 position에서 particle의 position으로 향하는 vector와 벽의 normal vector를 dot product하여 distance를 얻어 판별한다.

만약 이렇게 구한 distance가 particle의 radius와 임의로 준 epsilon을 합한 값보다 작다면 particle이 벽을 통과한 상태이므로 particle의 position을 벽의 normal 방향으로 radius와 epsilon을 더한 값에서 distance를 뺀만큼 이동시켜준다.

그리고 벽의 normal vector와 particle의 속도 vector를 dot product한 값의 부호를 따져 벽 안으로 들어오는 것에 대해서만 collision response를 수행한다.

여기서 collision response를 계산할 때는 velocity를 tangent 성분과 normal 성분으로 나눠 normal 성분에 restitution 계수를 곱해 계산을 진행한다.

마지막으로 매 frame마다 해당 physical law를 반영하여 update를 수행할 때는 상미분방정식을 풀어야 한다. 이러한 상미분방정식을 풀 때 1/120s를 Sub step의 개수만큼 나눈 값을 time step으로 사용함으로써 계산의 오차를 줄일 수 있다.

Modified Euler's method는 Euler's method와 달리 먼저 현재의 velocity를 사용하여 다음 time step의 velocity를 구하고, 이를 반영하여 다음 time step의 position을 계산하기 때문에 Euler's method보다 만족스러운 결과를 보이는 것을 확인할 수 있다.

그리고 Modified Euler's method는 Sub step의 개수가 1일 때에도 상당히 만족스러운 결과를 보여주기 때문에 Sub step의 개수를 키워도 큰 차이를 육안으로 확인하긴 쉽지 않다.

Exercise 01. Time integration with the midpoint method:



Sub step의 개수로 1을 주고 Midpoint method을 수행했을 때의 모습이다.

Midpoint method는 먼저 Euler step을 계산하고 이러한 Euler step의 midpoint에서의 velocity를 구해낸다. 그리고 이러한 midpoint의 값을 사용하여 다음 time step의 position을 구한다.

위와 같이 Midpoint는 먼저 Euler step의 midpoint에서의 velocity를 구해내고 이를 반영하여 다음 time step의 position을 구하기 때문에 Modified Euler's method처럼 보다 만족스러운 결과를 보이는 것을 확인할 수 있다.

또한 Modified Euler's method처럼 Sub step의 개수가 1일 때에도 상당히 만족스러운 결과를 보여주기 때문에 Sub step의 개수를 키워도 큰 차이를 육안으로 확인하긴 쉽지 않다.