

컴퓨터 애니메이션

과제 3

| | report | create | attach | nail | drag | Total |
|-------|------------------|-------------------|----------|--------|---------|-------|
| Score | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | 10/10 |
| | point damping | damping spring | unstable | stable | falling | |
| Score | 1/1 | 1/1 | 1/1 | 1/1 | 1/1 | |

제 출 일: 2020년 6월 19일

소 속: 수학과

학 번: 2016603009

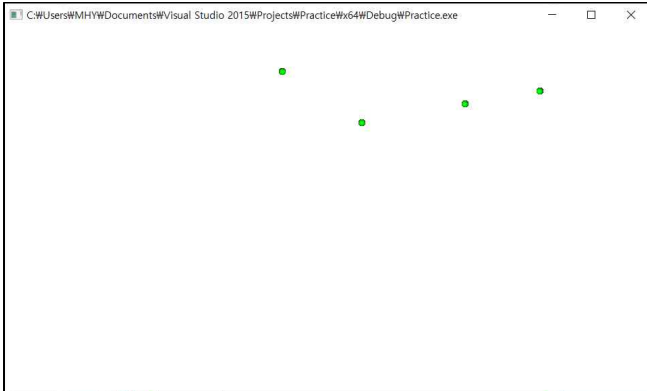
이 름: 문 하 영

담당교수: 최 민 규

Programming Assignment #3

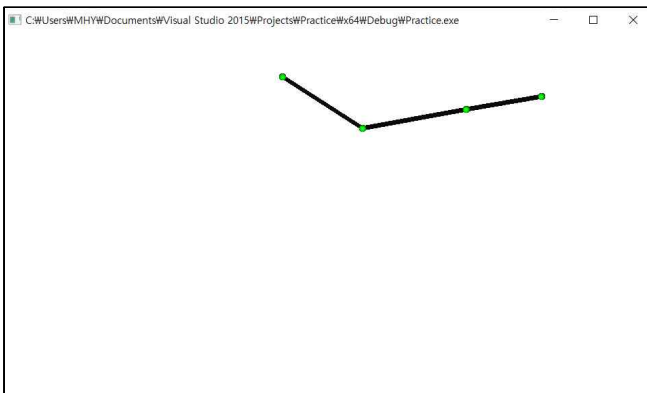
실습 자료 p10_mass_spring_model.cpp를 바탕으로 구현하였습니다.

1) Create particles



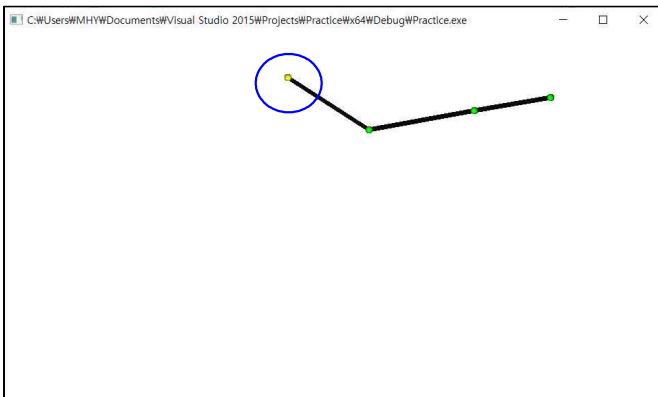
c키를 누르고, 마우스를 통해 실행 화면에 클릭을 할 때마다 particle을 찍습니다. particle은 10개까지 찍을 수 있도록 하였고, 마우스의 x, y 좌표를 변환하여 Vector3f형 배열 p[10]에 저장됩니다.

2) Attach particles



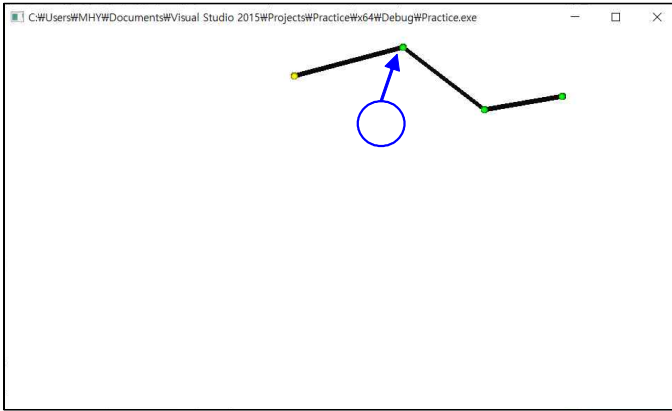
a키를 누르고, particle의 인접한 곳에서 마우스 클릭하면 해당 particle 값을 저장합니다. 그리고 이어서 다른 particle의 인접한 곳에서 마우스 클릭하면 처음 선택된 particle 값을 불러 해당 particle과 spring으로 서로 이어줍니다. particle은 마우스를 통해 클릭한 곳과 가장 가까운 control point의 거리가 0.25 이내일 때 선택되며, 2번 클릭할 때 같은 particle을 선택하면 spring이 이어지지 않도록 하였습니다.

3) Nail particles



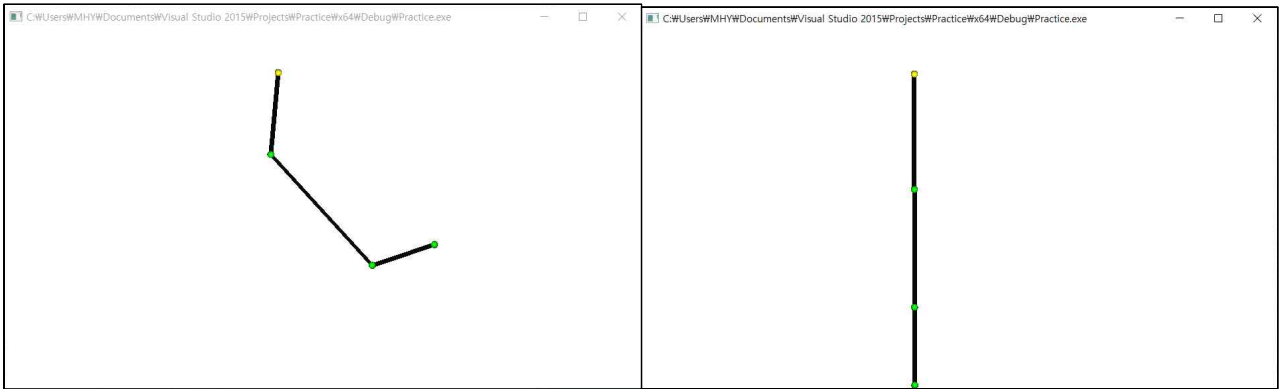
n키를 누르고, particle의 인접한 곳을 클릭하면 해당 particle이 고정되도록 하였습니다. 고정된 particle의 색은 위 그림과 같이 노란색으로 변합니다.

4) Drag a particle while simulating the system



d키를 누르고, particle의 인접한 곳을 클릭하여 드래그하면 해당 particle이 마우스를 따라 움직이도록 하였습니다. glfwSetCursorPosCallback 함수를 통해 마우스 입력이 있는 드래그를 하는 동안, 마우스를 따라 반복적으로 particle 값을 업데이트하고, 마우스를 따라 particle이 움직이도록 화면에 표시하였습니다.

5) Turn on/off point damping



처음에는 point damping이 꺼져있는 상태입니다. 왼쪽 그림처럼 space바를 눌러 실행시키면 활발하게 계속해서 운동하는 것을 볼 수 있습니다. p키를 누르면, point damping이 켜져 실행되도록 하였습니다. 오른쪽 그림처럼 point damping이 적용되어 운동이 감쇠하는 모습을 볼 수 있습니다.

solveODE함수에서 첫 번째 for문을 수정하였습니다. point damping이 켜지면, 아래 강의자료 식의 형광색 부분처럼 각 particle에 velocity의 반대방향으로 damping force를 주도록 하였습니다.

☐ Point damping

- ☒ Damping force in opposite direction to the velocity

$$\frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{f}_i(t) - c_i \mathbf{v}_i(t)}{m_i}$$

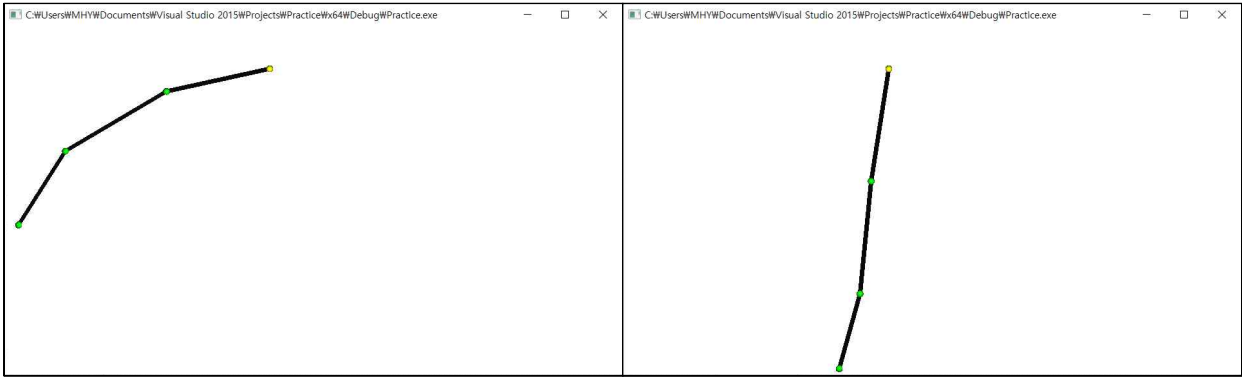
```
void solveODE()
{
    // Total force
    Vector3f f[10];

    for (int i = 0; i < nParticles; i++)
    {
        // Initialization
        f[i].setZero();

        // Gravity
        if (useGravity) f[i] += m*gravity;

        if (keyp)
            f[i] -= 0.01*v[i];
    }
}
```

6) Turn on/off damped spring



5번과 마찬가지로 처음에는 damped spring이 꺼져있는 상태로 시작합니다. 왼쪽 그림처럼 space바를 눌러 실행시키면 활발하게 계속해서 운동하는 것을 볼 수 있습니다. s키를 누르면, damped spring이 적용되어 오른쪽 그림처럼 운동이 감쇠하는 모습을 보입니다.

solveODE함수에서 두 번째 for문을 수정하였습니다. damped spring이 켜지면, 아래 강의자료 식의 형광색 부분처럼 spring의 velocity에 비례하는 damping force를 주도록 하였습니다. 변수 n_i 는 강의자료의 n_{ij} 이고, 변수 vel_i 는 강의자료의 v_{ij} 입니다.

```
for (int i = 0; i < nEdges; i++)
{
    Vector3f v_i = p[e1[i]] - p[e2[i]];
    float L_i = v_i.norm();
    Vector3f n_i = v_i.normalized();
    Vector3f f_i = k[i] * (L_i - l[i]) * n_i;

    f[e2[i]] += f_i;
    f[e1[i]] -= f_i;
}
```

□ Damped spring

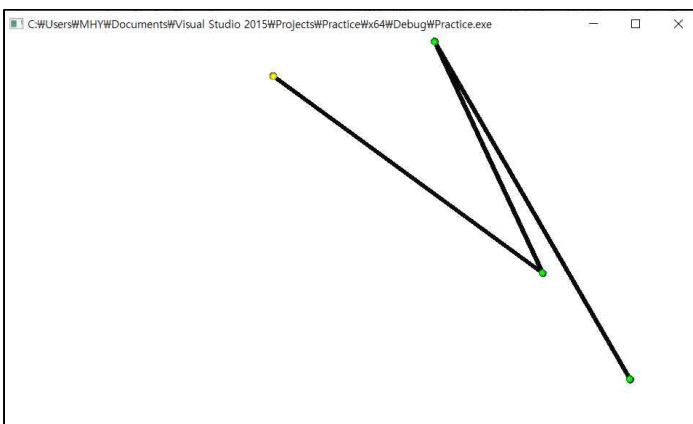
- Damping force proportional to the velocity of the spring

$$\frac{d\mathbf{v}_i(t)}{dt} = \frac{\mathbf{f}_i(t) - \sum_{j \in N_i} c_{ij} (\mathbf{v}_{ij}(t) \cdot \mathbf{n}_{ij}(t)) \mathbf{n}_{ij}(t)}{m_i}$$

```
if (keys)
{
    Vector3f vel_i = v[e1[i]] - v[e2[i]];
    Vector3f f2_i = 0.05 * vel_i.dot(n_i) * n_i;

    f[e2[i]] += f2_i;
    f[e1[i]] -= f2_i;
}
```

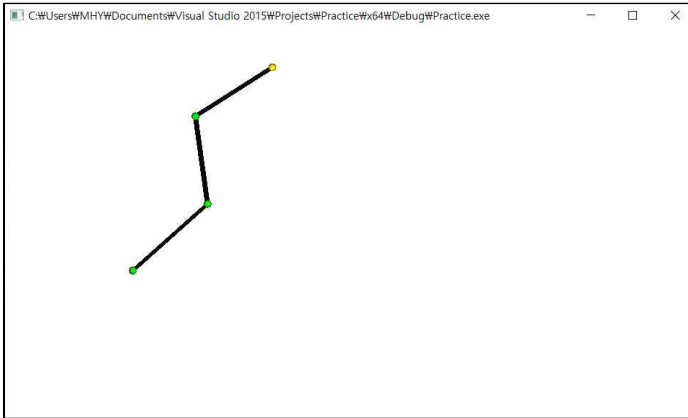
7) Make the system unstable by increasing the spring constant



```
Spring constant = 4.2
Spring constant = 4.3
Spring constant = 4.4
Spring constant = 4.5
Spring constant = 4.6
Spring constant = 4.7
Spring constant = 4.8
Spring constant = 4.9
Spring constant = 5
```

up키를 누르면 spring constant값이 0.1 증가합니다. spring constant값을 오른쪽 그림처럼 5정도까지 올려주고 e키를 눌러 Euler integration을 적용시켰습니다. 그러면 왼쪽 그림처럼 불안정상태를 보여주고 점점 폭발하는 것처럼 실행됩니다.

8) Stabilize the system by decreasing the time step size or sub-time step size



7번 상태에서 m키를 누르고 숫자 키 9를 눌러줍니다. sub-time step size를 9로 주고 modified Euler integration을 적용시켰습니다. 다시 그림처럼 안정된 상태로 점점 돌아오는 것을 볼 수 있습니다.

9) Free falling



r키를 누르면 모든 particle의 nail constraint가 제거됩니다. 그림처럼 고정된 particle이 없기 때문에 중력이 적용되어 있어 바닥으로 떨어진 것을 볼 수 있습니다.