2020년도 2학기 컴퓨터공학설계및실험Ⅰ

11주차 WaterFall Problem

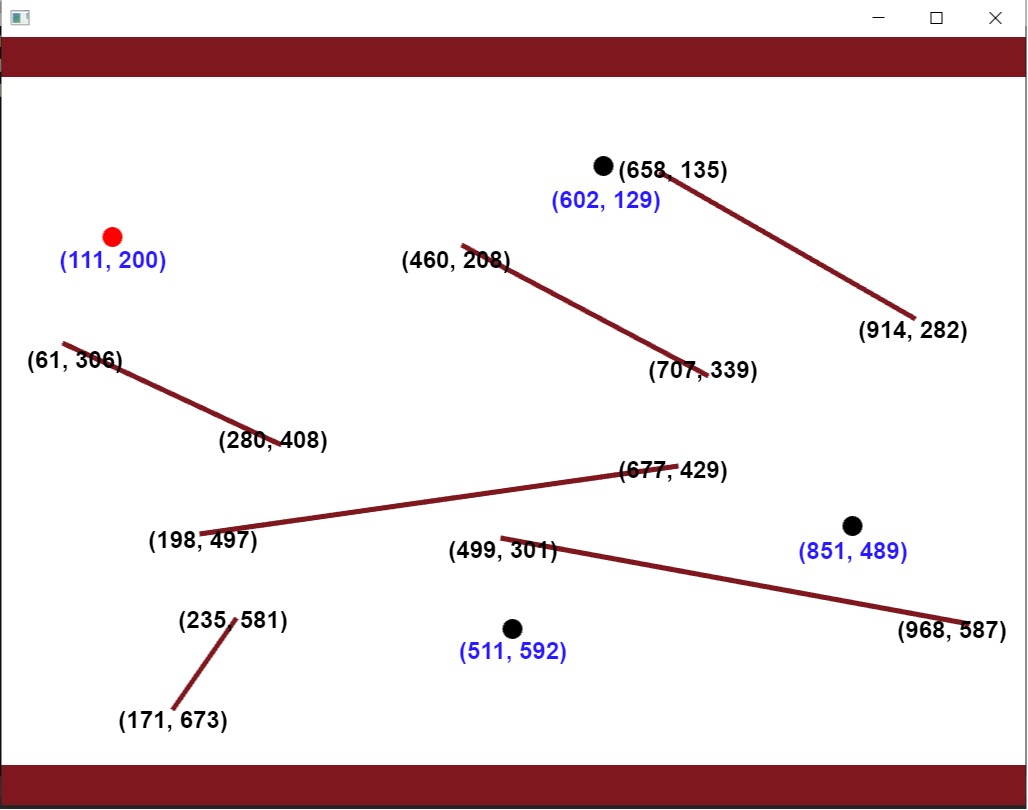
20161663 허재성

1. 실습 목적

10주차 과제에서 구현한 WaterFall 문제에서 물이 떨어져 흐르는 경로를 계산하여 화면에 나타나는 것을 openFrameWorks로 구현한다.

2. 관련 이론

10주차에서는 openFrameWorks로 물이 나오는 구멍과 선반들을 구현했다. 구현을 위해 물이 나오는 구멍을 표현하는 클래스 Point와 선반, 선분을 표현하는 클래스 LineSegment, 물 구멍과 선반들을 관리하는 Waterfall 클래스를 선언하여 사용하였다.

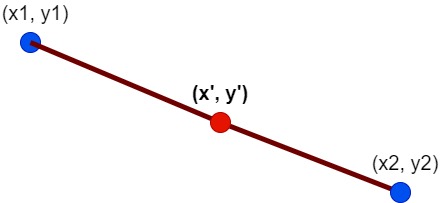


구현한 물구멍, 선반은 위와 같다. 파란 색 좌표는 점의 좌표고 검은 색 좌표는 선분의 양 끝 점의 좌표이다.

현재 가장 왼쪽 점에서 물이 나온다고 가정하자. 만약 구멍 아래에 어떤 선반도 없을 경우 물은 바닥까지 수직으로 계속 떨어질 것이다. 위의 경우, 중간에 선반(두 점 (61, 306), (280, 408)을 이은 선분)이 가로막고 있다. 따라서 물은 수직으로 떨어지다가 선반을 따라서 흐르게 된다.

물이 수직으로 떨어지다 선반과 만나는 위치를 알아야 한다. 위치를 알기 위해서는 선반의 기울기를 계산해야 한다.

물이 나오는 구멍의 좌표는 (111, 200)이고 물은 선반과 만날 때 까지 수직으로 떨어지므로 물이 떨어지는 경로의 x 좌표는 111을 유지한다. 선반과 만나는 위치는 선반을 표현하는 선분 위의 점들 중 x 좌표가 111인 점이다.



좀 더 일반적으로, 수학적인 방법으로 알아본다. 두 점 (x1, y1), (x2, y2)를 잇는 선분이 있을 때 선분의 기울기 m은 m = (y2 – y1) / (x2 – x1) (x1 과 x2는 같지 않음)이다. 한편 선분 위에 (x’, y’)가 존재하므로 두 점 (x’, y’), (x2, y2)를 이은 선분의 기울기도 마찬가지로 m이다. 즉 다음 식이 성립한다.

|  |
| --- |
| (y2 – y’) / (x2 – x’) = (y2 – y1) / (x2 – x1) |

y’을 제외한 모든 값을 알고 있는 경우이므로 y’에 대하여 정리하면 다음과 같다.

|  |
| --- |
| y’ = y2 – (x2 – x’) \* ((y2 – y1) / (x2 – x1)) |

위의 식을 이용하면 물 구멍에서 떨어지는 물이 선반과 만나는 점을 알아낼 수 있다.

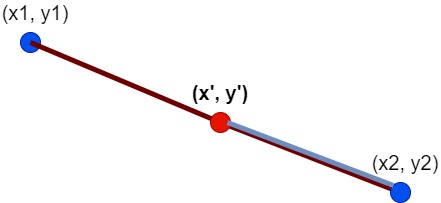
기울기 m이 간단할 경우 다음과 같이 구할 수 있다.

|  |
| --- |
| y’ = y2 – m \* (x2 -x’) |

실제로 위의 예시에 적용해 보면 x1 = 61, y1 = 306, x2 = 280, y2 = 408, m = 0.4657, ’ = 111 이므로 y’은 329.2876712라는 값이 나온다. 일반적으로 정확히 정수로 떨어지지는 않을 가능성이 높다. 329를 선택하면 선분 상의 점보다 살짝 위에 떠있게 되고 반대로 330을 선택하면 선분 상의 점보다 살짝 아래에 있게 된다. 문제의 요구에 맞게 정하면 된다.

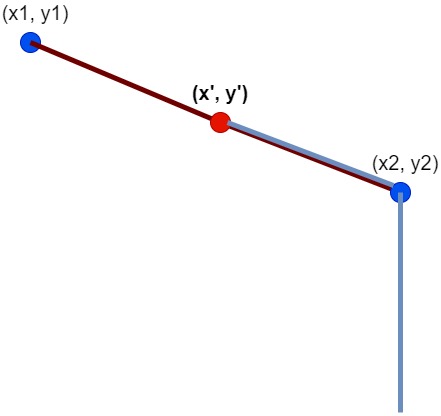
선반과 물이 만나는 점을 알아냈으니 물이 흐르는 방향을 알아야 한다. 물은 중력에 의해 위에서 아래로 흐르므로 아래쪽으로 기울어진 방향으로 흐르게 된다. 선분의 양 끝 점 중 y 좌표가 더 큰 점이 스크린 상 아래에 있다. 수학에서 흔히 생각하는 일반적인 좌표 평면과 다르게 y 좌표가 클수록 더 아래에 위치한다는 것을 주의해야 한다.

물을 가로막는 선반의 양 끝점의 y좌표를 비교해서 더 큰 점을 찾는다. 물이 처음 선반과 만나는 점과 양 끝점 중 y 좌표가 더 큰 점을 이어서 선분을 그리면 물이 흐르는 모양이 구현된다.



정확하지는 않지만 대략 위와 같은 그림이 그려진다.

선반을 따라 흐르는 물이 선분의 끝 점에 도달했다고 가정하자. 관성을 고려하지 않기로 했으므로 물은 선반의 끝에서 수직으로 떨어질 것이다. 따라서 다음 그림과 같은 상황이 된다.



물이 떨어지기 시작하는 점은 새로운 물 구멍과 마찬가지 역할을 하게 된다. 해당 점에서부터 물이 수직으로 떨어지기 시작해 만약 가로막는 선반이 없을 경우 바닥까지 계속 떨어질 것이다. 만약 물의 수직 경로에 가로막는 또 다른 선반이 있을 경우 위의 기울기로 교점을 찾는 방법을 이용해 만나는 점을 찾고 물이 흐르는 방향을 다시 찾으면 된다. 이 과정을 물이 더 이상 가로막는 선반 없이 바닥까지 떨어질 때까지 반복하면 된다.

3. 실습 방법

10주차에서 작성한 기존에 존재하는 Point 클래스와 LineSegment 클래스와 별개로 흐르는 물을 표현하는 Water 클래스를 추가하는 것을 고려할 수 있다. 하지만 기본적으로 흐르는 물도 두 점을 이은 선분으로 표현되므로 Water 클래스를 전혀 새로운 클래스로 작성하는게 아닌 LineSegment를 상속하는 유도 클래스로 작성하는 것을 고려할 수 있다. 처음 물이 떨어지는 구멍을 선분의 한쪽 끝 점으로 하고 물이 수직으로 떨어지다 선반과 만나는 점을 선분의 다른 한쪽 끝 점으로 생각할 수 있다. 다만 이럴 경우 기존의 LineSegment의 멤버 변수 leftP, rightP는 왼쪽 점, 오른쪽 점을 의미했는데 Water에서는 멤버 변수의 이름은 그대로지만 그 의미가 조금 달라진다는 것에 주의해야 한다. 또한 유도 클래스로 작성할 경우, 유도 클래스일지라도 기반 클래스의 private 멤버에는 직접 접근할 수 없으므로 LineSegement 클래스의 멤버 leftP, rightP의 접근 제어자를 protected로 변경하거나 기존에 존재하는 getter 멤버 함수들을 이용해야 한다.

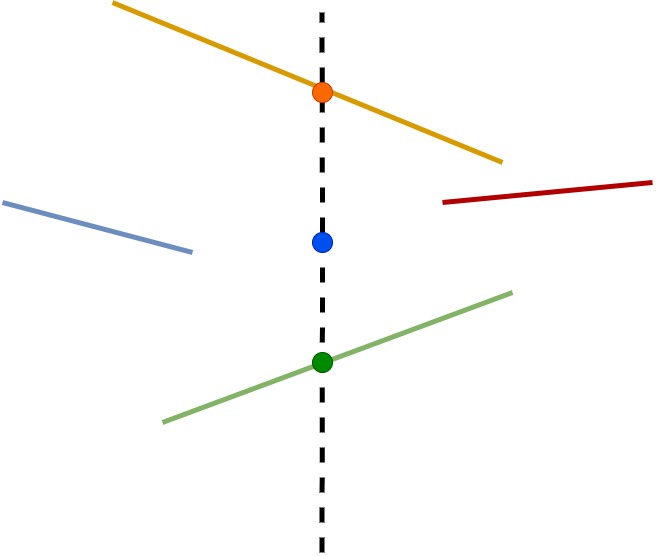
또한 10주차에서는 선분의 양 끝점은 더 이상 변경될 일이 없었지만 물의 경우 경로가 바뀌므로 양 끝점도 변경되어야 한다. LineSegment의 경우 점의 좌표가 변경될 일이 없으므로 getter 함수만 정의했고 setter 함수를 정의하지 않았지만 Water의 경우 선반이 가로막을 때마다 물의 새로운 시작점, 끝점을 정해야 하므로 setter 함수를 필요에 따라 정의하여 사용한다.

그 외에도 적절히 필요에 따라 멤버 함수들을 정의하여 사용한다.

물이 어떤 선반의 방해도 받지 않는 경우가 있을 수 있다. 이럴 경우 물은 수직으로 바닥까지 떨어진다. 바닥은 y좌표가 728(또는 729)부터 시작하므로 물이 떨어지는 점의 좌표는 (물구멍의 x좌표, 728)이다. 물 구멍의 좌표와 해당 점의 좌표를 잇는 선분을 그리면 수직으로 끝까지 떨어지는 물 줄기가 구현된다. 따라서 생성자로 Water 객체를 생성할 때 leftP의 좌표는 물구멍의 좌표, rightP의 좌표는 x 좌표는 leftP의 x 좌표, y 좌표는 728로 기본적으로 설정해둘 수 있다.

바닥의 y 좌표는 문제의 요구사항에 따라 달라질 수 있으므로 문제의 요구에 따른다.

물 구멍의 좌표가 주어졌을 때 물의 낙하를 방해하는 선분을 파악하는 것이 중요하다. 모든 선분의 양 끝점을 파악하여 양 끝점의 범위에 물 구멍의 x 좌표가 포함되는 지 확인한다. 만약 포함하는 선분이 있다면 해당 선분이 물줄기를 가로막을 선반의 후보가 된다. 해당 선분의 물 구멍의 x 좌표를 가진 점의 y 좌표를 물 구멍의 y 좌표와 비교한다. 만약 y 좌표가 물구멍의 y 좌표보다 크다면 해당 점이 물 구멍보다 아래에 있으므로 해당 선반은 물의 진로를 방해한다. 반대로 해당 점이 물 구멍보다 위에 있다면 물은 해당 선분을 만날 수 없다. 다음 그림을 통해 이해한다.

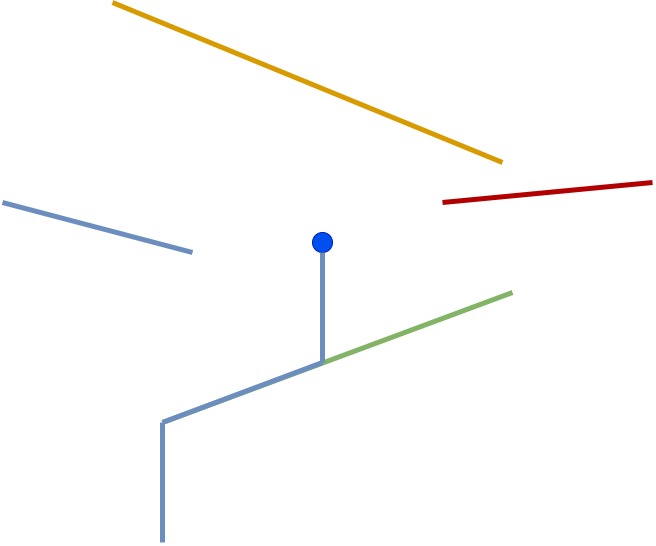


파란색 점이 물이 나오는 구멍일 때, 선분의 x 좌표의 범위가 파란색 점의 x 좌표를 포함하는 선분은 위의 그림에서 노란색 선분과 초록색 선분 뿐이다. (파란색 점을 관통하는 수직선을 그렸을 때 수직선과 만나는 선분이 노란색, 초록색 선분이다.) 따라서 빨간색 선분과 파란색 선분은 고려할 필요가 없다.

노란색 선분에서 해당 x 좌표를 가진 점을 주황색 점으로 표시했고, 초록색 선분에서 해당 x 좌표를 가진 점을 초록색으로 표시했다. 노란 색 점은 파란 색 점보다 위에 있으므로(y 좌표가 더 작으므로) 물줄기와 만나지 못한다. 따라서 더 아래에 있는(y 좌표가 더 큰) 초록색 점과 만나므로 초록색 선분이 물줄기를 가로막는 선분이 된다.

만약 초록색 선분 같은 선분이 여러 개 있다면 여러 선분들 중 가장 먼저 만나는 선분(교점의 y좌표가 가장 작은 선분)을 따라 흐르게 된다.

위의 그림에서 물이 흐르면 다음 그림과 같이 된다.



4. 기타

openFrameWorks에서 제공하는 ofGetHeight() 함수로 스크린의 높이를 구할 수 있다. 스크린의 높이가 곧 스크린의 가장 아래 좌표이다. 오픈프레임워크 공식 커뮤니티에는 다음과 같이 설명한다.

|  |
| --- |
| ofGetHeight()**int ofGetHeight()** This gets the height of your ofApp window. Useful for finding the middle of the screen like so: [ofVec2f](https://openframeworks.cc/documentation/math/ofVec2f) middle(ofGetWidth()/2, ofGetHeight()/2) [1] |

5. 참고 문헌

# [1] openframeworks.cc, “ofAppRunner”

https://openframeworks.cc/documentation/application/ofAppRunner/#show\_ofGetHeight