Final Project 보고서

전공: 컴퓨터공학과 학년: 4학년 학번: 20212022 이름: 이예준

**1.**

- 프로젝트 목표

이 프로젝트의 목표는 마우스를 클릭하거나 드래그할 때, 물방울들이 생성되어 아래로 떨어지게

되고, 경사면이 존재할 경우에는 물방울들이 경사면을 따라 흐르도록 하게 하는 것이다.

사용자는 파일을 통해 경사면의 정보를 가져와 그 정보를 기반으로 경사면을 그리고,

물방울과 경사면의 실시간 상호작용을 시각적으로 확인할 수 있다.

- 실험 환경

M1 MacBook Air에서 xcode를 기반으로 Openframeworks를 사용하였다.

Openframeworks의 버전은 v0.12.0이다.

**2.**

텍스트, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

- vector<Trickle> water : 물방울들의 위치와 상태를 저장하는 벡터

- vector<Slope> slopes : 경사면들의 정보를 저장하는 벡터

- bool isMousePressed : 마우스 클릭 여부를 저장하는 플래그

- bool load\_flag : 파일 로드 여부를 저장하는 플래그

- bool draw\_flag : 경사면 그리기 모드 여부를 저장하는 플래그

**3.**

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

- 프로그램의 초기 설정을 수행하는 함수로 배경색은 흰색으로 설정했고,

플래그 초기값은 모두 false로 설정했다.

**\* 상수 시간 작업(배경색 설정 및 플래그 초기화)을 수행하므로 시간 복잡도는 O(1) 이다.**

**\* 상수 공간을 사용하므로 공간 복잡도는 O(1) 이다.**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 마우스를 눌렀을 때, for문을 이용하여 5개의 물방울을 만들게 되는데,

ofRandom() 함수를 이용해 **x좌표와 y좌표 offset**을 -20부터 20까지의 무작위 값으로 설정하고,

**마우스의 중심을 기준으로 offset만큼 떨어진 위치**를 Trickle 구조체 circle의 x좌표와 y좌표로

저장한다. 최초의 물방울은 경사면과의 상호작용이 없으므로 onSlope는 false로 설정한 뒤

그 circle을 vector water에 저장한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 만약 경사면을 화면에 그렸다면 물방울이 경사면들 중 **상호작용하는 경사면이 있는지 확인**한다.

상호작용을 하는 경우 onSlope값을 true로 설정하고 break로 for문을 빠져나온다.

*\* getY() method를 이용해서 현재 물방울의 x좌표로 경사면의 y좌표를 얻어서*

*경사면의 y좌표 기준 오차범위 (-3 ~ -1)내에 물방울이 있으면 상호작용을 한다고 판단한다.*

*(선의 굵기 때문에 오차범위를 이렇게 지정했다)*

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

- 위의 과정에서 상호작용하는 경사면이 없었다면 그냥 수직으로 2만큼 내려오게 하고,

상호작용하는 경사면이 있었다면 해당 경사면을 찾는다. 해당 경사면을 찾으면 2점의 좌표를

이용해서 기울기를 구하고 cos(),sin() 함수를 이용해서 이동할 위치를 설정한다.

이때, 경사면의 오른쪽이 더 올라가 있다면 **중력의 작용**을 받아 왼쪽으로 물이 흐를 것이고,

경사면의 왼쪽이 더 올라가 있다면 중력의 작용을 받아 오른쪽으로 물이 흐를 것이기 때문에

경사면 두 점의 **y좌표의 대소관계에 따라서 물의 흐르는 방향을 알맞게 설정**한다.

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

- vector에 계속 push\_back하기만 하면 **메모리 관리가 안되기 때문**에

화면 범위를 넘어간 물방울들은 vector water에서 제거해준다.

\* ofGetHeight()-40으로 해준 이유는 화면 밑에 40길이의 직사각형 빨간 띠를 그려 놨기 때문에

물방울과 서로 겹칠 경우 제거해주기 위함이다.

\* isMousePressed가 참일 때 5개의 원을 생성하고, 이 부분은 상수 시간 작업이다.

water 벡터의 각 원에 대해 경사면과의 상호작용을 계산하고 업데이트한다.

각 원에 대해 모든 경사면을 체크한다.

경사면의 개수를 n, water 벡터의 원소 개수를 m이라고 하면, 최악의 경우 경사면을 체크하는

부분의 시간 복잡도는 O(n \* m)이다.

remove\_if 및 erase를 사용하여 water 벡터의 원소를 제거하는 작업도 수행한다.

이는 벡터의 크기와 비례하여 O(m) 시간이 걸립니다.

따라서 🡪 시간 복잡도: O(n \* m)

\* water 벡터와 slopes 벡터를 사용하며, water 벡터의 크기를 m, slopes 벡터의 크기를

n이라고 하면, 이 메서드의 공간 복잡도는 O(m + n)이다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

- 빨간 띠의 색을 설정한 뒤 화면의 위 아래로 직사각형을 그려준다.

그 다음 물방울은 파란색으로 색 설정, 1~3 사이의 무작위로 크기 설정을 해주고,

구조체에 저장되어 있는 물방울의 좌표에 그려준다.

만약 화면에 경사면 그리기 모드 플래그가 설정되어 있다면 선의 굵기는 3, 선의 색은

검은색으로 설정한다. 그 다음 vector slopes에 저장되어 있는 경사면의 정보를 기반으로

선을 그린다.

\* 빨간 띠와 물방울, 경사면을 그리는 작업은 모두 벡터의 크기에 비례한다.

물방울의 개수를 m, 경사면의 개수를 n이라고 하면, 이 메서드의 시간 복잡도는

O(n + m)이다.

따라서 🡪 시간 복잡도: O(n + m)

\* 이 메서드는 주로 화면에 그리는 작업을 수행하므로 추가적인 공간을 사용하지 않다.

따라서 공간 복잡도는 O(1)이다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 키보드의 ‘v’ 키를 누르면 스크린 샷을 저장한다.

키보드의 ‘q’ 키를 누르면 프로그램을 종료한다.

키보드의 ‘d’ 키를 누르면 경사면 그리기 모드를 허용한다.

키보드의 ‘e’ 키를 누르면 그려져 있는 경사면을 삭제한다.

\* 각 키에 대한 조건문을 체크하는 작업은 상수 시간 작업이므로 시간 복잡도는 O(1)이다.

\* 각 키에 대한 플래그를 설정하거나 해제하는 작업은 상수 공간을 사용하므로

공간 복잡도는 O(1)이다.

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

- 키보드의 ‘l’ 키를 누르면 경사면의 정보가 있는 파일을 불러온다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

- 마우스를 클릭하거나 드래그하면 마우스의 좌표를 저장하고,

클릭했을 때는 플래그 설정, 클릭을 해제했을 때는 플래그도 해제한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

- 선택된 파일의 이름을 가져온 뒤 파일을 읽기 위해 ofFile 객체를 생성한다.

만약 파일이 존재하지 않는 경우 에러매세지를 출력하고, 파일이 존재하는 경우 파일을

찾았다는 메세지를 출력한다. 그 다음 파일을 버퍼로 불러온다.

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

- 파일을 한 줄 씩 읽어오면서 공백을 기준으로 문자열을 분리한다.

만약 한 줄에 하나의 값이 있는 경우 경사면의 개수를 의미하므로 정수로 반환한 뒤

경사면의 개수가 몇개인지 출력해준다. 그 다음 vector의 크기를 경사면의 개수만큼 설정한다.

만약 한 줄에 두 개 이상의 값이 있는 경우 경사면의 좌표를 의미하므로 각 값들을

slopes 벡터에 저장한다. 만약 좌표가 화면의 범위를 넘어가게 될 경우 에러 메세지를 출력하고

프로그램을 종료한다.

\* 파일을 읽고 각 라인을 처리하는 작업을 수행한다. 각 라인에서 경사면의 좌표를 읽고

벡터에 저장한다. 파일의 라인 수를 L이라고 하면, 이 메서드의 시간 복잡도는 O(L)이다.

따라서 시간 복잡도는 O(L)이다.

\* 파일을 버퍼로 읽어오고 벡터에 저장하는 작업을 수행한다. 파일 라인 수를 L,

slopes 벡터의 크기를 n이라고 하면, 이 메서드의 공간 복잡도는 O(L + n)이다.

**4.**

텍스트, 폰트, 화이트이(가) 표시된 사진

자동 생성된 설명

- 물방울을 나타내는 구조체로 물방울의 위치와 경사면과 상호작용하는지에 대한 상태를

나타낼 수 있다.

텍스트, 폰트, 친필, 화이트이(가) 표시된 사진

자동 생성된 설명

-경사면을 나타내는 구조체로 경사면 양 끝 2개의 점의 위치와

x좌표를 이용해서 y좌표를 구하는 method getY를 나타내고 있다.

텍스트, 폰트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

- x1과 x2가 똑같을 경우 이 경사면은 수직이므로 y1값을 반환한다.

다를 경우 먼저 두 점의 좌표를 이용해서 경사면의 기울기를 구한 뒤

일차 방정식을 응용하여 으로 x값에 대응하는 y값을 반환한다.

**- Flow Chart**

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**5.**

- 물방울들 생성

마우스를 클릭했을 때, 무작위로 물방울들을 생성시켜서 물줄기를 연상시키도록 시각화했다.

- 떨어지는 물방울들

프레임마다 일정 높이만큼 물방울들을 떨어뜨려서 마치 현실처럼

중력이 존재하는 것 같이 구현했다.

- 물방울들과 경사면과의 상호작용

마우스의 클릭 및 드래그로 생성되는 물방울들이 떨어지면서 경사면에 붙게 될 때,

경사면을 그냥 통과하지 않고 경사면과의 상호작용이 일어나 경사면을 따라 흘러

내려갈 수 있도록 구현함으로써 현실감을 더했다.

**6.**

스크린샷, 직사각형, 화이트, 디자인이(가) 표시된 사진

자동 생성된 설명 < 기본 화면 >

스크린샷, 라인, 텍스트이(가) 표시된 사진

자동 생성된 설명 < 마우스 클릭 >

스크린샷, 그래픽, 디자인이(가) 표시된 사진

자동 생성된 설명 < 마우스 드래그 >

라인, 도표, 스크린샷, 텍스트이(가) 표시된 사진

자동 생성된 설명 < 경사면을 타고 흐르는 물방울들 >

**7.**

- 느낀 점

이번 프로젝트를 통해 물리 시뮬레이션을 구현할 때, 어떤 부분을 구현하면 더 현실적일

수 있을지에 대해 많은 고민을 하게 했고, 또 그만큼 구현 방법에 대해 배울 수 있었다.

- 개선 사항

경사면이 서로 교차되었을 때 어떻게 조치를 취할지 방법을 정하고,

그렇게 만들어진 공간에 물이 들어차서 웅덩이가 만들어 질 수 있도록 구현하면 더 좋은

사용성을 보장할 수 있을 것이다.