실험 WaterFall-1주차 결과보고서

전공: 컴퓨터공학 학년: 2학년 학번: 20201635 이름: 전찬

**1. 목적**

OpenFrameworks 실습 코드를 직접 수행해보며, 각 line이 어떤 기능을 수행하는지, 어떤 의미를 갖는지 서술한다.

**2. 실습 코드에서 각 line의 기능과 의미**

기본적으로 OpenFrameworks는 ofApp.cpp의 일정 부분(주로 이미 존재하는 method에 내용을 추가해주는 형식으로)에 원하는 작동을 작성하는 형식으로 여러 동작을 수행시킬 수 있다. 그 중에서도 이번 실습에서는 setup(), draw(), keyPressed(), keyReleased() 라는 ofApp의 이미 존재하는 method를 사용해서 WaterFall 프로그램에서 필요한 선분과 점을 그려주게 된다. 각 method가 어떤 역할을 수행하는지 알아보자.

우선 setup() method는 프로그램을 실행시킬 때, 처음 한 번만 실행되는 method이다. 이번 실습에서는 ofSetFrameRate(15); ofBackground(255, 255, 255); 등과 같이 프로그램을 실행하는 동안 필요한 초기 설정을 해주는 용도로 사용한다.

draw() method는 setup()과 다르게 반복적으로 계속해서 수행되는 method이다. setup()에서 호출해준 ofSetFrameRate(15)를 통해서 draw()의 수행간격에 제한을 걸어줄 수도 있다. 이번 실습에서는 위와 아래쪽의 빨간 사각형을 그려줌과 함께, draw\_flag가 set되면 input.txt에서 받은 정보들을 토대로 선분과 점을 그려주는 역할을 반복적으로 수행한다.

keyPressed()와 keyReleased()는 모두 키보드의 입력에 관련된 method 이다. 약간의 차이점이 존재하는데, keyPressed()는 키보드에서 일정 키를 눌렀을 때, 그 키를 입력으로 받지만, keyReleased()는 일정 키를 누른 이후 손을 뗀 경우에 그 키를 입력으로 받는다. 쉽게 말하자면 한 키를 꾹 누르고 있을 때, keyPressed()는 입력을 계속해서 받으며, keyReleased()는 한 번의 입력만 받는다고 할 수 있다. 이번 실습에서는 keyPressed()를 통해서 ‘q’를 입력 받은 경우에는 동적 할당 해제와 함께 프로그램의 종료를, ‘d’ 키를 입력 받은 경우에는 load\_flag가 1인 경우 draw\_flag를 1로 변경시켜주는 역할을 수행한다.

keyReleased()에서는 ‘l’ 키의 입력을 받은 경우에는 input.txt의 정보를 동적 할당 받으며 load\_flag를 1로 바꿔주는 역할을 수행한다. 또한 왼쪽 방향키와 오른쪽 방향키로 물이 나오는 점을 바꿔주는 역할 또한 수행한다. 여기에서 keyReleased()를 사용한 이유는 동적 할당을 한 번만 받아야 하며, 왼쪽, 오른쪽 방향키 또한 계속해서 입력을 받으면 output 부분이 너무 복잡해질 수 있어서 위와 같이 설정해 놓았다.

위에서 미리 정해진 method의 내용을 바꿔주는 형태와 달리, 이번 실습에서는 ofApp class를 변경시키거나, 새로운 method를 정의하는 방법 또한 사용하였다. 우선 ofApp class에서 여러 변수를 추가해주었는데, draw\_flag, load\_flag, num\_of\_line, selected\_dot, line\_counter 등이 있다. 위 변수를 통해서 동적 할당과 함께, 여러 method의 작동을 조절할 수 있다. 특히 개인적으로 2차원 동적 할당을 받을 변수 int\*\* line\_data, int\*\* dot\_data를 설정해 주었다. 또한 화면의 범위를 넘어가지 않는 점, 선의 개수를 저장하는 변수 int line\_counter, dot\_counter를 설정해 주었다.

마지막으로 ofApp class에 processOpenFileSelection이라는 method를 정의해 주었다. 이 method는 input을 정할 수 있는 화면을 나타나주며, 선택된 input.txt를 바탕으로 line\_data, dot\_data에 선분과 점을 동적할당 받는 역할을 수행한다. 결과적으로 위에서 설명한 ‘l’ 키의 입력을 통해서 keyReleased() 에서 processOpenFileSelection을 실행시키는 형태임을 알 수 있다.

**3. 실습에서 작성한 코드의 설명**

이번 실습에서는 크게 5가지 부분에서 일정 역할을 수행하는 코드들을 작성했다.

우선 첫 번째로는 ‘l’ 키의 입력과 함께 동적 할당을 받는 코드를 작성했는데, 처음에 받은 num\_of\_line, num\_of\_dot 만큼의 동적 할당을 먼저 받은 이후에(line\_data에는 num\_of\_line\*4, dot\_data에는 num\_of\_dot\*2의 배열을 동적 할당), 범위 안에 포함되는 좌표들만 저장해준 이후, 남은 부분은 동적 할당을 해제하는 형식으로 코드를 작성했다. 또한 input\_type을 0에서 1로 바꾸기 위해 num\_of\_line만큼 line 정보를 읽은 이후에 input\_type = 1로 바꾸어주는 코드를 작성했다.

두 번째로 ‘d’ 키를 입력 받은 이후에 load\_flag==1 이면, draw\_flag = 1, selected\_dot = 0 이 되게 코드를 작성했다.

세 번째로 draw() method에서 draw\_flag==1인 경우에 저장한 배열의 정보에 따라서 선분과 점을 line\_counter, dot\_counter을 바탕으로 그려주는 함수를 작성했다. 이때 selected\_dot에 해당하는(dot\_data[i]에서 i==selected\_dot 인 경우에) ofSetColor(255, 0, 0)으로 빨간색 점을 그려주었다.

네 번째로 왼쪽, 오른쪽 방향키를 입력 받은 경우에 selected\_dot을 바꿔주는 코드를 작성했다. 오른쪽 방향키인 경우에는 selected\_dot = (selected\_dot+1)/dot\_counter 의 형태로, 왼쪽 방향키인 경우에는 if(selected\_dot==0){~~~} else{~~~} 의 형태로 코드를 작성해 주었다.

마지막으로 ‘q’ 키를 입력했을 때, 각 data에서 필요한 2차원 동적 할당 해제를 수행해 주었다.