

## 2019학년도 2학기 정보과학II 데이터시각화 processing 제출 양식

간단한, 수업 중 프로젝트 수행평가입니다.

2. processing 예제 분석하고 살짝 바꿔서, 자기 관심 분야 관련 시뮬레이션 만들어보기

평가 관점은 다음과 같습니다.

- ① 프로젝트에 대한 설명이 구체적으로 설명되어 있다.
- ② 프로젝트 작품이 충실하게 구현되어있다.
- ③ 과학 및 수학 분야와 연관성을 가진 주제를 다루고 있다.
- ④ 과학 및 수학 분야에 대한 흥미와 호기심을 불러일으킬 수 있다.
- ⑤ 창의적 데이터 시각화 아이디어가 제시되어 있다.
- ⑥ 구체적 적용 상황이 제시되어 있다.
- ⑦ 과학 및 수학 분야의 연구 활동에서의 활용 가능성이 높다.

어떻게 만들어서 제출해야 하는가?

- 다음 페이지의 양식에 맞추어 내용을 작성해 제출하면 됩니다.
- 수업시간을 충분히 활용해서 과제를 수행하세요. 미루고 있다가 집에 가서? 하는 것 금지!
- 다음 시간 끝날 때, pdf파일로 만들어, 개인별 USB에 담아서 선생님에게 제출합니다.
- gnuplot, processing 각각 따로 한 번에 제출합니다!

### 2. processing 제출

- 폴더 이름 : (19.09.XX)정보과학2processing(학번,이름)
- 기본 제출
  - (19.09.XX)정보과학2processing(학번,이름).pdf 파일
  - \*.pde 파일(만들어 저장한 pde 폴더, 파일 및 모든 파일)

이 양식 자료를 이쁘게?(글꼴, 배치 등) 꾸미려고 시간 버리지 말 것!

핵심으로 정확하게! 꼭!

분량은 중요하지 않지만, 노력은 중요함!

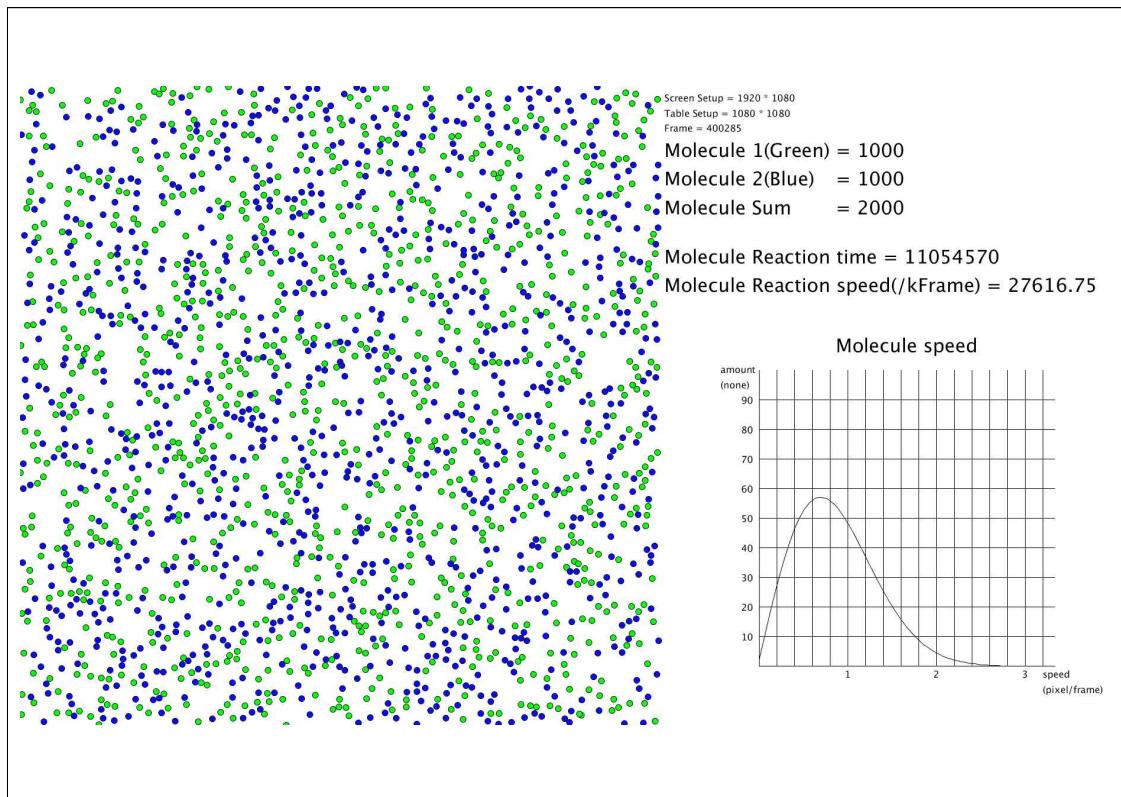
학번 : 19034

이름 : 문현호

제목 : 이상적인 분자 운동 구현과 분석

만든 것에 대한 설명(결과 그림 캡처 후 삽입)

- 관련 수학/과학 교과명 : 화학(+ 물리)
- 관련 내용 또는 주제 : 분자의 운동
- 만든 processing 시뮬레이션 결과 캡처 사진



- 만든 것의 과학/수학/정보 관점의 의미

많은 양의 분자를 구현하고, 2차원의 완전 탄성 충돌을 시킨다. 두 종류의 분자를 구성하여 단위 시간당 두 분자가 충돌하는 횟수를 분석하여 반응 속도에 비례하는 값을 표현할 수 있고, 완전 탄성 충돌을 통해 변하는 분자들의 운동 속력을 그래프로 나타내어 기체 분자 운동론에서의 분자 속력 분포를 보여줄 수 있다.

- 자신이 분석해 변경하거나 바꾼 것들

배열에  $x$ ,  $y$ ,  $v_x$ ,  $v_y$  값을 저장하고, 충돌 시의 속도 변화를 나타내는 함수를 완전 탄성 충돌 시의 운동에너지와 운동량이 모두 보존됨을 이용하여 '가장 간단한 방법'으로 제작하였다. 모든 분자들은 질량이 같고, 프로그램을 시작할 때의 속도 역시 1pixel/frame으로 같게 하였다.

여기에 화면을 분할하여 왼쪽 정사각형 공간에 입자들이 움직이도록 하고, 오른쪽 공간에는 현재 상태를 분석할 수 있는 텍스트들을 띄울 수 있게 하였다. 이 중에는 현재 입자들의 속도 분포를 나타내는 '그래프'도 있는데, 이 또한 배열에 값들을 저장하고, 좌표를 지정한 다음,  $x$ 축과  $y$ 축과 격자를 그리고, 그 위에 값들을 선으로 이어 나타내도록 하였다. 덕분에 입자계를 분석하는 것이 쉬워졌다.

분석을 더 쉽게 할 수 있도록 여러 가지 기능을 추가하였다. 스페이스바를 누르면 입자들의 속도 벡터를 나타내는 직선을 표시하도록 했다. 나머지는 별 것은 아니지만,  $s$ 키를 누르면 slow, 1프레임이 0.1초로 바뀌고,  $d$ 키를 누르면 delay, 1프레임이 1초가 된다.

- 참고한 예시 또는 내용

물리화학에서 분자의 운동을 나타낼 때는 탄성 충돌을 가정한다고 알고 있었기 때문에, 할리데이와 인터넷에 나와 있는 실제 탄성 충돌 식을 참고하였다. 화학 교과서에서 나오는 반응 속도와 분자 운동 속도 그래프도 참고하였다.

특히, processing의 문법을 배우기 위해 processing 사이트를 많이 참고했다. 생각보다 C 문법과 비슷한 부분이 많아 편했다. 물론, draw 함수 안에서 draw를 하는 알고리즘이 보이는 방법과는 조금 달라 조정해주는 부분이 힘들었다.

친구들 중에서도 이렇게 입자의 운동을 표현한 것이 많았고, 그들을 통해 processing에 대한 기초적인 지식들을 많이 배웠다. 입자의 위치와 속도를 저장해주는 내장 구조체가 있다는 사실은 좀 늦게 알게 되었다... 막상 지금 생각해보니, 내가 직접 이를 만들었다는 점이 뿌듯하다.

- 찾아보면서 재미있었던 예시와 그 이유는?

예시를 많이 찾아보지는 않았지만, 봤던 예시 중에서는 일단 가장 많은 도움을 주었던, 우리 반 찬빈이의 폭죽 알고리즘이 재미있었다. 중력가속도를 구현하여 쏘아올린 불꽃이 위에서 여러 방향으로 날아가는데, 날아가는 경로도 중력가속도를 구현한데다, 색깔 밝기도 점점 열리는 것이 멋지고 재밌었다. 색깔도 랜덤이라 여러 번 클릭하여 동시에 여러 폭죽이 터지는 장면은 정말 아름다웠다! 언젠가 꼭 나도 저런 것을 한 번 만들어봐야겠다는 생각이 들었다.

이외에도 7반의 서준이는 분자의 속력을 실제 계산식과 비교하기도 했고, 7반의 진호는 분자의 반응과 역반응을 구현하여 동적 평형을 구현하기도 했다. 분자나 입자를 이용하고, 탄성 충돌을 보이는 친구들은 굉장히 많았다.

- 만들면서 생각해보거나 시도해 본 것들

처음에는 그냥 분자의 반응 속도가 농도의 곱에 비례하는 이유를 알아보기 위해 개발을 시작했지만, 좀 더 실제 분자 같아 보이기 위해 탄성 충돌을 넣기로 했고, 이것이 제일 힘들었다. 물리를 못 하는 나는 맨 처음에는 분자들의 운동 속력이 항상 일정할 것이라고 생각했지만, 실제로는 아니었다. 알고리즘을 대충 설명하자면, 분자들의 질량은 모두 같기 때문에, 부딪히는 순간에 둘의 좌표 차이를 알아내고, 그 각도의  $\sin$  값과  $\cos$  값을 구하며, 그 둘이 부딪힌 방향의 속도를 서로 교환한다. 이렇게 되면 무조건 속력 값은 충돌할수록 처음과 달라질 것이고, 이것을 알고 나서 분자들의 운동 속력을 실제 그래프로 그려보면 어떨까라는 생각을 하게 되었다.

실제 분자 수는 보통 2000개까지 넣는데, 이것만으로는 그 분자 운동 속도 그래프가 예쁘게 나오지 않고, 매 프레임마다 굉장히 어지럽게 나오기 때문에, 입자들의 속력이 잘 섞였다고 가정할 수 있는 500프레임 이후부터는 값들의 평균을 표시하게 해주었다. 이러면 그래프가 2개가 나오는데, 두 배로 어지러워지기 때문에 2000프레임부터는 현재 값을 나타내지 않도록 하게 해주었다.

사실 입자 개수나 부피에 대한, 벽과 충돌할 때 벽에게 준 에너지를 분석하여 압력도 한 번 알아보고, 운동에너지(온도)에 따른 변화도 살펴 이상 기체 방정식을 알아보려고 했지만, 프레임 당 속도가 너무 빨라지면 지름이 10프레임인 입자끼리의 충돌이 어려워질 것이며, 분자량도 지정을 해 주어야 해서.. 이미 이 입자계도 시작 속력이 1이지만, 이후에 가다 보면 속력이 4에서 5 이상인 녀석도 아주 가끔 일시적으로 나오고, 이 경우 분자들끼리의 충돌이 서로 멀어지기 전에 몇 번 더 일어나는 등의 에러가 발생할 수 있어 포기했다.

이미 이전 경우에도 분자들끼리 충돌을 한 다음 프레임에서 서로의 거리가 10 이상이 되지 않으면, 다시 충돌을 일으키는 버그가 있었는데, 벽이나 다른 입자와 충돌하기 전 같은 입자와 다시 충돌할 때는 충돌 연산이 일어나지 않도록 입자를 나타내는 배열에 변수를 지정해주었다.

- 만들면서 이해하게 되었거나 느낀 점들

어릴 때 스크래치를 가지고 게임을 만들다 공과 공이 만났을 때 튀기는 알고리즘을 만들어 보고 싶었는데, 그 당시의 내 지식으로는 도저히 할 수 없어서 포기했던 기억이 난다. 고등학생이 되어 이를 만드는 데 성공해서 정말 기쁘다. 또한, 반응 속도가 농도의 곱에 비례하는 것은 언젠가 직관적으로 이해한 적이 있지만, 온도(운동에너지)에 따른 분자 운동 속도 그래프가 진짜 이런 식으로 나온다는 점은 정말로 놀라웠다! processing 생각보다 너무 신기하고 재밌다. 이후에 시뮬레이션 프로그래밍 수업 꼭 들어보고 싶다.