Mason Edwards

PHY3513

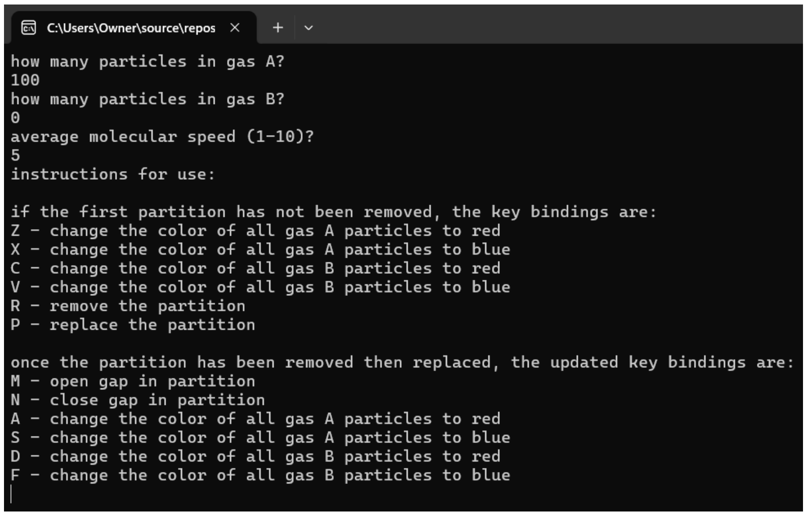
Programming Challenge

I. Program Description

Using Microsoft Visual Studio, C++, and the SFML graphics library, I programmed an animation of the Maxwell's Demon thought experiment we covered in class. The program provides an interactive visual representation of the experiment, which contains within it several of the concepts we have learned this semester. There are no quantitative data points to be analyzed, as the point of this program is to simulate a dynamic two-dimensional depiction of a thermally isolated physical system containing an ideal gas which the user can interact with in a way that allows them to play the part of the demon in Maxwell's thought experiment. Therefore, for my analysis I will explore the physical principles underlying the concept of Maxwell's Demon and review its compatibility with the laws of thermodynamics and information theory.

II. Demonstration

When the user runs the program, they are first given a series of three prompts. The physical system that will be constructed is a rectangular container divided in the middle by a partition, and for the first two prompts the user gets to choose how many ideal gas particles are (pseudo)randomly generated in the left and right chambers at the start of the simulation. To model Maxwell's demon the way it was covered in class, the user should only set a non-zero quantity of particles for one of the two chambers, so for this demonstration we will generate 100 in the left chamber. The third prompt lets the user define the average molecular speed for each particle, which will be randomized for each generated particle within a certain interval. An average speed of '5' units seems to work well for the animation.



```
how many particles in gas A?
100
how many particles in gas B?
0
average molecular speed (1-10)?
5
instructions for use:

if the first partition has not been removed, the key bindings are:
Z - change the color of all gas A particles to red
X - change the color of all gas A particles to blue
C - change the color of all gas B particles to red
V - change the color of all gas B particles to blue
R - remove the partition
P - replace the partition

once the partition has been removed then replaced, the updated key bindings are:
M - open gap in partition
N - close gap in partition
A - change the color of all gas A particles to red
S - change the color of all gas A particles to blue
D - change the color of all gas B particles to red
F - change the color of all gas B particles to blue
```

Figure 1: User Prompts in the Command Line Interface upon Starting the Program

Once the user completes the prompts, instructions for the program's use are displayed in the command line interface and a render window is displayed containing the animation.

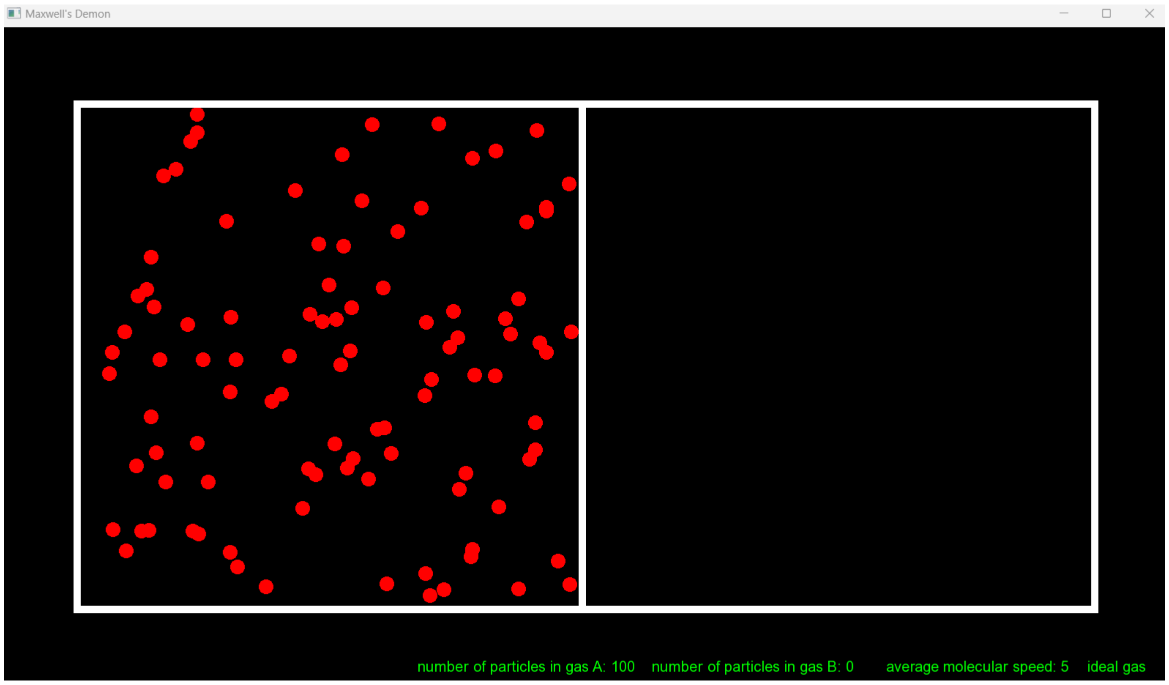

Maxwell's Demon
number of particles in gas A: 100
number of particles in gas B: 0
average molecular speed: 5
ideal gas

Figure 2: Render Window Showng the System in its Initial State

As seen above, the left chamber contains an ideal gas of a certain pressure, volume, and temperature restricted by a partition from the right chamber, which contains only a vacuum. From here, the user can remove the partition by pressing 'R' on the keyboard. Once this is done, the gas will undergoe a Joule expansion into the full volume of the container:

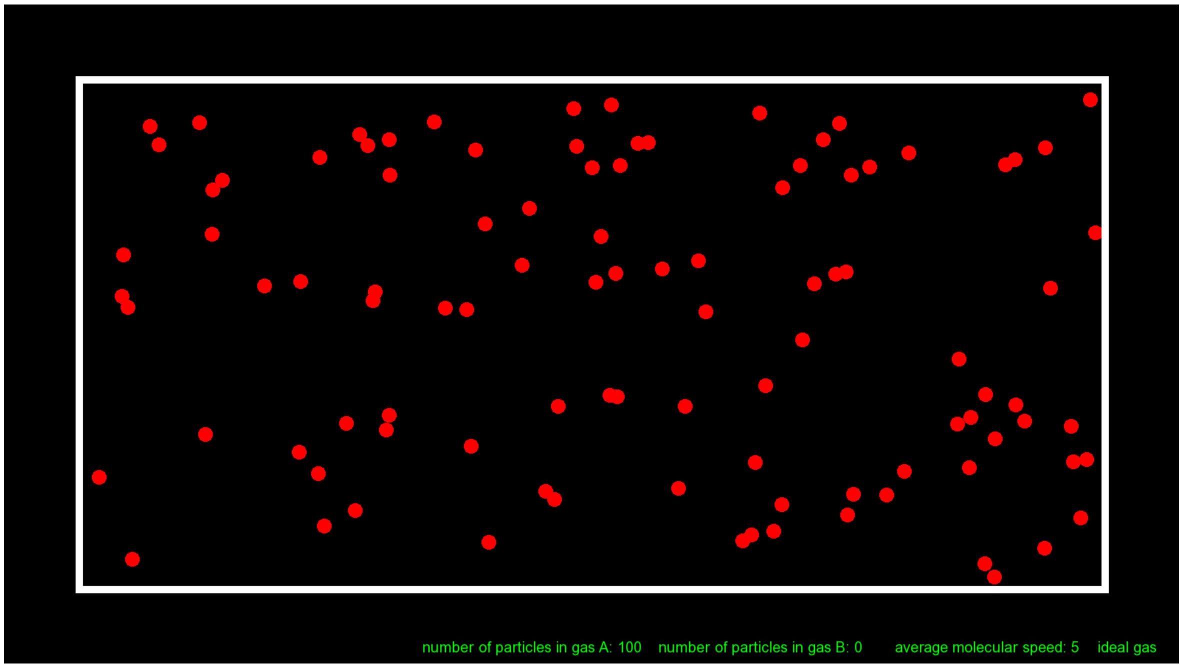

number of particles in gas A: 100
number of particles in gas B: 0
average molecular speed: 5
ideal gas

Figure 3: System after Undergoing a Joule Expansion

Now after waiting a short period of time and assuming the gas reaches equilibrium, the gas now has half its initial pressure, twice its initial volume, and has experienced no change in temperature. As we learned in class, an increase in entropy has now occurred, which will be explained in the analysis section. The next step for the user is to replace the partition by pressing ‘P’.

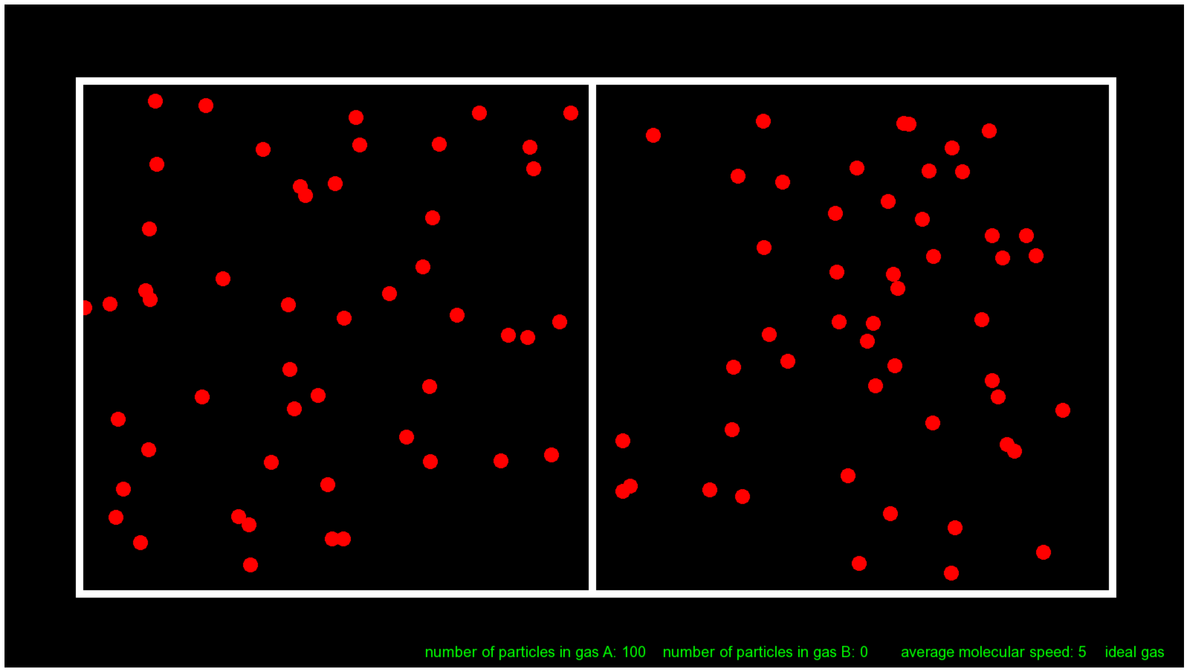

number of particles in gas A: 100   number of particles in gas B: 0   average molecular speed: 5   ideal gas

Figure 4: System with the Partition Replaced once Equilibrium is Reached after the Joule Expansion Occurs

At this point, the user is able to begin the thought experiment. If all the gas particles now in the right chamber can be moved through the partition back into the left chamber, the state variables of pressure, volume, and temperature could essentially be returned to their initial values at the moment the program started. Furthermore, if this could be done without doing any work on the gas particles, the entropy of the system would decrease without a corresponding increase of entropy in the surroundings. Thus, this would violate the second law of thermodynamics (this will be discussed in further detail in the analysis section). To accomplish this, the user is able to open a gap in middle of the partition by pressing ‘M’ when they observe a particle moving from the right chamber towards the left, and once the particle passes through they can close the gap by pressing ‘N’ before any particles escape from the left chamber.

Now to demonstrate this, I will press ‘F’ on the keyboard to change the color of every particle in the right chamber to blue:

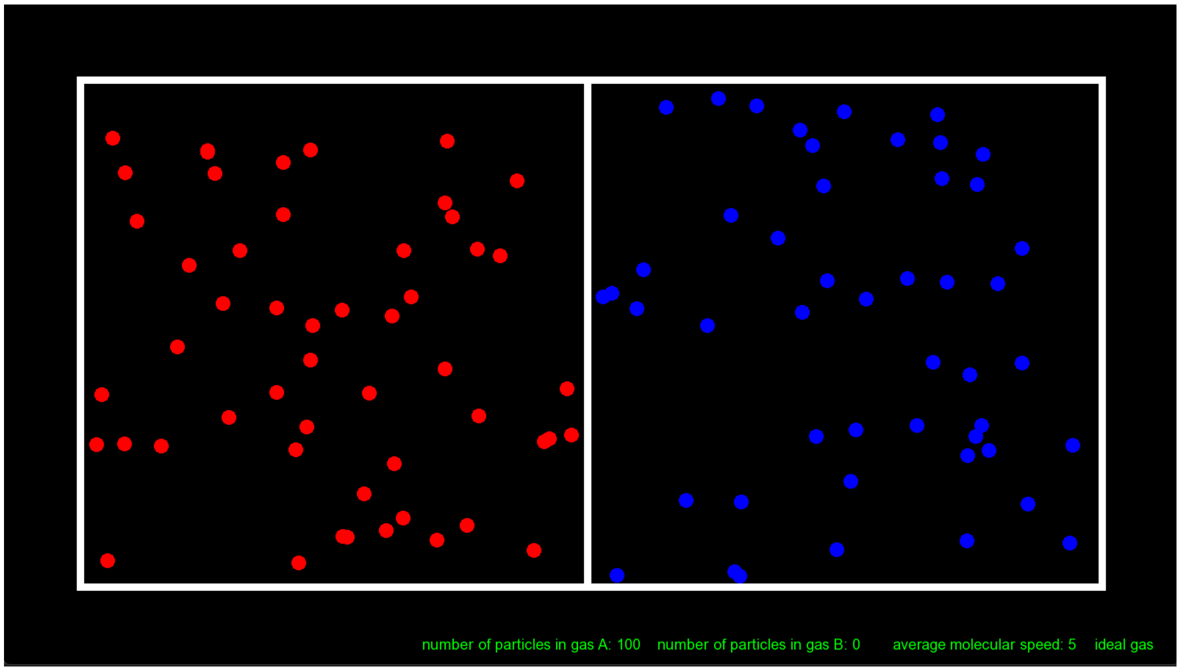

number of particles in gas A: 100
number of particles in gas B: 0
average molecular speed: 5
ideal gas

Figure 5: Color-Distinguished Particles in each Chamber after Partition Replacement

With the particles in each chamber now clearly distinguishable, we have a way of measuring our progress. After opening and closing the gap in the partition several times in the manner mentioned above, we can see that many particles have been transferred to the left chamber, and thus a pressure differential is starting to occur between the two chambers:

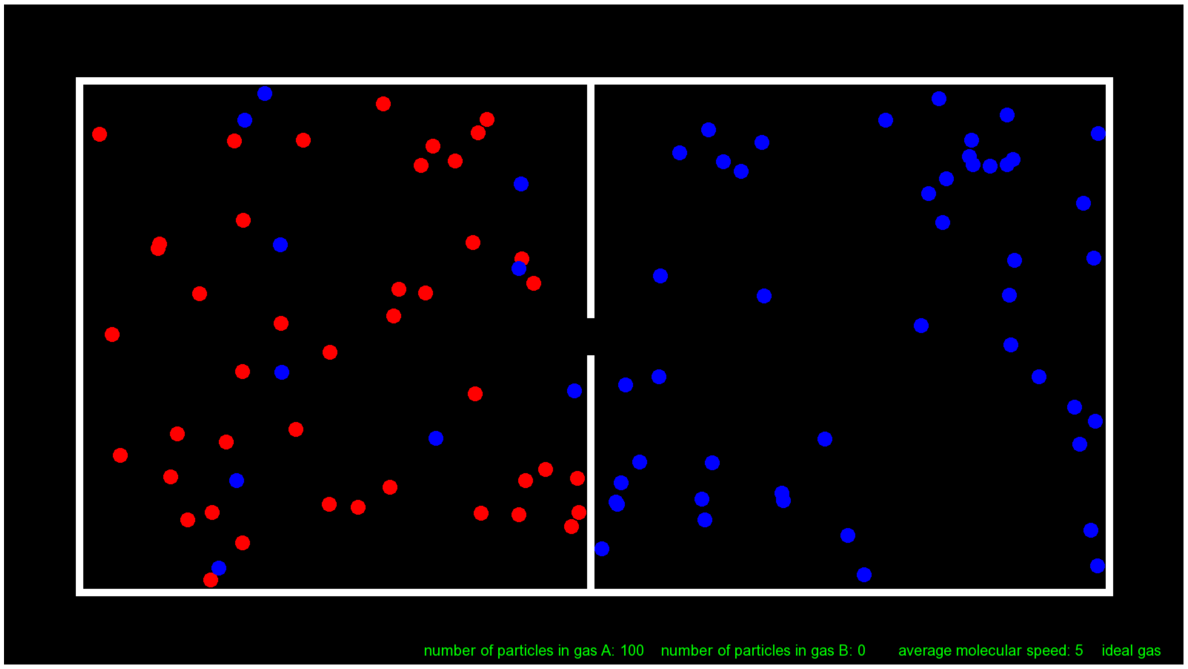

number of particles in gas A: 100
number of particles in gas B: 0
average molecular speed: 5
ideal gas

Figure 6: The Gap in the Partition is Momentarily Opened to let an Incident Particle from the Right Chamber Flow Through to the left Chamber

Therefore, if the user is able to successfully open and close the gap such that every blue particle passes through the partition into the left chamber without letting any red particles passing through to the right, they will have completed their job as Maxwell's Demon and have successfully violated the second law of thermodynamics, right?

III. Analysis: Entropy, Maxwell's Demon, and Information

In the introductory physics courses, we learned several principles of classical mechanics that are always true in nature, including the fact that energy can never be created or destroyed but transferred from one place to another, and that a macroscopic projectile in motion will follow a certain path through time. In studying thermodynamics this semester, we have learned (among other things) that heat and work are both forms of conserved energy, that heat will always flow from a hotter object to a colder one when the two are in thermal contact and isolated from their surroundings, and that while work can be completely converted into heat, heat can never completely convert into work. These last two points raise some pretty profound implications for the behavior of physical systems across time, and introduce the notion of irreversibility. If entropy is seen as a measure of the level of physical disorder, randomness, or chaotic complexity of a system, then intuitively, the flow of energy into a system (via thermal energy as heat or by work being done on the system) would excite its particles and have the effect of raising its entropy. Now compared to Newton's laws of motion, where time can seemingly flow backwards, could this increase in entropy similarly be reversed?

Not without exchanging energy with the surroundings, according to the second law of thermodynamics. In a Joule expansion, a thermally isolated container contains an ideal gas of a certain volume, pressure, and temperature confined to the left half by a partition, and when that partition is removed, the gas expands to fill the entire container. Since no heat is added to the system, the process is adiathermal, but the sudden removal of the partition does not allow for a quasi-static expansion so the process is not adiabatic. Though no work is done on the system nor heat supplied from the surroundings, its entropy does in fact increase, which can explained by the statistical definition of entropy. The entropy of a system is equivalent to the Boltzmann constant multiplied by the natural logarithm of the total number of possible microstates the system can exist in (omega), and as we learned in class, the doubling of the gas volume means every particle then has twice the amount of possible locations it can occupy, which corresponds to a change in omega for the system of 2^(number of particles in the gas), increasing the system's entropy despite no change in entropy of the surroundings. Now since the expansion was not quasi-static, it is implied that the process is irreversible. However, since entropy is a function of state and path independent, we can model this process with a reversible isothermal expansion. Doing so, the net change in entropy of the universe would be zero as the entropy change of the system would be equal and opposite to the change in entropy of the surroundings. Thus, no matter the path taken, the entropy of an isolated system will either increase or remain constant, but never decrease. Or *could* it?

Maxwell's Demon is a thought experiment proposed in the nineteenth century by the physicist James Clerk Maxwell that aims to model such a scenario. In the experiment, an ideal gas is confined to the left half of a thermally insulated container by a partition, and when the partition is removed a joule expansion occurs as explained above and the entropy of the system increases. Now once the expanded gas reaches equilibrium, the pressure of the gas is half what it initially was, volume is double, and temperature is the same. For this increase in entropy to be successfully reversed, the state variables would have to be returned to their initial values, which would require replacing the partition and somehow getting every particle back into the left chamber without doing any significant amount of work. If there were some intelligent being with the processing power required to analyze the positions and velocities of the particles momentarily close to the partition, it could open a roughly particle-sized gap to let particles moving from the right chamber towards the left through the gap, and then close it so that no particles escape from the left chamber. Doing this for every particle in the right chamber would re-confine the expanded gas particles back into to the left of the partition, returning the state variables to their initial values and decreasing the entropy of the isolated system without doing any work. Thus, if the demon could in fact accomplish this without causing a corresponding increase in entropy somewhere else, then this would seemingly violate the second law. Clearly, the demon will have a lot of computation to do, and will need to store information about the positions and velocities of several particles.

Therefore, to determine whether or not this is possible, the connection between information and entropy must first be established. Circling back to the statistical definition of entropy, knowing the Boltzmann constant and the total number of possible states a system can exist in is enough for us to be able to quantify its entropy. Interestingly, a very similar formula exists for quantifying information content. Claude Shannon defined the information content of a probabilistic statement as Q, which he equates to minus 1 multiplied by some positive constant k multiplied by the logarithm of the probability present in the statement. If we set k equal to 1 and define the logarithm as having a base of 2, the information equation has the units of bits. Conversely, if we choose the Boltzmann constant as the value for k and the base of the logarithm as e, the equation for information is very similar to that of entropy in its statistical definition. Furthermore, Shannon showed that if we have a series of probabilistic statements, we could define the average information content by a formula that is identical to the Gibb's expression for entropy in thermodynamics. Therefore the entropy of a physical system can be seen as a measure of the extent to which we can know the configuration of its elementary parts, given our inability to fully comprehend every microscopic detail they contain.

With this in mind, we can better understand what is required for Maxwell's demon to do its job. As mentioned earlier, there would be an immense amount of computation to be done, so the demon will need a pretty large storage device (perhaps a fancy solid-state drive). While my simulation only included 100 gas particles, in even just one mole of gas there would be over 20 orders of magnitude more particles to deal with, so even the best SSD on the market would fill up pretty quickly. That said, the demon would have to clear its storage drive quite often, and it is this act of erasing information that nullifies its ability to violate the second law. Given Shannon's definition of information, the entropy increase from erasing one bit of information would be equal to the Boltzmann constant multiplied by the natural logarithm of 2. While that may not seem like much, the demon would have to clear the memory on its hard drive often enough that the increase in entropy from repeatedly clearing its memory would be greater in absolute value than the decrease in entropy it accomplished by opening and closing its gap in the partition, so the net entropy change of the universe it brought about would still be greater than or equal to zero. While it would certainly be cool to be able to use the idea of Maxwell's demon to build an entropy-reversing machine, the idea nevertheless emphasizes the fact that information is as fundamental a feature of the natural world as entropy is.

IV. Installation Guide

To be able to run the program, Microsoft Visual Studio needs to be installed. Then, click file → new project → empty project (in C++). After naming and creating the project, right click its name in the solution explorer, then click Add → New item → C++ file(.cpp) → Add.

Now the SFML graphics library needs to be imported into the project, which can be done by following the tutorial in the following Youtube video:

https://www.youtube.com/watch?v=VWWSc2nqrEA&t=189s

Also, I imported the arial font into the project to use in the render window. The program can run without it, but it can be downloaded from the following website and then would need to be rerouted according to the file path for where it is saved.

https://github.com/matomo-org/travis-scripts/blob/master/fonts/Arial.ttf

Then copy and paste the code into the .cpp file and press the play button.