**Lab 1**

We made a rock, paper, scissors game against an AI. For lab 1, the AI has a memory of sets of three throws. The first two are important because they dictate what the last one was. When it picks the correct combination of three throws, the third being the most recent player move, it is upped in priority, or is more likely to be seen again. So eventually after playing it enough, it is used to rock paper scissors patterns and is able to efficiently guess what the next throw is by analyzing what the last two were. This is a very simple yet effective way of guessing throws.

i.e. You have been throwing rock rock paper a couple of times now, it will be doing paper when you throw rock and scissors when you throw paper because it’s used to rock rock paper, paper rock rock, and rock paper rock, and can guess it.

**Lab 2**

The AI for lab 2 is still rock paper scissors, but is much smarter. It keeps your last two moves in immediate memory, and calculates based off of the **frequency** of sets of three moves instead of just moves that were thrown like the last one. Instead of having priorities for guessing, it has priorities based on the frequency of which patterns are used. While it sounds similar, it uses these combinations to learn how you throw them.

i.e. Instead of a 1, 2, 3 etc priority on movesets like the last lab, it has a float value for the amount of times it’s thrown, like a .64 probability vs the other combination having a .23 probability. This allows it to adapt to playstyles quicker as the more times it’s used, the frequency values become closer and the AI can make more accurate decisions, even to rapidly changing patterns.