Interaction:

I started by playing the game on chain manually to figure out how the Game dynamics work. I started losing right away but also noticed the end pattern that won it for the AI every time. Now at this point I was sure a forced win strategy for a sequential extensive game like this, probably existed, but doing backward induction on this would be hell, because of the size of the decision tree of this game.
So, I just analyzed backward induction for 2 moved back before the final winning move, and if it was forced (There we're only one good move, I extend the induction analysis to one more move back). It became clear after running enough games that what I was looking for was some kind of symmetric configuration around the glass cell.

Approach:

I jotted down the positions that could led to winning moves and created a json file to populate whenever a board situation appears that can be turned symmetric.

AI Use:
I used some AI for blue coding, and this was predominantly Sonnet. I simulated the 5x8 board and asked two different LLMs to take sequential turns, and every other turn they would have to either add or delete from the chomping_glass_policy.json file, which would ultimately be the strategy we conduct on a per move basis. The more data is in the strategy file the more the chances of solidifying our induction strategy. This file turned out to be huge, and I 'm sure there is some garbage in it but overall it looked fine.
Apart from this I used AI for some debugging. The Rust compiler is goated but sometimes it was ambiguous on lib.rs specifically.

Conclusion:
I tried to make this something that I'd push top production, but I also didn't want to make it so extensive that it'd be very time consuming. The good news is the code haven't lost yet. The onchain AI is really good, in fact the idea of having the strategy file came from my notin that there might be some moves that are fully hardcoded for the AI, or it just has extensive knowledge of the game.

Proof of Victory:

The solescan of one of the wins is:
https://solscan.io/tx/3Sti5P7b3BUgyWDLHEvok8GXySkZpTJy4izoUu31KaKvCcWmHmSWdReYidvW1QRsKJgH63m8tcLi45ue36GbdhcL

I also recorded my code submitting moves an winning and that video is also included.