Stephen Round
Interactive Media Design
Project 2

Roshambo

There's few interactions as simple and well-known as Rock, Paper, Scissors. You have three options, and each one trumps another option and is beaten by the other. In this assignment, I created an environment in which these options live as entities, and will hunt down the one they personally beat and destroy them. When only one type of entity is left, a new round of entities respawns and the cycle continues. It's built in p5.js, a mixture of Javascript and Processing.

When presented with the task of creating a completely autonomous environment with no human component whatsoever, I knew I wanted to make an ecosystem of three separate types of agents that would all interact with each other accordingly. Roshambo, or Rock, Paper, Scissors, was the best way to convey this.

Development was a nightmare. I had to learn a lot about how vectors and arrays worked in tandem. My first obstacle came in the form of not realizing that I was inputting an entire array into an argument that only asked for one specific thing, instead of dissecting the array for the one thing it was asking for. Then came the problem of getting the killed prey to stay killed, instead of just immediately respawning as a ghost that would no longer adhere to the rules I had set in place, and would no longer interact with other objects that haven't been killed yet. Once every object died, they all simply floated around.

Once I overcame these hurdles, development went swimmingly. I had to cut a lot of code to clean it up, but I think I rocked it.

Features, Current and Dropped

Each option of Roshambo is represented by a colored hand icon that displays the option of rock, paper, or scissors. Originally, these icons were literal rocks, pieces of paper, and pairs of scissors, but the paper was hard to distinguish from the background. The hands look cleaner and are easy to tell what will beat what—red scissors beat green paper; green paper beats blue rock; blue rock beats red scissors.

Because the most development took place largely in the weekend and earlier weekdays preceding the due date, there are some features that I would have preferred to code in but couldn't make it. For starters, I wanted a flee option to make the combat between the agents to be more interesting. If a paper comes across a rock, but sees a scissors coming after it, it would abandon the rock to persevere.

Another thing you'll notice by watching this is that sometimes an agent escapes a predator before they're completely shrunk down. This leads to comical interactions where a tiny pair of scissors will obliterate an entire community of normal-sized papers. Perhaps down the road I could push a feature where small icons could be in flight mode instead of fight mode, and would gradually regain size if they stay alive long enough. While they're small, they can't make any size reductions to their prey.

An idea I had way back at the beginning of development was to have a defeated entity leave behind a corpse, like torn paper, covered rock, or shattered scissors. If an agent tried to move across the corpse of another, it would slow them down.

For surviving members of a round (e.g. a pair of papers remains while a new round is spawned) I wanted to incorporate an indicator of their victory, like a small crown or shape above their icon, so you could see which agents have lasted. However, this is probably the least important dropped feature, as I often observe the survivors of a round immediately get devoured when the new agents spawn in.

Finally, a scoreboard didn't make it in. To keep track of each entity's victory, a small score tally of each agent's survival would be displayed at the bottom.

Stephen Round
Interactive Media Design
Project 2

No Humans Allowed

The one stipulation of this assignment that was above all else was that no human component was allowed. There is a small, basically insignificant human component in this program, however. Resizing the window will make the environment larger or smaller, and therefore harder or easier for the objects to find each other. This is hardly a hindrance to the overall autonomy of the program. Resizing the window doesn't affect the AI of the agents.

The Future

This program has a lot of room to grow. All the aforementioned dropped features could be integrated to make a more interesting scene.

I originally mused the idea of neural networks, and I think given enough attention this program could benefit from one (or several). I remember watching a video in which a different neural network was assigned to each team of orbs with sharp ends to fight each other. Each network was trained separately until they were able to combat the other. In the future, I could write a network for each agent and allow them to learn how to interact with the others based on their identity. That's largely ambitious, however.

Conclusion

This assignment was incredibly satisfying to see come to fruition. It was also incredibly frustrating to see not work most of the time during development. In all, there is a lot it could do but it also does quite a lot. It is a dive into the world of autonomous programming and shows how interactivity isn't limited to human input.