

Acknowledgements

This program was written by David Lerner under the direction of Jerry Hughes and was funded by the PSFC from November '09 through January '10. My thanks go out to Andy Russell, Michael Jones, Henry Mei, and Will Barr for some pointers on the quirks of java and David Levonian for helping me hash out a couple of algorithms.

Gameplay

Pressing the number on the numpad with which a magnet is labeled toggles it on or off. Green is on; gray is off. Alternatively, one may use the tyu/ghj/bnm block of letter keys to control the magnets. The qwe/asd/zxc block of letter keys is mapped similarly for use with the button boards. A magnet that's turned on repels the plasma. When the plasma touches the wall, you have lost, and your score stops increasing. The plasma will turn from cyan to red. Pressing spacebar (at any time) restarts the game. You must click the applet before it will respond to any controls. This runs best in Chrome and on displays with vertical resolution of at least 1024 pixels, though the min resolution is easily modifiable by changing SCREEN_SIZE. Pressing the left or right arrow keys increases or decreases game speed linearly.

Underlying physics

The underlying physics of this game/simulation are intended to be qualitatively accurate only. And at that, probably not very. I've used "contour" and "plasma" pretty interchangeably to refer to the colorful object that you try to keep from crashing into walls; please pardon my inaccuracy.

The relative locations of the four plasmaWires in the arraylist called plasma and the one stored in plasmaWire.central determine the values of the parameters k (κ), a , and δ (see Analytic Contour.pdf, taken from Todd et al., Nuclear Fusion v.19 (1979) p.743.). From there, the R and Z (x and y) functions parameterized by θ on $[0, 2\pi]$ give an expression for the contour as it is drawn. The wires are added to the plasma arraylist by increasing θ starting at zero. When a peripheral wire is closer to central than it was, it experiences a repulsive force inversely proportional to separation; this requires a divergent sum of energy to bring separation to zero. When it is farther than it was initially, it experiences an attractive force proportional to separation. Torque on the plasma as a whole is constrained to zero by keeping plasmas at 0 and π constrained to move only in the x -direction relative to central, and plasmas at $\pi/2$ and $3\pi/2$ likewise constrained in the y . The constants of variation are specified in plasmaWire as $c1$ and $c2$ for repulsive and attractive forces respectively.

Each of the five wires in plasma carries equal current and they experience a force inversely proportional to separation with each magnet. This is split into x and y components. For the central wire, both components are experienced. For the others, only the parts of the force that would not cause a torque about the central wire are experienced.

Known bugs

Restarting the game occasionally throws a null thread exception that does not affect gameplay, but it will show up as a println if you are running this in jGrasp etc rather than a browser (this println could be removed but is diagnostically useful and doesn't show up in the game in browser).

If you restart the game multiple times in rapid succession, when you finally let it go, it will give you a plasma which appears to have a high seed velocity. It does not in fact have a higher one, nor is playbackSpeed modified. I could not determine the cause of this bug and could not fix it. It also does not seem to affect gameplay.