**Known Bugs**

**Shader-lang**

* Wireworld
  + Circle drawing tool issues
    - Only white, but not supposed to be
    - Canvas cleared upon play when using circle tool
      * Likely related to only being white
      * White is not a supported color meaning that circle tool will always fail
  + Other notes
    - Not clear how to use this automata
      * Basic explanation?
    - Seems to work
* Brian’s Brain
  + Minor performance issues
* No user tutorial
  + Site not clear to use

**Frontend**

* Generation Counter
  + Does not reset upon selecting new rule
  + Does not stop when continuous play is turned off
    - Only stops once you start to draw
* Cursor
  + Circle tool
    - Glitchy when resizing
    - Wraps around to other side when getting close to an edge
    - Sometimes the cursor does not properly change
  + Changing cursor
    - Have to change the brush type before the cursor changes for the first time
  + Drawing
    - Cannot draw while brush type selection drawer is open - must be closed first
    - Resizing cursor for drawing
      * When resizing square cursor to 1 pixel, in actuality it is 2x2 pixels large
      * When drawing with square cursor on small sizes the shades of the selected color shift
        + Goes from darker to lighter if you redraw over the pixels you have already drawn
  + Bucket fill
    - On occasion requires switching of colors in order to work
    - Does not work the first time
      * Need to change color first then click bucket color
  + Color select
    - Still displays color when its associated state is deleted
  + Exiting and Save Dialog
    - When closing the tab, the save dialog prompt is shown when you are not on the canvas page
* Canvas
  + Resizing the canvas will randomly scale the horizontal axis at a greater rate than the vertical axis
  + Inconsistent CA behavior along top-left to bottom-right diagonal. About 5 cells from either end of this diagonal exhibit this behavior.
  + When state is deleted, the associated color can still be seen on the canvas
* Community page
  + Post sorting functionality does not work
* LogIn/LogOut page
  + Emails are not checked for validity
    - Potential ddos security vulnerability
  + Stay logged in on reboot/refresh/page close
    - Authentication is done via a cookie, if the log out button is not pressed then it does not clear the cookie
* Top Nav Bar
  + Log out appears even when not signed in

**Backend**

* Security
  + No way to prevent a user from making infinite accounts
    - DDOS vulnerability
  + Since the cookie is not deleted on exit, there could be potential session vulnerabilities

**Backend-Frontend Integration**

* Cookie authentication
  + Cookie is only deleted when client makes a request to the server to delete the cookie
  + Meaning that the cookie will not be deleted if this request isn’t made, so if a user does not press the logout button the cookie will not be deleted
    - However, the user will be logged out since the token in the cookie will expire within an hour
  + The site does not indicate to the user that they are logged out once the cookie expires, so the frontend will still show the user as being logged in
* Post loading on community page
  + Attempting to load a post the first time will result in a blank canvas
  + But when attempting to load the same post a second time, the post loads with the proper state
    - Likely related to issues with loading the state by replacing the previous state with the new state

**Database**

* Setting up database
  + Depending on the operating system used, there is a completely different set of required steps to install and run the createDatabase.sql script

**Performance**

* Firefox
  + Firefox webgl performance is very bad
    - This is a known issue with the browser
* Brave (built off of chromium)
  + Brian’s brain tested with no lag spikes
* Rule Complexity
  + Creating a large or unoptimized rule will result in performance issues
  + Due to variety of factors:
    - P5 is a high level language
    - WebGL performance varies with browser use
    - The hardware of the client can heavily impact the performance
* Canvas Size
  + Creating a large enough canvas has the likelihood of performing poorly or freezing the site client side

**Compatibility**

* Tested to work with up to date versions of Windows and Linux on firefox and chromium based browsers
* Likely compatibility issues:
  + Operating systems that do not feature an up to date or recent version of chromium
    - Windows 7 and 8 likely supported (depending on version of chromium used)
    - Pre Windows 7 likely not supported
  + Gpu and OS must support webGL 2
  + Cpu must meet minimum requirements for chromium
* Known compatibility issues
  + Firefox performance is poor and will render the site unusable for less powerful computers
    - Issue with firefox itself, no way for us to fix this

**Testing Methodology**

Testing was conducted manually across different browsers, operating systems, and hardware with a focus on targeting the systems and software that the majority of internet users are likely to use.

During development, pair programming, peer review, and continuous testing were used frequently whenever possible considering time constraints. Pair programming was conducted via in person coding sessions. During the pair programming sessions, team members would alternate between writing the code and reviewing the previously written code. By alternating the person reviewing the code in this manner we were able to find more bugs via differing perspectives. Modules were tested for functionality as they were being written and were rewritten as necessary to fix issues that came up. Additionally, interfaces were tested for functionality as soon as they were in a state where it was possible to do so.

When testing manually, code was tested with an expected functionality in mind and if it did not perform according to this expected functionality, this would be addressed instantly.

**Testing Limitations:**

* Time Constraints
  + We were unable to utilize Unit Testing as we decided to prioritize coding time to ensure that the product would be in a useable state
  + Instead, we opted to use less comprehensive testing that could potentially result in more bugs slipping through
* Testing pool constraints
  + Testing was only conducted by the immediate members of our team
  + Outside testers could potentially find issues that may have been overlooked due to the relatively small team size
* Useability testing constraints
  + Since testing was only done by the members of the team who are familiar with cellular automata and how to use our site, it is impossible to know how the average internet user would react to the UI and general useability of the site

**Conclusion:**

In spite of testing limitations and time constraints, the main features of the site that we wished to implement are all currently working to a level where a user could reasonably use the website for the purpose of creating and sharing their own cellular automata rulesets.