Ross Lagoy

Lab3. BCB 502

02.13.15

github.com/rlag

openprocessing.org/user/46418

**A list of interactions your simulation currently supports, and why they support (or don’t support) user understanding:**

Currently, my Game Of Life simulation supports the following listed interactions, which I define as opportunities for the user to interact with the environment by providing physical inputs: key commands and mouse commands. The user can apply an infection at the start or in real-time during the simulation. This helps the user discover a new type of cell interaction and begin to discover which rules are applied to the simulation. Being able to pause, reset, and clear the simulation is valuable to understanding trends, which are quantified and plotted in real-time. Additionally, being able to increase and decrease the frame rate provides the user an opportunity to select an update speed that allows exploration into the applied rules (live or dead state) and trends. Finally, I added in a live cell color selection, which provides the user to customize the simulation environment, which could indirectly support user understanding. If the user is able to choose which color is used for the simulation, they may enjoy the experience more—they may not be understanding the details directly from being able to change colors, but they might have a better experience interacting and observing the screen.

1. Cell infection (mouse press).
2. Pause, reset, and clear (key press).
3. Frame rate selection (key press).
4. Live cell color selection (new) (mouse press).

**Three (or more) interaction opportunities you identified from your analysis, this lab, and the handout. Tell me: Describe what the user needs to be able to see/do; How you identified the problems; What interaction technique(s) can fill them.**

One opportunity I identified from this handout is the possibility of *searching and filtering* interactions within the cell-arena. At first glance, the simulation is overwhelming but I’ve attempted to make sense of it with bar graphs and line graphs. Yet, I believe it would be useful if the user could hover their mouse over the changing cell-arena, filtering the interactions between live and dead and infected neighbors over time within a range, and plot the rate.

I identified this problem after I plotted the complete arena of cell state over time. This provides a nice quantitative and trend visualization of the otherwise complicated behaviors in the cell arena. For example, the user may not notice the exponential or logarithmic decrease and increase in live vs dead cells at the start—the time plot brings this relationship to attention.

If the user could hover their mouse over the arena via a 50x50 pixel area grid (or customized area) it would capture the relationships (dying, appearing, and infecting cells) occurring within that selected area. This idea is illustrated in the ‘prototyped’ example below.

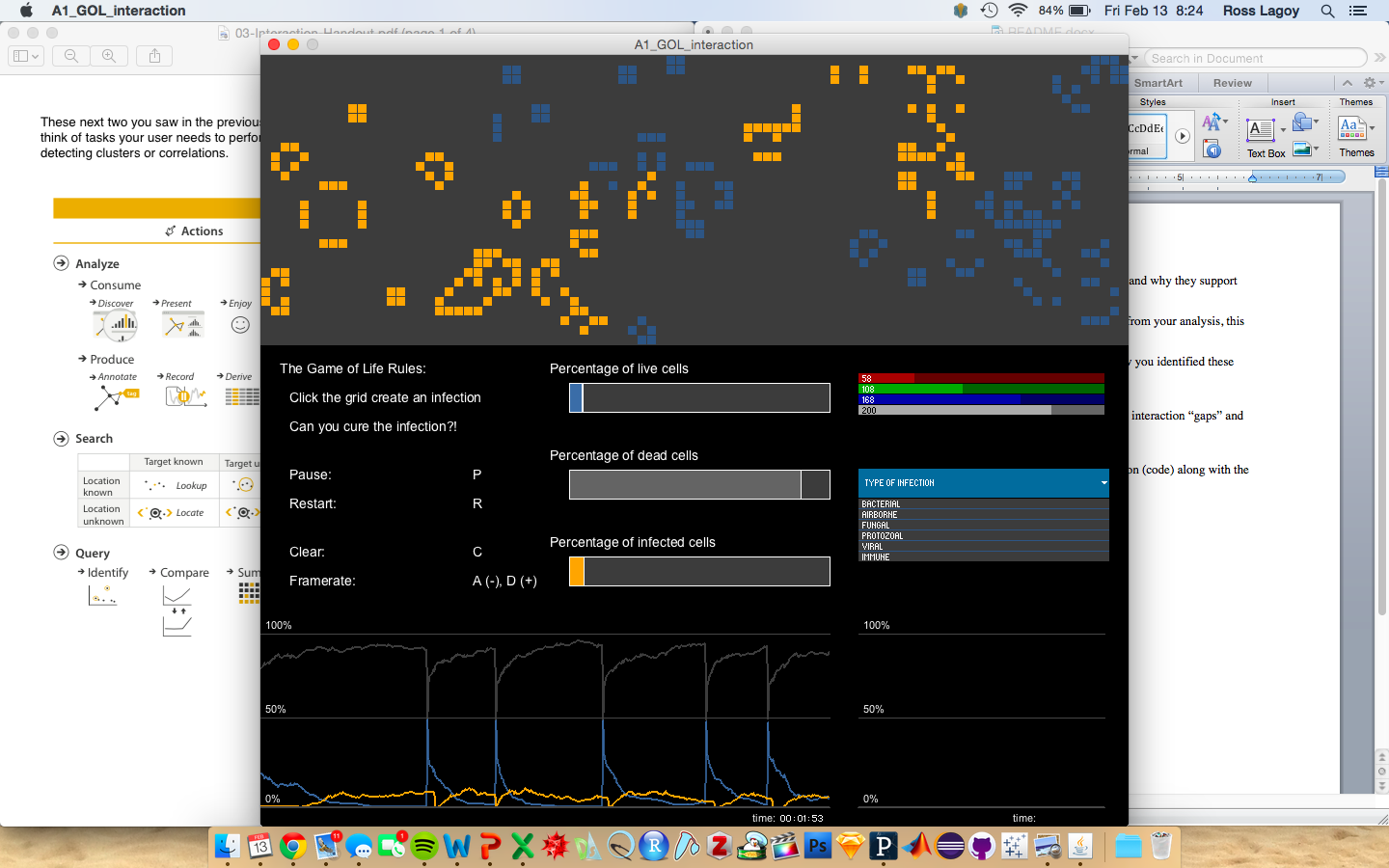
I identified these opportunities by watching Prof. Ryder use my simulation, as well as my classmates, Jason and Victor, in class during the lab:

1. **Set initial cell state(s) by user:** this could be implemented in a GUI where the user could type in an int between 0-100 (percent) for either live or dead cells. If a number was typed in for percent live, the dead cell number would be calculated, or vise versa. This would be doable using a controlP5 package. This could support user understanding by exploring the differences in initial cell state rate of change. Additionally, users could choose their desired live cell color (implemented) and frame rate (implemented). Cell color may most likely be an aesthetic and personalized feature, whereas a slower frame rate could allow the user to observe the applied rules over real-time. The live-dead rules are difficult to interpreted at a normal (30) frame rate, but if slowed down the user could observe the trend and predict the next frame.
2. **Highlighting a region in the cell arena that plots a subset of the array:** this idea is discussed above and implemented as a prototype for this lab below. This would help filter the larger, summated, cell state line plot graphic I have. This would allow the user to explore a determined range of cell states to investigate cell interactions.
3. **A graph that plots a weighted percentage:** I like showing two different graphical representations of the cell state (bar and line graph) because they portray different real-time information: quantities and rates. However, I think another plot could be added that overlays this kind of information, like a stacked bar plot, which would concisely summarize cell quantity in one visual, instead of three separate bars.
4. **User selected infection types:** I added a drop down GUI from the controlP5 package that is not functional now, but represents the opportunity for a user to select a different kind of infection that models biology. Each infection could be a different color with different interaction rules for spawning and death. If this feature worked, I could imagine adding these ‘new-infection’ trend lines to the cell state line plot and bar graphs, which could compare infection properties from one another allowing the user to understand the different applied rules and further model biological interactions.
5. **Plotting and quantifying individual cell state (cell tracking): I** think it would be cool to visualize individual cell state over time. I could imagine the user selecting an individual cell within the grid and observing its live, dead, or infected cycle over time—and/or predicting its chance of surviving, dying, or becoming infected given the real-time changing parameters. A percentage could be displayed through on-screen text that quantifies its cell state.
6. **Filtering the dashboard:** I think it would be useful for the user to be able to filter and arrange the dashboard as they wish, i.e. removing the rules (as they become aware of the key presses), moving the sliders, scaling the cell state line plot, removing the bar plots, etc. This could allow for a more personalized experience. The problem now could be overcrowding of data representation, which this feature would correct.

**From your interaction opportunities list, choose one of the interaction “gaps” and add it to your simulation or prototype it.**

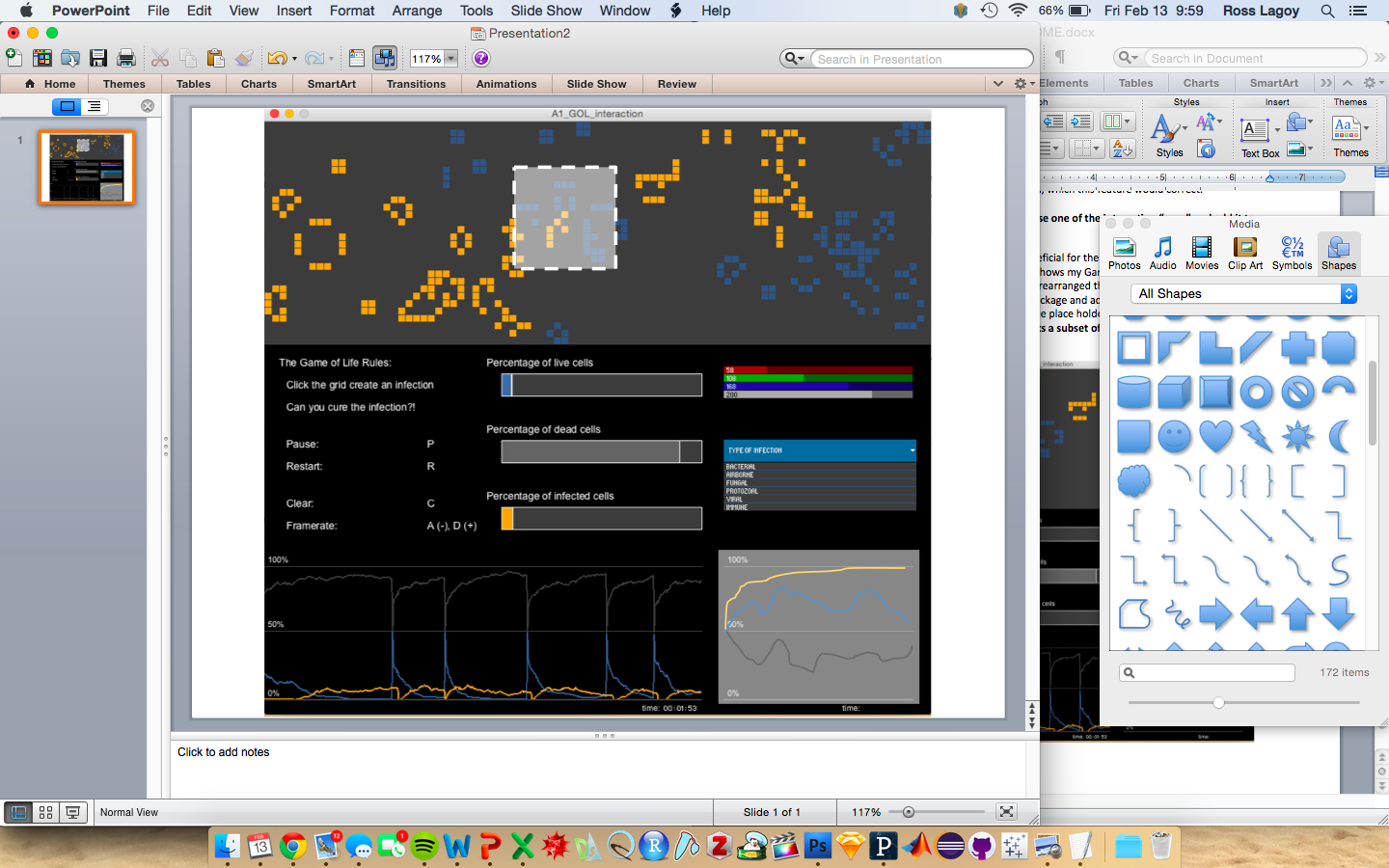
As mentioned above, I believe it would be beneficial for the user to be able to filter the trends in the simulation. The below figure prototype shows my Game Of Life v2.0 with an updated color scheme, fixed bugs, and cleaner layout. I rearranged the rules, the plots, and the window size. I also learned how to implement the controlP5 package and added a time, drop down menu (not yet functional), a color selector, and a prototype place holder for my chosen interaction gap: **highlighting a region in the cell arena that plots a subset of the array**.

Game Of Life v2.0 Without Prototype:



This is the current layout and updated game of life without the prototype, displaying all current updated features as discussed.

Game Of Life v2.0 With Prototype:



This is the prototype that represents what I imagine my filtering feature to represent. You can see a selected area within the simulation (represented by the gray box with a white dotted line) and in the bottom right (represented by the gray box quantifying the trend within the simulation selection). The interactions that travel through this statically highlighted region would be updated over time, allowing the user to focus on a selected range.

This interaction, in combination with the others already available in this version, will allow the user to understand the rules of the game and explore the otherwise ‘difficult to interpret’ trends.