GUI for zebrafish application

Joost Nibbeling, Haoran Ding

LIACS, Leiden University, Niels Bohrweg 1, Leiden, The Netherlands
j.h.nibbeling@umail.leidenuniv.nl, bitdhr@hotmail.com

supervised by Maarten van Hees
vanheesmaarten@hotmail.com

To make studying the infection process of the Mycobacterium marinum bacteria in a zebrafish easier, a model of this infection process was created. To make using this model easier, a visualization was made. This visualization however, had a very limited user interface. In this paper we describe the creation of a new graphical user interface that improves the experience by giving making all functionality accessible on this interface. We also extend the visualization with new capabilities.

1 Introduction

Infections in zebra fish are used to study the infection process in humans. One such infection is the Mycobacterium marinum infection. To aid this research, this infection was modeled as a petri net. Later, a 3D visualization of this model was made. However, the interface of this visualization is very limited. Our goal was to improve this GUI of this application, to make it easier to use. In this paper we will describe the process of creating this GUI, our evaluations to the new interface and finally our conclusions of how well we succeeded in our goal.

2 Petri Net Model and Visualization

To study a tuberculosis infection in humans, researches can make use of a similar infection in zebrafish: the Mycobacterium marinum infection. Because it is inconvenient to keep infecting zebrafish, a model of this infection was created by R. Calvalho. This model took the form a petri net[1]. This is a mathematical model that might be difficult to understand for biologists who might be interested in this model.
Therefore a visualization was created by Jeroen van den Heuvel to make the model easier to understand for these biologists.

The resulting visualization however had a very minimal user interface in which all manipulation of the visualization is done through keyboard inputs. The visualization also requires an input file that results from the petri net model. The exact file name

needs to be entered at startup. This is inconvenient. To make the use of the visualization more user friendly, a graphical user interface is required. Adding some new functionality is also desired to improve the usability of the visualization. Adding this GUI and functionality is the aim of this project. This brings us to the following research question: Has our new user interface and functionally improved the end user experience?

User analysis

The original petri net model was used created to study the effect of the Mycobacterium marinum infection in a zebra fish. These users are biologists. Because they are not mathematicians, the original visualization was created to aid them in understanding in understanding the model. The result however, had a very limited command line interface, that some might find cumbersome to use.
As we cannot assume a level of computer proficiency, a clear and intuitive graphical user interface is necessary. However, some might like to use keyboard shortcuts as they are often faster than moving your cursor to a button and pressing it. For these people it is necessary to keep the ability to use the application with just keyboard controls.

3 Designing the GUI

To create a good UI it is preferable that all existing functionality from the original visualization is accessible from elements of a GUI. This way the user does not have to remember all the key bindings. The already existing functionality that needs to be available in the GUI include pausing, resuming and starting the simulation, moving the camera, toggling the bounding boxes and fish model, adjusting the speed of the simulation and selecting the input file.

The original visualization also has the ability to manually input some data, but this was so limited in that we did not re-implement this feature and focused on the others. The old visualization also does not allow for easy replay of the infection process, except for completely resetting it. As this is inconvenient during research, we wanted to add the ability to move back and forth through the simulation of the infection process. Finally we needed to add more models that represent zebra fishes at different stages of development and a way to switch between these models.

For our user group, we need a clear user interface. Therefore, for every piece of functionality we need an interface elements that makes sense for this interaction. Furthermore we wanted to find as many appropriate icons as possible as icons are clearer than just text.

For the pausing, resuming, resetting and the back and forth buttons, we looked at existing video player programs. These use icons on their buttons that have similar functionality as ours and that should be familiar to most people.

The camera originally used a combination of mouse and keyboard inputs. We wanted to be able to do this with just the mouse. As this involved manipulating a 3D object, we looked at how this is done in other 3D modeling programs. As the model is not very complex we went for this simple solution also used in Microsoft 3D builder. We also added explicit buttons to move the camera.

We tried to group these buttons in a meaningful way and placed them underneath the visualization. Other less used functionality we moved to a menu that can be hidden to avoid cluttering the interface. This menu can be accessed with a button from the main interface.

The resulting interface has the following functionality and interface elements shown by default:

- Select file to visualize: Buttons with file icon.
- Pause/Resume visualization: Buttons with pause/resume icon.
- Resets visualization: Button with reset icon.
- Move through simulation: Buttons with resume/fast forward icons.
- Show/Hide extra options: Checkbox
- Camera Controls:
    - Move/Rotate: Buttons with arrows indicating direction
    - Change control mode: Radio buttons to change if above buttons move or rotate model
    - Zooming: buttons with magnifying glass icons.
    - Camera also controllable with mouse: dragging with left mouse button click moves the model, dragging with the right mouse button moves the model and the mouse wheel can be used to zoom in.

And in the options menu we have the following functionality:

- Toggle bounding boxes: Checkbox
- Toggle fish model: Checkbox
- Change simulation speed: Slider
- Change fish model: Radio buttons

Most of the old key bindings were retained, with some being adjusted if we thought it necessary. New functionality was assigned new key bindings.

The final resulting interface design with menu is shown in figure 1 and with menu in figure 2.

For the usability specifications we decided to focus on subjective measurements. For this we use a questionnaire that rates the system on the Likert Scale. Our target were results of 2 or 4 depending on how the questions were worded. Objective measurements turned out to be harder, as we had no measurements of the old version to compare the new results with and trying to set objective goals amounted to guess work.

4 Results and Evaluation

For the evaluation we asked a group of people to use to first use to old version without GUI and after that the new version. We had people try and do all the different interactions. After that they filled out a questionnaire that had statements in them about the system and they were asked how much they agreed with that statement using the Likert scale. The questions and average results of all users can be found in table 1. After that we asked them if they thought the new version was better than the old one and what they thought about how we could improve our GUI. They all answered this last question with yes.

Looking at the results of the questionnaire we are very close to the goals in the usability specifications. We intended the GUI to be intuitive and easy to understand. The users indicated that this was indeed the case with them indicating that the system was not cumbersome to use and not too complex. We did not meet the goal in regards to how easy the system was to learn, although this might be because what the visualization shows is difficult to understand and this is not something we can solve with a better interface. We also missed the goal on how well the interface integrates all the different functionality, although not by much.

To improve on the interface we were given a few solutions. Most of these amount to some repositioning the interface elements. These suggestions were:

- Assign camera movement to arrow keys on keyboard. This came from a left handed person that preferred to use the arrow keys instead of W,A,S,D. As the arrow keys were unassigned this is an easy feature to add.
- Integrating the start and pause button as one. This makes sense to do as the pause button serves no purpose when the simulation is paused and the start button serves no purpose when the simulation is running.
- Prompt for an input file immediately on startup. This also make sense, because the application does not do much without an input file.
- Create separate buttons for rotating. The initial setup was made to avoid cluttering the UI with more buttons, but this setup was preferred by the users.
- A timeline to see how far in the simulation you are. This was intended originally, but there were implementation difficulties, so this will not be implemented for now.

- The speed option was deemed important enough to give it a place in the main screen to avoid having to continuously go to the options screen.
- Add an explanation about the simulation itself. While possibly useful, our project focuses on the GUI and not on explaining what is simulated, so we will skip this one for now.
- A reset button for the camera. This can be useful if you've accidentally moved the model of screen an can no longer find it.

5 Conclusion and Discussion

The question we asked in the beginning was if our new interface improves the experience of the users. As all users answered this question with yes, it is safe to assume we managed to do this. As the original application had no graphical user interface at all, reaching this goal was probably not the hardest thing to do. The users also indicated that our interface was easy to use, but that we can still iterate further upon the interface design to improve it.

6 Future Work

Future work could include evaluating the interface again with all the suggestions implemented. By improving the interface using the results of these evaluations we could keep on iterating on this design.
Another possibility is to keep extending the application with new features. For example the manual inputting of data could be brought back and improved. If the users would like to see more features these can also be considered.

References

1. R. V. Carvalho, J. Kleijn, A. H.Meijer and F. J. Verbeek. “Modeling Innate Immune Response to Early Mycobacterium Infection”. *Computational and Mathematical Methods in Medicine*. Vol. 2012. 2012
2. J. van den Heuvel. “Dynamic 3D visualization of Mycobacterium marinum infection modelling in zebrafish using the Petri Net formalism.”. 2014

Appendix

Figures

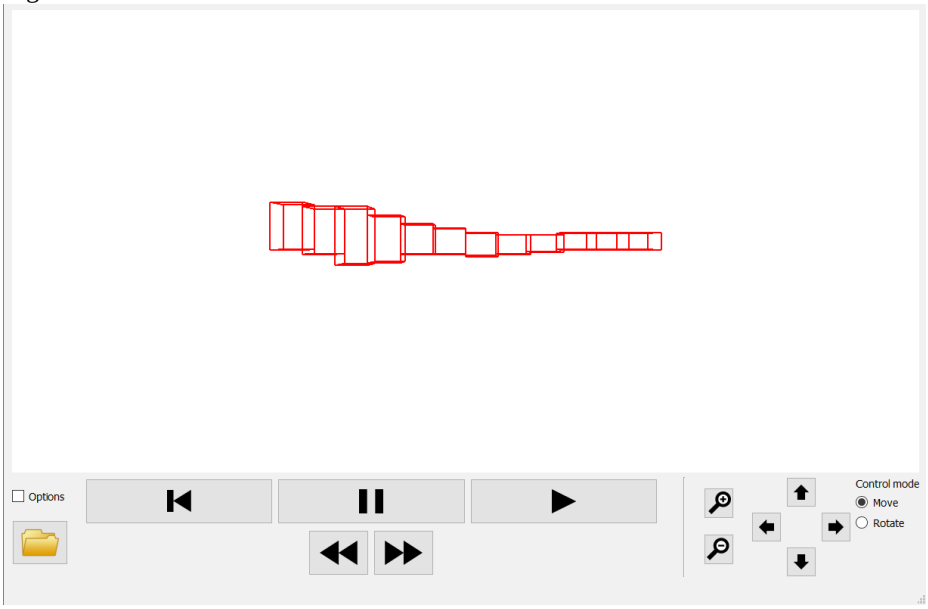

Figure 1: GUI without menu

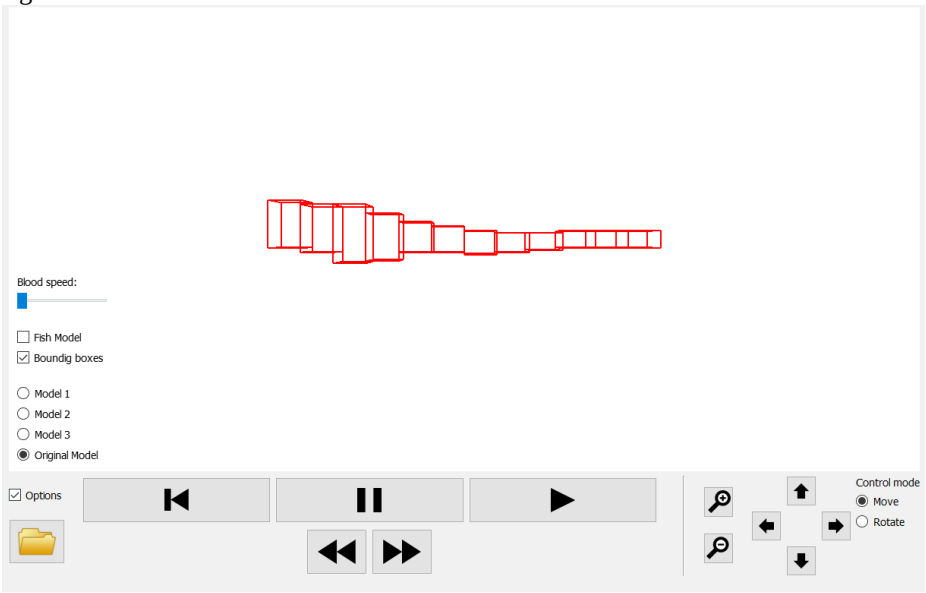

Figure 2: GUI with menu

Tables

Question	Average Result
I found the system unnecessarily complex	2
I thought the system was easy to use	4,4
I think that I would need the support of a technical person to use this	2
I found the various functions well integrated in the system	3,8
I thought there was too much inconsistency in the system	2,2
I would imagine most people would learn to use this system very quickly	4,4
I found the system very cumbersome to use	1,6
I felt very confident using the system	4
I needed to learn a lot of things before I could get going with this system	2,6

Table 1: Average results of questionnaire (Likert Scale)