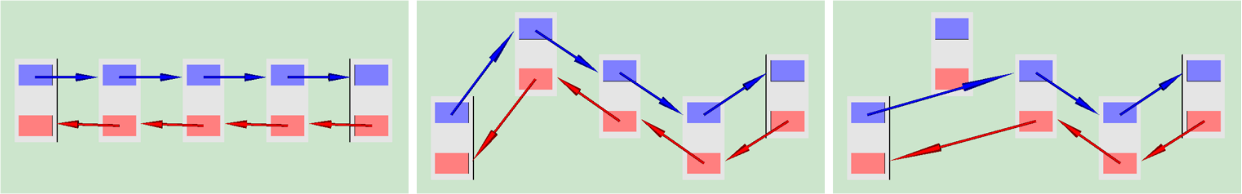
CSCI 4972 Introduction to Visualization
Assignment #3 Interaction with a Process
Due: Tuesday September 21st at 11:59pm

Interaction is a important mechanism for visualization, which can enhance the engagement and thus the effectiveness of visualization for communication and education. By “kicking the tires” of a visualization, the user can formulate questions about the world and discover the answers through exploration. The example below is a visualization of the classic doubly-linked list data structure and a simple operation that may be performed on the data. Using the left mouse button the nodes can be moved around in 3D, but the connections between the elements remain. Furthermore, by clicking with the right mouse button a node can be unlinked (deleted) from the main chain and the arrows on either side automatically reconnect to maintain the consistency of the overall data structure.



To implement the interaction and visualization above, we need to collect information from the user, in the form of mouse clicks and/or key presses. Furthermore, in many cases we want to map the 2D screen location of the mouse click to the object and/or 3D location of the click. VTK provides this functionality and uses graphics hardware acceleration to implement “picking”, which can be used to determine which actor in the scene was “clicked”.

To begin this assignment, familiarize yourself with the mechanisms of interaction (a.k.a. user input) in VTK including: mouse events (left/middle/right & press/move/release), key press/release events, callbacks, and how to subclass the camera and/or actor interactors. Furthermore, study examples of using the VTK pick infrastructure to grab world coordinates of the click and alternatively of retrieving a pointer to the object that was “picked”.

Once you have explored the basics, the challenge is to select an interesting “process” to visualize and design a simple interaction that will enhance the visualization of that process. You are especially encouraged to tackle a visualization of an abstract computational process (as in the example of linked list operations above). Another type of process that you could choose to visualize would be the gameplay for a simple board game, for example, the game of checkers. For a given board state, the user may click to select any of his/her pieces and then the legal moves (if any) for that piece are highlighted and then chosen with a second click. The user interaction may cause other side effects (e.g., capture of the opponents piece), and if you are especially ambitious you may want to look into VTK animations to present these results.

A word of caution: Designing and implementing intuitive, easy-to-use, and compelling interfaces is extremely time consuming! Keep in mind that this is a one week assignment and you will have further opportunities to incorporate interaction into your assignments and final project.

Assignment Requirements

- Select a “process” to visualize (abstract computational processes are encouraged!)
- Design a minimal visual representation and interaction scheme for the process
- Create a VTK program for the visualization and interaction
- Write a couple paragraphs describing the purpose, design, and implementation of your visualization (use the README.txt template)

Optional Learning Objective

- Learn to use a video screen capture software (examples include camtasia, http://www.debugmode.com/wink/ , http://ostatic.com/blog/six-top-screen-capture-tools-for-linux , etc.)

Grading Criteria

(5 pts) Creativity of selected process and design of visualization and interaction

(5 pts) Written description and overall effectiveness of the visualization & interaction scheme for communication

(10 pts) Technical content (e.g.,mouse event and/or keypress events, subclassing interactors, transformations, and possibly animation)

How to Submit:

- Post at least two screenshots of your visualization and a short paragraph describing the purpose and design of your visualization interaction on LMS.
- *Optionally,* post a video of your interaction to LMS.
- Submit the source code (and any necessary data files) for your program to the CS homework server.