

# GeometryEditor

Xun Lai

Oct. 18, 2006

# Authoring Supports Implemented

- **Arbitrary Drawing**
- **Drawing primitives:** Making it simple to create basic geometric shapes such as points, lines (segments, rays and vectors), circles (ellipses and arcs), polygons, conics, etc
- **Geometric object construction:** Constructing a new geometric object subject to mathematical relations with existing objects. For example, creating a line parallel to an existing line and through an existing point.
- **Measurement:** Measuring length, slope, radius, distance, area, circumference, perimeter, angle, and coordinates.
- **Animation:** Moving and changing objects to illustrate and to demonstrate.
- **Iteration:** Repeated execution of user commands.
- **Calculation:** Creating and evaluating mathematical expressions based on existing measurements.
- **Graphing:** Plotting points and function graphs in coordinate systems.
- **Defining Macros:** Grouping several steps into one command.
- **Defining GUI Operations:** Creating a variety of buttons, user inputs, and tables in a manipulative.

## Authoring Supports to be Implemented

- **Geometric transforms:** Translation, reflection, dilation, and rotation of objects.
- **Loci and Envelops:** Constructing loci of moving points and envelops of moving lines.

# Major Algorithms

- eventLogic, statusObject
- selectionLogic
- macro
- objectManager
- iteration
- history: unlimited redo/undo
- json
- synCopy

# System Files and Classes

- graph: coordinateSystem
- calculator
- window manager
- utility: BFS
- valueSrc: math expression
- animation
- menu/toolbar manager
- ruler/protractor

# Current Status of GeometryEditor

- New classes of geometric objects can be added without significant modification of the above algorithms and classes
- No longer use XUL for menus/toolbar and dialogs. (still thanks to XUL that helped me focus on the authoring algorithms)

# GeoSite

- GeometryEditor for editing geometric manipulatives
- A modified version of FCKeditor for editing the html sections in a page
  - a well developed HTML composer
  - able to create math formula to relate html sections and manipulatives
    - quantities, buttons, inputs
- Able to dynamically load manipulatives from another GeoSite

# GeoSite Demos



# Integrate GeometryEditor into your Application

- No difference between viewing and authoring any more
- To create an instance of GeometryEditor

```
editor = GeometryEditor.newInstance( instanceName, divId );
editor.systemBasePath = "/geosvg/geosite/system/";
editor.width = 700; //optional
editor.height = 520; //optional
editor.menuset = { "Edit": ["Undo","Redo"] }; //optional
editor.toolbarset = []; //optional
editor.dataValue = "... ";
editor.dataURL = "http://geosite/username/path/manipulativeName";
editor.create();
```

- Data describing the manipulative has its own width/height/menuset/toolbarset values. They will be overwritten by the values you set

# Integrate GeometryEditor into your Application (cont.)

- Only `editor.dataValue` or `editor.dataURL` should be set. If both are set, GeometryEditor will use `dataURL`
- To update the content in the editor: do the same thing as creating, and then call `editor.update()`
- APIs:
  - `getDataValue()`  
//width/height already in the data; no longer need to get them separately

# Integrate GeometryEditor into your Application (cont.)

- **Callback Functions:**
  - After create(): if defined, user function `GeometryEditor_OnEditorLoadDone()` will be called
  - After update() using dataURL: if defined, user function `GeometryEditor_OnDataURLLoadDone()` will be called
- Both callback functions will have the `editorInstance` passed in

# Difference from the old GDrawing Library

- No difference between authoring/viewing
- GeometryEditor won't open editing window for you
  - Write your own html file with the editor
    - Example: the manipulative editing window in GeoSite
  - Write codes like `window.open( ... )` yourself
  - Use the API `getDataValue` in the child window and pass the data back to the opener yourself

# Future Work – GeometryEditor Authoring

- Transform of objects
- Locus
- Save macros as separate files via AJAX

# Future Work - GeoSite

- AJAX will be used to save a particular section or manipulative instead of refreshing the whole page
- Measurements in manipulatives and quantities/inputs/buttons in an html section act like input/output interface. I will have a more complete design of input/output interface for
  - GeometryEditor generated manipulatives
  - SVG/Flash/Applet-based manipulatives (atomic)
  - HTML-based manipulatives (composite)
  - HTML-based sections
- Web services to supply the above three components
  - GeometryEditor generated manipulative is done