

Software Engineering

Project 1 Software Maintenance

Team 14

Aditi Mavalankar (201201049)

Vanshika Srivastava (201201065)

Ramakrishna Vadla (201350850)

Project : Rasterization Polygon

Objective of this experiment is to understand the steps of filling a polygon in a 2D frame buffer. Given the vertices of the polygon, the experiment outputs a filled polygon. Here we have demonstrated the steps of Scan Line Polygon Fill algorithm.

Milestone 1

- Test Cases for existing lab

The application was tested on the following parameters and a total of about 15 test cases were used.

- Usability
- Responsiveness
- Intuitiveness of the GUI
- Data type verification
- Out of range input values, and others

Milestone 1

- Defects / Issues

A total of about 4 issues were identified

- The application doesn't run in as a stand alone. It gives access control exception, java.io.FilePermission error. Looks to be a applet error.
- The invalid input co-ordinates enter into the "Enter co-ordinates" text box doesn't give the error message but simply ignores it. It should give the proper error message
- When clicked on "Start Experiment with Default Values" it displays the "Enter" button behind "Next iteration" button. The "Enter" button should not be displayed

Milestone 1

- Quality metrics of the existing code base
 - Lines of code : 980
 - Cyclomatic complexity : 5
 - Coupling : Monolithic design
 - Cohesion : Low
 - Maintainability Index :
 - Halstead Complexity Measure Volume: = 593.6
 - $MI = 171 - 5.2 * \ln(V) - 0.23 * (G) - 16.2 * \ln(LOC) = 25.114$
 - Web page loads in moderate time.
 - Web page loads in moderate time.
 - Web page responds in moderate time. Could not test the experiment loading time due to bugs in the code.

Milestone 2

- Identifying
 - code smells
 - Anti-patterns
 - design flaws
 - Monolithic design
- Java to JavaScript conversion
- Integration and Testing
- New Quality Metrics

Milestone 2

- Code Smells
 - Unnecessary comments
 - Long Parameter List
 - High conditional complexity
 - Combinatorial Explosion
 - Large Class
 - Very uncommunicative names
 - Inconsistent names
 - Data Clumps
 - Inappropriate Intimacy
 - Middle man

Milestone 2

- Anti-Patterns
 - The Blob
 - Single controller class
 - Lava Flow
 - Large commented-out code with no explanations
 - Lot's of TODOs
 - Golden Hammer

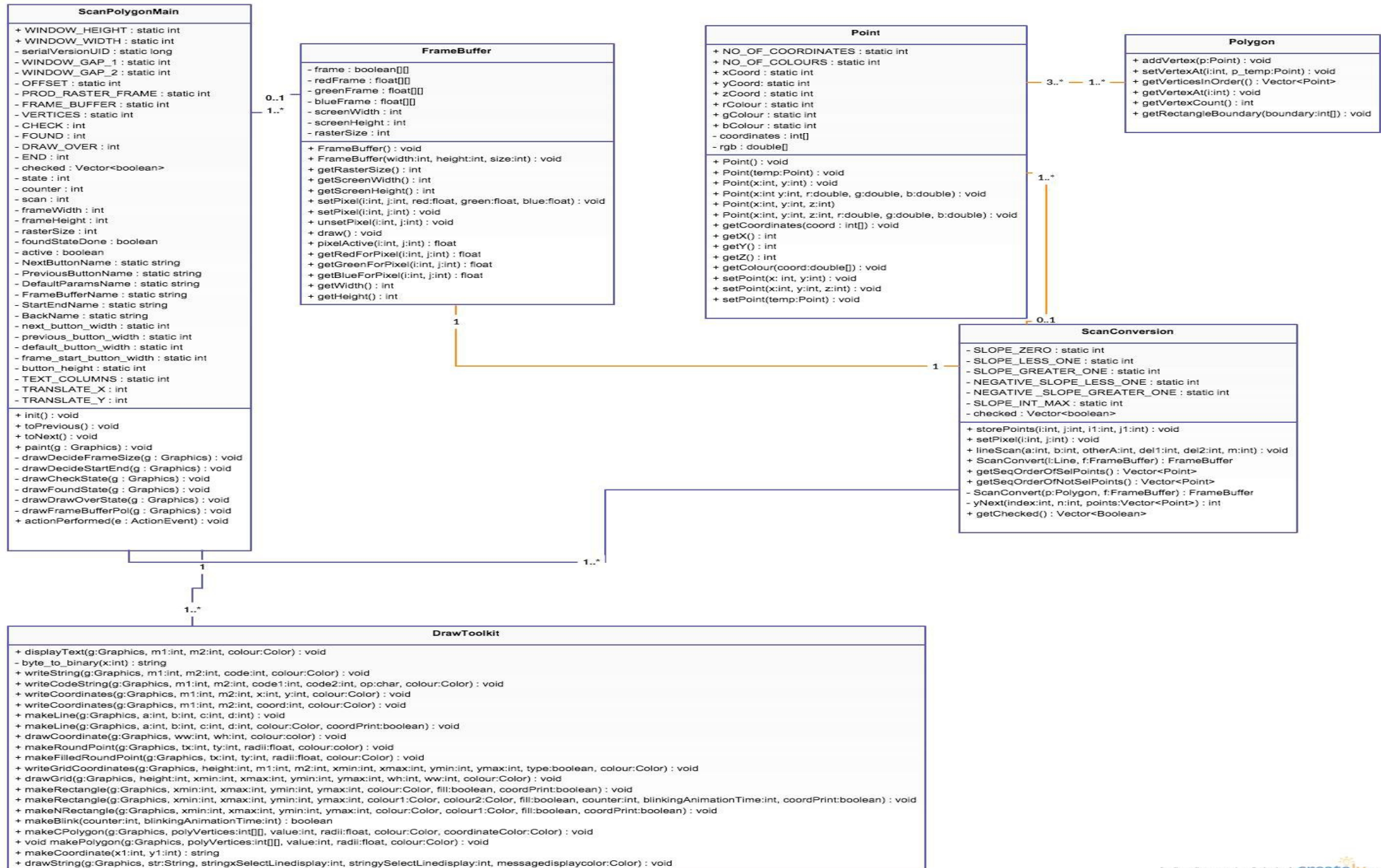
Static Code analysis and metrics

- Re-written entire code using Java script and HTML5
- ▮ Can be executed off-line.
- ▮ Used Builder Pattern while refactoring

Quantitative Measures

Metrics	Ideal and Suggested Value	Javascript Value
Cyclomatic Complexity	01-05	03 (Average) 06(Max)
Lengthy Methods	15-20	16(Average) 31(Max)
Number of Parameter	05-10	0-4
LOC		250

UML Class Diagram for Java Code



Thank You