

Sprint Review Meeting Minutes

Attendees: Alex Castaneira, Anthony Thompson, Jinpan Zheng, Alex Narvaez, Joseph Mejia

Start time: 1:45 P.M. Zoom Sunday March 17th

End time: 2:35 P.M. Zoom Sunday March 17th

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story #3: Utilize computational geometry capabilities from CGAL to visualize geometric objects.
- User Story #7: Visualize or familiarize with online resources and Hernandez notes on the introductory classes/objects like points, line segments, doubly connected edge list(DCEL), and voronoi diagram.
- User Story #9: Formatting Java switch case statements to construct geometric objects in Java these objects include triangle, circle, rectangle, line, line segments, ray, and polygon.
- User Story #10: Format and break existing C++ side of project codebase, regarding toString implementation and text file save into appropriate C++ architecture with a .h file, .cpp file, and main file for appropriate submission guidelines.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- User Story #1: The library provides utilization for both Java and C++ programming languages, and must be compatible with both.
- User Story #2: Receive feedback on the simplicity of the process or software for teachers, developers, and students on ease of use.
- User Story #4: Make it compatible for the educational setting in which computer graphics, algorithm techniques, and computational geometry teachers find it useful.
- User Story #5: Make the library extensible for future geometric algorithms additions for future capstone groups.
- User Story #6: Abstract framework that is a simple method for adding graphing collections of geometric objects to java programs.
- User Story #8: Create simple program exercises that can implement the doubly connected edge list (DCEL) and voronoi diagram in a bare bones all can use aspect.
- User Story #9: Formatting of and standardization of toString for objects such as circle, triangle, rectangle, and polygon in the C++ language in which appropriate measures are taken for saving radius, vertices, RGB, and location into the text file.
- User Story #10: Format and break existing C++ side of project codebase, regarding toString implementation and text file save into appropriate C++ architecture with a .h file, .cpp file, and main file for appropriate submission guidelines.
- User Story #11: Formatting of and standardization of toString objects lines and rays that are not simple shapes, that later on create or are the building blocks for voronoi diagram and Delaunay triangulation into the text file.

- User Story #12: Create simple parser in java for the appropriate text files in which information derived from C++ toString is printed in desired format in java program for values such as red color, blue color, green color, boundary, transparency, x-coordinate, y-coordinate, point, vertices.