It sounds like you have been tasked with developing a graph library using JavaScript and integrating it as a module with the Semantic UI framework. The library, called Graphnet, will use websockets to dynamically render nodes and edges. Additionally, you will need to implement various node and edge positioning algorithms, such as force, random, and circular.

Here are some general steps you can follow to complete this project:

Research existing graph libraries and algorithms: Before you start coding, it's important to understand the current landscape of graph libraries and algorithms. Take some time to research and compare different options, and consider which algorithms are most suitable for your needs.

Plan out the architecture of the library: Once you have a good understanding of the available options, you can start planning out the architecture of your library. This includes defining the interfaces for the various components of the library, such as the websocket connection and the positioning algorithms.

Implement the websocket connection: The first major component of your library will be the websocket connection. You will need to write code that can establish and maintain a connection to the server, and handle incoming data from the server.

Implement the positioning algorithms: With the websocket connection in place, you can start implementing the various positioning algorithms. These should be designed as standalone modules that can be easily integrated into the rest of the library.

Integrate with Semantic UI: Once you have a working prototype of your library, you can start integrating it with the Semantic UI framework. This will involve creating a module that can be easily imported and used within the framework.

Test and debug: With the library fully integrated with Semantic UI, you will need to thoroughly test and debug your code. Make sure to test the library with a variety of different data sets and configurations to ensure that it is robust and reliable.

Document and optimize: Finally, you should document your code and optimize it for performance. This includes writing clear documentation that explains how to use the library, as well as optimizing the code for speed and memory usage.

Good luck with your project!