# Requirements for Wave Simulation Environment for Wave Propagation

# Introduction

This requirements document details the requirements for a wave simulation environment. Each requirements section is subdivided into "Primary Requirements" and "Secondary Requirements" subsections.

The primary requirements are requirements that must be implemented in order to be considered complete. The secondary requirements are requirements that are bonus requirements, these are requirements that we'd like to have but understand that there may not be time to complete them. Those secondary requirements are not description complete. If you'd like to try to complete some of the requirements then we can provide more detail at a later time.

The requirements contain 4 primary blocks:

1. UI Requirements
2. Visualization
3. Solver
4. Documentation

Feel free to ask us questions if you need more detail. The primary contact will be Danny deSousa ([ddesousa@lumerical.com](mailto:ddesousa@lumerical.com)) and the secondary contact will be Jens Niegemann (jniegemann@lumerical.com).

# UI Requirements

## Primary Requirements

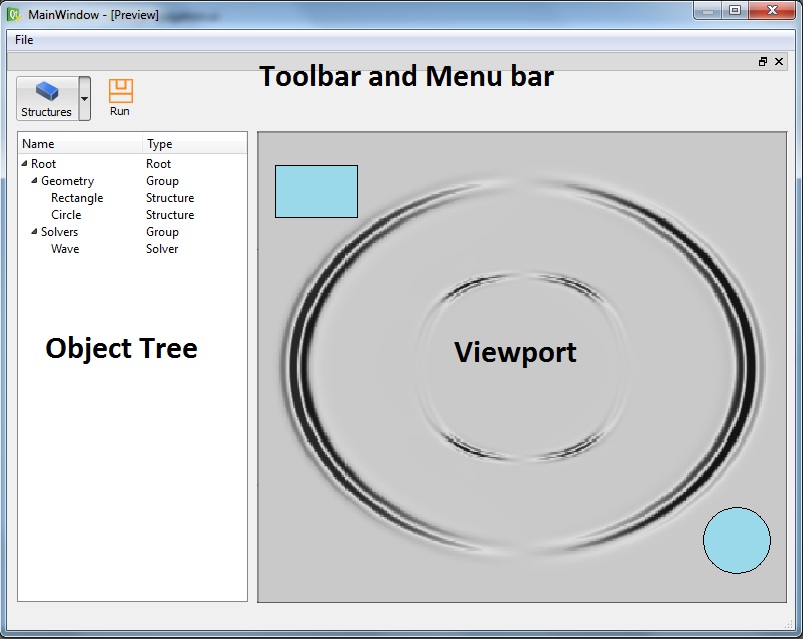
The UI is a key component in any simulation environment and it will be important in this project. A first version of the simulation environment should have 4 UI components:

1. **Menu bar** and **toolbar**, the menu bar will have:
   1. Open and Close menu items that loads/closes simulation projects. How the projects are stored and in what file-format is completely up to you, the only requirement we demand is that we can load and save simulations.
   2. The toolbar will have 2 tool buttons that allow for the insertion of structures into the simulation environment and as well an option to then run that simulation environment. The structures, from the toolbar that we'd like to be able to add are:
      1. Rectangle
      2. Circle

Default coordinates of the structures will be at the center (0,0) of the viewport.

1. The **object tree** is a tree representation of what's in the simulation environment.
   1. Items added through the toolbar will appear in the object tree.
   2. The object tree will contain a root node with no properties other than to act as a root to the object tree.
   3. The structures are to be grouped into a node called "Geometry".
   4. The wave propagation solver is to be grouped by a parent node called "Solvers".
   5. A **context menu** when right-clicking on a structure will allow for editing of structure properties. The following properties for structures are required:
      1. **X, Y coordinates** in the case of a rectangle specify the bottom left coordinates and a **X,Y span** that specify the lengths of the side.
      2. **X, Y coordinates** in the case of a circle specify the center of the circle and **radius** that specifies the length from center.
2. The **viewport** is a 2D rendering of the structures and a visualization of the wave propagating within the simulation environment.

The figure below gives a visual depiction of what we'd like to see.



We'd also like to see the UI components use some of the standard UI design patterns that are commonly used in software development.

1. An implementation of the Hierarchal Model-View-Controller (HMVC) pattern within the design environment would be very useful knowledge to us as we've been considering using that pattern within our own products.
2. A secondary design problem that we'd like you to explore is the integration of existing MVC (Qt's Model-View pattern) implementations with your HMVC implementation.

## Secondary Requirements (Bonus)

If time permits, we'd like to see some interactions supported between the object tree and the viewport or just through the viewport.

1. Change the source of the wave by selecting any spot in the viewport though left mouse-click.
2. Change the dimensions of the mesh, the mesh is a structure that allows you discretize the simulation space into quadrilaterals (square cells). This is very important for the solver and will be hard-coded initially however it would be nice to be able to edit the dimensions of the mesh.
3. Selection of a structure in the object tree that results in the structure in the viewport being highlighted.
4. Add materials to structures, materials can have an effect on how a wave behaves. (You can ask us for help here)

# Visualization Requirements

## Primary Requirements

We'd like to see, within the viewport UI, the ability to, in real-time:

1. Visualize, in 2D, a wave propagating through its environment and;
2. Demonstrating the change effects in how the wave propagates when structures are in the way.
3. Source of the wave defaults to center (0, 0) in the viewport.
4. The viewport should scale or stretch without visual anomalies This will require a coordinate system (world space) that is independent of the device screen (screen space). Reference QPainter has support for this.

## Secondary Requirements (Bonus)

If time permits, we'd like to see some interactions supported between the object tree and the viewport or just through the viewport.

1. Changing the source of the wave by left-clicking any spots in the viewport.

# Solver Requirements

The solver component will be provided as a "black box" and there is no need to modify the solver. However, some interaction between the solver and the UI needs to be implemented:

1. The solver will provide "addRectangle", "addCircle" and "resetMaterials" method. Those can be used to initialize the solver according to the object tree.
2. The solver will provide a "resetField" and a "doTimestep" method. From those, a time-stepping should be constructed and executed when the "run" button is clicked
3. The solver provides a "getField" method which provides access to the raw data. The raw data will be a floating point type (either single or double precision). The conversion to an Image needs to be implemented.

Secondary Requirements (Bonus)

1. When implementing the timestepping naively, it will lock the UI. To avoid that, the simulation should be performed in a separate thread.
2. The solver will have a "setField" which can be used to excited the field at a given cell. This can be used to make the excitation dynamic. E.g. when clicking on the visualization, there could be a wave launched from the current mouse location.

# Documentation Requirements

## Primary Requirements

We'd like to see a design document using UML that demonstrates:

1. High-level architecture using a component diagram
2. Class diagrams of the major concepts (I.e., UI and Visualization)
3. An activity diagram that describes from a high-level the workflow demonstrating the major software steps that occur from beginning of a simulation to the end of a simulation.
4. An activity diagram that describes from some high-level the primary UI interactions.