

Project Plan

Yu Duo Zhang & Yixin Liu

Vanier College

2025 - 2026

420-SF3-RE

Program Development in a Graphical Environment

Prof.: Nagat Drawel

October 17, 2025

Contents

1. Project Summary	3
1.1. Project Idea	3
1.2. Model-View-Controller (MVC)	3
1.2.1. Model (Logic & Data Structure)	3
1.2.2. View (UI/UX Design)	3
1.2.3. Controller (Event Handling)	4
2. Task Breakdown	5
2.1. Model	5
2.2. View	5
2.3. Controller	5
2.4. Other	6
3. Responsibilities and Task Allocation	7
3.1. WaveSimulator.java Class	7
3.2. WaveDisplay.java Class	7
3.3. Utils Class	7
3.4. Event Handlers	7
4. Trello Board Integration	9
4.1. Links (Trello Board & Git Repo)	9
4.2. Trello Board Screenshot:	9
5. Risk Management	10
5.1. Technical Issues	10
5.2. Scheduling Issues	10
5.3. Teamwork-Related Issues	10

1. Project Summary

1.1. Project Idea

The *Interactive Wave Physics Simulator* is an interactive application that demonstrates various types of waves. The purpose is to provide an educational tool for college students to understand better the concept of waves as it is also one of the **Physics** course for *Computer Science and Mathematic* students in Quebec.

The application will allow users to modify parameters (frequency, amplitude, wavelength, etc.) in real time and visualize waves equations and their mathematical relationships.

1.2. Model-View-Controller (MVC)

1.2.1. Model (Logic & Data Structure)

The core of the application is built up with the wave equations and their parameters. Thus, the main data structure will include all parameters needed to define a wave, configured by the user through the user interface.

The application would also guide the user to understand how to use the simulator through the *Help* menu. Aside, user can also save the parameters defined in the simulator in a local file and load it back when needed. Graph exporting feature will also be implemented to allow user to save the wave graph as an image file.

1.2.2. View (UI/UX Design)

The root of user interface will be implemented by using BorderPane Layout from JavaFx. This root will be divided into different sections:

- Top: Menu Bar (File, Exit, Help, etc.)
 - Each of the control item will be a drop-down menu that contains different options for user to choose from.
- Left: Parameter Panel (Frequency, Amplitude, Wavelength, Wave Type, etc.)
 - Each Parameter will be represented by a Label and a corresponding TextField and ComboBox to input the value and unit.
- Center: Visualization (Wave Graph)
 - This section will be implemented using ChartFX library to display the corresponding wave graph based on the user input.
- Right: Control Panel (Additional Controls)
 - Each control will first have a CheckBox to enable or disable the feature, followed by corresponding controls to adjust the settings.
- Bottom: Equation Display
 - The bottom section of the root layout will display the mathematical equation corresponding to current parameters defined by the user and to the graph. This will be shown by using a Label that updates dynamically as user changes the parameters.

1.2.3. Controller (Event Handling)

All controls in the user interface will have event handlers or listeners attached. When user interacts with any control, the corresponding controller method will be involved to process the change, update the graph and equation.

2. Task Breakdown

2.1. Model

- Wave Equations
 - General Equation: $y(x, t) = A \sin(kx - wt + \delta)$
- Core Logic
 - Default Constructor
 - Variable Setters and Getters
 - Wave Calculation Methods
- Data Structures
 - All parameters will be saved as data members in the `WaveSimulator.java` class.
 - Parameters:

VARIABLE	MEANING
A	Amplitude
k	Angular Wave Number
w	Angular Frequency
δ	Phase Difference

2.2. View

- UI Layout (`WaveDisplay.java`)
 - Left Control Panel
 - Center Graph Visualization
 - Right Control Panel
 - Top Menu Bar
 - Bottom Equation Display
- Styling (`style.css`)
 - CSS Implementation

2.3. Controller

- Event Handlers
 - Menu Actions
 - Graph Update
 - Equation Display Update
 - Additional Controls
- User Input
 - Parameter Update

2.4. Other

- File I/O
 - Parameter Saving and Loading
- Graph Exporting
 - Image Exporting
- Help Menu
 - User Guide with Pop-up Windows
- Testing
 - Unit Testing for Calculation Methods
 - Integration Testing for User Interface
- Documentation
 - Code Comments

3. Responsibilities and Task Allocation

3.1. `WaveSimulator.java` Class

TASK	TEAM MEMBER	DEADLINE
Data Field		
<code>WaveSimulator</code> Constructor		
Getters and Setters		
Calculation Methods		

3.2. `WaveDisplay.java` Class

TASK	TEAM MEMBER	DEADLINE
Left Control Panel		
Right Control Panel		
Bottom Equation Display		
Top Menu Bar		
Center Graph Visualization		
CSS Styling		

3.3. `Utils` Class

TASK	TEAM MEMBER	DEADLINE
File I/O		
Graph Exporting		

3.4. Event Handlers

TASK	TEAM MEMBER	DEADLINE
Menu Actions		
Parameter Update		

TASK	TEAM MEMBER	DEADLINE
Graph Update		
Equation Display Update		
Additional Controls		
Help Menu		
Unit Testing		
Integration Testing		
Code Comments	Yu Duo Zhang & Yixin Liu	Along the Development

4. Trello Board Integration

4.1. Links (Trello Board & Git Repo)

Trello: <https://trello.com/b/QBSKjbH8/project>◦

GitHub: https://github.com/Cloumy074/Vanier_420-SF3-RE_F25_Team-Project.git◦

4.2. Trello Board Screenshot:

5. Risk Management

5.1. Technical Issues

5.2. Scheduling Issues

5.3. Teamwork-Related Issues