Project Plan

Yu Duo Zhang & Yixin Liu

Contents

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
- Help Menu
  - ‣ User Guide with Pop-up Windows
- Graph Exporting
  - ‣ Image Exporting
- 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	Yixin Liu	Week 8
Constructor	Yu Duo Zhang	Week 8
Getters and Setters	Yu Duo Zhang	Week 8
Calculation Methods	Yu Duo Zhang	Week 10

3.2. `WaveDisplay.java` Class

Task	Team Member	Deadline
Root Layout	Yixin Liu	Week 8
Left Control Panel	Yixin Liu	Week 10
Right Control Panel	Yu Duo Zhang	Week 10
Bottom Equation Display	Yixin Liu	Week 10
Top Menu Bar	Yixin Liu	Week 10
Center Graph Visualization	Yu Duo Zhang	Week 14
CSS Styling	Yu Duo Zhang	Week 14

3.3. `Utils` Class

Task	Team Member	Deadline
File I/O	Yixin Liu	Week 14
Graph Exporting	Yu Duo Zhang	Week 14

3.4. Event Handlers

Task	Team Member	Deadline
Menu Actions Except `Help`	Yu Duo Zhang	Week 10

Task	Team Member	Deadline
Help Menu	Yu Duo Zhang	Week 14
Parameter Update	Yu Duo Zhang	Week 10
Graph Update	Yu Duo Zhang	Week 14
Equation Display Update	Yixin Liu	Week 10
Additional Controls	Yu Duo Zhang	Week 14

3.5. Other Tasks

Task	Team Member	Deadline
Unit Testing	Yixin Liu	Week 14
Integration Testing	Yixin Liu	Week 14
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

If any technical issue arises to any team member, the member should inform the other member as soon as possible, so the team can find a solution together before the due date / deadline.

If the issue cannot be resolved in time / is a long term issue, the member can reallocate some of their task to the other member to not affect the project timeline.

5.2. Scheduling Issues

If any scheduling conflict arises due to exams / personal scheduling, the team member should inform the other member in advance to adjust the workload distribution accordingly, so that the project timeline is not affected.

5.3. Teamwork-Related Issues

If codes written by one member cannot be merged properly to the main branch due to conflicts, the member should inform the other member for an review and assistance to resolve the conflicts.

If any communication related issue arises, the team member should inform the other member by other ways (in class, MIO, etc.) to ensure an effective update.