Geometric Optics Simulator

By JFKT

Joshua Indig, Farhan Haque, Kevin-Yichen Li, Tristan

Project

- Objective: A simulator that provides the user an application to visualize the different concepts in geometric optics.
- Concepts: Refraction, Thin Lenses, Mirrors and Educational Quiz.
- Workflow: Separating each concept workload and following the Model-View-Controller structure to facilitate application building.
- Needs Addressed: Offers multiple interactive learning tools for students learning the geometric optic concept in physics.
- Examples of developed functionalities: Animatable multi-lens system,

Challenges and Solutions-Kevin

- Implementing Robust Control Logic
- Identifying and Resolving Bugs
- Translating Physical Behaviors into Realistic Animations
- Handling Diverse User Input Scenarios

Challenges and Solutions - Joshua

- Challenge 1: Animating multiple lens stages while preserving accurate ray paths
- Solution: Recursive animation by reusing the final image from each stage as the next object.
- Challenge 2: Maintaining consistent drawing while dragging or changing the scale of the scene.
- Solution: Implemented offsetX, offsetY and a scale factor to translate the coordinates and to scale the lens system.
- Challenge 3: Integrating my own functionality into the shared BaseView
- Solution: Modular design of BaseView

Challenges and Solutions - Farhan

- Challenge 1:Learning and applying CSS to design and customize the user interface was difficult, especially when implementing dynamic themes. Ensuring consistent styles like background colors and fonts across all scenes required significant trial and error. Balancing CSS properties without breaking functionality added to the complexity.
- Solution 1 : I overcame this by experimenting extensively with CSS, using modular stylesheets and testing each scene individually. This approach helped me isolate and fix styling issues while maintaining a consistent visual theme across the application.
- Challenge 2: Time management was a major challenge as I had to balance this project with multiple academic responsibilities, including other assignments, exams, and deadlines. It was difficult to find the proper/sufficient time for coding, testing and refining the project.
- Solution 2: I managed to solve this problem by establishing a proper personal schedule and working during moments of free time. This allowed me to stay on track with both the project and my personal work without compromising the quality of either.

Challenges and Solutions - Tristan

- Challenge 1: Finding the path coordinates required to for the ray to travel
- Solution: Created a class to handle base cases calculations using unit circle and mirror formulas to find points of contact
- Challenge 2: Handling edge cases
- Solution: Changing method of coordinate calculation using slopes and know points of contact for edge cases.
- Challenge 3: Integrating my own functionality into the shared BaseView
- Solution: Modular design of BaseView

Lessons Learned - Joshua

- Learning to write test cases to verify my rays and image position calculations.
- Recursive animations to chain multiple different stages.
- Using GitHub more effectively.
- Creating sprint reports to track progress.

Lessons Learned - Farhan

- Gained more experience with CSS styling for dynamic themes and visual customization.
- Learned how to test methods effectively with JUnit testing.
- Developed project management skills like task prioritization and time allocation.
- Learned how to effectively work in a team environment with proper communication which ensured an efficient project experience.

Lessons Learned - Tristan

- Learned how to handle user input in generative animation using physics and math equations
- Learned how to adapt to edge cases for animation generation
- Developed skills to handle sequence and simultaneous animation
- Learn how to develop modular classes to utilize for efficiency like the Coordinate class or BaseView Class.
- Improved time management skills towards reaching required checkpoints on time.

Lessons Learned - Kevin

- Enhanced javafx technique, OOP concepts and data structure learned throughout the two-year program.
- Communicate and cooperate into a larger group of development
- Better understanding of MVC framework