Pd6 Final Project 2023-06-04 Petr Ermishkin

Worked with no one.

## Light Ray Simulator

### Description

The purpose of this project is to simulate light rays in different environments. Planned functionalities include light ray reflection and refraction with rectangular objects and lenses as well as presets to test different situations with user input. Additional functionality could include objects with more complex shapes, objects at different angles, the ability to draw virtual parts of a ray, and the ability to determine the location of an object’s image given two rays.

### List of Current Functionalities

0. Create a ray by clicking and dragging to specify start and heading.

1. Create a rectangle by clicking and dragging while holding shift to specify the top left and bottom right corners.

2. Rays reflect upon encountering rectangles, changing the direction of the photon and creating a new line.

3. Rays can now be contained inside of rectangles. By default the entire screen is bounded by a single rectangle.

4. Pressing R resets the simulation.

### List of Functionalities Planned to be Done by the Next Meeting

0. Implement presets / more user input.

1. Fix back-to-front rectangle creation (currently the rectangle breaks if not created with the starting point as the top left corner). Also fix accidentally holding / letting go of the shift key while creating rays and rectangles.

2. Figure out the gosh darn angles (for both reflection and refraction).

### Problems / Concerns

0. I encountered a LOT of problems since the last meeting with trying to transition the rays to being able to be drawn / updated after being erased each tick. Earlier parts of code kept breaking and it took a while to isolate the issues.

1. Despite the above largely being a success, the lines “wiggle” a bit when moving. Likely due to them now being erased and updated every time the line switches to a new pixel (which is obviously not smooth since pixels come in integer values only).

2. I’m considering how I would code which object the rays are inside at any given time and it seems like it would be complicated. My instinct is to use the order in which they were created as precedence for which one the ray is “inside”.

3. I’m a bit disappointed in the scale of the project. So much of it ended up me trying to figure out seemingly simple but actually quite complicated things.

4. The program is starting to bloat quite a bit. I need to clean it up / prettify it.

5. There’s a really weird and specific bug where 1 out of 10 rectangles created right after a reset don’t seem to collide with rays. No idea how to consistently reproduce it.

### UML

Still not 100% on the UML diagram. It hasn’t had significant changes since the last one though.

