Basic Optical Instrument Experiment

Developer Documentation

This project contains files and folders. The structure of the contents of this folder is outlined below:

- images/
  This contains any images used by the lens_experiment.html, mirror.html, prism.html and optical_fiber.html files.

- js/
  This contains the JavaScript files used.
  - function.js - This contains all JavaScript code used by the project.

- Lens_experiment.html
- mirror.html
- prism.html
- optical_fiber.html
- style.css – Style Sheet for lens_experiment and mirror in this project
- prism.css – Style Sheet for prism in this project.
- optical_fiber.css – Style Sheet for optical_fiber in this project.

  All These files contain all HTML code used by the project.

The lens.js File

The lens.js file contains all main functionality functions used by this project. These are defined with several method and variables, explained below.

Methods

intersect() – Find the intersection point of two rays.

deleteray(), deleteray2() – To remove previously drawn ray lines when dragging the lens or object.

dragDrop() – This function works after the lens drag from toolbar & drop into the container.

drawImage() – When dragging object or lens, It is used to draw all set of ray lines and image of object for one lens, two lens experiment. This two categories works five different cases such as object beyond 2F , object between F and 2F , object at 2F , object at F , object within F.

document.getElementById('object_arrow').addEventListener('click', function() {}) – When click the object in the toolbar, the object will be created into the container.

arrow_object.on("dragmove", function() {}) –When drag the arrow object, this function will work. It limits the dragging area of object up to first lens.

triangle_object.on("dragmove", function() {}) –When drag the triangle object, this function will work. It limits the dragging area of object up to first lens.

square_object.on("dragmove", function() {}) –When drag the square object, this function will work. It limits the dragging area of object up to first lens.

image.on("dragmove", function(evt) {}) –When drag the lens, this function will work. If we drag the first lens, it limits the dragging area between object and second lens. If we drag the second lens, it limits the dragging area of object up to first lens.

image.on('dblclick', function(evt){}) – When double click the lens, clicked lens, related ray lines, related f,2f will be deleted.

Variables

object_name – Find the name of object in the container (Arrow, Triangle, Square).

principal_axis –Create principal axis line for the experiment.

arrow_object – Create Arrow object.

triangle_object – Create Triangle object.

square_object – Create Square object.

lense_id – After the dropping the lens into the container, find id of the dropped lens.

lense_1_name – Find the name of the fist lens in the container.

lense_2_name – Find the name of the second lens in the container.

twof_1 - Create left 2f of the first lens.

twof_2 - Create right 2f of the first lens.

focal_1 - Create left f of the first lens.

focal_2 - Create right f of the first lens.

twof_3 - Create left 2f of the second lens.

twof_4 - Create right 2f of the second lens.

focal_3 - Create left f of the second lens.

focal_4 - Create right f of the second lens.

object_position_x – Find the position of the object (Arrow, Triangle, Square).

lense_1_position – Find the position of the first lens.

lense_2_ position – Find the position of the second lens.

slope_1 - Find slope between principal axis and the central ray for first lens.

slope_2 - Find slope between principal axis and the refracted ray for first lens, if it is convex lens.

lense1_ray1_end_point_x – Find the end point x coordinate of central ray for first lens.

lense1_ray1_end_point_y – Find the end point y coordinate of central ray for first lens.

lense1_ray2_end_point_x – Find the end point x coordinate of refracted ray for first lens.

lense1_ray2_end_point_y – Find the end point y coordinate of refracted ray for first lens.

central_ray – Create the central ray.

incident_ray – Create the incident ray.

focal_point_lense_1 – Find the focal point of the first lens .According to the lens type. If convex lens = f, if concave lens = -f.

focal_point_lense_2 – Find the focal point of the second lens .According to the lens type. If convex lens = f, if concave lens = -f.

intersect_point_1 - Find intersected point of refracted ray and central ray of first lens for all three objects.

intersect_point_1_tri_squ - Find intersected point of refracted ray and central ray of first lens for triangle, square object's left upper corner point.

image_1 – Create the image of the object for both lens, if it is one lens Experiment.

refracted_ray_convex – Create refracted ray, if the first lens is convex lens.

intersect_point_2 - Find intersected point of refracted ray and central ray of second lens if the first lens is Convex for all three objects.

intersect_point_2_tri_squ - Find intersected point of refracted ray and central ray of second lens if the first lens is Convex for triangle, square object's left upper corner point.

image_2 – Create the image of the object, if the first lens is convex for two lens Experiment.

reflected_ray – Create reflected ray for the first lens, if it is convex.

slope_3 - Find slope between principal axis and the refracted ray for first lens, if it is concave lens.

refracted_ray_concave – Create the refracted ray, if the first lens is concave lens.

reflected_ray_concave - Create reflected ray for the first lens, if it is concave.

intersect_point_3 - Find intersected point of refracted ray and central ray of second lens if the first lens is Concave for all three objects.

intersect_point_3_tri_squ - Find intersected point of refracted ray and central ray of second lens if the first lens is Concave for triangle, square object's left upper corner point.

image_3 – Create the image of the object, if the first lens is concave for two lens Experiment.

reflected_ray_lense2_1, reflected_ray_lense2_2, reflected_ray_lense2_3, reflected_ray_lense2_4 – Create the reflected ray for the second lens.

lense2_ray1_end_point_y - Find the end point x coordinate of refracted ray for second lens.

lense_instance – Find the lens, When double click the object to remove.

lense_instance_id - Find id of the lens, when double click the object to remove.

The mirror.js File

Methods

intersect() – Find the intersection point of two rays.

deleteray(), deleteray2() – To remove previously drawn ray lines when dragging the mirror or object.

dragDrop() – This function works after the mirror drag from toolbar & drop into the container.

drawImage() – When dragging object or mirror, It is used to draw all set of ray lines and image of object for mirror experiment. This works five different cases such as object beyond 2F , object between F and 2F , object at 2F , object at F , object within F .

document.getElementById('object_arrow').addEventListener('click', function() {}) – When click the object in the toolbar, the object will be created into the container.

arrow_object.on("dragmove", function() {}) –When drag the arrow object, this function will work. It limits the dragging area of object.

image.on("dragmove", function(evt) {}) –When drag the mirror, this function will work. If we drag the mirror, it limits the dragging area between object and mirror.

image.on('dblclick', function(evt){}) – When double click the mirror, clicked mirror, related ray lines, related f,2f will be deleted.

Variables

object_name – Find the name of object in the container (Arrow).

principal_axis –Create principal axis line for the experiment.

arrow_object – Create Arrow object.

mirror_id - After the dropping the mirror into the container, find id of the dropped mirror.

mirror_1_name - Find the name of the mirror in the container.

image1posi – Find the image position in the container area.

twof_1 - Create left 2f of the Mirror.

twof_2 - Create right 2f of the first lens.

focal_1 - Create left f of the first lens.

focal_2 - Create right f of the first lens.

mirror_instance - Find the mirror, When double click the object to remove.

mirror_1_position - Find the position of the Mirror.

focal_point_mirror_1 - Find the focal point of the mirror. According to the mirror type. If convex mirror = f, if concave mirror = -f.

slope_1 - Find slope between principal axis and the central ray for mirror.

slope_2 - Find slope between principal axis and the refracted ray for mirror, if it is convex mirror.

slope_3 - Find slope between principal axis and the refracted ray for mirror, if it is concave mirror.

slope_4 - Find slope between principal axis and the ray go through the optical centre.

mirror1_ray1_end_point_x - Find the end point x coordinate of central ray for mirror.

mirror1_ray1_end_point_y - Find the end point y coordinate of central ray for mirror.

mirror1_ray2_end_point_x - Find the end point x coordinate of refracted ray for mirror.

mirror1_ray2_end_point_y - Find the end point y coordinate of refracted ray for mirror.

intersect_point_1 - Find intersected point of refracted ray and central ray of mirror for all three objects.

The prism.js File

Methods

intersect() – Find the intersection point of two rays.

deleteray(), deleteray1(), deleteray2() – To remove previously drawn ray lines when dragging the mirror or object.

dragDrop() – This function works after the mirror drag from toolbar & drop into the container.

drawImage() – When dragging object or mirror, It is used to draw all set of ray lines and image of object for mirror experiment. This works five different cases such as object beyond 2F , object between F and 2F , object at 2F , object at F , object within F .
document.getElementById('object_arrow').addEventListener('click', function() {}) – When click the object in the toolbar, the object will be created into the container.

arrow_object.on("dragmove", function() {}) –When drag the arrow object, this function will work. It limits the dragging area of object.

image.on("dragmove", function(evt) {}) –When drag the mirror, this function will work. If we drag the mirror, it limits the dragging area between object and mirror.

image.on('dblclick', function(evt){}) – When double click the mirror, clicked mirror, related ray lines, related f,2f will be deleted.

Variables

object_name – Find the name of object in the container (Arrow).

arrow_object – Create Arrow object.

prism_id - After the dropping the prism into the container, find id of the dropped prism.

prism_1_name - Find the name of the first prism in the container.

prism_2_name - Find the name of the second prism in the container.

image1posi - Find the image position in the container area.

object_position_y - Find the position of the object (Arrow).

prism_1_position - Find the position of the prism.

prism1_ray2_end_point_y - Find the end point y coordinate of refracted ray for prism.
prism1_ray2_end_point_x - Find the end point x coordinate of refracted ray for prism.
slope_1 - Find slope between principal axis and the central ray for mirror.

slope_2 - Find slope between principal axis and the refracted ray for mirror, if it is convex mirror.

incident_ray - Create the incident ray.

refracted_ray - Create reflected ray for the prism.

orthogonally_ray – Create the orthogonally ray for the first prism in the prism panel.

orthogonally_ray1 - Create the orthogonally ray for the second prism in the prism panel.

out_side_ray – Create the outside ray for the prism

refracted_ray_R – Create the reflected ray red light ray for the prism

refracted_ray_O – Create the reflected ray orange light ray for the prism

refracted_ray_Y - Create the reflected ray yellow light ray for the prism

refracted_ray_G - Create the reflected ray green light ray for the prism

refracted_ray_B - Create the reflected ray blue light ray for the prism

refracted_ray_I - Create the reflected ray indigo light ray for the prism

refracted_ray_V - Create the reflected ray violet light ray for the prism

The optical_fiber.js File

Methods

intersect() – Find the intersection point of two rays.

deleteray(), deleteray1(), deleteray2() – To remove previously drawn ray lines when dragging the mirror or object.

dragDrop() – This function works after the mirror drag from toolbar & drop into the container.

drawImage() – When dragging object or mirror, It is used to draw all set of ray lines and image of object for mirror experiment. This works five different cases such as object beyond 2F , object between F and 2F , object at 2F , object at F , object within F .
document.getElementById('object_arrow').addEventListener('click', function() {}) – When click the object in the toolbar, the object will be created into the container.

arrow_object.on("dragmove", function() {}) –When drag the arrow object, this function will work. It limits the dragging area of object.

image.on("dragmove", function(evt) {}) –When drag the mirror, this function will work. If we drag the mirror, it limits the dragging area between object and mirror.

image.on('dblclick', function(evt){}) – When double click the mirror, clicked mirror, related ray lines, related f,2f will be deleted.

Variables

object_name – Find the name of object in the container (Arrow).

arrow_object - Create Arrow object.

lense_id - After the dropping the optical fibre into the container, find id of the dropped optical fibre.

fiber_optic - Find the name of the optical fibre in the container.

lense_instance - Find the optical fibre, When double click the object to remove.

lense_1_position - Find the position of the optical fibre.

slope_1 - Find slope of the initial ray.

slope_2 - Find slope of the refracted ray for optical fibre

slope_3 - Find slope of the refracted ray for optical fibre

slope_4 - Find slope of the refracted ray for optical fibre

slope_5 - Find slope of the final refracted ray for optical fibre

lense1_ray1_end_point_x - Find the end point x coordinate of ray inside the optical fibre.

lense1_ray1_end_point_y - Find the end point y coordinate of ray inside the optical fibre.

lense1_ray2_end_point_x - Find the end point x coordinate of refracted ray for optical fibre.

lense1_ray2_end_point_y - Find the end point x coordinate of refracted ray for optical fibre.

intersect_point_1 - Find the first intersected point of refracted ray.

intersect_point_2 - Find the second intersected point of refracted ray.

intersect_point_3 - Find the third intersected point of refracted ray.