Project Design

Team: **ReflectoRay**
Team Members:

202201079	**SalahDin Ahmed Salh Rezk**	**s-salahdin.rezk@zewailcity.edu.eg**
202201293	**Ahmed Muhammad Abdullah**	**s-ahmed.abdullah@zewailcity.edu.eg**
202201517	**Salah Mahmoud Gamal**	**s-salah.gamal@zewailcity.edu.eg**

Team Contact: **s-salahdin.rezk@zewailcity.edu.eg**

Abstract

The Ray Reflection Simulation is a Python program that simulates the reflection of rays off mirrors. The simulation uses the Turtle graphics library to visualize the behavior of rays as they interact with mirrors, allowing users to explore principles of reflection and geometric optics. The project aims to provide an educational and interactive tool for understanding the principles of ray reflection. By allowing users to configure initial conditions and visualize the behavior of rays, the simulation promotes learning in the field of geometric optics.

Name	**Input**	**Return**	**Description**	**Member**
`parse_arguments`	None	`argparse.Namespace`	Parse command line arguments.	Ahmed
`load_initial_conditions`	`file_path: str`	`dict`	Load initial conditions from a JSON file.	Ahmed
`setup_screen`	None	`turtle.Screen`	Set up the turtle screen for the simulation.	Salah
`draw_mirrors`	`mirrors: list`	None	Draw mirrors on the turtle screen.	Salah
`create_ray`	`angle: int, start: tuple, color: str`	`turtle.Turtle`	Create a turtle object representing a ray.	Salah
`create_rays_from_sources`	`angles: list, sources: list`	`list`	Create rays from the sources.	Salah
`distance`	`point: tuple, line_start: tuple, line_end: tuple`	`float`	Calculate the distance between a point and a line.	Salah
`reflect_ray`	`incident_angle: float, line_start: tuple, line_end: tuple`	`float`	Calculate the reflection angle based on incident angle and mirror orientation.	SalahDin
`extend_ray`	`ray: turtle.Turtle`	None	Extend the ray to simulate reflection.	SalahDin
`setup_simulation`	`mirrors: list, sources: list, angles: list`	`turtle.Screen, list`	Set up the simulation with mirrors, sources, and angles.	SalahDin
`simulate_rays`	`rays: list, mirrors: list`	None	Simulate the reflection of rays off mirrors.	SalahDin
`run_simulation`	`screen: turtle.Screen, rays: list, mirrors: list, iterations: int, video: str, tmp: TemporaryDirectory`	None	Run the simulation with progress tracking and optional video recording.	SalahDin
`save_image`	`screen: turtle.Screen, output: str`	None	Save the screen as a PNG image.	Ahmed

`convert_eps_to_png`	`folder: str`	None	Convert EPS images in the input folder to PNG.	Ahmed
`save_video`	`folder: str, output: str, fps: int, codec: str`	None	Save images in the input folder as a video.	Ahmed
`save_output`	`screen: turtle.Screen, image: str, video: str, tmp: TemporaryDirectory`	None	Save the output as an image or video.	Ahmed
`main`	None	None	Main function to run the ray reflection simulation.	Salah

Table 1: Functions Implementation

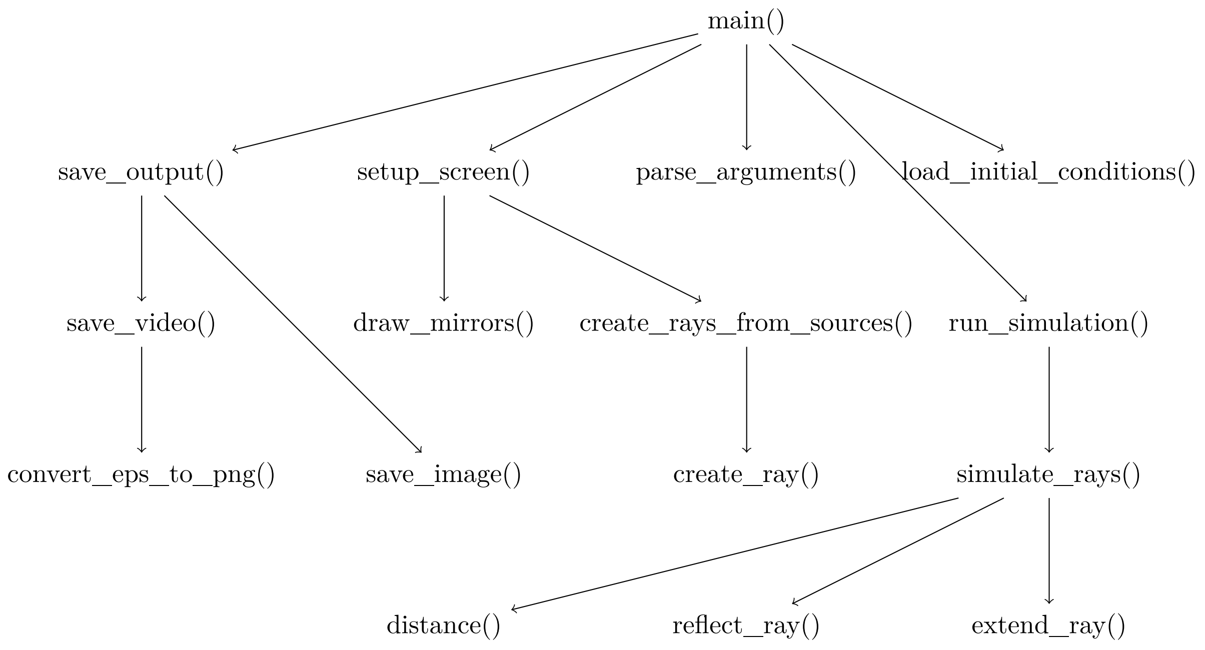

Figure 1: Flow Chart of Functions