

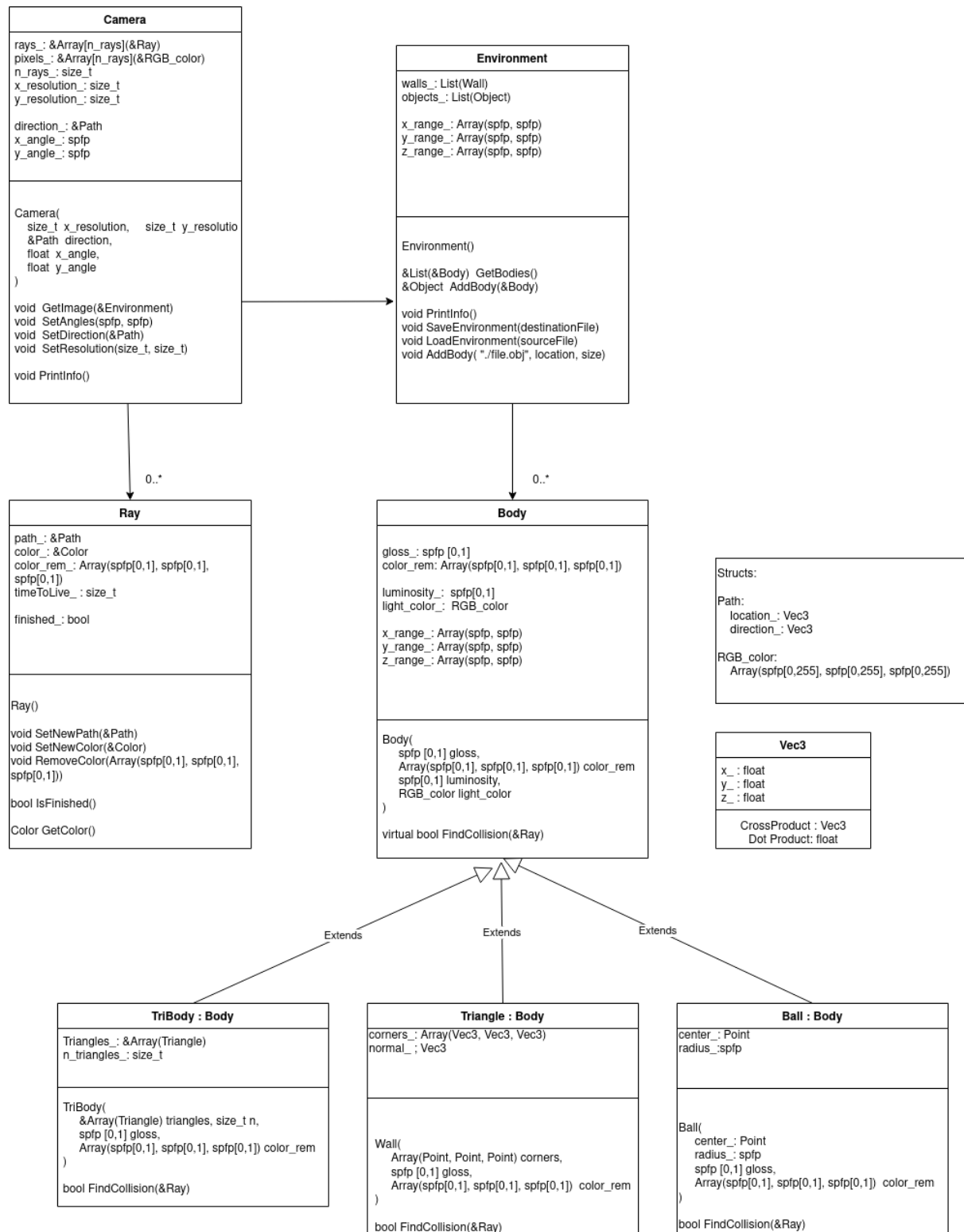
# Project plan

Path tracer, group 3

Scope of the work: what features and functionalities will be implemented, how is the program used, and how does it work

- The program is used via command line. (We might make a simple GUI if we have extra time)
- The program takes the following arguments:
  - The file from which the environment is loaded.
  - A file name for the output file.
  - Camera parameters.
  - Other?
- The program writes the rendered image to a file using ppm, png or other such format.
- Features to be implemented (in addition to the required ones):
  - Monte Carlo -integration
  - Area lights
  - Triangle meshes
  - Parallelization (using CPU or GPU) and other optimization

# High-level structure of the software: main modules, main classes (according to current understanding)



## Planned use of external libraries

- OpenMP for CPU parallelization or Cuda or OpenCL for GPU acceleration
- Possibly some library for image exporting
- We won't likely benefit from the use of math libraries apart from the standard C++ one

## Division of work and responsibilities between the group

- We try to keep the workload even.
- We use Trello board to keep track of the tasks and their status (Backlog, In progress, In Code Review, Done)
- We use feature branches to develop tasks and adding them to master after another team member has reviewed the Pull Request
- The responsibilities are divided based on the interests of the group members.
  - Each sprint a team member gets one or more tasks to do.
  - The tasks include for example familiarizing oneself with the mathematics of the algorithm and exploring libraries and file formats. To minimize total workload we try to assign these kinds of tasks to a single person who can later brief the group on the subject.
- Division of work for the first week:
  - Anton: Environment class, Reading hardcoded data to memory
  - Atso: Camera
  - Joona: Body and its subclasses
  - Kalle: Initial project template on git repository (makefiles etc.)
  - Kerttu: Ray and structs
  - **The class member functions - especially the more complex ones - do not need to be implemented during the first week!**

## Planned schedule and milestones before the final deadline of the project

- Using sprints as milestones. Each sprint lasting 2 weeks. In the beginning of a sprint each team member gets a responsibility of a feature to be completed during the sprint.
- Good communication between sprints
  - Weekly meeting within the group at 14-16 on Tuesdays.
  - Weekly meetings team members update what they have completed, what are they working on and if they have any problem
  - Weekly meeting with assistant possibly on Mondays?
- Sprints:
  - Weeks 44-45: Basic functionality
    - Working project skeleton that renders a picture using path tracing
    - Full camera implementation
    - Hardcoded environment

- 3 basic shapes:
    - ball
    - triangle
    - cube
  - 2 basic material types:
    - specular (mirror)
    - emitting (light source)
  - colors?
    - Write the rendered image to a file as ppm, png or some other format
- Weeks 46-47: More features
  - Triangle-based objects
  - different environments from file
  - ability to save environments to file
  - Monte Carlo implementation
    - diffuse material
  - Loading from file
- Weeks 48-49: Even more features, optimization. Writing documentation
  - More complicated textures/colors?
    - transparent materials
    - lacquered material
  - Different light types
    - sun
  - Optimization:
    - Parallelization with CPU/GPU
    - Optimizing collision finding algorithm by grouping objects into blocks
  - Cool scenarios for showcasing the capabilities of the program
    - Water in a glass with a straw in it?
  - Simple Graphical User Interface for choosing a file, changing parameters and outputting image?
  - Writing documentation of the program