Department of Computers

Course Number: 151055
Course Name: Mini project for Introduction to Software Engineering

Mini project 1 (MP1) instructions (10 points to the course grade)

General instructions:

- Read through the presentation for the MP1 before reading the instructions – most of it appears there in Hebrew
- Different MP1 features will be assigned by the teacher of the group while evenly distributed across the student pairs in each group! For a standard group (up to 14 students – 7 pairs), each feature will be done by at most two pairs of students. Grouping together of two or more pairs for working on the feature is not allowed! The MP1 is given for **one** week and must be presented in the next lab after a pair was assigned an MP1 feature.
- For presenting this mini-project feature, it is mandatory to create at least one picture which contains at least 10 different bodies and at least 3 different light sources in different positions for best showing of the feature functionality.
- It must be allowed to turn the feature developed in the mini project ON and OFF from unit test(s) code. In the unit tests, for each picture it is mandatory to run it twice (separate picture file names) – with and without the feature being activated, showing both resulting pictures and timing of their image rendering.
    - Modularity approach must be used
- All the features in this mini project create a cone/pyramid beam of sample rays instead (or in addition to?) the main (basic) ray (by creating and using a blackboard with set of points for target area of the ray beam)
- The number of sample rays produced for each basic ray must be 50 or higher (for X/Y rectangle-based method it will be 9X9 = 81). For final pictures production even higher number of sample rays in the beam (~300 or even ~1000) for reaching higher quality
- The basic (central) ray probably will **not** be included in the set of sample rays – depending on your algorithm, the main purpose is stated in the next point
- The rays must be distributed evenly in the rays' cone/pyramid, therefore the target points used for producing the sample rays must cover the whole target area evenly.
- The students will choose rectangular or circle area in accordance with their specific feature and they must be able to explain their decisions with appropriate reasoning.
- The students must make architectural decisions about code responsibilities (according to RDD) for their mini-project, and as a result – location (classes, methods) and way of implementation. The solution must follow the design principles and avoid "code smells" and anti-patterns. The students must be able to explain their decisions with appropriate reasoning (that may or may not be accepted by the teacher).
- The architectural decisions of the students must consider maximization of “future” code reuse as like all the super-sampling improvements are to be implemented!
    - it does not mean that each pair of students will do all the features, but the solution being able to be used to implement all the features is a part of the grading of the mini-project
- The resulting color of a point (or color multiplier for a ray) will be equally weighed color from all the sample rays.
- The amount of sample rays and all other parameters will be stored in appropriate classes according to RDD decisions and they must be changeable **from the unit tests code** by using appropriate setters in appropriate classes.
- **It is strictly forbidden to use any hard-coded values.**

Suggestions and questions for consideration in each feature

Anti-aliasing:

- The sample rays are produced from camera location through points in a pixel
- Will the target area be a pixel rectangle? An inscribed circle in the pixel? Something else?
- How will we avoid undesirable distortions of the picture?
- What will be the method of calculating evenly distributed points in the target area?
- Where will the parameters be stored?
- Where will the sample rays be produced?
- How and where will the resulting color be calculated?

Depth of Field:

- Sample rays of the beam are produced from a source area (aperture window) through a point in the focus plane (focus point)
  - The aperture window is located at the camera location, its X/Y vectors are v_{right} and v_{up}
  - Will the target area be a square or a circle?
  - It is mandatory that the distance between Focal Plane and the camera (along the v_{to} direction) or the view plane (by orthogonal line segments) will be a configuration parameter
  - It is prohibited to create a Plane object for Focal Plane
  - If the calculated focal point will be generated on a curved area instead of a plane – it will cost one grade point down (calculate it correctly basing on appropriate trigonometric formulas)
- Who is responsible for calculating the focus point and the aperture window source area?
- What are the parameters for focus plane and the aperture window source area parameters? NB: consider camera parameters like aperture, focus distance, etc.
- How will we avoid undesirable distortions of the picture (when calculating the target point)?
- What will be the method of calculating evenly distributed points in the source area?
- Where will the parameters be stored?
- Where will the sample rays be produced?
- How and where will the resulting color be calculated?

Soft Shadows:

- Will be the target area a circle or a square?
- Which light sources are affected, and which aren't (it was discussed in the lectures)?
- The light sources classes (which ones?) will get radius or square edge size parameter or something else?
- If it is a circle or a square – will it have configurable "direction" (normal to its plane) or it will always be orthogonal to the basic shadow ray? May it be different for various light source types?
- What will we get from the sample rays (color? multiplier?) and how we will average the results?
- How will we avoid undesirable distortions of the picture (when calculating the target point)?
- What will be the method of calculating evenly distributed points in the target area?
- Where will the parameters be stored?
- Where will the sample rays be produced?
- How and where will the resulting color be calculated?
- **Attention**: partial lighting and appropriate shadows **must exist** even for a light source "under horizon" (when $sign(v \cdot n) \neq sign(l \cdot n)$ for the central ray) if a part of the light source (round or square) is "over horizon". That is, this condition must be checked separately for each ray in a beam.

Glossy Surface and Diffuse Glass:

- How will you ensure that the DRY principle is kept?
- Will we use "virtual" target area or angle generator to produce a beam of rays?
- If virtual target area approach chosen:
  - Will be the target area a circle or a square?
  - How will the "virtual" target area location and size be determined to ensure the desirable diffusion?
- How will we avoid sample rays going to other side (relatively to the basic ray) of plane tangent to a surface at a point?
- How will we avoid undesirable distortions of the picture (when calculating the target point)?
- What will be the method of calculating evenly distributed points in the source area?
- Where will the parameters be stored?
- Where will the sample rays be produced?
- How and where will the resulting color be calculated?

It is highly recommended to implement Multi-Threading in this stage

(to save your time when running the tests)

See separate presentation and examples

Bonuses (the points are added to the accumulated grade of the course, up to 100):

- Jitter – 1point
- Perfect/brilliant and optimal (performance) design – 2 points
- Additional feature from the above list – 1-2 points for each one (1 point for working code and 1point for good design decisions) – up to total 6 points