Department of Computer Science

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

Mini project 2 (MP2) instructions (20 points to the course grade)

General instructions:

- **Your teacher will decide which feature each team is assigned with**
  - In a full group (7 teams) there usually won't be more that 3 teams doing the same improvement
- For presenting this mini-project feature, it is mandatory to create at least one 3D model which includes many dozens of different (all kind of) bodies and at least total of 5 light sources of different (all) types in different positions.
- It must be possible 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 with and without the feature and to show both resulting pictures and timing of their image rendering.
- It is mandatory to demonstrate time measuring when running the project for the same picture with all performance features deactivated and with all of them activated.
- The feature is built on top of mini-project 1 (MP1) feature which must be activated and effectively working
- The students must make architectural decisions about code responsibilities (according to RDD) for their mini-project, and as a result location and way of implementation. Their solution must follow the design principles and avoid "smells" as learned in the course Introduction to Software Engineering. They must be able to explain their decisions with appropriate reasoning.
- **All** feature configuration other parameters will be stored in appropriate classes according to RDD decisions and they must be changeable by appropriate setters from the unit tests code. **It is strictly forbidden to use any hard-coded values.**
- **NB:** the working code can give 40% of MP2 grade – all the rest is for the improvement level (how faster it is), architecture decisions, keeping design principles, coding style, etc.
- **Multithreading** (MT) is a must for MP2. The students who applied MT only will be granted 5 points for MP2 part of the project grade – however only if can explain what it done there and how does it work. If MP2 is done MT adds 3 points to the MP2 grade. Improvement expected to be achieved by the MT is ~2.1-2.3 times vs performance without MT.

Adaptive Super-sampling (applied [on top of / to] all the features you did for MP1) :
This feature can grant up to 17 points for MP2 (+ 3 points including MT)

- Expected improvement is 5-10 times (not including MT)
- What will be minimum and maximum super-sampling?
- How the results will be weighed?
- How reproducing and re-calculating of the same rays and their color will be avoided?
- What will be the method used for the feature in order to avoid re-calculating (think in terms of dynamic programming or recursive parameters)?

Boundary Volume:
This feature can grant up to 17 points for MP2 (+ 3 points including MT)
Stage1: **Conservative Boundary Region**, using AAB (up to 7 points)

- Expected improvement is 3-4 times (not including MT)
- What, where and how will be done (data, methods, communication) for the boundary regions of geometries (Box)
- When the box will be calculated?
- Where and when the CBR will be checked?

- Keep DRY, use RDD
- How much the performance has been improved?

Stage2: Boundary Volume Hierarchy (up to 2 points more on top of Stage1 as above)

- Expected improvement is 2-3 times (comparing to the performance with flat CBR)
- If CBR is done correctly – it can be easily applied on Geometries…
    - Actually – it is expected to be done based on Template Method / NVI design pattern
- Create manually the hierarchy in the scene.geometries
- Keep DRY, use RDD
- How much the performance has been improved (vs Stage 1 and overall)?

Stage3: Automatic building of hierarchy (up to 8 points more on top of Stage2 as above)

- Expected overall improvement is 10-15 times (not including MT)
- Where and how will be rebuilt the 3D model (scene.geometries) to create hierarchical structure?
- What algorithm to use?
- Keep DRY, use RDD
- NB: the performance will not be improved comparing to Stage2
- Need hundreds or thousands of bodies to demonstrate
- Compare performance vs flat scene.geometries structure

Regular Grid (for advanced students):

This feature can grant up to 17 points for MP2 (+ 3 points including MT)

- Expected overall improvement is 25-40 times (not including MT)
- It should be done in a separate (new) ray tracer implementation class
    - Will it inherit RayTracerBase or SimpleRayTracer?
- First, create CBR for all the basic geometries
- Create CBR or a box for the scene in the picture frame
- How the CBR will be divided to voxels?
- How to determine optimal voxelization? You can try, you can find approach to pre-calculate recommended voxelization in internet
- How will the info be stored with the voxels (bodies, etc.)?
- How will the ray be traced (3DDDA – search in internet)?
- How will repetitive intersections for the same body on the same ray trace be avoided?
- Compare performance with and without the feature

Bonus:

- Personal factor from your teacher for full attendance, well standing with submitting due dates, motivation, improvement show during the semester (for whom made mistakes and lost points in the beginning, but learned quickly and fixed it, “not stepping on a rake 2nd time”), interesting non-standard work – **up to 5 points**