Take-Home Exam Instructions, 18th to 21st of May 2020

Course: IMT2531 – Graphics Programming, Spring 2020, NTNU Gjøvik

Note:

- **Read this entire document before you start!**
- **Any questions must be posted via the course issue tracker and carry the tag 'Exam'. Answers are visible to all students.**
- **This exam, in combination with the associated oral exam, is worth 60% of your final mark. The Take-Home exam counts 35%, the oral exam 25%.**
- **Deadline: Thursday 21st of May 2020, noon (more details in the section 'Deadlines' – there is more than one!)**

General theme

You will render a landscape based on a grey scale heightmap. The produced scene should be textured either by using custom textures, or by sampling the scene along a sensible color to signal elevation ranges. The landscape should have a forest line and lake. The scene should be lit by sunlight, ideally having a day/night cycle with changing light position and color. You should allow the user to freely navigate the rendered scene using the mouse and/or keys of your choice. It should also be possible to switch between perspectives. You will further position several models into the scene that move around on their own and/or could be controlled by the user. There should be ground-, air- and water-based models that are constrained to move in their respective elements.

Evaluation criteria:

There are 3 tasks in total, each consisting of a main task and additional features. Each main task has the same weight. To pass you will have to address two of the main tasks. To obtain a good grade you need to address all three main tasks, either by completing them, or compensate for aspects you cannot solve with additional tasks. Obtaining a top grade (A) will in addition require that your solution is well composed and that your coding is of a professional standard (see *professionalism* below). The additional features can provide you with bonus points (just like in the assignments) and may compensate some if you only partly solve the main tasks. Simple additional features can add up to 5-10% bonus points while the hardest tasks can add up to 15-20% bonus points.

Task 1: (Generate scene)

- Load one of the heightmaps provided under *resources* (see below) into a scene and color or texture elevation ranges to resemble a somewhat realistic representation of altitudinal zones. You can assume that the lowest elevation level is water (black in heightmap), which should be bluish, and the highest are hill or mountain tops (white in heightmap), which can be colored yellow, brown or white depending on your preference. Intermediate heights should correspond to a "tree-line" and be greenish with some trees scattered around. You may generate trees on you own or load a model (several pine trees are provided in the resources).

Note that the height representation does not need to correspond to actual geographic metrics, but the elevation levels should be distinguishable for the observer.

- Illuminate the scene by emulating sunlight using the Phong model. You will have to associate normals with each vertex of the terrain to include diffuse and specular components. A way to compute normals at each vertex is provided under *links* below (you can choose another method if you like).
- **ALTERNATIVE:** If you are unable to load and render the terrain described above, then generate a flat terrain. To gain any score of significance with this choice you will need to be more involved with the textures, objects etc. For example, add a forest of trees as well as forest floor/grass/plants. There should be a lake in this model as well. You should also emulate sunlight as described above. To increase the level of detail on your terrain and make the lighting appear more interesting, you can consider adding a suitable normal mapping to your terrain (see *links* below for description). NOTE: with this option you will only obtain about 25% score on task 1. You will also be penalized to some extent on task 3. You may add extra points to this by including additional features though.

- Additional features:
    - Generate your own (non-flat) terrain. It should have hills, valleys and lake(s) as the one provided in the resources.
    - Add shadows to the scene.
    - Render clouds. You may implement a skybox,
    - Introduce rain or snow. If you want to be more advanced here, use particle systems.

Task 2: (Moving cameras and lights)

- Provide a free-moving camera whose orientation is controlled by the mouse and/or WASD keys.
- Allow switching between global camera ("bird view"), 3rd person and 1st person perspectives using the toggle button 't'. 3rd person perspective could follow one of the objects in task 3.
- Allow dynamic movement of sunlight to produce day/night cycle. Let the light color change throughout the cycle: white at midday, yellowish at evening/morning and red/orange at dusk/dawn. Set night light to some dim ambient dark blue color (or whatever choice you find moody).
- Emulate ripples with specular highlight on the water surface. You may use a normal mapping (see *links* below) for this purpose. Suitable normal mappings are provided in the resources.

- Additional features:
    - Allow the user to control the speed of day/night cycle using GUI.
    - Integrate zooming functionality for the camera.
    - Add "mini map" showing the position in the terrain from above for orientation in 1st person perspective.
    - Try to position different dim light sources throughout the terrain during nighttime to make the scene "moodier". Consider using the GUI to control light positions and color.
    - Add sound and try to make it change appropriately with the scene.

Task 3: (Objects)

- Load land-, air- and water-based objects to the scene (one of each). You may use the models provided in the resources (moose, deer, duck, fish, eagle, plane), or find a model of your own choice online. The models should move randomly around in their own element. The air- and water-based models can move in 2D by default: The air-based in a plane surface above the terrain somewhere. The fish and/or duck can be confined to the water surface.
- Try to avoid that objects move through the ground, trees or other objects. They may partly sink into the water if you allow them to walk into the lake though. NOTE: you will only obtain a 50% score here if you chose the alternative in Task 1.
- Allow the user to switch between the different objects to obtain "land view", "bird view" and "lake view", and take control over them using mouse and/or WASD keys.

- Additional features:
    - Make your own object loader to load relevant objects.
    - Let the air-based model maneuver in in three dimensions above the landscape.
    - Move the objects systematically, or let several models interact when the user does not control them.
    - Let the fish jump in and out of the water, or the duck dive under water occasionally
    - Animate some suitable model of your choice (look for one online)
    - Add sound corresponding to some of the objects.
    - Let the user choose between following the movement of the objects or control the model as above.

+Additional features:

Although additional features add "bonus points", priority should be given to main tasks, especially if you want a high grade! Note that additional features are open ended. I.e., they can go beyond any of the ones mentioned in the tasks. Remember that all additional features are weighted based on the challenge level they impose.

+Professionalism:

In addition to the explicit requirements, we expect adherence to professional standards as we have done in the assignments. This involves the structure, modularity and documentation of your code, as well as deployment. Your repository should further contain a readme file (!) that provides an overview of your project and describes the deployment process. This file should include instructions for the general use of the application (e.g., key bindings, debug functionality), including description of additional features.

Throughout the exam, we further expect frequent commits (roughly one commit per working hour) with meaningful commit messages. This to reduce the possibility for plagiarism. This will also serve to decrease the chance of losing parts of your work.

+Communication with Teaching team:

Comments or clarifying questions regarding the exam are exclusively posted via the issue tracker of the course Gitlab repository and tagged with the label 'Exam'. It may be worthwhile to subscribe to the label, so you get email notifications if anything is posted there. Please do not send direct emails or discord messages related to such concerns to any member of the teaching team during the exam period.

+Links:

-Terrain normals : https://stackoverflow.com/questions/13983189/opengl-how-to-calculate-normals-in-a-terrain-height-grid

- Normal mapping: https://learnopengl.com/Advanced-Lighting/Normal-Mapping

+Resources:

All necessary resources are provided as a single zip file on the exam page on the course wiki. Exam wiki page URL: https://git.gvk.idi.ntnu.no/course/imt2531/imt2531-2020/-/wikis/Exam

There are heightmaps from different locations in high and low resolution. Choose location after your preference and resolution depending on your computer's performance. You are also free to use other objects than the ones provided.

+Deliverables:

1.) **URL to code repository** posted alongside your NTNU username in the spreadsheet linked from the course wiki page. https://git.gvk.idi.ntnu.no/course/imt2531/imt2531-2020/-/wikis/Exam
2.) **Fill in the self-assessment form** linked from the wiki. It will ask you to provide a self-assessment of your work and give you the opportunity to highlight strengths and shortcomings of your work. It also allows you to give us feedback on the exam.
3.) **Video of your exam** highlighting the functionality with a maximum length of 3 minutes. It should be hosted on a video hosting platform, such as Youtube, and for the course of the exam evaluation, not be listed on the platform itself (set to unlisted). The link must be added in the corresponding column in the submission spreadsheet linked in the wiki before deadline of the video submission (see below).

Note: Ensure that the URL you posted works (and that the repo is accessible). If we can't find your work, we can't assess it! Consider making your repo and place its link into the spreadsheet ASAP. On the deadline you will then simply need to make the last commit...

Deadlines:

- The deadline for **Deliverables 1 is on Thursday, 21st of May 2020, noon**.
- For Deliverable 2 you have 12 more hours: Thursday 21st of May 23:59:59
- For Deliverable 3 the deadline is Monday 25th of May 2020 at noon