3D Graphics project
DJAN Dennis-Mintah, QAMHIYEH Bashar, SOSNIN Andrey
MoSIG M1, 2017-2018

1 Disclaimer

The main program `project.py`, which contains all of the features we've implemented, is too computationally intense to run adequately on most machines. We've spent a lot of time to find how to improve performance, but in our opinion it is still too slow to run in real time. Because of that, we decided to showcase some of the features in separate programs.

2 List of files

The archive contains 4 main executables:

- `project.py`: the main program - contains a terrain with trees and a geyser, as well as an animated sun, some keyframe-animated 3D clouds and a statically-animated non-hierarchical non-textured model of a flying dinosaur. Can be played as a land-based dinosaur with a third-person-view camera and multiple animations.
- `world.py`: same as `dino.py`, but without the playable model.
- `dino.py`: just the ground with a playable dinosaur and a skybox, without other objects.
- `geizer.py` : a single model of a geyser animated with particle effects, because we think that it looks nice.

Moreover, we generate some number of assets ourselves:

- `generate_grid.py`: generation of an uneven terrain using Perlin noise.
- `gen_texture.py`: generation of the texture for the ground (slopes are gray, flats are green)
- `generate_assets.py`: contains generators of a sphere mesh (used in clouds), geyser, as well as wrappers for the terrain generators. At each generation we check whether a textured ground has already been generated, and if so, we reuse it. We submit the archive with a the ground pregenerated because the generation takes about a minute.

3 Short description of the scene

A sun raises in the sky. A lonely dinosaur stands in a field next to a geyser and gazes at the clouds drifting across the firmament. In the distance he sees a pterosaur gliding, then dropping to the ground, only to raise back and continue its flight. The dinosaur decides that he wants to walk around.

4 List of known issues

- **Animation and controls**: controlling the dinosaur and the camera from the keyboard results in a very staccato-like twitching animation. We think that it may be fixed either by either introducing an FPS cap, or by buffering the control commands (e.g. when the user holds "W" pressed, we accumulate the value of translation while the animation is executing)
- **3rd person view camera**: switching between the trackball camera mode and the following mode introduces inconsistencies into camera's position.
- **Textures with alpha channel**: the leaves' texture on the trees is mapped incorrectly. For some reason there are some problems with all the image formats, we've tried, that support transparency.
- **Lights** are not being applied transforms to change the coordinate basis.

5 Implemented features

- **Multiple skinned animations**: we've used the animated model of the dinosaur provided in the Lab 7, which uses skinned keyframe animation. We load the entire set of files to get the animations, then take the mesh and the materials from only one of them. By default the dinosaur is executing the "Idle" animation, other animations can be triggered by the user. Triggering "Walk" and "Run" is associated with an additional translation forward (running is two times further away than walking). Only "Idle" can be interrupted.
- **Rotating camera**: the camera follows the dinosaur. It can be rotated around its head. If the camera finds itself inside the ground (for example when the dinosaur stands on the slope facing away from it and the camera is located behind its back) we move the camera closer to the dinosaur.
- **Controls**: the user can control the dinosaur's action (eating, attacking, ...), its movement speed (walking/running) and the direction it faces. The user can also rotate the camera around its head.
- **Animated clouds**: the clouds consist of several spheres (coloured, not textured), arranged in a hierarchical structure. Animating the nodes with keyframe animation in our opinion achieved a decent looking realistic movement.
- **Sun**: a small model of a sun moves across the sky. It is the source of light for models that use shaders with lighting.

- **Walking on uneven terrain**: we do not implement a full collision-detection mechanism. Instead, when we want to translate the model, before applying the translation, we get the ground elevation at the destination point and modify the translation's vertical coordinate accordingly.
- **Instancing**: we use OpenGL instancing on two occsasions: first when we place the trees (we load and create only one `VertexArray`), and second when we animate the particles of water coming out of the geyser.
- **Skybox**: we surrounded the scene with a skybox. When we render it, we set the translational component of the view matrix to zero.

6 Improvements

We thought of several improvements and additions that we did not have time to implement:

- **Runtime ground generation**: originally, we decided to generate the terrain using Perlin noise (rather than, say, a static 2D height map) in order to be able to generate new ground if we move too close to the current border. However we did not have a chance to work on that because of the time concerns and a significant complexity of this task from the programming point of view.
- **Capping the FPS** would probably allow to avoid the twitching during animations
- **Dynamic texture** of the skybox which changes depending on the position of the sun.
- Adding more computer controlled dinosaurs would be simple (with our collision model), but they would be difficult to observe with the current framerate

7 Assets

We used the following 3rd-party assets available for free:

- A model of a pterosaur
- A model of a tree with textures
- A model of the sun
- A texture for the skybox

as well as the contents of the `dino` archive provided in the Lab 7.