Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Real-time Graphics Assignment 5

Date Published: November 29th 2017, Date Due: December 6th 2017, 11:44

- The assignments have to be done in groups of 2 students.
- Hand in the solutions to the exercises via L^{2}P.
- You are only allowed to change code inside the marked strips (STUDENT CODE BEGIN/END)!
- Any questions? → L^{2}P discussion forum or rtg@cs.rwth-aachen.de!

If not done yet, obtain the (publicly accessible) exercise framework and assignments from `https://www.graphics.rwth-aachen.de:9000/Teaching/rtg-ws17-assignments/`.
Use **git pull** to fetch the newest changes of the framework (including the code for this exercise).

The **only** files that you should modify and **upload**:

- line.vsh
- Assignment05.cc
- RigidBody.cc

Description In this assignment you will create a roly-poly toy (German: "Stehaufmännchen"), a child's toy that always returns to its upright position as its center of mass is rather low.
You will also create a simple pendulum.
Both objects are realized via rigid bodies that can collide with the ground plane. The center of mass of the pendulum is additionally constrainted by a wire.

Controls You can apply an *impulse* to the objects with the left mouse button.
If you hold the Shift key pressed while clicking left, you attach a thruster to the object which adds a *force* to the respective rigid body.
W, A, S and D keys can be used for navigation while the right mouse button allows you to rotate the camera. Double-clicking the center mouse button resets the camera position.

Further Help Each subtask corresponds to a code strip with more detailed comments and hints.
You can find some screenshots in the folder `screenshots`.
We will upload a "Rigid Body Cheat Sheet" to the L^{2}P.

Exercise 1 Render Code [2 + 1 = 3 Points]

(a) Create a line mesh and draw it (`Assignment05.cc, Task 1.a`). Write the vertex shader for the line drawing (`shader/line.vsh`).

(b) Compute the model matrices of the rigid body shapes and draw them (`Assignment05::renderScene(...), Task 1.b`).

Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 2 Rigid Body Simulation [2 + 2 + 3 = 7 Points]

(a) Compute mass, center of gravity and inertia via sampling in `RigidBody::calculateMassAndInertia()`.

(b) In `RigidBody::update(...)`, apply the motion equations for linear momentum and position as well as angular momentum and position.

(c) In the last strip of `RigidBody.cc`, implement the following methods:

- `glm::vec3 RigidBody::linearVelocity() const { ... }`
- `glm::vec3 RigidBody::angularVelocity() const { ... }`
- `glm::vec3 RigidBody::velocityInPoint(glm::vec3 worldPos) const { ... }`
- `glm::vec3 RigidBody::invInertiaWorld() const { ... }`
- `glm::vec3 RigidBody::applyImpulse(glm::vec3 impulse, glm::vec3 worldPos) { ... }`
- `glm::vec3 RigidBody::addForce(glm::vec3 force, glm::vec3 worldPos) { ... }`

You may already need/want some of them for `Task 2.b`. After this task, all features of the simulation should work.