Prof. Dr.-Ing. Ralf Steinmetz
Multimedia communications Lab

Dr. Florian Mehm
Dipl. Inf. Robert Konrad

TECHNISCHE
UNIVERSITÄT
DARMSTADT

„Game Technology“
Winter Semester 2014/2015

Exercise 9

For bonus points upload your solutions until **Friday the 19th of December 2014, 11:40**

General Information

- The exercises may be solved by teams of up to three people.
- The solutions have to be uploaded to the Git repositories assigned to the individual teams.
- **The submission date (for practical and theoretical tasks) is noted on top of each exercise sheet.**
- If you have questions about the exercises write a mail to game-technology@kom.tu-darmstadt.de or use the forum at https://www.fachschaft.informatik.tu-darmstadt.de/forum/viewforum.php?f=557

1. Practical Tasks: Physics (5 Points)

In this exercise, the overall task is to build an early version of “Marbellous”, the game we are building in the exercise. The extended physics code which handles collisions between the ball and the triangle mesh are provided for the most parts.

Note: Like the last exercise, the implementation can be quirky at some points. If you are unsure if your solution is working as intended, either ask on the forum or check the video that will be released.

1.1 Triangle-Sphere-Intersection

In Collision.h, you can find the source code for the SAT intersection test for triangles and spheres. (Note that the remaining code is an optimized of the test). Provide the code for the function IsSeparatedByA, which should be true iff the axis from the vertex a to the sphere is a separating axis.

1.2 Collision Basis

Implement the function GetCollisionBasis(vec3 x) you can find in Collision.h. The first vector in the basis vector is provided to you as the collision normal. Find two other vectors that are perpendicular and which make up the collision basis. You can use the global z-axis (0, 0, 1), but make sure to handle the case in which the collision normal is parallel to this axis.

1.3 Sphere-Box-Intersection

(See also the theoretical task.) Implement your box-sphere-intersection algorithm. Use it to detect when the ball has reached the goal area. Play the provided sound when the goal area is reached. The goal is located at approximately (x, y, z) = (-46, -44, -4) and can be modelled by a unit-length box scaled by (5.5, 2.2, 2.0).

2. Theoretical Task: Physics (5 Points)

2.1 Sphere-Box-Intersection

Research a method for intersection between a box and a sphere or derive your own.

Describe the chosen intersection test and write it in pseudocode.