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

Dipl. Inf. Robert Konrad
Polona Caserman, M.Sc.

TECHNISCHE UNIVERSITÄT DARMSTADT

„Game Technology“
Winter Semester 2017/2018

Exercise 12

For bonus points upload your solutions until **Tuesday, January 30th, 13:29**

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: Sound location (5 Points)

In this exercise, we implement basic positional audio. The exercise is based on "Superball" – you control a sphere at the bottom and have to evade spheres coming at you from above (there is no collision implemented though).

You can find the relevant source code at the top of `Exercise.cpp`, in the class `DynamicSound`. The sound is provided as two 16 bit streams (stereo). In the original array, you find the unmodified version of the sound. In this array, all values with even array indices are for the left side, all values with odd array indices on the right side. Your task is to modify the sound in realtime to account for the relative position of the listener (located in the player-controlled ball) and the sound source.

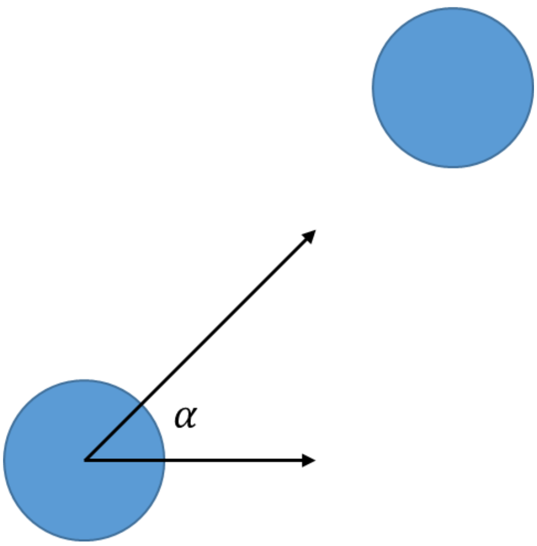
This includes two tasks:

1) Mixing between left and right ear

To implement this, you can use the cosine of the angle between the listener and the sound source. Use it to set the values of two float values which indicate the relative intensity of the sound. If the sound is very close to the listener in the horizontal direction, choose equal factors of 0.5 and 0.5.

Hint: You can normalize the direction vector and use the x-axis for this. See the following figure for an illustration of this. If the second sphere is straight ahead, the cosine will be 0, and if the second sphere is directly to the right of the sphere, it will reach its maximal value.

(Update: In the source code comments, it said to use 0.5 as the maximal value for right and left. The original intent of this comment was to choose 1.0 as the maximal value. If you already completed the exercise and used 0.5, this will be counted as correct)



2) Distance attenuation

The sound should get fainter the further away it is positioned. You can do this by dividing the mixed sound value by the exponential function of the distance.

Combining 1) and 2), the value of the sound sample should be:

$$x' = x \cdot \frac{volume}{e^{distance}}$$

https://github.com/TUDGameTechnology/Exercise12.git contains additional code to help you out. You can either copy the code changes manually or just pull them into your own repository using `git pull https://github.com/TUDGameTechnology/Exercise12.git`

Please remember to push into a branch called Exercise12.

2. Theoretical Tasks (5 Points)

2.1 Doppler Effect (2 Points)

Consider a car driving with 150 km/h and a person running away from the car with 15 km/h (the two move in the same direction). The car emits lots of different sound effects. How much does the frequency of those sound effects change for the person when the car passes him or her, i.e. how large is the maximal difference between the frequencies?

2.2 Sound location simulation without headphones (1.5 Points)

Directional sound can be simulated effectively using headphones. Can this also be done using regular speakers? What are the expected limitations?

2.3 Sound reflection data (1.5 Points)

Considering the data available to a physically based rendering engine – what data can be reused to simulate realistic sound reflections?