**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**

**Solution 12**

# 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).

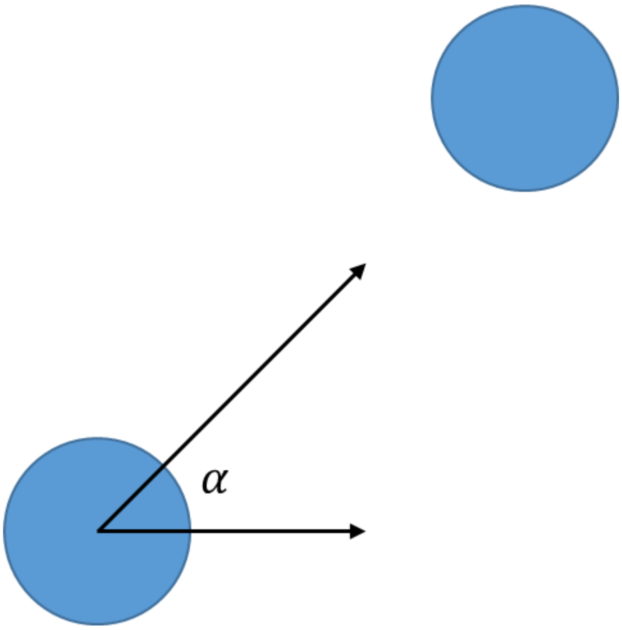
You can find the relevant source code at the top of Exercise.cpp, in the class DynamicSound. The sound is provided as a 16 bit interleaved stereo stream. In the original array, you find the unmodified version of the sound. In this array, all values with even array indices are for the left side, all values 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.




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

*You can find the solution code for the practical exercises at https://github.com/TUDGameTechnology/Solution12.git.*

## 2. Theoretical Tasks: Compression (5 Points)

### 2.1 Doppler Effect

Consider a car driving with 150 km/h and a person running away from the car with 15 km/h. 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?

*We consider the frequency the listener hears when the sound is approaching vs. when it is driving away from the listener.*

*The formula for increasing distance is:*

$$f_B = f_S \cdot \frac{c + v_B}{c - v_s}$$

*Therefore, we find that*

$$f_1 = f_S \cdot \frac{c + 15}{c - 150}$$

*Similarly for decreasing distance we find:*

$$f_2 = f_S \cdot \frac{c - 15}{c + 150}$$

*The difference between the frequencies is*

$$\begin{aligned}
|f_2 - f_1| &= \left| f_S \cdot \frac{c - 15}{c + 150} - f_S \cdot \frac{c + 15}{c - 150} \right| \\
&= \left| f_S \cdot \left( \frac{c - 15}{c + 150} - \frac{c + 15}{c - 150} \right) \right| \\
&= \left| f_S \cdot \left( \frac{c - 15}{c + 150} \cdot \frac{(c - 150)}{(c - 150)} - \frac{c + 15}{c - 150} \cdot \frac{(c + 150)}{(c + 150)} \right) \right| \\
&= \left| f_S \cdot \left( \frac{(c - 15)(c - 150) - (c + 15)(c + 150)}{(c + 150)(c - 150)} \right) \right| \\
&= \left| f_S \cdot \left( \frac{c^2 - 150c - 15c + 15 \cdot 150 - c^2 - 150c - 15c - 15 \cdot 150}{c^2 - 150^2} \right) \right| \\
&= \left| f_S \cdot \left( \frac{-300c - 30c}{c^2 - 150^2} \right) \right| \\
&\approx |f_2 \cdot -0.27| \\
&\approx \mathrm{f_s} \cdot 0.27
\end{aligned}$$

*We see that the difference is depending on the frequency that the source emits. If we insert an "a" with 440 Hz as $f_S$ and the speed of sound of 1236 km/h for c, we find that*

$$|f_2 - f_1| \approx 119\ Hz$$

*If we calculate $f_1$ and $f_2$ directly, we find*

$$f_1 \approx 387\ Hz, f_2 \approx 506\ Hz$$

*While the vehicle is approaching, the A sounds like a tone between B and C (1 semitone up), while it sounds like a tone between F# and G when the distance is increasing (3 semitones down).*

### 2.2 Sound location simulation without headphones

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

*This can only possibly work somewhat correctly for a single person, because the sound simulation would have to adjust to the position of the listener.*

## 2.3 Sound reflection data

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

*Geometry, normal maps and roughness maps – because these data sets define the structure of the environments.*

***Geometry:*** *We can calculate how the surroundings are shaped, e.g. how much space there is to the next walls.*
***Normal maps & Roughness maps:*** *We can approximate finer details of the surfaces from these maps, since they encode how the surface is shaped on a smaller level than the geometry level. For example, using this data, we can differentiate between a concrete wall and a wall covered with tapestry.*