Отчет по лабораторной работе №4

«3D звук. Библиотека FMod»

по дисциплине  
«Архитектура ЭВМ»

|  |  |
| --- | --- |
| Выполнили:  студенты гр. 3530904/80004 | Захарова А. А.  Федоров Д. В.  Селезнев В. А. |
| Преподаватель | А. В. Петров |
|  |  |

«\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г.

**Задание на выполнение работы**

1. Загрузите и запустите программу, которая проигрывает \*.mp3 файл с использованием библиотеки FMOD 4.х.

2. Познакомьтесь с 3D-моделью проигрывания звука и с набором 3D-функций из библиотеки FMOD 4. Используйте описание FMOD API, установленное на компьютере. (Version 4.44.11 Built on Mar 26, 2013) (см. рисунок)

3. Напишите программу проигрывания файла со звуковым 3D-эффектом перемещения звука по объему.

4. Напишите программу проигрывания файла со звуковым эффектом. Звуковые эффекты у студентов должны отличаться.

**Код программы**

#define \_USE\_MATH\_DEFINES

#include <cmath>

#include <conio.h>

#include <Windows.h>

#include "fmod.hpp"

#include <iostream>

void check(FMOD\_RESULT res)

{

if (res != FMOD\_OK)

{

std::cout << ":(\n";

}

}

int main()

{

FMOD::System\* system;

FMOD::Sound\* sound;

FMOD::Channel\* channel = 0;

FMOD\_RESULT result;

FMOD::System\_Create(&system);

result = system->init(16, FMOD\_INIT\_NORMAL, 0);

check(result);

result = system->createSound("D:\\kavinsky-nightcall.mp3", FMOD\_3D, 0, &sound);

check(result);

result = sound->set3DMinMaxDistance(10.0f, 1000.0f);

check(result);

result = system->playSound(sound, 0, true, &channel);

check(result);

FMOD\_VECTOR pos = { 0.0f, 0.0f, 0.0f };

result = channel->set3DAttributes(&pos, 0);

check(result);

FMOD::DSP\* effect;

result = system->createDSPByType(FMOD\_DSP\_TYPE\_ECHO, &effect);

//result = system->createDSPByType(FMOD\_DSP\_TYPE\_TREMOLO, &effect);

check(result);

//result = channel->addDSP(0, effect);

check(result);

channel->setPaused(false);

double angle = 0;

bool isplaying;

channel->isPlaying(&isplaying);

while (isplaying)

{

++angle;

if (angle > 360)

{

angle -= 360;

}

pos.x = 100 \* std::cos(angle / 180 \* M\_PI);

pos.z = 100 \* std::sin(angle / 180 \* M\_PI);

channel->set3DAttributes(&pos, 0);

system->update();

channel->isPlaying(&isplaying);

Sleep(10);

if (\_kbhit())

{

break;

}

}

return 0;

}