

Лабораторна робота №3

Дудник Олексій КН-19-2

Програмування засноване на тестуванні

Мета роботи: навчитися розробляти й реалізовувати модульні тести з використанням систем автоматизованого тестування серії Unit.

Хід виконання роботи

1. Вивчити теоретичні відомості.
2. Ознайомитися із принципами роботи системи автоматизованого тестування JUnit
3. Реалізувати модульні тести, що забезпечують перевірку основної функціональності кожного класу відповідно до розробленого проекту програмного продукту.
4. Домогтися успішного виконання всіх модульних тестів.
5. Зробити висновки з проекту програмного продукту.

Для написання юніт-тестів буду використовувати C++ бібліотеку GoogleTest
Напишемо тести для класу Envelope:

```

1  #include <iostream>
2  #include <gtest/gtest.h>
3
4  #include "../include/audio/Envelope.h"
5  #include "../include/audio/Oscillator.h"
6
7  // Envelope Tests
8  class EnvelopeTest : public testing::Test
9  {
10 protected:
11     void SetUp() override
12     {
13         m_env = new Envelope(&m_attack, &m_decay, &m_sustain, &m_release);
14     }
15
16     float m_attack = 0.1f;
17     float m_decay = 0.1f;
18     float m_sustain = 0.8f;
19     float m_release = 0.2f;
20     Envelope *m_env;
21 };
22

```

Рис. 1: Ініціалізуємо конфігурацію класу Envelope для декількох тестів

```

23 TEST_F(EnvelopeTest, AmplitudeOnSilent)
24 {
25     EXPECT_TRUE(m_env->isNoteOff());
26     EXPECT_EQ(0, m_env->getAmplitude(0));
27 }
28
29 TEST_F(EnvelopeTest, AmplitudeOnAttack)
30 {
31     m_env->noteOn(140.0);
32     EXPECT_FALSE(m_env->isNoteOff());
33     EXPECT_LE(0, m_env->getAmplitude(140.1));
34     EXPECT_GE(1, m_env->getAmplitude(140.1));
35 }
36
37 TEST_F(EnvelopeTest, AmplitudeOnDecay)
38 {
39     m_env->noteOn(140.0);
40     EXPECT_FALSE(m_env->isNoteOff());
41     EXPECT_LE(m_sustain, m_env->getAmplitude(140.1));
42     EXPECT_NEAR((double)m_sustain, m_env->getAmplitude(140.2), 0.0001);
43 }
44
45 TEST_F(EnvelopeTest, AmplitudeOnSustain)
46 {
47     m_env->noteOn(140.0);
48     EXPECT_FALSE(m_env->isNoteOff());
49     EXPECT_EQ(m_sustain, m_env->getAmplitude(140.4));
50 }
51
52 TEST_F(EnvelopeTest, AmplitudeOnRelease)
53 {
54     m_env->noteOn(140.0);
55     EXPECT_FALSE(m_env->isNoteOff());
56     m_env->noteOff(150.0);
57     EXPECT_TRUE(m_env->isNoteOff());
58     EXPECT_GE(m_sustain, m_env->getAmplitude(150.1));
59     EXPECT_EQ(0, m_env->getAmplitude(150.3));
60 }

```

Рис. 2: Самі тести

Повторимо ції дії для класу Oscillator

```
62 // Oscillator Test
63 class OscillatorTest : public testing::Test
64 {
65 protected:
66     void SetUp() override
67     {
68         m_osc = new Oscillator(&m_type, &m_isActive, &m_noteOffset, &m_gain);
69     }
70
71     WaveType m_type = SINE;
72     bool m_isActive = true;
73     int m_noteOffset = 0;
74     float m_gain = 1.0;
75     Oscillator *m_osc;
76 };
77
78 TEST_F(OscillatorTest, Sine)
79 {
80     EXPECT_EQ(0, m_osc->getSample(48, 0, 0.0));
81     EXPECT_GE(1, m_osc->getSample(48, 1, 0.3));
82 }
83
84 TEST_F(OscillatorTest, Square)
85 {
86     m_type = SQUARE;
87     EXPECT_EQ(-0.8, m_osc->getSample(48, 0, 0.0));
88     EXPECT_EQ(0.8, m_osc->getSample(48, 0.02, 0.0));
89 }
90
91 TEST_F(OscillatorTest, Saw)
92 {
93     m_type = SAW;
94     EXPECT_EQ(0, m_osc->getSample(48, 0, 0.0));
95     EXPECT_GE(1, m_osc->getSample(48, 0, 0.3));
96 }
97
98 TEST_F(OscillatorTest, Triangle)
99 {
100     m_type = TRIANGLE;
101     EXPECT_EQ(0, m_osc->getSample(48, 0, 0.0));
102     EXPECT_GE(1, m_osc->getSample(48, 0, 0.3));
103 }
```

Рис. 3: Тести для класу Oscillator

```
105 int main(int argc, char **argv)
106 {
107     testing::InitGoogleTest(&argc, argv);
108     return RUN_ALL_TESTS();
109 }
110
```

Рис. 4: Main-функція для запуску тестів

Тепер запусимо юніт-тести

```

Aleksejs-MacBook-Pro :: build/bin/Debug <main*> % ./synth-test
[=====] Running 9 tests from 2 test suites.
[-----] Global test environment set-up.
[-----] 5 tests from EnvelopeTest
[ RUN      ] EnvelopeTest.AmplitudeOnSilent
[ OK       ] EnvelopeTest.AmplitudeOnSilent (0 ms)
[ RUN      ] EnvelopeTest.AmplitudeOnAttack
[ OK       ] EnvelopeTest.AmplitudeOnAttack (0 ms)
[ RUN      ] EnvelopeTest.AmplitudeOnDecay
[ OK       ] EnvelopeTest.AmplitudeOnDecay (0 ms)
[ RUN      ] EnvelopeTest.AmplitudeOnSustain
[ OK       ] EnvelopeTest.AmplitudeOnSustain (0 ms)
[ RUN      ] EnvelopeTest.AmplitudeOnRelease
[ OK       ] EnvelopeTest.AmplitudeOnRelease (0 ms)
[-----] 5 tests from EnvelopeTest (0 ms total)

[-----] 4 tests from OscillatorTest
[ RUN      ] OscillatorTest.Sine
[ OK       ] OscillatorTest.Sine (0 ms)
[ RUN      ] OscillatorTest.Square
[ OK       ] OscillatorTest.Square (0 ms)
[ RUN      ] OscillatorTest.Saw
[ OK       ] OscillatorTest.Saw (0 ms)
[ RUN      ] OscillatorTest.Triangle
[ OK       ] OscillatorTest.Triangle (0 ms)
[-----] 4 tests from OscillatorTest (0 ms total)

[-----] Global test environment tear-down
[=====] 9 tests from 2 test suites ran. (0 ms total)
[ PASSED  ] 9 tests.
Aleksejs-MacBook-Pro :: build/bin/Debug <main*> % █

```

Рис. 5: Результат тестування

Висновок: під час виконання роботи було здобуто навички розробки модульних тестів.