



МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»
(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова)

БГТУ.СМК-Ф-4.2-К5-01

Факультет	<u>И</u> шифр	<u>Информационные и управляющие системы</u> наименование
Кафедра	<u>ИС</u> шифр	<u>Информационные системы и программные технологии</u> наименование
Дисциплина		<u>Программирование на языке высокого уровня</u>

ПРАКТИЧЕСКАЯ РАБОТА №1

на тему

Начало работы

с библиотекой

SDL 2.0

Выполнил студент группы И-582

Махнев П.С.

Фамилия И.О.

РУКОВОДИТЕЛЬ

Спирин Д.О.

Фамилия И.О.

Подпись

Оценка _____

« _____ » _____ 2019 г.

САНКТ-ПЕТЕРБУРГ

2019 г.

1 Лабораторная работа №1

1.1 Цель работы

Изучить основные функции библиотеки SDL2.0, разобраться с проектированием простейших приложений с графическим интерфейсом

1.2 Задание

- 1) Подключить SDL 2.0 и вывести на экран любимую геометрическую фигуру, а также текст с ФИО автора и версией SDL. Версию SDL получать при помощи `SDL_GetVersion()`.

Работа выполняется на языке C++. Программа обязательно должна содержать пользовательские класс(ы). Например, класс 1 содержит инициализацию SDL и основной цикл обработки сообщений, класс 2 содержит инициализацию окна приложения, отрисовку и т.д., класс 3 содержит способ заполнения текстур для отрисовки. Функция `main` обязательно должна содержать только одну строчку пользовательского кода. Все сообщения на русском языке.

1.3 Результат выполненной работы

В результате работы была написана программа удовлетворяющая условиям задания.

Было создано 4 класса.

- 1) Класс `App`:

Реализует логику программы, содержит инициализацию SDL и основной цикл обработки сообщений.

Ниже представлен заголовочный файл класса.

Файл **app.h**

```
#pragma once

#include "../sdl_headers/sdl_headers.h"
#include "../window/main-window/main_window.h"
```

```

#include "iostream"
using namespace std;

class App
{
public:
    static App* get_instance();
    ~App();

private:
    MainWindow* main_window;

private:
    static App* instance;
    App();

private:
    bool running;
    SDL_Event e;

public:
    bool init();
    void setup();
    void update();
    void on_event();
    void quit();

    int run();
};

```

2) Класс Window:

Базовый класс для всех окон приложения, реализующий абстракцию над SDL_Window и SDL_Renderer.

Ниже представлен заголовочный файл класса.

Файл **window.h**

```

#pragma once

#include "../sdl_headers/sdl_headers.h"

#include "iostream"
using namespace std;

class Window
{
protected:
    SDL_Rect size;
    string title;

    Uint32 flags;

    SDL_Window* window;

```

```

        SDL_Renderer* renderer;

        SDL_Point mouse;

        bool display;

    public:
        Window(string title, SDL_Rect size, Uint32 flags);
        ~Window();

    public:
        bool init();
        virtual void setup() = 0;
        virtual void update() = 0;

        void show();
        void hide();
        bool is_show();

        void close();

        SDL_Rect get_size();
        SDL_Renderer* get_renderer();
        SDL_Window* get_window();
};

```

3) Класс MainWindow, производный от Window:

Класс реализующий главное окно программы. Класс реализует в себе добавление элементов и их отображение.

Ниже представлен заголовочный файл класса.

Файл **main_window.h**

```

#pragma once

#include "../window.h"
#include "../../label/label.h"
#include "vector"
#include "string"

class MainWindow : public Window
{
private:
    vector <WLabel*> labels;

public:
    MainWindow(string title, SDL_Rect sizes, Uint32 flags);

public:
    void setup();
    void update();
};

```

4) Класс WLabel:

Класс представляющий из себя виджет Надписи.

Ниже представлен заголовочный файл класса.

Файл **label.h**

```
#pragma once

#include "../sdl_headers/sdl_headers.h"

#include "iostream"
using namespace std;

class WLabel
{
private:
    SDL_Rect size;

    SDL_Texture* texture;

    SDL_Renderer* renderer;
    SDL_Texture* parent_target;

    SDL_Texture* text_texture;
    SDL_Rect text_rect;
    string text;
    TTF_Font* font;

public:
    WLabel(SDL_Renderer* renderer, SDL_Texture* parent_target, string
text, SDL_Rect size, string path, int font_size);

private:
    void init();
    void render_text(string text, SDL_Rect place);

public:
    void render();

};
```

Программа была протестирована, далее представлен снимок экрана с работающей программой.

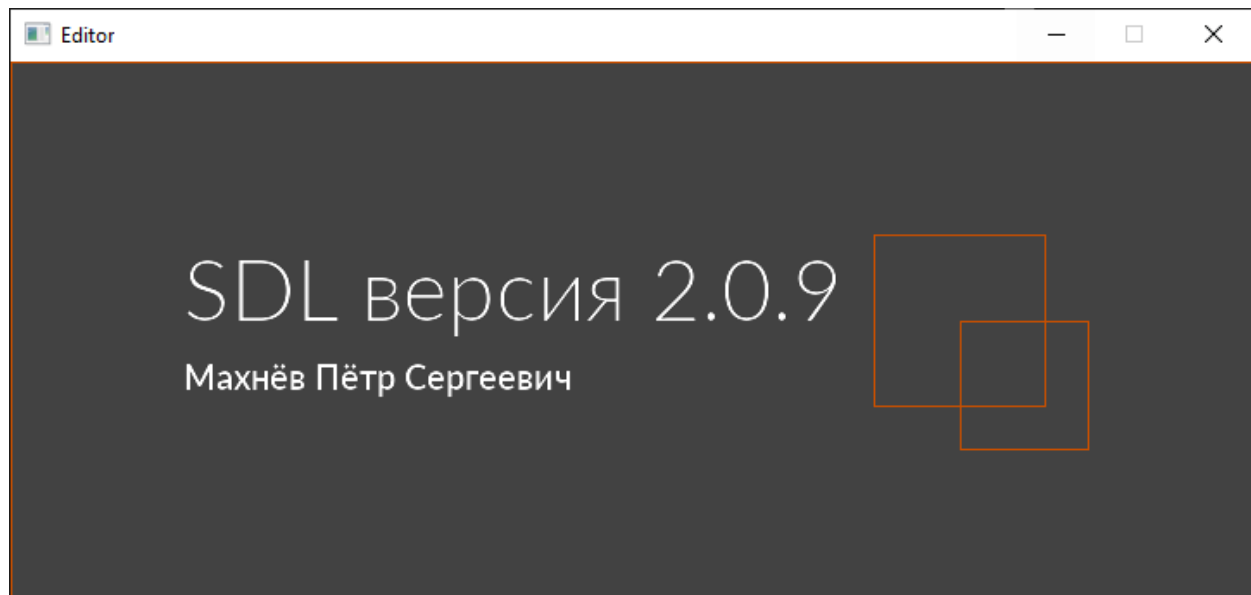


Рисунок 1 — Демонстрация работы программы