МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования «Балтийский федеральный университет имени Иммануила Канта»

ОНК «Институт высоких технологий»

ОТЧЁТ О ПРОХОЖДЕНИИ

УЧЕБНОЙ ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКИ

на базе Высшей школы компьютерных наук и искусственного интеллекта

Выполнил Кравчук Светозар Кириллович \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студент очной формы обучения 1 курса

направления подготовки 01.03.02 Прикладная математика и информатика

профиль обучения «Анализ данных и машинное обучение»

Руководитель практики

Ст. преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тарачков М.В.

г. Калининград 2024 г.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc171711451)

[ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 4](#_Toc171711452)

[SFML 4](#_Toc171711453)

[Основные библиотеки SFML 4](#_Toc171711454)

[Основные возможности библиотек SFML 5](#_Toc171711455)

[ГЛАВА 2. ЗАДАНИЕ НА ПРАКТИКУ 7](#_Toc171711456)

[Задача №1 «Выбор языка программирования» 7](#_Toc171711457)

[Задача №2 «Графика» 7](#_Toc171711458)

[Задача №3 «Логика» 7](#_Toc171711459)

[Задача №4 «Проверка на ошибки» 7](#_Toc171711460)

[Задача №5 «Сборка проекта» 7](#_Toc171711461)

[ГЛАВА 3. ВЫПОЛНЕНИЕ ЗАДАНИЙ НА ПРАКТИКУ 8](#_Toc171711462)

[Решение задачи №1 8](#_Toc171711463)

[Решение задачи №2 9](#_Toc171711464)

[Решение задачи №3 9](#_Toc171711465)

[Решение задачи №4 10](#_Toc171711466)

[Решение задачи №5 10](#_Toc171711467)

[ЗАКЛЮЧЕНИЕ 11](#_Toc171711468)

[СПИСОК ЛИТЕРАТУРЫ 12](#_Toc171711469)

[ПРИЛОЖЕНИЯ 13](#_Toc171711470)

[ПРИЛОЖЕНИЕ 1 13](#_Toc171711471)

[ПРИЛОЖЕНИЕ 2 15](#_Toc171711472)

**ВВЕДЕНИЕ**

Вид практики – Учебная технологическая (проектно-технологическая) практика (далее Учебная практика).

Цель учебной практики: получение первичных профессиональных умений навыков.

Задачи учебной практики:

1. Закрепление и углубление теоретических знаний в области информационных технологий;
2. Приобретение и развитие первичных профессиональных навыков и умений в области прикладной математики и информатики.

# ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

## SFML

SFML (Simple and Fast Multimedia Library) — это кроссплатформенная библиотека для разработки игр и мультимедийных приложений на C++. Она предоставляет простой и удобный в использовании интерфейс для работы с графикой, звуком, вводом данных, сетью и другими мультимедийными возможностями.

## Основные библиотеки SFML

* **sf::RenderWindow:**
  + Основной объект для отрисовки и управления окном.
* **sf::Texture:**
  + Загрузка и хранение текстур (изображений).
* **sf::Sprite:**
  + Представление спрайта, используемого для отображения текстуры.
* **sf::Font:**
  + Загрузка и отображение шрифтов.
* **sf::Text:**
  + Представление текста, используемого для отрисовки текста.
* **sf::Sound:**
  + Воспроизведение звуков.
* **sf::Event:**
  + Представление события (например, нажатия клавиши).

## Основные возможности библиотек SFML

1. Графика:

* Отрисовка:
  + Рисование геометрических фигур (прямоугольники, круги, линии).
  + Загрузка и отображение изображений (текстуры).
  + Работа со спрайтами.
  + Создание и управление анимациями.
  + Поддержка различных форматов изображений (PNG, JPG, GIF, BMP).
* Окна и рендеринг:
  + Создание окон с различными стилями.
  + Управление циклом событий (ввод, закрытие окна).
  + Поддержка полноэкранного режима.
  + Управление частотой кадров.

2. Звук:

* Воспроизведение звуков:
  + Загрузка звуковых файлов (wav, ogg, mp3).
  + Воспроизведение звуков и музыки.
  + Управление громкостью и положением звука.
  + Поддержка многоканального звука.
* Запись звука:
  + Запись звука с микрофона.
  + Сохранение записанного звука в файл.

3. Ввод данных:

* Мышь:
  + Определение положения курсора.
  + Обработка кликов мыши.
  + Обнаружение движения мыши.
* Клавиатура:
  + Проверка состояния клавиш.
  + Обработка нажатия клавиш.
* Джойстик:
  + Поддержка джойстиков.
  + Чтение данных о положении джойстика и кнопок.

4. Сеть:

* TCP-соединения:
  + Установка и отключение TCP-соединений.
  + Отправка и получение данных по TCP.
* UDP-соединения:
  + Отправка и получение данных по UDP.
* Обнаружение сети:
  + Поиск устройств в сети.

5. Таймеры, случайные числа и файловая система:

* Таймеры:
  + Установка таймеров для запуска событий.
* Случайные числа:
  + Генерация случайных чисел.
* Файловая система:
  + Доступ к файлам и папкам.

**ГЛАВА 2. ЗАДАНИЕ НА ПРАКТИКУ**

## Задача №1 «Выбор языка программирования»

Важно выбрать язык программирования, который больше всего подойдёт для задачи, которую мы перед собой поставили. Необходимо учесть плюсы и минусы всех языков программирования, чтобы задачу можно было выполнить, избегая серьёзных ошибок.

## Задача №2 «Графика»

Необходимо сделать красивый внешний вид для игры. Нужно: создать карты, фишки, придумать дизайн для игры, подобрать музыку, сделать приятный и удобный интерфейс.

## Задача №3 «Логика»

Необходимо продумать алгоритм для игры: создать классы для карт, фишек и создать бота для раздачи карт.

## Задача №4 «Проверка на ошибки»

Нужно попросить людей опробовать игру, чтобы они проверили её на работоспособность и оценили качество выполненной работы.

## Задача №5 «Сборка проекта»

Чтобы игра запускалась не через компилятор, необходимо осуществить сборку готового проекта (нужно объединить всю работу, которую мы делили между собой), чтобы можно было запускать игру через ярлык.

# ГЛАВА 3. ВЫПОЛНЕНИЕ ЗАДАНИЙ НА ПРАКТИКУ

## Решение задачи №1

Выбор пал на язык программирования C++. C++ является мощным языком программирования, который обладает рядом преимуществ при создании игр:

**1. Высокая производительность:**

* Низкоуровневый доступ: C++ позволяет работать с памятью и ресурсами напрямую, что обеспечивает высокую производительность, особенно важную для игр, требующих максимальной скорости.
* Оптимизация: C++ компилируется в машинный код, что позволяет оптимизировать код и снизить накладные расходы.
* Управление памятью: C++ дает разработчику полный контроль над управлением памятью, позволяя оптимизировать ее использование для повышения производительности.

**2. Гибкость и расширяемость:**

* Объектно-ориентированное программирование: C++ поддерживает объектно-ориентированное программирование, что позволяет создавать модульные и расширяемые структуры кода.
* Шаблоны: C++ позволяет использовать шаблоны, которые делают код более гибким и универсальным.
* Библиотеки: C++ имеет богатую экосистему библиотек, которые можно использовать для решения различных задач, таких как работа с графикой, звуком, физикой и т.д.

Также в C++ есть кроссплатформенная библиотека SFML, что существенно облегчает разработку игры.

## Решение задачи №2

Для того, чтобы решить задачу №2, пришлось воспользоваться большим количеством функций библиотеки SFML. Были использованы библиотеки sf::RenderWindow, sf::Texture, sf::Sprite, sf::Font и т.д. Все эти библиотеки помогли сделать игру красивой (был добавлен задний фон, музыка, красивый шрифт, интерактивное меню, рабочие настройки звука и смены фона). Возникли и некоторые сложности, пришлось изучать дополнительную литературу для того, чтобы, к примеру, сделать рабочий ползунок громкости.

Не обошлось и без использования возможностей CMake, так, благодаря ему, программа понимает из каких папок необходимо взять картинки заднего фона, карточек, фишек, музыки и т.д.

Также пришлось использовать и PhotoShop, чтобы сделать красивые текстуры карт, фишек, курсора. Он помог создать карты с нуля, сделать одну текстуру для фишек, а после остальные перекрасить и осветлить.

Пример кода программы расположен в приложении №1

## Решение задачи №3

Выполняя данную задачу, были созданы классы для кнопок, класс игрока для взятия карт, пропуска хода, а также отдельные функции для взятия первой и последней карты. Также был создан счёт игрока и денежный счёт. Если “деньги” у игрока заканчиваются, то игра выдаёт ему бонусные 2500, выводя сообщение, что вы банкрот. Всё это составляет логическую часть игры, то есть основную её часть.

Пример кода программы расположен в приложении №2

## Решение задачи №4

После того как игра готова, её необходимо проверить на критические ошибки. Это необходимо для того, чтобы игра была готова к показу публике. Для этого проект был показан тестировщикам. Они в свою очередь получили доступ к репозиторию на GitHub, скачали файлы игры, проверили элементы меню; работу кнопок, смены заднего фона, ползунка громкости; цикличность музыки, а также и работу самой игры. В ходе проверки найденные ошибки были исправлены, замечания были учтены, желания выполнены.

## Решение задачи №5

Наконец, когда проект полностью прошёл проверку, его необходимо собрать. Для этого была настроена библиотека Visual C ++ «runtime library». Это необходимо для того, чтобы создать .exe файл проекта, чтобы игра запускалась не через компилятор, а через ярлык, для удобства. Проект готов.

# ЗАКЛЮЧЕНИЕ

В ходе практики были изучены основы языка программирования С++. Задачи были направлены на закрепление полученных знаний по языку программирования C++ и кроссплатформенной библиотеке SFML.

Благодаря практики я стал лучше разбираться в языке программирования, понял, как использовать CMake и GitHub. Научился работать в команде и развил навык коммуникации между людьми.

Также я научился делать простейшие игры, используя SFML. Узнал все базовые библиотеки, понял, как их использовать в той или иной ситуации. Теперь я умею связывать «бэкэнд» и «фронтэнд» игры, а также осуществлять сборку проекта в единое целое.

По мере прохождения учебно-технологической практики я выполнил пять задач, в которых использовал полученные знания и закрепил навыки работы с C++ и SFML.

В течение практики задачи были выполнены, а цели достигнуты.

# СПИСОК ЛИТЕРАТУРЫ

**Перечень учебной литературы ресурсов сети «Интернет», необходимой для проведения практики**

* 1. “Game Programming Patterns” (автор: Robert Nystrom) - Книга, посвященная шаблонам программирования для разработки игр. Издана в “O’Reilly Media”.
  2. “Programming Game AI by Example” (автор: Mat Buckland) - Книга, посвященная искусственному интеллекту в играх. Издана в “Wordware Publishing”.
  3. “Essential C++ for Games” (автор: David R. Ashley) - Книга, посвященная использованию C++ для разработки игр. Издана в “CRC Press”.
  4. “C++ Primer” (5-е издание) (авторы: Stanley B. Lippman, Josée Lajoie, Barbara E. Moo) - Классический учебник по C++, охватывающий все основные концепции языка. Книга доступна в мягкой обложке и в твердой обложке, издан в “Pearson”.

**Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Искусство программирования на языке C++/C# - Геометрические фигуры SFML, C++. – Ссылка: <https://youtu.be/taxJkOXFsUE?si=jIxvXmH8ygFr-bzU>

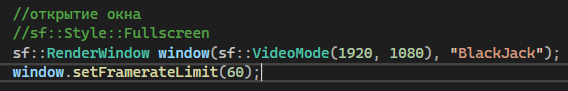
2. Павел Букреев - sfml урок 4, вывод участка изображения. – Ссылка: <https://youtu.be/RE0HDuYHx5U?si=AMIiOcBgg780ATCv>

3. SFML Graphics Library | Quick Tutorial. Ссылка: <https://www.geeksforgeeks.org/sfml-graphics-library-quick-tutorial/>

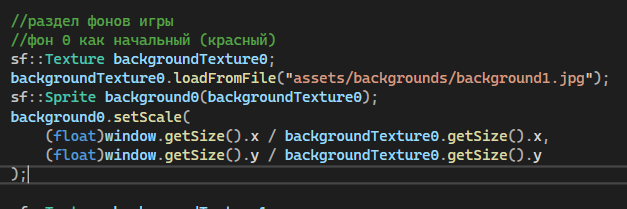
# ПРИЛОЖЕНИЯ

## ПРИЛОЖЕНИЕ 1

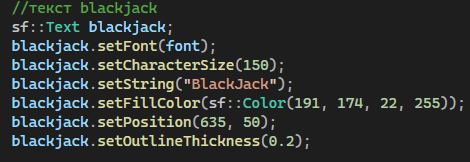
Пример использования **sf::RenderWindow**:



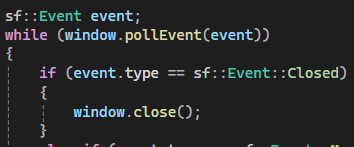
Пример использования **sf::Texture** и **sf::Sprite**:



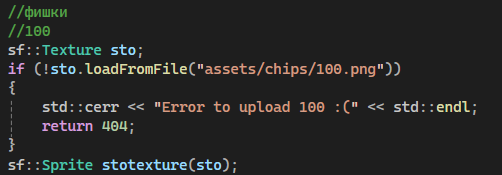
Пример использования **sf::Text**:



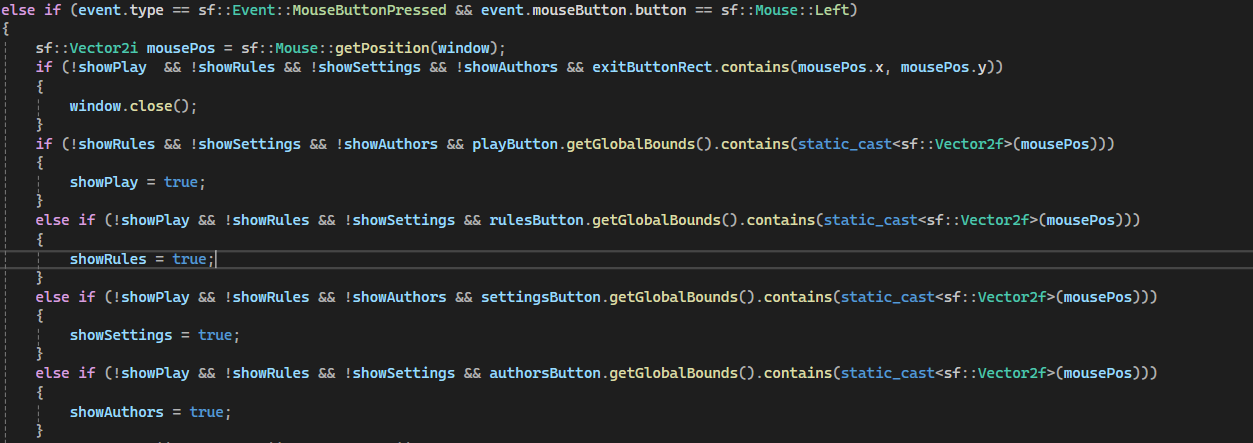
Пример использования **sf::Event**:

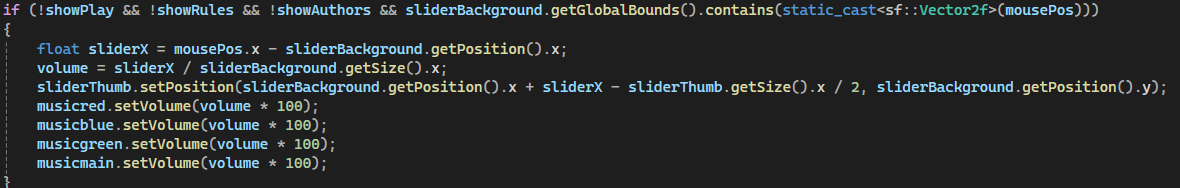


Пример реализации текстуры отдельной фишки:



Множество условий, чтобы кнопки из главного меню не работали во всех остальных вкладках:

  
Реализация ползунка звука:



## ПРИЛОЖЕНИЕ 2

Пример функций логики:

