|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

Студент Климов Илья Сергеевич \_

*фамилия, имя, отчество*

Группа ИУ7-52Б \_

Тип практики \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Название предприятия\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Климов И.С. \_

*подпись, дата фамилия, и.о.*

Руководитель практики **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2021 г.*

**Оглавление**

[**Введение** 3](#_Toc80880127)

[**1.** **Аналитическая часть** 5](#_Toc80880128)

[**1.1.** **Разбор существующих программных обеспечений** 5](#_Toc80880129)

[**1.1.1.** **Приложение «Кубик Рубика»** 5](#_Toc80880130)

[**1.1.2.** **Приложение «Free Super Cube»** 6](#_Toc80880131)

[**1.1.3.** **Приложение «Cube Genius»** 7](#_Toc80880132)

[**1.2.** **Алгоритмы удаления невидимых поверхностей.** 8](#_Toc80880133)

[**1.3.** **Методы закрашивания** 9](#_Toc80880134)

[**2.** **Конструкторская часть.** 10](#_Toc80880135)

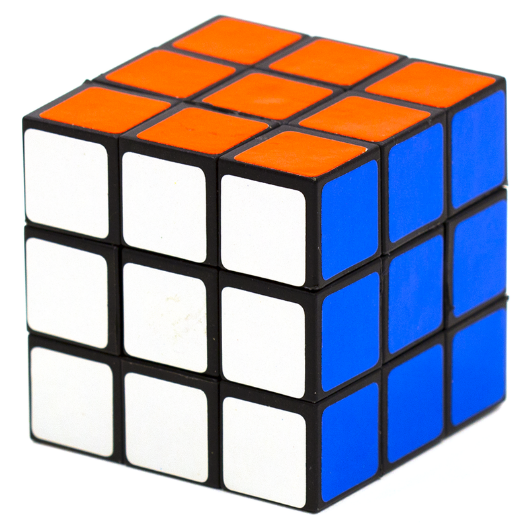
[**3.** **Технологическая часть.** 11](#_Toc80880136)

[**Заключение.** 12](#_Toc80880137)

[**Список литературы** 13](#_Toc80880138)

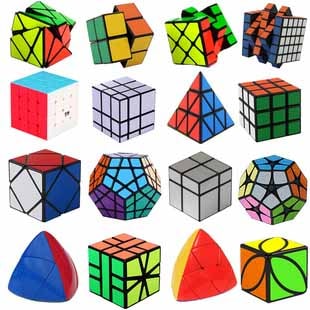
# **Введение**

В 1974 году венгерский скульптор Эрнё Рубик изобрел головоломку, которая впоследствии стала одной из самых популярных в мире, и, наверное, нет такого человека, который бы не слышал про нее. Головоломка носит название в честь её изобретателя – кубик Рубика и представляет собой пластмассовый куб с подвижными деталями.



*Рис.1 – Кубик Рубика в собранном состоянии.*

Изначально размерность составляла только 3x3x3, то есть в таком случае кубик состоит из крестовины с 6-ю центральными элементами, 8 углов и 12 реберных элементов. Затем начали появляться различные модификации, включающие в себя не только другую размерность, но и другую форму с отличающимся внутренним строением. Одними из самых популярных являются «пирамидка Мефферта» и «мегаминкс» (головоломка в форме додекаэдра). Они имеют схожий принцип с кубиком, но разное количество граней и деталей.



*Рис.2 – Разновидности головоломок.*

Задача решения кубика состоит в его сборке в изначальное положение путем поворотов граней. Сложность обуславливается тем, что число всех достижимых состояний составляет приблизительно 43 квинтиллиона (43 \* 1018). Занимательно, что из любого положения можно привести к единственному собранному всего за 20 ходов. На сегодняшний день проводится множество соревнований по различным дисциплинам. Классический кубик Рубика 3 на 3 был собран буквально за 3 секунды. [1]

Однако не все хотят тратиться на покупку головоломки, и гораздо удобней воспользоваться виртуальным аналогом. Благо, что с развитием технологий начали появляться приложения с визуализацией этой головоломки и возможностью ее сборки.

Цель работы – создание программного продукта, визуализирующего головоломки «кубик Рубика», «пирамидка Мефферта» и «мигаминкс» и позволяющего собирать их. Для достижения цели были выделены следующие задачи:

1. Определить требования для приложения основе анализа аналогичных;
2. Провести анализ алгоритмов удаления невидимых поверхностей и выбрать подходящий;

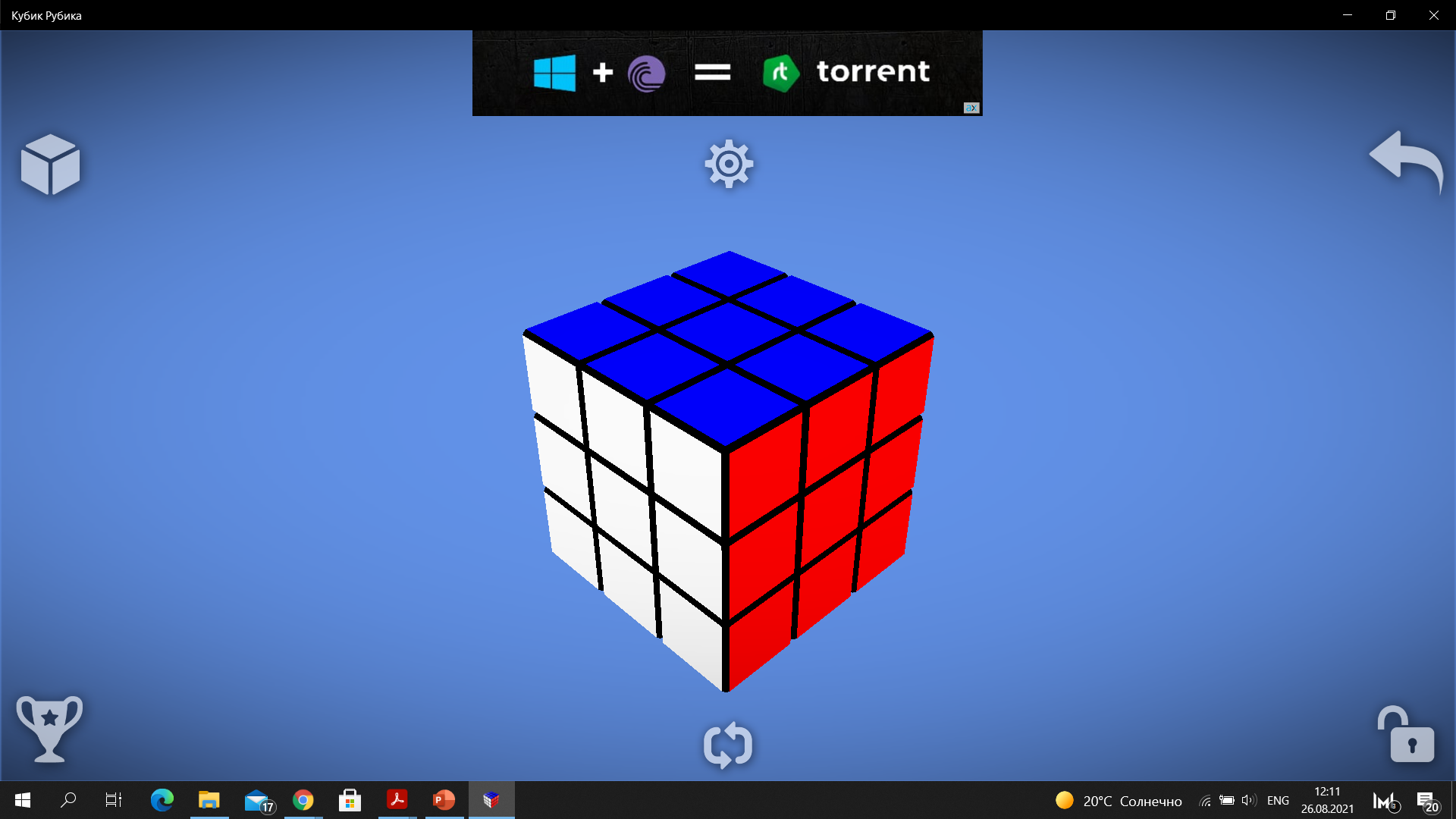
# **Аналитическая часть**

В данном разделе рассматриваются существующие программы по заданной теме, анализируются алгоритмы удаления невидимых поверхностей, закрашивания, отсечения. Исходя из анализа выбирается наиболее подходящий вариант.

## **Существующие программные обеспечения**

В качестве аналогов для проекта были взяты приложения из Microsoft Store[[1]](#footnote-1). По соответствующему запросу было найдено три продукта, каждый из которых имеет как плюсы, так и минусы.

### **Приложение «Кубик Рубика»**



*Рис.3 – Экран приложения «Кубик Рубика»*

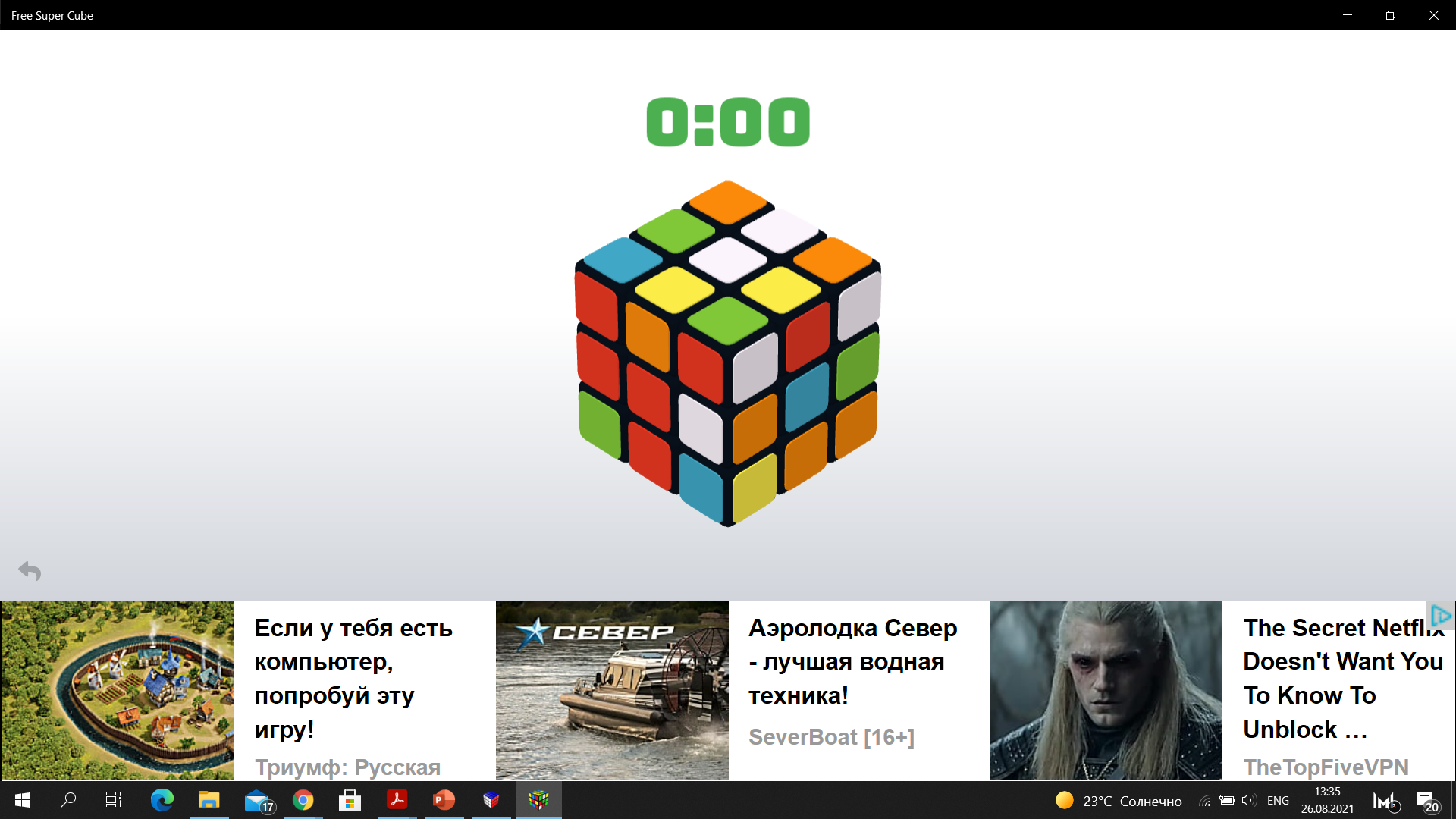
**Достоинства:**

* Возможность выбора головоломки (кубик Рубика, пирамидка Мефферта, мегаминкс) и её размерности;
* Возможность масштабирования;
* Возможность вращения головоломки в разных плоскостях;
* Возможность вернуть предыдущее состояние.

**Недостатки:**

* Грани поворачиваются быстро, без контроля;
* На гранях головоломки виден «лестничный эффект»;
* Отсутствие возможности кастомизации;
* Отсутствие кнопок для управления головоломкой;
* Наличие рекламы.

### **Приложение «Free Super Cube»**



*Рис.4 – Экран приложения «Free Super Cube»*

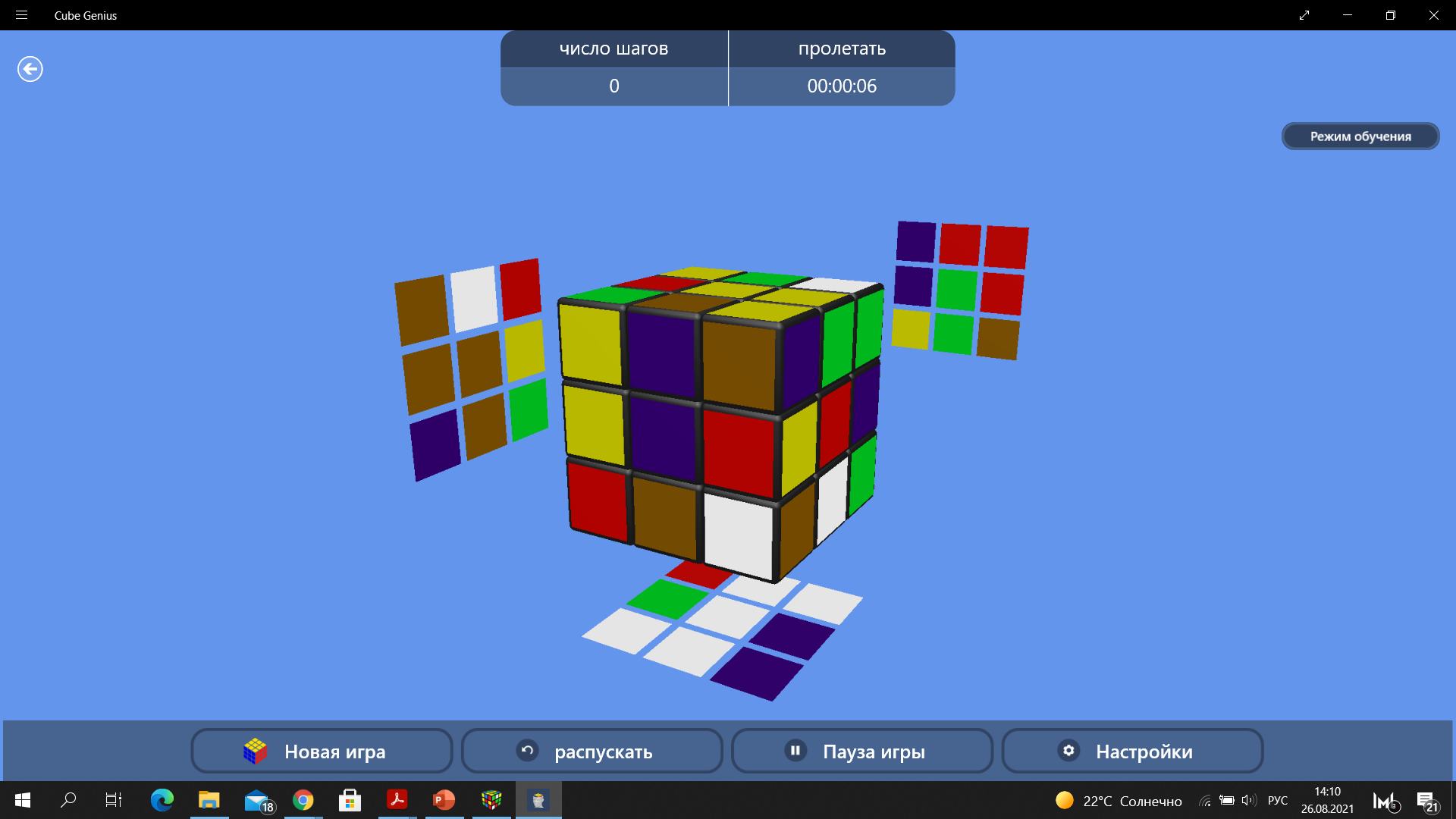
**Достоинства:**

* Плавный поворот граней мышкой, возможность контроля;
* Головоломка хорошо прорисована, гладкие линии;
* Наличие настроек головоломки;
* Возможность выбора размерности головоломки.

**Недостатки:**

* Поворот вокруг осей четко фиксирован;
* Отсутствие выбора головоломок;
* Отсутствие кнопок для управления головоломкой.

### **Приложение «Cube Genius»**



*Рис.5 – Экран приложения «Cube Genius»*

**Достоинства:**

* Видны цвета нелицевых граней;
* Отсутствие рекламы.

**Недостатки:**

* Грани поворачиваются быстро, без контроля;
* На гранях головоломки виден «лестничный эффект»;
* Отсутствие возможности кастомизации;
* Поворот вокруг осей четко фиксирован;
* Отсутствие выбора головоломок;
* Отсутствие кнопок для управления головоломкой.

**Вывод:** анализ аналогичных приложений, выявление их плюсов и минусов показали, какие требования следует определить для будущего приложения:

* Плавный поворот граней мышкой, возможность контроля;
* Головоломка хорошо прорисована, гладкие линии;
* Возможность выбора головоломки (кубик, пирамидка, мегаминкс) и её размерности;
* Возможность вращения головоломки в разных плоскостях;
* Возможность масштабирования;
* Наличие настроек головоломки;
* Возможность поворота граней с помощью кнопок и клавиш.

## **Алгоритмы удаления невидимых поверхностей.**

Задача удаления невидимых поверхностей является одной из самых важных и сложных в компьютерной графике. Выделяют три класса данных алгоритмов:

1. Работающие в объектном пространстве – работают с физической системой координат, в которой описаны объекты. Получаются весьма точные результаты, объем вычислений растет, как квадрат числа объектов;
2. Работающие в пространстве изображения – работают с системой координат экрана, на котором визуализируются объекты. Точность вычислений ограничена разрешением экрана, объем вычислений растет, как произведение количества объектов на число пикселей;
3. Формирующие списки приоритетов – работают попеременно с физической и системой координат экрана. [2]

Так как взаимодействие между пользователем и программой происходит в режиме реального времени, то одной из главных характеристик будет время, затрачиваемое на алгоритм.

## **Алгоритм Робертса**

Алгоритм Робертса является первым известным решением задачи удаления невидимых линий. Он позволяет определить, какие рёбра или часть рёбер видимы, а какие заслонены другими гранями. [3] Относится к алгоритмам, работающим в объектном пространстве.

Алгоритм Робертса проходит в два этапа:

1. Определение нелицевых для каждого объекта;
2. Определение и удаление невидимых ребер. Каждое из ребер последовательно сравнивается со всеми остальными телами. В классической версии количество вычислений растет теоретически как квадрат числа объектов. [4]

**Достоинства:**

* Относительная простота.

**Недостатки:**

* Объем вычислений растет, как квадрат числа объектов;
* Ориентирован на работу с выпуклыми многогранниками;
* Невозможность работы с прозрачными объектами.

### **Алгоритм художника**

Алгоритм художника позволяет определить, какие грани видимы, а какие заслонены, выводит каждую из них целиком по мере приближения к наблюдателю, то есть подобно тому, как художник наносит изображением на холст слой за слоем. Изначально все объекты необходимо отсортировать от заднего плана к переднему. [5] Данный алгоритм относится к алгоритмам, работающим в пространстве изображения.

**Достоинства:**

* При тривиальной сортировке обладает высоким быстродействием;
* Простота использования.

**Недостатки:**

* Сортировки по оси Z может быть нетривиальной задачей;
* При некотором расположении граней не может дать верный результат;
* Происходит отрисовка всех граней объектов, из-за чего время работы может сильно увеличиться.

### **Алгоритм Z-буфера**

Алгоритм Z-буфера позволяет определить, какие пиксели граней видны, а какие заслонены другими. Является одним из простейших алгоритмов удаления невидимых поверхностей. Сам по себе Z-буфер является двухмерным массивом, размеры которого равны размерам окна, и в каждой ячейке хранится значение глубины пикселя. [3]

## **Методы закрашивания**

# **Конструкторская часть.**

# **Технологическая часть.**

# **Заключение.**

# **Список литературы**

1. [Электронный ресурс]: Википедия
2. С сайта того
3. ПДФка
4. Алгоритм Робертса с того сайта
5. <https://studfile.net/preview/954959/page:8/>
6. http://compgraph.tpu.ru/zbuffer.htm

1. Все программные продукты, которые будут упоминаться далее, рассмотрены на операционной системе Windows 10 [↑](#footnote-ref-1)