

Департамент образования и науки города Москвы  
ГБОУ «@@@@@»

ПРОЕКТ  
«FAST MATH»

*Выполнил:*

@@@@@@@@@@@@  
@@@@@@@@@@@@  
@@@@@@@@@@@@

*Научный руководитель:*

@@@@@@@@@@@@  
@@@@@@@@@@@@

\_\_\_\_\_

# Содержание

<b>1</b>	<b>Методологический паспорт</b>	<b>2</b>
<b>2</b>	<b>Введение</b>	<b>3</b>
<b>3</b>	<b>Создание продукта</b>	<b>4</b>
3.1	Инструменты . . . . .	4
3.1.1	Manim . . . . .	4
3.1.2	EManim . . . . .	5
3.1.3	IDE . . . . .	5
3.1.4	Git . . . . .	5

# 1 Методологический паспорт

## 2 Введение

Проект Fast Math заключается в создании коротких образовательных видеороликов по математике, которые содержат основную информацию по какой-либо теме. Они позволяют понять тему на начальном уровне, что ускоряет её будущее изучение. Видео подходят для просмотра на уроке и индивидуального просмотра. Также они направлены на полное понимание тема, а не на заучивание. Поэтому в видео объясняются некоторые вещи, которые на обычном уроке могут быть поданы просто для запоминания. Например, вывод формул и их связь с другими, геометрические интерпретации. Все видеоролики распространяются на видеохостинге YouTube<sup>TM</sup> и на сайте проекта<sup>1</sup>.

---

<sup>1</sup>К сожалению, на данный момент сайт не доступен

## 3 Создание продукта

### 3.1 Инструменты

#### 3.1.1 Manim

Для создания роликов я использовал язык программирования Python и модуль Manim. Он позволяет конвертировать программный код в видеоролики. В Manim большинство инструментов заточены для визуализации математики. Создал его блогер Грант Сандерсон, также известный как 3b1b<sup>2</sup>. У Manim множество плюсов:

- Простые и понятные названия элементов и функций
- Поддержка L<sup>A</sup>T<sub>E</sub>X
- Распространение под свободной лицензией MIT<sup>3</sup>
- Хорошая документация <sup>4</sup>
- Большое кол-во уроков по библиотеке
- CLI интерфейс пакета

Manim поддерживает использование сторонних SVG-файлов и работу с ними. Для рендера видео могут использоваться две библиотеки: OpenGL и Cairo. OpenGL работает быстрее, но при этом имеет неудобную работу с окнами, плохой предпросмотр и некоторые другие проблемы, поэтому я использую Cairo. Поскольку Manim сначала создаёт небольшие кусочки видео, их нужно соединить, это возможно сделать с помощью ffmpeg. Подробнее об этом я буду говорить в пункте про видеоредакторы. Несмотря на то, что Manim эффективный и быстрый модуль, оно имеет ряд недостатков. Самый большой из них - это отсутствие поддержки кириллицы в L<sup>A</sup>T<sub>E</sub>X. Для его исправления необходимо внести изменения в код пакета. Также у Manim есть две основные версии: Manim3b1b и ManimCommunity. Manim3b1b разрабатывается только Грантом Сандерсоном, а Community версию поддерживает сообщество. Я использую Community версию, поскольку анимации в ней работают лучше.

---

<sup>2</sup>[Официальный сайт](#)

<sup>3</sup>[Текст лицензии](#)

<sup>4</sup>[Ссылка на документацию](#)

### 3.1.2 EManim

Поскольку Manim не содержит некоторых функций, которые были мне необходимы, мной был создан модуль Python EManim<sup>5</sup>. Он включает в себя возможность автоматического вывода субтитров, уведомления, начальную заставку и множество других функций. Для удобства установки и обновления модуль загружен в PyPi и на данный момент доступен для установки любым пользователем пакетного менеджера pip. EManim распространяется под лицензией GNU GPL 3.0<sup>6</sup>

### 3.1.3 IDE

Для работы с кодом необходим хотябы редактор кода, а лучше использовать IDE<sup>7</sup>. В качестве данного инструмента я использовал Neovim, улучшенный форк Vim. Поскольку Neovim изначально не обладает основными функциями IDE, нужно использовать плагины. В качестве менеджера плагинов я использовал Packer и установил следующие плагины:

- TreeSitter для подсветки синтаксиса
- Nvim-lsp для поддержки LSP серверов
- Nvim-cmp для автодополнения
- Lspkind
- LuaLine
- Bufferline

Также я установил LSP сервер pyright. Все эти настройки позволили значительно увеличить скорость разработки по сравнению с другими IDE, например с VS Code.

### 3.1.4 Git

При разработке необходимо использовать систему контроля версий. Это позволит контролировать изменения в продукте, значительно упростит работу в команде и резервное копирование. Самая эффективная, быстрая, простая и популярная из них - это Git. Git была разработана Линусом Торвальдсом для поддержки ядра Linux. Эта VCS<sup>8</sup> является децентрализованной и распространяется под свободной лицензией GNU GPL 2. Для

---

<sup>5</sup>Extendet Manim

<sup>6</sup>[Ссылка на лицензию](#)

<sup>7</sup>Integrated Development Environment (Интегрированная Среда Разработки)

<sup>8</sup>Version Control System (Система Контроля Версий)

более эффективной работы желательно использовать Git-сервер, куда будет дублирован репозиторий. Изначально я выбрал решение от компании Microsoft-Github. Github удобная и популярная система, но есть несколько причин, из-за которых этот сайт является нежелательным к использованию. Во-первых Github блокирует некоторые репозитории только по причине происхождения разработчиков, например недавно пострадали многие российские разработчики. Этот фактор не позволяет доверять Github. Также Фонд Свободного ПО не одорил этот шаг со стороны Microsoft и перестал сотрудничать с компаниями, которые используют Github. Во-вторых, код сервера Github не является открытым, поэтому никто не может развернуть его на своём сервере, дополнить или улучшить. По этим причинам я был вынужден задуматься о переходе на другой хостинг.

Существует множество открытых аналогов Github, распространяемых под свободной лицензией. Самые популярные из них: Gitlab, Gitea и Gogs. Лучшим решением было бы развернуть свой собственный сервер и разместить на нём репозиторий проект, но, к сожалению, моя рабочая станция не обладает достаточной мощностью, а аренду виртуального сервера я не могу позволить из-за отсутствия финансирования проекта. Поэтому мной был рассмотрен переход на сторонний хостинг, основанный на одном из вышеперечисленных решений. Среди многих вариантов я выбрал сайт [codeberg.org](https://codeberg.org).