

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

«Костромской государственный университет»  
(ФГБОУ ВПО «КГУ»)

Институт Автоматизированных Систем и Технологий

Кафедра информационных систем и технологий

Направление подготовки «Информационные системы и технологии»  
Дисциплина: Технологии разработки программного обеспечения

## **ОТЧЁТ К ПРОЕКТУ**

«Игра 2048»

Выполнил студент:

Сорокин Михаил Александрович

Группа: 19-ВТбо-1а

Проверил: к.т.н. доцент

Барило Илья Иванович

Оценка: \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

Кострома,

2022 г.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ.....	4
1.1. Выбор идеи .....	4
1.2. Выбор платформы.....	4
1.3. Этапы разработки.....	5
2. ФУНКЦИОНИРОВАНИЕ ПРОГРАММЫ.....	7
2.1. Правила игры.....	7
2.2. Геймплей .....	7
3. РАЗРАБОТКА ПРОЕКТА .....	11
3.1. Дизайн .....	11
3.2. Код из Scratch .....	13
4. СИСТЕМА КОНТРОЛЯ ВЕРСИЙ GITHUB.....	20
5. ТАБЕЛЬ РАБОЧЕГО ВРЕМЕНИ .....	21
ЗАКЛЮЧЕНИЕ .....	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	24

## ВВЕДЕНИЕ

В наше непростое время чем раньше начнешь готовить работника к будущей профессии, тем более высококвалифицированным специалистом он станет. Значит прививать определенные навыки человеку нужно с еще дошкольного-школьного возраста, а в этом возрасте знания лучше воспринимаются в игровой форме. А значит интересные образовательные игры могут разнообразить процесс обучения и привлечь к важной общественной теме больше человек.

И именно поэтому в ВУЗе мне была поставлена задача-разработать игру, благодаря которой обучающиеся могли постичь определенные знания, которые помогут им в будущем лучше овладеть профессиональными навыками (конкретнее-математическими и с уклоном в программирование)

Цель – разработать игру, которая поможет постичь определённые математические навыки или же наглядно показать функционирование программ и языков программирования.

Задачи:

- Проанализировать доступное ПО.
- Выбрать основную идею игры.
- Разработать дизайн игры.
- Разработать алгоритмы для функционирования игры.
- Загрузить все файлы по проекту в репозиторий на GitHub.
- Написать подробную документацию по проекту.

# 1. ТЕОРЕТИЧЕСКИЙ АНАЛИЗ

## 1.1. Выбор идеи

Главные критерии при выборе идеи будущей игры я выделил такие:

- Простота правил
- Математический уклон
- Простота реализации
- Интерес аудитории к жанру

Одним из самых популярных жанров игр, безусловно, считаются головоломки. А потому задача по выбору идеи заметно сузилась: нужно реализовать простую в использовании головоломку с математическим уклоном. А потому я решил реализовать на платформе Scratch всем давно известную игру 2048, чтобы на ее примере познакомить обучающихся со степенями двойки (а все знают, что работа со степенями двойки в информатике и логике крайне важна) и с функционалом выбранного ПО, которое уже изначально предназначалось для знакомства молодых программистов с работой алгоритмов и циклов.

## 1.2. Выбор платформы

Перед тем, как прийти к реализации выбранной идеи, мне необходимо было рассмотреть ПО, которое поможет привести проект к итоговому виду.

Под анализ попали следующие языки и среды программирования:

- JavaScript
- C#
- Python
- Unity

- Scratch

В результате выбор пал на платформу Scratch, потому что она обладает решающими преимуществами:

- Простота реализации проекта
- Наличие опыта использования платформы
- Наглядность работы алгоритмов

### **1.3. Этапы разработки**

Для всего проекта мною были определены следующие этапы к созданию итогового результата:

#### **1. Подготовка.**

На этом этапе был изучен теоретический материал, необходимый для дальнейшей работы: проанализировано различное ПО, которое могло пригодиться, просмотрены курсы по программированию на Scratch (видео и специализированные книги), изучен функционал системы контроля версий GitHub.

#### **2. Дизайн.**

На этапе «Дизайн» были нарисованы спрайты для игрового поля и для каждой ячейки с номером от 2 до 2048 (степени двойки) прямо на платформе Scratch. Цвета фонов и спрайтов специально подбирались, чтобы быть читаемыми на фоне и при этом были приятны глазу.

#### **3. Разработка.**

На этапе «Разработка» в среде Scratch были созданы основные алгоритмы, подсчитывающие и перемещающие значения и соответствующие им спрайты, благодаря которым в итоге и создана игра.

#### **4. Тестирование.**

На этапе «Тестирование» был использован функционал игры по максимуму, чтобы определить устойчивость и целостность всего проекта. Здесь, если обнаруживаются ошибки, баги, несостыковки, они сразу же и исправляются.

#### 5. Документация.

На этапе «Документация» был написан отчет по проекту, сделана презентация для защиты перед заказчиком (преподавателем) и демонстрации основного функционала игры, заполнен Табель рабочего времени, в котором расписан каждый этап проектирования, создан публичный репозиторий в системе контроля версий GitHub и добавлены туда все файлы, связанные с проектом.

#### 6. Защита проекта.

## **2. ФУНКЦИОНИРОВАНИЕ ПРОГРАММЫ**

### **2.1. Правила игры**

1. В каждом раунде появляется плитка номинала «2».
2. Нажатием стрелки игрок может скинуть все плитки игрового поля в одну из 4 сторон. Если при сбрасывании две плитки одного номинала «налетают» одна на другую, то они превращаются в одну, номинал которой равен сумме соединившихся плиток. После каждого хода на свободной секции поля появляется новая плитка номиналом «2» или «4». Если при нажатии кнопки местоположение плиток или их номинал не изменится, то ход не совершается.
3. Если в одной строчке или в одном столбце находится более двух плиток одного номинала, то при сбрасывании они начинают соединяться с той стороны, в которую были направлены. Например, находящиеся в одной строке плитки (4, 4, 4) после хода влево превратятся в (8, 4), а после хода вправо — в (4, 8). Данная обработка неоднозначности позволяет более точно формировать стратегию игры.
4. За каждое соединение игровые очки увеличиваются на номинал получившейся плитки.
5. Игра заканчивается поражением, если после очередного хода невозможно совершить действие.

### **2.2. Геймплей**

После нажатия на кнопку запуска программы появляется поле, на котором в случайном месте появляются два спрайта со значением «2» (см. рис. 2.1).

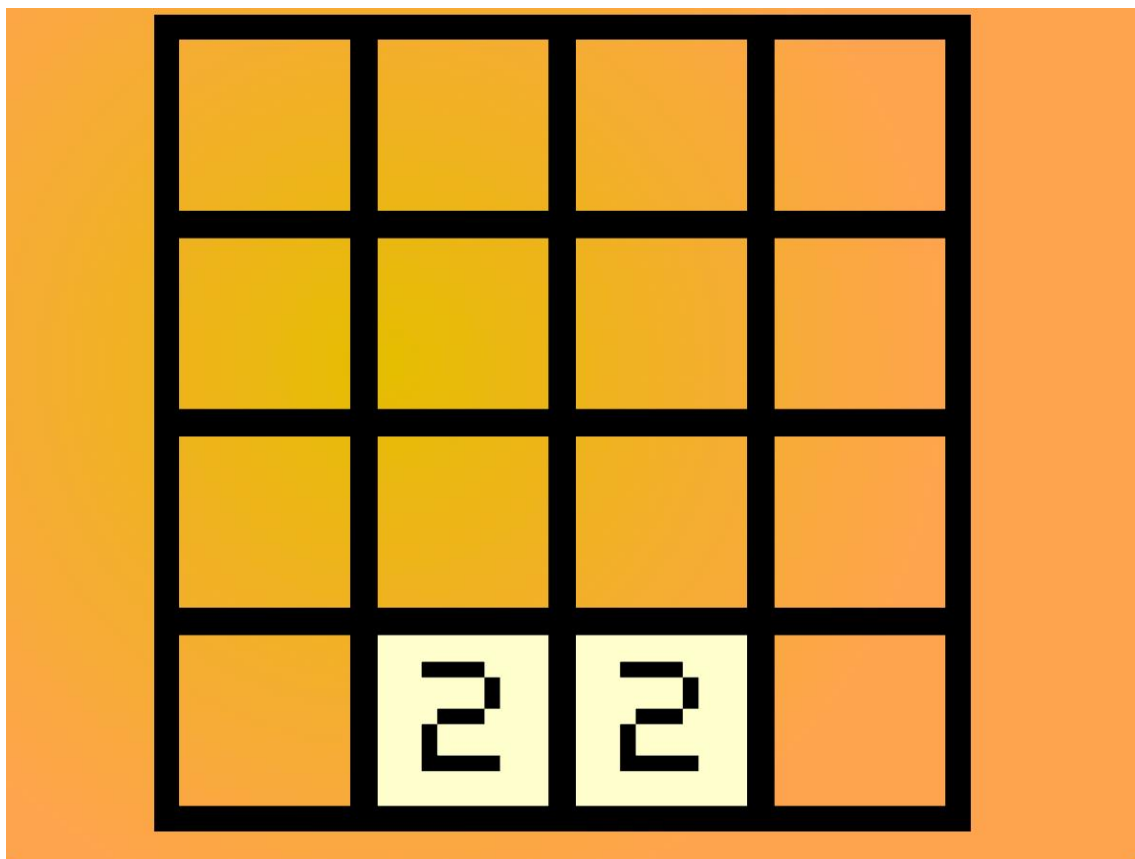


Рис. 2.1. Начальное поле

Далее, согласно вышеописанным правилам, мы можем переместить ячейки в крайнее левое, правое, верхнее, нижнее положение поля, но для того, чтобы ячейки сложились, нам нужно нажать на левую или правую стрелку на клавиатуре. После нажатия на правую стрелку ячейки перемещаются в крайнее правое положение, складываются в «4», а после в случайно выбранной программой ячейке появляется ячейка со значением «2» (см. рис. 2.2).



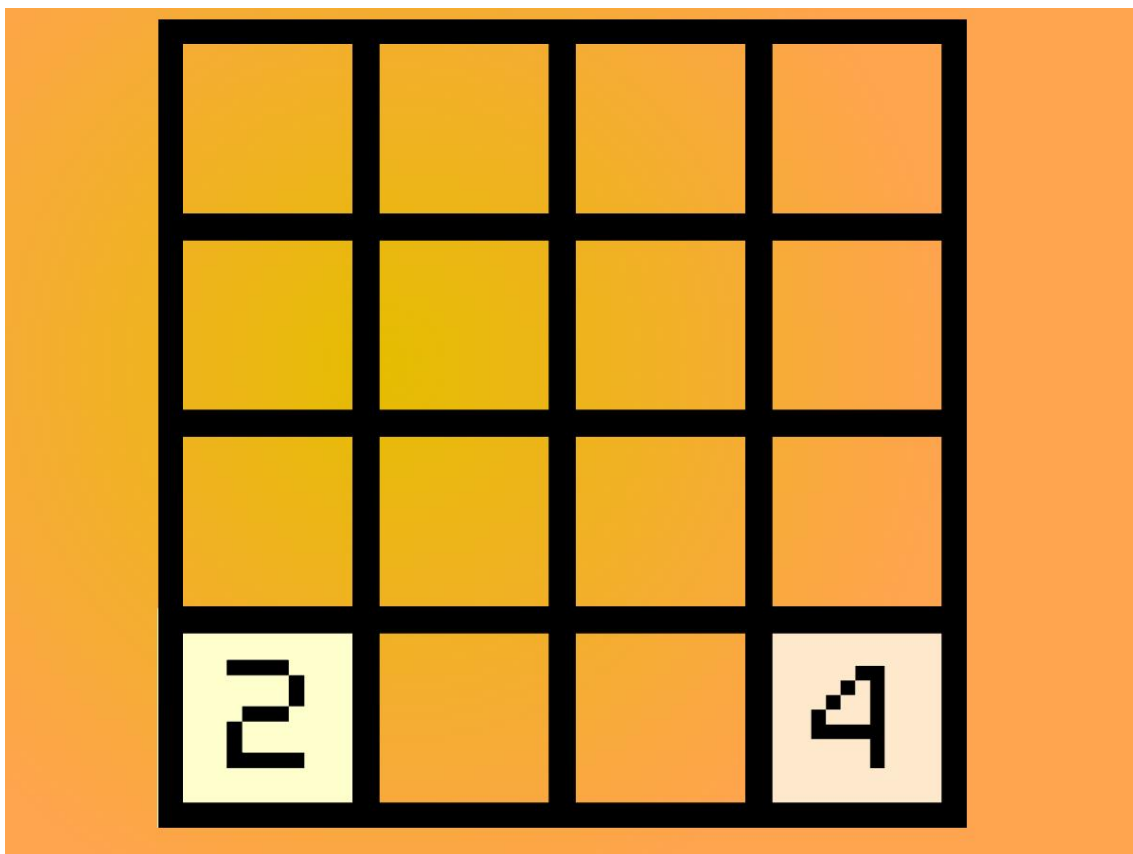
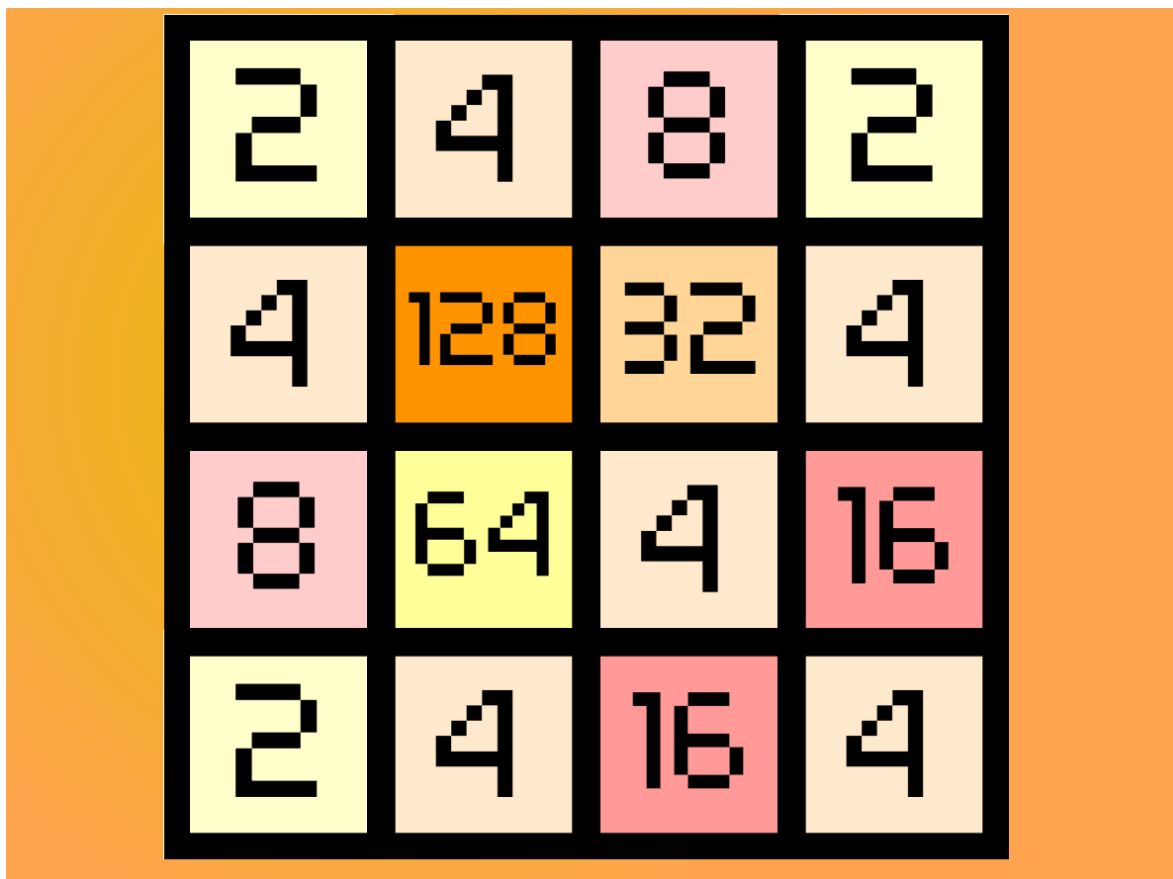


Рис. 2.2. Результат хода. Сложение

Игра заканчивается в двух вариантах-когда игрок добивается значения 2048 в какой-либо ячейке и когда все ячейки заполнены значениями, которые невозможно более сложить. Это, соответственно, выигрыш и проигрыш (см. рис. 2.3).



2	4	8	2
4	128	32	4
8	64	4	16
2	4	16	4

Рис. 2.3. Демонстрация проигрыша

### 3. РАЗРАБОТКА ПРОЕКТА

#### 3.1. Дизайн

Дизайн для спрайтов (границы ячейки, сами ячейки с числами) и фона были нарисованы и спроектированы прямо на платформе Scratch. Для фона был выбран градиент (см. рис. 3.1.), границы ячейки сделаны черными (см. рис. 3.2), а ячейки с разными числами и цветами (см рис. 3.3 и рис. 3.4).

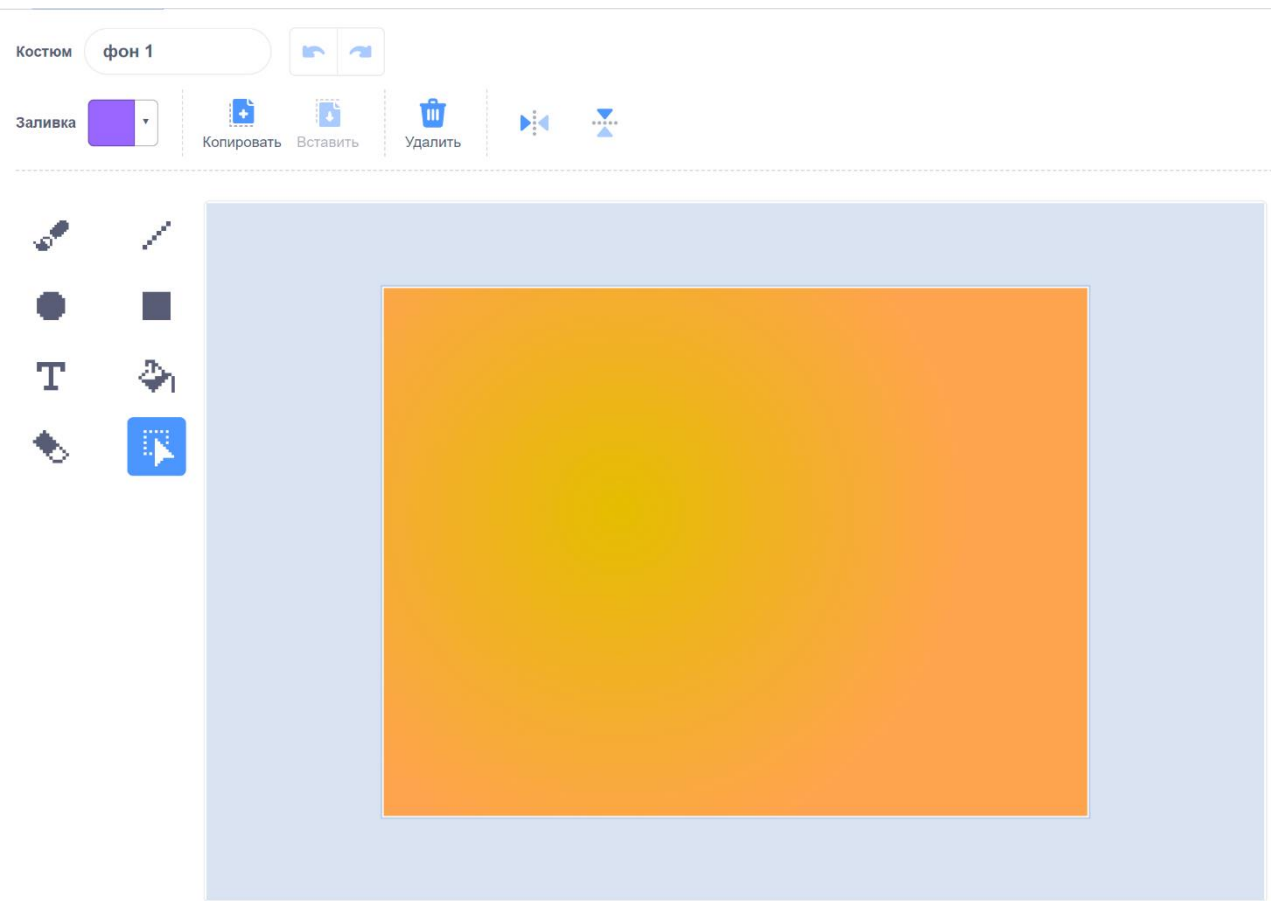


Рис. 3.1. Фон

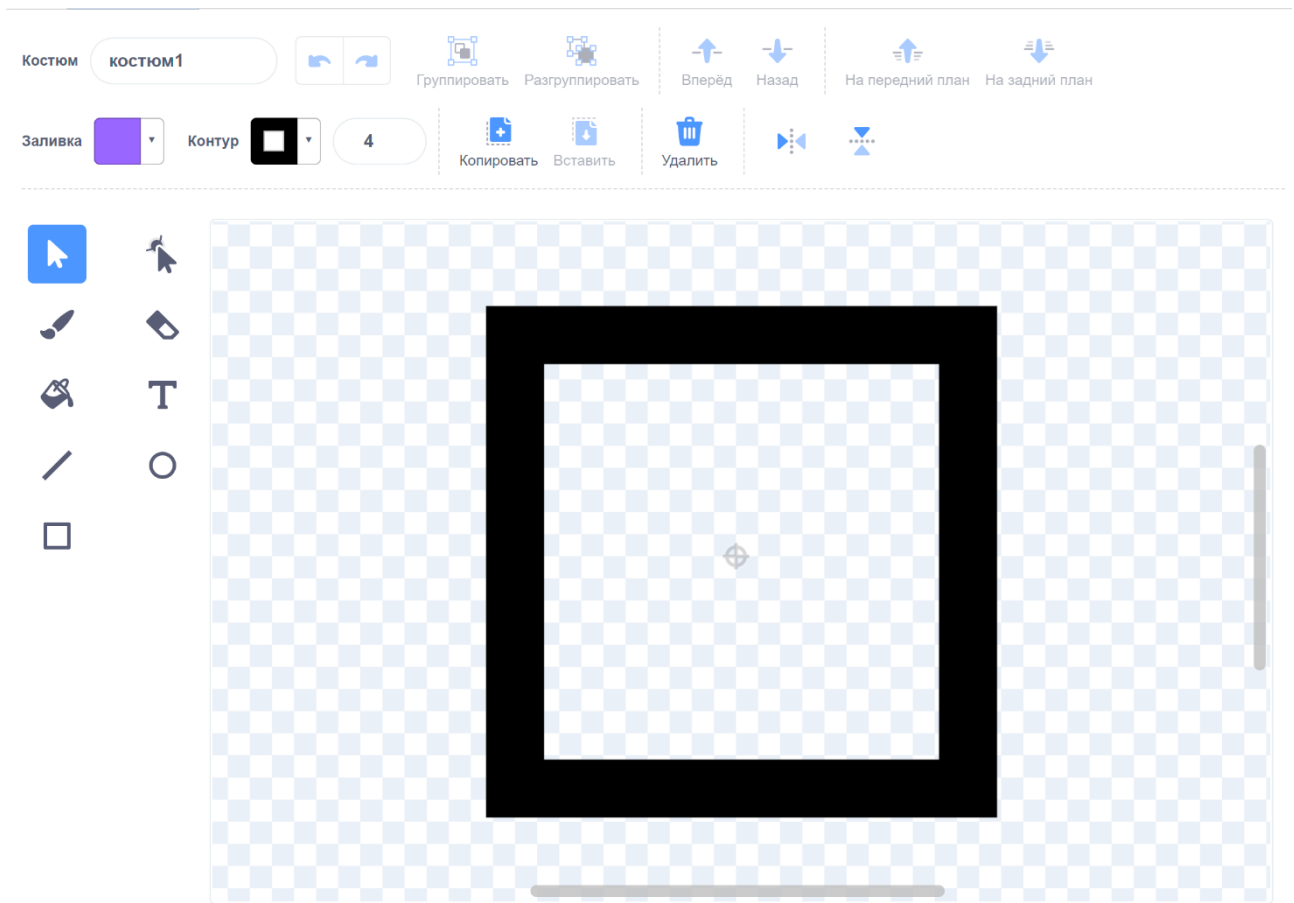


Рис. 3.2. Границы ячеек

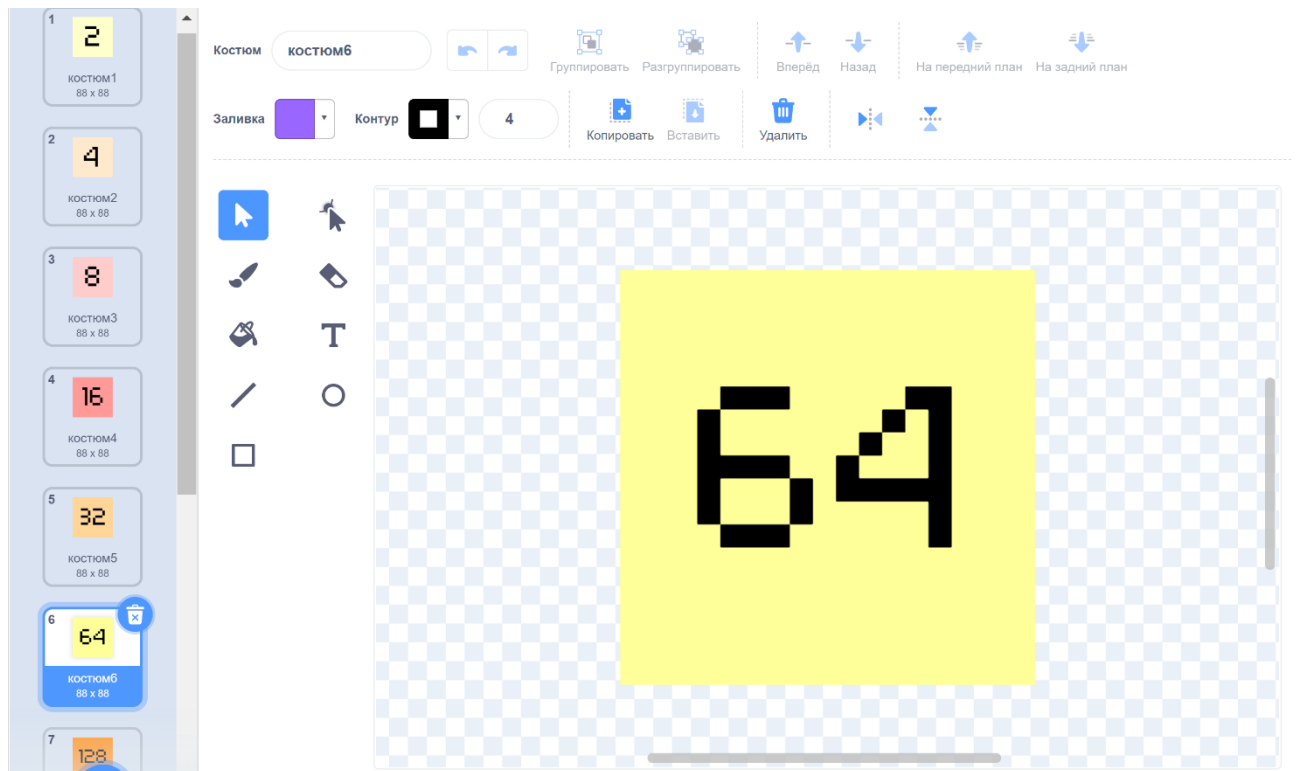


Рис. 3.3. Ячейки

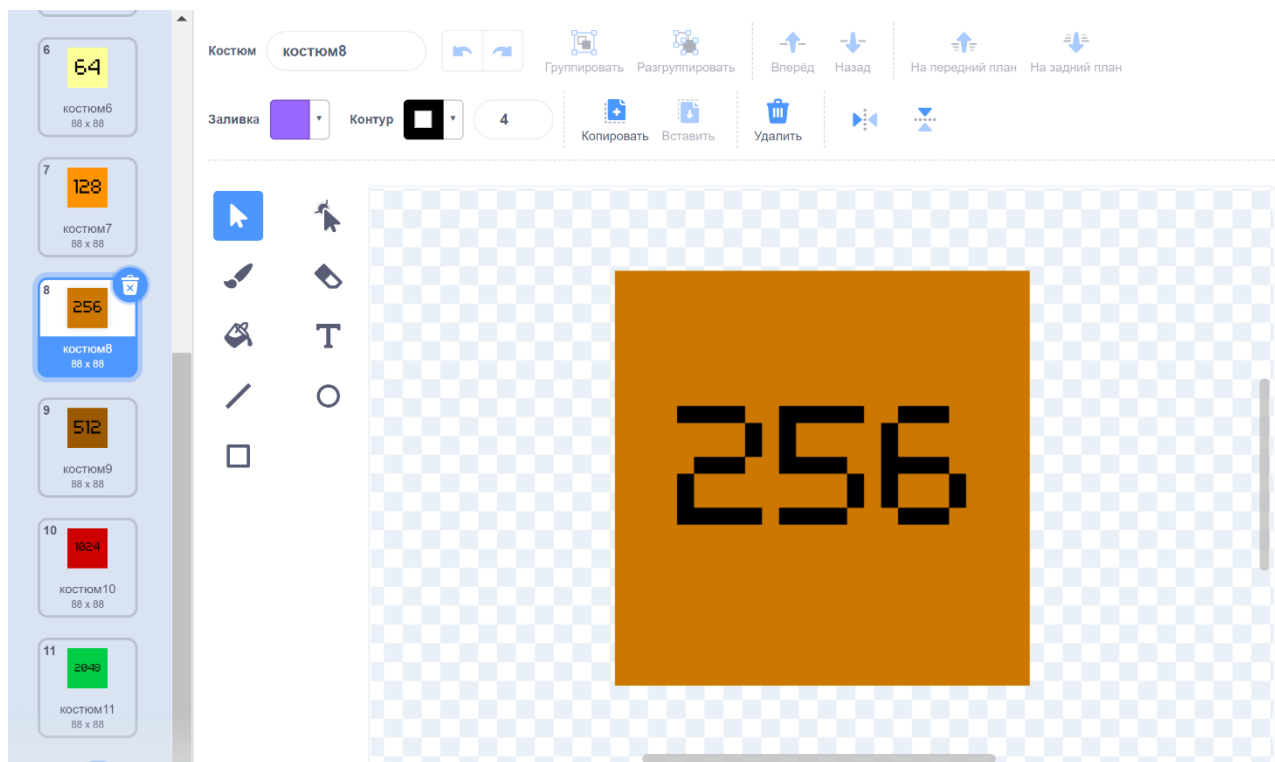


Рис. 3.4. Ячейки

### 3.2. Код из Scratch

Ниже представлен код, который был создан на платформе Scratch. Он наглядно продемонстрирует весь «скелет» игры и расскажет подробно о ее функционировании и логике работы.

Ниже представлен код для спрайта, из которого составляется поле игры. Все происходит благодаря копированию одного спрайта и расставлению его копий по определённым координатам (см. рис. 3.5).

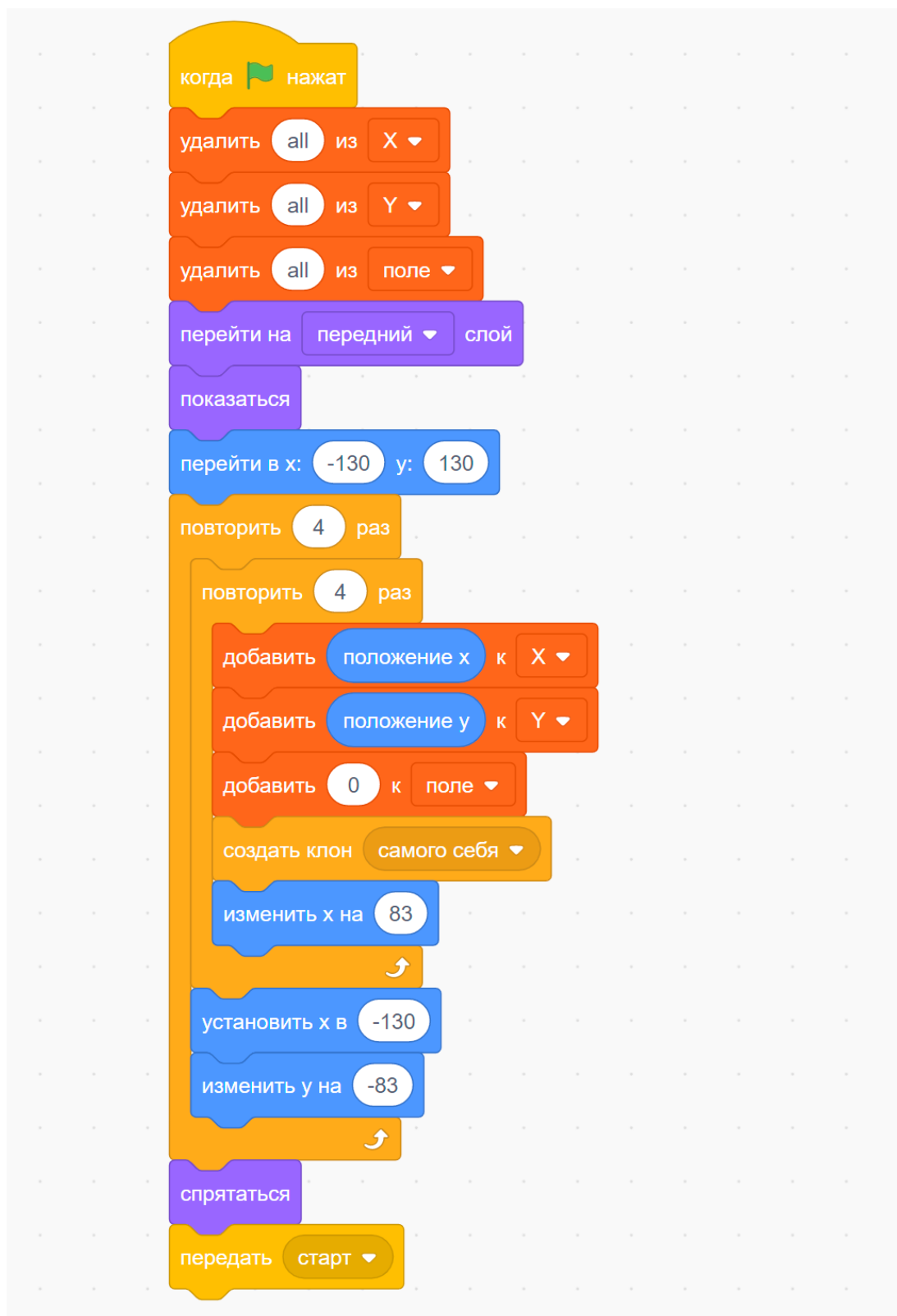


Рис. 3.5. Код для границ ячеек

Ну а далее будет представлен код с кратким описанием для спрайта ячеек, который и занимается всеми событиями, связанными с ним.

Это один из основных алгоритмов (см. рис. 3.6), он вызывает функции по созданию новых ячеек, считывает нажатия на клавиши управления и т.д.

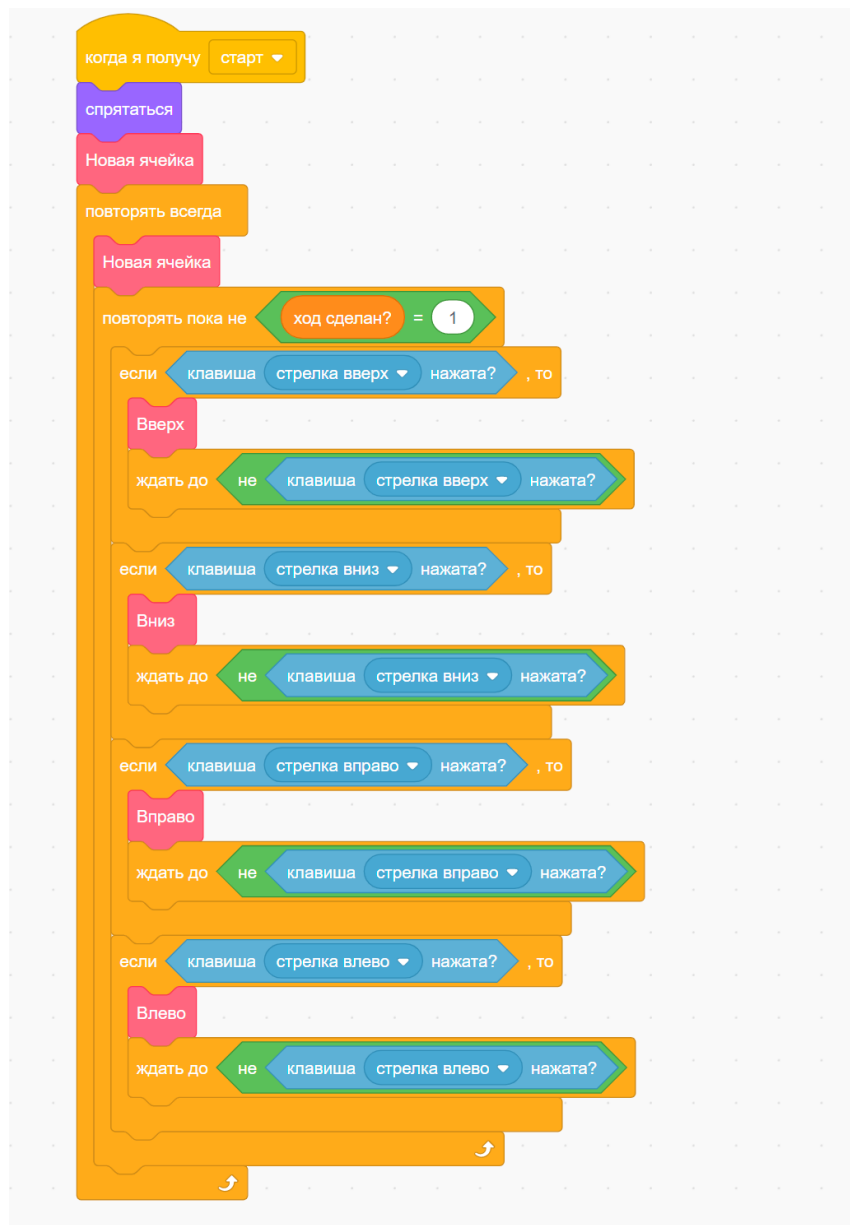


Рис. 3.6. Основной алгоритм

А эти три алгоритма (см. рис. 3.7, рис. 3.8, рис. 3.9) отвечают за создание новой ячейки в случайно выбранном месте.

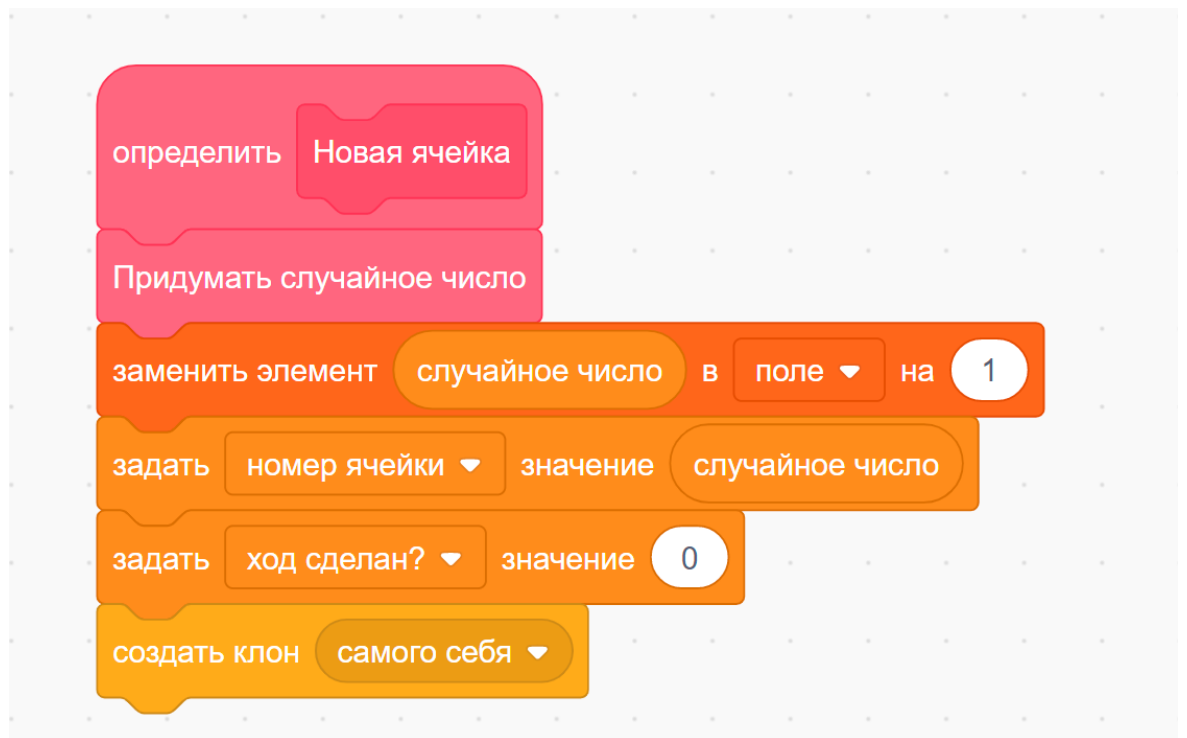


Рис. 3.7. Создание новой ячейки

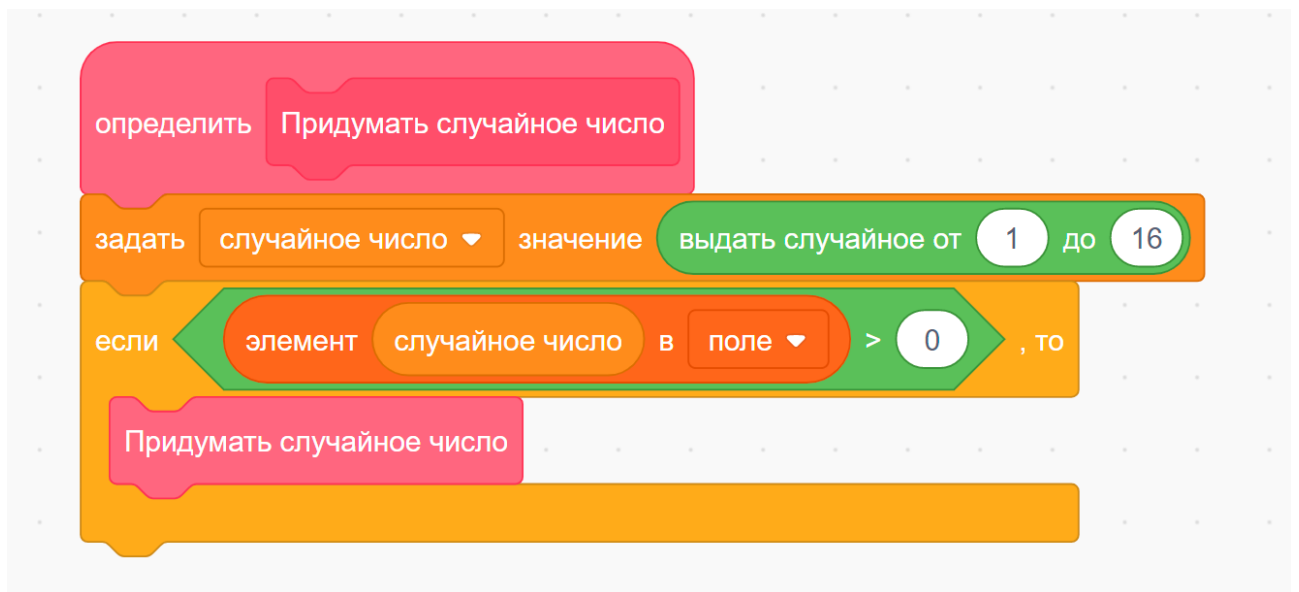


Рис. 3.8. Определение случайного числа



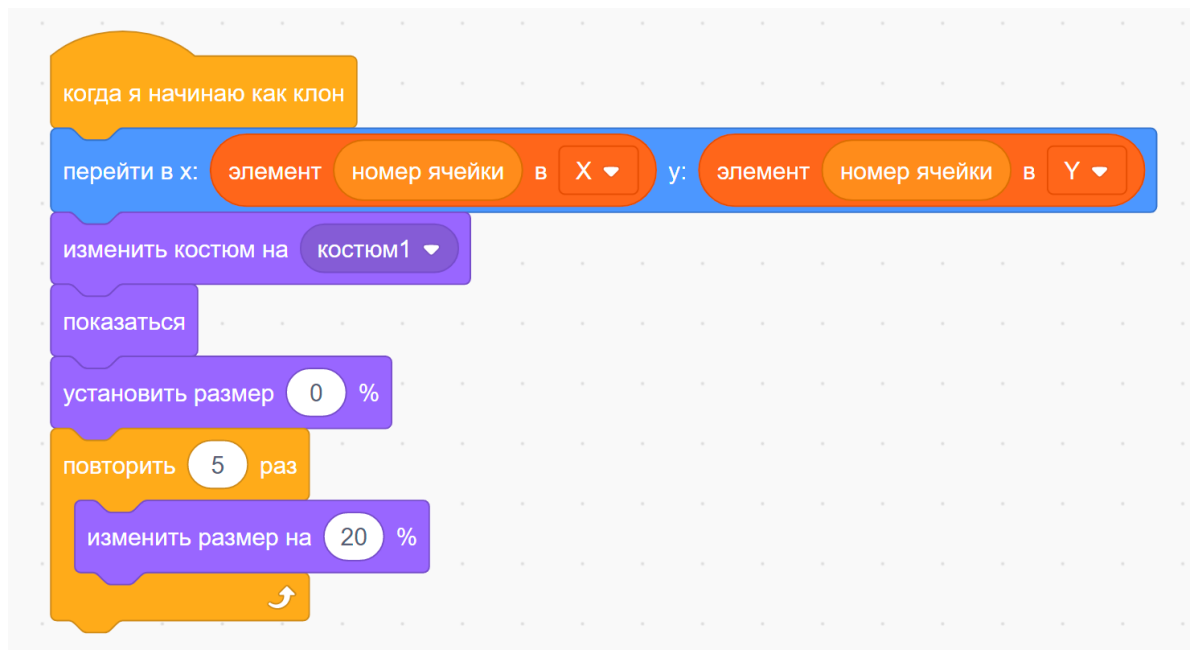


Рис. 3.9. Создание новой ячейки

А следующие алгоритмы (см. рис. 3.10, рис. 3.11, рис. 3.12) отвечают за попытку хода, за перемещение и за сложение спрайтов.

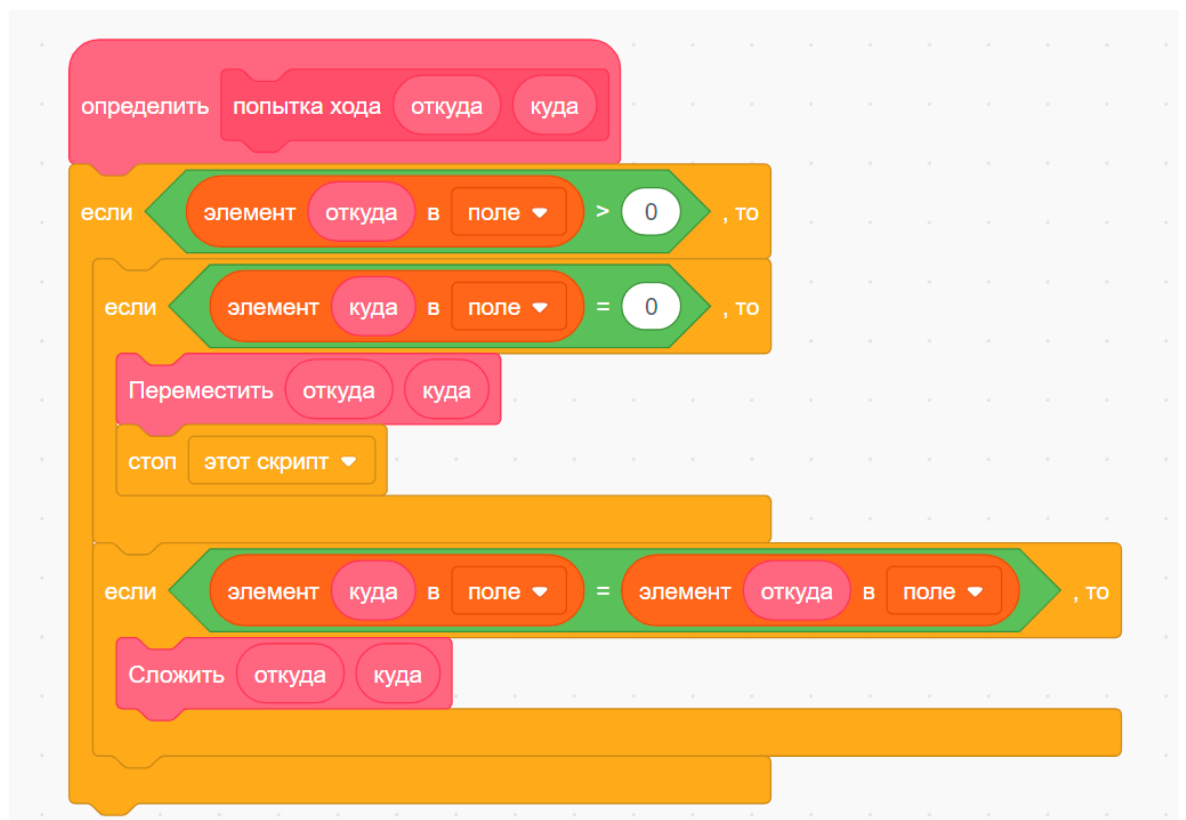


Рис. 3.10. Попытка хода

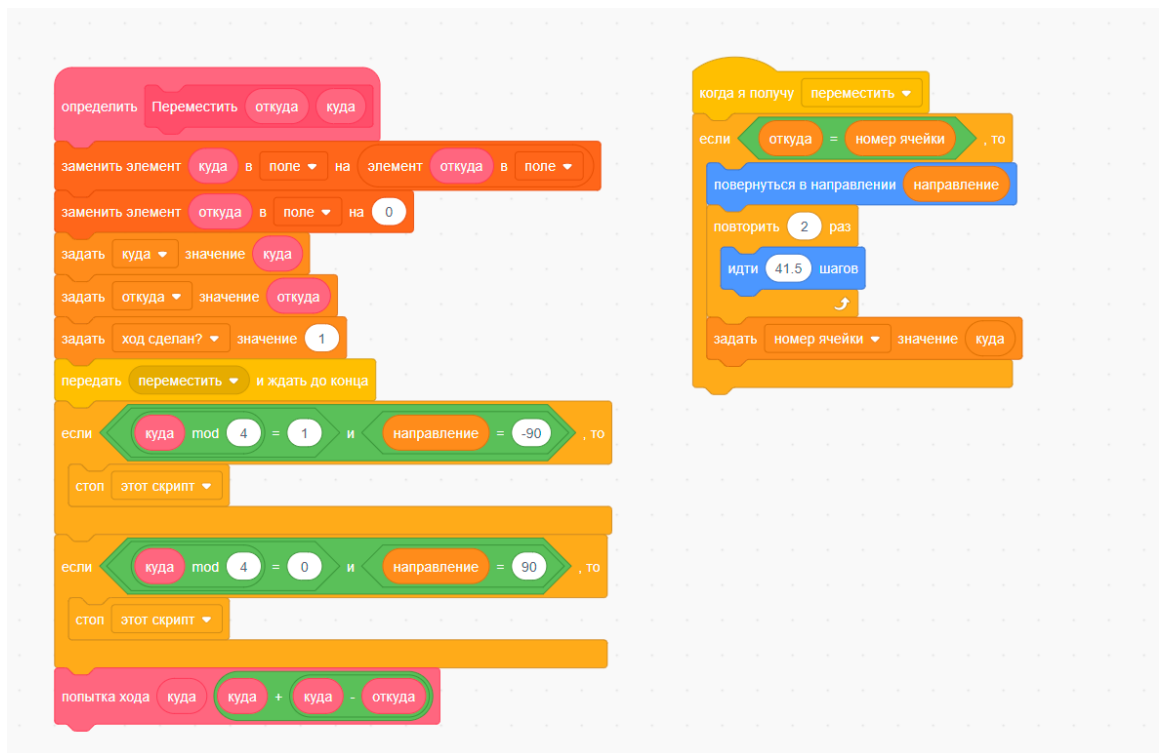


Рис. 3.11. Перемещение ячейки

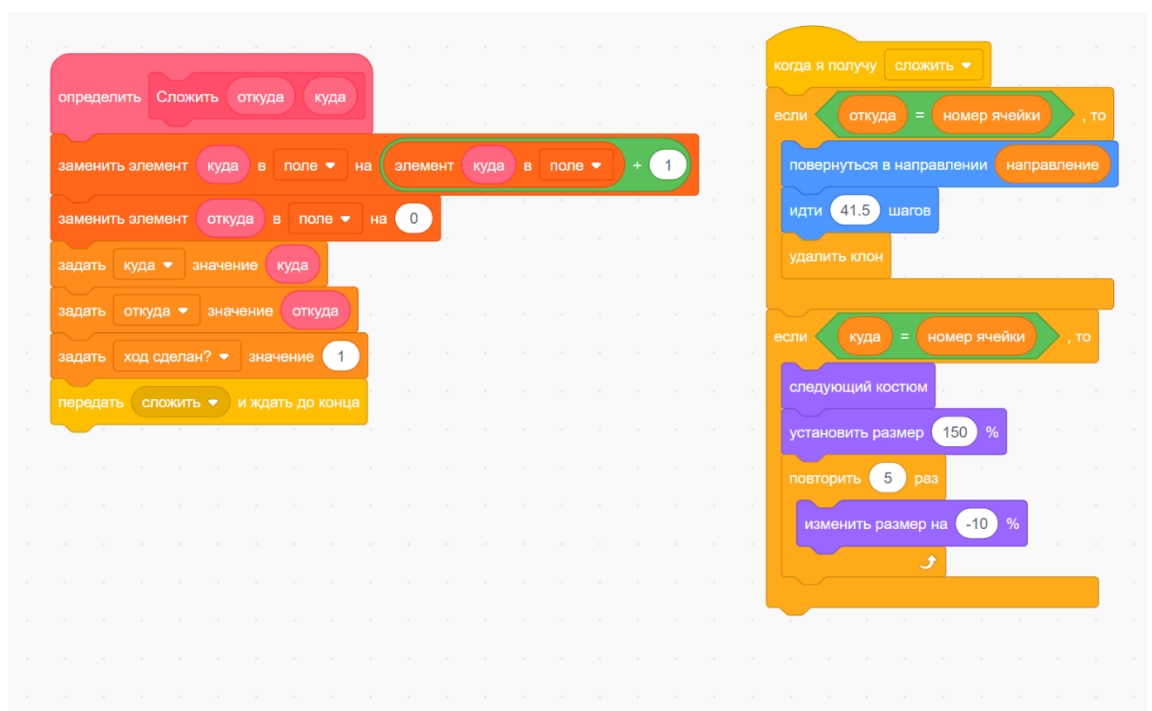


Рис. 3.12. Сложение ячеек

И наконец, последние алгоритмы (см. рис. 3.13) в ответственности за перемещение значений между ячейками.

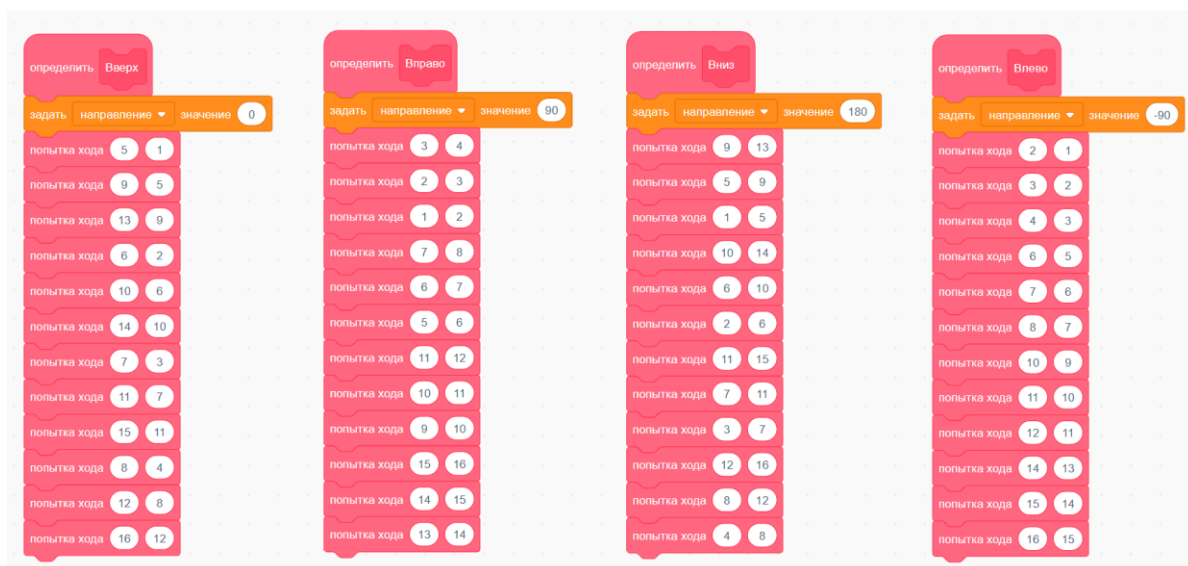


Рис. 3.13. Перемещение значений ячеек

## 4. СИСТЕМА КОНТРОЛЯ ВЕРСИЙ GITHUB

Специально для этого проекта был создан репозиторий (см. рис. 4.1) с файлами, имеющими какое-либо отношение к разработке-программа самой игры, отчет, презентация, табель рабочего времени, скриншоты функционирования.

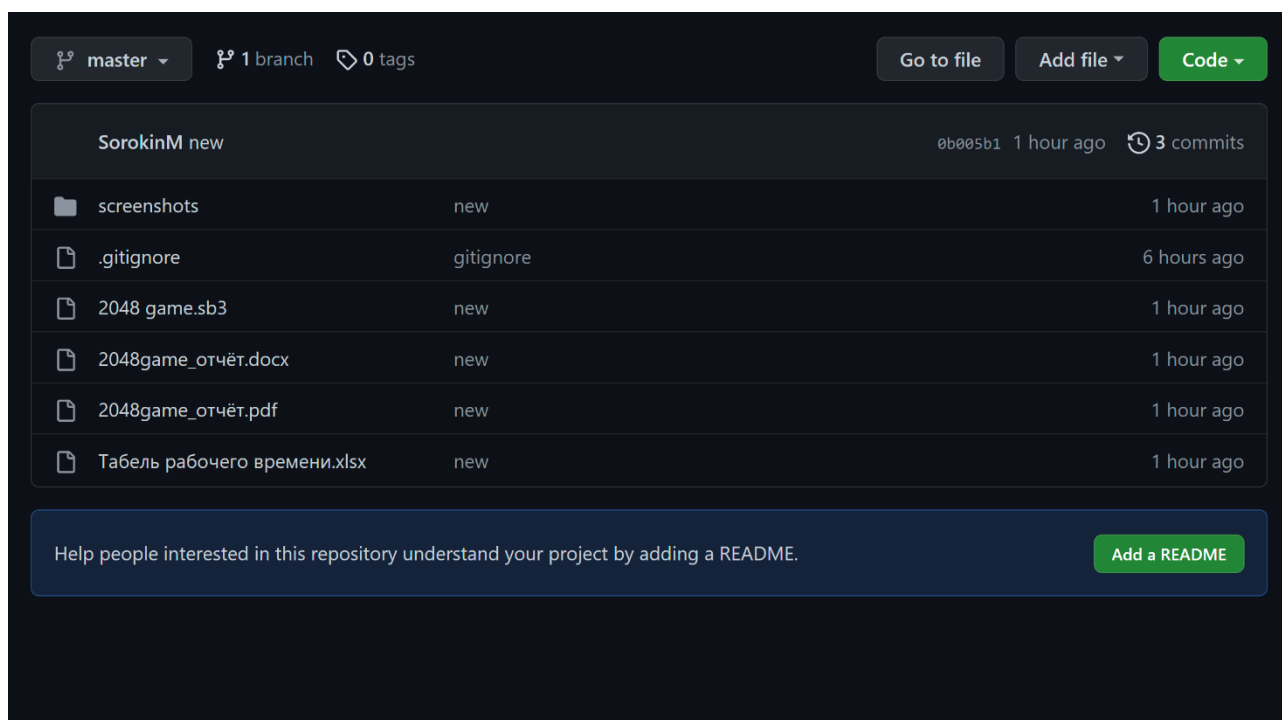


Рис. 4.1. Репозиторий в GitHub

## 5. ТАБЕЛЬ РАБОЧЕГО ВРЕМЕНИ

Для контроля и вычисления времени, потраченного для выполнения проекта, разработки и описания документации, параллельно основной работе мною велся табель рабочего времени (см. рис. 5.1 и рис. 5.2).

Отдельная задача может достигать 60 минут. Если же, по предположению, она заняла времени больше, то ее делил на подзадачи.

В таблице рабочего времени указаны номер задачи, исполнитель, время, потраченное на выполнение задачи, категория задачи, название задачи и ее краткое описание (в отчете представлен табель без краткого описания задач для лучшей читаемости изображений).

Подсчет времени в таблице показал, что на выполнение проекта было затрачено более 18 часов (1091 минута).

№	Исполнитель	Время (в мин.)	Категория	Задача
1	Сорокин М. А.	60	Подготовка	Главная идея
2	Сорокин М. А.	60	Подготовка	Анализ ПО
3	Сорокин М. А.	60	Подготовка	Изучение Scratch_1
4	Сорокин М. А.	60	Подготовка	Изучение Scratch_2
5	Сорокин М. А.	55	Подготовка	Изучение GitHub
6	Сорокин М. А.	45	Сценарий	Установка правил игры
7	Сорокин М. А.	15	Дизайн	Дизайн поля
8	Сорокин М. А.	6	Дизайн	Дизайн фона
9	Сорокин М. А.	35	Дизайн	Дизайн ячеек
10	Сорокин М. А.	35	Дизайн	Подгонка элементов
11	Сорокин М. А.	25	Разработка	Создание поля
12	Сорокин М. А.	20	Тестирование	Тестирование поля
13	Сорокин М. А.	20	Разработка	Создание случайного числа
14	Сорокин М. А.	30	Разработка	Создание новой ячейки
15	Сорокин М. А.	30	Тестирование	Тестирование создания ячейки
16	Сорокин М. А.	20	Разработка	Проверка возможности хода
17	Сорокин М. А.	15	Разработка	Перемещение значения ячейки вверх
18	Сорокин М. А.	5	Тестирование	Тестирование. Вверх.
19	Сорокин М. А.	15	Разработка	Перемещение значения ячейки вниз
20	Сорокин М. А.	5	Тестирование	Тестирование. Вниз.
21	Сорокин М. А.	15	Разработка	Перемещение значения ячейки влево

Рис. 5.1. Табель рабочего времени. 1 половина.

22	Сорокин М. А.	5	Тестирование	Тестирование. Влево.
23	Сорокин М. А.	15	Разработка	Перемещение значения ячейки вправо
24	Сорокин М. А.	5	Тестирование	Тестирование. Вправо.
25	Сорокин М. А.	35	Разработка	Перемещение спрайтов
26	Сорокин М. А.	10	Тестирование	Тестирование. Перемещение спрайтов
27	Сорокин М. А.	20	Разработка	Сложение/замена спрайтов
28	Сорокин М. А.	10	Тестирование	Тестирование. Сложение спрайтов.
29	Сорокин М. А.	15	Разработка	Анимации
30	Сорокин М. А.	5	Тестирование	Тестирование анимаций
31	Сорокин М. А.	40	Тестирование	Основное тестирование
32	Сорокин М. А.	60	Документация	Оформление GitHub
33	Сорокин М. А.	60	Документация	Написание отчёта_1
34	Сорокин М. А.	60	Документация	Написание отчёта_2
35	Сорокин М. А.	60	Документация	Подгонка отчёта под стандарты
36	Сорокин М. А.	60	Документация	Презентация

Рис. 5.2. Табель рабочего времени. 2 половина.

## ЗАКЛЮЧЕНИЕ

В результате выполнения проекта «Игра 2048» мною была спроектирована игра на платформе Scratch.

Были изучены основные элементы программы Scratch и углублены уже имеющиеся познания.

Впервые мною был изучен базовый функционал системы контроля версий GitHub, а потому мой проект и документация выложены в открытый доступ на платформе GitHub.

Во время выполнения работы параллельно велся Табель рабочего времени, благодаря которому можно определить, что на все этапы разработки и написания документации было потрачено суммарно более 18 часов времени (1091 минута). Ну, а благодаря данному проекту обучающихся можно познакомить, как я говорил, со степенями двойки и на примере кода рассказать им о функционировании «более взрослых» языков программирования.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Статья об оригинальной игре «2048». [Электронный ресурс]: портал. – Режим доступа: [https://ru.wikipedia.org/wiki/2048\\_\(игра\)](https://ru.wikipedia.org/wiki/2048_(игра)) (дата обращения: 17.06.2022).
2. Scratch для юных программистов [Электронный ресурс]: портал. – Режим доступа: <https://mpa71.ru/wp-content/uploads/Scratch-dlya-yunyh-programmistov.pdf> (дата обращения: 17.06.2022).
3. Scratch Викторина | Списки, текст в речь, подсчет правильных ответов [Электронный ресурс]: портал. – Режим доступа: <https://www.youtube.com/watch?v=eZRxbgLPec0> (дата обращения: 17.06.2022).
4. Уроки Scratch 3.0 [Электронный ресурс]: портал. – Режим доступа: [https://www.youtube.com/watch?v=Vc8moYRG-bE&list=PLdzeMLV8u\\_l4ssPz6sa5VeN6xvJv6QAQk](https://www.youtube.com/watch?v=Vc8moYRG-bE&list=PLdzeMLV8u_l4ssPz6sa5VeN6xvJv6QAQk) (дата обращения: 17.06.2022).
5. Игра 2048. Как победить. [Электронный ресурс]: портал. – Режим доступа: <https://www.youtube.com/watch?v=wh8U1lXtGgE> (дата обращения: 17.06.2022).
6. Знакомство с Git и GitHub. [Электронный ресурс]: портал. – Режим доступа: <https://medium.com/nuances-of-programming/знакомство-с-git-и-github-руководство-для-начинающих-54ea2567d76c> (дата обращения: 17.06.2022).
7. Как начать работать с GitHub. [Электронный ресурс]: портал. – Режим доступа: <https://habr.com/ru/post/125799/> (дата обращения: 17.06.2022).
8. GitHub: настройка и первая публикация проекта. [Электронный ресурс]: портал. – Режим доступа: <https://gb.ru/posts/github-nastrojka-i-pervaya-publikaciya-proekta> (дата обращения: 17.06.2022).



9. Изучение GitHub. [Электронный ресурс]: портал. – Режим доступа: <https://www.youtube.com/watch?v=JfpCicDUMKc&t=7s> (дата обращения: 17.06.2022).