

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №1

Выполнила:

Красюк Карина

K3441

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

СОДЕРЖАНИЕ

Задача.....	3
Ход работы	3
1 Игра Flexbox Froggy	3
2 Игра Grid Garden	6
3 Игра LearnGitBranching.....	8
Вывод	11

Задача

Практическое освоение базовых технологий веб-разработки и систем контроля версий через прохождение трёх интерактивных обучающих игр:

- Flexbox Froggy (по CSS Flexbox)
- CSS Grid Garden (по CSS Grid)
- Learn Git Branching (по основам Git)

Ход работы

1 Игра Flexbox Froggy

Были пройдены все 24 уровня игры. На рисунках 1,2,3,4,5,6 можно увидеть результаты.

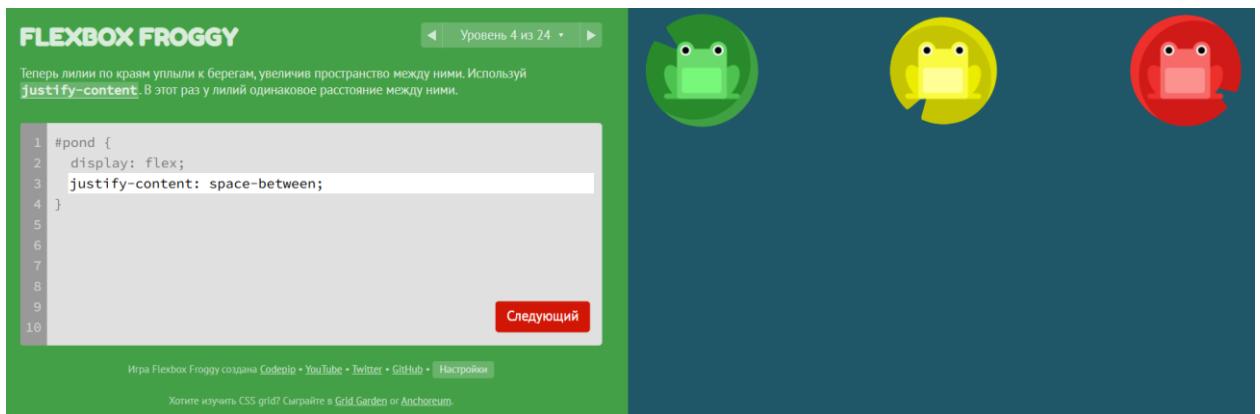


Рисунок 1 – Прохождение 4 уровня игры Flexbox Froggy

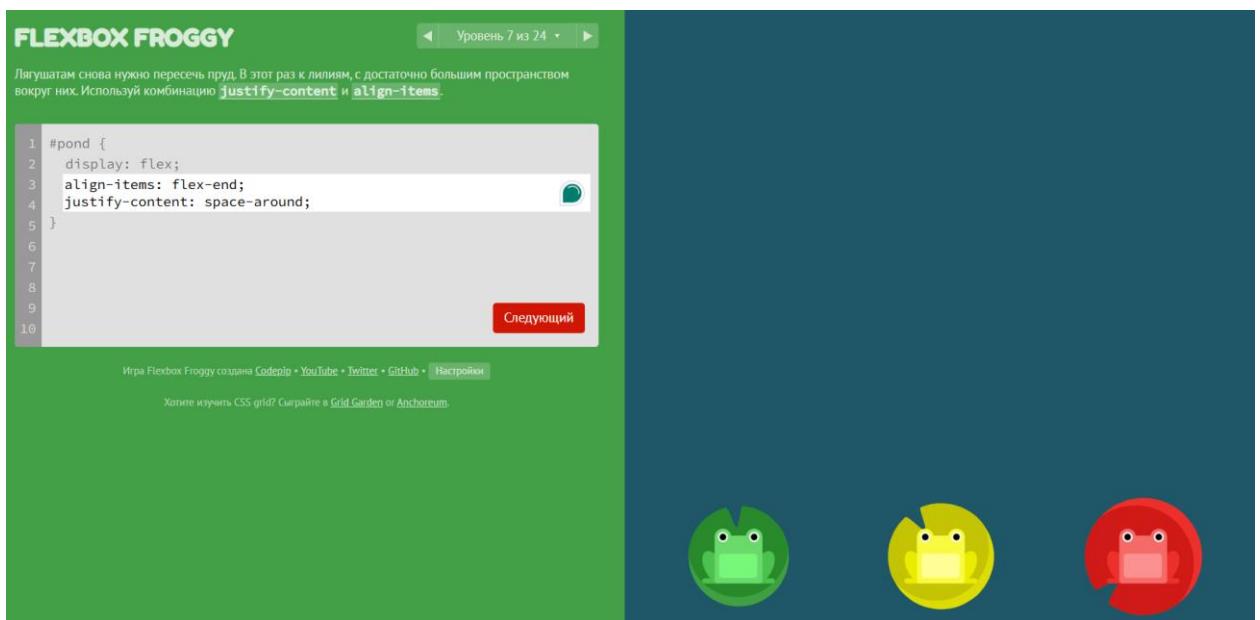


Рисунок 2 – Прохождение 7 уровня игры Flexbox Froggy

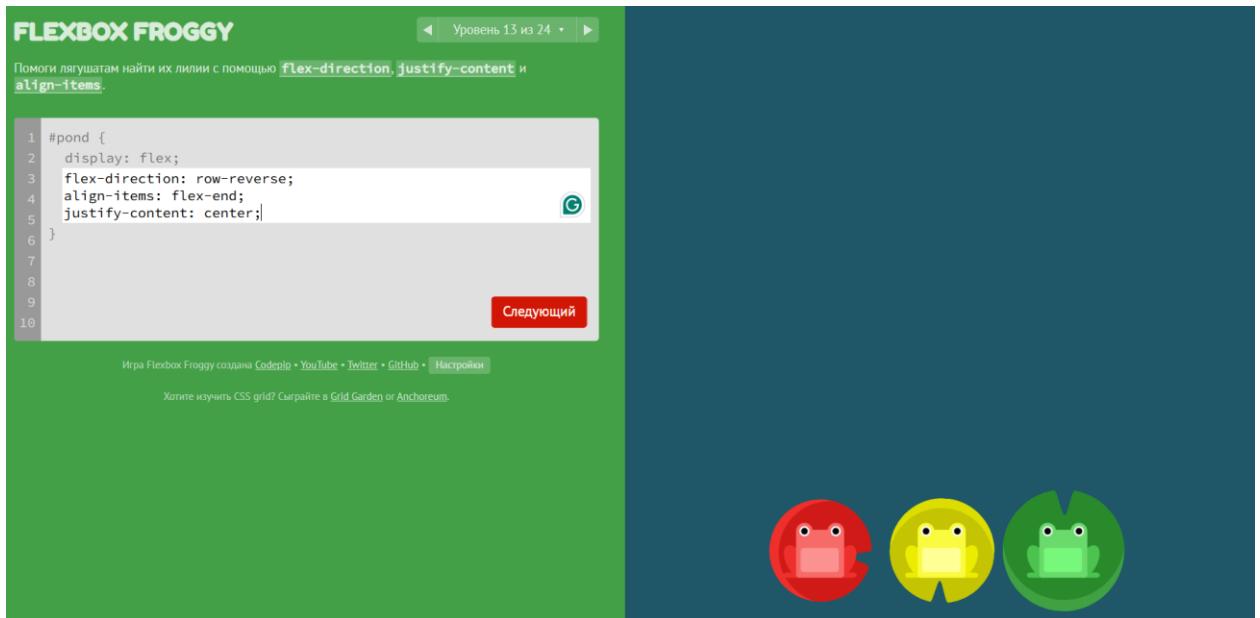


Рисунок 3 – Прохождение 14 уровня игры Flexbox Froggy

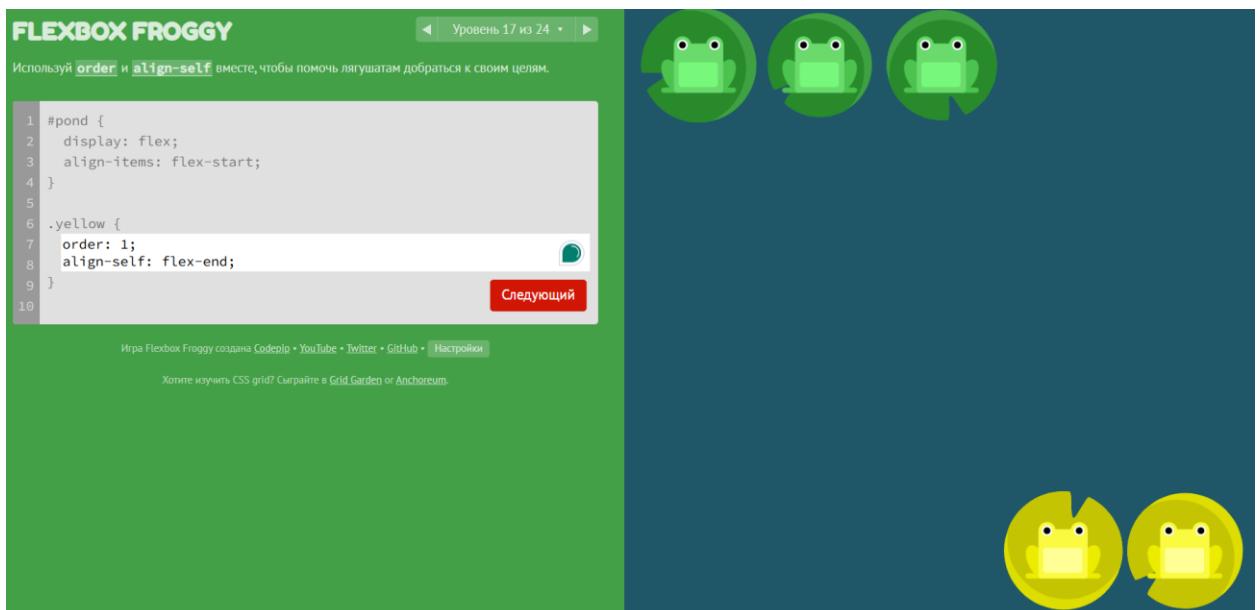


Рисунок 4 – Прохождение 17 уровня игры Flexbox Froggy

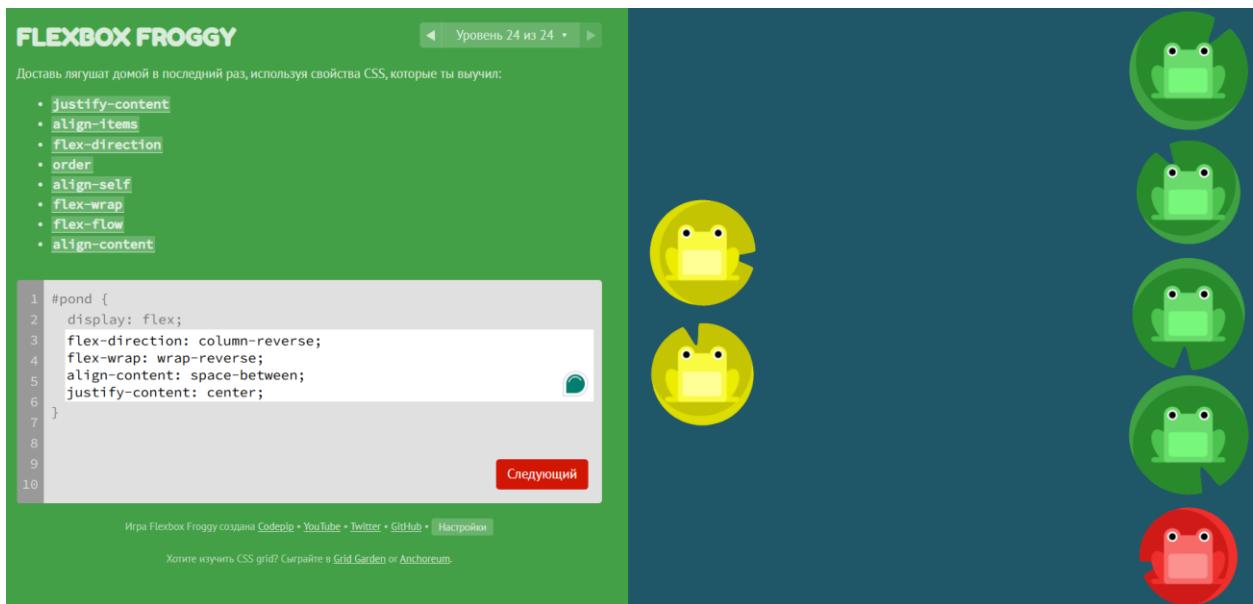


Рисунок 5 – Прохождение 24 уровня игры Flexbox Froggy



Рисунок 6 – Пройденные все уровни игры Flexbox Froggy

В процессе прохождения Flexbox Froggy были рассмотрены основные свойства CSS Flexbox:

- `justify-content` – задаёт, как элементы будут распределяться по основной оси: прижаты к началу или концу, размещены по центру либо разнесены с равными интервалами.

- align-items – определяет, как элементы выстраиваются по поперечной оси: снизу, сверху, по центру, по базовой линии или растягиваются на всю высоту контейнера.
- flex-direction – указывает, в каком направлении будут расположены элементы: горизонтально, вертикально или в обратном порядке.
- order – позволяет менять порядок отдельных элементов.
- flex-wrap – задаёт, могут ли элементы переноситься на следующую строку или должны оставаться в одной.
- align-content – управляет тем, как располагаются сразу несколько рядов элементов внутри контейнера, когда их больше одного.

2 Игра Grid Garden

Были пройдены все 28 уровней игры. На рисунках 7,8,9,10 можно увидеть результаты.

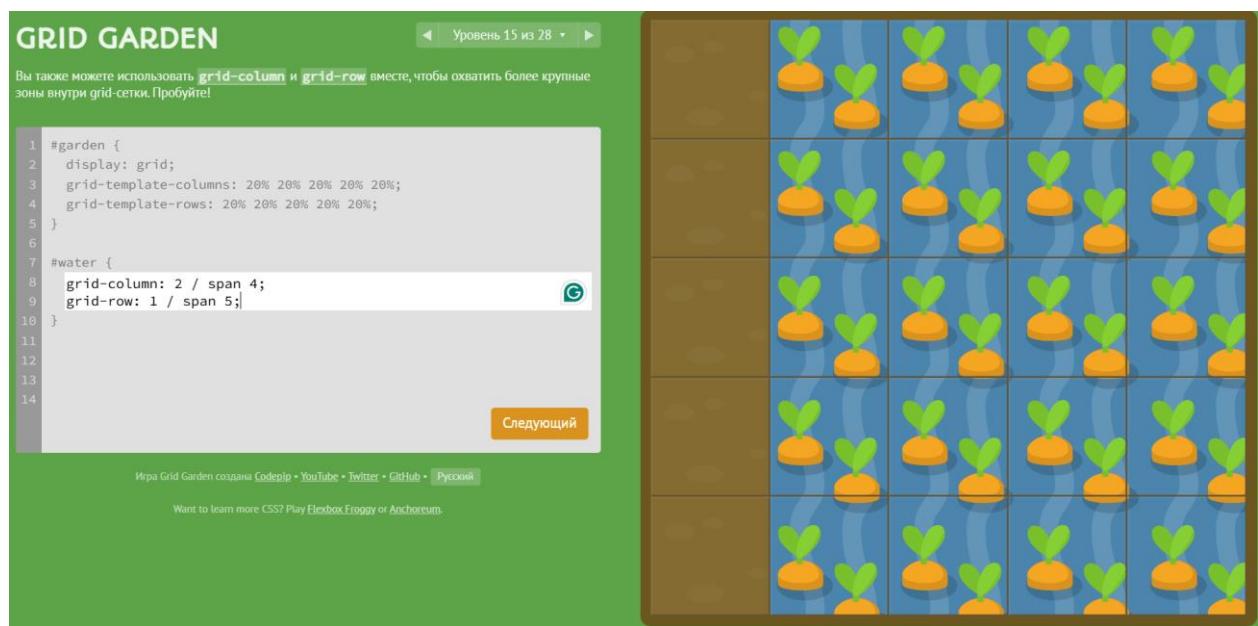


Рисунок 7 – Прохождение 15 уровня игры Grid Garden

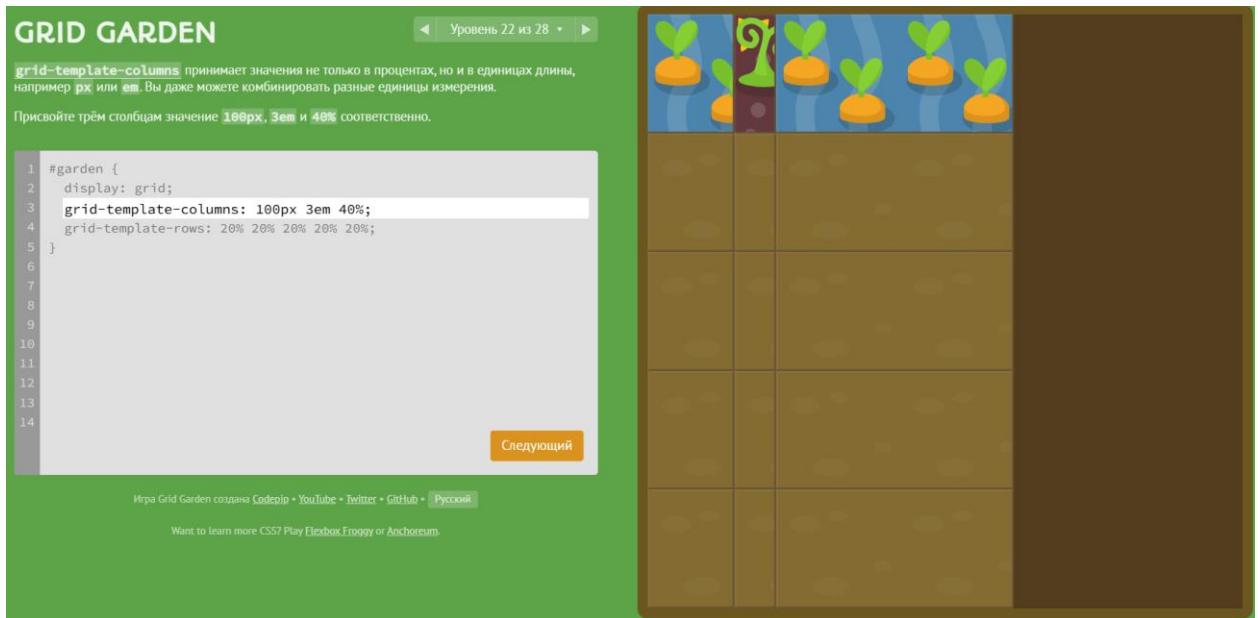


Рисунок 8 – Прохождение 22 уровня игры Grid Garden

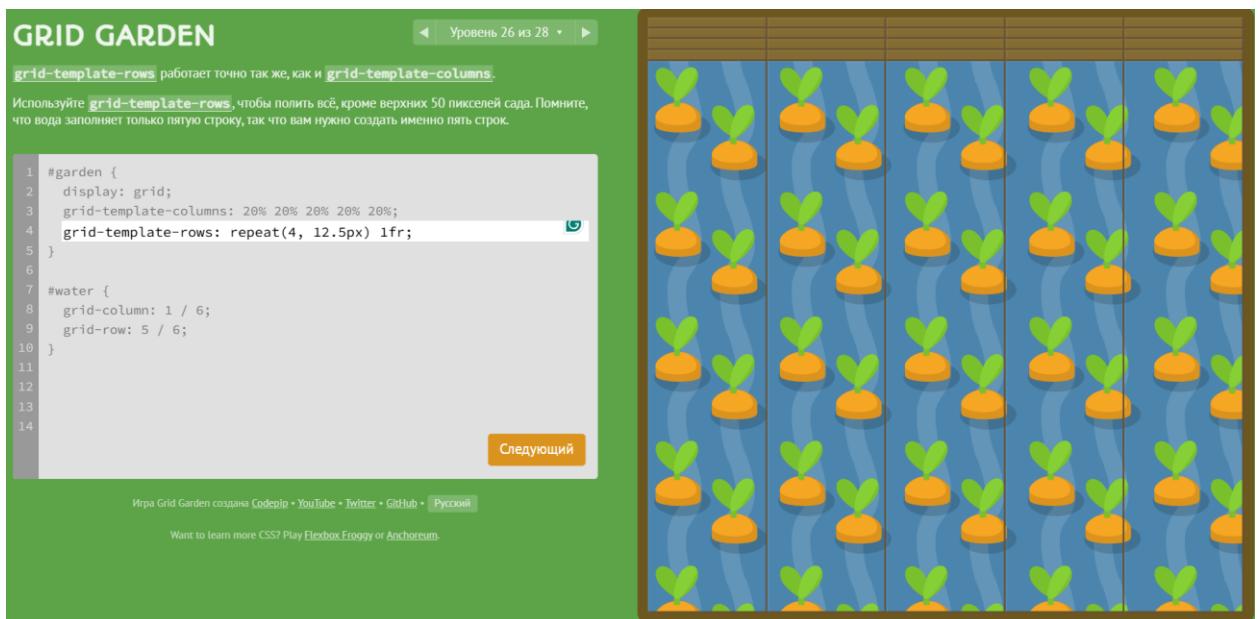


Рисунок 9 – Прохождение 26 уровня игры Grid Garden

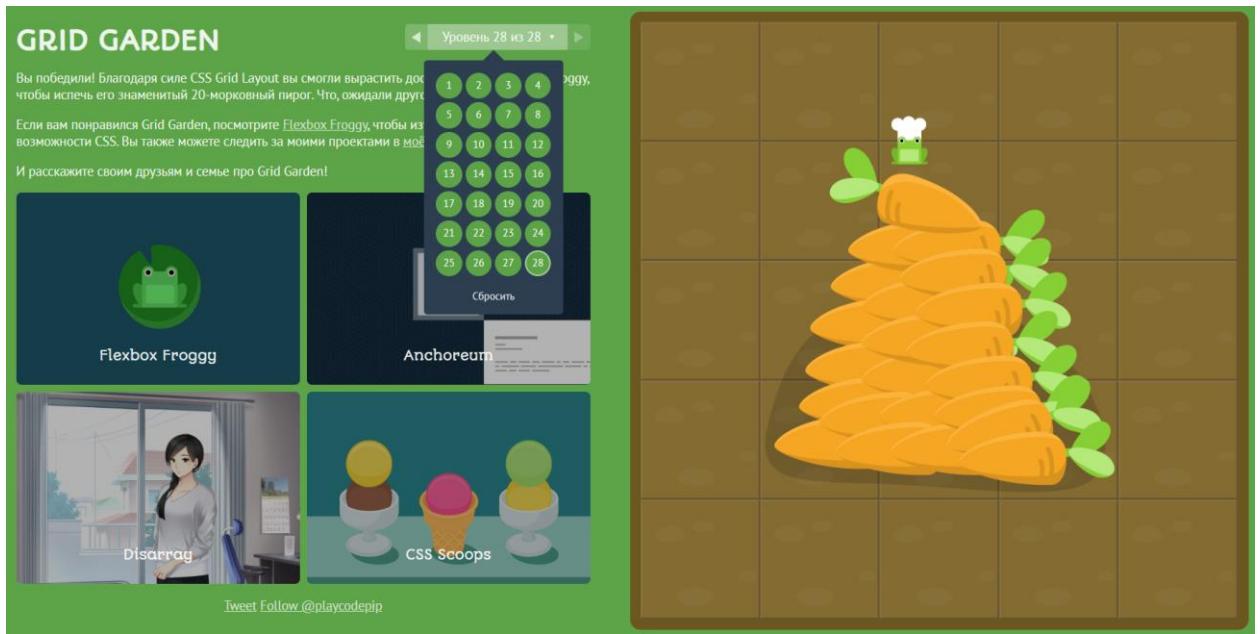


Рисунок 10 – Пройденные все уровни игры Grid Garden

При прохождении игры Grid Garden были освоены ключевые свойства CSS Grid:

- `grid-column-start` и `grid-column-end` – с их помощью элемент размещается в нужных колонках.
- `span` – это ключевое слово, которое задает, сколько строк или колонок должен занять элемент.
- `grid-column` – короткое свойство, объединяющее `grid-column-start` и `grid-column-end` для удобства.
- `grid-area` – свойство для размещения элемента в заранее заданной именованной области сетки.
- `grid-template` – свойство, которое определяет основную структуру сетки, задавая количество и размеры строк и колонок.

3 Игра [LearnGitBranching](#)

Были пройдены первые 4 урока “Основы” и все уроки “Удалённые репозитории”. На рисунках 11,12 можно увидеть результаты.

Основы Удаленные репозитории

Введение

Хорошо подобранные введение в основные команды git

2: Введение в Git



Едем дальше

Следующая порция абсолютной git-крутизны. Проголодались?

1: Теряем голову, или detached HEAD



Перемещаем труды туда-сюда

Не стесняйтесь менять историю

1: Введение в Cherry-pick



Сборная солянка

Ассорти из приёмов работы с Git, хитростей и советов

1: Выберем один коммит.



Продвинутый уровень

Рисунок 11 – Пройденные первые 4 урока “Основы” игры LearnGitBranching

Выбери уровень

Основы Удаленные репозитории

Push & Pull - удалённые репозитории в Git!

Настало время поделиться своими единичками и нулями. Время коллективного программирования

1: Введение в клонирование



Через origin – к звёздам. Продвинутое использование Git Remotes

Весело было быть всесильным мудрым правителем...

1: Push Master!



Рисунок 12 – Пройденные уроки “Удалённые репозитории” игры LearnGitBranching

В рамках изучения системы контроля версий Git были освоены основные принципы работы с ветками и коммитами:

- `git commit` – создание точек сохранения текущих изменений в репозитории
 - `git branch` и `git checkout` – инструменты для создания новых направлений разработки и переключения между ними
 - `git merge` – слияние различных веток в единую кодобазу
 - `git rebase` – перестраивание истории изменений для создания линейной последовательности коммитов

Вывод

Работа позволила приобрести важнейшие навыки для фронтенд-разработчика:

- создание адаптивных макетов с помощью технологий Flexbox и CSS Grid
- освоение системы контроля версий Git, включая операции с ветками и объединение изменений

Наработанные умения будут активно использоваться при выполнении последующих практических заданий.