

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №1

Выполнила:
Крапивина Яна

Группа
J3212

Проверил:
Добряков Д. И.

Санкт-Петербург

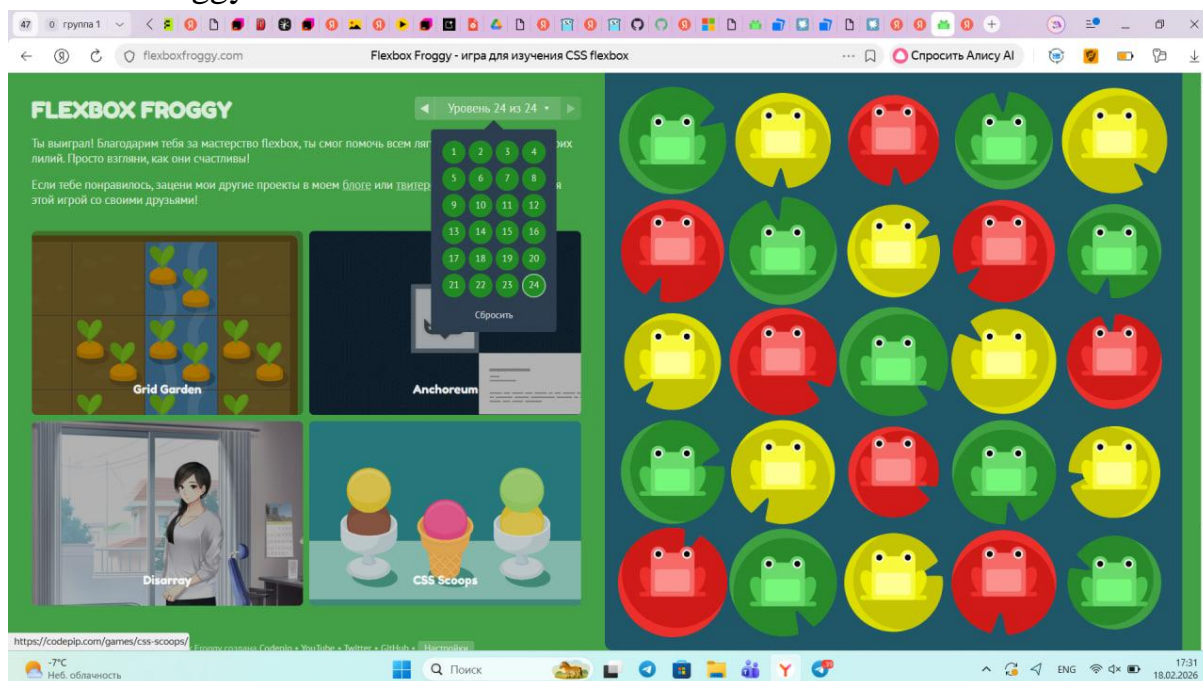
2026 г.

Задача

Целью работы являлось получение и закрепление практических навыков работы с CSS Flexbox и CSS Grid с помощью игр, а также изучение команд Git, включая работу с удаленными репозиториями.

Ход работы

1. Flexbox Froggy

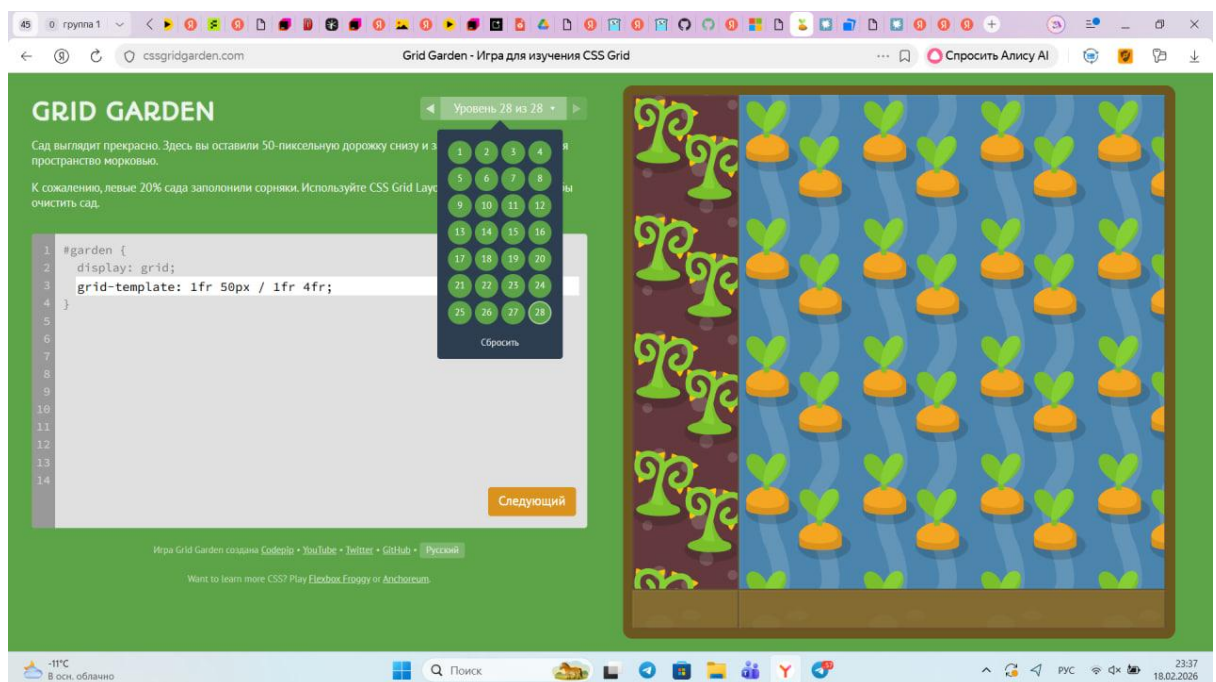


Были изучены и отработаны основные свойства Flexbox:

- **Justify-content** (выравнивает элементы горизонтально и принимает следующие значения: flex-start, flex-end, center, space-between, space-around)
- **Align-items** (выравнивает элементы вертикально и принимает следующие значения: flex-start, flex-end, center, baseline, stretch)
- **Flex-direction** (задает направление, в котором будут расположены элементы в контейнере, и принимает следующие значения: row, row-reverse, column, column-reverse)
- **Order** (изменение порядка отображения конкретных элементов: значение этого свойства у элементов равно 0, но мы можем задать положительное или отрицательное целое число)

- Align-self (применяется к определенному элементу и принимает такие же значения как и align-items)
- Flex-wrap (определяет поведение элементов, когда они не помещаются в одну линию внутри контейнера: nowrap, wrap, wrap-reverse)
- Flex-flow (комбинирует flex-direction и flex-wrap: принимает их значения, разделенные пробелом)
- Align-content (как несколько рядов должны отделяться друг от друга: flex-start, flex-end, center, space-between, space-around, stretch)

2. Grid Garden

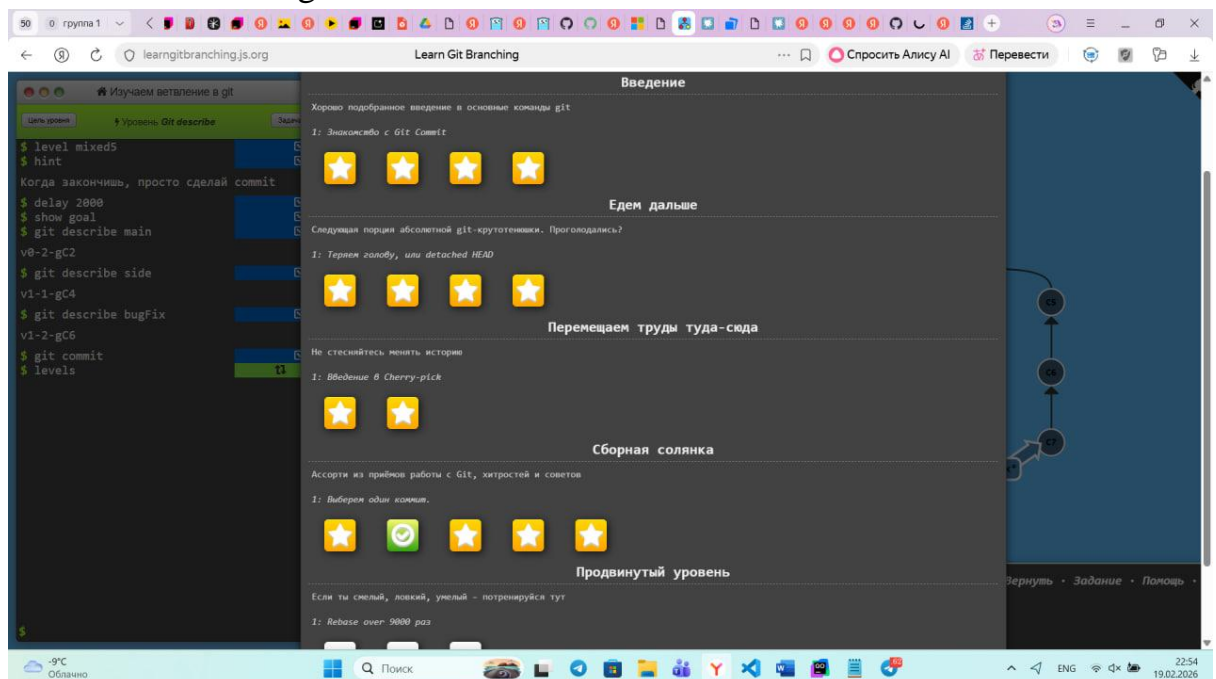


При прохождении всех уровней были изучены возможности CSS Grid:

- Grid-column-start: num (покрывает зону, начинающуюся на num grid-линии по вертикали)
- Grid-column-end: num (чтобы увеличить элемент между несколькими столбцами)
- Span (вместо начальной и конечной позиции строк можно задавать ширину)
- Grid-column(row): start/end (краткая форма)
- Grid-area: row-start/column-start/row-end/column-end

- Order (изначально все элементы имеют order, равный 0, но этому свойству можно присвоить любое положительное или отрицательное значение)
- Grid-template-columns и grid-template-rows (задают размеры и количество колонок и строк в сетке)
- Функция repeat (5, 20%) (5 столбцов, по 20% ширины каждый)
- Единица измерения fr (если двум элементам задано 1fr и 3fr соответственно, то пространство будет поделено на 4 одинаковые части. Первый элемент займет $\frac{1}{4}$, а второй оставшиеся $\frac{3}{4}$ пространства)
- Grid-template (сокращенный вариант)

3. Learn Git Branching



Основы Git:

- Git commit
- Git branch
- Git checkout
- Git checkout -b
- Git merge

Перемещение по истории:

- Git checkout <commit>
- Git checkout HEAD^

- Git checkout HEAD~n
- Git branch -f
- Git reset
- Git reset --hard

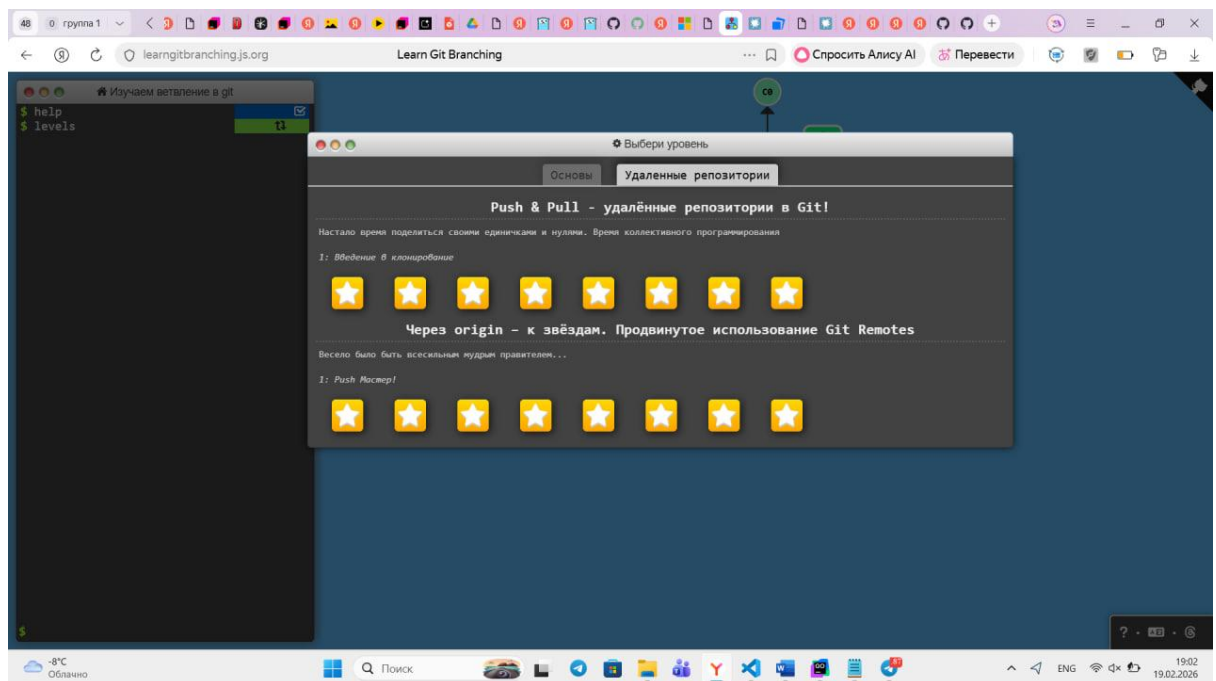
Работа с историей:

- Git cherry-pick
- Git rebase
- Git rebase -i

Работа с тэгами и описанием:

- Git tag
- Git describe

Удаленные репозитории:



- Git clone
- Git remote
- Git remote add
- Git fetch
- Git pull
- Git push
- Git push origin <branch>

- `Git push origin <local branch>:<remote branch>`
- `Git checkout -b <branch> origin/<branch>`
- `Git branch -u origin/<branch>`

Вывод

В ходе выполнения работы я разобралась с принципами работы Flexbox и CSS Grid. Многие вещи проще воспринимались именно через интерактивные задания.

Также я изучила базовые команды Git и поняла, как работает система контроля версий: создание веток, слияние, перемещение по истории и работа с удалёнными репозиториями. Теперь стало понятнее, как организуется работа над проектом и как происходит обмен изменениями между разработчиками.