

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа 1: CSS Flexbox, CSS Grid

Выполнил:

Едигарева Дарья

Группа К3439

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

Задача

Пройти три игры:

<https://flexboxfroggy.com/#ru>

<https://cssgridgarden.com/#ru>

https://learngitbranching.js.org/?locale=ru_RU (основы работы с git, достаточно выполнить первые 4 урока + все уроки по удалённым репозиториям)

Ход работы

1. Flexbox Froggy

Игра с 24 уровнями, каждый из которых вводит новое CSS-свойство flexbox или комбинацию.

Ключевые свойства, с которыми учишься работать в процессе:

justify-content — выравнивание вдоль основной оси. (Gist)

align-items — выравнивание всех элементов вдоль поперечной оси.

flex-direction — направление основной оси (row/column и их reverse-версии).

order — порядок отображения flex-элементов, переопределяет исходный документный порядок.

align-self — позволяет отдельному элементу отличаться от остальных по поперечной оси.

flex-wrap — позволяет переносить элементы на новую строку/столбец.

flex-flow — сокращение для flex-direction + flex-wrap.

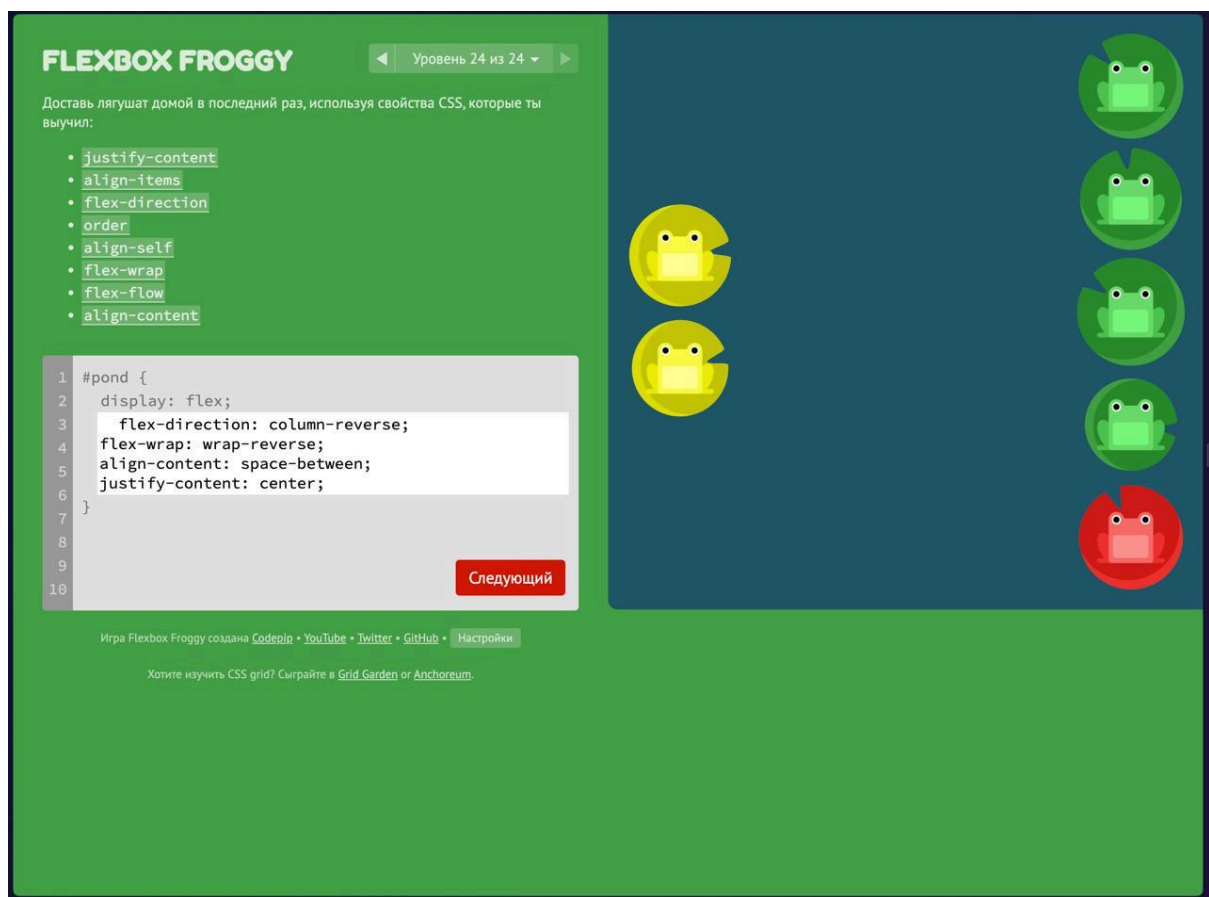
align-content — выравнивание нескольких строк/строковых линий

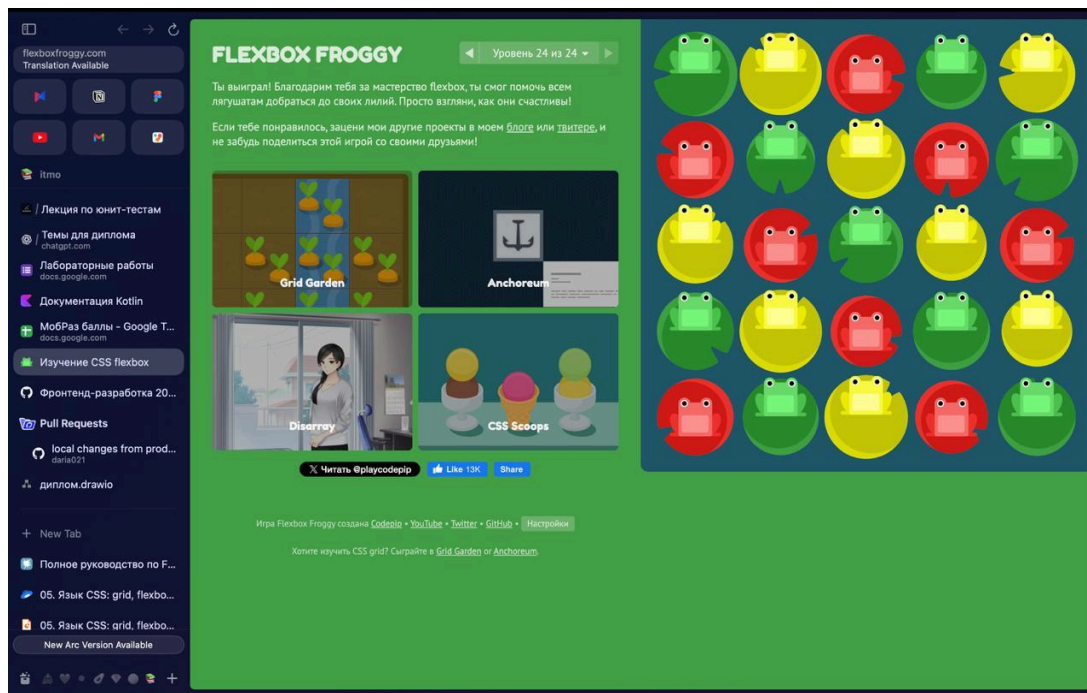
flex-контейнера в поперечном направлении.

Чему научилась

1. Как контейнер (родительский блок) управляет расположением своих дочерних flex-элементов.
2. Как изменение направления оси (flex-direction) полностью меняет смысл justify-content и align-items.

3. Чем отличается поведение одиночной строки (без wrap) от многострочного flex (с wrap + align-content).
4. Как с помощью order и align-self можно ломать «логический» порядок и визуально переставлять элементы — важно для адаптивной верстки.
5. Как комбинировать свойства (например, flex-flow: column wrap, flex-direction: column-reverse + wrap-reverse) и понимать, как «падают» линии элементов.
6. Почему flexbox удобен для одномерных макетов (одна строка или один столбец) и как это отличие от CSS Grid помогает проектировать простые адаптивные интерфейсы.





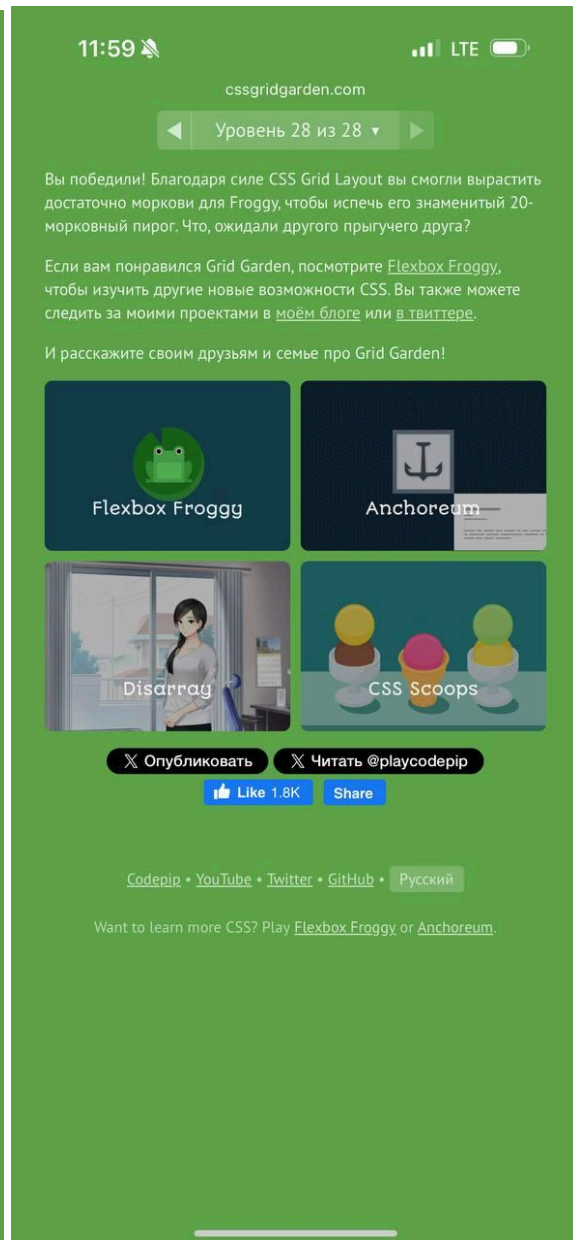
2. CSS Grid Garden

Игра с уровнями, целью которой является «полить морковки» или «убрать сорняки», используя CSS Grid-свойства.

Введены такие вещи, как `grid-template-rows`, `grid-template-columns`, `grid-column-start/end`, `grid-row-start/end`, `span`, и т.п.

Чему научилась

1. Поняла базовую двухмерную модель layout: строки и столбцы.
2. Как задавать шаблон сетки (`grid-template`), управлять размерами ячеек.
3. Как позиционировать элемент в сетке: начинать с `grid-column-start` и `—end`, `grid-row-start` и `—end` или задавать `grid-area`.
4. Как один элемент может занимать несколько строк или столбцов (используя `span`).
5. Как комбинировать Grid с Flexbox: знать, когда что удобнее использовать.



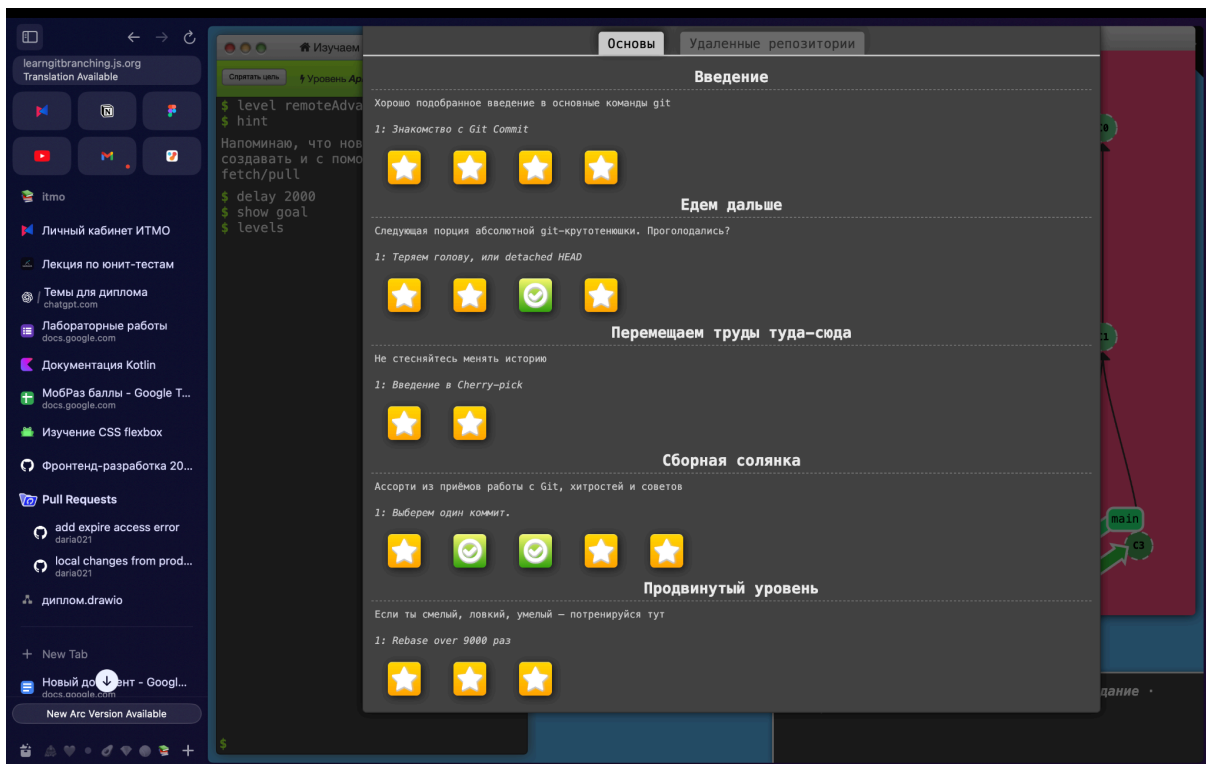
3. Learn Git Branching

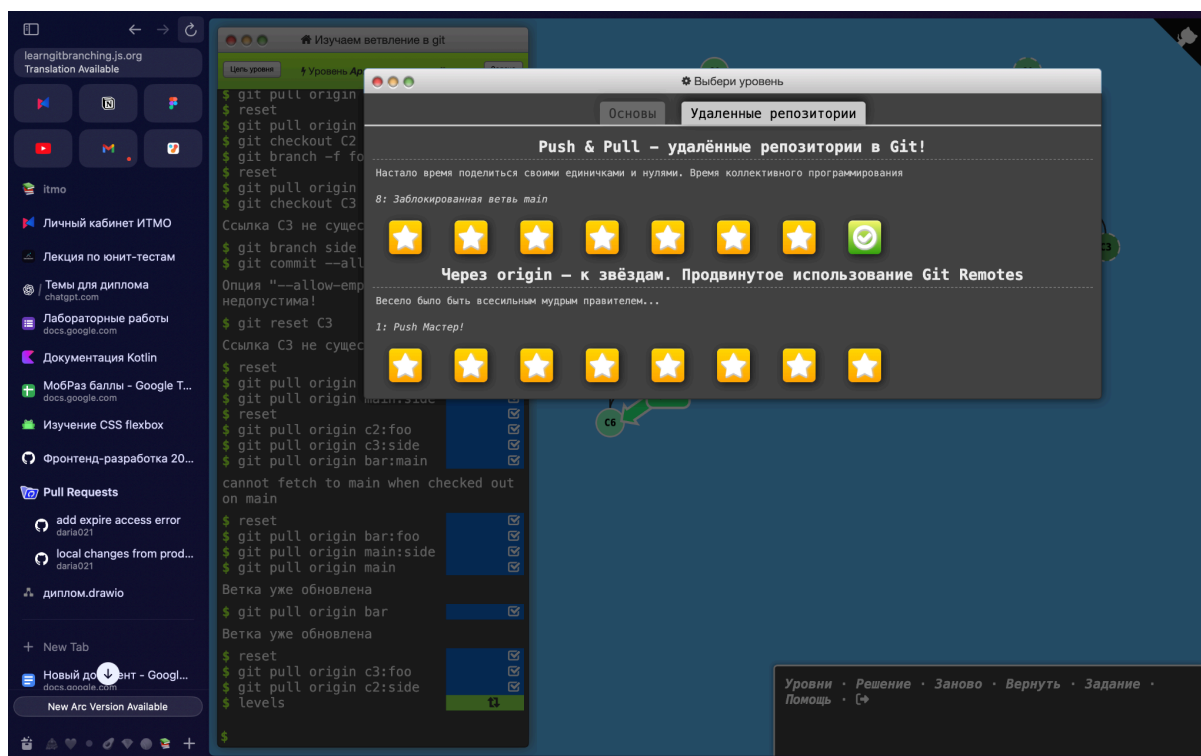
Интерактивная игра/симулятор, где ты визуально управляешь репозиторием Git: создаёшь коммиты, ветки, сливаешь, делаешь rebase, работаешь с удалёнными.

Чему научилась

1. Что такое коммит: снимок состояния файлов, фиксация изменений.
2. Как работает ветвление (branch): лёгкий указатель на коммит, не копия проекта.

3. Как работает слияние (merge): когда ветка встраивается в другую и история объединяется.
4. Что такое rebase: перенести изменения ветки на другой базовый коммит, переписать историю.
5. Работа с удалёнными ветками, fetch, pull, push — симуляция команд реального взаимодействия с remote.
6. Как визуально видеть и управлять ситуацией, где история «отстала», есть конфликты, есть пересечения веток.





Вывод

Flexbox Froggy позволила на практике понять механизмы работы Flexbox-модели: выравнивание вдоль основной и поперечной осей, управление направлением, переносом элементов, их порядком и индивидуальным позиционированием. Эти знания необходимы для построения адаптивных интерфейсов, где элементы должны гибко реагировать на ширину экрана и перестраиваться без «костылей» и лишних медиазапросов.

В CSS Grid Garden отрабатывалась двумерная модель расположения элементов. Grid дал понимание структуры макета как сетки строк и столбцов, научил управлять областями, пересечениями и сложными схемами размещения.

В Learn Git Branching через визуальную симуляцию, удалось освоить историю коммитов, принципы ветвления, слияние, перенос базовой ветки (rebase), работу с удалёнными репозиториями, fetch/pull/push-циклы, указатели origin, HEAD, fast-forward и не-fast-forward историй.