

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

Отчет

Практическая 1

Выполнил:  
Ле Хоанг Чыонг

Группа: K33392

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](https://learngitbranching.js.org/?locale=ru_RU) (основы работы с git, достаточно выполнить первые 4 урока)

## Ход работы

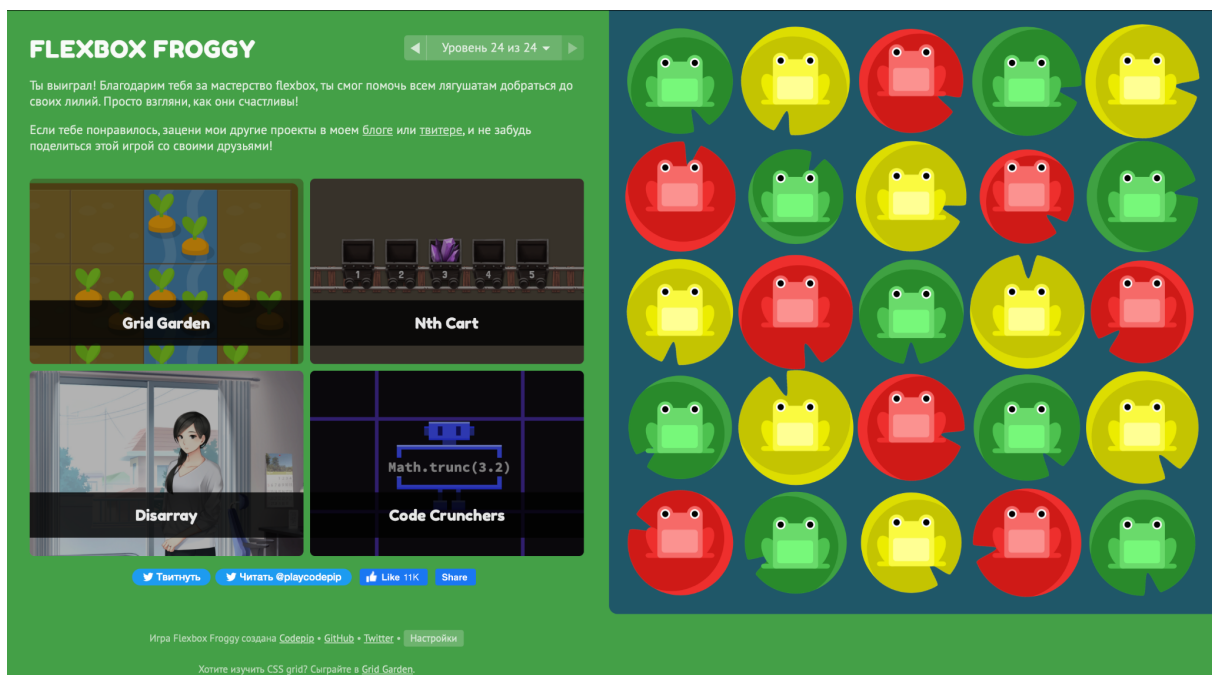
### 1. Flexbox froggy

В игре "FLEXBOX FROGGY" я практиковался со следующими командами :

1. **justify-content (выравнивание по горизонтали):** Это свойство используется для выравнивания элементов в контейнере по горизонтали. С помощью justify-content можно настроить расстояние между элементами или выровнять их по определенному значению, такому как flex-start, flex-end, center, space-between, space-around и space-evenly.
2. **align-items (выравнивание по вертикали):** Это свойство используется для выравнивания элементов в контейнере по вертикали. С помощью align-items можно настроить вертикальное положение элементов внутри контейнера, такое как flex-start, flex-end, center, baseline и stretch.
3. **flex-direction (направление гибкости):** Это свойство используется для определения направления элементов в контейнере. С помощью flex-direction вы можете задать, должны ли элементы быть расположены в горизонтальном направлении (row), вертикальном направлении (column), обратном горизонтальном направлении (row-reverse) или обратном вертикальном направлении (column-reverse).
4. **order (порядок):** Это свойство используется для определения порядка отображения дочерних элементов в контейнере. Значение order - это число, и элементы будут отсортированы в порядке возрастания значения.
5. **align-self (выравнивание самостоятельно):** Это свойство используется для настройки вертикального положения

конкретного элемента в контейнере. С помощью `align-self` вы можете переопределить значение `align-items` для конкретного элемента.

6. **flex-wrap (перенос гибкости)**: Это свойство используется для определения, должны ли дочерние элементы контейнера быть помещены в одну строку (`nowrap`), разрешить перенос элементов на следующую строку в случае недостатка места (`wrap`) или разрешить перенос элементов на следующую строку и столбец в случае недостатка места (`wrap-reverse`).
7. **flex-flow (поток гибкости)**: Это сокращенное свойство для одновременной установки `flex-direction` и `flex-wrap`. С помощью `flex-flow` вы можете задать как направление, так и тип компоновки элементов в контейнере.
8. **align-content (выравнивание контента)**: Это свойство используется для выравнивания строк в контейнере, когда они не достаточно высоки, чтобы заполнить контейнер. С помощью `align-content` вы можете настроить вертикальное положение строк, такое как `flex-start`, `flex-end`, `center`, `space-between`, `space-around` и `stretch`.

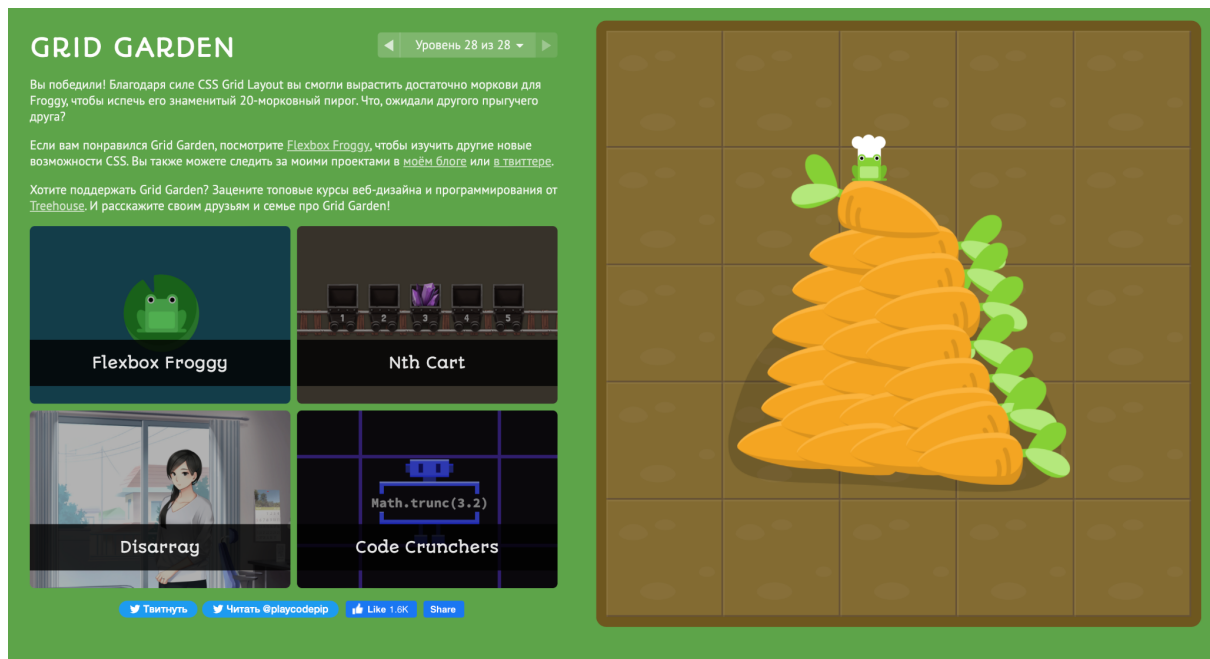


git describe bugFix commit "Finish the task"

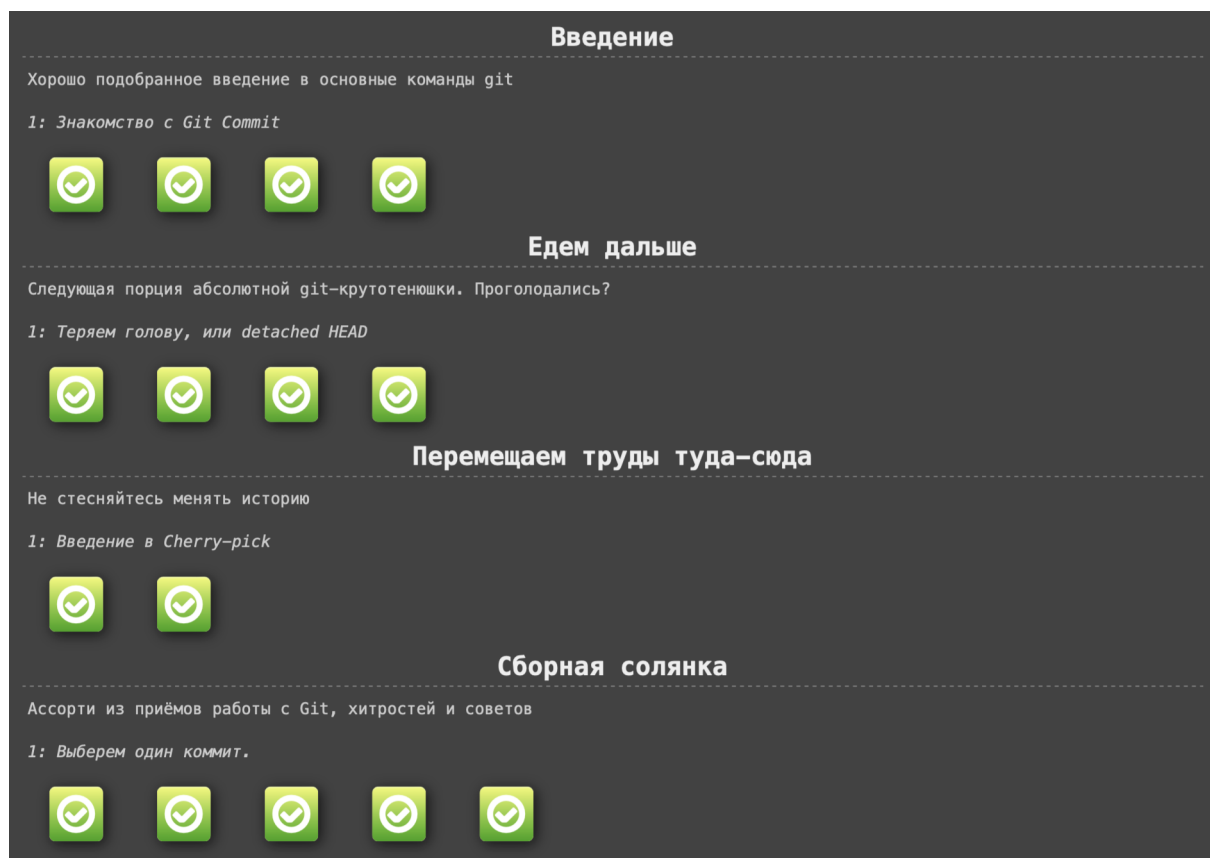
## 2. Grid garden

В игре "GRID GARDEN" я практиковался со следующими командами :

1. **grid-column (сетка-колонка)**: Эта команда используется для определения позиции элемента в горизонтальном направлении сетки. Используя grid-column, вы можете указать номера столбцов, в которых находится элемент, и настроить их размеры. Например, grid-column: 2 / 4 установит элемент от столбца 2 до столбца 4 в сетке.
2. **grid-row (сетка-строка)**: Эта команда используется для определения позиции элемента в вертикальном направлении сетки. Используя grid-row, вы можете указать номера строк, в которых находится элемент, и настроить их размеры. Например, grid-row: 1 / 3 установит элемент от строки 1 до строки 3 в сетке.
3. **grid-area (сетка-область)**: Эта команда позволяет определить область на сетке, указав позицию элемента как в горизонтальном, так и в вертикальном направлении. Используя grid-area, вы можете разместить элемент в конкретной области на сетке. Например, grid-area: 1 / 2 / 3 / 4 установит элемент от строки 1, столбца 2 до строки 3, столбца 4 на сетке.
4. **grid-template (сетка-шаблон)**: Эта команда используется для определения структуры и размеров сетки. Используя grid-template, вы можете указать количество столбцов, количество строк и их размеры в сетке. Например, grid-template: repeat(3, 1fr) / 1fr 2fr создаст сетку с 3 строками и 2 столбцами, где размер первого столбца будет 1 гибкой единицей (1fr), а второго столбца - 2 гибких единицы (2fr).



### 3. LearnGitBranching



## **Вывод**

Я узнал больше о командах css и git, считаю эти знания очень полезными для будущего.