

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Дисциплина:** Фронт-энд разработка

**Отчет**

**Домашняя работа**

**Выполнила:**

**Нугаева Камилла**

**J3212**

**Проверил:  
Добряков Д. И.**

**Санкт-Петербург**

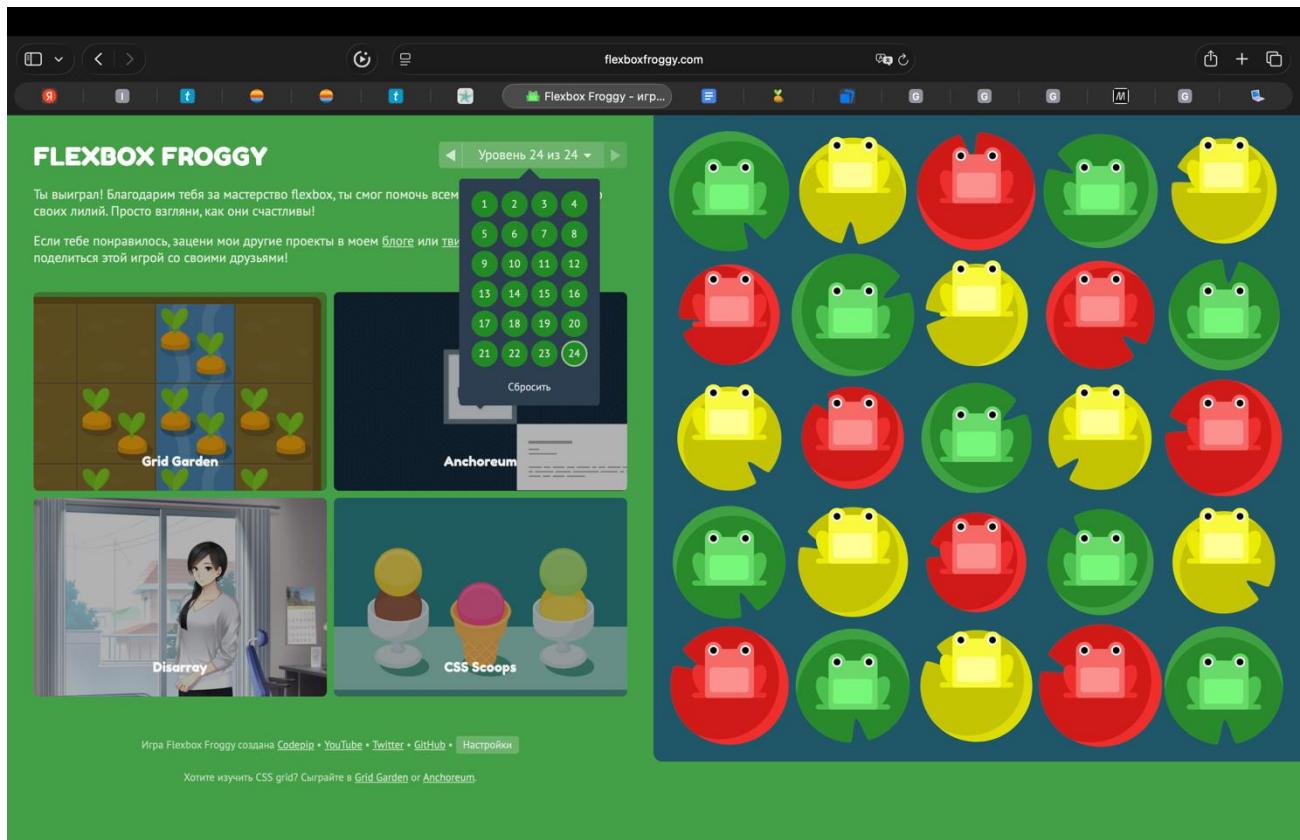
**2026 г.**

## 1. Flexbox Froggy

**Свойства CSS, которые были применены:**

1. **justify-content** - выравнивает все элементы вдоль главной оси. (flex-start, center, flex-end, space-between, space-around, space-evenly).
2. **align-items** - выравнивает все элементы вдоль пересекаемой оси. (flex-start, center, flex-end, baseline, stretch (default)).
3. **flex-direction** - задаёт направление основной оси. (row (default), row-reverse, column, column-reverse).
4. **order** - указывает порядок элемента. По умолчанию 0. Меньше число - раньше элемент, больше - позже.
5. **align-self** - выравнивает элемент вдоль пересекаемой оси, перекрывая значения свойства align-items. (flex-start, flex-end, center, baseline, stretch)
6. **flex-wrap** - указывает, нужно ли что бы элементы принудительно находились в одном ряду или автоматически переносились. (nowrap (default), wrap, wrap-reverse).
7. **flex-flow** - свойство для быстрой записи flex-direction и flex-wrap. (<flex-direction> <flex-wrap>)
8. **align-content** - выравнивает ряды контейнера внутри него (работает только, если элементы расположены больше чем в один ряд)(flex-start, center, flex-end, space-between, space-around, space-evenly, stretch (default)).

Очень крутая игра, реально помогает разобраться, над некоторыми уровнями пришлось посидеть))



## 2. Grid Garden

Свойства CSS, которые были применены:

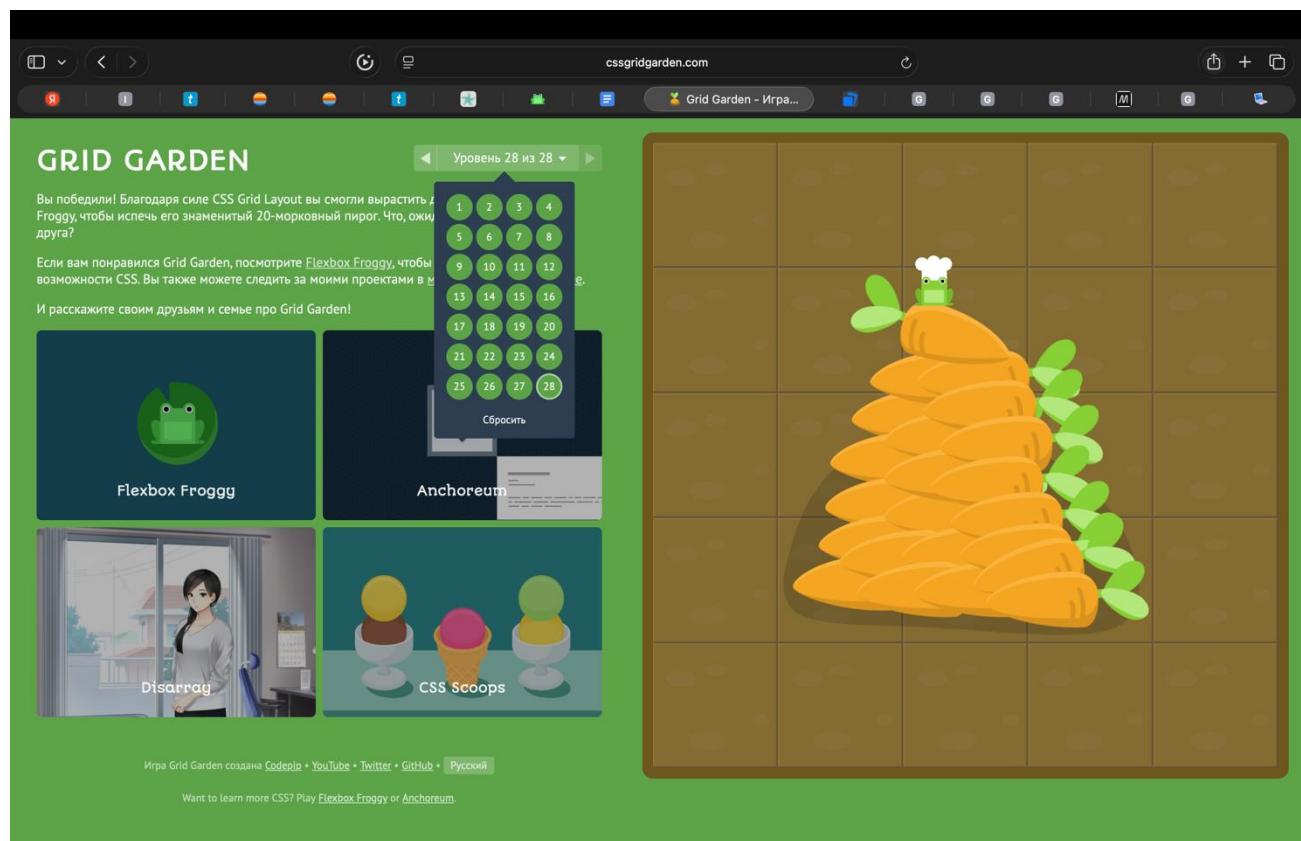
1. **grid-template-rows** - задаёт размеры строк grid-контейнера.
2. **grid-template-columns** - задаёт размеры колонок grid-контейнера.
3. **grid-template** - сокращённая запись для grid-template-rows и grid-template-columns.
4. **order** - меняет порядок отображения grid-элементов (как и во flex), но в grid используется реже.

По умолчанию 0: меньше - раньше, больше - позже.

5. **grid-area** - определяет позицию и размер grid-элемента внутри grid-сетки.
6. **grid-row** - сокращённо задаёт вертикальный диапазон элемента: `grid-row: start / end;`
7. **grid-column** - сокращённо задаёт горизонтальный диапазон элемента: `grid-column: start / end;`
8. **grid-row-start** - линия, с которой начинается элемент по строкам (верхняя граница).
9. **grid-column-start** - линия, с которой начинается элемент по колонкам (левая граница).
10. **grid-column-end** - линия, на которой заканчивается элемент по колонкам (правая граница).

### 11. Размеры:

- px** - фиксированный размер в пикселях.
- %** - процент от размера родителя (контейнера).
- em** - размер относительно текущего элемента.



### 3. Learn Git Branching

Разобрава несколько ключевых моментов:

- **Merge и Rebase:**  
`merge` = склеиваем две ветки и оставляем историю как есть  
`rebase` = переписываем историю так, будто я делала работу сразу на одной линии
- **fetch только скачивает изменения, pull скачивает и применяет, а push отправляет мои изменения на сервер**

Стало проще ориентироваться в Git: как устроены ветки, чем отличается merge от rebase, и как безопасно работать с удалёнными репозиториями через fetch/pull/push.

