

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет по
домашней работе “CSS Flexbox, CSS Grid”

Выполнил:

Пронина Мария

Группа:

К33392

Проверил:

Добряков Д. И.

Санкт-Петербург

2023 г.

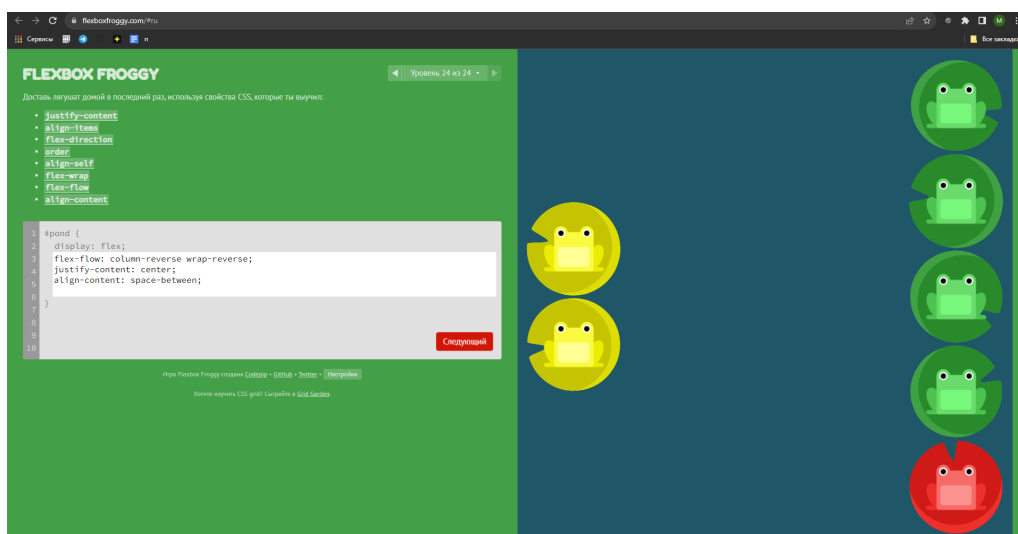
Задача

Пройти игры по CSS Flexbox, CSS Grid и работе с git.

Ход работы

1. Я прошла игру Flexbox Froggy, в чем очень помогла [статья](#). В процессе я разобралась с модулем Flexbox Layout, его свойствами и как их комбинировать, чтобы добиться желаемого позиционирования элементов:
 - a. *justify-content*: свойство определяет положение элементов относительно главной оси.
 - b. *align-items*: свойство, определяющее положение элементов относительно оси, перпендикулярной главной.
 - c. *flex-direction*: определяет, в каком направлении будут располагаться элементы.
 - d. *order*: (применяется к flex-элементам) позволяет менять порядок элементов, давая им “приоритеты” (по умолчанию у всех 0).
 - e. *align-self*: по сути то же, что и align-items, но применяется к конкретному элементу.
 - f. *flex-wrap*: позволяет элементам смещаться в следующий ряд.
 - g. *flex-flow*: свойство, объединяющее flex-direction и flex-wrap.
 - h. *align-content*: по сути то же, что justify-content, но не для элементов, а для рядов.

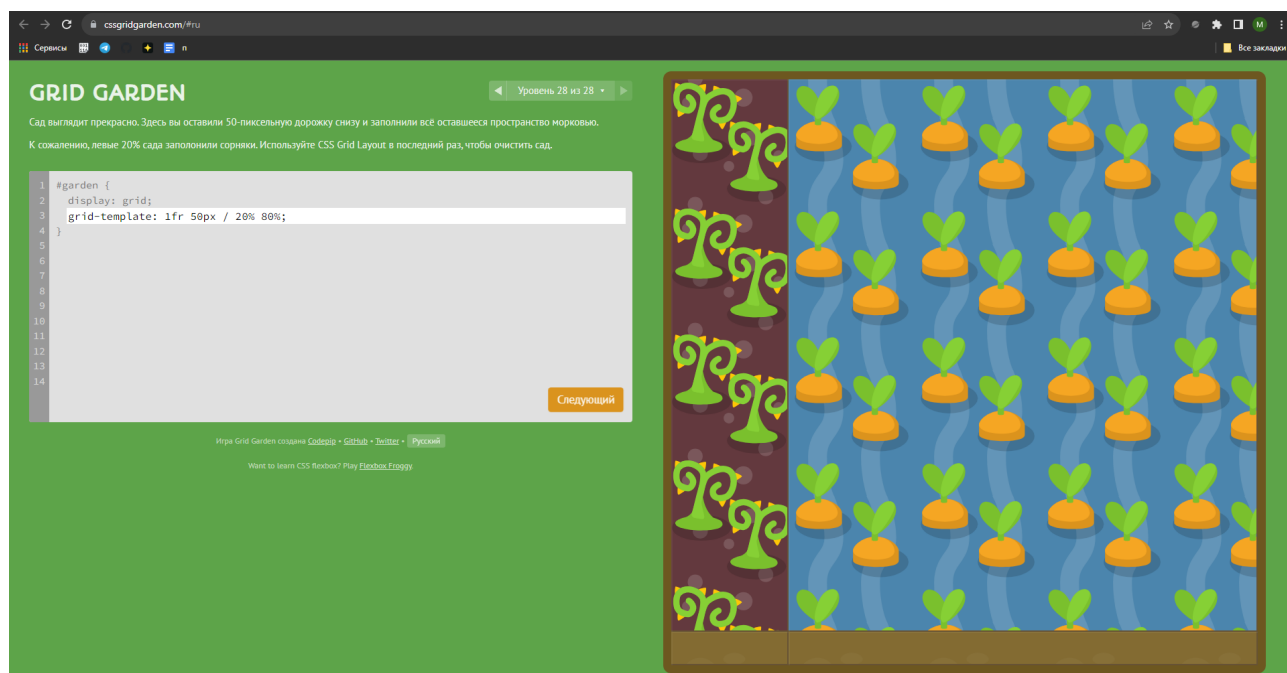
Для примера, решение последнего уровня игры:



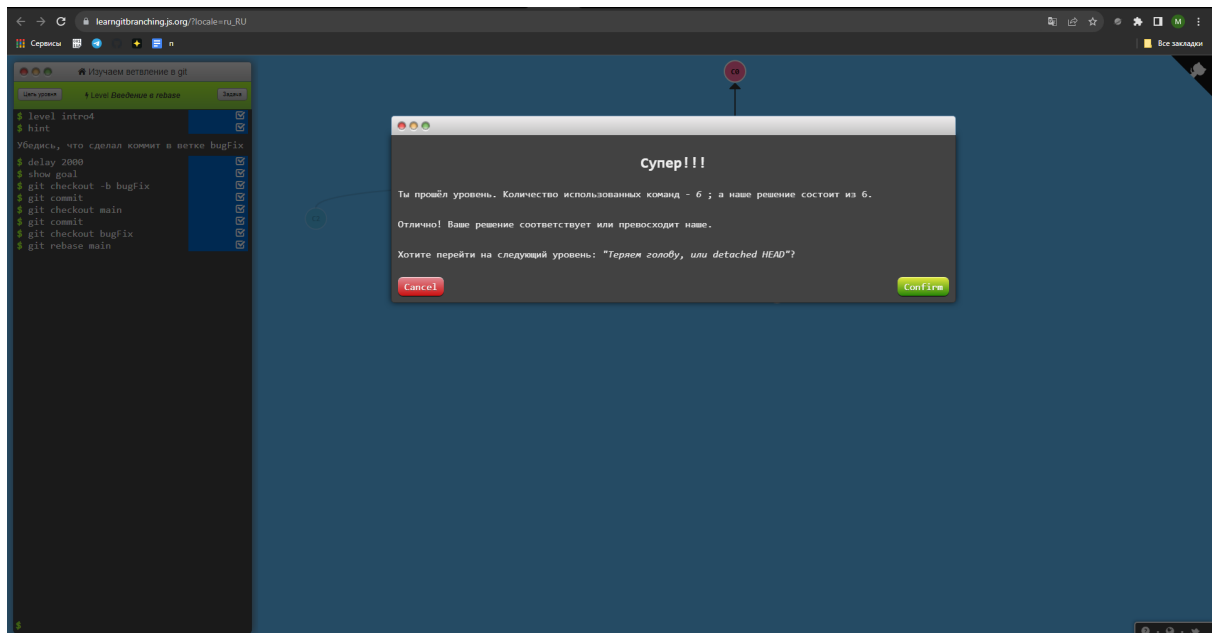
2. Я прошла игру Grid Garden, используя [эту статью](#). В процессе я разобралась с Grid моделью, принципами работы с ней и научилась использовать ее свойства для создания макета страницы:

- a. *grid-column*: это свойство дочернего элемента (объединение *grid-column-start* и *grid-column-end*), определяющее сколько и какие столбцы занимает элемент.
- b. *grid-row*: это свойство дочернего элемента (*grid-row-start* / *grid-row-end*), которое определяет сколько и какие строки занимает элемент.
- c. *grid-area*: свойство, которое полностью определяет позицию элемента (объединение *grid-row* и *grid-column*).
- d. *order*: (применяется к дочерним элементам) позволяет менять порядок элементов, давая им “приоритеты” (по умолчанию у всех 0).
- e. *grid-template-columns*: свойство, определяющее, сколько столбцов будет в сетке и какого они размера.
- f. *grid-template-rows*: свойство, определяющее, сколько строк будет в сетке и какого они размера.
- g. *grid-template*: объединение *grid-template-rows* и *grid-template-columns*.

Для примера, решение последнего уровня игры:



3. Затем я прошла игру по основам работы с git. В этой игре в целом все было просто и известно. Я повторила и наглядно увидела, как происходят git checkout, git merge и git rebase.



Вывод

Во время выполнения, я узнала и попробовала использовать макеты для позиционирования элементов в css. Оказалось, что это удобные способы создания структуры страницы, а выбор Grid, Flexbox или их комбинация зависит от конкретной цели и ожидаемого результат.