

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Домашняя работа №1

Выполнила:

Гергокова Амина

К4242

Проверил:

Добряков Д. И.

Санкт-Петербург

2024 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)

Собрать отчёт по шаблону. Экспортировать в pdf и открыть Pull Request в этот репозиторий.

Ход работы

В первой игре представлены 24 уровня, в которых сложность растёт с каждым последующим. Далее представлены прохождения всех уровней:

```
1 #pond {  
2   display: flex;  
3   justify-content: flex-end;  
4 }
```

Уровень 1

```
1 #pond {  
2   display: flex;  
3   justify-content: center;  
4 }
```

Уровень 2

```
1 #pond {  
2   display: flex;  
3   justify-content: space-around;  
4 }
```

Уровень 3

```
1 #pond {
2   display: flex;
3   justify-content: space-between;|
4 }
```

Уровень 4

```
1 #pond {
2   display: flex;
3   align-items: flex-end;|
4 }
```

Уровень 5

```
1 #pond {
2   display: flex;
3   justify-content: center;
4   align-items: center;|
5 }
```

Уровень 6

```
1 #pond {
2   display: flex;
3   justify-content: space-around;
4   align-items: flex-end;|
5 }
```

Уровень 7

```
1 #pond {
2   display: flex;
3   flex-direction: row-reverse;|
4 }
```

Уровень 8

```
1 #pond {
2   display: flex;
3   flex-direction: column;|
4 }
```

Уровень 9

```
1 #pond {
2   display: flex;
3   flex-direction: row-reverse;|
4   justify-content: flex-end;
5 }
```

Уровень 10

```
1 #pond {
2   display: flex;
3   flex-direction: column;|
4   justify-content: flex-end;
5 }
```

Уровень 11

```
1 #pond {
2   display: flex;
3   flex-direction: column-reverse;|
4   justify-content: space-between;
5 }
```

Уровень 12

```
1 #pond {
2   display: flex;
3   flex-direction: row-reverse;
4   justify-content: center;
5   align-items: flex-end;|
```

Уровень 13

```
1 #pond {
2   display: flex;
3 }
4
5 .yellow {
6   order: 2;|
7 }
```

Уровень 14

```

1 #pond {
2   display: flex;
3 }
4
5 .red {
6   order: -1;|
7 }

```

Уровень 15

```

1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   align-self: flex-end;|
8 }

```

Уровень 16

```

1 #pond {
2   display: flex;
3   align-items: flex-start;
4 }
5
6 .yellow {
7   align-self: flex-end;
8   order: 1;|
9 }

```

Уровень 17

```

1 #pond {
2   display: flex;
3   flex-wrap: wrap;|
4 }

```

Уровень 18

```
1 #pond {  
2   display: flex;  
3   flex-direction: column;  
4   flex-wrap: wrap;  
5 }
```

Уровень 19

```
1 #pond {  
2   display: flex;  
3   flex-flow: column wrap;  
4 }
```

Уровень 20

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   align-content: flex-start;  
5 }
```

Уровень 21

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   align-content: flex-end;  
5 }
```

Уровень 22

```
1 #pond {  
2   display: flex;  
3   flex-wrap: wrap;  
4   flex-direction: column-reverse;  
5   align-content: center;  
6 }
```

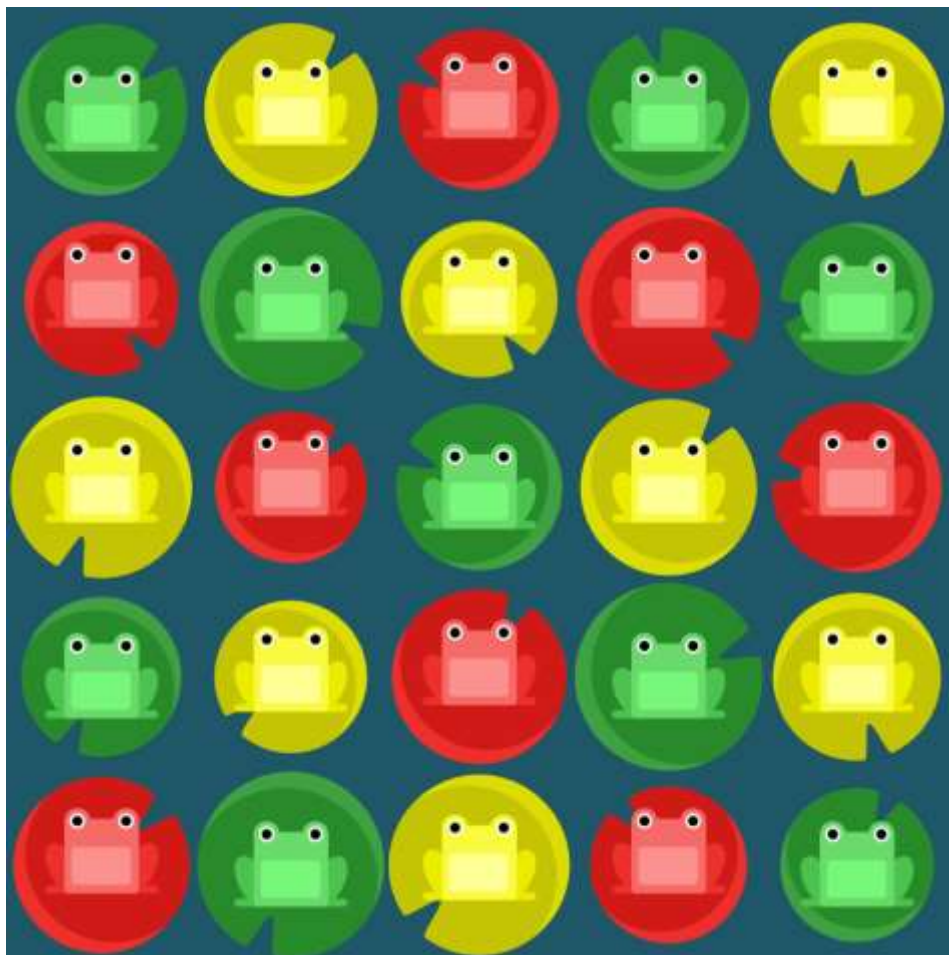
Уровень 23

```

1 #pond {
2   display: flex;
3   flex-flow: column-reverse wrap-reverse;
4   justify-content: center;
5   align-content: space-between;
6 }
7

```

Уровень 24



Уровень 25. Экран победы

Затем идет аналогичная игра, однако имеющая 28 уровней и направленная на изучение CSS Grid Layout. Далее представлены прохождения всех 28 уровней:

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 3;|
9 }

```

Уровень 1

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #poison {
8   grid-column-start: 5;|
9 }

```

Уровень 2

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 1;
9   grid-column-end: 4;|
10 }

```

Уровень 3


```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 5;
9   grid-column-end: 2;|
10 }

```

Уровень 4

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 1;
9   grid-column-end : -2;|
10 }

```

Уровень 5

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #poison {
8   grid-column-start: -3;
9 }

```

Уровень 6

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 2;
9   grid-column-end: span 2;|
10 }

```

Уровень 7

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: 1;
9   grid-column-end: span 5;
10 }

```

Уровень 8

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column-start: span 3;
9   grid-column-end: 6;
10 }

```

Уровень 9

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 4 / 6;|
9 }

```

Уровень 10

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 2 / span 3;|
9 }

```

Уровень 11

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-row-start: 3;|
9 }

```

Уровень 12

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-row: 3 / span 3;|
9 }

```

Уровень 13

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #poison {
8   grid-column: 2;
9   grid-row: 5;|
10 }

```

Уровень 14

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 2 / span 4;
9   grid-row: span 5;|
10 }

```

Уровень 15

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-area: 1 / 2 / 4 / 6;|
9 }

```

Уровень 16

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water-1 {
8   grid-area: 1 / 4 / 6 / 5;
9 }
10
11 #water-2 {
12   grid-area: 2 / 3 / 5 / 6;|
13 }

```

Уровень 17

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 .water {
8   order: 0;
9 }
10
11 #poison {
12   order: 5;|
13 }

```

Уровень 18

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 .water {
8   order: 0;
9 }
10
11 .poison {
12   order: -1;
13 }

```

Уровень 19

```

1 #garden {
2   display: grid;
3   grid-template-columns: 50% 20% 20% 20% 20%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 1;
9   grid-row: 1;
10 }

```

Уровень 20

```

1 #garden {
2   display: grid;
3   grid-template-columns: repeat(8, 12.5%);
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-column: 1;
9   grid-row: 1;
10 }

```

Уровень 21

```

1 #garden {
2   display: grid;
3   grid-template-columns: 100px 3em 40%;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }

```

Уровень 22

```

1 #garden {
2   display: grid;
3   grid-template-columns: 1fr 5fr;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }

```

Уровень 23

```

1 #garden {
2   display: grid;
3   grid-template-columns: 50px 1fr 1fr 1fr 50px;
4   grid-template-rows: 20% 20% 20% 20% 20%;
5 }
6
7 #water {
8   grid-area: 1 / 1 / 6 / 2;
9 }
10
11 #poison {
12   grid-area: 1 / 5 / 6 / 6;
13 }

```

Уровень 24

```

1 #garden {
2   display: grid;
3   grid-template-columns: 75px 3fr 2fr;
4   grid-template-rows: 100%;
5 }

```

Уровень 25

```

1 #garden {
2   display: grid;
3   grid-template-columns: 20% 20% 20% 20% 20%;
4   grid-template-rows: 50px 0% 0% 0%;
5 }
6
7 #water {
8   grid-column: 1 / 6;
9   grid-row: 5 / 6;
10 }

```

Уровень 26

```

1 #garden {
2   display: grid;
3   grid-template: 60% / 200px;
4 }
5
6 #water {
7   grid-column: 1;
8   grid-row: 1;
9 }

```

Уровень 27

```

1 #garden {
2   display: grid;
3   grid-template: 1fr 50px / 20% 80%;
4 }

```

Уровень 28



Уровень 29. Экран победы

Следующая игра является симулятором Git функций, содержащая множество различных уровней, которые позволят лучше освоиться с работой как с удаленными, так и локальными репозиториями. Далее представлено прохождение всех уровней:

```
$ level intro1
$ hint
Попробуй дважды выполнить команду 'git
commit' ;)
$ delay 2000
$ show goal
$ git commit
$ git commit
```

Уровень 1. Знакомство с Git Commit

```

$ level intro2
$ hint

Создай новую ветку при помощи "git branch
[name]" и перейди на неё при помощи "git
checkout [name]"

$ delay 2000
$ show goal
$ git checkout -b bugFix

```

Уровень 2. Ветвление в Git

```

$ level intro3
$ hint

Не забудь делать коммиты в правильном
порядке (сначала bugFix, потом main)

$ delay 2000
$ show goal
$ git checkout -b bugFix
$ git commit
$ git checkout main
$ git commit
$ git merge bugFix

```

Уровень 3. Ветвление в Git

```

$ level intro4
$ hint

Убедись, что сделал коммит в ветке bugFix

$ delay 2000
$ show goal
$ git checkout -b bugFix
$ git commit
$ git checkout main
$ git commit
$ git checkout bugFix
$ git rebase main

```

Уровень 4. Введение в rebase

```

$ level rampup1
$ hint

Ориентируйся по идентификаторам (hash)
коммитов.

$ delay 2000
$ show goal
$ git checkout C4

```

Уровень 5. Теряем голову, или detached HEAD

```

$ level rampup2
$ hint
Не забудь оператор ^^^
$ delay 2000
$ show goal
$ git checkout bugFix^

```

Уровень 6. Относительные ссылки (^)

```

$ level rampup3
$ hint
Понадобится использовать как минимум одну
прямую ссылку (хеш), чтобы пройти этот
уровень
$ delay 2000
$ show goal
$ git branch -f bugFix C3~2
$ reset
$ git checkout C1
$ git branch -f main C6
$ git branch -f bugFix HEAD^

```

Уровень 7. Относительные ссылки №2

```

$ level rampup4
$ hint
Обрати внимание, что revert и reset
принимают разные параметры.
$ delay 2000
$ show goal
$ git reset HEAD^
$ git checkout pushed
$ git revert HEAD

```

Уровень 8. Отмена изменений в Git

```

$ level move1
$ hint
git cherry-pick основывается на именах
коммитов!
$ delay 2000
$ show goal
$ git cherry-pick C3 C4 C7

```

Уровень 9. Введение в Cherry-pick

```
$ level move2
$ hint

Можно использовать либо ветки, либо
относительные ссылки (HEAD~), чтобы
указать цель для Rebase

$ delay 2000
$ show goal
$ git rebase -i HEAD~4
```

Уровень 10. Введение в интерактивный Rebase

```
$ level mixed1
$ hint

Не забывай, что интерактивный rebase и
cherry-pick - это твои друзья!

$ delay 2000
$ show goal
$ git rebase -i HEAD~3
$ git branch -f main HEAD
```

Уровень 11. Выберем один коммит.

```
$ level mixed2
$ hint

Первой командой должна быть git rebase
-i HEAD~2

$ delay 2000
$ show goal
$ git rebase -i HEAD~2
$ git rebase -i HEAD~2
$ reset
$ git rebase -i HEAD~2
$ git rebase -i HEAD^
$ git rebase -i HEAD~2
$ git branch -f main HEAD
```

Уровень 12. Жонглируем коммитами

```

$ level mixed3
$ hint
Не забудь переместить main на последние
изменения.

$ delay 2000
$ show goal
$ git checkout main
$ git cherry-pick C2
$ git branch -f main HEAD^
$ git cherry-pick C2 C3

```

Уровень 13. Жонглируем коммитами №2

```

$ level mixed4
$ hint
Можно сделать checkout напрямую на
коммит или же на тег

$ delay 2000
$ show goal
$ git tag v0 C1
$ git tag v1 C2
$ git checkout v1

```

Уровень 14. git tag

```

$ level mixed5
$ hint
Когда закончишь, просто сделай commit

$ delay 2000
$ show goal
$ git describe bugFix
v1_2_gC6
$ git describe main
v0_2_gC2
$ git commit

```

Уровень 15. Git describe

```
$ level advanced1
$ hint
Не забудь, что лучше всего сдвинуть
мастер в самом конце...

$ delay 2000
$ show goal
$ git cherry-pick C3 C4 C5 C6 C7
На этом уровне нельзя
использовать эту команду!

$ git rebase another
$ git rebase side
$ git rebase bugFix
$ git rebase -i bugFix~2
```

Уровень 16. Rebase over 9000 раз

```
$ level advanced2
$ hint
`git branch bugWork` на нужном коммите
поможет создать нужную ссылку.

$ delay 2000
$ show goal
$ git branch bugWork HEAD^^2^
```

Уровень 17. Здоровая семья, или несколько родителей

```
$ show goal
$ hint
Убедись, что у нас всё по порядку!
Сначала ветка `one`, потом `two` и
только потом `three`

$ delay 2000
$ show goal
$ git checkout one
$ git cherry-pick C4 C3 C2
$ git branch -f two main
$ git checkout two
$ git rebase -i three
$ git branch -f three main~3
```

Уровень 18. Спутанные ветки

```
$ level remote1
$ hint
Простой git clone!
$ delay 2000
$ show goal
$ git clone
! local branch "main" set to track
remote branch "o/main"
```

Уровень 19. Введение в клонирование

```
$ level remote2
$ hint
Уделяйте внимание очередности -- сперва
commit на main
$ delay 2000
$ show goal
$ git commit
$ git checkout o/main
$ git commit
! Внимание! Репозиторий в состоянии
detached HEAD, то есть не находится ни
на какой ветке!
```

Уровень 20. Удалённые ветки

```
$ level remote3
$ hint
Просто выполните git fetch!
$ delay 2000
$ show goal
$ git fetch
```

Уровень 21. Git fetch

```
$ level remote4
$ hint
Запустите команду git pull !
$ delay 2000
$ show goal
$ git pull
```

Уровень 22. Git pull

```

$ level remote5
$ hint
помните, Вы можете указать количество
фейковых коммитов

$ delay 2000
$ show goal
$ git clone
! local branch "main" set to track
remote branch "o/main"

$ git fakeTeamwork main 2
$ git commit
$ git pull

```

Уровень 23. Коллективная работа

```

$ level remote6
$ hint
Помните, что прежде чем push-ить вам
нужно клонировать репозиторий!

$ delay 2000
$ show goal
$ git commit
$ git commit
$ git push

```

Уровень 24. Git push

```

$ level remote7
$ hint
проверьте сортировку в визуализации
цели

$ delay 2000
$ show goal
$ git clone
! local branch "main" set to track
remote branch "o/main"

$ git fakeTeamwork main 1
$ git commit
$ git pull --rebase
$ git push

```

Уровень 25. Расхождение в истории


```
$ level remote8
$ hint

Создайте новую feature ветвь от main
перед тем, как откатить изменения в
main до состояния o/main.

$ delay 2000
$ show goal
$ git checkout -b feature
$ git branch -f main o/main
$ git push

! local branch "feature" set to track
remote branch "o/feature"
```

Уровень 26. Заблокированная ветвь main

```
$ level remoteAdvanced1
$ hint

Помни - ты всегда можешь отменить
команды с помощью undo или reset

$ delay 2000
$ show goal
$ git rebase side2
$ git rebase side1
$ git pull --rebase
!

side3 is not a remote tracking branch!
I don't know where to push

$ git branch -f main side3
$ git checkout main
$ git pull --rebase
$ git push
```

Уровень 27. Push Мастер!

```

$ level remoteAdvanced2
$ hint
Внимательно посмотрите на цель уровня!

$ delay 2000
$ show goal
$ git checkout main
$ git branch -f main side1
$ git pull
$ git push
$ git branch -f main side2
$ git pull
$ git push
$ git branch -f main side3
$ git pull
$ git push

```

Уровень 28. Слияние с удалённым репозиторием

```

$ level remoteAdvanced3
$ hint
Помни, есть два способа установить
слежку за удалённым репозиторием!

$ delay 2000
$ show goal
$ git checkout -b side o/main
! local branch "side" set to track
remote branch "o/main"

$ git commit
$ git pull --rebase
$ git push

```

Уровень 29. Слежка за удалённым репозиторием

```

$ level remoteAdvanced4
$ hint
Вы всегда можете ознакомиться с
последним слайдом, воспользовавшись
"objective".

$ delay 2000
$ show goal
$ git push origin main
$ git push origin foo

```

Уровень 30. Аргументы git push

```
$ level remoteAdvanced5
$ hint

Помните, Вы всегда можете признать своё поражение, набрав команду "show solution" (показать решение) :P

$ delay 2000
$ show goal
$ git push origin foo:main
$ git push origin main^:foo
```

Уровень 31. Аргументы для push -- расширенная версия!

```
$ level remoteAdvanced6
$ hint

Обратите внимание на то, как номера коммитов могут меняться! Вы можете прочесть слайды вновь, воспользовавшись командой "help level"

$ delay 2000
$ show goal
$ git fetch origin foo:main
$ git fetch origin main^:foo
$ git checkout foo
$ git merge main
```

Уровень 32. Аргументы для fetch

```
$ level remoteAdvanced7
$ hint

Команда branch недоступна на этом упражнении, пользуйтесь командой fetch!

$ delay 2000
$ show goal
$ git push :foo

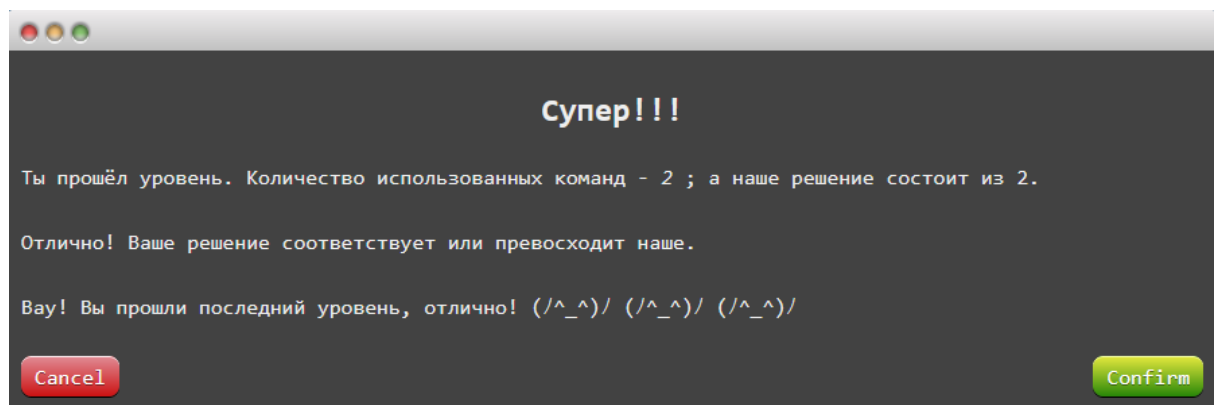
:foo is not a remote in your repository! try adding origin to that argument

$ git push origin :foo
$ git fetch origin :bar
```

Уровень 33. Пустой источник

```
$ level remoteAdvanced8
$ hint
Напоминаю, что новые ветки можно
создавать и с помощью команд fetch/pull
$ delay 2000
$ show goal
$ git pull origin bar:foo
$ git pull origin side:main
cannot fetch to main when checked out
on main
$ git pull origin main:side
```

Уровень 34. Аргументы для pull



Уровень 35. Экран победы

Вывод

Первая игра была направлена на изучение различных видов выравнивания во flex-контейнерах CSS. На уровнях игры были рассмотрены такие CSS свойства, как: justify-content, align-items, flex-direction, order, align-self, flex-wrap, flex-flow, align-content. Игра была успешно пройдена, и все лягушки благополучно добрались домой.

Вторая игра устроена аналогичным образом, однако в ней рассматривалось взаимодействие с CSS Grid Layout. Было рассмотрено создание столбцов, колонок, а также задание им ширины и высоты, используя как относительные, так и абсолютные значения. В результате сорняки были побеждены, а морковь выращена.

Наконец, третья игра дает нам довольно полное знакомство с Git. Она показывает различные команды, возможности их использования, а также раскрывает назначение. В ней содержится описание по работе как с локальным репозиторием, так и с удаленным.

Подводя итог, можно отметить, что прохождение всех 3 игр даёт множество полезных знаний как в CSS вёрстке, так и в работе с Git.