



# DEVELOP 2019



# Только CSS без JS? А так можно?

**CSS** трюки, **Кодо-генерация**, Сила **Google Chrome**,  
Ловкость **Puppeteer**





#DevFest

# Александр Вишняков

Full Stack Web Engineer

- Разработчик **ПО** с **2006** года, как бекенд, так и фронтенд.
- Влюблен в **JS**, но им не ограничиваюсь
- Использовал стек **Microsoft** в разработке, затем перешел на стек **OpenSource** языков и стек **Linux**.
- Увлекаюсь **ML** и не только, развиваюсь в разных направлениях.



#DevFest

# Только CSS без JS? А так можно?

- Одна из CSS задач на Yandex Contest 2019 Frontend





#DevFest

# Только CSS без JS? А так можно?

- Одна из CSS задач на Yandex Contest 2019 Frontend
- Технологии, для решения задачи



#DevFest

# Только CSS без JS?

## А так можно?

- Одна из CSS задач на Yandex Contest 2019 Frontend
- Технологии, для решения задачи
- Технологии, для проверки задачи





#DevFest

# Только CSS без JS?

## А так можно?

- Одна из CSS задач на Yandex Contest 2019 Frontend
- Технологии, для решения задачи
- Технологии, для проверки задачи
- Пошаговый разбор задачи



DEVFEST

2019



#DevFest

# Только CSS без JS?

## А так можно?

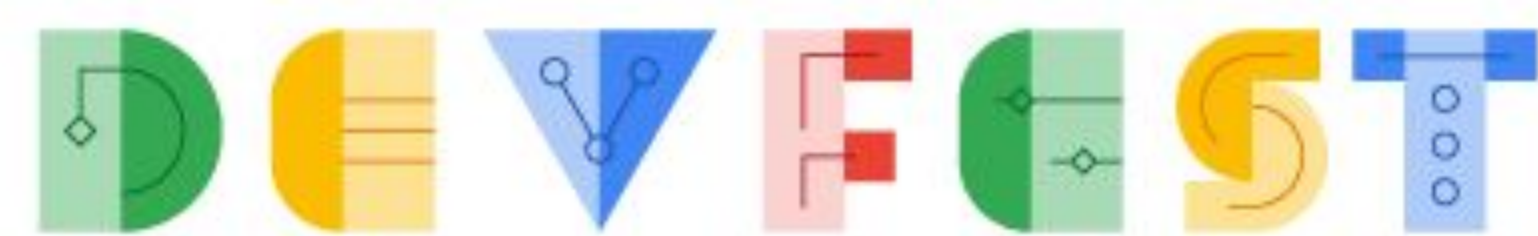
- Одна из CSS задач на Yandex Contest 2019 Frontend
- Технологии, для решения задачи
- Технологии, для проверки задачи
- Пошаговый разбор задачи
- Подведение итогов



**ОДНА ИЗ CSS ЗАДАЧ НА  
YANDEX CONTEST 2019 FRONTEND**

## D. Ход конём (40 баллов) Yandex Contest 2019

**Геннадий** - интеллектual. Он любит знакомиться с интересными людьми. Но будучи человеком осмотpительным и недоверчивым, делает он это только в интернете. Недавно Геннадий обнаружил, что сопоставимых по **IQ** собеседников можно отыскать на шахматном форуме, но вот беда - в шахматы играть Геннадий не умеет, а все обучаторы основаны на **javascript-e**, который Геннадий **осмотpительно отключает**, чтобы избежать вероятности **подцепить вирус**.



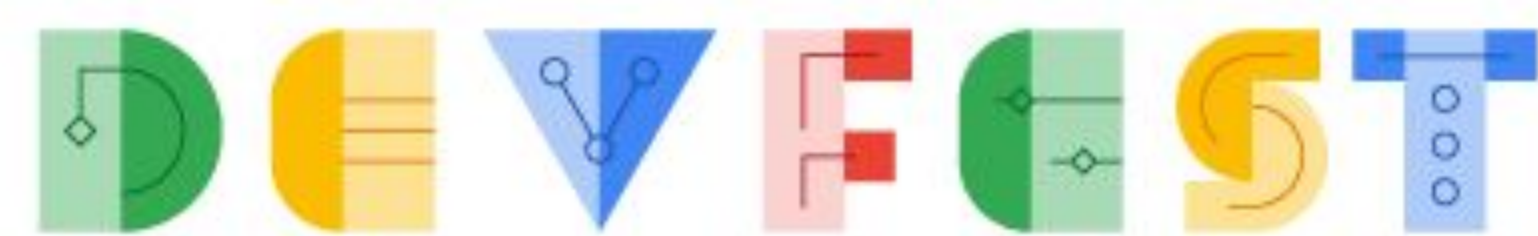
2019





## D. Ход конём (40 баллов) Yandex Contest 2019

Чтобы помочь Геннадию - предлагаем сделать **обучатор для игры в шахматы без javascript**, который будет показывать, как ходит конь. Обучатор должен выглядеть как шахматная доска. Кликаешь по клетке - тебе показывают, куда с этой клетки может пойти конь.



2019



# D. Ход конём (40 баллов) Yandex Contest 2019

Формат ввода

**html-документ**, при загрузке которого рисуется шахматная доска



2019





# D. Ход конём (40 баллов) Yandex Contest 2019

## Формат вывода

Задание будет протестировано в реальном браузере (**Chrome 77**).

В браузере будет загружен ваш **html-документ**.

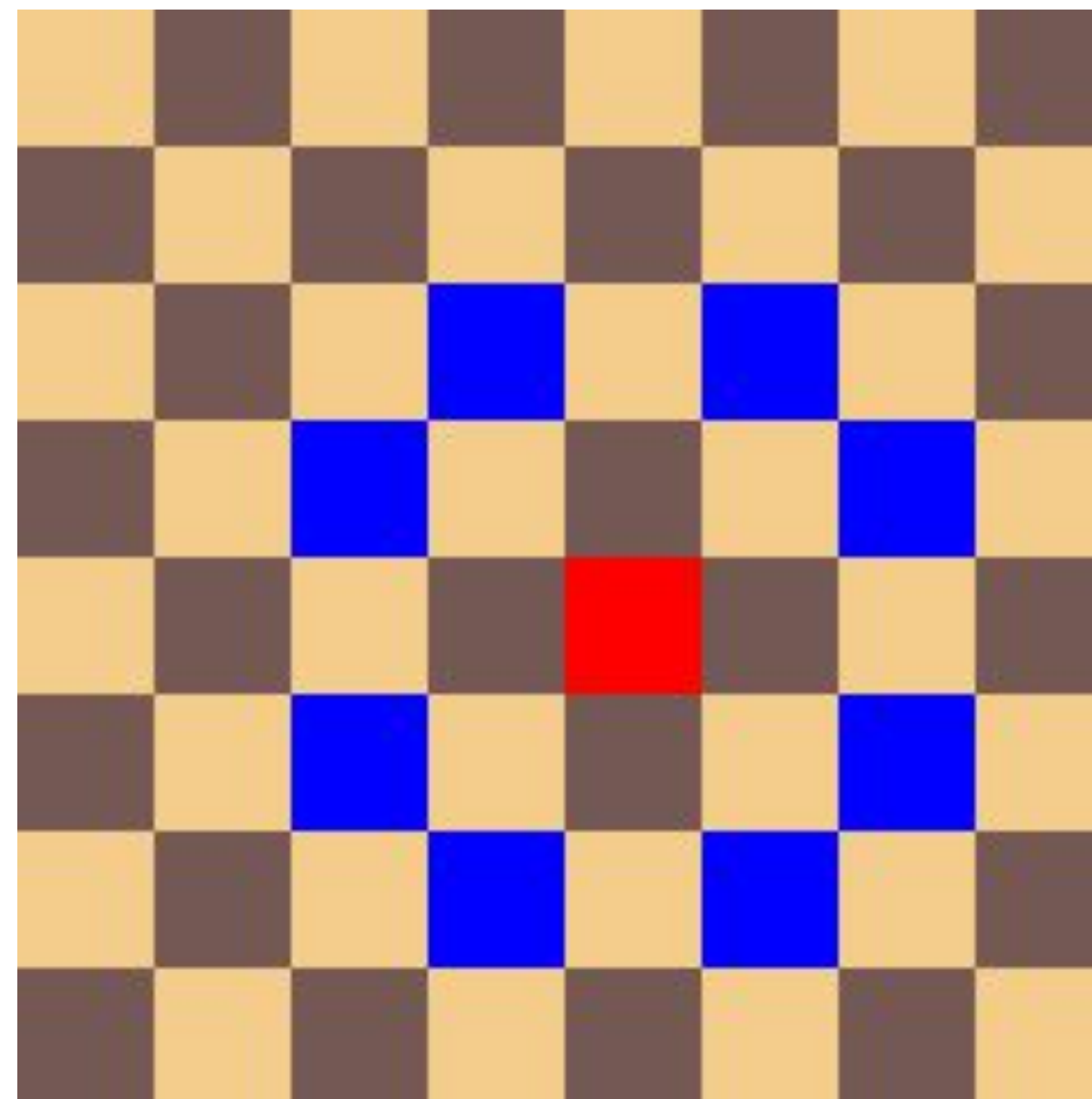
**Робот кликает в различные ячейки шахматного поля** и снимает скриншоты после кликов. **Скриншоты должны соответствовать эталонным**



2019



# D. Ход конём (40 баллов) Yandex Contest 2019



2019

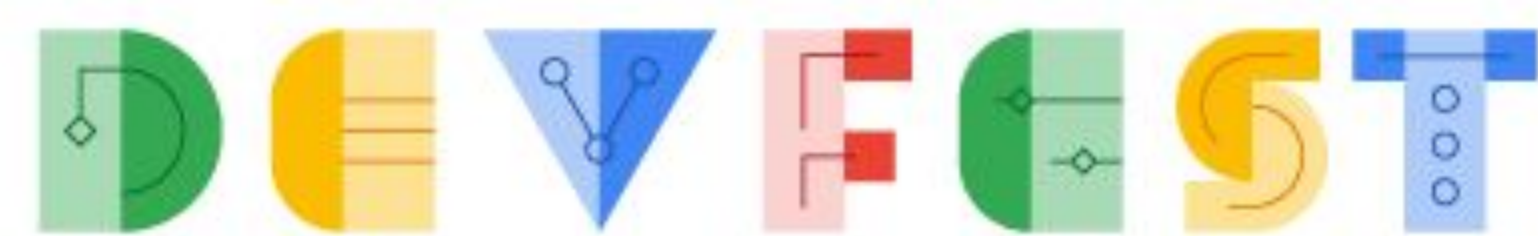




# D. Ход конём (40 баллов) Yandex Contest 2019

## Примечания

- Реализация на **CSS** и **HTML**.  
**Javascript использовать нельзя.**
- Вся верстка должна быть квадратной, без теней, градиентов, скруглений и т.п.
- Ширина и высота ячейки - **30** пикселей

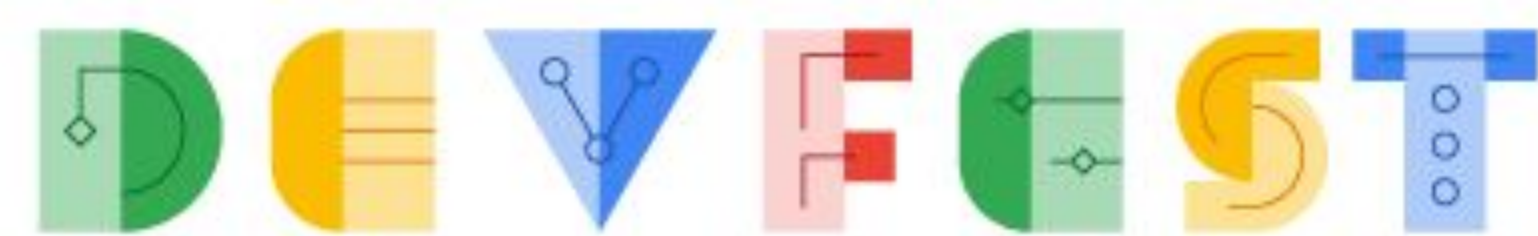


2019



## D. Ход конём (40 баллов) Yandex Contest 2019

- Шахматное поле находится на странице слева сверху, без отступов
- Цвет выделенной ячейки **#ff0000**
- Цвет ячейки, на которую может ходить фигура **#0000ff**
- Цвет светлой ячейки **#f4cd8d**
- Цвет темной ячейки **#745853**



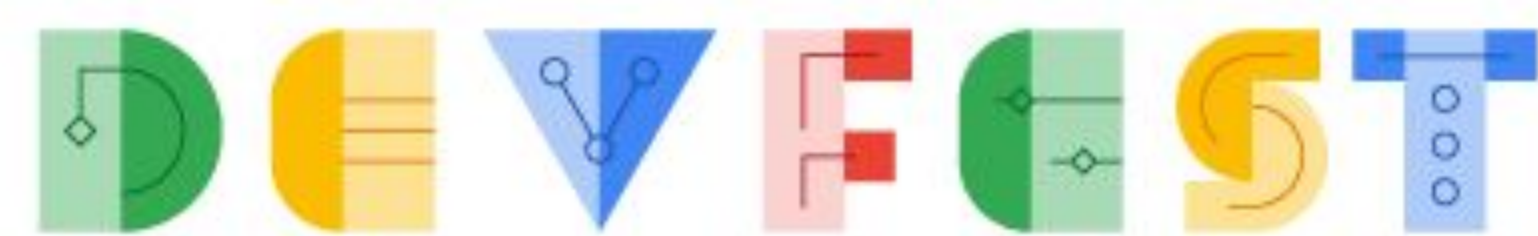
2019





## D. Ход конём (40 баллов) Yandex Contest 2019

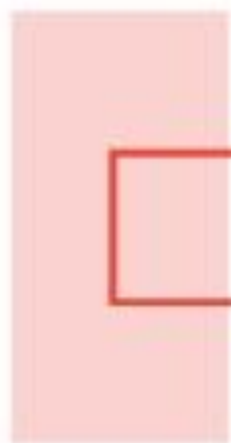
- Левая верхняя ячейка светлая
- Изначально ни одна ячейка не выделена
- Выделение происходит по клику в конкретную ячейку и сохраняется до следующего клика



2019



# ТЕХНОЛОГИИ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ





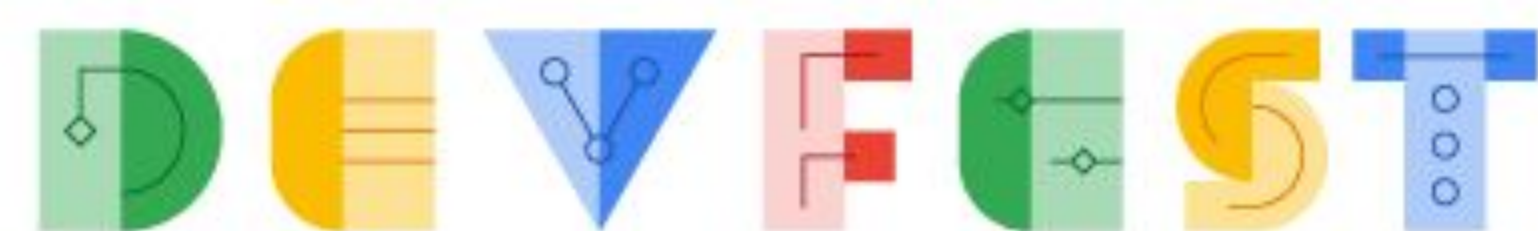
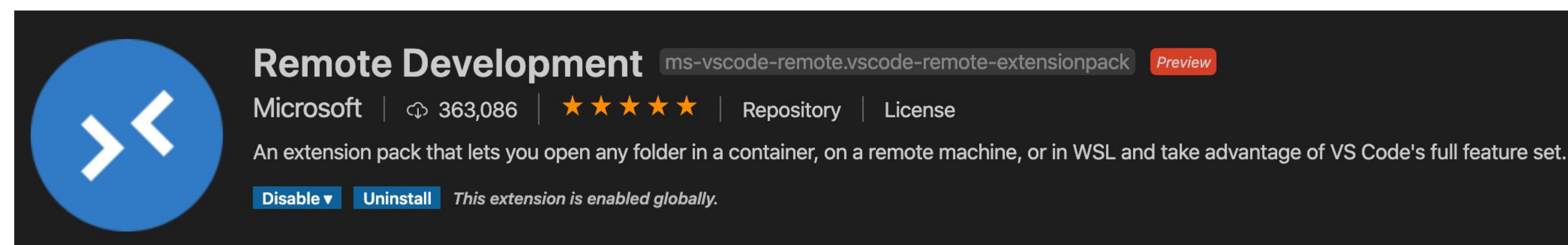
# ТЕХНОЛОГИИ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ

- VSCode



# ТЕХНОЛОГИИ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ

- VSCode
- Remote-Development



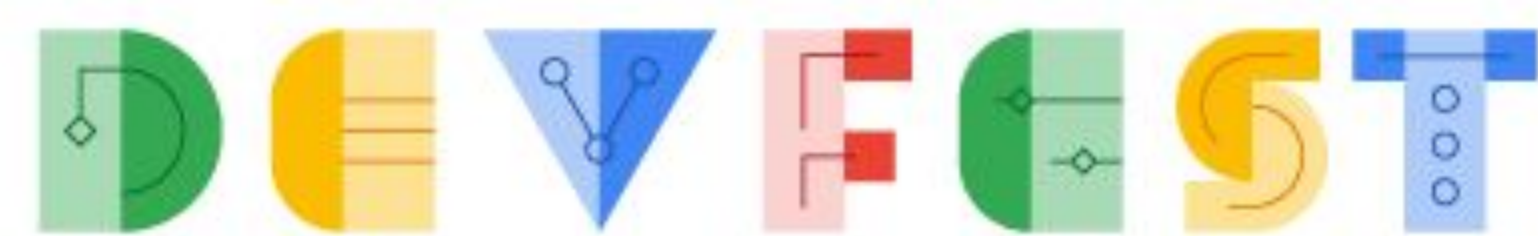
2019





# ТЕХНОЛОГИИ, ДЛЯ РЕШЕНИЯ ЗАДАЧИ

- **VSCode**
- **Remote-Development**
- **Docker Desktop**



2019



# ТЕХНОЛОГИИ, ДЛЯ ПРОВЕРКИ ЗАДАЧИ

# ТЕХНОЛОГИИ, ДЛЯ ПРОВЕРКИ ЗАДАЧИ

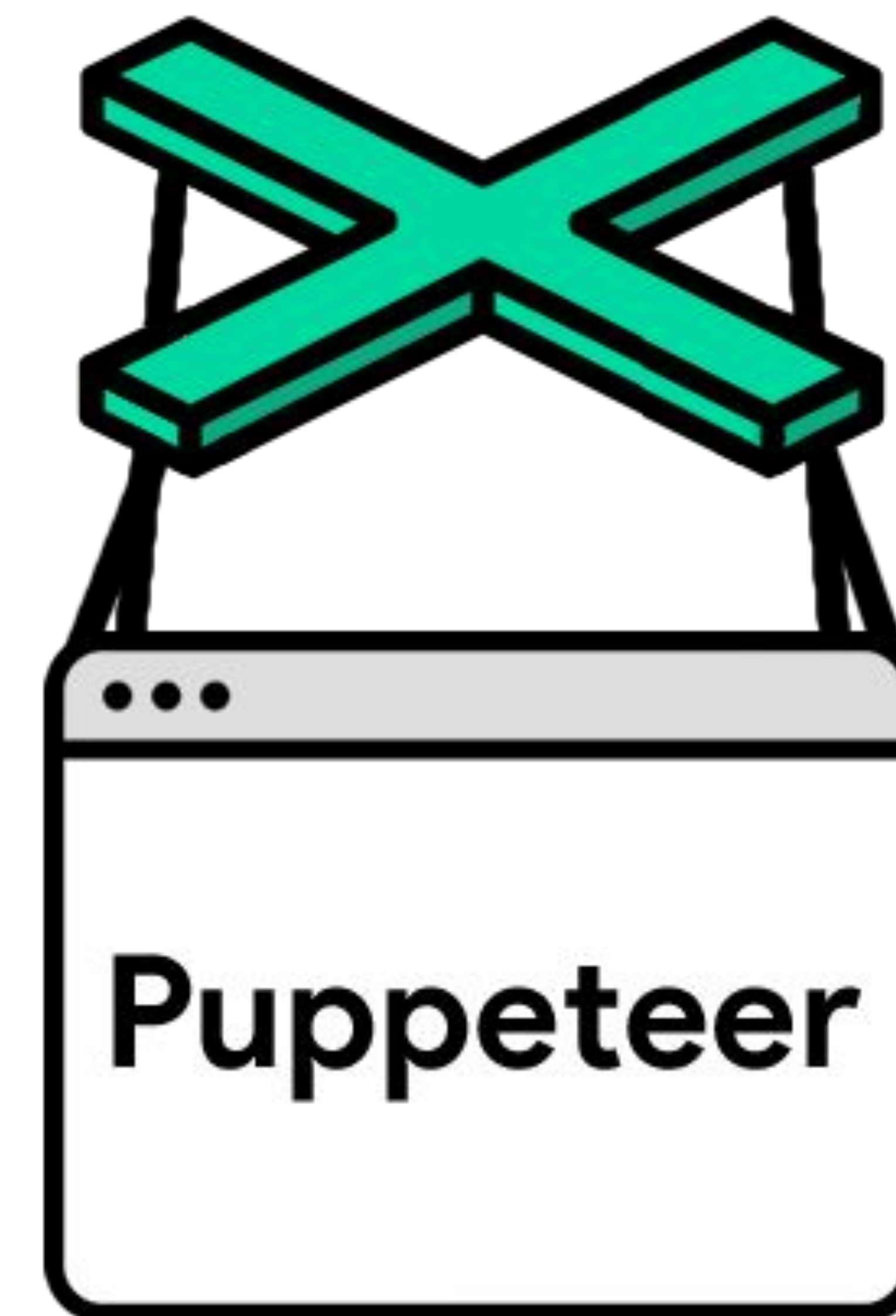
- Google Chrome





# ТЕХНОЛОГИИ, ДЛЯ ПРОВЕРКИ ЗАДАЧИ

- Google Chrome
- Puppeteer



## A trail of black footprints starts at the bottom left and leads diagonally upwards towards the top right. The footprints are of varying sizes and orientations, suggesting a path. The trail ends near the text 'ПОШ' on the right side of the image.

# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Реализация на **CSS** и **HTML**.

**Javascript использовать нельзя.**



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

CSS !== JS



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Вся верстка должна быть **квадратной, без теней, градиентов, скруглений** и т.п.
- Ширина и высота ячейки - **30** пикселей



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
:root {  
  --cell-width: 30px;  
}
```



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Шахматное поле находится на странице слева сверху, без отступов



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
body {  
  margin: 0;  
  padding: 0;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Цвет выделенной ячейки **#ff0000**




2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
:root {  
  --cell-width: 30px;  
  --cell-selected-bg: #ff0000;   
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Цвет ячейки, на которую может ходить фигура **#0000ff**

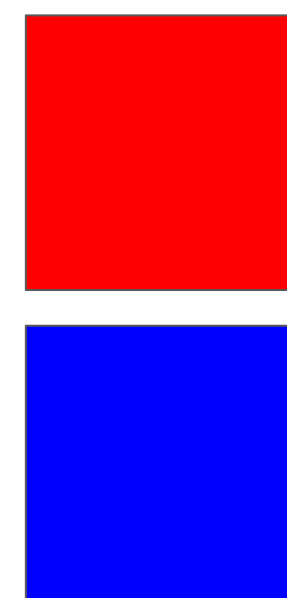


2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
:root {  
  --cell-width: 30px;  
  --cell-selected-bg: #ff0000;  
  --cell-marker-bg: #0000ff;  
}
```



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Цвет светлой ячейки **#f4cd8d** 

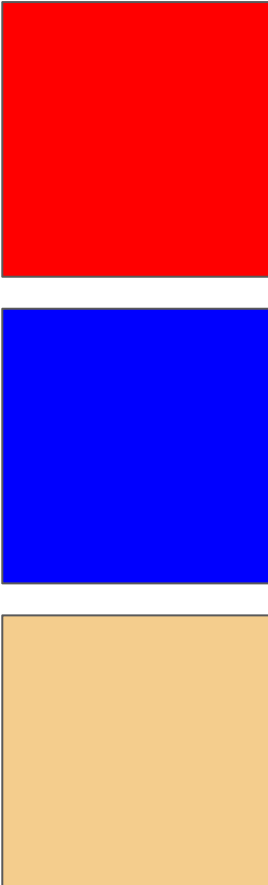


2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
:root {  
  --cell-width: 30px;  
  --cell-selected-bg: #ff0000;  
  --cell-marker-bg: #0000ff;  
  --cell-light-bg: #f4cd8d;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Цвет темной ячейки **#745853** 



2019



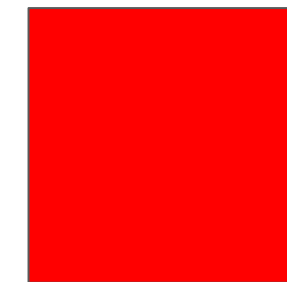


# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**:root** {

--cell-width: 30px;

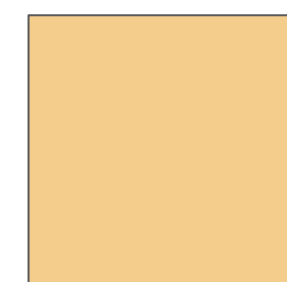
--cell-selected-bg: #ff0000;



--cell-marker-bg: #0000ff;



--cell-light-bg: #f4cd8d;



--cell-dark-bg: #745853;



}

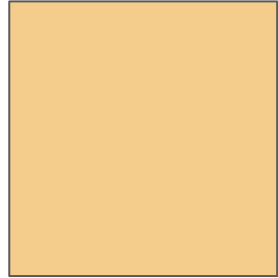


2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Левая верхняя ячейка светлая 

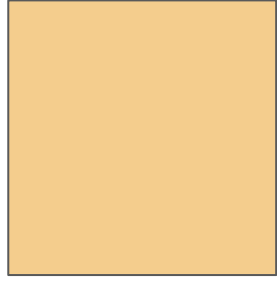


2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Левая верхняя ячейка светлая 
- Изначально ни одна ячейка не выделена



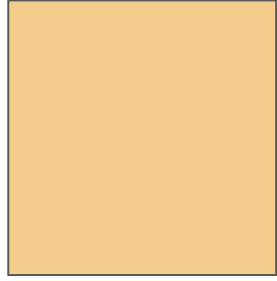
2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## Анализ условий:

- Левая верхняя ячейка светлая 
- Изначально ни одна ячейка не выделена
- Выделение происходит по клику в конкретную ячейку и сохраняется до следующего клика



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
:root {  
  ...  
  --cell-count: 8;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
#chess {  
  width: calc(var(--cell-width) * var(--cell-count));  
  height: calc(var(--cell-width) * var(--cell-count));  
  margin: 0;  
  border: 0;  
}
```



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
#chess button {  
  float: left;  
  border: none;  
  outline: none;  
  width: var(--cell-width);  
  height: var(--cell-width);  
}
```



2019

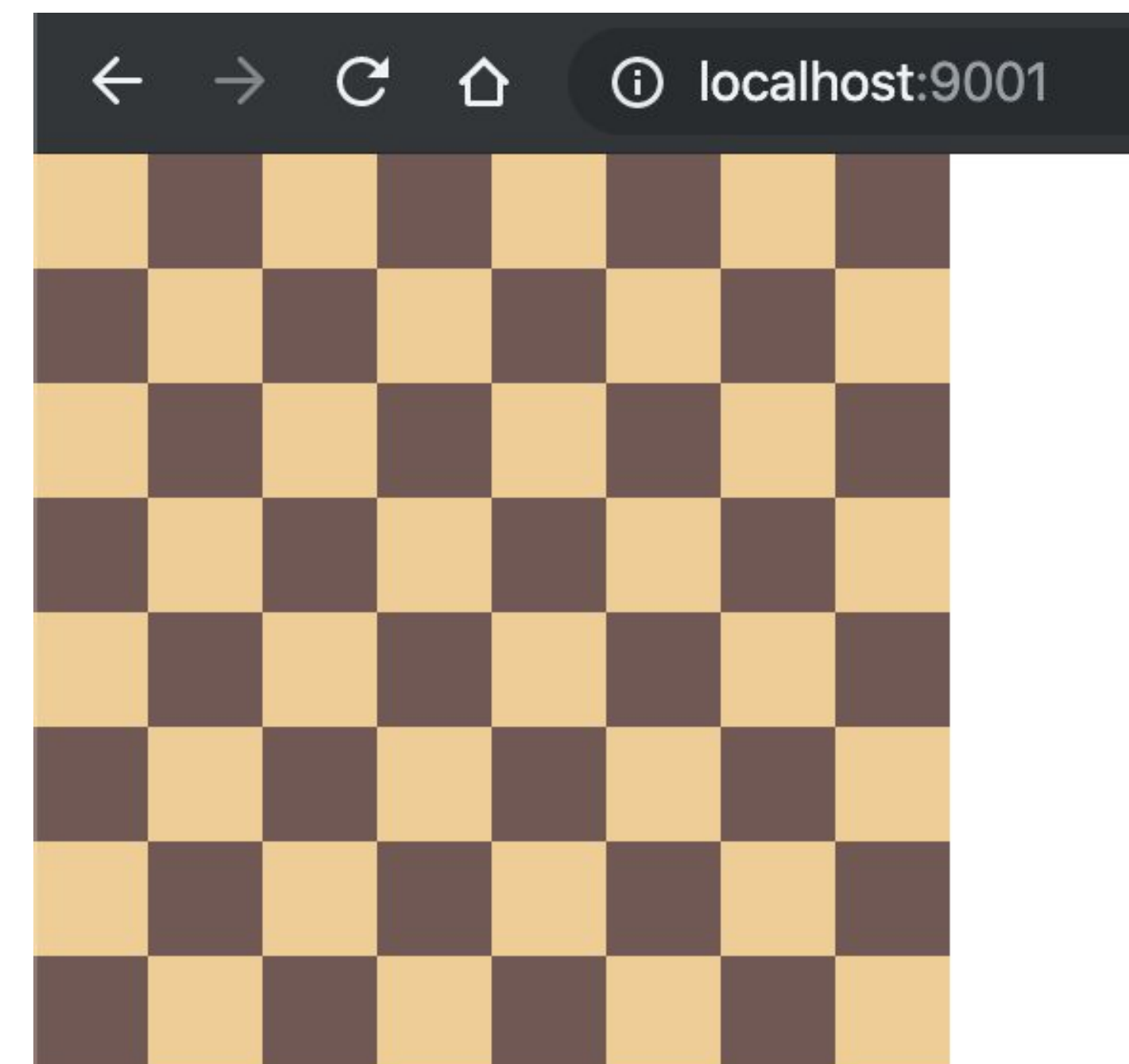


# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
#chess button:nth-child(-2n+8):not(:focus),  
#chess button:nth-child(8) ~ button:nth-child(-2n+15):not(:focus),
```

.....

```
#chess button:nth-child(56) ~ button:nth-child(-2n+63):not(:focus) {  
  background-color: var(--cell-dark-bg);  
  color: white;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**<div id="chess">**

**<button index=1></button>**

....

**<button index=64></button>**

**<div class="marker m1"></div>**

....

**<div class="marker m8"></div>**

**64 элемента <button>**

**8 маркеров**

**</div>**

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
<div id="chess">
```

```
  <button index=1></button>
```

....

```
  <button index=64></button>
```

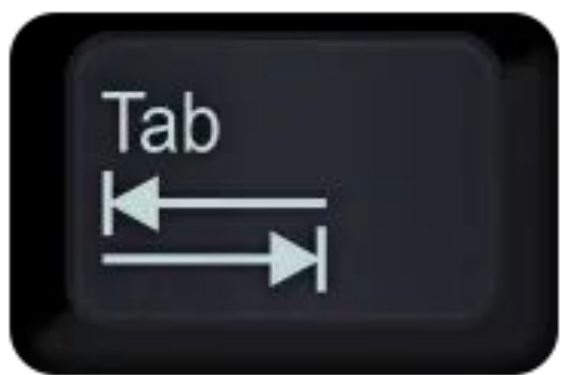
```
  <div class="marker m1"></div>
```

....

```
  <div class="marker m8"></div>
```

```
</div>
```

**Emmet**: `button[index=$]*64`



64 элемента `<button>`

8 маркеров



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
<div id="chess">
```

```
  <button index=1></button>
```

....

```
  <button index=64></button>
```

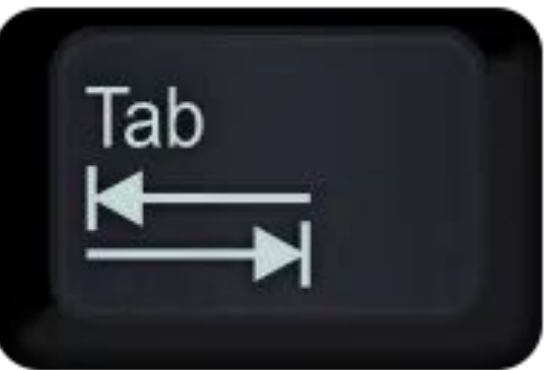
```
  <div class="marker m1"></div>
```

....

```
  <div class="marker m8"></div>
```

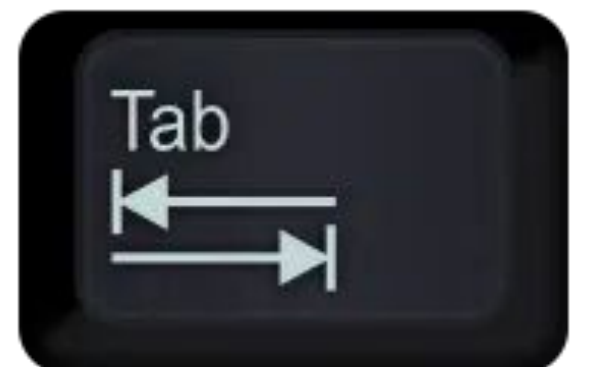
```
</div>
```

**Emmet**: `button[index=$]*64`

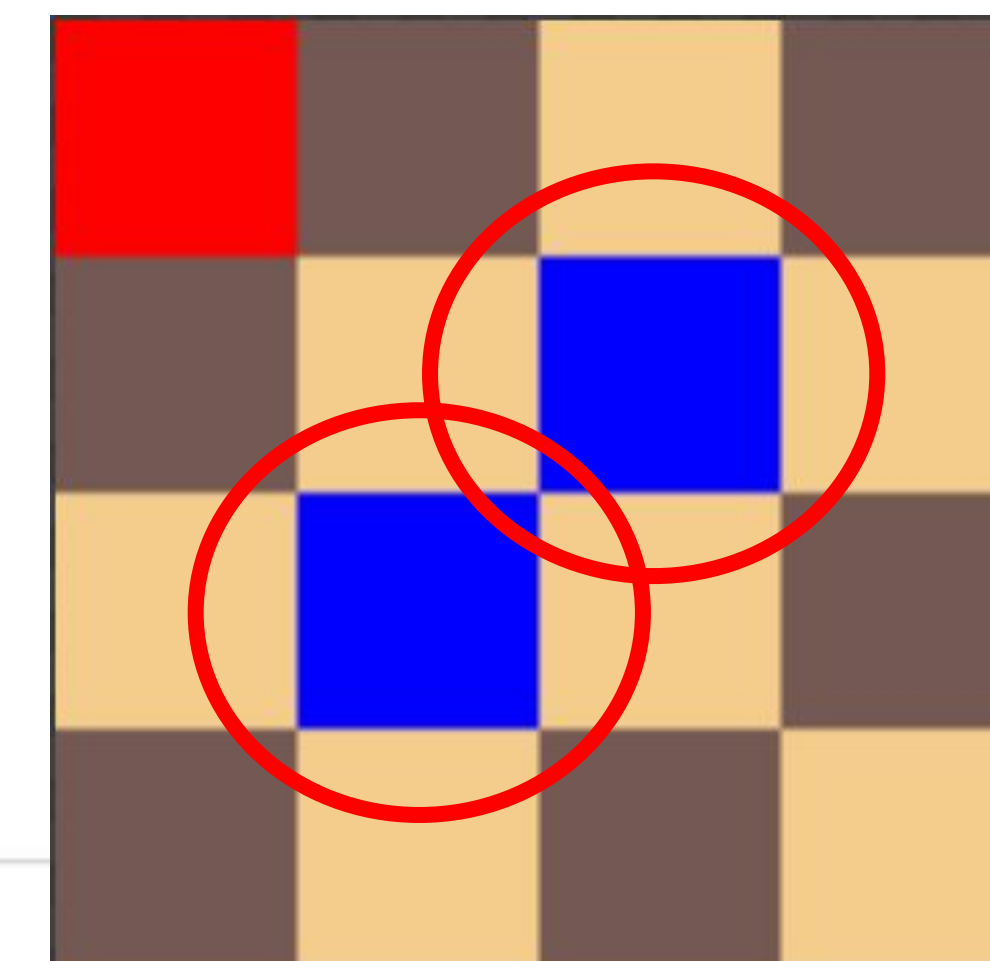


64 элемента `<button>`

**Emmet**: `div.marker.m$*8`



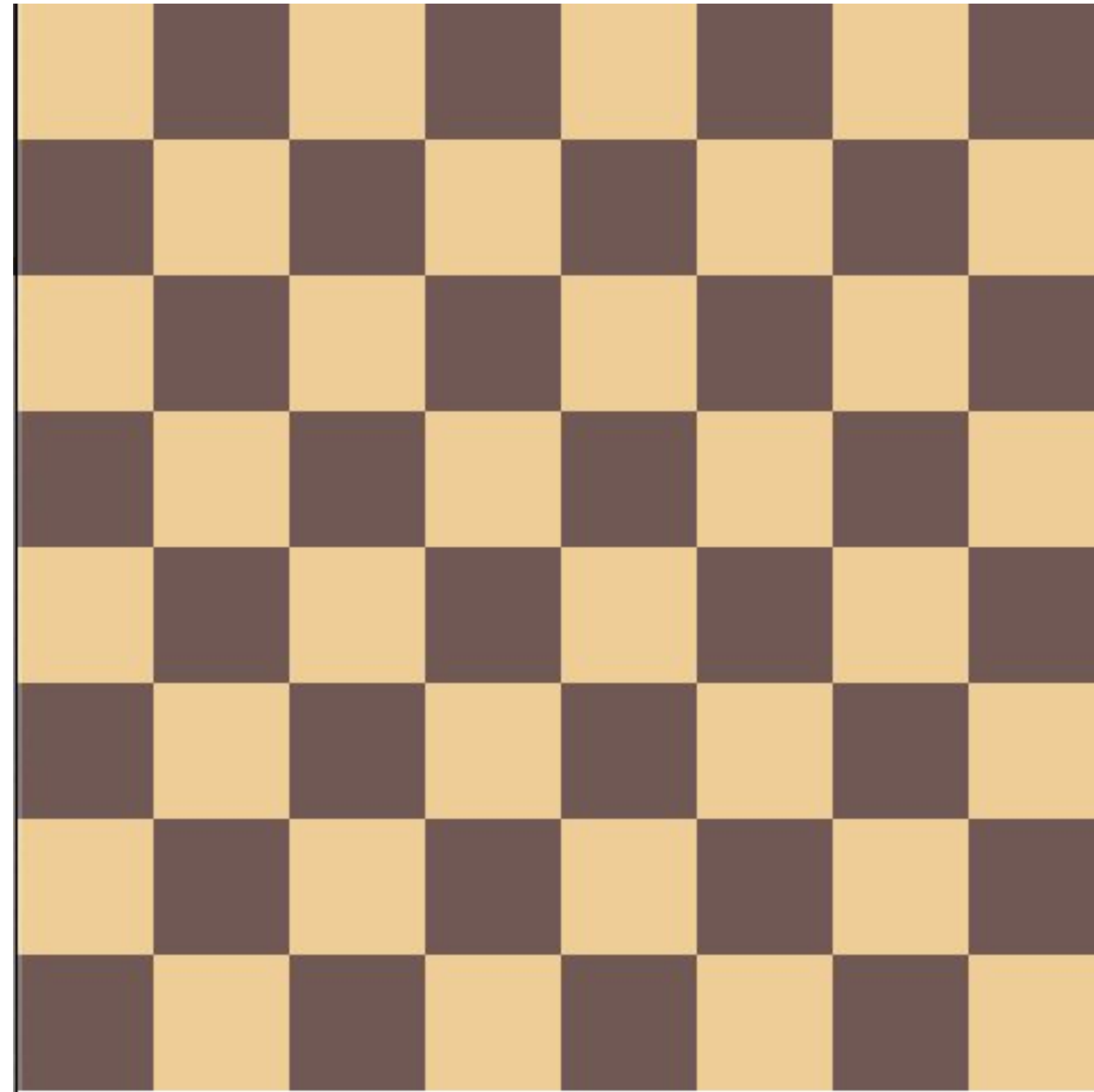
8 маркеров



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**selector1 ~ selector2 { style properties }**

**комбинатор смежных селекторов**

CSS 1	CSS 2	CSS 2.1	CSS 3
—	—	—	✓

# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**selector1 ~ selector2 { style properties }**

- Находит **selector2**, если ему предшествует **selector1**



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**selector1 ~ selector2 { style properties }**

- Находит **selector2**, если ему предшествует **selector1**
- **selector1** и **selector2** имеют общего родителя



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**selector1 ~ selector2 { style properties }**

- Находит **selector2**, если ему предшествует **selector1**
- **selector1** и **selector2** имеют общего родителя
- Свойства применяются ко второму селектору



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

**:nth-child(...)** псевдокласс

CSS 1	CSS 2	CSS 2.1	CSS 3
—	—	—	✓

# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## **:nth-child( $A_n+B$ )** псевдокласс

- Находит **1** или **более** элементов, основываясь на их позиции среди группы соседних элементов



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

## **:nth-child( $An+B$ )** псевдокласс

- Находит **1** или **более** элементов, основываясь на их позиции среди группы соседних элементов
- **tr:nth-child( $2n+1$ )**  
Описывает нечётные элементы HTML: **1, 3, 5**, и т. д.
- **tr:nth-child( $-2n+8$ )**  
Описывает четные элементы HTML: **8, 6, 4, 2** и т. д.

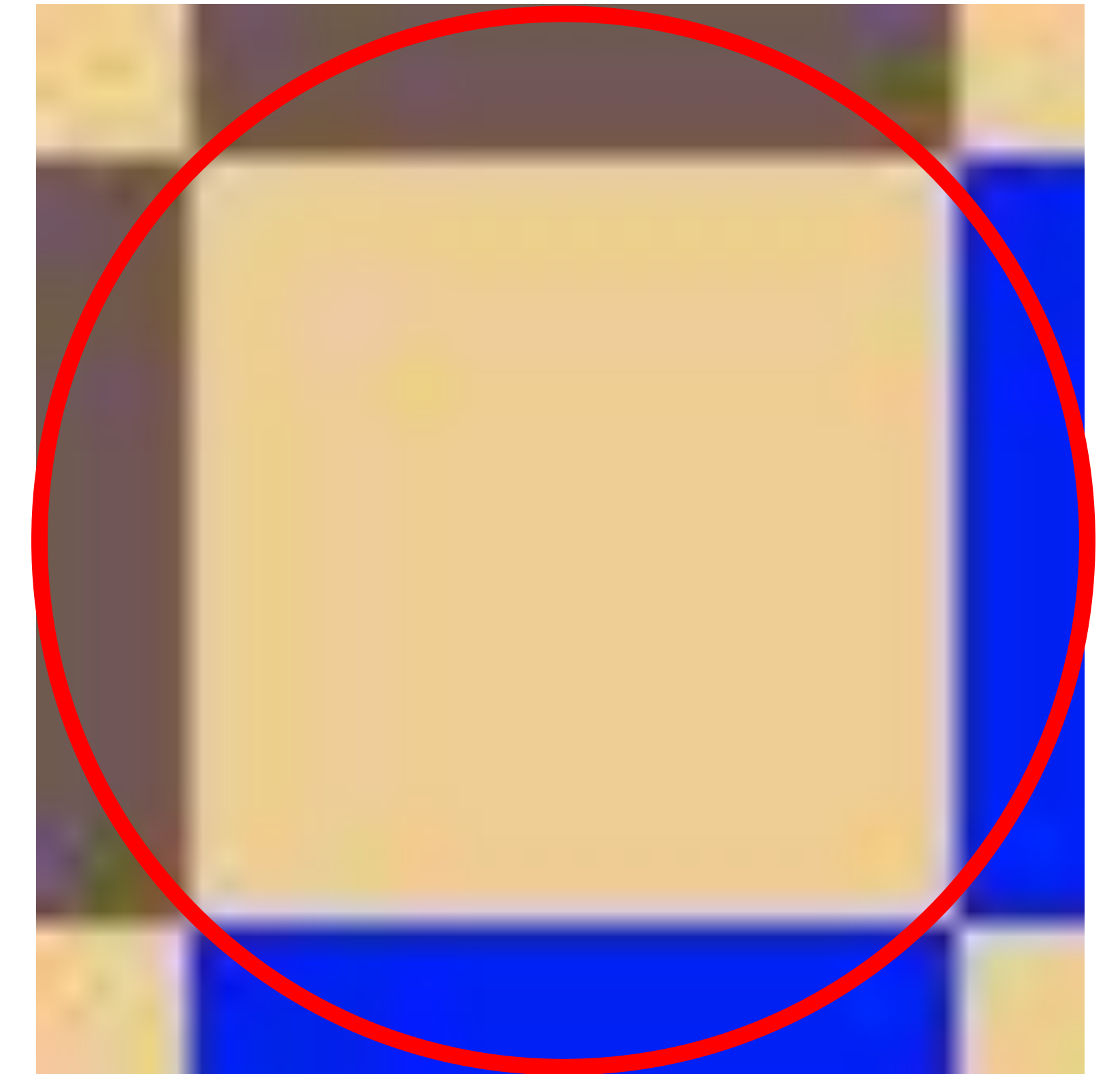


2019



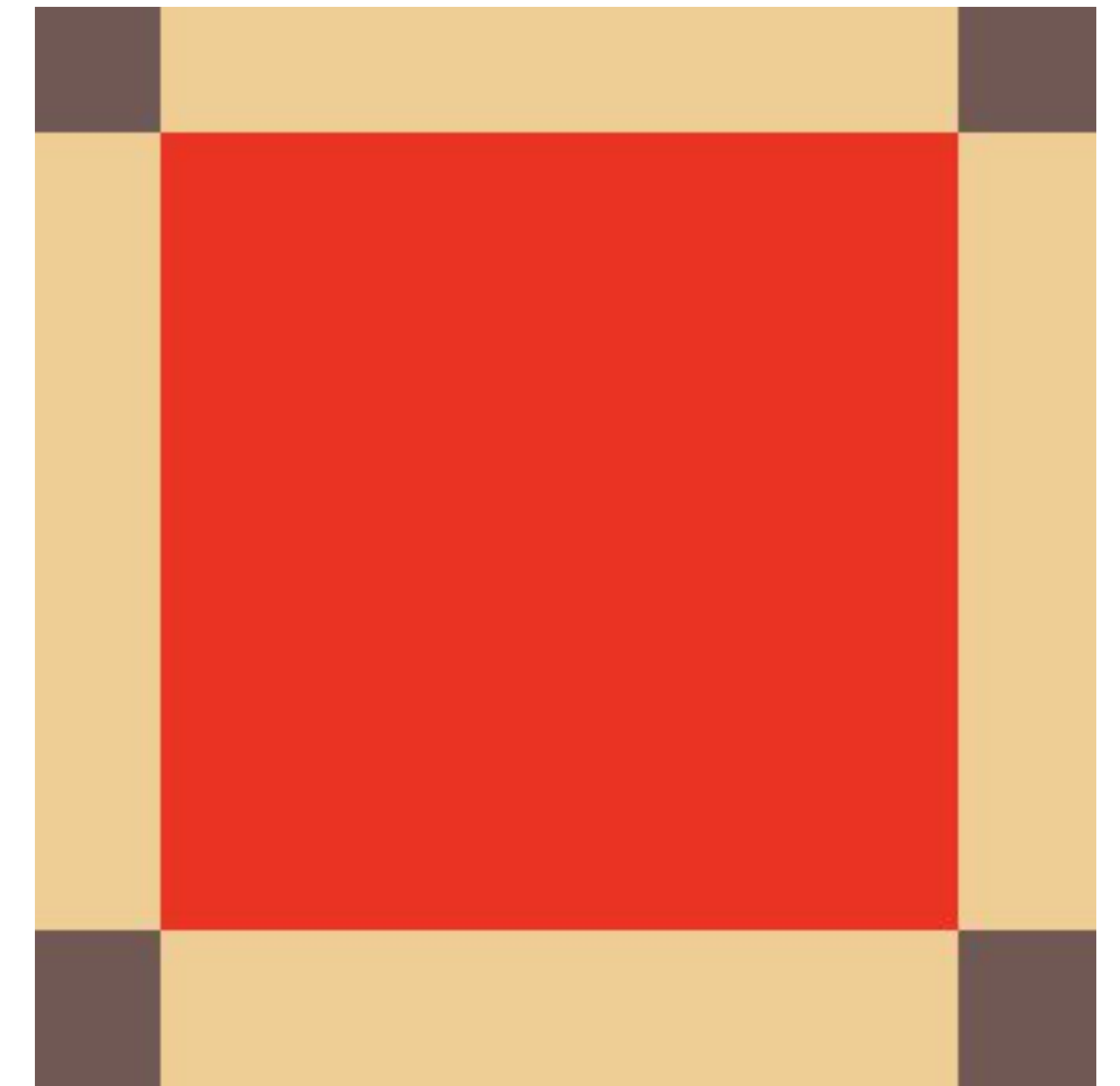
# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
#chess button:not(:focus) {  
  background-color: var(--cell-light-bg);  
}
```



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
#chess button:focus {  
  background-color: var(--cell-selected-bg);  
}
```



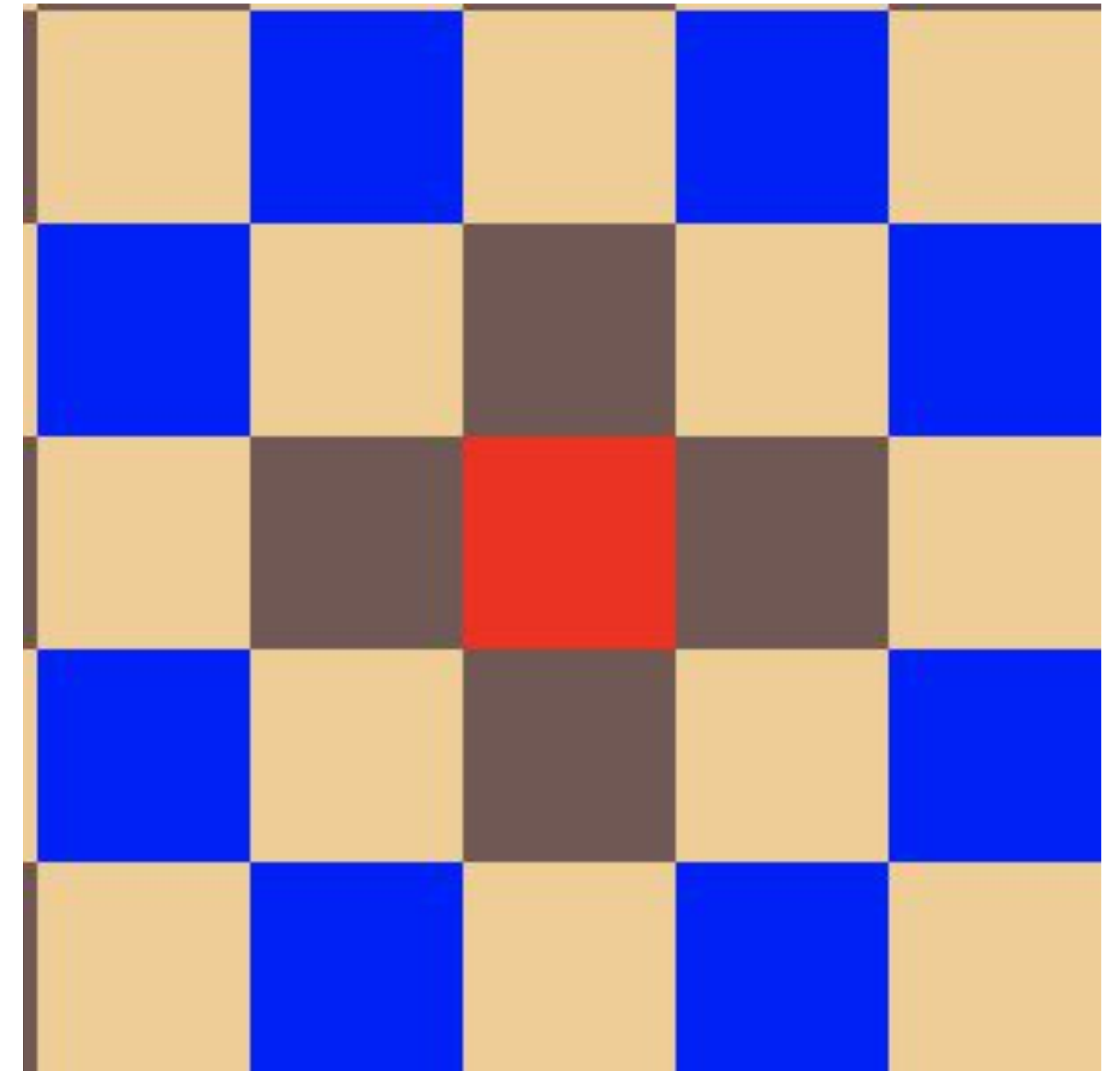
2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
.marker {  
  position: absolute;  
  left: calc(var(--cell-width) * -1); /* Скрытие маркеров */  
  top: calc(var(--cell-width) * -1); /* Скрытие маркеров */  
  width: var(--cell-width);  
  height: var(--cell-width);  
  pointer-events: none;  
  background-color: var(--cell-marker-bg);  
  color: white;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- А теперь рутина. Как быть?
  - Нужен css для маркеров **64-х** клеток шахматной доски  $64 * 8 \approx \mathbf{512}$  ВОЗМОЖНЫХ ПОЗИЦИЙ



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- А теперь рутина. Как быть?
  - Писать руками?



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- А теперь рутина. Как быть?
  - Писать руками?
  - Нет, спасибо! Поможет кодо-генерация!



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- А теперь рутина. Как быть?
  - SASS или SCSS

The Sass logo is written in a purple, cursive script font.

# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- А теперь рутина. Как быть?
  - Будем использовать SCSS





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

//Функция SCSS для проверки значения в диапазоне 1..8

```
@function is-in-range($v) {  
  @return $v >= 1 and $v <= 8;  
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
//Функция SCSS генерации raw css для маркера
@mixin marker-at($el-idx, $row, $col, $marker-idx) {
  @if is-in-range($row) and is-in-range($col) {
    button:nth-child(#{ $el-idx }):focus ~ .m#{ $marker-idx } {
      top: calc(#{ $row - 1 } * var(--cell-width));
      left: calc(#{ $col - 1 } * var(--cell-width));
    }
  }
}
```



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

// SCSS листы смещений для маркеров (по строкам, по столбцам)

\$row-offsets: 1, 2, 2, 1, -1, -2, -2, -1;

\$col-offsets: 2, 1, -1, -2, -2, -1, 1, 2;



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

```
// SCSS циклы генерации raw css для ~ 8 маркеров, каждой клетки шахматного поля
```

```
@for $row from 1 through 8 {
```

```
  @for $col from 1 through 8 {
```

```
    $el-idx: (($row - 1) * 8) + $col;
```

```
    @for $maker-idx from 1 through 8 {           // Генерируем CSS для маркеров
```

```
      $row-offset: nth($row-offsets, $maker-idx); // Смещение из листа по строке
```

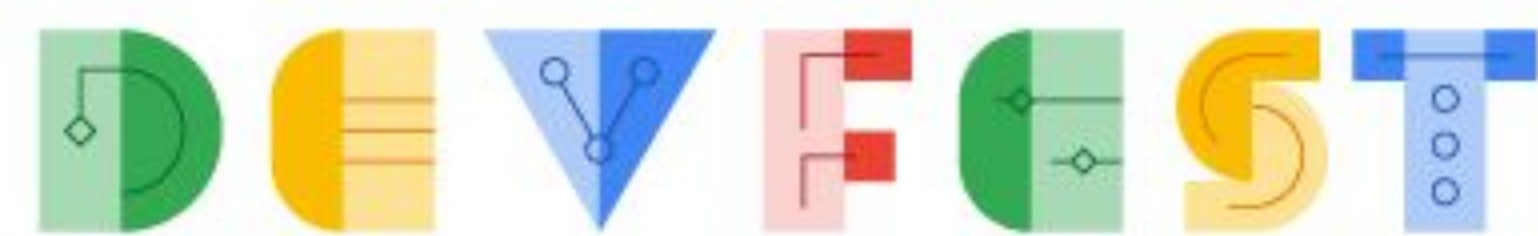
```
      $col-offset: nth($col-offsets, $maker-idx); // Смещение из листа по колонке
```

```
      @include marker-at($el-idx, $row + $row-offset, $col + $col-offset, $maker-idx);
```

```
    }
```

```
  }
```

```
}
```



2019





# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

- SCSS файл - 90 строк



2019



# ПОШАГОВЫЙ РАЗБОР ЗАДАЧИ

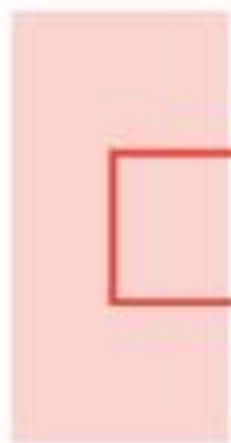
- SCSS файл - **90** строк
- CSS на выходе - **1740** строк  
(**несжатый CSS**)



2019



**КАК ТЕСТИРОВАТЬ?**



# КАК ТЕСТИРОВАТЬ?

- Что нужно?



2019





# КАК ТЕСТИРОВАТЬ?

- **Что нужно?**
  - **Эталонные скриншоты с расположением маркеров**



2019



# КАК ТЕСТИРОВАТЬ?

- **Что нужно?**

- **Эталонные скриншоты с расположением маркеров**
- **Сравнение текущего скриншота с эталонным**



2019



# КАК ТЕСТИРОВАТЬ?

- Как делать скриншоты?



2019



# КАК ТЕСТИРОВАТЬ?

- Как делать скриншоты?
  - **Puppeteer** + **Google Chrome** + **Немного кода**



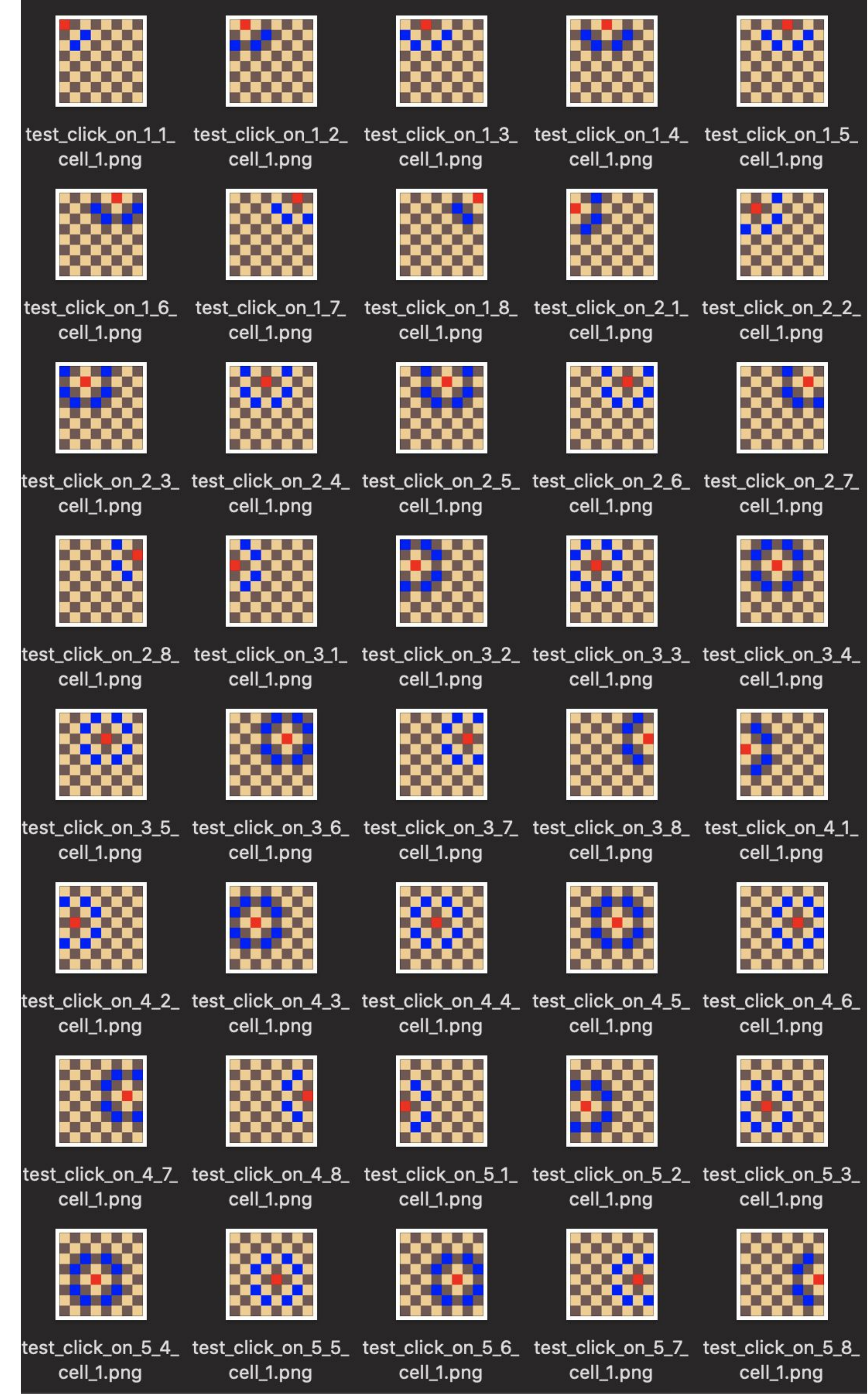
2019





# Эталонные скриншоты

Всего **80**  
скриншотов  
для **72** тестов

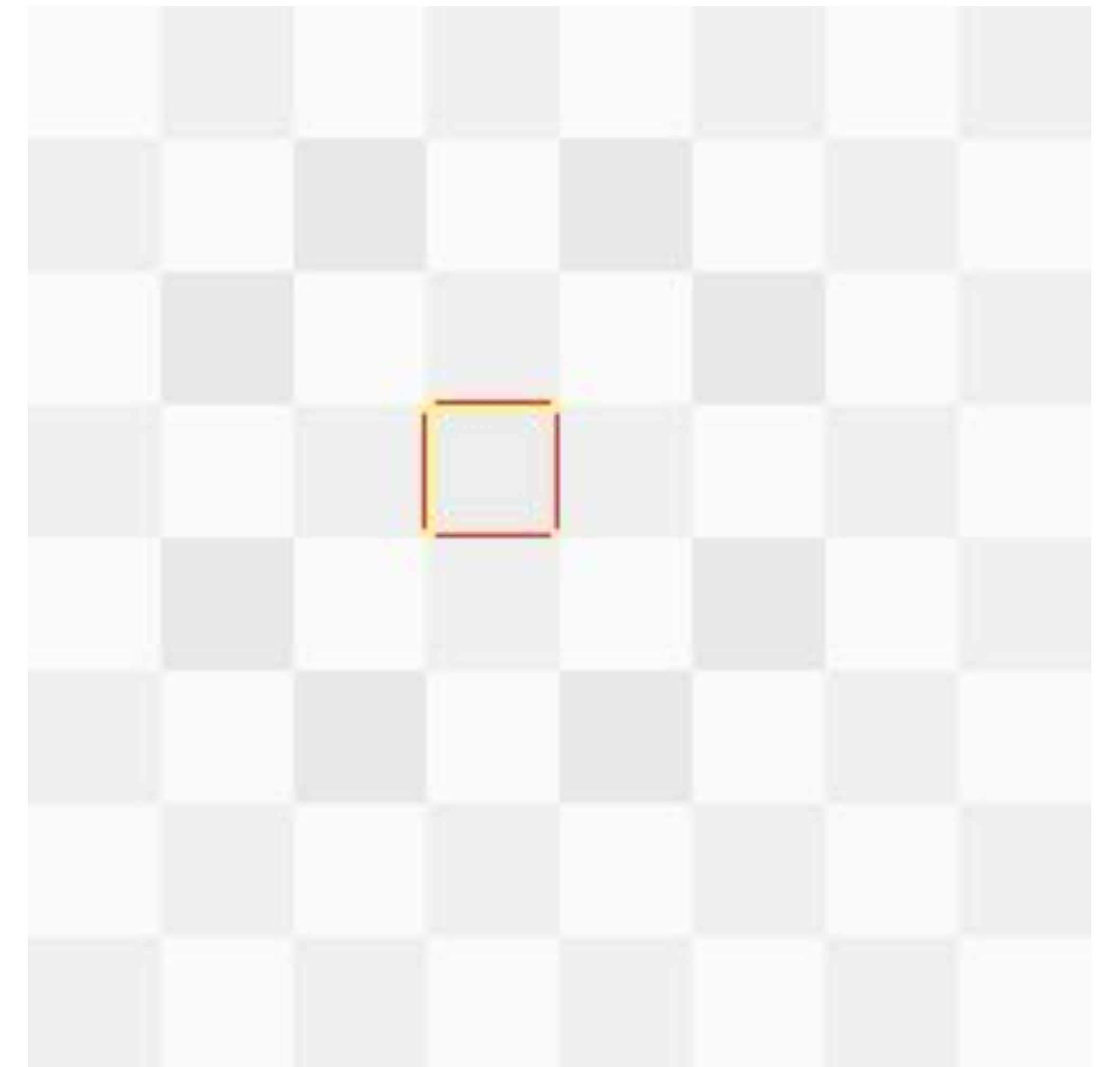




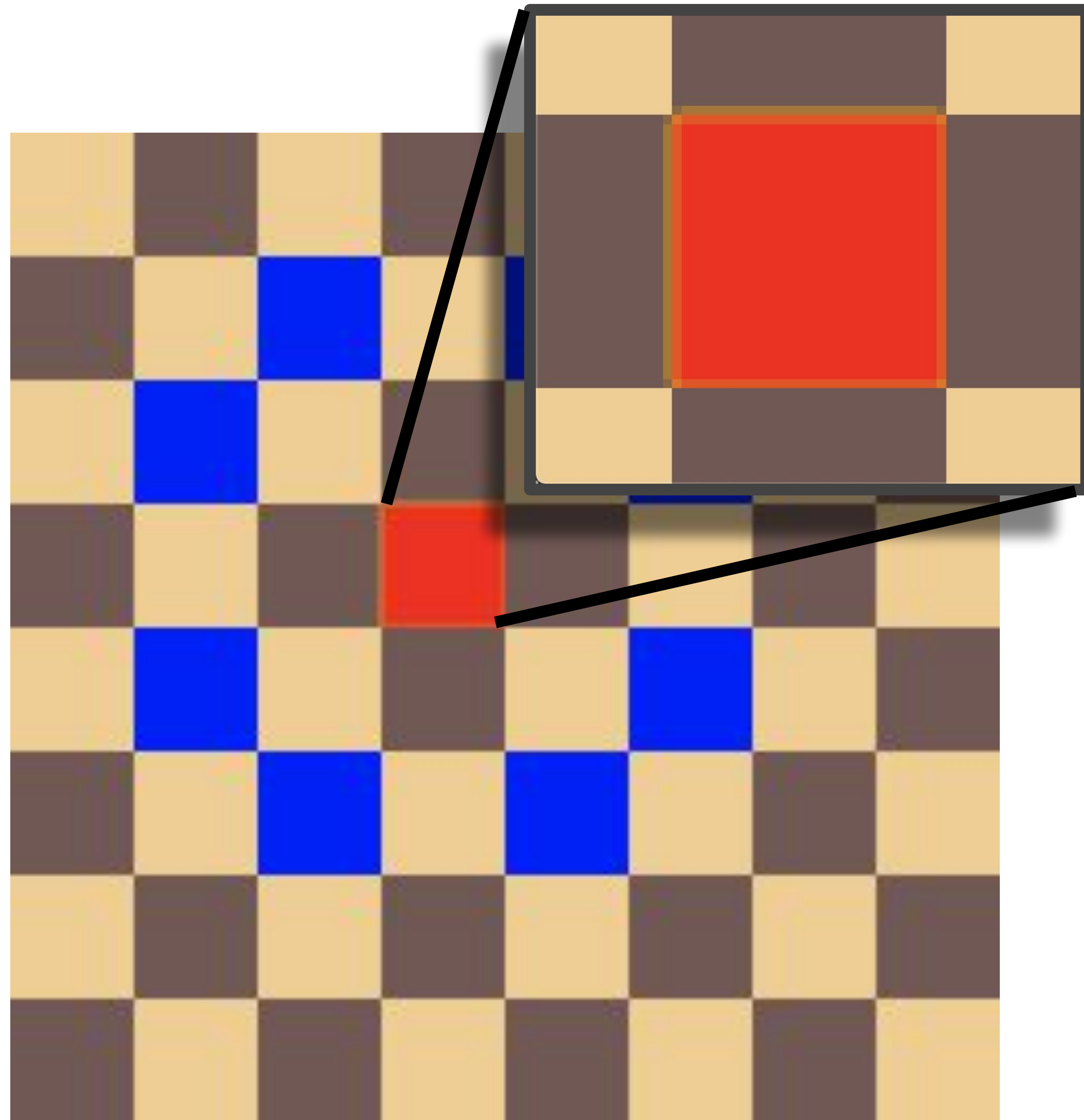
# КАК ТЕСТИРОВАТЬ?



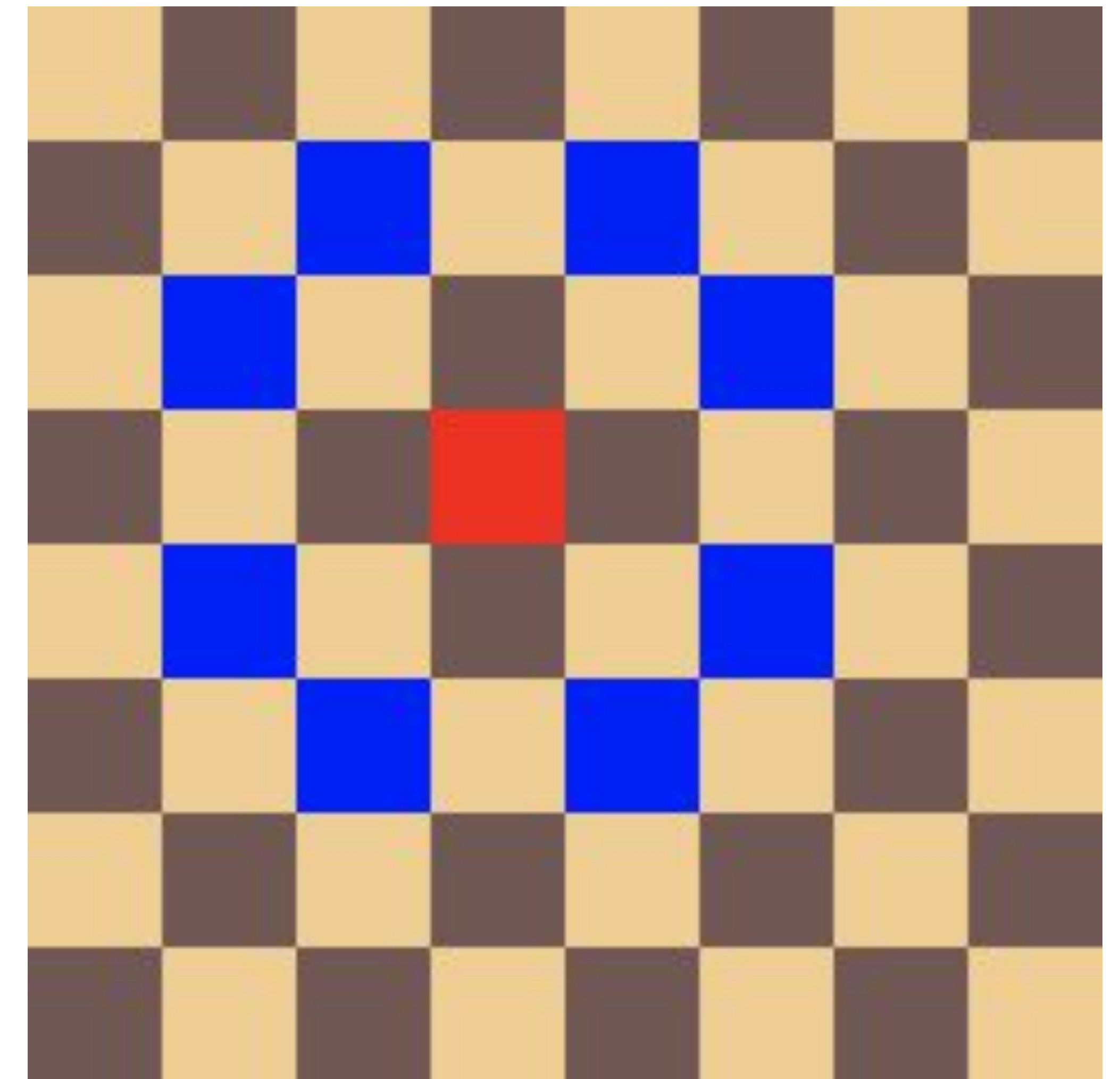
```
#chess button {  
  float: left;  
  border: none;  
  width: var(--cell-width);  
  height: var(--cell-width);  
}
```



# КАК ТЕСТИРОВАТЬ?

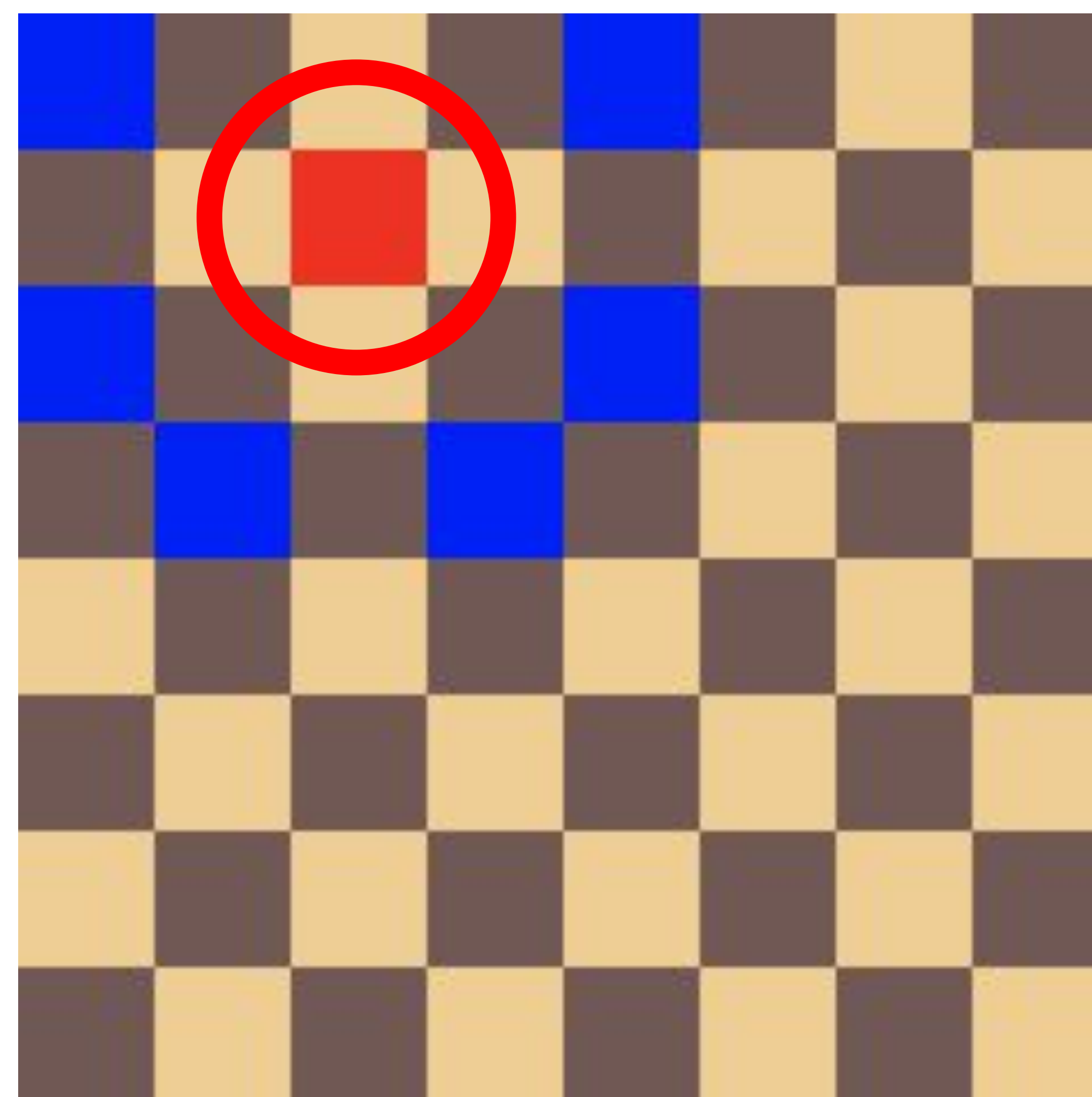
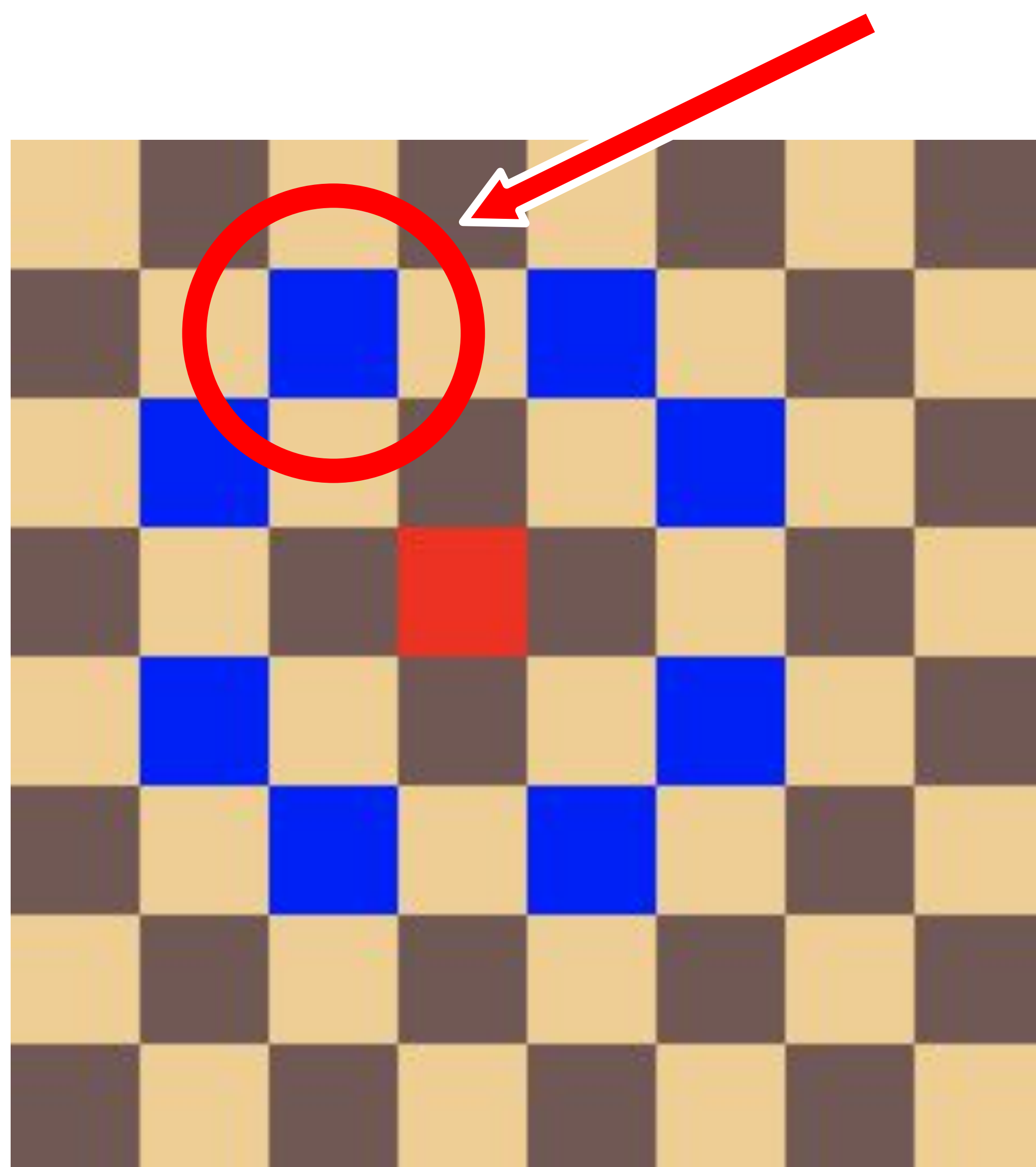


```
#chess button {  
  float: left;  
  border: none;  
  outline: none; // FIXED  
  width: var(--cell-width);  
  height: var(--cell-width);  
}
```





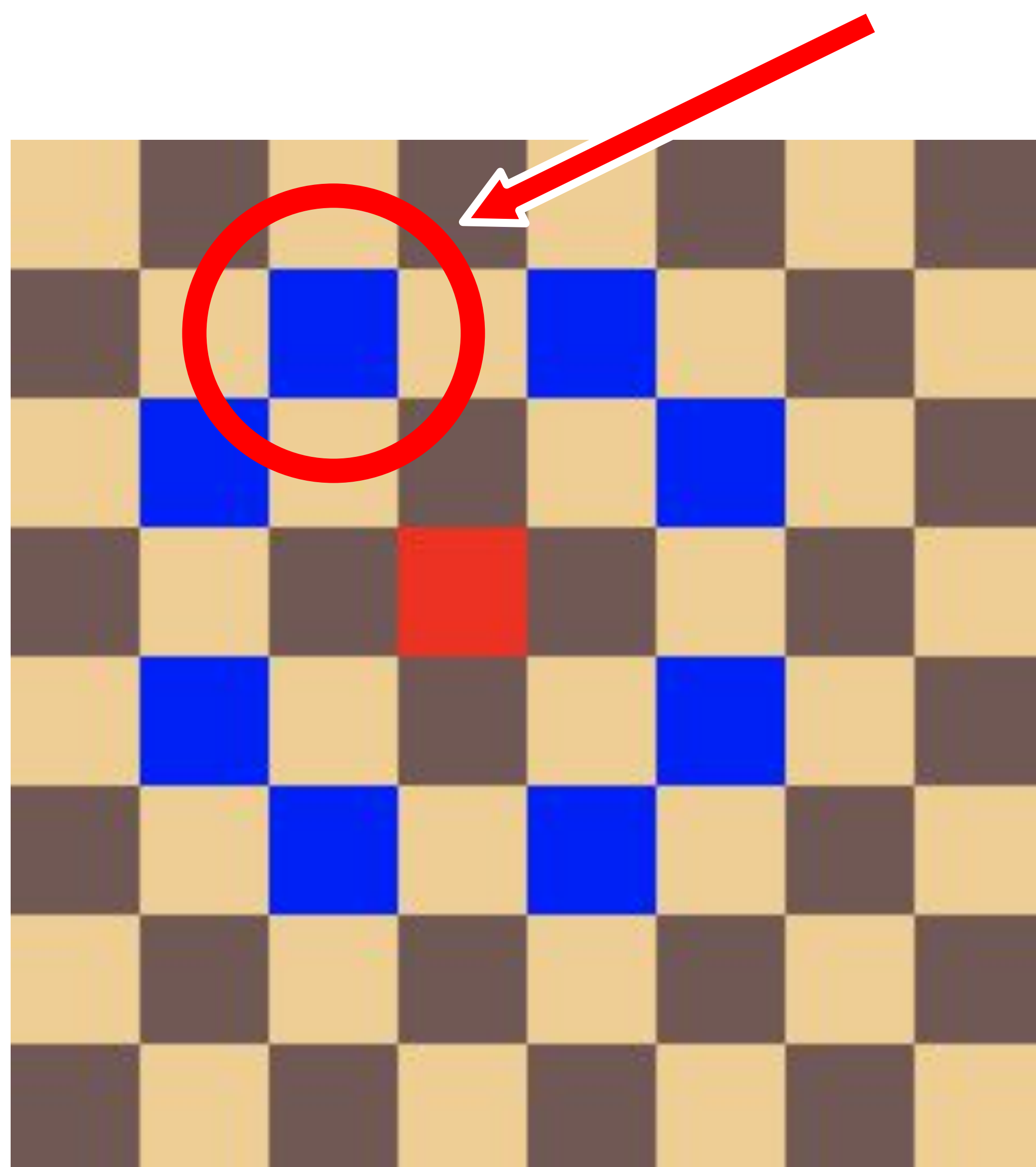
# КАК ТЕСТИРОВАТЬ?



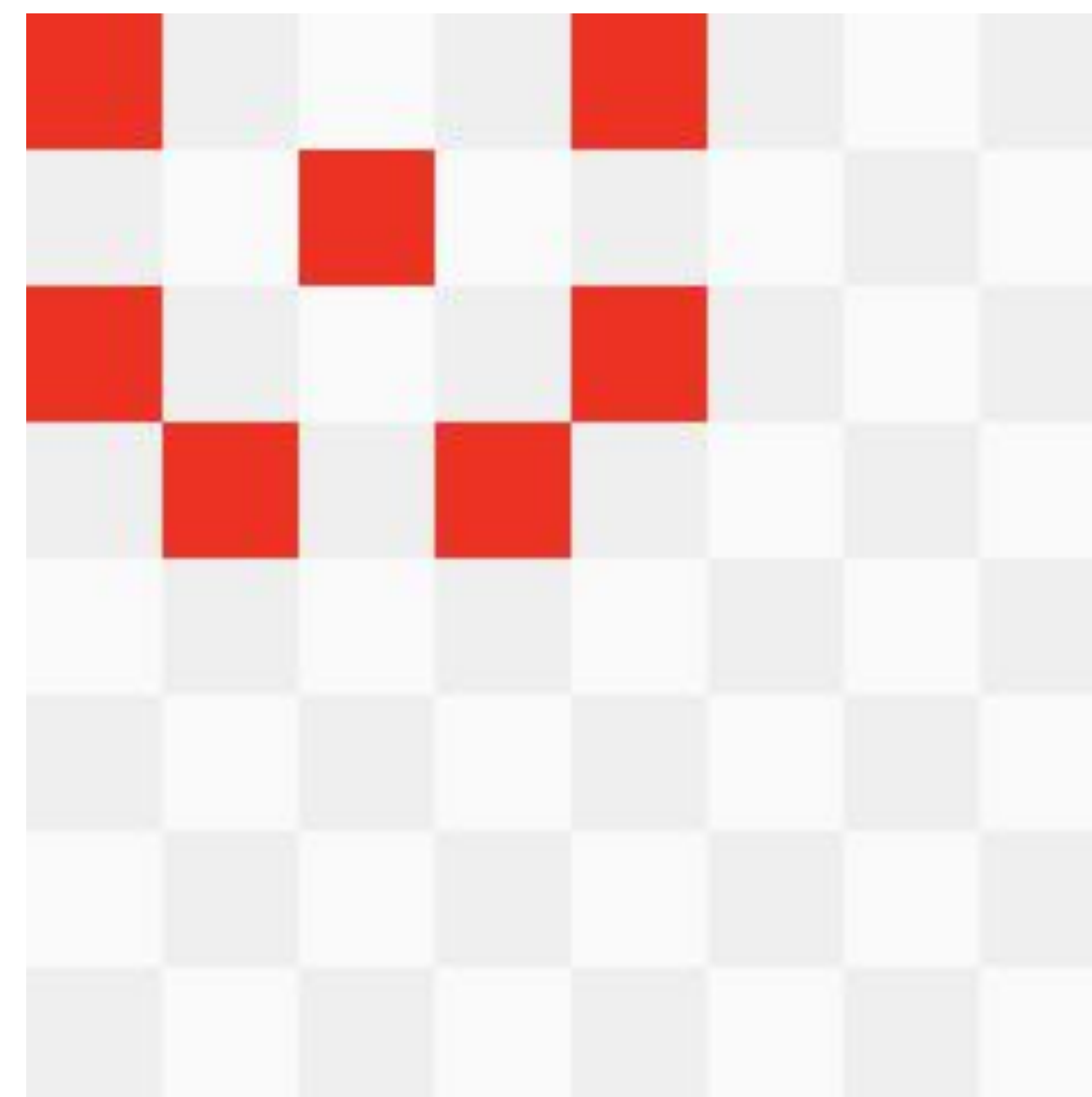
УСПЕШНЫЙ  
СЦЕНАРИЙ



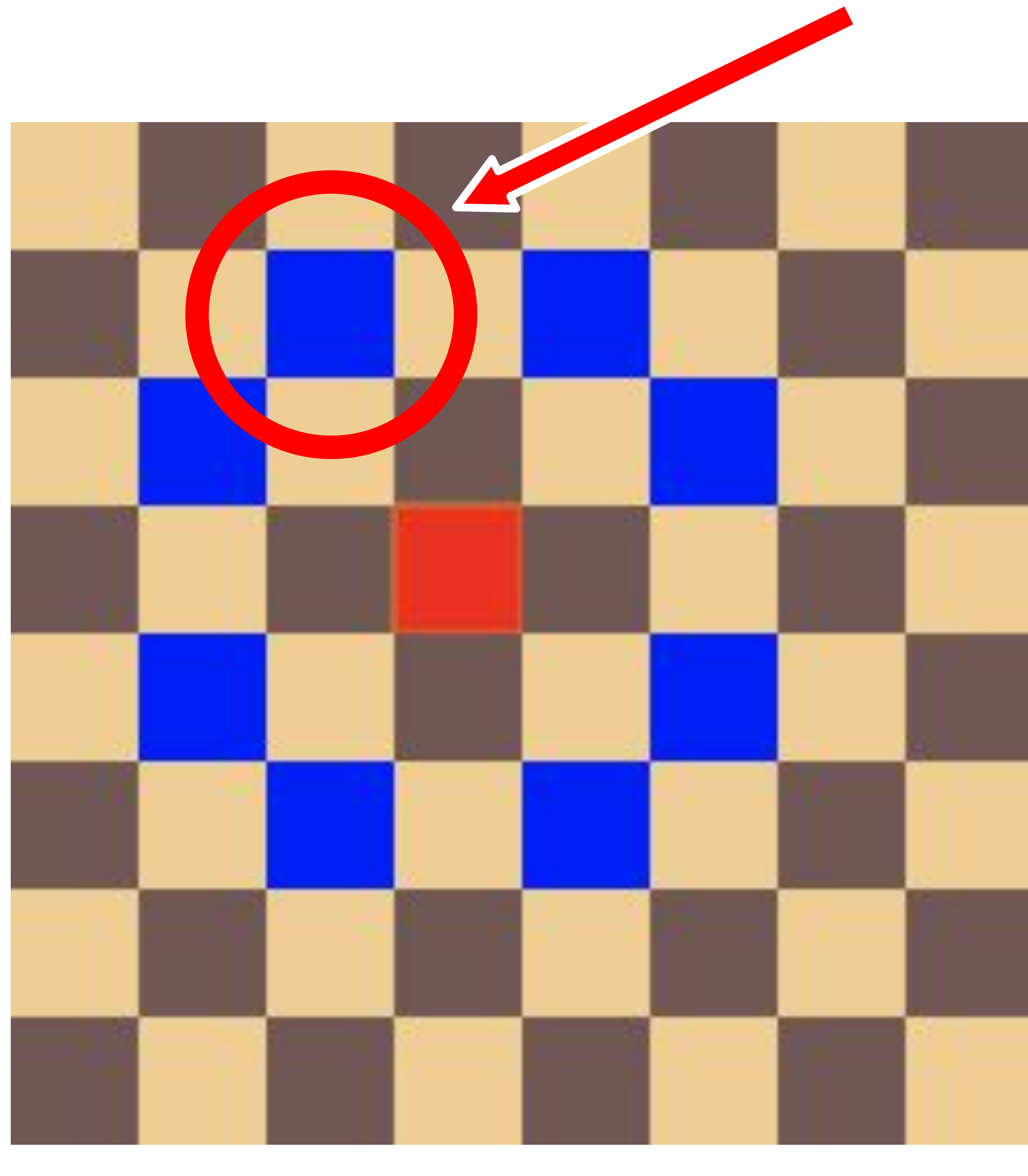
# КАК ТЕСТИРОВАТЬ?



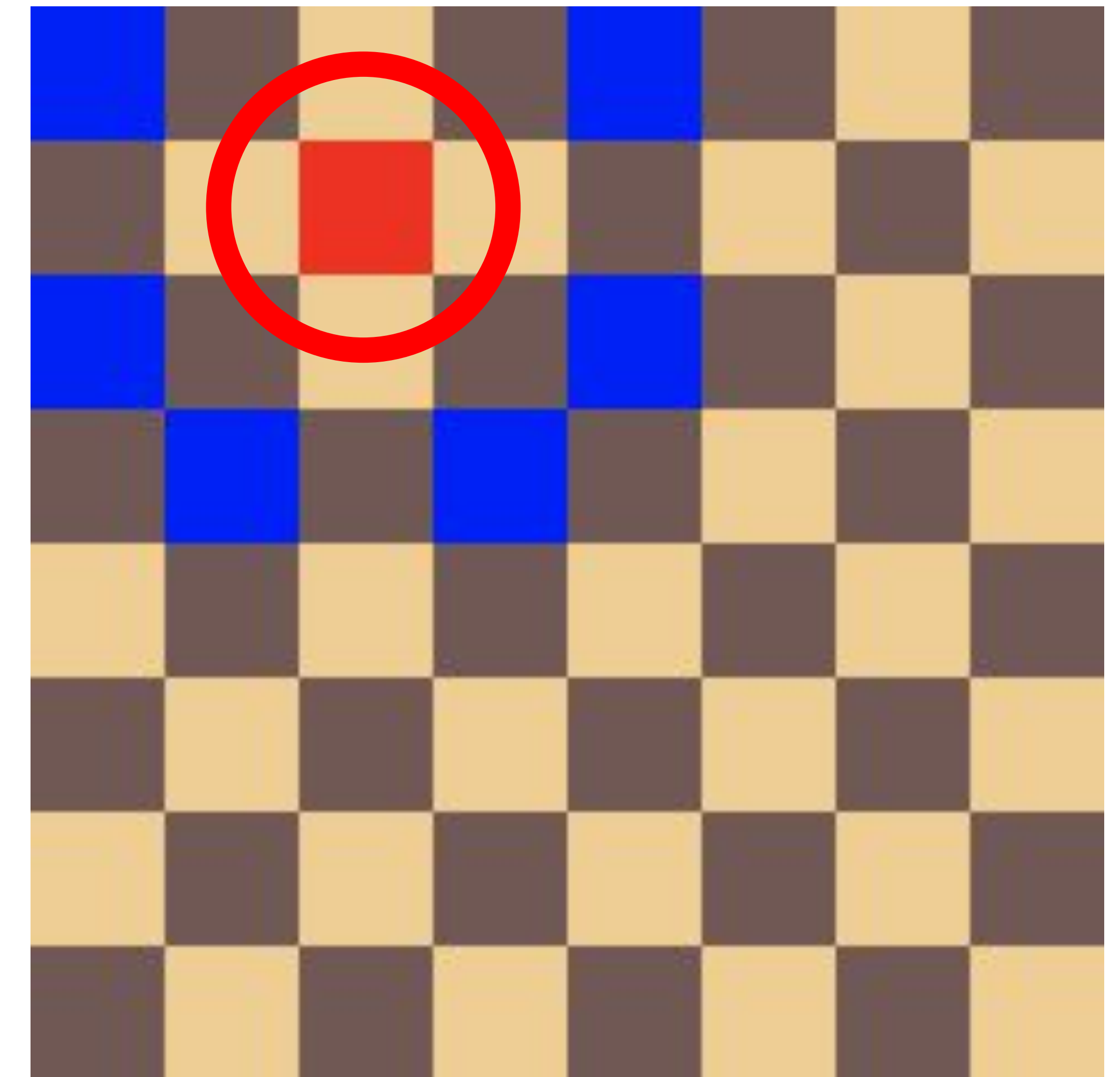
НЕУСПЕШНЫЙ  
СЦЕНАРИЙ



# КАК ТЕСТИРОВАТЬ?



```
.marker {  
  position: absolute;  
  left: calc(var(--cell-width) * -1);  
  top: calc(var(--cell-width) * -1);  
  width: var(--cell-width);  
  height: var(--cell-width);  
  pointer-events: none; // FIXED  
  background-color: var(--cell-marker-bg);  
  color: white;  
}
```





Example app listening on port 9001!

**Scenario:** markers\_click.test.js

X test\_click\_on\_top\_left\_marker

-----  
TESTS PASSED: 0

TESTS FAILED: 1

ERRORS: JestAssertionError: expect(received).toEqual(expected) // deep equality

Expected: 0

Received: 104

at checkScreenshots (/workspaces/devfest-gdg-2019/tests/index.js:102:28)

at processTicksAndRejections (internal/process/task\_queues.js:93:5)

at async test\_click\_on\_top\_left\_marker (/workspaces/devfest-gdg-2019/tests/markers\_click.test.js:4:5)

at async /workspaces/devfest-gdg-2019/tests/index.js:113:21 {

matcherResult: {

actual: 104,

expected: 0,

message: [Function],

name: 'toEqual',

pass: false

}

}



2019





PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
✓ test_click_on_7_2_cell  
✓ test_click_on_7_3_cell  
✓ test_click_on_7_4_cell  
✓ test_click_on_7_5_cell  
✓ test_click_on_7_6_cell  
✓ test_click_on_7_7_cell  
✓ test_click_on_7_8_cell
```

**Scenario:** single\_cell\_click\_row8.test.js

```
✓ test_click_on_8_1_cell  
✓ test_click_on_8_2_cell  
✓ test_click_on_8_3_cell  
✓ test_click_on_8_4_cell  
✓ test_click_on_8_5_cell  
✓ test_click_on_8_6_cell  
✓ test_click_on_8_7_cell  
✓ test_click_on_8_8_cell
```

-----  
TESTS PASSED: 72

TESTS FAILED: 0

root@4242ffab01ef:/workspaces/devfest-gdg-2019#



2019



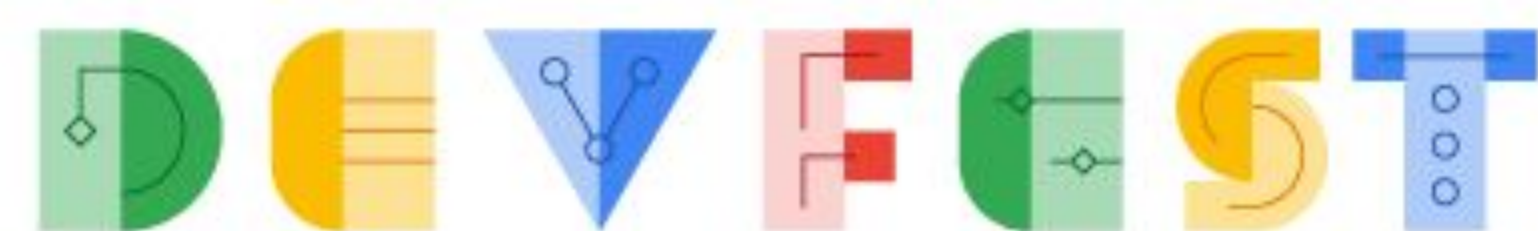
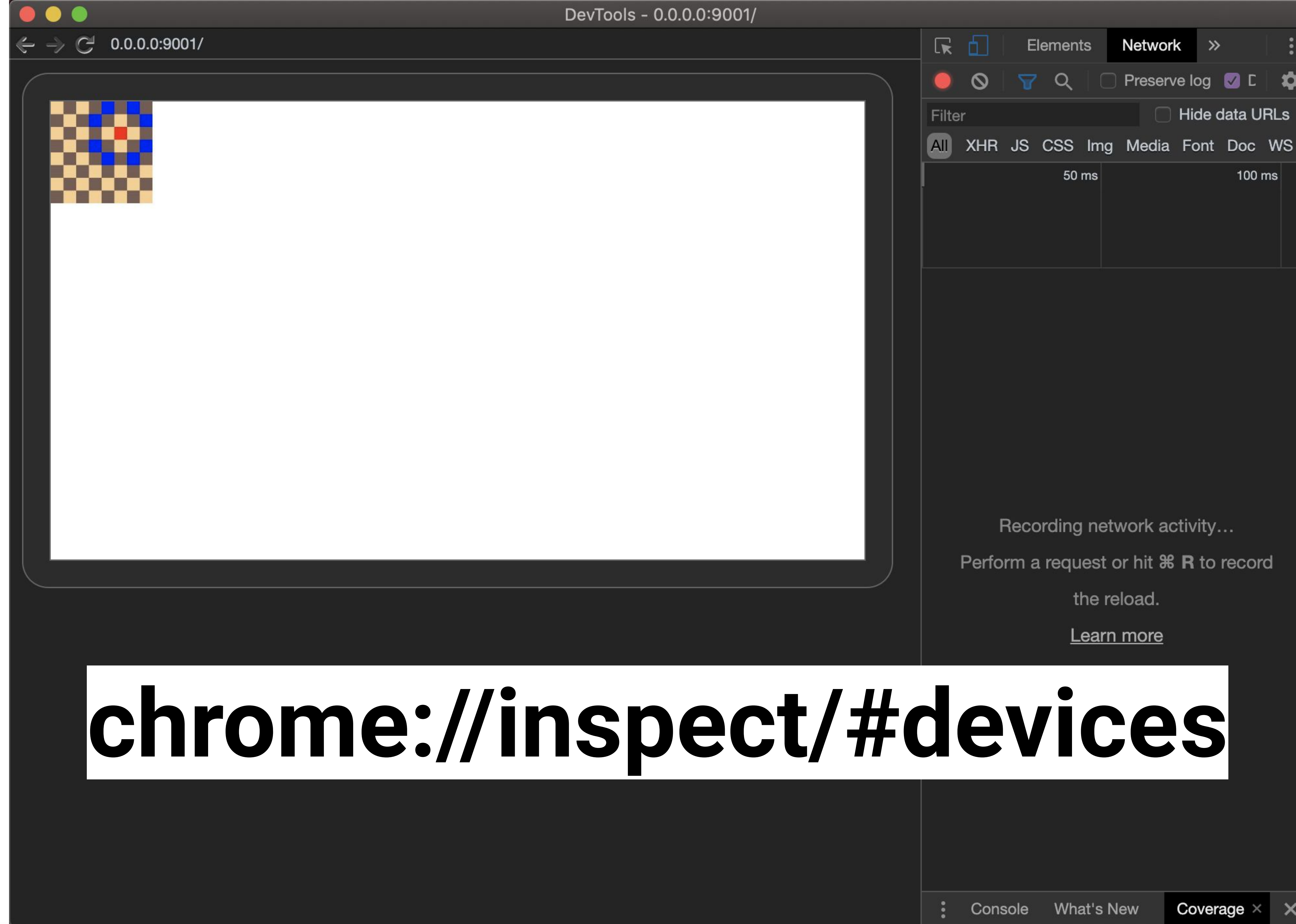


```
async function test_click_on_1_1_cell({ page, diff, checkScreenshots, getCoordsByRowCol }) {  
  await page.mouse.click( ...getCoordsByRowCol(1, 1));  
  await checkScreenshots(diff);  
}
```



2019



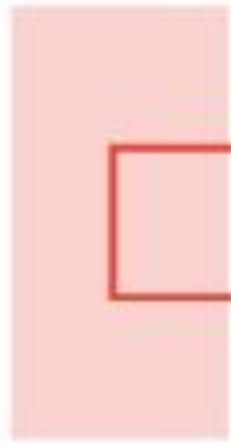


2019





# ПОДВЕДЕНИЕ ИТОГОВ



# ПОДВЕДЕНИЕ ИТОГОВ

- Если попадаются задачи с состояниями в CSS, то:



2019





# ПОДВЕДЕНИЕ ИТОГОВ

- Если попадаются задачи с состояниями в CSS, то:
  - Невозможно решить за приемлемое время, вручную

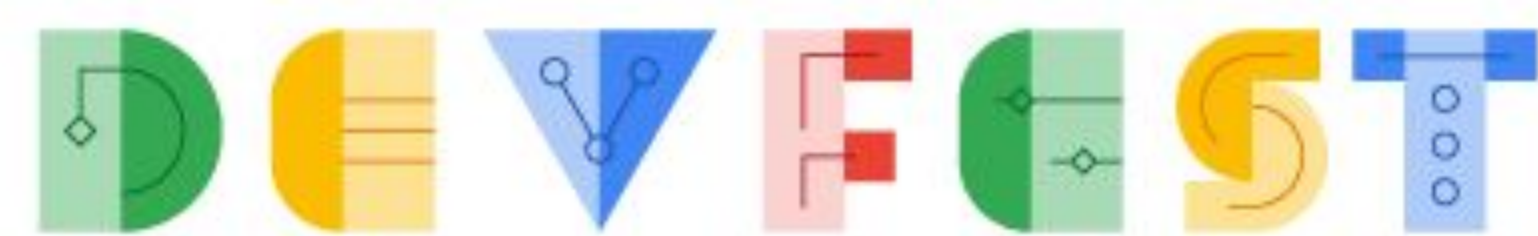


2019



# ПОДВЕДЕНИЕ ИТОГОВ

- Если попадаются задачи с состояниями в CSS, то:
  - Невозможно решить за приемлемое время, вручную
  - Но можно использовать генерацию!

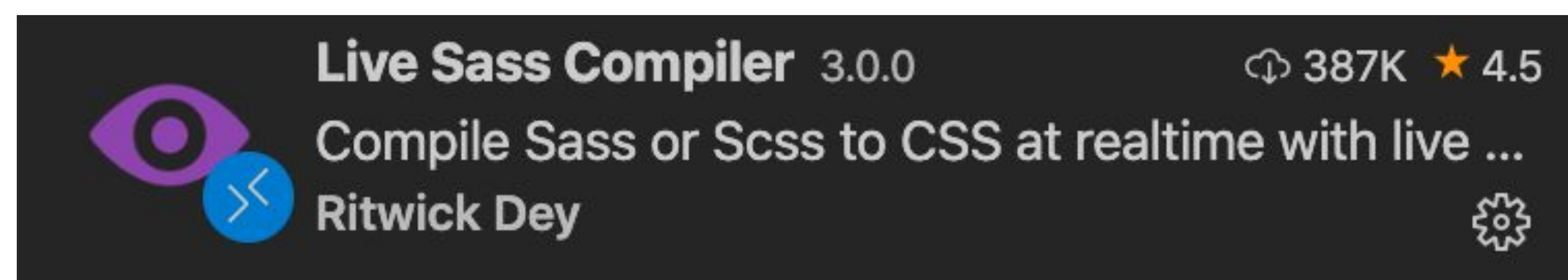


2019



# ПОДВЕДЕНИЕ ИТОГОВ

- Если попадаются задачи с состояниями в CSS, то:
  - Невозможно решить за приемлемое время вручную
  - Но можно использовать генерацию!
  - Использовать возможности вашей IDE (к примеру, **Live SASS Compiler**)



2019





# ПОДВЕДЕНИЕ ИТОГОВ

- Если попадаются задачи с состояниями в CSS, то:
  - Невозможно решить за приемлемое время вручную
  - Но можно использовать генерацию!
  - Использовать возможности вашей IDE (к примеру, **Live SASS Compiler**)
  - Для генерации большого кол-ва HTML



2019





Код решения задачи

**<https://git.io/JeySd>**

SCAN ME

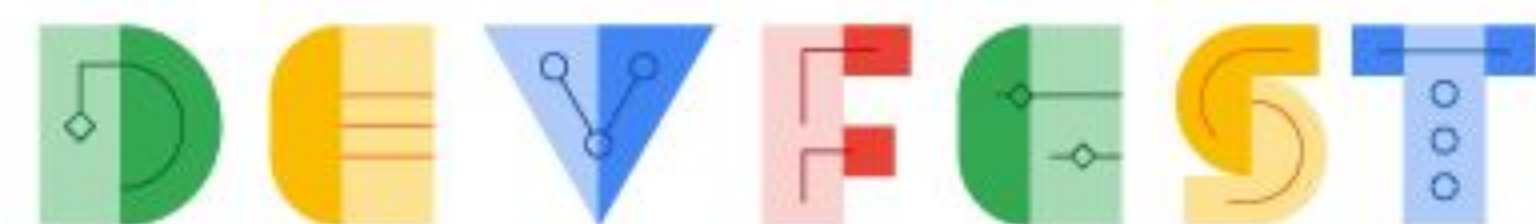




**Всем спасибо!**

Вопросы?

SCAN ME



2019