

Выполнение работы.

Были использованы следующие переменные:

var **canvas**, **figure** — для связи HTML-кода и основного игрового поля, поля, отражающего следующую фигуру соответственно;

var **ctx**, **ctx_figure** - для отрисовки основного поля и поля, где изображена следующая фигура;

var **xm** = 10 — количество клеток на игровом поле по горизонтали;

var **ym** = 15 - количество клеток на игровом поле по вертикали;

var **N** = 30 — ширина одной клетки в пикселях;

var **Time** — время падения фигуры, которое будет уменьшаться при увеличении уровня;

var **matrix** = матрица, хранящая информацию о заполнении поля (0 — если клетка свободна и цифры от 1 до 10 в зависимости от цвета, если клетка несвободна);

var **active_figure** - массив, хранящий координаты по x и y каждой клетки активной фигуры, а также тип фигуры;

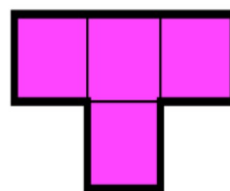
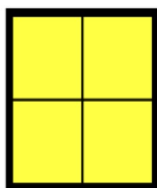
| Номер элемента | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|----------------|----|----|----|----|----|----|----|----|------------|
| Что хранит | x1 | y1 | x2 | y2 | x3 | y3 | x4 | y4 | Тип фигуры |

x1..x4 — значения по x для 1..4 клеток соответственно

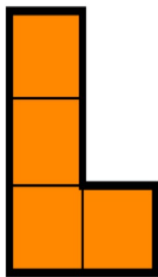
y1..y4 — значения по y для 1..4 клеток соответственно

Было разработано 7 типов фигур, используемых в тетрисе:

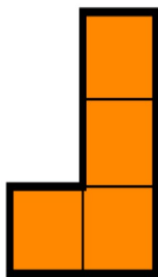
1) 2) 3)



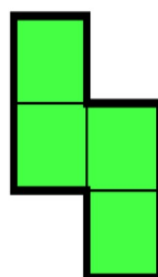
4)



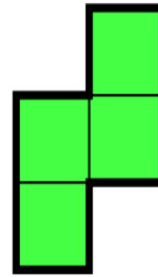
5)



6)



7)



var **rgb_active_figure** - массив, хранящий значения r, g, b и номер цвета активной фигуры;

| | | | | |
|----------------|---|---|---|-------------|
| Номер элемента | 0 | 1 | 2 | 3 |
| Что хранит | r | g | b | Номер цвета |

var **next_figure** - хранит описание следующей фигуры, аналогично, **active_figure**;

var **rgb_next_figure** - хранит описание цветов следующей фигуры, аналогично, **rgb_active_figure** ;

var **matrix_figure** — матрица для изображения следующей фигуры;

var **r, g, b** — значения цвета red, green, blue;

var **flag** — флаг для определения конца игры (false, если стакан заполнен);

var **level** - переменная, отвечающая за подсчет уровня;

Были реализованы следующие функции:

1) color(col) — задает значения r, g, b для переданного цвета;

2) figure1, ..., figure7 - для описания фигур;

3) changeMatrix() - меняет матрицу, при появлении новой фигуры;

4) turnF2(), ..., turnF7() - обеспечивают повороты фигур, меняют координаты матрицы;

5) getRandomIntInclusive(min, max) — возвращает рандомное значение от min до max включительно;

- 6) `game()` - запускает игру, заполняет матрицу для хранения информации о состоянии игрового поля и матрицу о следующей фигуре нулями;
- 7) `startGame()` - обеспечивает задание основных переменных, вызывает функции для отрисовки поля, взаимодействия пользователя с игрой, движение фигур;
- 8) `liveFigure()` - функция, реализующая «жизнь» фигуры, в ней проверяется условие окончания игры, вывод таблицы рекордов;
- 9) `init()` - инициализирует имя пользователя и уровень, связывает javascript-код и HTML-код;
- 10) `drawNet(ctx1, xmax, ymax)` — рисует сетку на полях;
- 11) `drawNextFigure()` - отрисовывает следующую фигуру;
- 12) `square1(col, x, y)` — отрисовывает квадрат на поле следующей фигуры;
- 13) `new_figure()` - случайно выбирает следующую фигуру;
- 14) `checkStr()` - проверяет можно ли убрать строку и уменьшает время, если можно;
- 15) `canCleanStr(n)` — условие для проверки заполненности строки;
- 16) `cleanStr(n)` — очистка строки;
- 17) `canGoDown()` - проверка условия, что можно двигать фигуру вниз;
- 18) `square(col, x, y)` - отрисовывает квадрат на игровом поле;
- 19) `go()` - движение фигуры вниз на одну клетку;
- 20) `CanGoRight()` - условие для проверки возможности двигать фигуру вправо;
- 21) `CanGoLeft()` - условие для проверки возможности двигать фигуру влево;
- 22) `moving_figure()` - взаимодействие пользователя с игрой, обеспечение перемещения фигуры вправо, влево, поворота и падения, при нажатии клавиш.
- 23) `store()` - функция, достающая введенное пользователем имя в окне ввода;
- 24) `read()` - возвращает имя пользователя;
- 25) `setUsername()` - функция для сохранения имени игрока и отображения его в поле ввода при повторном входе;
- 26) `saveRecord(level)` — сохраняет рекорды игроков;
- 27) `getRecords()` - возвращает записи о рекордах.

Введите имя:

Ввод

[illegible]

Рисунок 2 — Основная страница

В файле style.css разработан дизайн основной страницы и всплывающего модального окна с результатами.

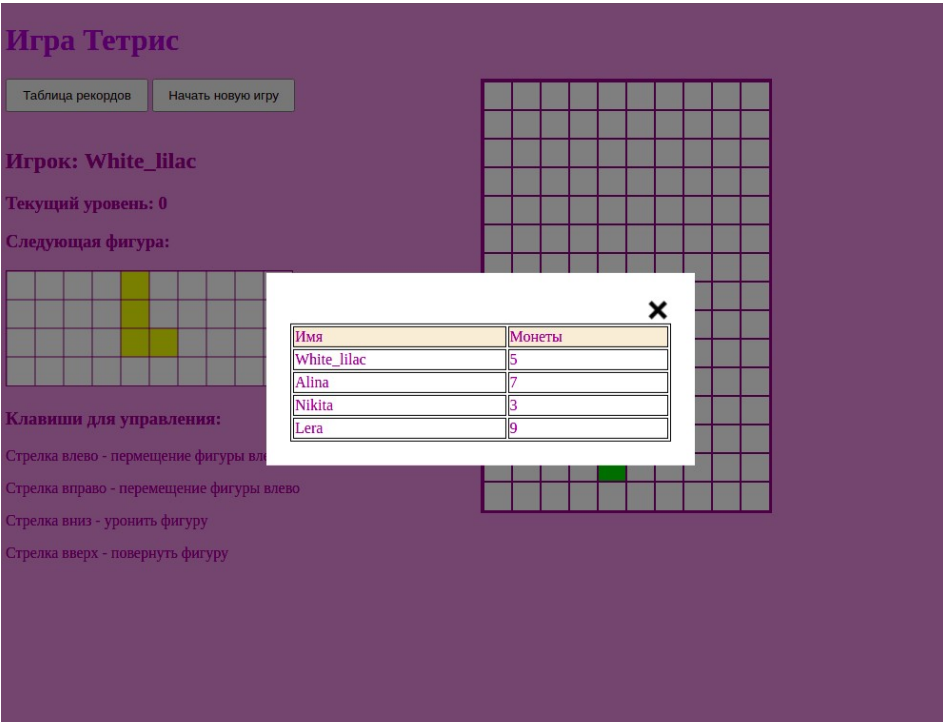


Рисунок 3 — Всплывающее окно — таблица результатов