Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию

Выполнил:
студент группы ИУ5-32Б
Кудрявцев Андрей
Подпись и дата:

Проверила:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2023 г.

Постановка задачи

Разработать игру „змейка“ для двух игроков, используя языки «HTML», «CSS» и «JavaScript».

Для создания проекта и знакомства с новыми синтаксическими конструкциями за основу была взята данная статья: https://thecode.media/snake-js/

Текст программы

```
<!DOCTYPE html>
<html>

<head>
  <title>Змейка</title>
  <style>
    html,
    body {
      height: 100%;
      margin: 0;
    }

    /*Задаём глобальные параметры*/
    body {
      background: black;
      display: flex;
      align-items: center;
      justify-content: center;
    }

    /*Делаем границу вокруг игрового поля*/
    canvas {
      border: 1px solid white;
    }
  </style>
</head>

<body>
  <!-- Рисуем игровое поле -->
  <canvas width="400" height="400" id="game"></canvas>
  <!-- Сам скрипт с игрой -->
  <script>
    // Поле, на котором всё будет происходить, — тоже как бы переменная
    var canvas = document.getElementById('game');
    // Классическая змейка — двухмерная, сделаем такую же
    var context = canvas.getContext('2d');
    // Размер одной клеточки на поле — 16 пикселей
    var grid = 16;
    // Служебная переменная, которая отвечает за скорость змейки
    var count = 0;
    // А вот и сама змейка
    var score1 = 0; // переменная для хранения количества набранных очков
    var score2 = 0;
    var snake1 = {
      // Начальные координаты
      x: 160,
      y: 160,
      // Скорость змейки — в каждом новом кадре змейка смещается по оси X или Y.
      dx: grid,
      dy: 0,
      // Тащим за собой хвост, который пока пустой
      cells: [],
      // Стартовая длина змейки — 4 клеточки
      maxCells: 4
    };

    var snake2 = {

      x: 320,
      y: 320,

      dx: -grid,
      dy: 0,

      cells: [],

      maxCells: 4
    };

```

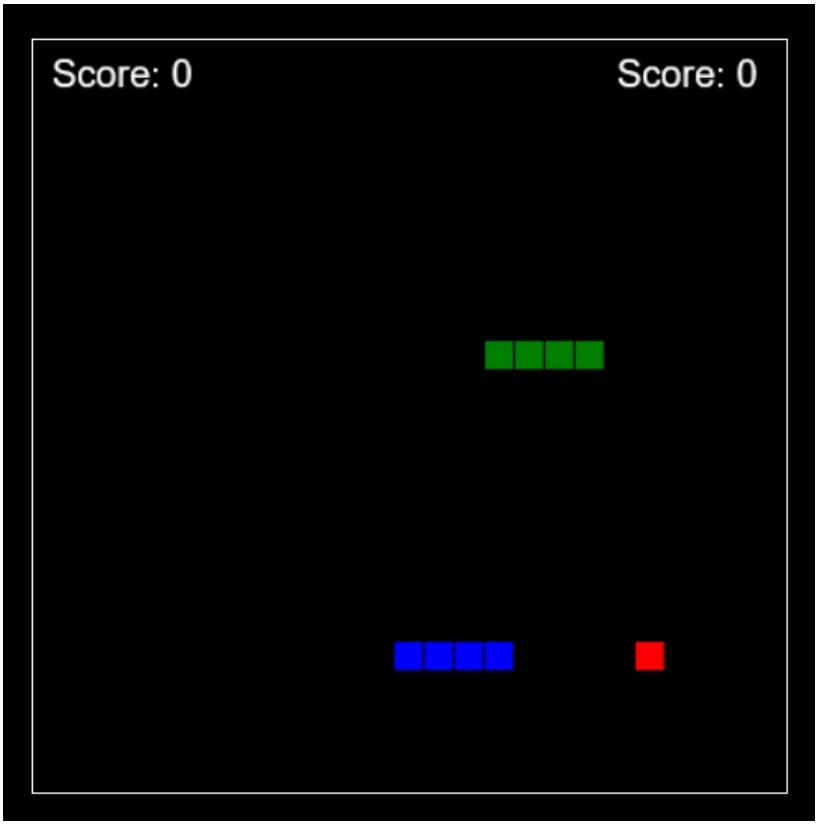
```
// А это — еда. Представим, что это красные яблоки.
var apple = {
  // Начальные координаты яблока
  x: 320,
  y: 320
};
// Делаем генератор случайных чисел в заданном диапазоне
function getRandomInt(min, max) {
  return Math.floor(Math.random() * (max - min)) + min;
}
// Игровой цикл — основной процесс, внутри которого будет всё происходить
function loop() {
  // Хитрая функция, которая замедляет скорость игры с 60 кадров в секунду до 15 (60/15 = 4)
  requestAnimationFrame(loop);
  // Игровой код выполнится только один раз из четырёх, в этом и суть замедления кадров, а пока переменная
  if (++count < 4) {
    return;
  }
  // Обнуляем переменную скорости
  count = 0;
  // Очищаем игровое поле
  context.clearRect(0, 0, canvas.width, canvas.height);
  // Двигаем змейку с нужной скоростью
  snake1.x += snake1.dx;
  snake1.y += snake1.dy;
  // Если змейка достигла края поля по горизонтали — продолжаем её движение с противоположной строны
  if (snake1.x < 0) {
    snake1.x = canvas.width - grid;
  }
  else if (snake1.x >= canvas.width) {
    snake1.x = 0;
  }
  // Делаем то же самое для движения по вертикали
  if (snake1.y < 0) {
    snake1.y = canvas.height - grid;
  }
  else if (snake1.y >= canvas.height) {
    snake1.y = 0;
  }
  // Продолжаем двигаться в выбранном направлении. Голова всегда впереди, поэтому добавляем её координаты
  snake1.cells.unshift({ x: snake1.x, y: snake1.y });
  // Сразу после этого удаляем последний элемент из массива змейки, потому что она движется и постоянно ос
  if (snake1.cells.length > snake1.maxCells) {
    snake1.cells.pop();
  }

  snake2.x += snake2.dx;
  snake2.y += snake2.dy;

  if (snake2.x < 0) {
    snake2.x = canvas.width - grid;
  }
  else if (snake2.x >= canvas.width) {
    snake2.x = 0;
  }

  if (snake2.y < 0) {
    snake2.y = canvas.height - grid;
  }
  else if (snake2.y >= canvas.height) {
    snake2.y = 0;
  }

  snake2.cells.unshift({ x: snake2.x, y: snake2.y });

  if (snake2.cells.length > snake2.maxCells) {
    snake2.cells.pop();
  }

```

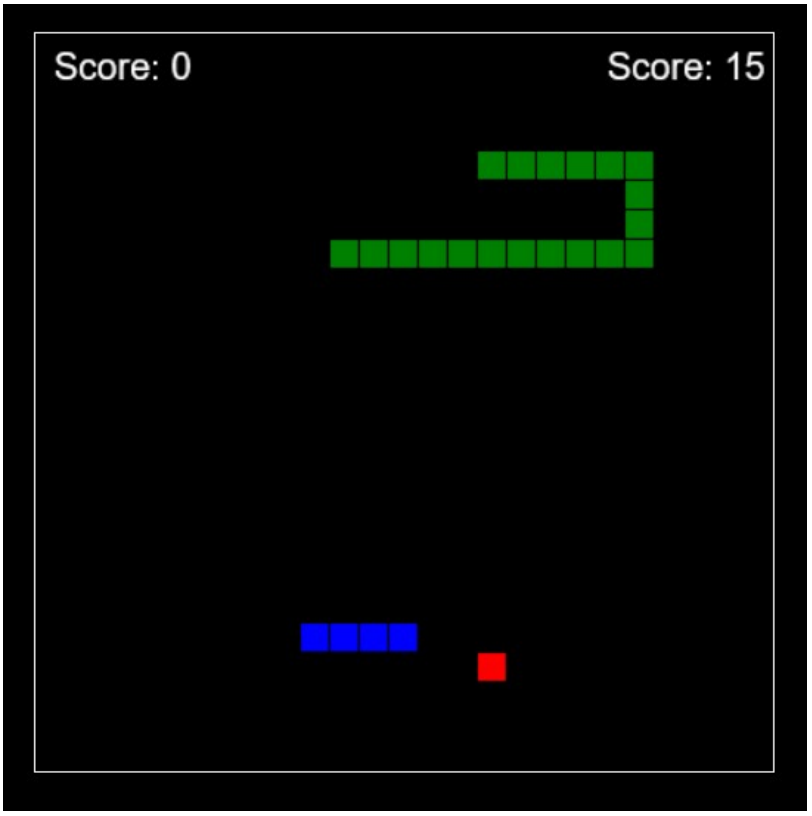
```
      // Рисуем еду — красное яблоко
      context.fillStyle = 'red';
      context.fillRect(apple.x, apple.y, grid - 1, grid - 1);
      // Одно движение змейки — один новый нарисованный квадратик
      context.fillStyle = 'green';
      // Обрабатываем каждый элемент змейки
      snake1.cells.forEach(function (cell, index) {
        // Чтобы создать эффект клеточек, делаем зелёные квадратики меньше на один пиксель, чтобы вокруг них
        context.fillRect(cell.x, cell.y, grid - 1, grid - 1);
        // Если змейка добралась до яблока...
        if (cell.x === apple.x && cell.y === apple.y) {
          // увеличиваем длину змейки
          snake1.maxCells++;
          // Рисуем новое яблочко
          // Помним, что размер холста у нас 400x400, при этом он разбит на ячейки — 25 в каждую сторону
          apple.x = getRandomInt(0, 25) * grid;
          apple.y = getRandomInt(0, 25) * grid;

          score1++; // увеличиваем количество набранных очков
        }
        // Проверяем, не столкнулась ли змея сама с собой
        // Для этого перебираем весь массив и смотрим, есть ли у нас в массиве змейки две клетки с одинаковы
//----------
        for (var i = index + 1; i < snake1.cells.length; i++) {
          // Если такие клетки есть — начинаем игру заново
          if (cell.x === snake1.cells[i].x && cell.y === snake1.cells[i].y) {
            // Задаём стартовые параметры основным переменным
            snake1.x = 160;
            snake1.y = 160;
            snake1.cells = [];
            snake1.maxCells = 4;
            snake1.dx = grid;
            snake1.dy = 0;

            score1 = 0; // сбрасываем количество набранных очков
          }


          if (snake2.x === snake1.cells[i].x && snake2.y === snake1.cells[i].y) {
            // Задаём стартовые параметры основным переменным
            snake2.x = 320;
            snake2.y = 320;
            snake2.cells = [];
            snake2.maxCells = 4;
            snake2.dx = -grid;
            snake2.dy = 0;

            score2 = 0; // сбрасываем количество набранных очков
          }

        }
      });

      context.fillStyle = 'blue';
      snake2.cells.forEach(function (cell, index) {

        context.fillRect(cell.x, cell.y, grid - 1, grid - 1);

        if (cell.x === apple.x && cell.y === apple.y) {

          snake2.maxCells++;

          apple.x = getRandomInt(0, 25) * grid;
          apple.y = getRandomInt(0, 25) * grid;

          score2++; // увеличиваем количество набранных очков
        }
//
```

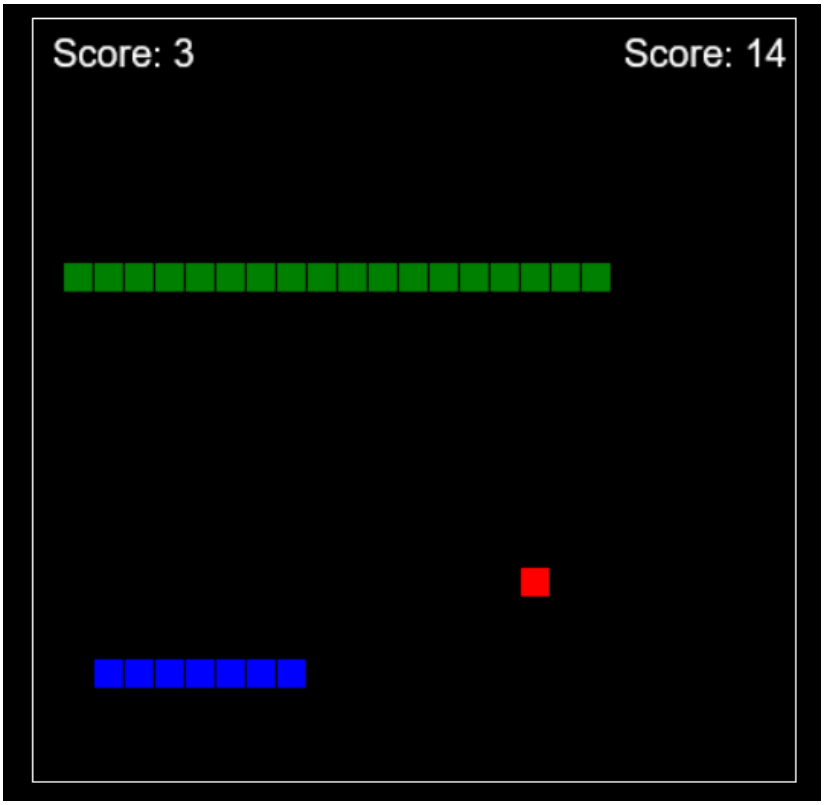
```
//----------
        for (var i = index + 1; i < snake2.cells.length; i++) {

          if (cell.x === snake2.cells[i].x && cell.y === snake2.cells[i].y)  {

            snake2.x = 320;
            snake2.y = 320;
            snake2.cells = [];
            snake2.maxCells = 4;
            snake2.dx = -grid;
            snake2.dy = 0;



            score2 = 0; // сбрасываем количество набранных очков
          }

          if (snake1.x === snake2.cells[i].x && snake1.y === snake2.cells[i].y) {
            // Задаём стартовые параметры основным переменным
            snake1.x = 160;
            snake1.y = 160;
            snake1.cells = [];
            snake1.maxCells = 4;
            snake1.dx = grid;
            snake1.dy = 0;

            score1 = 0; // сбрасываем количество набранных очков
          }

        }
      });



      // выводим количество набранных очков
      context.fillStyle = 'white';
      context.font = '20px Arial';
      context.fillText('Score: ' + score2, 10, 25);

      context.fillStyle = 'white';
      context.font = '20px Arial';
      context.fillText('Score: ' + score1, 310, 25);

    }
    // Смотрим, какие нажимаются клавиши, и реагируем на них нужным образом
    document.addEventListener('keydown', function (e) {
      // Дополнительно проверяем такой момент: если змейка движется, например, влево, то ещё одно нажа
      // Стрелка влево
      // Если нажата стрелка влево, и при этом змейка никуда не движется по горизонтали…
      if (e.which === 37 && snake1.dx === 0) {
        // то даём ей движение по горизонтали, влево, а вертикальное — останавливаем
        // Та же самая логика будет и в остальных кнопках
        snake1.dx = -grid;
        snake1.dy = 0;
      }
      // Стрелка вверх
      else if (e.which === 38 && snake1.dy === 0) {
        snake1.dy = -grid;
        snake1.dx = 0;
      }
```

```
      // Стрелка вправо
      else if (e.which === 39 && snake1.dx === 0) {
        snake1.dx = grid;
        snake1.dy = 0;
      }
      // Стрелка вниз
      else if (e.which === 40 && snake1.dy === 0) {
        snake1.dy = grid;
        snake1.dx = 0;
      }

      else if (e.which === 65 && snake2.dx === 0) {

        snake2.dx = -grid;
        snake2.dy = 0;
      }

      else if (e.which === 87 && snake2.dy === 0) {
        snake2.dy = -grid;
        snake2.dx = 0;
      }

      else if (e.which === 68 && snake2.dx === 0) {
        snake2.dx = grid;
        snake2.dy = 0;
      }

      else if (e.which === 83 && snake2.dy === 0) {
        snake2.dy = grid;
        snake2.dx = 0;
      }

    });
    // Запускаем игру
    requestAnimationFrame(loop);
  </script>
</body>

</html>
```

Результат выполнения программы






Score: 3
Score: 14