Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Теоретическая механика»**

**КУРСОВой проект**

**Реализация и визуализация игры «Змейка»**

по дисциплине «Математическое моделирование»

Выполнили

студенты гр.13632/1 <*подпись*> К.В. Демина

<*подпись*> В.Н. Малышева

<*подпись*> И.Г. Прохоренкова

Преподаватель <*подпись*> Д.В. Цветков

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 201\_\_ г.

Санкт-Петербург

2019

**Содержание**

Введение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .3

Задачи. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .3

Код JavaScript. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

Код HTML. . .. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .12

Результаты. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

Распределение задач, вывод и заключение. . . . . . . . . . . . . . . . . . . . . . . . . . . . .14

**Введение**

JavaScript - специальный язык программирования, разработанный для использования в браузерах. Представленная работа посвящена созданию игры "Змейка". В современном общество невозможно жить без всевозможных игр, разработанных на базе различных языков программирования. Поэтому важно понимать главные принципы и механизмы создания и визуализации подобных программ. В этом и заключается актуальность данной работы. Целью является изучение основных принципов создания игры "Змейка". Объект исследования – процесс использования языков программирования для создания игры. Предмет исследования – программные инструменты JavaScript и HTML.

**Задачи**

Реализовать и визуализировать игру “Змейка” с рядом существенных отличий:

* Существует три вида еды: красная - увеличивает длину змейки, желтая - уменьшает длину змейки, синяя - ничего не меняет.
* При достижении счета в 3 и 5 очков скорость увеличивается.
* Если счет больше 5, то змейка имеет возможность проходить сквозь стены.

**Код JavaScript**

window.addEventListener("load", main\_code, false);

function main\_code(){

 var ctx = SnakeS.getContext("2d");

 var h = SnakeS.height;

 var w = SnakeS.width;

 var n = 45;

 var intervalID;

 var direction = "left";

 var ms =200;

 var score = 0;

 var wCode = 87;

 var aCode = 65;

 var sCode = 83;

 var dCode = 68;

 var upCode = 38;

 var leftCode = 37;

 var downCode = 40;

 var rightCode = 39;

 var pause = false;

 function get\_mouse\_coords(e){

 var m = {};

 var rect = SnakeS.getBoundingClientRect();

 m.x = e.clientX - rect.left;

 m.y = e.clientY - rect.top;

 return m;

 }

 SnakeS.onclick = function(e){

 var m = get\_mouse\_coords(e);

 console.log(m.x, m.y);

 if ((m.x>(w/2-95))&&(m.x<(w/2+95))&&(m.y>(h/2+30))&&(m.y<(h/2+70))) {

 console.log(1);

 location.reload();

 }

 }

 var sl=1;

 var nofood = false;

 var nofoodno = false;

 var x = w/n;

 var y = h/n;

 console.log(x,y);

 function box(){

 this.conteins = "empty";

 }

 function Snake(){

 this.head = null;

 this.tail = null;

 this.body = [];

 this.length = 1;

 }

 var field = [];

 for (var i=0; i<n; i++) {

 var m = [];

 for (var j=0; j<n; j++){

 m.push(new box());

 }

 field.push(m);

 }

 var snake = new Snake();

 function finish() {

 ctx.textAlign = 'center';

 ctx.font = '30px Georgia';

 pause = true;

 console.error("Вы проиграли");

 clearInterval(intervalID);

 ctx.fillStyle = 'rgb(180, 234, 255)';

 ctx.fillRect(0.25\*w, 0.25\*h, 0.5\*w, 0.5\*h);

 ctx.strokeStyle = 'black';

 ctx.strokeRect(0.25\*w, 0.25\*h, 0.5\*w, 0.5\*h);

 ctx.fillStyle = 'black';

 ctx.fillText('Вы проиграли!', w/2, h/2-40);

 ctx.fillText('Ваш счет: '+score, w/2, h/2);

 ctx.fillStyle = 'rgb(14, 151, 184)';

 ctx.fillRect(w/2-95, h/2+30, 190, 40);

 ctx.strokeRect(w/2-95, h/2+30, 190, 40);

 ctx.fillStyle = 'black';

 ctx.fillText('Сыграть еще', w/2, h/2+60);

 console.info('rtr');

 }

 function CreateItem(item){

 var i=Math.round(Math.random()\*(n - 1));

 var j=Math.round(Math.random()\*(n - 1));

 while (field [i][j].conteins != "empty"){

 var i=Math.round(Math.random()\*(n - 1));

 var j=Math.round(Math.random()\*(n - 1));

 }

 field [i][j].conteins = item;

 }

 function Init(){

 var i=Math.round(Math.random()\*(n - 1));

 var j=Math.round(Math.random()\*(n - 1));

 field [i][j].conteins = "food";

 var i=Math.round(Math.random()\*(n - 1));

 var j=Math.round(Math.random()\*(n - 1));

 while (field [i][j].conteins == "food"){

 var i=Math.round(Math.random()\*(n - 1));

 var j=Math.round(Math.random()\*(n - 1));

 }

 field [i][j].conteins = "snake";

 snake.head = {x: i, y: j};

 snake.tail = {x: i, y: j};

 snake.body.push({x: i, y: j}) ;

 console.log(direction, snake.body, snake.head);

 CreateItem('nofood');

 }

 function Draw() {

 for (var i=0; i<n; i++) {

 for (var j=0; j<n; j++){

 if (field[i][j].conteins == "empty") {

 ctx.fillStyle = 'white';

 ctx.fillRect(x\*i, y\*j, x, y);

 }

 if (field[i][j].conteins == "food") {

 ctx.fillStyle = 'red';

 ctx.fillRect(x\*i, y\*j, x, y);

 ctx.strokeStyle = 'black';

 ctx.strokeRect(x\*i, y\*j, x, y);

 }

 if (field[i][j].conteins == "snake") {

 ctx.fillStyle = 'green';

 ctx.fillRect(x\*i, y\*j, x, y);

 ctx.strokeStyle = 'black';

 ctx.strokeRect(x\*i, y\*j, x, y);

 }

 if (field[i][j].conteins == "nofood") {

 ctx.fillStyle = 'blue';

 ctx.fillRect(x\*i, y\*j, x, y);

 ctx.strokeStyle = 'black';

 ctx.strokeRect(x\*i, y\*j, x, y);

 }

 if (field[i][j].conteins == "nofoodno") {

 ctx.fillStyle = 'yellow';

 ctx.fillRect(x\*i, y\*j, x, y);

 ctx.strokeStyle = 'black';

 ctx.strokeRect(x\*i, y\*j, x, y);

 }

 }

 }

 document.getElementById('scre').innerHTML = score;

 }

 window.onkeydown = function(evt) {

 evt = evt || window.event;

 var charCode = evt.keyCode || evt.which;

 console.log(charCode);

 if (((charCode == wCode)||(charCode == upCode)) && direction != "down") {

 direction = "up";

 }

 if (((charCode == sCode)||(charCode == downCode)) && direction != "up") {

 direction = "down";

 }

 if (((charCode == aCode)||(charCode == leftCode)) && direction != "right") {

 direction = "left";

 }

 if (((charCode == dCode)||(charCode == rightCode)) && direction != "left") {

 direction = "right";

 }

 }

 function CheckLength() {

 if ((score == 3) && (sl != 2)){

 sl = 2;

 ms = ms-100;

 clearInterval(intervalID);

 intervalID = setInterval(control, ms);

 return ;

 }

 if ((score == 5 ) && (sl != 3)){

 sl = 3;

 ms = ms-50;

 clearInterval(intervalID);

 intervalID = setInterval(control, ms);

 }

 }

 function Calcul(){

 var next

 try {

 if (direction == "left") {

 next = {x: snake.head.x-1, y: snake.head.y}

 }

 if (direction == "right") {

 next = {x: snake.head.x+1, y: snake.head.y}

 }

 if (direction == "up") {

 next = {x: snake.head.x, y: snake.head.y-1}

 }

 if (direction == "down" && snake.length == 1) {

 next = {x: snake.head.x, y: snake.head.y+1}

 }

 if (direction == "down" && snake.length > 1) {

 next = {x: snake.head.x, y: snake.head.y}

 }

 if (score > 5) {

 if (next.x<0) {

 next.x = n-1

 console.log('TP')

 }

 if (next.y<0) {

 next.y = n-1

 console.log('TP')

 }

 if (next.x>n-1) {

 next.x = 0

 console.log('TP')

 }

 if (next.y>n-1) {

 next.y = 0

 console.log('TP')

 }

 } else {

 if (next.x<0) {

 finish();

 return;

 }

 if (next.y<0) {

 finish();

 return;

 }

 if (next.x>n-1) {

 finish();

 return;

 }

 if (next.y>n-1) {

 finish();

 return;

 }

 }

 if (field[next.x][next.y].conteins == "snake") {

 finish();

 return;

 }

 if (field[next.x][next.y].conteins == "food") {

 snake.body.push({x:next.x, y:next.y});

 snake.head = {x:next.x, y:next.y};

 field[next.x][next.y].conteins = "snake";

 CreateItem ('food');

 score++;

 }

 if (field[next.x][next.y].conteins == "nofoodno") {

 snake.body.push({x:next.x, y:next.y});

 snake.head = {x:next.x, y:next.y};

 field[next.x][next.y].conteins = "snake";

 field[snake.body[0].x][snake.body[0].y].conteins = "empty";

 snake.body.shift();

 if(snake.body.length!=1)

 field[snake.body[0].x][snake.body[0].y].conteins = "empty";

 snake.body.shift();

 score++;

 nofoodno=false;

 }

 if (field[next.x][next.y].conteins == "nofood") {

 snake.body.push({x:next.x, y:next.y});

 snake.head = {x:next.x, y:next.y};

 field[next.x][next.y].conteins = "snake";

 field[snake.body[0].x][snake.body[0].y].conteins = "empty";

 snake.body.shift();

 score++;

 nofood=false;

 }

 if (field[next.x][next.y].conteins == "empty") {

 snake.body.push({x:next.x, y:next.y});

 snake.head = {x:next.x, y:next.y};

 field[next.x][next.y].conteins = "snake";

 field[snake.body[0].x][snake.body[0].y].conteins = "empty";

 snake.body.shift();

 }

 } catch (err){

 console.error("Что то пошло не так; X=", next.x, "Y=", next.y);

 clearInterval(intervalID);

 }

 if (((score+1) % 5 == 0)&&(!nofood)){

 CreateItem ('nofood');

 nofood=true;

 }

 if (((score+1) % 8 == 0)&&(!nofoodno)){

 CreateItem ('nofoodno');

 nofoodno=true;

 }

 }

 function control (){

 if (!pause)

 Calcul ();

 if (!pause)

 Draw();

 CheckLength();

 }

 Init();

 Draw ();

 intervalID = setInterval(control, ms);

}

**Код HTML**

<!DOCTYPE html>

<head>

 <title>SNAKE</title>

 <meta charset='utf-8'>

 <script src="zmeika.js"></script>

</head>

<body>

<canvas id="SnakeS" width="600" height="600" style="border: 2px solid #000000"></canvas>

<tr>

<pre><span style="font-size: 30px;">Ваш счет:</span> <span id = 'scre' style="font-size: 30px;">0</span></pre>

</tr>

<p></p>

</body>

</html>

**Результаты**

С помощью методов объектно-ориентированного программирования был составлен простой код, позволяющий пользователю сыграть в игру “Змейка”. Управление змеей осуществляется клавишами со стрелками. На скриншотах можно ознакомиться с полным интерфейсом игры.







**Вывод**

Таким образом, реализация и визуализация игры «Змейка» оказались успешными, удалось выполнить поставленные задачи. Каждым студентом были освоены новые методы, использованные в данном проекте, которые упрощают работу с объектно-ориентированным программированием. В итоге мы получили новый и полезный опыт в работе над подобными заданиями и составлением курсовых работ, который будет существовать в дальнейшем.

**Заключение**

В заключении хотелось бы сказать, что JavaScript – это действительно функциональный и актуальный язык, позволяющий выполнять большой спектр задач, включающий и разработку простейших браузерных игр.

**ФОРМА ЗАДАНИЯ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА (КУРСОВОЙ РАБОТЫ)**

Санкт-Петербургский политехнический университет Петра Великого

**ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

студентке группы 13632/1 Деминой Ксении Вячеславовной

студентке группы 13632/1 Прохоренковой Ирине Георгиевне

студентке группы 13632/1 Малышевой Веронике Николаевной

***1. Тема проекта:*** Реализация и визуализация игры «Змейка»

***2. Срок сдачи студентом законченного проекта:***  «\_\_\_».\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

***3. Исходные данные к проекту***: курс лекций по математическому моделированию за первый и второй семестры

***4. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения.

Примерный объём пояснительной записки 16 страниц печатного текста.

***5. Перечень графического материала*** (с указанием обязательных чертежей и плакатов): отсутствует

***6. Консультанты***

***7. Дата получения задания***: «\_\_\_».\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 *(подпись) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 *(подпись студента) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 *(подпись студента) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 *(подпись студента) (инициалы, фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_*(дата)*