Санкт-Петербургский политехнический университет Петра Великого

Институт прикладной математики и механики

**Кафедра «Теоретическая механика»**

**КУРСОВОЙ ПРОЕКТ**

**Реализация и визуализация игры “MegaBall2D”**

по дисциплине «Математическое моделирование»

Выполнил

студент гр.13632/1 Грешников Павел

Руководитель

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2019 г.

Санкт-Петербург

2019

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  Постановка задачи. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 3  3 |
| 1. Реализация. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| 1.1. Описание кода HTML и JavaScript. . . . . . . . . . . . . . . . . . . . . . . . . . . .  Результаты. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5  8 |
| Вывод и заключение. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |

**Введение**

JavaScript –язык программирования, который позволяет выполнять задачи различной направленности и сложности. В данной работе будет создана игра «MegaBall2D». Объект исследования – процесс использования языков программирования для создания простейшей игры.   
Предмет исследования – программные инструменты языков JavaScript и HTML, которые применяются при разработке простейших игр.

**Постановка задачи**

Реализовать и визуализировать игру “MegaBall2D” со следующими правилами:

1. В начале игроку требуется «выстрелить» правым мячиком в левый, оттянув его компьютерной мышью.

2. Координаты второго шарика выбираются случайным образом на правой половине поля.

3. При выборе траектории полета шарика необходимо учитывать, что в игре присутствует гравитация, отскоки от стен и стенка, мешающая мячику пролететь.

4. С каждым новым уровнем радиус шариков уменьшается (повышается сложность).

5. Если игрок бросил шарик так, что он точно не столкнется с другим, игра начнется с первого уровня.

**Реализация**

Рассматриваем шарик как окружность с центром в точке с координатами (х,у), которые меняются с течением времени так же, как и у тела, брошенного под углом к горизонту. Также в каждый момент времени проверяем расстояние между шариками: если оно меньше двух радиусов, значит шарики столкнулись, в таком случае начинается новый уровень.

**Описание кода HTML и JavaScript**

Создаем canvas:

<canvas id="example\_canvas" width="1500" height="500" style="border: 1px solid #000000"></canvas>

Считываем координаты мыши и при нажатии на ЛКМ внутри шарика начинаем его двигать:

function get\_mouse\_coords(e){

var m={}; // объявление пустого объекта

var rect = example\_canvas.getBoundingClientRect();

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

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

return m;

}

example\_canvas.onmousedown=function(e){

m=get\_mouse\_coords(e);

var dx=m.x-x;

var dy=m.y-y;

padding.x=dx;

padding.y=dy;

if (Math.sqrt(dx\*dx+dy\*dy)<r) {

x0=m.x;

y0=m.y;

flag = true;

ctx.fillStyle=TouchColor;

draw();

}

}

star();

example\_canvas.onmouseup=function(e){

flag = false;

ctx.fillStyle=defaultColor;

draw();

m=get\_mouse\_coords(e);

var dx=m.x-x;

var dy=m.y-y;

padding.x=dx;

padding.y=dy;

if (Math.sqrt(dx\*dx+dy\*dy)<r) {

x\_k=m.x;

y\_k=m.y;

speed();

start=true;

}

}

example\_canvas.onmousemove = function(e){

if (flag) {

m=get\_mouse\_coords(e);

x=m.x-padding.x;

y=m.y-padding.y;

draw();

ctx.beginPath();

ctx.moveTo(Math.round(x),Math.round(y));

ctx.lineTo(Math.round(x0),Math.round(y0));

ctx.stroke();

}

}

После отпускания ЛКМ, задаем шару скорость:

function speed(){

vx=-(x\_k-x0)\*2;

vy=-(y\_k-y0)\*2;

}

После этого, описываем его движение:

function phys(){

if (vy\*vy<(y-YSL-2\*r)) console.log('imp');

vy+=100\*dt;

x=x+vx\*dt;

y=y+vy\*dt+100\*dt\*dt/2;

if(x>=w-(r)) {

vx=(-0.9)\*Math.abs(vx);

}

else

if(x<=(r))

vx=0.9\*Math.abs(vx);

if(y>=h-(r)){

vy=-0.9\*Math.abs(vy);

if (Math.abs(vy)<10){

vy=-150\*dt;

}

}

else

if(y<=(r))

vy=0.9\*Math.abs(vy);

if ((x >w/2-5-r)&&(x<w/2)&&(y>h\*1/3-r)) {

vx=-0.9\*Math.abs(vx);

}

if ((x>w/2)&&(x<w/2+5+r)&&(y>h\*1/3-r)){

vx=0.9\*Math.abs(vx);

}

if ((x>w/2-5)&&(x<w/2+5)&&(y<h/3-r+5)&&(y>h/3-r)){

vy=-1000\*Math.abs(vx);

}

}

Задаем координаты второго шара:

function star(){

XSL=Math.random()\*(700-2\*(r))+w/2+r;

YSL=Math.random()\*(h-2\*(r))+r;

}

Рисуем шары и стенку между ними:

function draw() {

ctx.clearRect(0,0,w,h);

ctx.beginPath();

ctx.arc(x,y,r,0,2\*Math.PI);

ctx.arc(XSL,YSL,r,0,2\*Math.PI);

ctx.moveTo(Math.round(x0),Math.round(y0));

ctx.lineTo(x,y);

ctx.font = 'bold 60px sans-serif';

ctx.strokeText("CURRENT LEVEL:", 800, 100);

ctx.strokeText(lvl, 1330, 100);

ctx.fill();

ctx.beginPath();

ctx.moveTo(w/2-5,h);

ctx.lineTo(w/2-5,h/3);

ctx.moveTo(w/2+5,h/3);

ctx.lineTo(w/2+5,h);

ctx.moveTo(w/2+5,h/3);

ctx.lineTo(w/2-5,h/3);

ctx.stroke();

}

Проверяем, не столкнулись ли шары:

function confront(){

if (((x-XSL)\*(x-XSL)+(y-YSL)\*(y-YSL)<=4\*(r)\*(r))&&(start)){

levelup();

}

}

Повышение уровня:

function levelup(){

x=250;

y=h/2;

vx=0;

vy=0;

lvl++;

r=40\*10/(10+lvl);

star();

start=false;

}

Проверка на невозможность прохождения уровня и сброс уровня до первого:

function imposible() {

if ((start)&&((vy\*vy<(y-YSL-2\*r))||((x<w/2)&&(vy\*vy<(2\*y/3+r))))) {

imp++;

}

if (imp>40) {

imp=0;

restart();

}

}

}

function restart(){

x=250;

y=h/2;

vx=0;

vy=0;

lvl=1;

r=40\*10/(10+lvl);

star();

start=false;

}

Задаем движение шара с течением времени:

setInterval(control,1000/fps);

function control() {

if (start)

phys();

draw();

confront();

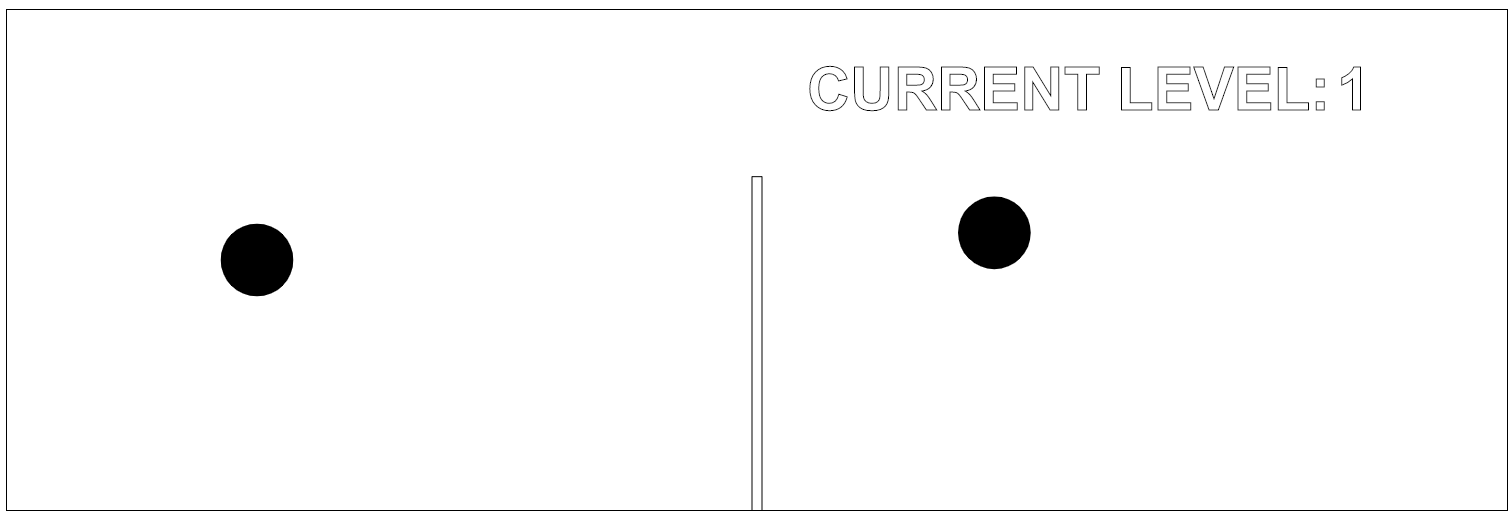
imposible();

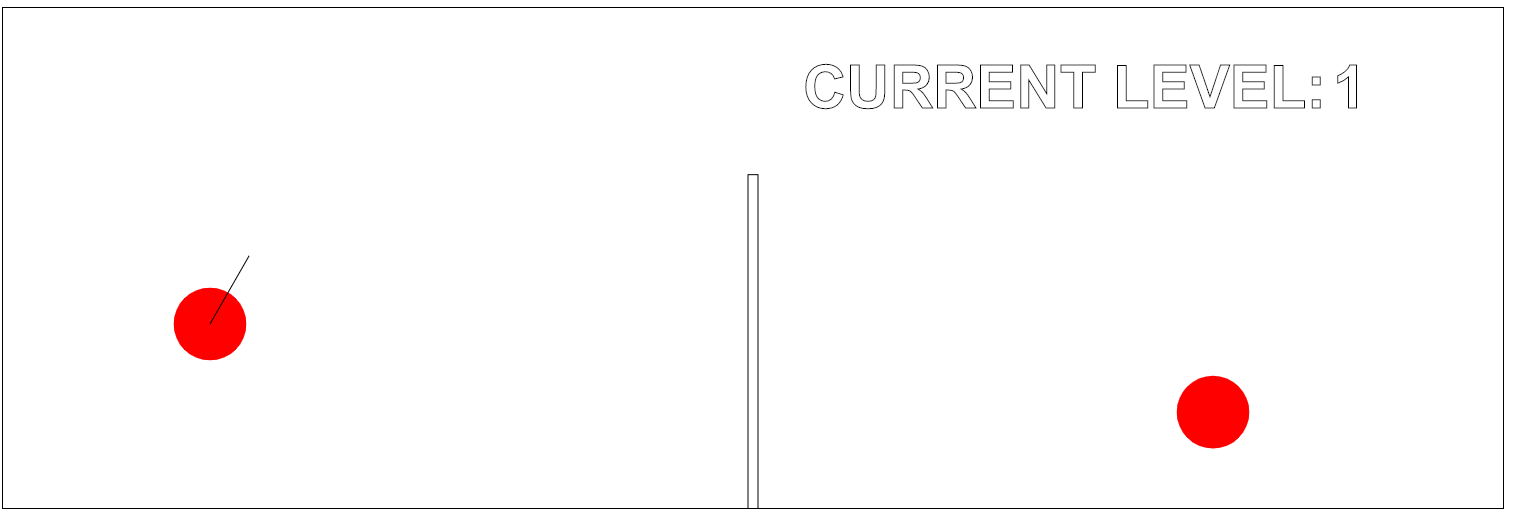
}

**Результаты**

Была составлена игра «MegaBall2D» на языке программирования JavaScript.

На рисунках, представленных ниже, можно ознакомиться с игровым процессом и интерфейсом игры. В левой половине поля расположен шар, который игроку предлагается «бросить» в шар, находящийся в правой половине.





**Вывод**

Таким образом, реализация и визуализация игры «MegaBall2D» оказались успешными. Можно сказать, что JavaScript - подходящий язык программирования для визуализации различных несложных для реализации игр.

**Форма задания на выполнение   
курсового проекта**

Санкт-Петербургский политехнический университет Петра Великого

**ЗАДАНИЕ**

**НА ВЫПолнение курсового проекта**

студенту группы 13632/1 Грешникову П. С.

***1. Тема проекта:*** Реализация и визуализация игры «MegaBall2D»

***2. Срок сдачи студентом законченного проекта***

***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 9 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)*

\_\_\_\_\_\_\_\_\_\_\_\_*(дата)*