Департамент образования города Москвы

Государственное бюджетное общеобразовательное учреждение города Москвы

Гимназия № 1505  
«Московская городская педагогическая гимназия-лаборатория»

ДИПЛОМНОЕ ИССЛЕДОВАНИЕ на тему:

**Эффект параллакса и его использование**

Автор:

Ерёмин Игорь Артурович,

ученица 10А класса

Руководитель:

Наумов Алексей Леонидович

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Рецензент:

подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2018 год

**Оглавление**

Введение

Глава 1. Эффект параллакса.

Глава 2. Работа с p5.js.

2.1. Функция createCanvas()

2.2. Библиотека WEBGL

2.3. Функция map()

2.4. Функция camera()

2.5. Функция translate()

Глава 3. Программа

3.1 Перемещение

3.2 Вращение

Заключение

Список источников

**Глава 1. Эффект параллакса.**

1. 1. Основы геометрии

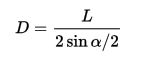
Параллакс — изменение видимого положения объекта относительно удалённого фона в зависимости от положения наблюдателя.

Параллакс используется в геодезии и астрономии для измерения расстояния до удалённых объектов. На явлении параллакса основано бинокулярное зрение.

Для понимания явления параллакса необходимо знать из геометрии, как связаны значения углов между сторонами треугольника и их длинами.

Если известна длина основания треугольника и величина углов, прилегающих к нему, то можно узнать значение длин двух сторон.

Сделать это можно с помощью математической формулы, связывающей значения длин сторон и величин углов, лежащих напротив них. Это формула: a/(sinA) = b/(sinB)= c/(sinC) Если дан треугольник ABC и известно его основание с прилегающими углами, то через формулу: AB / sin(BCA) = BC / sin(BAC) = CA / sin(ABC) можно найти все остальное. Таким образом, если нам известно перемещение тела и углы при перемещений, мы можем узнать начальное и конечное расстояние до тела при помощи теоремы синусов. Существует формула:



где L - расстояние между точками наблюдения, D - расстояние до объекта, а уголь Альфа - угол между начальными и конечными точками тела.

1.2. Фотография

Самым известным видом фотографического параллакса можно считать бинокулярный параллакс. Самым явный способ наблюдать бинокулярный параллакс - поднести к глазам палец и по очереди закрывать каждый глаз, при этом можно заметить, что угол зрения на объект меняется.

В объектив мы видим изображение под одним углом зрения, но на самом деле фотография получится с немного другим углом, так как есть разница в расстоянии между объективом и видоискателем.

**Глава 2. Работа с P5.js.**

2.1. createCanvas()

Чтобы начать работать с программой openProcessing и быть готовым выводить какие-либо данные, фигуры на экран, необходимо выделить место, где все это будет происходить. Функция «createCanvas()» создаёт окно программы, в пределах которого в будущем можно будет, что-то выводить. Размеры окна задаются переменными «Width» и «Height», которые указываются в скобках после функции. Если размеры отсутствуют, то размер получится 100 на 100 пикселей. Функция «createCanvas()» может вызываться только один раз за всю программу, иначе произойдет сбой.

2.2. WEBGL

В процессе создания приложений и различимых программ может потребоваться работа с различными 3D объектами или работа в трёхмерном пространстве. Для этого потребуется указать после параметров в функции createCanvas() программную библиотеку WEBGL. Эта библиотека позволяет создавать интерактивную 3D-графику. Библиотека построена на основе OpenGL.

2.3. map()

Функция «map()» позволяет перевести число из одного диапазона значений в другой. Эта функция зависит от 6 параметров, указанных в скобках после неё. Первый параметр указывает на значение, которое предстоит переработать. Второй и третий параметр указывают на текущий диапазон значений. Далее идут два параметра указывающие на желаемый диапазон. Следующий параметр не так важен, как остальные. Он представляет собой переменную Boolean, которая содержит значения из созданного диапазона.

2.4. camera()

Функция camera() играет роль глаз человека в openProcessing. Она позволяет «смотреть» в разные точки пространства, используя девять параметров, отвечающих за вектор, куда направлена камера, за положение камеры в пространстве и за положение центра объекта относительно камеры. Каждый параметр представлен для оси X, оси Y, оси Z. Если не задавать параметров камере, то она будет использована в таком виде:

camera() = camera(windowWidth/2.0, windowHeight/2.0, (windowHeight/2.0) / tan(PI\*30.0 / 180.0), windowWidth/2.0, windowHeight/2.0, 0, 0, 1, 0)

2.5. translate()

Функция translate() перемещает любой объект на заданную координату в пространстве. Она зависит от трёх компонентов: X - перемещение влево и вправо на экране, Y - перемещение вверх и вниз на экране, Z - перемещение вперёд и назад от экрана. Центр экрана имеет значения (0,0,0).

**Глава 3. Программа с виртуальной реальностью.**

3.1. Эффект параллакса

*var img1;*

*var img2;*

*var img3;*

*var img4;*

*var ySpeed;*

*var xSpeed;*

*var a, b, c, d, e, x, y, z;*

*function preload() {*

*img1 = loadImage("1.png");*

*img2 = loadImage("2.png");*

*img3 = loadImage("3.png");*

*img4 = loadImage("4.png");*

*img5 = loadImage("5.png");*

*img6 = loadImage("6.png");*

*img7 = loadImage("7.png"); }*

*function setup() {*

*createCanvas(windowWidth, windowHeight);*

*imageMode(CENTER);*

*image(img1, 0, 0,windowWidth+400,windowHeight);*

*image(img2, 0, 0,windowWidth+400,windowHeight);*

*image(img3, 0, 0,windowWidth+400,windowHeight);*

*image(img4, 0, 0,windowWidth+400,windowHeight);*

*image(img1, 0, 0,windowWidth+400,windowHeight);*

*a = windowWidth/2;*

*b = windowHeight/4;*

*c = windowWidth/2;*

*d = windowHeight/4;*

*e = windowWidth/2;*

*f = windowHeight/2;*

*g = windowWidth/2;*

*h = windowHeight/2;*

*i = windowWidth/2;*

*j = windowHeight/2;*

*k = windowWidth/2;*

*l = windowHeight/3+windowHeight/5;*

*m = windowWidth/2;*

*n = windowHeight/4; }*

*function draw(){*

*background(255);*

*xSpeed = rotationY - pRotationY;*

*ySpeed = rotationX - pRotationX;*

*a = a - (xSpeed\*3)/2;*

*b = b - (ySpeed\*3)/2;*

*image(img1, a, b,windowWidth+700,windowHeight+850);*

*c = c - (xSpeed\*4)/2;*

*d = d - (ySpeed\*4)/2;*

*image(img2, c, d,windowWidth+700,windowHeight+850);*

*e = e - (xSpeed\*5)/2;*

*f = f - (ySpeed\*5)/2;*

*image(img3, e, f,windowWidth+400,windowHeight);*

*g = g - (xSpeed\*7)/2;*

*h = h - (ySpeed\*7)/2;*

*image(img4, g, h,windowWidth+400,windowHeight);*

*i = i - (xSpeed\*8)/2;*

*j = j - (ySpeed\*8)/2;*

*image(img5, i, j,windowWidth+400,windowHeight);*

*k = k - (xSpeed\*9)/2;*

*l = l - (ySpeed\*9)/2;*

*image(img6, k, l,windowWidth+700,windowHeight+100);*

*m = m - (xSpeed\*10)/2;*

*n = n - (ySpeed\*10)/2;*

*image(img7, m, n,windowWidth+700,windowHeight+1250)}*

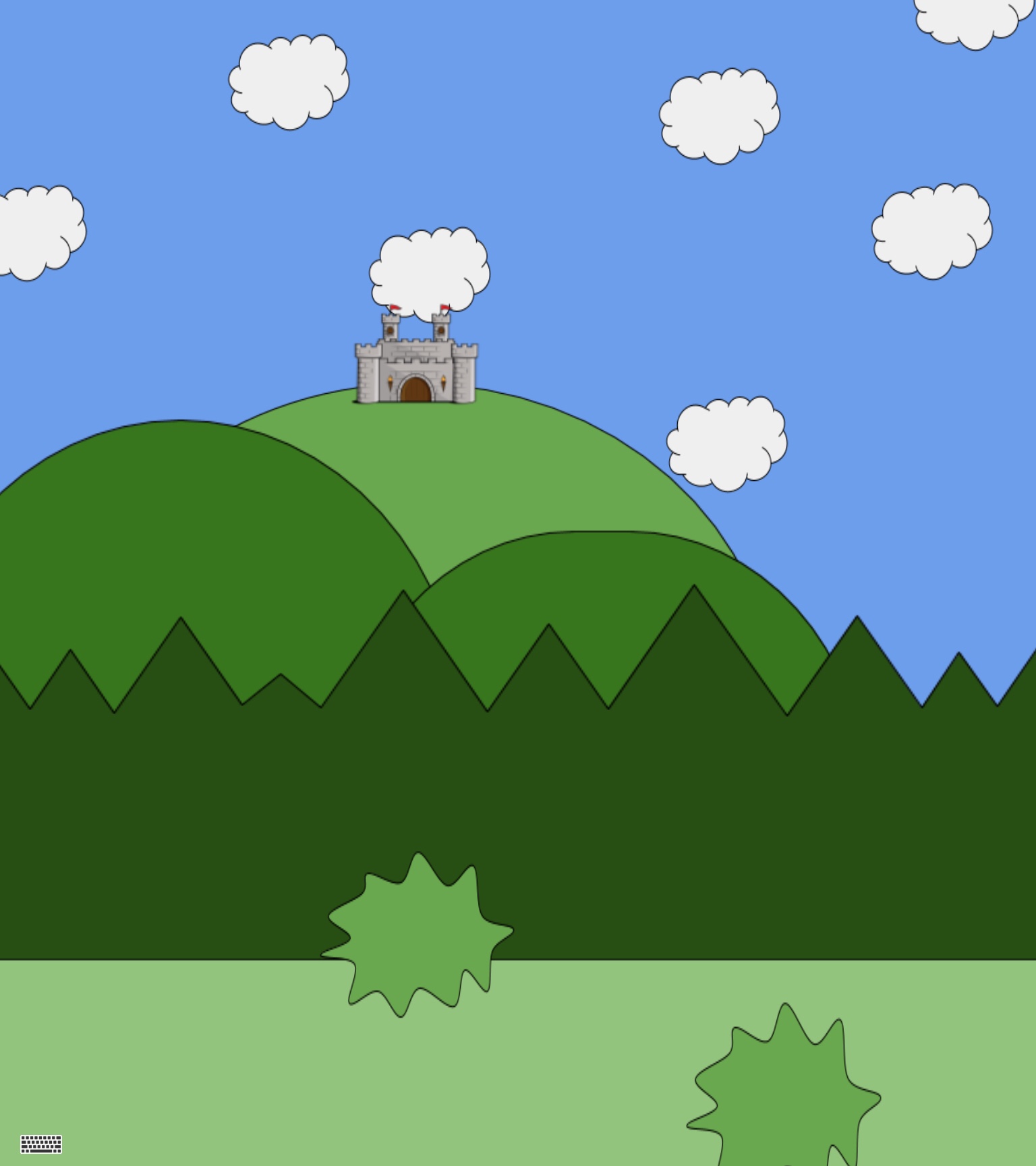
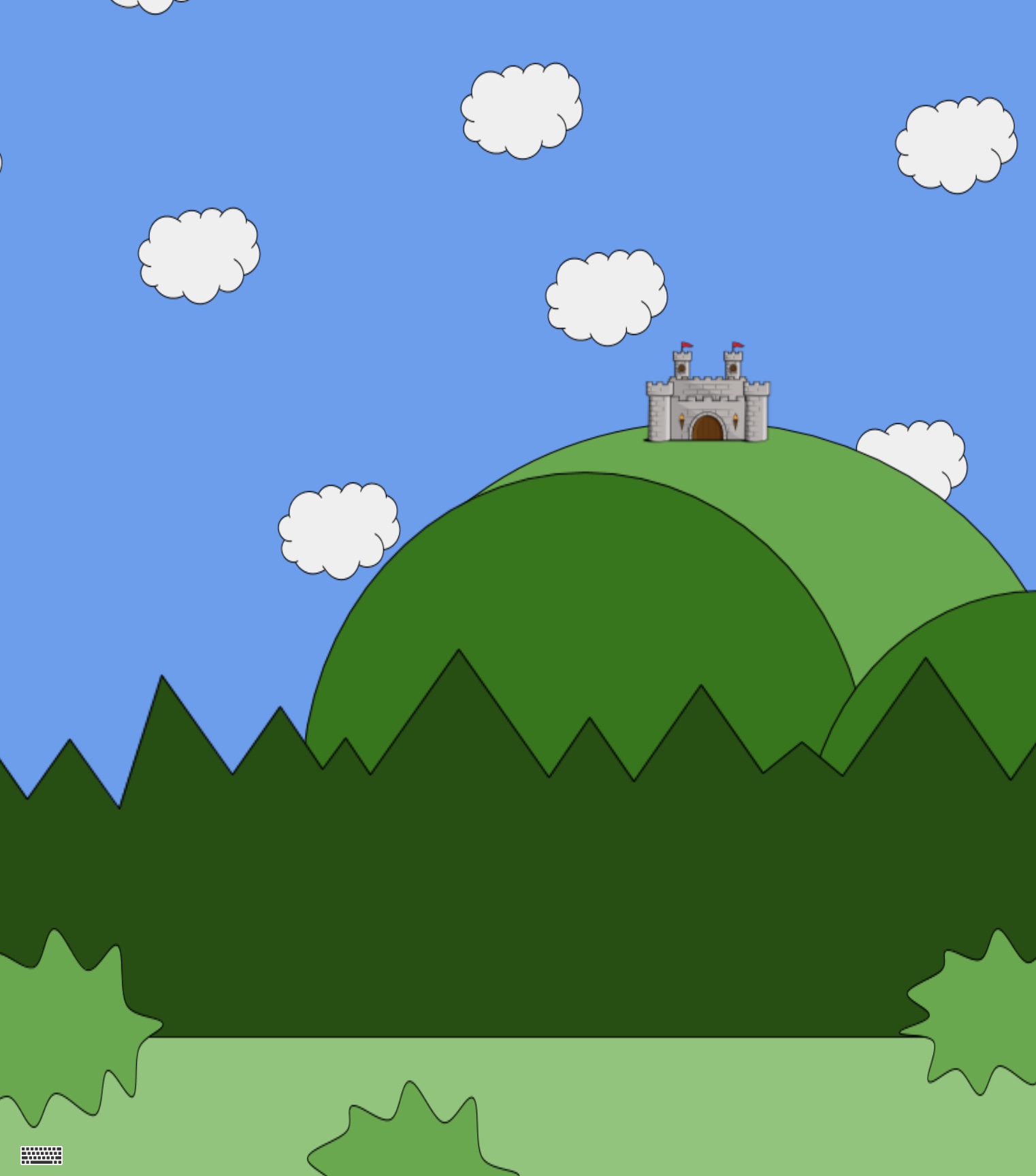
Эта программа помогает увидеть эффект параллакса на любом девайсе. Во время создания было начато знакомство с языком, была изучена работа с загрузкой картинок и изменением положения устройства, а именно функции *rotationX* и *rotationY*.

Рис. 1 «Скриншот программы» Рис. 2 «Скриншот программы»

3.2. Перемещение

*var camX, camY, camZ;*

*var k = 300;*

*var m = 300;*

*var n = 300;*

*function setup() {*

*createCanvas(windowWidth, windowHeight, WEBGL);}*

*function draw(){*

*background(90);*

*camX = map(m, 0, width,-200,200);*

*camY = map(k, 0, width,-200,200);*

*camZ = map(n, 0, width,-200,200);*

*camera(camX,camY+125,camZ+125,0,camY+125, сamZ+125,0,-1,0);*

*fill(0,255,0);*

*strokeWeight(0.5);*

*box(20);}*

*function keyPressed(){*

*if (key === "1"){*

*k = k + 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}*

*else if (key === "2"){*

*k = k - 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}*

*else if (key === "3"){*

*m = m + 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}}*

*else if (key === "4"){*

*m = m - 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}}*

*else if (key === "5"){*

*n = n + 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}}*

*else if (key === "6"){*

*n = n - 50;*

*println('camX = ' + camX + ‘ camY = ' + camY + ‘ camZ = ' + camZ );}}}*

Эта программа позволяет создать имитацию движения с помощью функции камера. Благодаря изменению значений кнопками можно перемещаться в виртуальном пространстве относительно объекта. Объектом в данном случае является куб, размеры которого 50х50 пикселей.

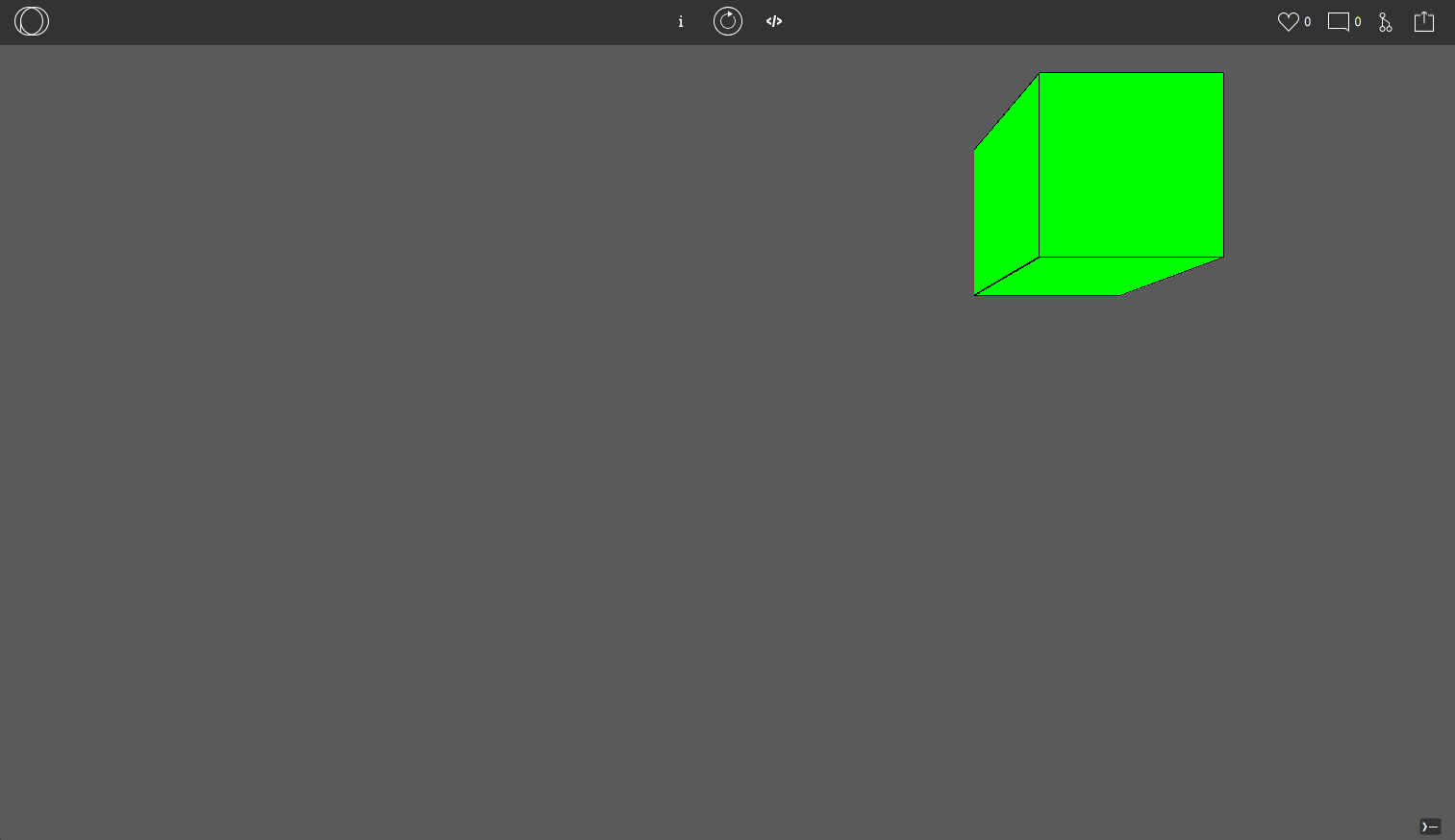


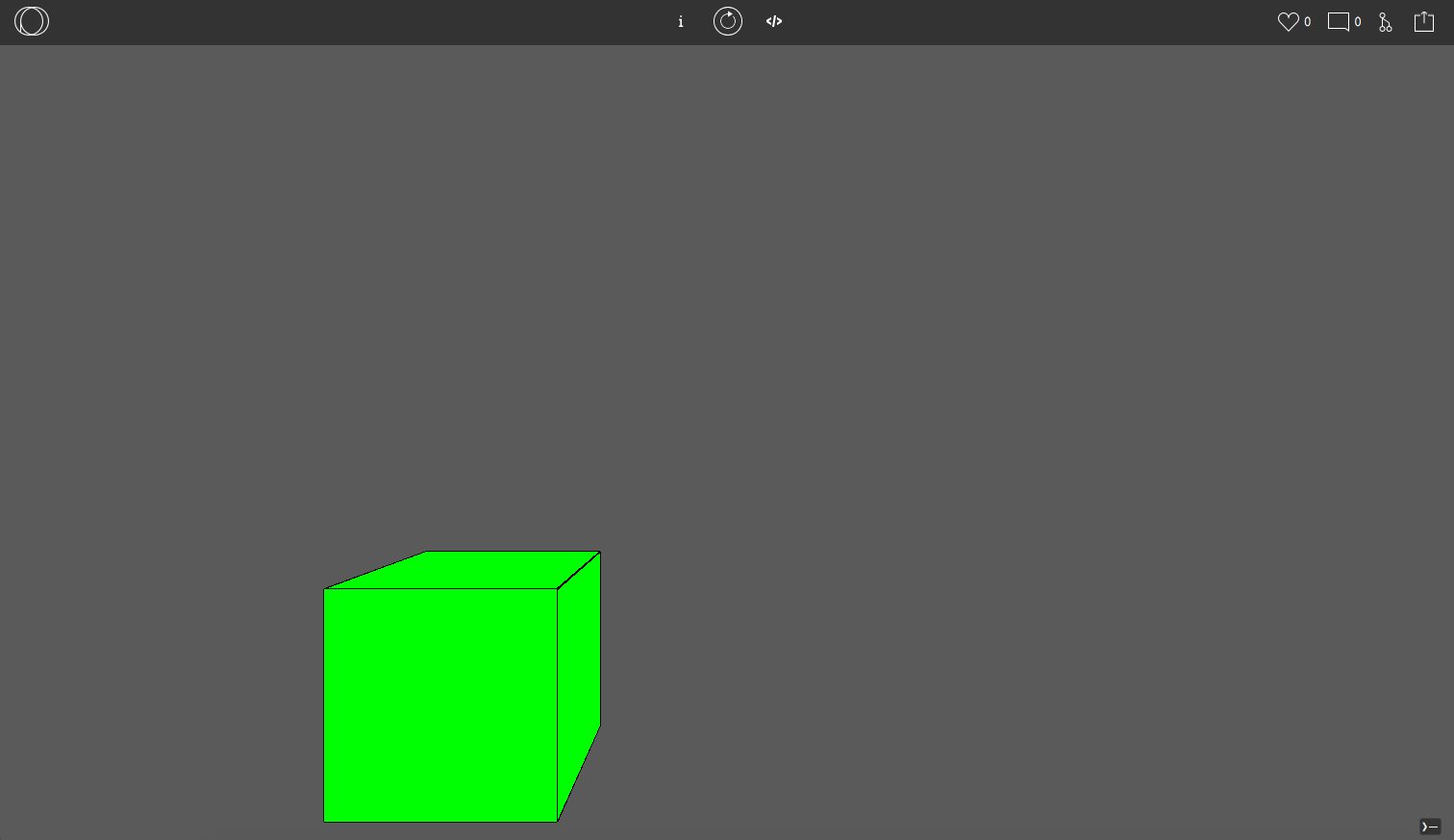
Рис. 3 «Скриншот программы»

Рис. 4 «Скриншот программы»

3.3. Обзор куба

*var r = 0; var d = 0;*

*function setup() {*

*createCanvas(windowWidth, windowHeight, WEBGL);}*

*function draw() {*

*background(1);*

*zSpeed = rotationZ - pRotationZ;*

*d = d - zSpeed/2*

*ySpeed = rotationY - pRotationY;*

*r = r - ySpeed/2*

*camX1 = 500 \* cos(radians(d\*2));*

*camY1 = 500 \* cos(radians(r\*4));*

*camZ1 = 500 \* sin(radians(d\*2));*

*camera(-camX1, camY1, camZ1, 0, 0, 0, 0, -1, 0);*

*fill(0, 255, 0); box(200);}*

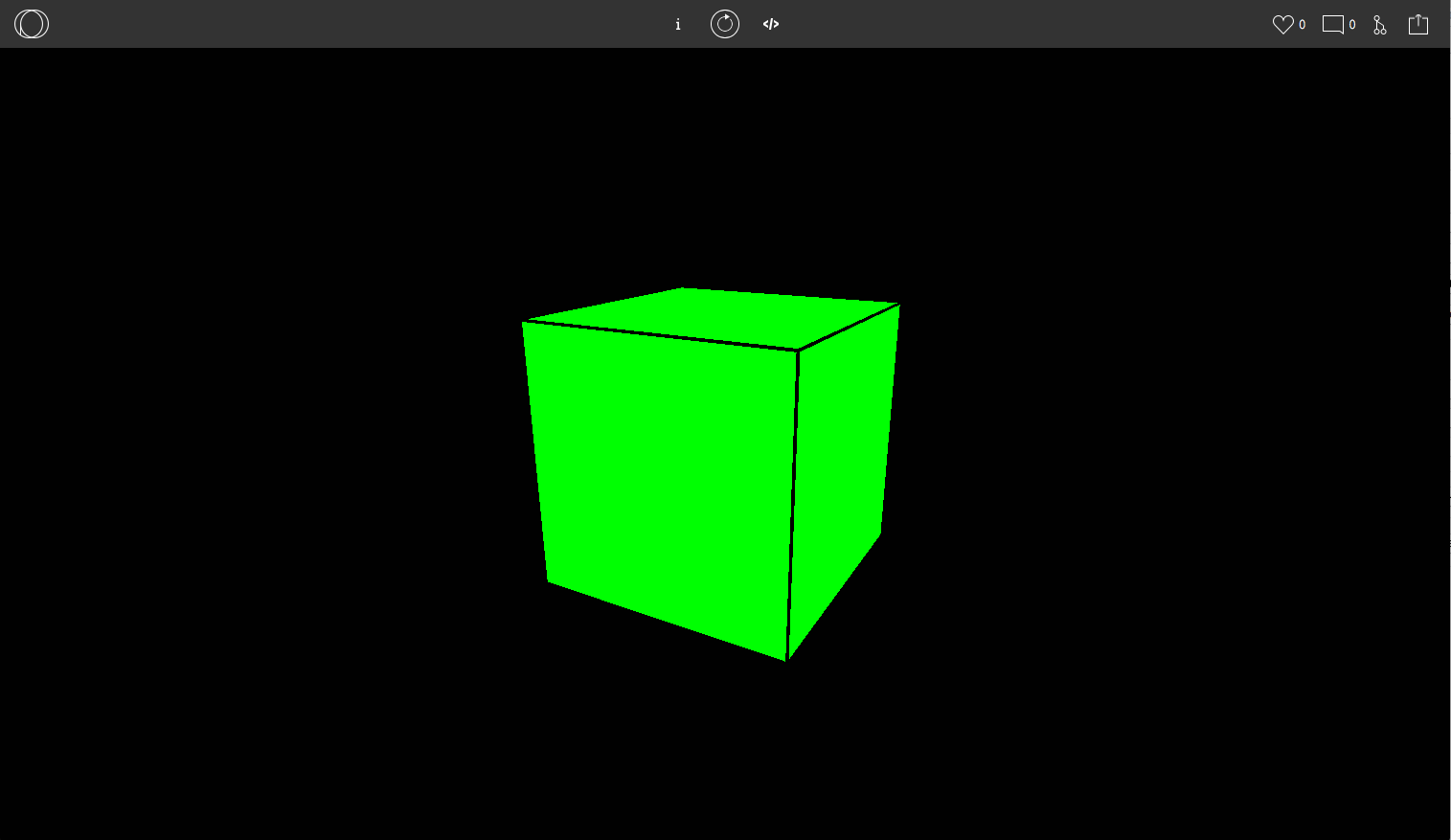
Эта программа позволяет «смотреть» на куб с разных сторон, используя угол поворота устройства. Это все происходит при использовании функции камеры.

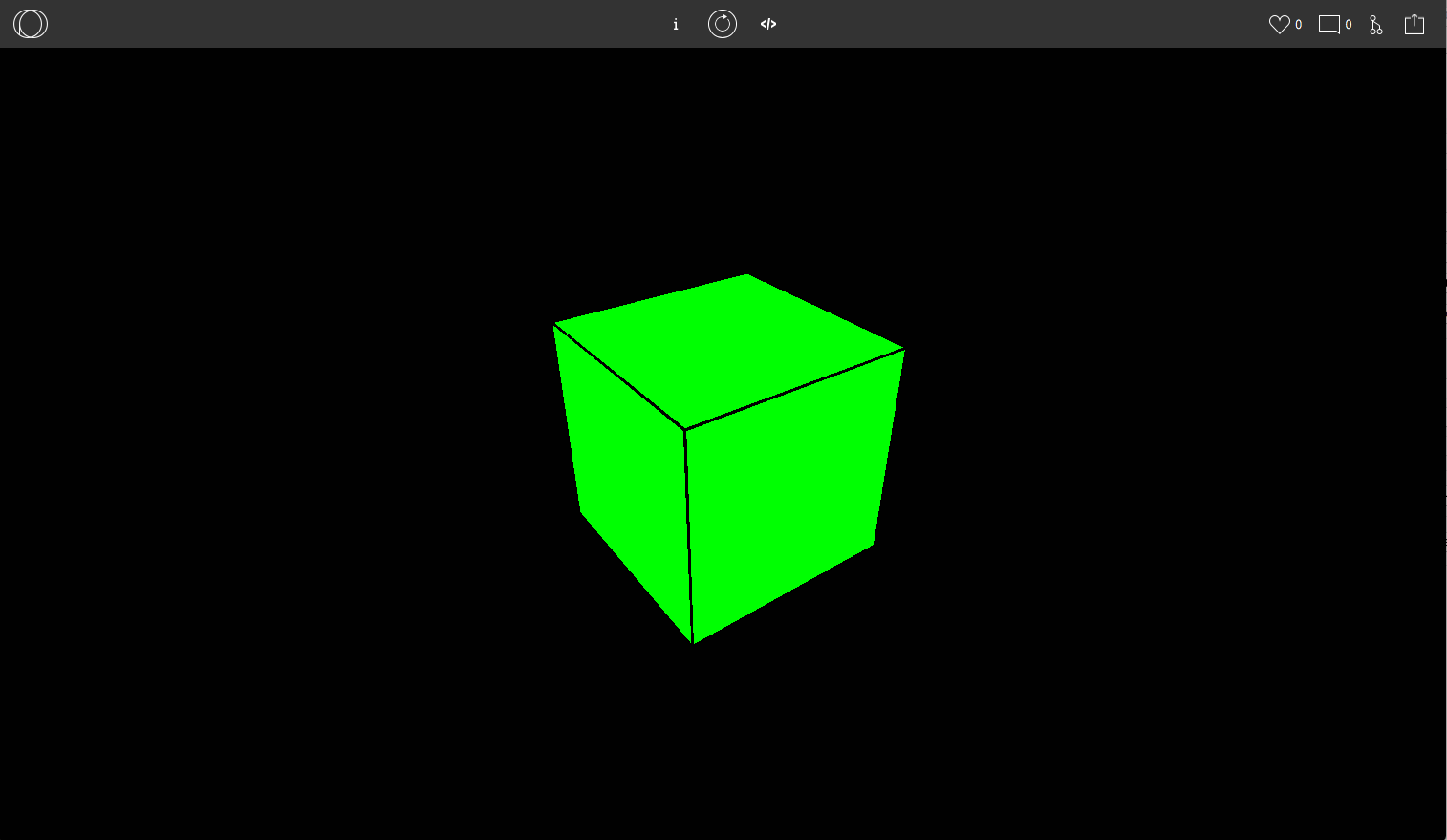
Рис. 5 «Скриншот программы»

Рис. 6 «Скриншот программы»

3.4. Изменение взгляда

*var r = 4500;*

*var d = 1500;*

*function setup() {*

*createCanvas(windowWidth, windowHeight, WEBGL);}*

*function draw() {*

*background(1);*

*zSpeed = rotationZ - pRotationZ;*

*d = d - zSpeed/2*

*ySpeed = rotationY - pRotationY;*

*r = r - ySpeed/2*

*camX1 = map(r, 0, width,-200,200);*

*camY1 = map(d, 0, width,-200,200);*

*camera(0, 0, (height/2) / tan(PI/6), camX1, camY1, 0, 0, 1, 0);*

*translate(windowWidth/2-100, 100, -100);*

*fill(0, 255, 0);*

*box(200);}*

*function keyPressed(){*

*if (key === "W"){*

*d = d + 200;*

*println(camX1 + ' ' + camY1);}*

*else if (key === "S"){*

*d = d - 200;*

*println(camX1 + ' ' + camY1);}*

*else if (key === "Q"){*

*r = r + 200;*

*println(camX1 + ' ' + camY1);}*

*else if (key === "A"){*

*r = r - 200;*

*println(camX1 + ' ' + camY1);}}*

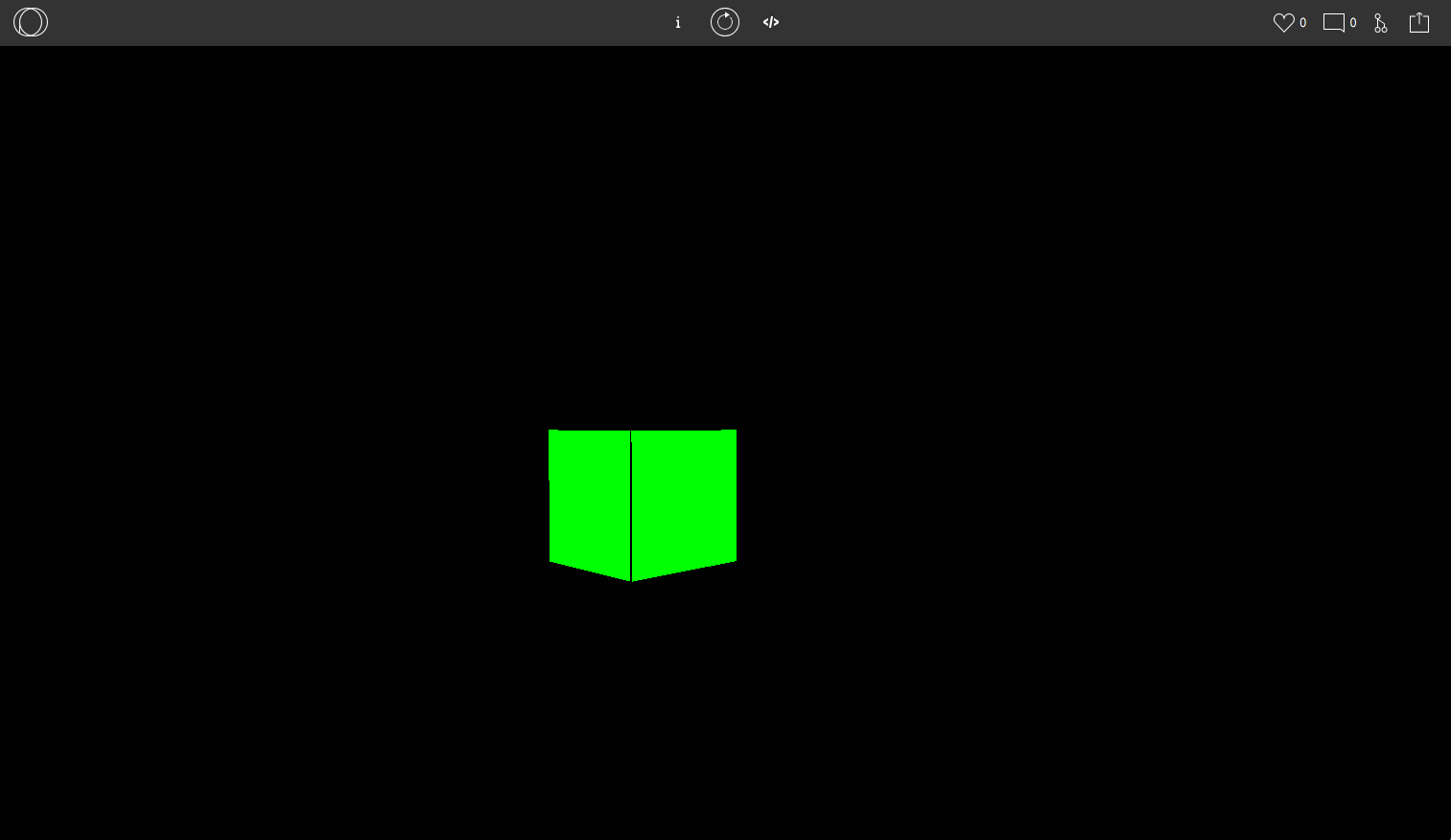
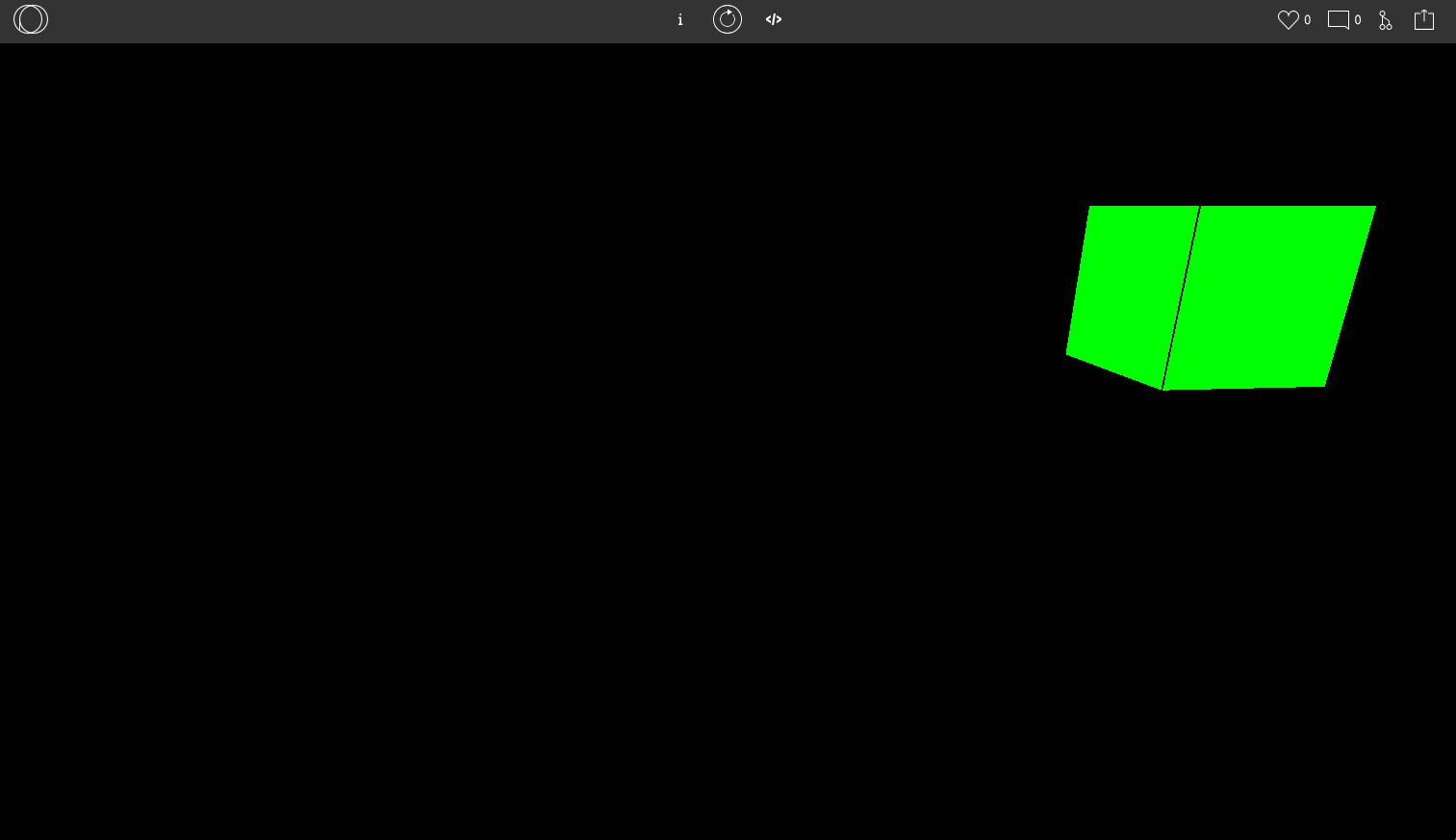
Эта программа позволяет направлять взгляд на разные точки в пространстве. Для наглядности был создан объект, чтобы наблюдать за правильностью работы программы.

Рис. 7 «Скриншот экрана» Рис. 8 «Скриншот экрана»

Заключение

Результатом моего диплома является ряд приложений, которые были созданы на основе компьютерного параллакса. Желаемого приложения, которое было задумано, как финальный продукт, сделать не удалось. Я столкнулся с несколькими проблемами. Первая была в программе «Перемещение». Сейчас объект изменяется вовремя нажатия кнопок, но была цель изменять объект в программе в зависимости от передвижения устройства, но программа неправильно считала ускорение и скорость, из-за этого результат получился неудовлетворительным. Если бы было получено это задание сейчас, я бы построил определенное количество графиков, показывающих время, скорость и ускорение, для определения ошибок и изъянов. Также проблемой было совмещение программы «Перемещение» и «Изменение взгляда». Каждая из них зависит от двух параметров: положения камеры в пространстве и точки, куда направлена камера. При совмещении этих программ в одну ничего не получалось. Если бы было получено это задание сейчас, я бы подумал, как это сделать по-другому, так как сейчас не считаю реальным, создание программы с используемыми компонентами. Также я бы изменил сам объект, так как сейчас я использую куб. Можно было бы создать другой объект для наглядности и эффективного анализа результата. Так как раньше я не сталкивался в программировании в среде p5.js и был не осведомлен в этой теме, у меня возникли трудности. Если такая работа будет получена в будущем, я уверен, что она будет более успешной.

**Источники**

1. [https://ru.wikipedia.org/wiki/Параллакс](https://ru.wikipedia.org/wiki/%25D0%259F%25D0%25B0%25D1%2580%25D0%25B0%25D0%25BB%25D0%25BB%25D0%25B0%25D0%25BA%25D1%2581) - онлайн библиотека
2. <http://fb.ru/article/239791/parallaks> —eto-parallaks-effekt- primeryi
3. [p5.js.org](http://p5.js.org) - онлайн библиотека для языка