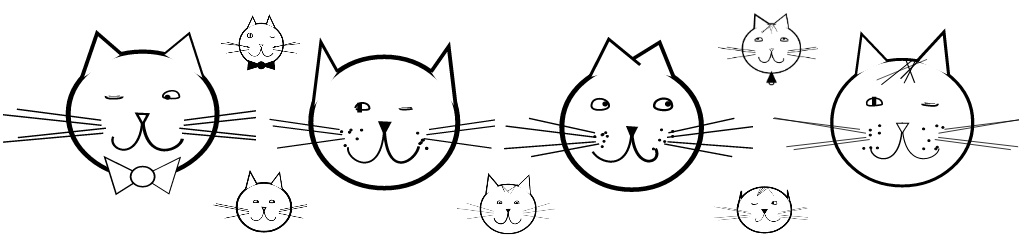
Генератор случайных котов за 8 шагов

  
 Я всегда очень любил котов, и любил их рисовать, особенно морды кошачьи. Чуть изменишь форму, линию — и совершенно другое выражение, другое настроение. У меня листы А4 были изрисованы под предел. И тут мне недавно стукнуло в голову — а что если сделать генератор морд котов? Чтобы нажал на кнопочку и тебе выкинется случайная морда кота. Как можно более случайная и интересная. Давайте же посмотрим, как же сделать такую штуку.   
Прошу под кат, любители котов.  
  
Всё будем делать на JS и Canvas'е, и я не привожу код инициализации, настройки и подобного. Этого в интернетах полно, а нам же интересно как рисовать котов, правда?  
  
Приведу лишь несколько предопределенных функций, просто для дальнейшего удобства, вот они:

**function** **add**(func, scale){

Graphics.ctx.scale(scale.x, scale.y);

func(Graphics.ctx);

Graphics.ctx.scale(1, 1);

}

Меняем размер канваса, что-то рисуем, возвращаем назад.

**function** **draw**С**ircle**(ctx, pos, radius, fillColor, strokeColor, lineWidth)

{

ctx.beginPath();

ctx.arc(center.x - pos.x, center.y + pos.y, radius, 0, 2\*Math.PI, false);

ctx.fillStyle = fillColor;

ctx.fill();

ctx.lineWidth = lineWidth;

ctx.strokeStyle = strokeColor;

ctx.stroke();

}

А это рисование окружностей\кругов. Тоже ничего особенного, всё это можно прочитать миллионы раз и на хабре и где угодно, давайте скорее к котам!

Шаг 1 — Голова

Начнем мы с головы. Шо есмь голова? Окружность. Которую можно немного сжать, или расширить.  
Ещё можно задать толщину обводки, это тоже сделаем. И есть маленькая деталь, смотрим в коде комментарии.  
Код весь прокомментирован, думаю проблем с понимаем не возникнет.

*//Генерим случайный радиус*

**var** radius = Math.sRandom(60, 70);

*//Изменяем размеры*

**var** scaleCircle = {x:Math.sRandom(1,1.1),y:Math.sRandom(0.9,1.0)};

*//Задаем цвета*

**var** whiteColor = "#fff";

**var** blackColor = "#000";

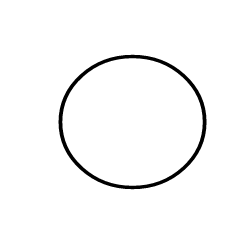
*//Рисуем*

add(**function**(ctx){

*// (Math.sRandom(0, 100) < 95 ? blackColor : whiteColor) - эта строчка, иногда, редко, делает контур белым, т.е. голова оказывается без контура. Уверяю, это получаеться здорово.*

drawСircle(ctx, {x:0,y:0}, radius, whiteColor, (Math.sRandom(0, 100) < 95 ? blackColor : whiteColor), Math.sRandom(2,5));

}, scaleCircle);

Вот и голова появилась.   
  
  
Идем дальше!

Шаг 2 — Уши

Уши у котов — одно из самых выразительных элементов морды. Поэтому чем более уши будут разные, по разному повернуты, разной ширины, тем больше будет разнообразия выражений.  
Делаем.

*//Вектор, который будем поворачивать, в поисках точек ушей*

**var** dir = {x:0, y:radius};

*//Первый угол*

**var** angleOne = Math.PI + Math.PI/Math.sRandom(2, 5);

*//Поворачиваем вектор и получаем первую точку*

**var** pointR1 = VectorRot(dir, angleOne);

*//Делаем вторую точку, через поворт вектора на первый угол минус отклонение*

**var** pointR2 = VectorRot(dir, angleOne - Math.PI/Math.sRandom(4, 7));

*//Считаем верхнюю точку уха*

**var** topPointR = {x:((pointR1.x + pointR2.x) / 2)+Math.sRandom(-10, 10), y:pointR2.y - 30 + Math.sRandom(0, 5)};

*//Рисуем*

add(**function**(ctx){

*//Рисуем правое ухо*

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor;

ctx.lineWidth = Math.sRandom(2, 4);

topPointR = {x:topPointR.x+Math.sRandom(-1,5), y:topPointR.y+Math.sRandom(-5,5)};

ctx.moveTo(center.x + pointR1.x,center.y + pointR1.y);

ctx.lineTo(center.x + topPointR.x,center.y + topPointR.y);

ctx.lineTo(center.x + pointR2.x,center.y +pointR2.y);

ctx.fill();

ctx.stroke();

*//Рисуем левое ухо*

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor;

ctx.lineWidth = Math.sRandom(2, 4);

**var** topPointL = VectorXInvert({x:topPointR.x+Math.sRandom(-5,5), y:topPointR.y+Math.sRandom(-5,5)});

**var** pointL1 = VectorXInvert(pointR1);

**var** pointL2 = VectorXInvert(pointR2);

ctx.moveTo(center.x + pointL1.x,center.y + pointL1.y);

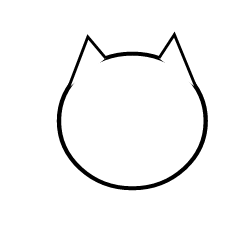
ctx.lineTo(center.x + topPointL.x,center.y + topPointL.y);

ctx.lineTo(center.x + pointL2.x,center.y +pointL2.y);

ctx.fill();

ctx.stroke();

}, {x:1,y:1});

Иииии вот что у нас получилось:  
  
  
  
Уже напоминает кота, правда?

Шаг 3 — Усы

Второй во выразительности, после ушей, элемент морды кота. Сделать его не сложно, но сделать так чтобы они выглядели более правдоподобно сложнее. Одним рандомом тут не обойдёшься, приходится крутить цикл, и по счетчику высчитывать точки, для более линейных результатов.

*//Расчитываем точки начала усов*

*//путем выбора вектора, и поворота вектора на почти случайное значение*

*//Результат - точка начала уса*

**var** pointsR = [];

*//Выбираем число усов*

**var** count = Math.floor(Math.sRandom(3, 5));

**for** (**var** i = 0; i < count; i++) {

*//Делаем вектор, с длиной из радиуса круга / на случайное значение*

**var** dir = {x:0, y:radius/Math.sRandom(1.6,1.9)};

*//Выбираем угол, плюс некоторые действия, для красоты*

**var** angleOne = Math.PI/(2 + ((i+1)/4));

*//Пвоворачиваем вектор*

**var** pointR1 = VectorRot(dir, angleOne);

*//Выбираем y конечной точки, так чтобы первую половину count усы отлонялись в одну сторону*

*//а во второй половине в другую*

**var** y = pointR1.y+(i < count / 2 ? -Math.sRandom(8, 25) : Math.sRandom(7, 15) );

*//Записываем точки начала и конца в массив*

pointsR.push({begin:pointR1, end:{x:pointR1.x - Math.sRandom(60, 100),y:y}});

}

*//Случайное значение ширины линии*

**var** lineWidth = Math.sRandom(0.5, 2);

*//Рисуем*

add(**function**(ctx){

*//Правая сторона*

**for** (**var** i = 0; i < pointsR.length; i++) {

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor;

ctx.lineWidth = lineWidth;

ctx.moveTo(center.x - pointsR[i].begin.x,center.y + pointsR[i].begin.y);

ctx.lineTo(center.x - pointsR[i].end.x,center.y + pointsR[i].end.y);

ctx.stroke();

}

*//Левая сторона*

**for** (**var** i = 0; i < pointsR.length; i++) {

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor;

ctx.lineWidth = lineWidth;

**var** pointLBegin = VectorXInvert(pointsR[i].begin);

**var** pointLEnd = VectorXInvert(pointsR[i].end);

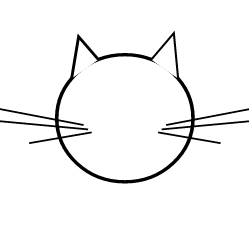
ctx.moveTo(center.x - pointLBegin.x,center.y + pointLBegin.y);

ctx.lineTo(center.x - pointLEnd.x,center.y + pointLEnd.y);

ctx.stroke();

}

}, {x:1,y:1});

Функция VectorXInvert пусть вас не смущает, просто приходилось часто отражать по x различные элементы, поэтому сделана простая функция, которая делает x = -x;  
  
Смотрим как прорисовывается наш кот:  
  


Шаг 4 — Рот

Это довольно сложно сделать реалистично, поэтому сделаем просто линию рта, схематично, но для нашего графического рисунка вполне пойдет.  
Пишем код, рисующий линию рта. Сделано на кривых Безье.

*//---Рот*

*//Рот строем с помощью кривых Безье. Задаем четыре точки + 2 точки для отражения*

*// P0*

*// P3 | iP3*

*// | | |*

*// P2--P1--iP2*

**var** P0 = {x:center.x, y:center.y};

**var** P1 = {x:center.x, y:center.y + Math.sRandom(40, 65)};

**var** P2 = {x:center.x - Math.sRandom(29, 36),y: center.y + 40};

**var** P3 = {x:center.x - Math.sRandom(20, 40), y:center.y + Math.sRandom(23, 28)};

**var** iP2 = {x:center.x + Math.sRandom(29, 36),y: center.y + 40};

**var** iP3 = {x:center.x + Math.sRandom(20, 40), y:center.y + Math.sRandom(23, 28)};

*//Рисуем*

add(**function**(ctx){

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor ;

ctx.lineWidth = Math.sRandom(1,3);

ctx.moveTo(P0.x,P0.y );

ctx.bezierCurveTo(P1.x, P1.y, P2.x, P2.y, P3.x, P3.y);

ctx.stroke();

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = fillColor ;

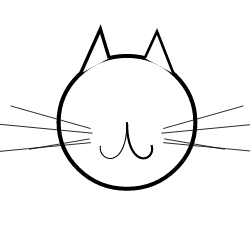
ctx.lineWidth = Math.sRandom(1,3);

ctx.moveTo(P0.x,P0.y );

ctx.bezierCurveTo(P1.x, P1.y, iP2.x, iP2.y, iP3.x, iP3.y);

ctx.stroke();

}, {x:1,y:1});

Глядим на то, что получилось. Терпение, ещё чуть-чуть осталось!  


Шаг 5 — Нос

Нос есть нос. Сделаем его двух вариантов — маленький круг, и треугольничек. Круг будет появляться намного реже. Ну так, для интереса.  
Разуметься примешиваем рандом, в том числе в цвета, чтобы получались разные носы — закрашенные, и просто контур. Нам ведь нужно много котов?

*//Два типа носа - либо треугольник, либо круг*

**var** chance = Math.sRandom(0, 100);

**if**(chance < 98)

{

*//Коефициент размера*

**var** scale = {x:Math.sRandom(0.9,1.3), y:Math.sRandom(0.9,1.3)};

*//Правая точка*

**var** pointR ={x:Math.sRandom(4,5)\*scale.x,y:Math.sRandom(-5,-4)\*scale.y};

*//Левая точка*

**var** pointL ={x:Math.sRandom(-4,-5)\*scale.x,y:Math.sRandom(-5,-4)\*scale.y};

*//Нижняя точка*

**var** bottomPoint ={x:0,y:Math.sRandom(5,6)\*scale.y};

*//Рисуем*

add(**function**(ctx){

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = (Math.sRandom(0, 100) > 50 ? fillColor : strokeColor);

ctx.lineWidth = Math.sRandom(1,3);

ctx.moveTo(center.x + pointR.x,center.y + 5 + pointR.y);

ctx.lineTo(center.x + bottomPoint.x,center.y + 5 + bottomPoint.y);

ctx.lineTo(center.x + pointL.x,center.y + 5 + pointL.y);

ctx.closePath();

ctx.fill();

ctx.stroke();

}, {x:1,y:1});

}

**else**

{

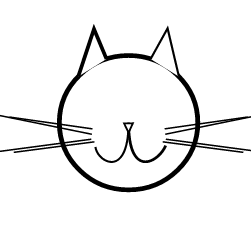
*//Рисуем кот*

add(**function**(ctx){

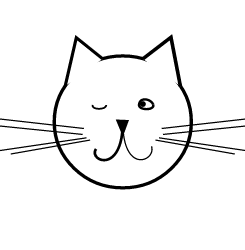
drawСircle(ctx, {x:0,y:0}, Math.sRandom(7, 10), strokeColor, strokeColor, 1);

}, {x:1,y:1});

}

Наш кот постепенно превращается в кота. Это не может не радовать.  
  


Шаг 6 — Глаза

Глаза штука сложная. Нет, нарисовать не сложно — просто две кривых Безье. Но надо ещё добавить закрывающееся глаза, котик же может спать, а может и один глаз приоткрыть. И это надо сделать. В этот раз просто кот, без кода. Ибо большой объем, не хочется утомлять читателя кучей текста. В конце будут приведены исходники, кто захочет поглядит. А мы просто посмотрим на то что получилось.  
  
  
  
Кот уже есть, но добавим ещё некоторый декор.

Шаг 7 — Точки усов на щеках

Знаете, бывают такие. Вероятность появления естественно, не 100%. Вот код, потом результат.  
Код простой — просто раскидываем точки на некотором расстояниие от носа.

**if**(Math.sRandom(0, 100) > 60)

{

add(**function**(ctx){

**for** (**var** i = 0; i < Math.sRandom(5, 7); i++) {

**var** P = {x:Math.sRandom(20, 40),y:Math.sRandom(0, 30)};

drawСircle(ctx, P, 1, strokeColor, strokeColor, 1);

}

**for** (**var** i = 0; i < Math.sRandom(5, 7); i++) {

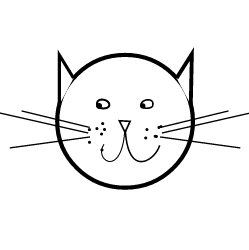
**var** P = {x:-Math.sRandom(20, 40),y:Math.sRandom(0, 30)};

drawСircle(ctx, P, 1, strokeColor, strokeColor, 1);

}

}, {x:1,y:1});

}

Результат:  


Шаг 8 — «Челка»

Просто иногда рисуем парочку линий сверху. На словах тяжело описать, смотрим код, смотрим что получается.

**if**(Math.sRandom(0, 100) > 75)

{

add(**function**(ctx){

**for** (**var** i = 0; i < Math.sRandom(3, 5); i++) {

*//Берем вектор равный по длине радиусу*

**var** radiusVector = {x:0,y:-radius};

*//Поворачиваем вектор на некоторое отлонение*

*//Нужно чтобы волосы выходили ровно из окружности*

radiusVector = VectorRot(radiusVector, Math.sRandom(-0.01, 0.01))

*//Строим нижнии точки*

**var** P0 = {x:Math.sRandom(-25, 25), y: radiusVector.y};

**var** P1 = {x:Math.sRandom(-25, 25), y: Math.sRandom(-50, -40)};

*//Ширина волоса*

**var** lineWidth = Math.sRandom(0.5, 1.5);

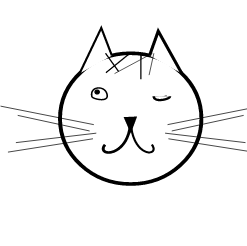
*//Рисуем*

drawLine(ctx, P0, P1, strokeColor, strokeColor, lineWidth);

}

}, {x:1,y:1});

}

Здорово, правда?   
  


Шаг 8 — Колокольчик или бантик

Ну а это уже просто для интереса, я думал чтобы ещё добавить, и решил добавить бантик и колокольчик. Не судите строго, я развлекаюсь :)  
Как всегда код, потом красивые картинки с котиками.

*//Декорации*

*//Либо бабочка либо колокольчик*

**var** chanceBottom = Math.sRandom(0, 100);

**if**(chanceBottom > 50)

{

**var** chance = Math.sRandom(0, 100);

**if**(chance > 90)

{

*//Бабочка - два треугольника + круг, с точками в случайном дипазаоне*

**var** P = {x:0,y:radius};

**var** P0 = {x:Math.sRandom(20, 45), y:radius- Math.sRandom(13, 22)};

**var** P1 = {x:Math.sRandom(20, 45), y:radius+ Math.sRandom(13, 22)};

add(**function**(ctx){

**var** color = (Math.sRandom(0, 100) > 50 ? fillColor : strokeColor);

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = color;

ctx.lineWidth = Math.sRandom(2, 5);

ctx.moveTo(center.x + P.x,center.y + P.y);

ctx.lineTo(center.x + P0.x,center.y + P0.y);

ctx.lineTo(center.x + P1.x,center.y + P1.y);

ctx.closePath();

ctx.stroke();

ctx.fill();

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = color;

ctx.lineWidth = Math.sRandom(2, 5);

ctx.moveTo(center.x - P.x,center.y + P.y);

ctx.lineTo(center.x - P0.x,center.y + P0.y);

ctx.lineTo(center.x - P1.x,center.y + P1.y);

ctx.closePath();

ctx.stroke();

ctx.fill();

drawСircle(ctx, P, Math.sRandom(6, 12), (Math.sRandom(0, 100) > 50 ? fillColor : strokeColor), strokeColor, Math.sRandom(1, 3));

}, {x:1,y:1});

}

}

**else**

{

*//Колокольчик*

**var** chance = Math.sRandom(0, 100);

**if**(chance > 90)

{

*//Колокольчик - треугольник, внизу с кругом. Строится по трём точкам, всё аналогично, как и выше*

**var** P = {x:0,y:radius};

**var** P0 = {x:Math.sRandom(8, 15), y:radius+ Math.sRandom(23, 29)};

**var** P1 = {x:-P0.x, y:P0.y};

**var** P3 = {x:0,y:P0.y + Math.sRandom(0, 7)};

add(**function**(ctx){

drawСircle(ctx, P3, Math.sRandom(2, 6), (Math.sRandom(0, 100) > 50 ? fillColor : strokeColor), strokeColor, Math.sRandom(1, 3));

**var** color = (Math.sRandom(0, 100) > 50 ? fillColor : strokeColor);

ctx.beginPath();

ctx.strokeStyle = strokeColor;

ctx.fillStyle = color;

ctx.lineWidth = Math.sRandom(2, 5);

ctx.moveTo(center.x + P.x,center.y + P.y);

ctx.lineTo(center.x + P0.x,center.y + P0.y);

ctx.lineTo(center.x + P1.x,center.y + P1.y);

ctx.closePath();

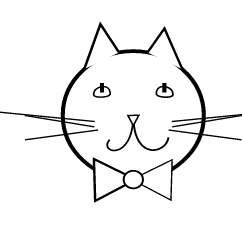
ctx.stroke();

ctx.fill();

}, {x:1,y:1});

}

}

Как видите ничего сложного, просто геометрические фигуры.  
Код был бы скучен, если бы не было котов, да будет кот с бантиком!  
  


Заключение

Ну вот и закончил я свое повествование о котах.  
Привожу код на гитхабе: [github.com/MagistrAVSH/random-cat](https://github.com/MagistrAVSH/random-cat)  
А вот результат всех трудов, можете пощелкать: [magistravsh.github.io/random-cat](http://magistravsh.github.io/random-cat/)  
А теперь брысь от монитора, и погладь кота! :)