var s;

var scl = 20;

var food;

playfield = 600;

// p5js Setup function - required

function setup() { // 캔버스 생성 뱀설정 프레인레이트 설정

  createCanvas(playfield, 640);

  background(51);

  s = new Snake();

  frameRate (10);

  pickLocation();

}

// p5js Draw function - required

function draw() { 점수판 설정 죽었을 때 업데이트 보여주는거

  background(51);

  scoreboard();

  if (s.eat(food)) {

    pickLocation();

  }

  s.death();

  s.update();

  s.show();

  fill (255,0,100);

  rect(food.x,food.y, scl, scl); 사각형 먹이

}

// Pick a location for food to appear

function pickLocation() { 줍는거 먹이설정

  var cols = floor(playfield/scl);

  var rows = floor(playfield/scl);

  food = createVector(floor(random(cols)), floor(random(rows)));

  food.mult(scl);

  // Check the food isn't appearing inside the tail

  for (var i = 0; i < s.tail.length; i++) { 먹으면 길이 증가

    var pos = s.tail[i];

    var d = dist(food.x, food.y, pos.x, pos.y);

    if (d < 1) {

      pickLocation();

    }

  }

}

// scoreboard

function scoreboard() { 점수판 설정

  fill(0);

  rect(0, 600, 600, 40);

  fill(255);

  textFont("Georgia");

  textSize(18);

  text("Score: ", 10, 625);

  text("Highscore: ", 450, 625)

  text(s.score, 70, 625);

  text(s.highscore, 540, 625)

}

// CONTROLS function

function keyPressed() { 키보드

  if (keyCode === UP\_ARROW){

      s.dir(0, -1);

  }else if (keyCode === DOWN\_ARROW) {

      s.dir(0, 1);

  }else if (keyCode === RIGHT\_ARROW) {

      s.dir (1, 0);

  }else if (keyCode === LEFT\_ARROW) {

      s.dir (-1, 0);

  }

}

// SNAKE OBJECT

function Snake() { 뱀

  this.x =0;

  this.y =0;

  this.xspeed = 1;

  this.yspeed = 0;

  this.total = 0;

  this.tail = [];

  this.score = 1;

  this.highscore = 1;

  this.dir = function(x,y) { 속도

    this.xspeed = x;

    this.yspeed = y;

  }

  this.eat = function(pos) { 먹었을때 점수설정

    var d = dist(this.x, this.y, pos.x, pos.y);

    if (d < 1) {

      this.total++;

      this.score++;

      text(this.score, 70, 625);

      if (this.score > this.highscore) {

        this.highscore = this.score;

      }

      text(this.highscore, 540, 625);

      return true;

    } else {

      return false;

    }

  }

  this.death = function() {

    for (var i = 0; i < this.tail.length; i++) {

      var pos = this.tail[i];

      var d = dist(this.x, this.y, pos.x, pos.y);

      if (d < 1) {

        this.total = 0;

        this.score = 0;

        this.tail = [];

      }

    }

  }

  this.update = function(){ 업데이트 설정 먹으면 꼬리 길이 늘어남

    if (this.total === this.tail.length) {

      for (var i = 0; i < this.tail.length-1; i++) {

        this.tail[i] = this.tail[i+1];

    }

    }

    this.tail[this.total-1] = createVector(this.x, this.y);

    this.x = this.x + this.xspeed\*scl; 한칸씩 이동

    this.y = this.y + this.yspeed\*scl;

    this.x = constrain(this.x, 0, playfield-scl);

    this.y = constrain(this.y, 0, playfield-scl);

  }

  this.show = function(){ 바로바로 적용되서 보여진다.

    fill(255);

    for (var i = 0; i < this.tail.length; i++) {

        rect(this.tail[i].x, this.tail[i].y, scl, scl);

    }

    rect(this.x, this.y, scl, scl);

  }

}

색깔 랜덤하게 변하는

var r;

var g;

var b;

var a;

var s;

var scl = 20;

var food;

playfield = 600;

// p5js Setup function - required

function setup() {

  createCanvas(playfield, 640);

  background(51);

  s = new Snake();

  frameRate (10);

  pickLocation();

}

// p5js Draw function - required

function draw() {

  background(51);

  scoreboard();

  if (s.eat(food)) {

    pickLocation();

  }

  s.death();

  s.update();

  s.show();

  r = random(255);

  g = random(100,200);

  b = random(100);

  a = random(200,255);

  fill (255,0,100);

  rect(food.x,food.y, scl, scl);

}

// Pick a location for food to appear

function pickLocation() {

  var cols = floor(playfield/scl);

  var rows = floor(playfield/scl);

  food = createVector(floor(random(cols)), floor(random(rows)));

  food.mult(scl);

  // Check the food isn't appearing inside the tail

  for (var i = 0; i < s.tail.length; i++) {

    var pos = s.tail[i];

    var d = dist(food.x, food.y, pos.x, pos.y);

    if (d < 1) {

      pickLocation();

    }

  }

}

// scoreboard

function scoreboard() {

  fill(0);

  rect(0, 600, 600, 40);

  fill(255);

  textFont("Georgia");

  textSize(18);

  text("Score: ", 10, 625);

  text("Highscore: ", 450, 625)

  text(s.score, 70, 625);

  text(s.highscore, 540, 625)

}

// CONTROLS function

function keyPressed() {

  if (keyCode === UP\_ARROW){

      s.dir(0, -1);

  }else if (keyCode === DOWN\_ARROW) {

      s.dir(0, 1);

  }else if (keyCode === RIGHT\_ARROW) {

      s.dir (1, 0);

  }else if (keyCode === LEFT\_ARROW) {

      s.dir (-1, 0);

  }

}

// SNAKE OBJECT

function Snake() {

  this.x =0;

  this.y =0;

  this.xspeed = 1;

  this.yspeed = 0;

  this.total = 0;

  this.tail = [];

  this.score = 1;

  this.highscore = 1;

  this.dir = function(x,y) {

    this.xspeed = x;

    this.yspeed = y;

  }

  this.eat = function(pos) {

    var d = dist(this.x, this.y, pos.x, pos.y);

    if (d < 1) {

      this.total++;

      this.score++;

      text(this.score, 70, 625);

      if (this.score > this.highscore) {

        this.highscore = this.score;

      }

      text(this.highscore, 540, 625);

      return true;

    } else {

      return false;

    }

  }

  this.death = function() {

    for (var i = 0; i < this.tail.length; i++) {

      var pos = this.tail[i];

      var d = dist(this.x, this.y, pos.x, pos.y);

      if (d < 1) {

        this.total = 0;

        this.score = 0;

        this.tail = [];

      }

    }

  }

  this.update = function(){

    if (this.total === this.tail.length) {

      for (var i = 0; i < this.tail.length-1; i++) {

        this.tail[i] = this.tail[i+1];

    }

    }

    this.tail[this.total-1] = createVector(this.x, this.y);

    this.x = this.x + this.xspeed\*scl;

    this.y = this.y + this.yspeed\*scl;

    this.x = constrain(this.x, 0, playfield-scl);

    this.y = constrain(this.y, 0, playfield-scl);

  }

  this.show = function(){

    fill(r, g, b, a);

    for (var i = 0; i < this.tail.length; i++) {

        rect(this.tail[i].x, this.tail[i].y, scl, scl);

    }

    rect(this.x, this.y, scl, scl);

  }

}

var r; // 랜덤한 색상 나타내기위한 값들 (r,g,c,a)

var g;

var c;

var a;

var col = [];

var b = -1;

var s;

var scl = 20;

var food = [];

var playfield = 600;

var cols = [];

var rows = [];

// p5js Setup function - required

function setup() {

createCanvas(playfield, 640);

background(51);

s = new Snake();

frameRate (10);

for(var i = 0; i < 3 ; i++)

pickLocation(i);

}

// p5js Draw function - required

function draw() {

r = random(100); // (r,g,c,a)각 랜덤한 색상

g = random(100,200);

c = random(255);

a = random(200,255);

background(51);

scoreboard();

for(var i = 0; i < 3 ; i++){

if (s.eat(food[i],i)) {

pickLocation(i);

if(i==0)

col[b] = 0;

else if(i==1)

col[b] = 1;

else

col[b] = 2;

}

}

s.death();

s.update();

s.show();

for(var i = 0; i < 3 ; i++){

if(i==0)

fill(255,0,0);

else if(i==1)

fill(0,255,0);

else if(i==2)

fill(r,g,c,a); //(i==2)일때 랜덤한 rgca값

rect(food[i].x,food[i].y, scl, scl);

}

}

// Pick a location for food to appear

function pickLocation(i) {

cols[i] = floor(playfield/scl);

rows[i] = floor(playfield/scl);

food[i] = createVector(floor(random(cols[i])), floor(random(rows[i])));

food[i].mult(scl);

// Check the food isn't appearing inside the tail

for (var i = 0; i < s.tail.length; i++) {

var pos = s.tail[i];

for(var j =0; j < 3; j++){

var d = dist(food[j].x, food[j].y, pos.x, pos.y);

if (d < 1) {

pickLocation(i);

}

}

}

}

// scoreboard

function scoreboard() {

fill(0);

rect(0, 600, 600, 40);

fill(255);

textFont("Georgia");

textSize(18);

text("Score: ", 10, 625);

text("Highscore: ", 450, 625)

text(s.score, 70, 625);

text(s.highscore, 540, 625)

}

// CONTROLS function

function keyPressed() {

if (keyCode === UP\_ARROW){

s.dir(0, -1);

}else if (keyCode === DOWN\_ARROW) {

s.dir(0, 1);

}else if (keyCode === RIGHT\_ARROW) {

s.dir (1, 0);

}else if (keyCode === LEFT\_ARROW) {

s.dir (-1, 0);

}

}

// SNAKE OBJECT

function Snake() {

this.x =0;

this.y =0;

this.xspeed = 1;

this.yspeed = 0;

this.total = 0;

this.tail = [];

this.score = 0;

this.highscore = 0;

this.dir = function(x,y) {

this.xspeed = x;

this.yspeed = y;

}

this.eat = function(pos,k) {

var d = dist(this.x, this.y, pos.x, pos.y);

if (d < 1) {

if(k==0){

this.x = floor(random(30))\*scl; //빨간색을 먹으면 랜덤한 위치로 옮겨주는 코드

this.y = floor(random(30))\*\*scl; //빨간색을 먹으면 랜덤한 위치로 옮겨주는 코드

b++;

this.total++; //빨간색을 먹으면 +1점 상승

this.score++; //빨간색을 먹으면 +1점 상승

}

else if(k==1){

this.score+=2; //초록색을 먹으면 스코어 2점 상승

this.total++; // 총길이 +1

b++; // 컬러값을 저장해줄 b값도 +1

col[b]=1; // col 배열의 b 인덱스에 1 추가

s.update(); // s.update() 함수 한번 더 실행

this.total++; // 총 길이 +1

b++; // 컬러값을 저장해줄 b값도 +1

if (this.score > this.highscore) {

this.highscore = this.score;

}

}

else{

this.total++;

this.score++;

b++;

}

return true;

}

else {

return false;

}

}

this.death = function() {

for (var i = 0; i < this.tail.length; i++) {

var pos = this.tail[i];

var d = dist(this.x, this.y, pos.x, pos.y);

if (d < 1) {

this.total = 0;

this.score = 0;

this.tail = [];

b = -1; //b 값을 초기값 -1로 지정

}

}

}

this.update = function(){ // 함수 그대로

if (this.total === this.tail.length) {

for (var i = 0; i < this.tail.length-1; i++) {

this.tail[i] = this.tail[i+1];

}

}

this.tail[this.total-1] = createVector(this.x, this.y);

this.x = this.x + this.xspeed\*scl;

this.y = this.y + this.yspeed\*scl;

this.x = constrain(this.x, 0, playfield-scl);

this.y = constrain(this.y, 0, playfield-scl);

}

this.show = function(){

for (var i = 0; i < this.tail.length; i++) {

//먹은 음식의 색깔대로 꼬리 생성

if(col[i]==0)

fill(255,0,0);

else if(col[i]==1)

fill(0,255,0);

else if(col[i]==2)

fill(r,g,c,a);

rect(this.tail[i].x, this.tail[i].y, scl, scl);

}

//뱀의 머리부분

fill(255);

rect(this.x, this.y, scl, scl);

}

}