```
//Colorful Bouncy Balls Program:


var velocity;
var colorArray;
var ballArray = [];
var ballRadius;
var velocity;
var numberBalls;




var velocityX;
var velocityY;
//velocity trackers preserve the current velocity over each 20 ms periodic iteration
var velocityXTracker = [];
var velocityYTracker = [];


var buttonArray = ["tenBalls", "hundredBalls", "thousandBalls", "smallRadius", "mediumRadius", "largeRadius", "slowSpeed",
"normalSpeed", "fastSpeed", "warmColors", "rainbowColors", "coldColors"];
var buttonTextArray = ["tenBalls", "hundredBalls", "thousandBalls", "smallRadius", "mediumRadius", "largeRadius", "slowSpeed",
"normalSpeed", "fastSpeed", "warmColors", "rainbowColors", "coldColors"];
var colorButtonArray = [Color.RED, Color.ORANGE, Color.YELLOW, Color.RED, Color.ORANGE, Color.YELLOW,
Color.GREEN, Color.BLUE, Color.PURPLE, Color.CYAN, Color.BLUE, Color.PURPLE];
var questionsArray = [];


var colorPalette;
var moveBallsCounter = 0;


var phase = "1number"; //holds a string that stores the current phase of buttons


/* The following "Colorful Bouncy Balls" program asks the user for the number
of desired balls, their radius, and their speed, and then creates each ball,
each with a randomly assigned color, then assigns each ball a random direction
to move. Whenever a ball collides with an edge of the canvas, it will bounce
away in the opposite direction. The end result should be continuously bouncing
mess of different colored balls, though the result can greatly differ based upon
what the user inputs for the number of balls, their radius, and their speed. */




function start(){
    createButtonArray();
    createButtonTextArray();
    createQuestionsArray();
    executePhase(phase);
    mouseClickMethod(mouseClick);
    setTimer(moveBalls, 50);
}


/* the executePhase function manages each button stage, ensuring that each button
along with its corresponding text, in addition to the question prompts, appears
at the correct time, and disappears once a button has been clicked */
function executePhase(phase){
    var button;
    var text;
    switch(phase){
        case "1number":
            questionsArray[0].move(55, 0);
            add(questionsArray[0]);
        break;
    case "2size":
        remove(questionsArray[0]);
        questionsArray[1].move(50, 0);
        add(questionsArray[1]);
        break;
    case "3speed":
        remove(questionsArray[1]);
        questionsArray[2].move(20, 0);
        add(questionsArray[2]);
        break;
    case "4color":
        remove(questionsArray[2]);
        questionsArray[3].move(15, 0);
        add(questionsArray[3]);
        break;
    case "5end":
        remove(questionsArray[3]);
        break;
}
if(phase == "4color"){
    for(var i = 9; i <= 11; i++){
        text = buttonTextArray[i];
        add(text);
    }
    createColorButtons();
}
if(phase == "5end"){
    for(var i = 9; i <= 11; i++){
        text = buttonTextArray[i];
        remove(text);
    }
    for(var i = 0; i <= colorButtonArray.length; i++){
        button = colorButtonArray[i];
        remove(button);
    }
    colorArray = assignColors(numberBalls, colorPalette);
    createBalls(numberBalls, colorArray, ballRadius);
    velocityX = velocity;
    velocityY = velocity;
}
var firstIndex;
var lastIndex;
switch(phase){
    case "1number":
        firstIndex = 0;
        lastIndex = 2;
        break;
    case "2size":
        firstIndex = 3;
        lastIndex = 5;
        break;
    case "3speed":
        firstIndex = 6;
        lastIndex = 8;
        break;
    case "4color":
        firstIndex = 9;
        lastIndex = 11;
}

//removing previous phase's buttons
if(phase != "1number"){
    for(var i = (firstIndex - 3); i <= (lastIndex - 3); i++){
        button = buttonArray[i];
        text = buttonTextArray[i];
        remove(button);
```

```
            remove(text);
        }
    }
    //adding current phase's buttons
        for(var i = firstIndex; i <= lastIndex; i++){
            button = buttonArray[i];
            text = buttonTextArray[i];
            if(phase != "4color"){
                add(button);
            }
            add(text);
    }
}


//detects button clicks and reacts correspondingly to each button press
function mouseClick(e){
    if(phase != "5end"){
        if(e.getX() > (getWidth() * (1/4)) && e.getX() < (getWidth() * (3/4))){
            //check whether the strings are actually functional in boolean comparisons
            if(e.getY() > (getHeight() * (2/9)) && e.getY() < (getHeight() * (3/9))){ //bottom
                if(phase == "4color"){
                    phase = "5end";
                    colorPalette = "warmColors";
                    executePhase(phase);
                }
                if(phase == "3speed"){
                    phase = "4color";
                    velocity = 30; //fast speed
                    velocityX = velocity;
                    velocityY = velocity;
                    executePhase(phase);
                }
                if(phase == "2size"){
                    phase = "3speed";
                    ballRadius = 40; //large
                    executePhase(phase);
                }
                if(phase == "1number"){
                    phase = "2size";
                    numberBalls = 1000;
                    executePhase(phase);
                }
            }
            if(e.getY() > (getHeight() * (4/9)) && e.getY() < (getHeight() * (5/9))){ //middle
                if(phase == "4color"){
                    phase = "5end";
                    colorPalette = "rainbowColors";
                    executePhase(phase);
                }
                if(phase == "3speed"){
                    phase = "4color";
                    velocity = 15; //average speed
                    velocityX = velocity;
                    velocityY = velocity;
                    executePhase(phase);
                }
                if(phase == "2size"){
                    phase = "3speed";
                    ballRadius = 25; //medium
                    executePhase(phase);
                }
                if(phase == "1number"){
                    phase = "2size";
                    numberBalls = 100;
                    executePhase(phase);
                }
            }
            if(e.getY() > (getHeight() * (6/9)) && e.getY() < (getHeight() * (7/9))){ //top
                if(phase == "4color"){
                    phase = "5end";
                    colorPalette = "coldColors";
                    executePhase(phase);
                }
                if(phase == "3speed"){
                    phase = "4color";
                    velocity = 5; //slow speed
                    velocityX = velocity;
                    velocityY = velocity;
                    executePhase(phase);
                }
                if(phase == "2size"){
                    phase = "3speed";
                    ballRadius = 5; //small
                    executePhase(phase);
                }
                if(phase == "1number"){
                    phase = "2size";
                    numberBalls = 10;
                    executePhase(phase);
                }
            }
        }
    }
}




function createButtonArray(){
    for (var i = 0; i < buttonArray.length; i++){
        var rect = new Rectangle((getWidth() / 2), (getHeight() / 9));
        if((i == 0) || (i == 3) || (i == 6)){
            rect.setColor(Color.YELLOW);
            rect.setPosition(getWidth() / 4, getHeight() * (6/9));
        }
        if((i == 1) || (i == 4) || (i == 7)){
            rect.setColor(Color.ORANGE);
            rect.setPosition(getWidth() / 4, getHeight() * (4/9));

        }
        if((i == 2) || (i == 5) || (i == 8)){
            rect.setColor(Color.RED);
            rect.setPosition(getWidth() / 4, getHeight() * (2/9));
        }
        buttonArray[i] = rect;
    }
}


/* creates the buttons for selecting the balls' colors, using multiple rectangles
for each button to accurately represent the color palette of each option */
function createColorButtons(){
    for(var i = 0; i <= 2; i++){ //warm
        var rect = new Rectangle((getWidth() / 6), (getHeight() / 9));
        rect.setColor(colorButtonArray[i]);
        switch(i){
            case 0:
                rect.setPosition(getWidth() / 4, getHeight() * (2/9));
                break;
            case 1:
                rect.setPosition(((getWidth() / 4) + ((getWidth() / 2) / 3)), (getHeight() * (2/9)));
                break;
            case 2:
                rect.setPosition(((getWidth() / 4) + ((2 * (getWidth()) / 2) / 3)), (getHeight() * (2/9)));
                break;
        }
        colorButtonArray[i] = rect;
        add(rect);
    }
    for(var i = 3; i <= 8; i++){ //rainbow
        var rect = new Rectangle((getWidth() / 12), (getHeight() / 9));
        rect.setColor(colorButtonArray[i]);
        switch(i){
            case 3:
                rect.setPosition((getWidth() / 4) + (0 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;
            case 4:
                rect.setPosition((getWidth() / 4) + (1 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;

            case 5:
                rect.setPosition((getWidth() / 4) + (2 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;

            case 6:
                rect.setPosition((getWidth() / 4) + (3 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;

            case 7:
                rect.setPosition((getWidth() / 4) + (4 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;

            case 8:
                rect.setPosition((getWidth() / 4) + (5 * (getWidth() / 2) / 6), getHeight() * (4/9));
                break;
        }
        colorButtonArray[i] = rect;
        add(rect);
    }
    for(var i = 9; i <= 11; i++){ //cold
        var rect = new Rectangle((getWidth() / 6), (getHeight() / 9));
        rect.setColor(colorButtonArray[i]);
        switch(i){
            case 9:
                rect.setPosition(getWidth() / 4, getHeight() * (6/9));
                break;
            case 10:
                rect.setPosition(((getWidth() / 4) + ((getWidth() / 2) / 3)), (getHeight() * (6/9)));
                break;
            case 11:
                rect.setPosition(((getWidth() / 4) + ((2 * (getWidth()) / 2) / 3)), (getHeight() * (6/9)));
                break;
        }
        colorButtonArray[i] = rect;
        add(rect);
    }
}


//creatButtonTextArray creates an array containing each text box for all 12 buttons
function createButtonTextArray(){
     for (var i = 0; i < buttonTextArray.length; i++){

        //bottom slot
        if(i == 0){
            buttonTextArray[i] = createText("10 Balls", (getWidth() * (3/8)), (getHeight() * (53/72)));
        }
        if(i == 3){
            buttonTextArray[i] = createText("Small", (getWidth() * (3/8)), (getHeight() * (53/72)));
        }
        if(i == 6){
            buttonTextArray[i] = createText("Slow", (getWidth() * (3/8)), (getHeight() * (53/72)));
        }
        if(i == 9){
            buttonTextArray[i] = createText("Cold", (getWidth() * (3/8)), (getHeight() * (53/72)));
        }

        //middle slot
        if(i == 1){
            buttonTextArray[i] = createText("100 Balls", (getWidth() * (3/8)), (getHeight() * (37/72)));
        }
        if(i == 4){
            buttonTextArray[i] = createText("Medium", (getWidth() * (3/8)), (getHeight() * (37/72)));
        }
        if(i == 7){
            buttonTextArray[i] = createText("Average", (getWidth() * (3/8)), (getHeight() * (37/72)));
        }
        if(i == 10){
            buttonTextArray[i] = createText("Rainbow", (getWidth() * (3/8)), (getHeight() * (37/72)));
        }

        //top slot
        if(i == 2){
            buttonTextArray[i] = createText("1000 Balls", (getWidth() * (3/8)), (getHeight() * (21/72)));
        }
        if(i == 5){
            buttonTextArray[i] = createText("Large", (getWidth() * (3/8)), (getHeight() * (21/72)));
        }
        if(i == 8){
            buttonTextArray[i] = createText("Fast", (getWidth() * (3/8)), (getHeight() * (21/72)));
        }
        if(i == 11){
            buttonTextArray[i] = createText("Warm", (getWidth() * (3/8)), (getHeight() * (21/72)));
        }
    }
}


//creates an array containing each text box necessary for the questions/prompts that the user receives
function createQuestionsArray(){
    questionsArray.push(createQuestion("Select the desired amount of balls."));
    questionsArray.push(createQuestion("Select the desired size of each ball."));
    questionsArray.push(createQuestion("Select the desired base speed of each ball."));
    questionsArray.push(createQuestion("Select the desired color palette for the balls."));
}


// createText function creates text for each button
function createText(text, xcoord, ycoord){
    var txt = new Text(text, "16pt Montserrat");
    txt.setPosition(xcoord, ycoord);
    txt.setColor(Color.BLACK);
    return(txt);
}


//creates a general text box for each prompt/question that the user receives above the buttons
function createQuestion(text){
    var txt = new Text(text, "12pt Montserrat");
    txt.setPosition(0, getHeight() * (1/8));
    txt.setColor(Color.BLACK);
    return(txt);
}

/* The assignColors function creates an array of randomized colors based upon
the number of balls selected by the user, and returns the array to the start
function */
function assignColors(numberBalls, colorPalette){
    var colorArray = [];
    for(var i = 0; i < numberBalls; i++){
        if(colorPalette == "warmColors"){
            var randomWarmColor = Randomizer.nextInt(1, 3);
            colorArray.push(indexColor(randomWarmColor));
        }
        if(colorPalette == "rainbowColors"){
            var randomRainbowColor = Randomizer.nextInt(1, 6);
            colorArray.push(indexColor(randomRainbowColor));
        }
        if(colorPalette == "coldColors"){
            var randomColdColor = Randomizer.nextInt(5, 7);
            colorArray.push(indexColor(randomColdColor));
        }
    }
    return(colorArray);
}


/* given an integer from 1-7, the indexColor function returns the corresponding
color value (sort of an enum key), intended to be used with a random integer
generator with desired values  */
function indexColor(number){
    switch(number){
        case 1:
            return(Color.RED);
        case 2:
            return(Color.ORANGE);
        case 3:
            return(Color.YELLOW);
        case 4:
            return(Color.GREEN);
        case 5:
            return(Color.BLUE);
        case 6:
            return(Color.PURPLE);
        case 7:
            return(Color.CYAN);
    }
}

/* The createBalls function creates however many balls were requested by the
user, assigns each ball a random location within the canvas, assigns a color to
each ball based on the colorArray created by the assignColors function. Each
ball created by the function is then pushed to an array called ballArray for
future use. */
function createBalls(numberBalls, colorArray, ballRadius){
    for(var i = 0; i < numberBalls; i++){
        var ball = new Circle(ballRadius);
        var randomX = Randomizer.nextInt(ballRadius + 1, getWidth() - ballRadius - 1);
        var randomY = Randomizer.nextInt(ballRadius + 1, getHeight() - ballRadius - 1);
        ball.setPosition(randomX, randomY);
        ball.setColor(colorArray[i]);
        add(ball);
        ballArray.push(ball);
    }
}
/* The moveBalls function moves each ball described by the array by a certain
amount on the x-axis and y-axis, as defined by the variable "velocity" which the
the user provides. With each iteration of this function, it checks whether each
ball has begun to touch any side of the canvas. If it has reached one side, it
will then reverse the horizontal or vertical velocity, depending on whether the
ball reached one of the sides, or the top/bottom of the canvas. This function
also makes it so that half the balls go towards positive x and half towards
negative x, and the same for the y-axis to prevent all the balls from going the
same direction initially. Additionally, the function was slightly modified
```

```
to create a more convincing appearance of "random" movement. */
function moveBalls(){
    if(phase == "5end"){
        var randomFlick;
        for(var i = 0; i < ballArray.length; i++){
            var ball = ballArray[i];

            if(ball.getX() + ballRadius >= getWidth()){
                velocityXTracker[i] = -velocity;
                velocityX = -velocity;
                randomFlick = Randomizer.nextInt(1, 2, 3);
                    switch(randomFlick){
                        case 1:
                            break;
                        case 2:
                            velocityX = velocityX - 5;
                            velocityXTracker[i] = velocityX;
                        case 3:
                            velocityX = velocityX - 10;
                            velocityY = -velocityY;
                            velocityYTracker[i] = velocityY;
                    }
            }else{
                if(ball.getX() - ballRadius <= 0){
                    velocityXTracker[i] = velocity;
                    velocityX = velocity;
                    randomFlick = Randomizer.nextInt(1, 2, 3);
                    switch(randomFlick){
                        case 1:
                            break;
                        case 2:
                            velocityX = velocityX + 5;
                            velocityXTracker[i] = velocityX;
                        case 3:
                            velocityX = velocityX + 10;
                            velocityY = -velocityY;
                            velocityYTracker[i] = velocityY;
                    }
                }else{
                    if(moveBallsCounter == 0){
                        if(i % 2 == 0){
                            velocityXTracker[i] = velocity;
                            velocityX = velocity;
                        }
                        if(i % 2 != 0){
                            velocityXTracker[i] = -velocity;
                            velocityX = -velocity;
                        }
                    }else{
                        velocityX = velocityXTracker[i];
                    }
                }
            }
            if(ball.getY() + ballRadius >= getHeight()){
                velocityYTracker[i] = -velocity;
                velocityY = -velocity
                randomFlick = Randomizer.nextInt(1, 2, 3);
                    switch(randomFlick){
                        case 1:
                            break;
                        case 2:
                            velocityY = velocityY - 5;
                            velocityYTracker[i] = velocityY;
                        case 3:
                            velocityY = velocityY - 10;
                            velocityX = -velocityX;
                            velocityXTracker[i] = velocityX;
                }
            }else{
                if(ball.getY() - ballRadius <= 0){
                    velocityYTracker[i] = velocity;
                    velocityY = velocity;
                    randomFlick = Randomizer.nextInt(1, 2, 3);
                    switch(randomFlick){
                        case 1:
                            break;
                        case 2:
                            velocityY = velocityY + 5;
                            velocityYTracker[i] = velocityY;
                        case 3:
                            velocityY = velocityY + 10;
                            velocityX = -velocityX;
                            velocityXTracker[i] = velocityX;
                    }
                }else{
                    if(moveBallsCounter == 0){
                        if(i % 2 == 0){
                            velocityYTracker[i] = velocity;
                            velocityY = velocity;
                        }
                        if(i % 2 != 0){
                            velocityYTracker[i] = -velocity;
                            velocityY = -velocity;
                        }
                    }else{
                        velocityY = velocityYTracker[i];
                    }
                }
            }
            ball.move(velocityX, velocityY)
        }
        moveBallsCounter = 1;
    }
}
```