SASTA OLIPHANT SCIENCE AWARDS

SOUTH AUSTRALIAN SCIENCE TEACHERS ASSOCIATION

Highly Commended

Computer Programming, Apps & Robotics

Year 3-4

Louis Kent

Pembroke School

<u>Project report:</u> Gravity and Wind – by Louis Kent

- <u>**The aim of the entry, and its scientific purpose and potential applications**</u>

The aim of this entry is to use a computer program to demonstrate how gravity and wind interact with each other in a step by step computer simulation.

There are three linked simulations:

1. A ball dropping and bouncing demonstrating the force of gravity in a vacuum from a height of 10 metres where gravity is accelerating at 9.8 metres per second.
2. A ball blowing and bouncing against the wall in the wind in a zero-gravity environment at speed of sixty pixels per second.
3. A combination of scenarios one and two with variable wind speed as determined by user input.

Some applications for this code in the real world might include:

1. Analysing hailstorms; they fall with gravity and there is lots of wind to push the hail around.
2. Dropping food packets in countries requiring aid.
3. Simulating potential damage to crops/fruit on trees orchards in storms.

- <u>**The type of robot or computer required to run the program**</u>

Any web browser (e.g. Chrome, Internet Explorer, Firefox) which supports HTML.

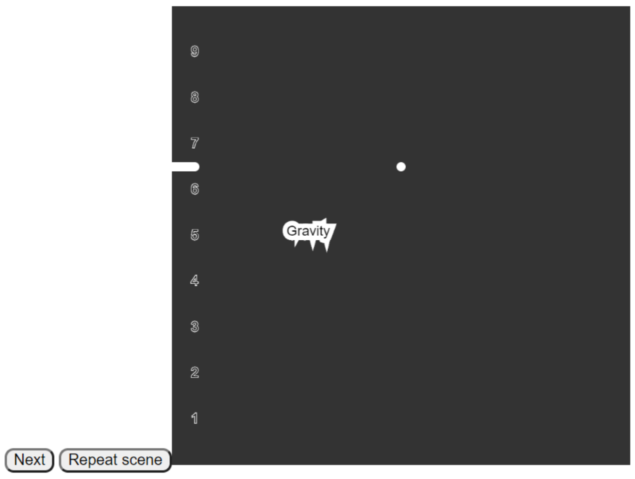
The project is coded in JavaScript because it is a universal language which allows any user who has a web browser to run the program, regardless of whether their operating system is Windows, Linux, or Mac.

The file where all the information for the particles/balls is stored is called PARTICLE.JS.

- <u>**Clear instructions on loading or using the entry**</u>

Instructions:

1. *Turn on your laptop/desktop/ipad/iphone/phone/smartwatch.*
2. *Click on this link:* https://filedn.com/lyEeHFgLQpLu1vUo9KCzXYk/index.html.
3. *(i) The first animation shows a ball dropping from ten metres where gravity is 9.8 metres per second,*



(ii) No. 2 shows wind

(iii) No. 3 shows both (Press and hold mouse to use wind.).

4. *Watch and press next when you have finished looking.*
5. *Enjoy!*
6. *Go here:* https://filedn.com/lyEeHFgLQpLu1vUo9KCzXYk/sketch.js *or* https://filedn.com/lyEeHFgLQpLu1vUo9KCzXYk/particle.js to view the code (shown below in Appendix – Code).
7. *The dependencies and meanings of gravity and wind are here (source – WIKIPEDIA):* https://filedn.com/lyEeHFgLQpLu1vUo9KCzXYk/dependencies.txt

Appendix – Code

Sketch.js:

```
/*
step 1 - Initialize the button which will allow the user to move to the
next scene.
*/

let nextButton;

/*step 2 - scene setup for defining the phenomena we will be simulating
â€¢ SCENE 1 will be gravity;
â€¢ SCENE 2 will be wind;
â€¢ SCENE 3 will be gravity and wind
*/
let scene1 = true;
let scene2 = false;
let scene3 = false;

//Time variable to keep track of when to reset to the middle because it is the next scene.
let t = 0;




let ball;

function setup() {
  nextButton = document.getElementById("next");
  //Make a canvas
  createCanvas(400, 400);

  //Change scenes when the button is clicked
  nextButton.addEventListener("click", () => {
    if (scene1) {
      scene2 = true;
      scene1 = false;
    } else if (scene2) {
      scene3 = true;
      scene2 = false;
      nextButton.style.visibility = "hidden";
    }
  });

  //Make a new Particle in the middle of the screen
  ball = new Particle(width / 2, 0);
}

function draw() {
  //When it is scene one do this:
  if (scene1) {
    background(51);
    //initialize gravity
    let force = createVector(0, 1);
    //animate and show
    ball.animate();
    ball.bounce();
    ball.display();
    //apply gravity
    ball.addForce(force);
    line(0, ball.pos.y, 20, ball.pos.y);
    for (let i = 9; i >= 1; i--) {
      push();
      strokeWeight(1);
      //Make text
      textAlign(CENTER, CENTER);
      if (i !== -1) {
        text(i, 20, height - (i * 40));
      }
      pop();
    }
    text("Gravity", width/ 4, height / 2);
    //When scene two do this:
  }
  else if (scene2) {
    background(51);
    if (t == 0) {
      ball.pos.x = width / 2;
      ball.pos.y = 0;
      ball.acc.set(0, 0);
      ball.vel.set(0, 0);
      t += 1;
    }
    //initialize wind
    let force = createVector(1, 0);
    //animate and show
    ball.animate();
    ball.bounce();
    ball.display();
    //apply wind
    ball.addForce(force);
    //draw text
    text("Wind", height / 4, width / 2);
    //When scene three do this:
  }
  else {
    background(51);
    if (t == 1) {
      ball.pos.x = width / 2;
      ball.pos.y = 0;
      ball.acc.set(0, 0);
      ball.vel.set(0, 0);
      t -= 1;
    }
    let g = createVector(0, 1);
    if (mouseIsPressed) {
      let w = createVector(1, 0);
      ball.addForce(w);
    }
    ball.animate();
    ball.bounce();
    ball.display();
    ball.addForce(g);
    text("Gravity and wind (Press mouse to use wind)", height / 4, width / 2);
  }
}
```

Particle.js:

```
/*A constructor/object class to make a particle.
 (It is an es2015 / es6 feature)
 */
class Particle {
  constructor(x, y) {
    /*
    Initializing the position,
    the velocity and the acceleration of the Particle
    */
    this.pos = createVector(x, y);
    this.vel = createVector(0, 0);
    this.acc = createVector(0, 0);
  };
  //A function/method to update and move the Particle
  animate() {
    this.vel.add(this.acc);
    this.pos.add(this.vel);
    this.acc.set(0, 0);
  };

  //A function/method to show the Particle as a point on the screen.
  display() {
    strokeWeight(8);
    stroke(255);
    point(this.pos.x, this.pos.y);
  };

  //Make the ball bounce!
  bounce() {
    if (this.pos.x <= 0) {
      this.pos.x = 0;
      this.vel.x *= -1;
    } else if (this.pos.x >= width) {
      this.pos.x = width;
      this.vel.x *= -1;
    } else if (this.pos.y <= 0) {
      this.pos.y = 0;
      this.vel.y *= -1;
    } else if (this.pos.y >= height) {
      this.pos.y = height;
      this.vel.y *= -1;
    }
  };

  // Add a force to the Particle. (eg. wind, gravity, friction, centrifugal)
  addForce(force) {
    this.acc.add(force);
  };
}
```