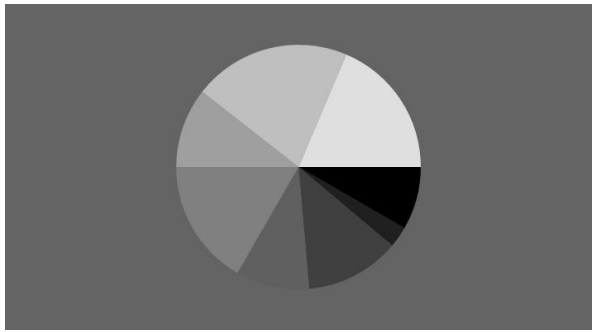
p5.js

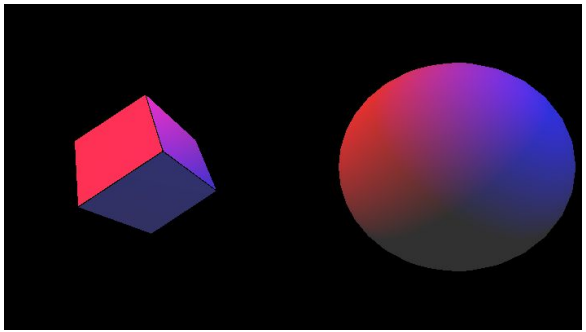
An accessible and lightweight graphics library

Introduction

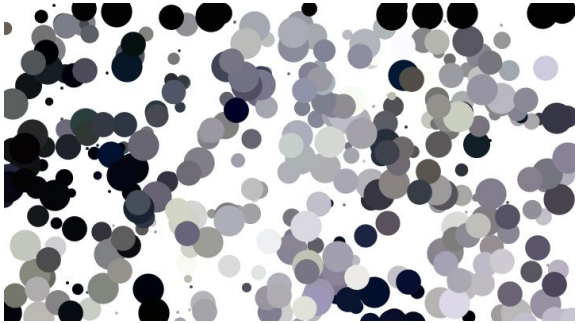
- https://p5js.org/
- Free and open source library adding drawing functionality to a web page
- Designed with a simple, but flexible API
- Includes support for 2D, 3D, text, input, video, and sound



https://p5js.org/examples/form-pie-chart.html



https://p5js.org/examples/3d-multiple-lights.html



https://p5js.org/examples/image-pointillism.html

Setup

- Editor available online: https://editor.p5js.org/
  - Node package available to integrate into other apps
- setup() function is called once to initialize
- createCanvas(400, 400) creates a 400 by 400 pixel area to draw on
- draw() function is called repeatedly to create the scene
  - All drawing related code should go here

```
1  function setup() {
2    createCanvas(400, 400);
3  }
4
5  function draw() {
6    background(220);
7  }
```

2D Drawing

- Several types of shapes can be drawn with a single function call
  - rect(), circle(), line(), arc(), ellipse(), etc.
- First two arguments are the x,y position
- Width and height, radius, or second x,y position come afterwards
- fill() function defines the RGB colour to use for the interior of the shape
- stroke() function defines the RGB colour for the border of the shape

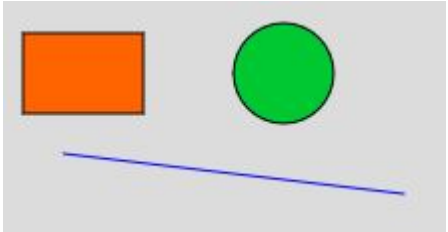
```
function setup() {
  createCanvas(400, 400);
}

function draw() {
  background(220);
  stroke(0, 0, 0)

  // Rectangle
  fill(255, 100, 0)
  rect(20, 20, 60, 40)

  // Circle
  fill(0, 200, 50)
  circle(150, 40, 50)

  // Line
  stroke(0, 0, 255)
  line(40, 80, 210, 100)
}
```



Keyboard Input

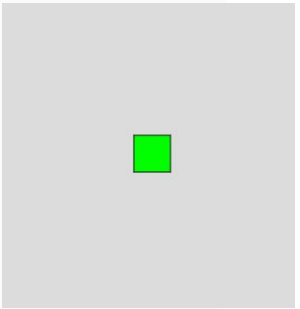
- The keyPressed() and keyReleased() functions can be used to detect keyboard input
- The keyCode variable holds the key which triggered the event
- A square was created for this example, which moves with the arrow keys
- keyIsDown() can also be used in the draw function

```
function setup() {
  createCanvas(400, 400);
  this.x = 200
  this.y = 200
  this.xvel = 0
  this.yvel = 0
}

function draw() {
  background(220);
  fill(0, 255, 0)
  this.x += this.xvel * 2
  this.y += this.yvel * 2
  rect(this.x - 20, this.y - 20, 40, 40)
}

function keyPressed() {
  if (keyCode == UP_ARROW) {
    this.yvel = -1
  } else if (keyCode == DOWN_ARROW) {
    this.yvel = 1
  } else if (keyCode == LEFT_ARROW) {
    this.xvel = -1
  } else if (keyCode == RIGHT_ARROW) {
    this.xvel = 1
  }
}

function keyReleased() {
  if (keyCode == UP_ARROW || keyCode == DOWN_ARROW) {
    this.yvel = 0
  } else if (keyCode == LEFT_ARROW || keyCode == RIGHT_ARROW) {
    this.xvel = 0
  }
}
```



Vue Integration

- A few more steps required
  - `npm install –save p5` ← Command line
  - `import p5 from “p5”;` ← Vue component
- Create sketch object containing desired built-in p5 functions which will be automatically called
  - The p5 context is passed into the lambda
- All p5 functions are now part of the s object rather than in global scope
- p5Context variable can be stored within the Vue page

```
const sketch = (s) => {
    s.setup = () => {
        // ...
    }

    s.draw = () => {
        // ...
    }
}
```

```
const p5Context = new p5(sketch);
```

Vue Integration – Our Project

- The following games all use the techniques specified in the last slide to integrate with Vue:
  - Snake
  - Word Search
  - Stacker
- The code is present in the following files:
  - src/components/SnakeGame.vue
  - src/components/WordSearchGame.vue
  - src/components/StackerGame.vue
- All game-related code is contained within a class for better organization