My Generative Clock

Amir Ghorbani

November 2020

Y1 USE18103 Coding, Figures, Visuals 20-21

Tutor: Nick Rothwell.

vvv

The Clock’s sound reactive feature only works in p5.js editor.

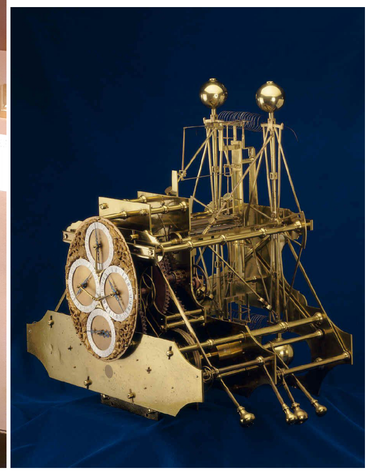
https://editor.p5js.org/AmirGhorbani/sketches/Sh4Zlg_AI

Design document

Museum

Natural History Museum

Royal Observatory Greenwich



Skeleton clock:
- It reminds me of Philippe Starck lemon squeezer.
- The designer exposes the mechanical part of the clock.

John Harrison's first marine timekeeper:

- sea clock

- It took 5 years to build. In 1736, it was tested on a sea voyage to Lisbon and back.

- Similar design to skeleton clock.

George Daniel is the greatest watchmaker of the 20th century.
Interface of his designs:
- Simple shapes (circles).
- Different compositions of primary shapes.
- Mostly symmetrical
- Complicated mechanical watches come with a minimal interface.

Space Traveller's Watch by George Daniel:
- It shows mean solar time and sidereal (star) time.
- Once joked ' When you are on your package tour to Mars you need a watch like this'.
- It is easy to simplify the interface of his clocks to primary shapes (mostly circles) and ignore other components.

Solar time clock by DE FOSSARD:

- Time of sunrise.
- Time of local solar noon.
- Time of sunset.
- Adjustable for almost any longitude and latitude.
- Greenwich Mean Time.
- The phase of the moon.
- A year going calendar.
- Human characteristics.
- looks like a Fuel pump.
- Mostly decorative rather than practical.

Solar time clock by DE FOSSARD VS Space Traveller's Watch:

Similarities:

1- Use of primary geometric shape (circle)
2- Minimal design
3- Multifunctional
4- On Some occasions it is not practical.

Differences:

1- different sizes
2- One pocket watch and the other stationary clock.
3- De Fossard design is more futuristic.
4- Daniel's design is elegant.
5- De Fossard's design exposes the mechanical parts.


S. Riefler
Glashütte


PEDOMETER BY BENJAMIN GRAY
A pedometer in a silver case, recording up to a maximum of 100 miles. It is engraved 'Made by B. Gray, clockmaker to George 3rd.' Gives to the Company of Clockmakers by his great grandson B.L. Vulliamy 1887.
Museum No. 667. Presented by B.L. Vuli


THE STANDARD TIME & TELEPHONE CO LIMITED
SYNCHRONIZED CLOCK (LUNDS PATENT
DENT
London


WATCH BY A-L BREGUET
France, c.1800
Gold case. White enamel 'souscription' dial and hour hand. Signed 'Breguet', also with secret signature. Ruby cylinder escapement of Breguet's special type. Compensation curb. Brass balance. Movement signed 'Breguet No. 518'. An early example of this type.
Museum No. 368. Presented by Miss V.L. Blackman, 1955


E. DENT & CO.
LONDON


E. DENT & CO.
LONDON


DENT,
Clock Makers
LONDON.
to the Queen.

Generative Art

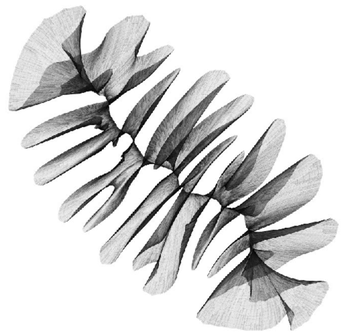
Generative Art is a process of algorithmically generating new ideas, forms, shapes, colours or patterns. First, you create rules that provide boundaries for the creation process. Then a computer (or less commonly a human) follows those rules to produce new works.
Generative code artists use computers to generate thousands of ideas in milliseconds.

https://www.behance.net/manoloide

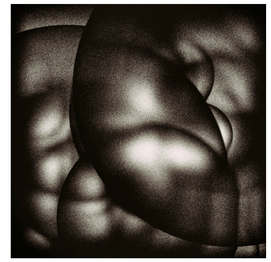
http://www.michael-hansmeyer.com/

Anders Hoff
https://inconvergent.net/

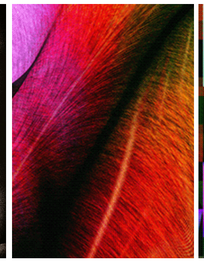
Repetition of primary shapes (mostly circle)
Simple geometric shapes in different sizes and colours
This work is computer generated. (algorithmic tools including Processing)

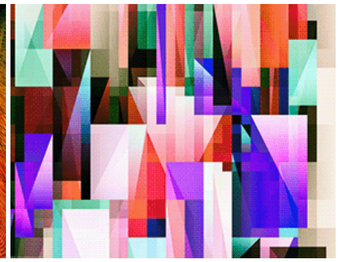


Rotation is the main property of this art work. (Computer generated)

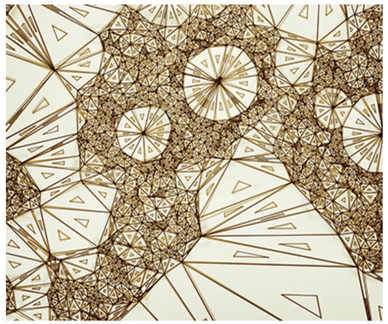


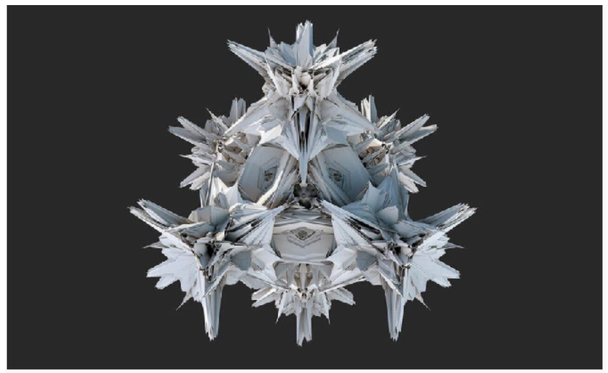
Transformation of a mass

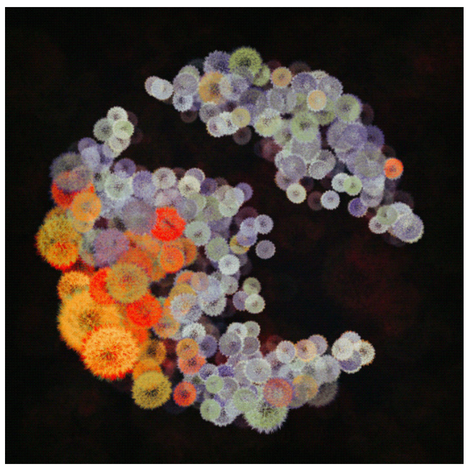


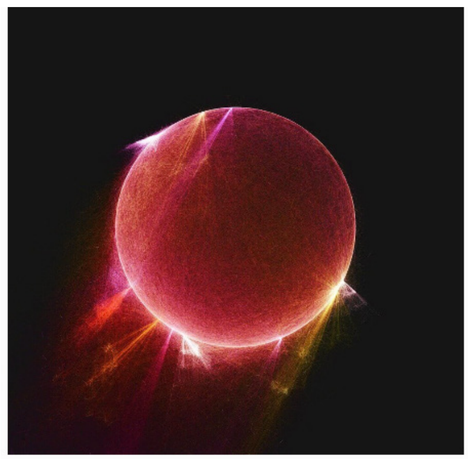


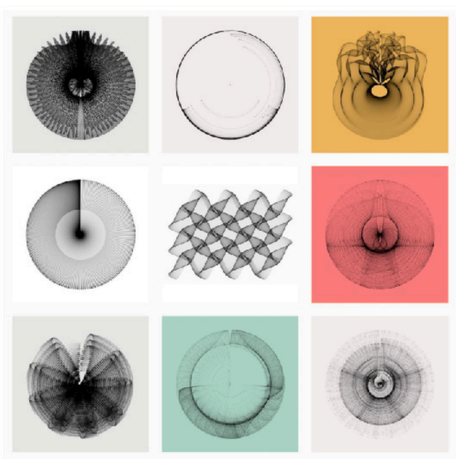
Repetition of colourful triangles

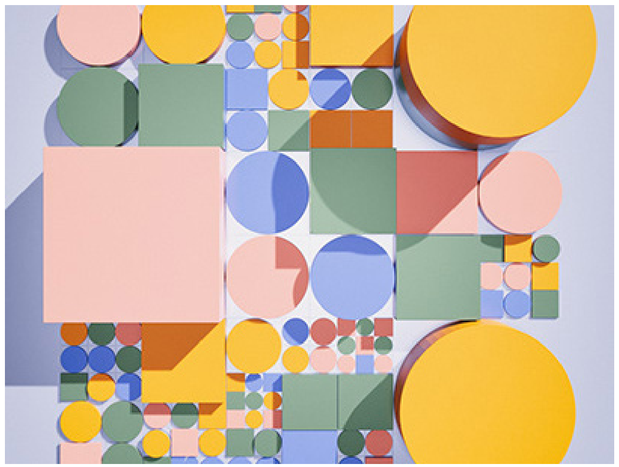












Study of Sine and Cosine

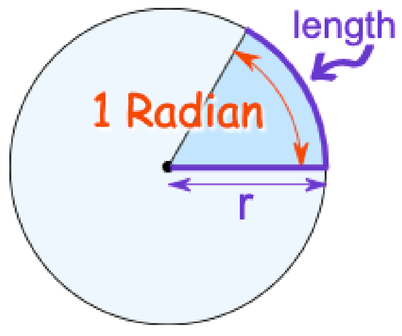
Why are sine and cosine important in my project?

I have used sine and cosine to define the button area for the mouse hover effect and also the mouse click event.

3.1416... * Radians = π * Radians = 180°
1 radian = 180°/π = 57.2958...° (approximately)

Radians Preferred by Mathematicians: Because the radian is based on the pure idea of "the radius being laid along the circumference", it often gives simple and natural results when used in mathematics.

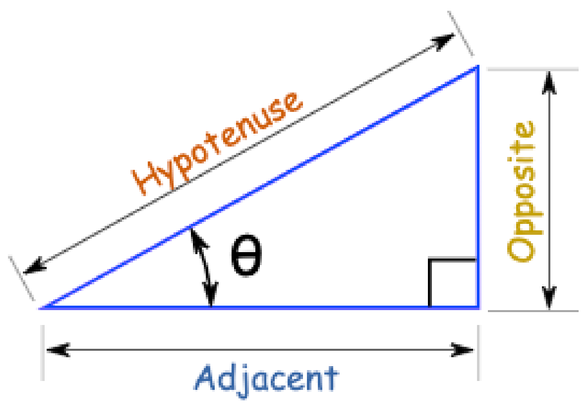
Degrees are easier to use in everyday work, but radians are much better for mathematics.



$$\sin\theta = \frac{\text{Opposite}}{\text{Hypotenuse}}$$

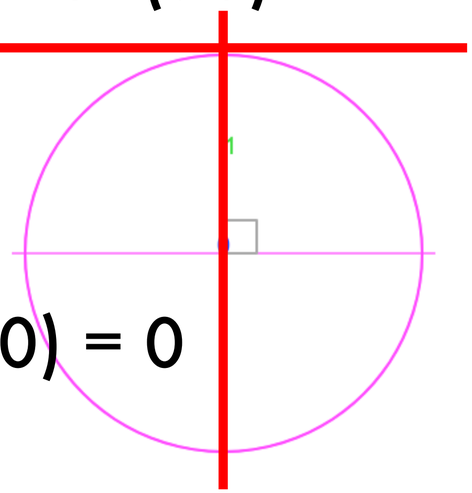
$$\cos\theta = \frac{\text{Adjacent}}{\text{Hypotenuse}}$$

$$\tan\theta = \frac{\text{Opposite}}{\text{Adjacent}}$$

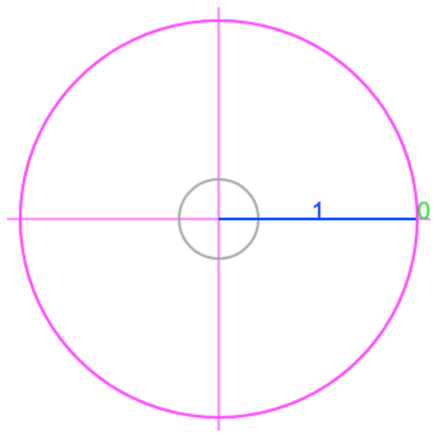


sin(90) = 1

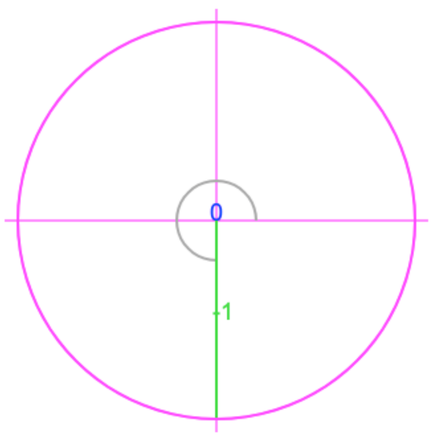
cos(90) = 0



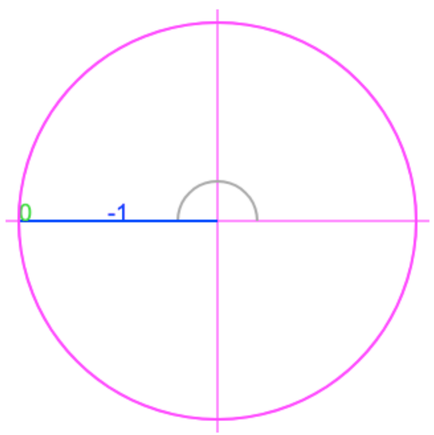
sin(0) = 0
cos(0) = 1



sin(270) = -1
cos(270) = 0



sin(180) = 0
cos(180) = -1



Some examples of clock interface

Because of the small dimensions of an ordinary clock or watch, normally the controllers are difficult to use.
(small buttons or rotary controllers)

My design journey

- THE DESIGN OF FUTURE THINGS:
Some clocks are obstinate. Others are temperamental. some are delicate, some rugged. We can apply human attributes to our clock, and often these terms are fittingly descriptive, even though we use them as metaphors or similes.
Autonomous clock: creates its own assessments, makes its own decisions.

- DOET:
A few decades ago, clocks were simple. All you had to do was set the time and keep the clock wound. The standard control was stem: a knob. Turning the knob would wind the string that provided power to the watch movement. Pulling out the knob and turning it rotated the hands. The operations were easy to learn and easy to do.

Over time, with the introduction of digital technology, accurate clocks were incorporated in many appliances, from phones to musical keyboards. They became fashion accessories, where one changed the watch with each change in activity and each change of clothes.

The technology has allowed more functions: the watch can give the day of the week, the month, and the year; it can act as a stopwatch (which itself has several functions), a countdown timer, and an alarm clock (or two); it has the ability to show the time for different time zones; it can act as a counter and even as a calculator. A radio receiver to allow it to set its time with official time stations around the world. built in compasses and barometers, accelerometers, and temperature gauges. Some have GPS and Internet receivers so they can display the weather and news, e-mail messages, and the lates from social networks. Some have built in cameras. Some work with buttons, knobs, motion, or speech. Some detect gestures. The watch is no longer just an instrument for telling time: it has become a platform for enhancing multiple activities and lifestyles.
How can all these functions fit into a small screen?

Time:
In some cultures, time is represented mentally as if it were a road stretching out ahead of the person. As a person moves through time, the person moves forward along the timeline. Other cultures use the same representation, except now it is the person who is fixed and it is time that moves: an event in the future moves toward the person.
Some cultures represent the time line vertically: up for the future, down for the past. Other cultures have rather different views. For example, does the future lie ahead or behind? to most of us, the question makes no sense: of course, the future lies ahead - the past is behind us. We speak this way, discussing the "arrival" of the future; we are pleased that many unfortunate events of the past have been "left behind." But the way South American Indian group, the Aymara, represents time. When they speak of the future, they use the phrase back days and often gesture behind them. It is a perfectly logical way to view the world.

Random Notes:
- Looking at the time from a psychological perspective.
- How we perceive things changing around us.
- Natural rhythms and repetitions.
- Perception of the passage of time.
- Day & the night cycle - waxing & the waning of the moon - seasons - sun - tropical year (Nile) - Analemma (shape of the sun movement) - Position of the stars (Side-aerial year) - Lunar solar calendar.
- Roman calendar.
- Sundial / shadow stick
- Water clocks.
- First dial clocks had the full 24hrs around them.
- World split up into 15 segments.
- Quartz clock - Atomic clock - Digital clock

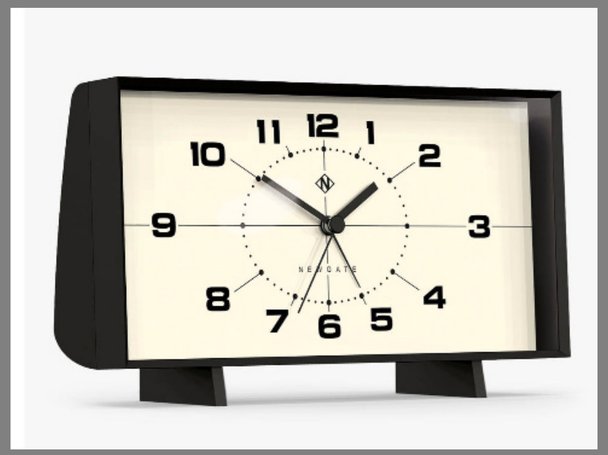
Ideas:
- A clock that does not tell you the time.
- How to develop a clock appropriate to a person's emotions.
- Time shows on glasses.
- A joker.
- Tobar magic 8 ball

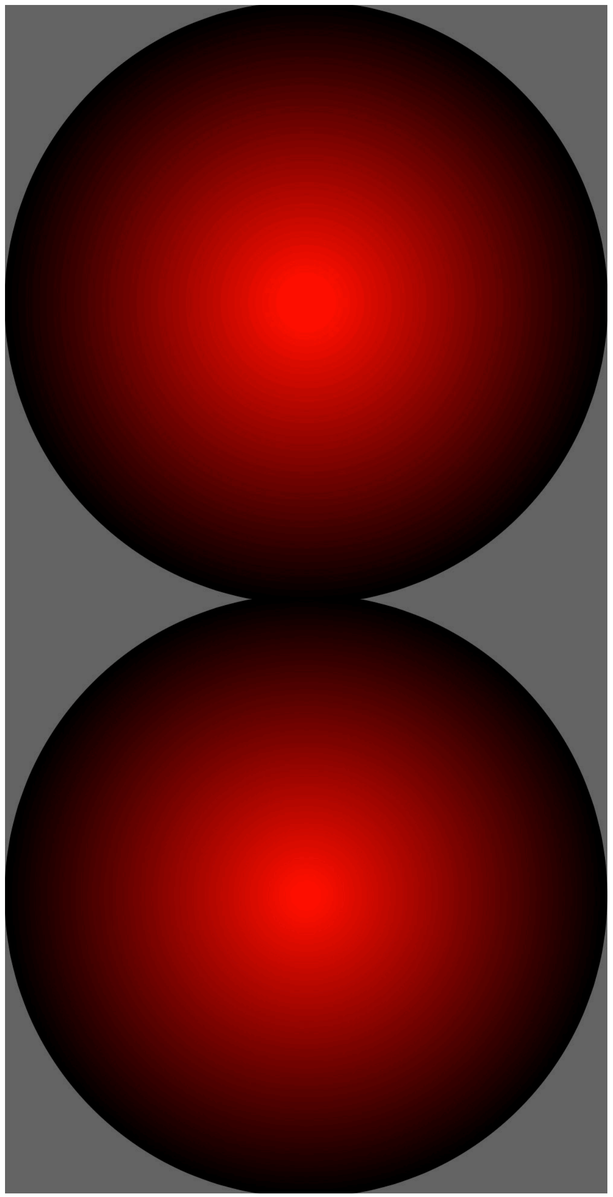
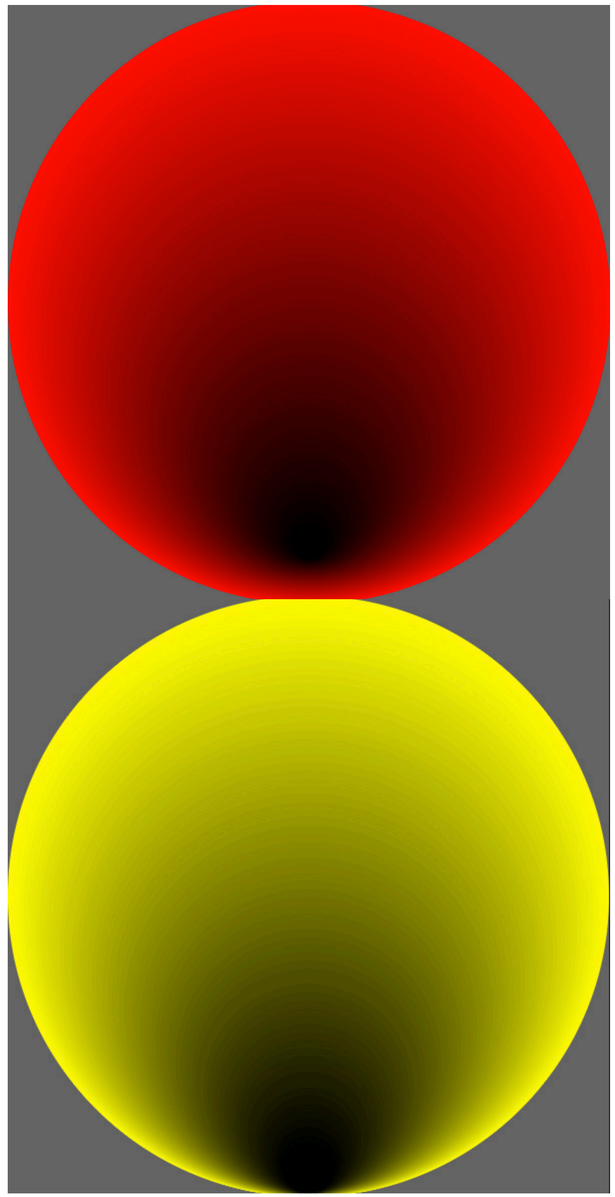
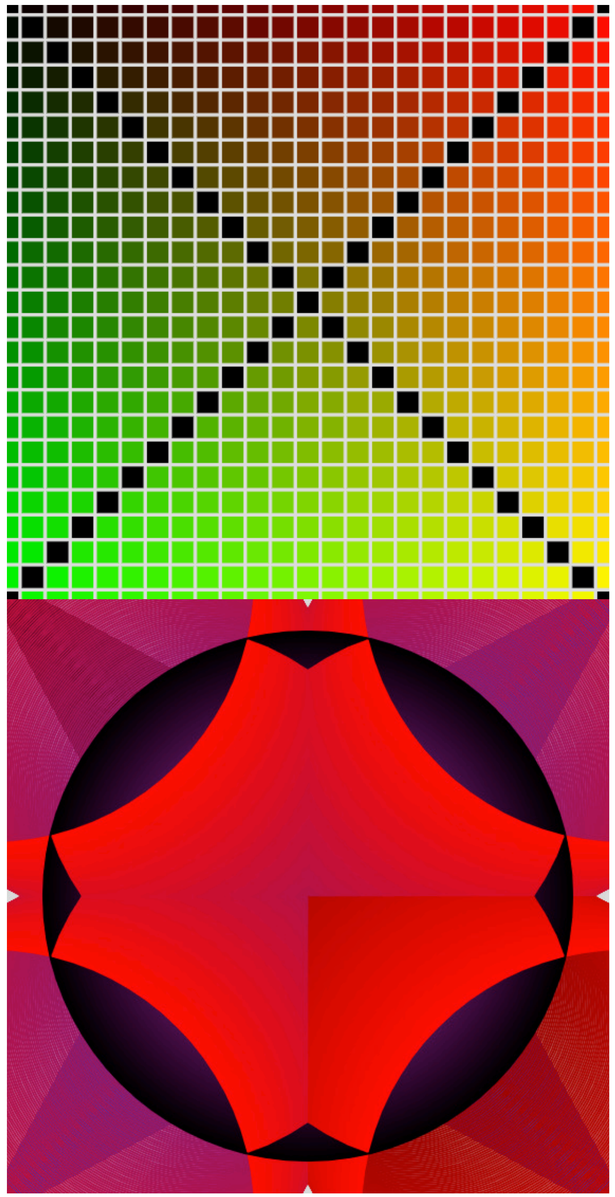
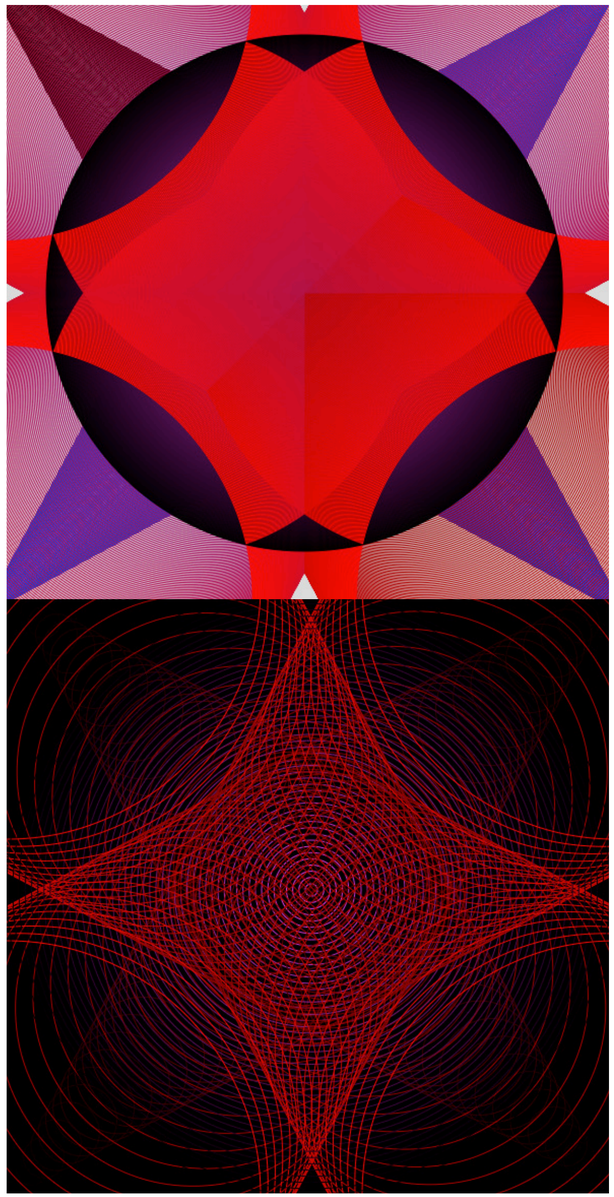
A few decades ago, clocks were simple.

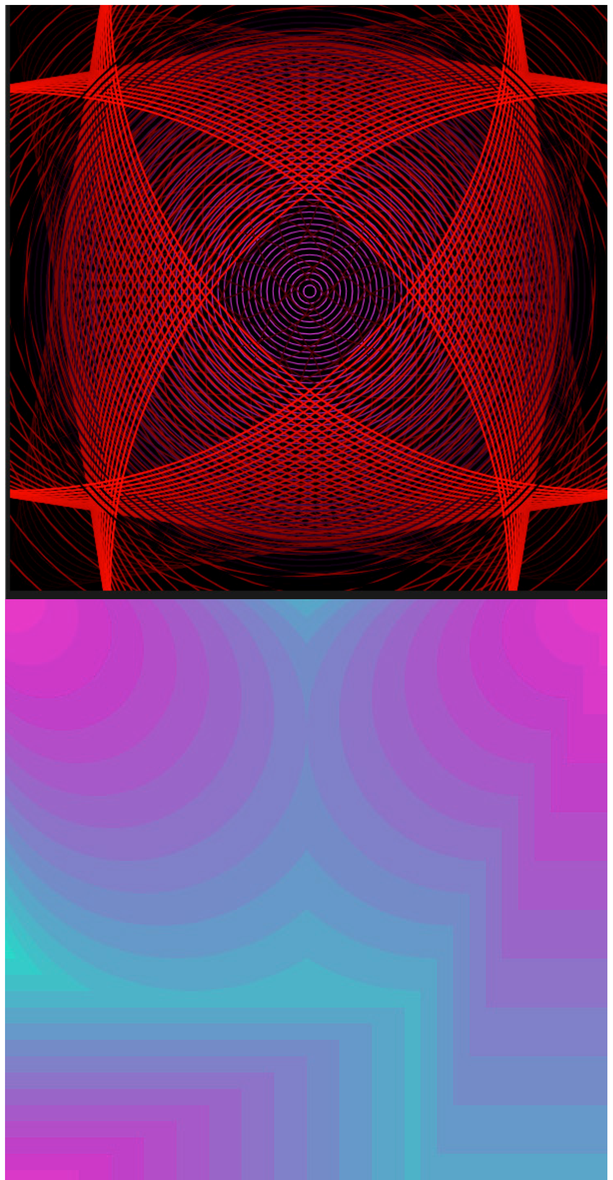
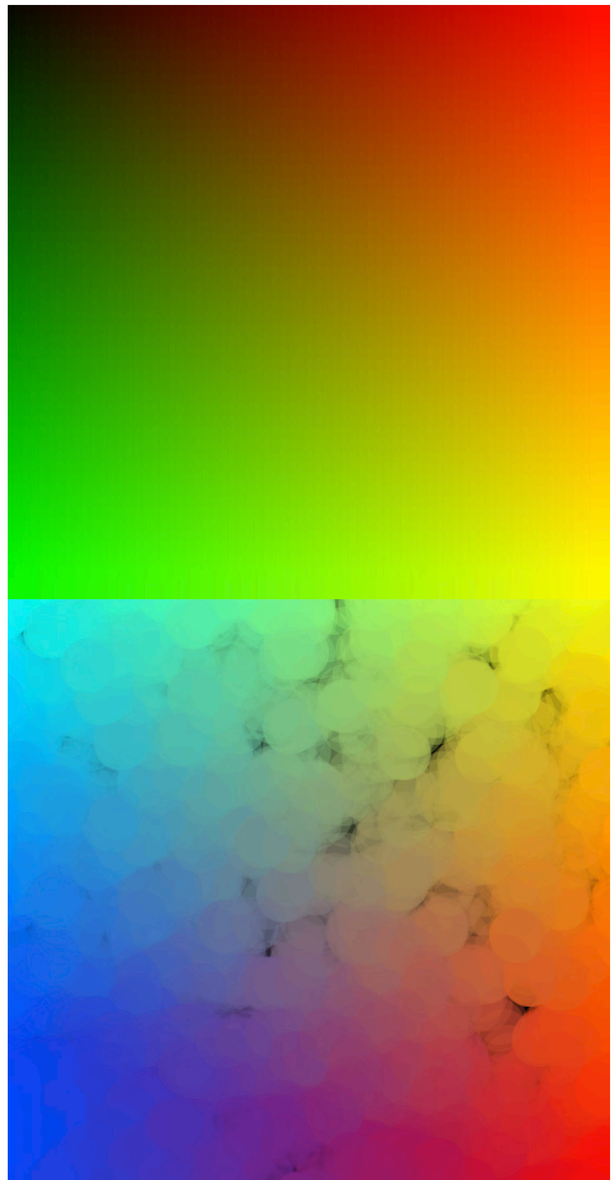
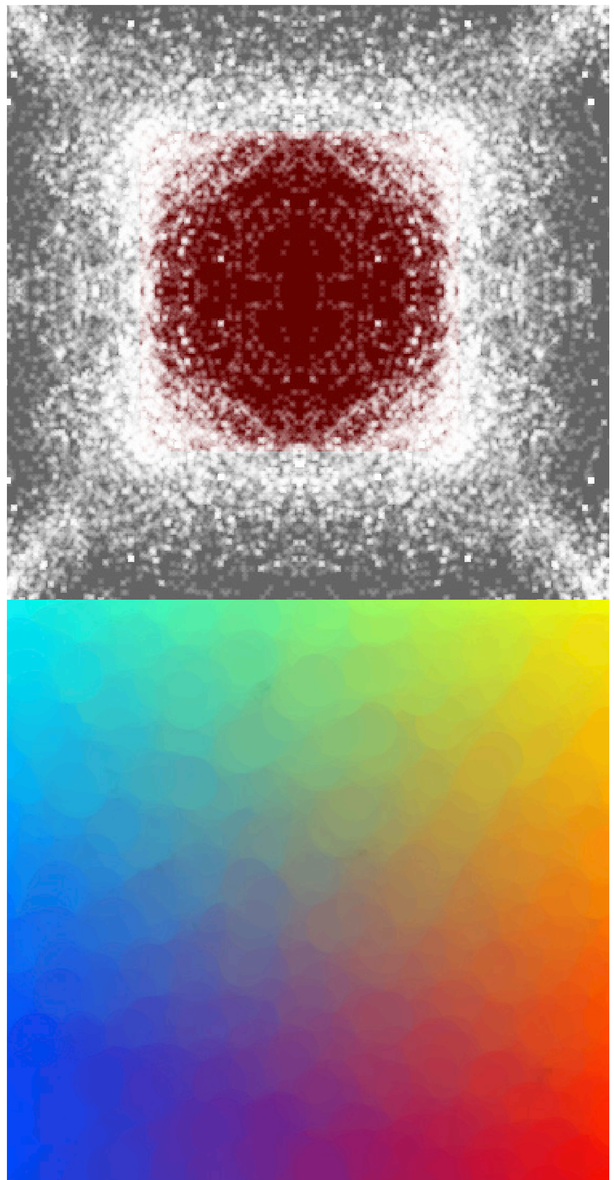
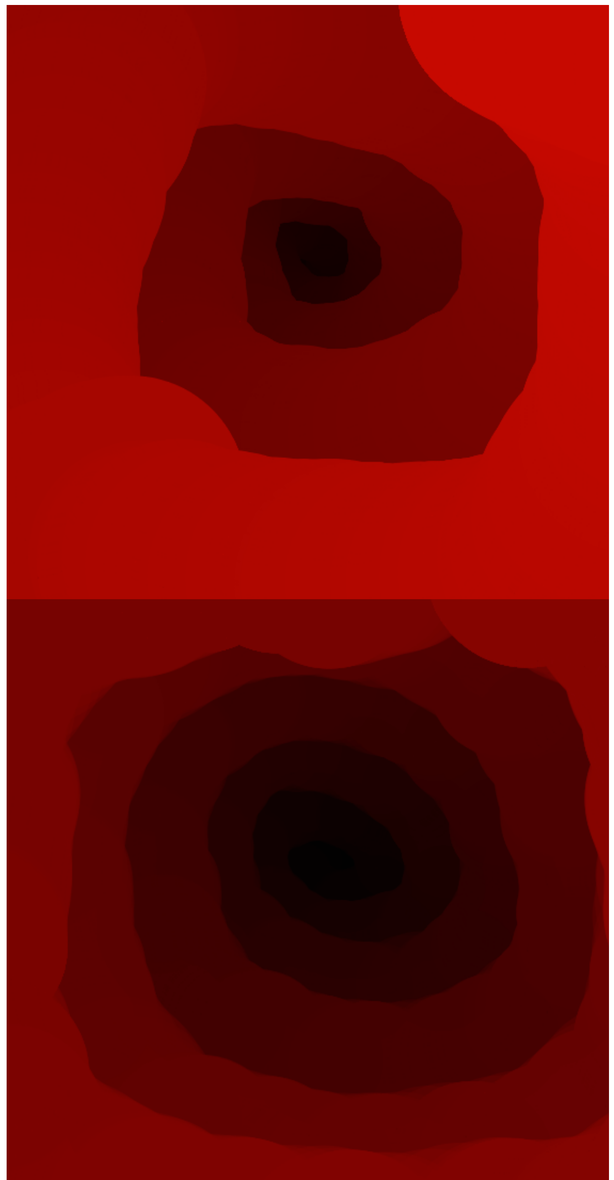
The technology has allowed more functions:
Week
Month
Year
Stopwatch
Countdown timer
Alarm clock
Different time zones
Counter
Calculator
Radio
Compasses
Barometers
Accelerometers
Temperature gauges
GPS
Internet
Weather
News
E-mail messages
Social networks
Cameras
Buttons, knobs,
Motion
Speech
Detect gestures
A platform for enhancing multiple activities and lifestyles.

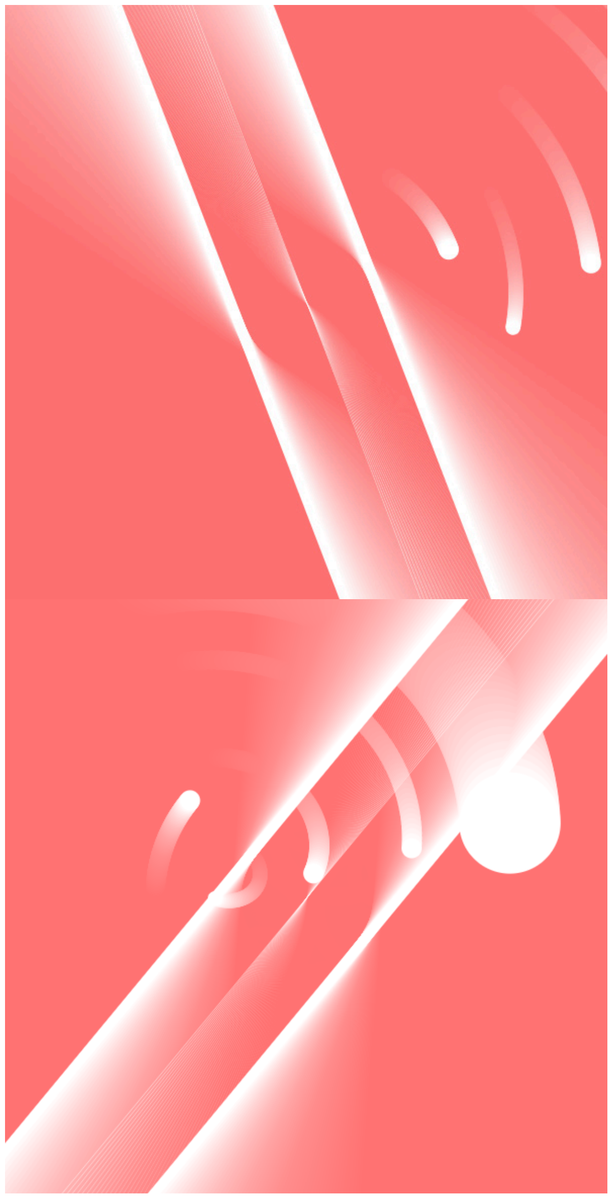
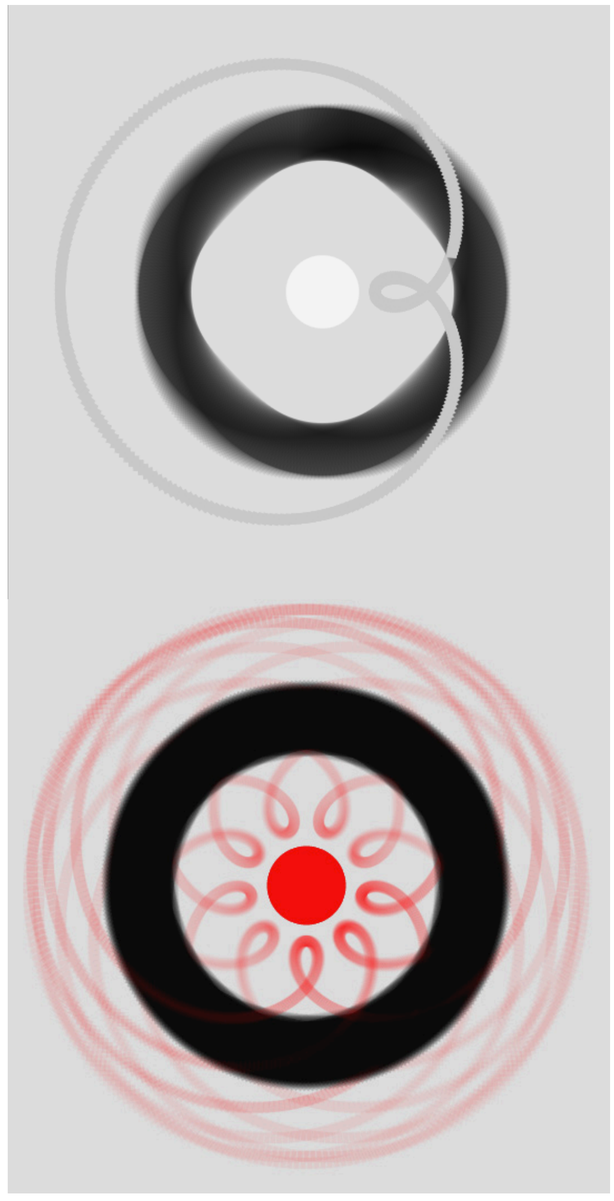
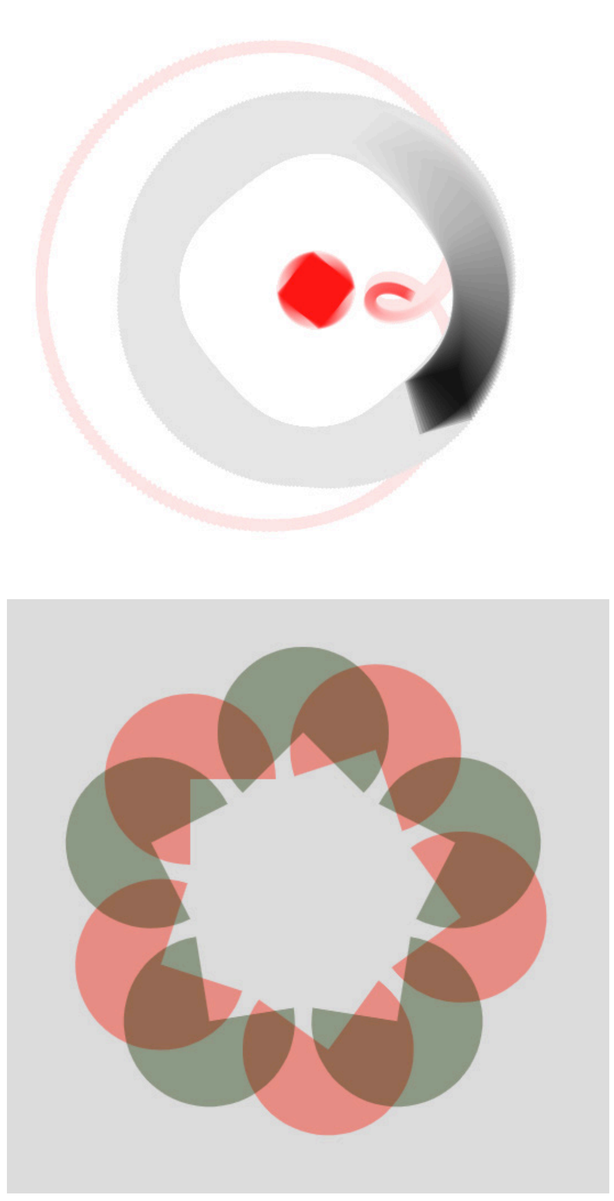
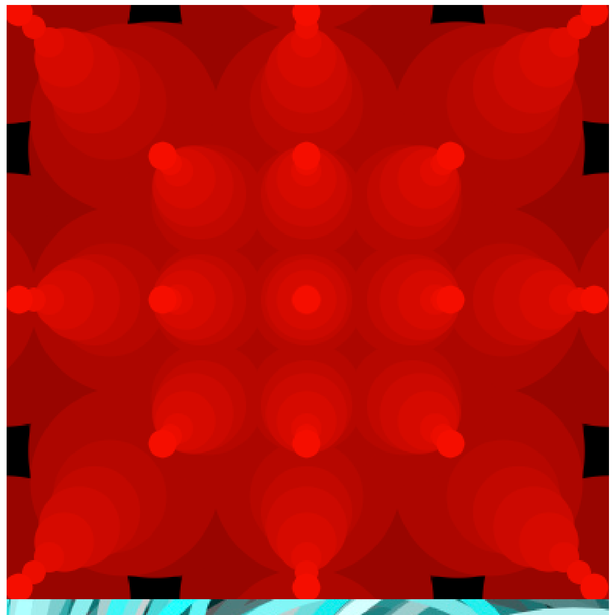
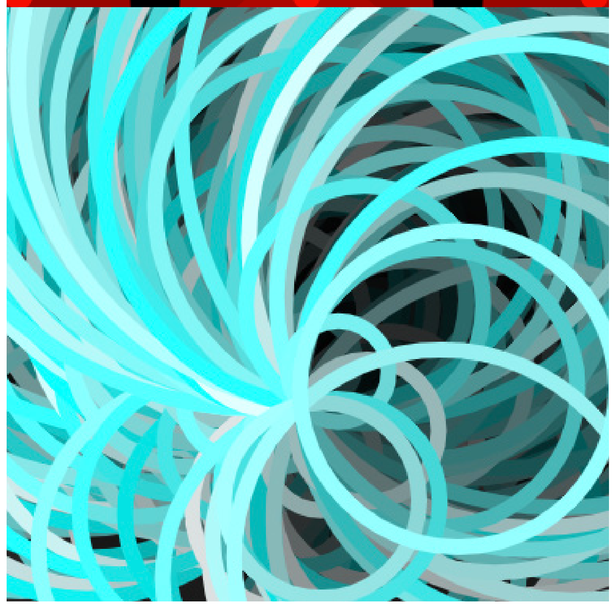
Motion
Speech
Detect gestures
TimeZone

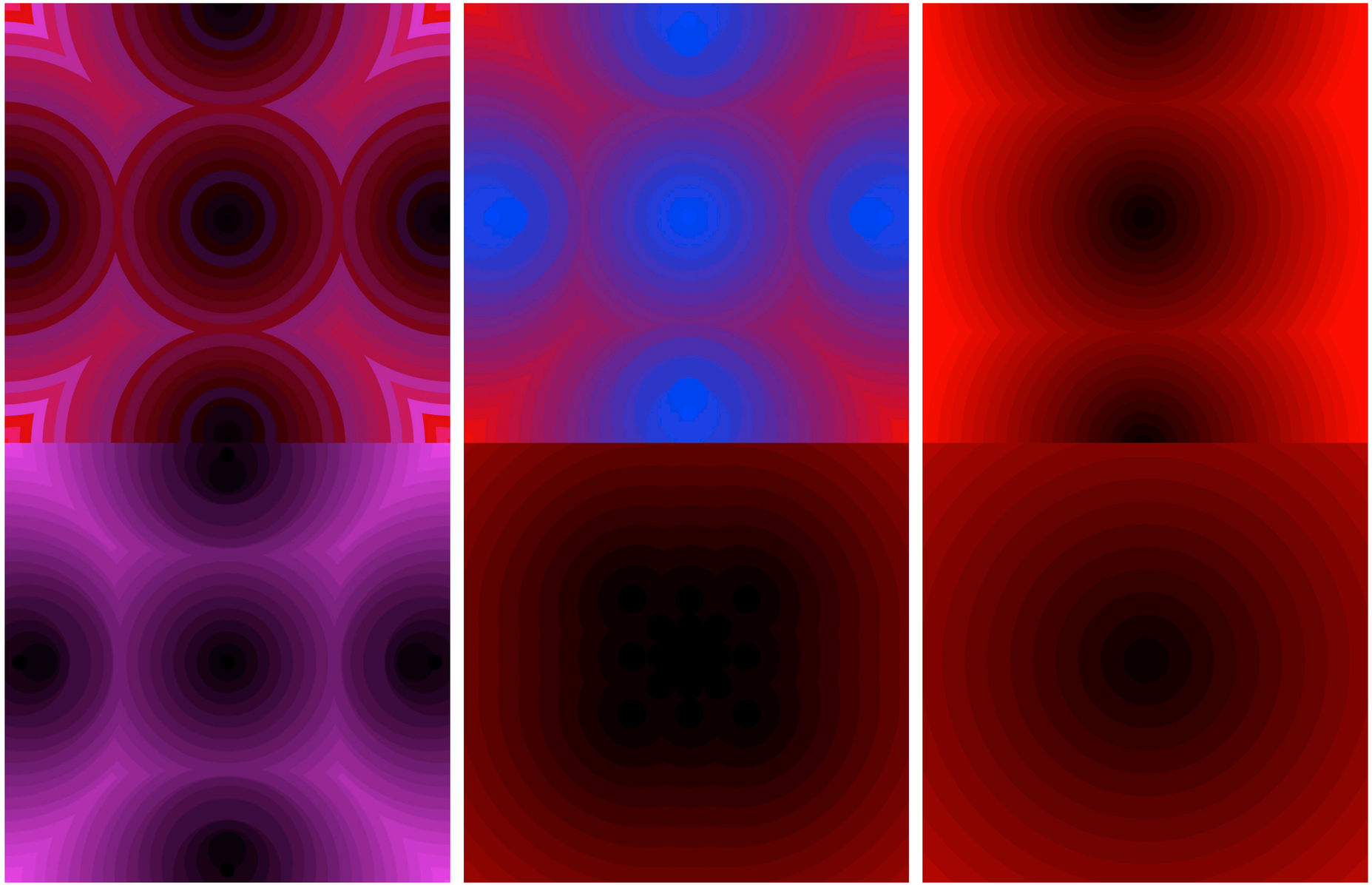
p5.js code

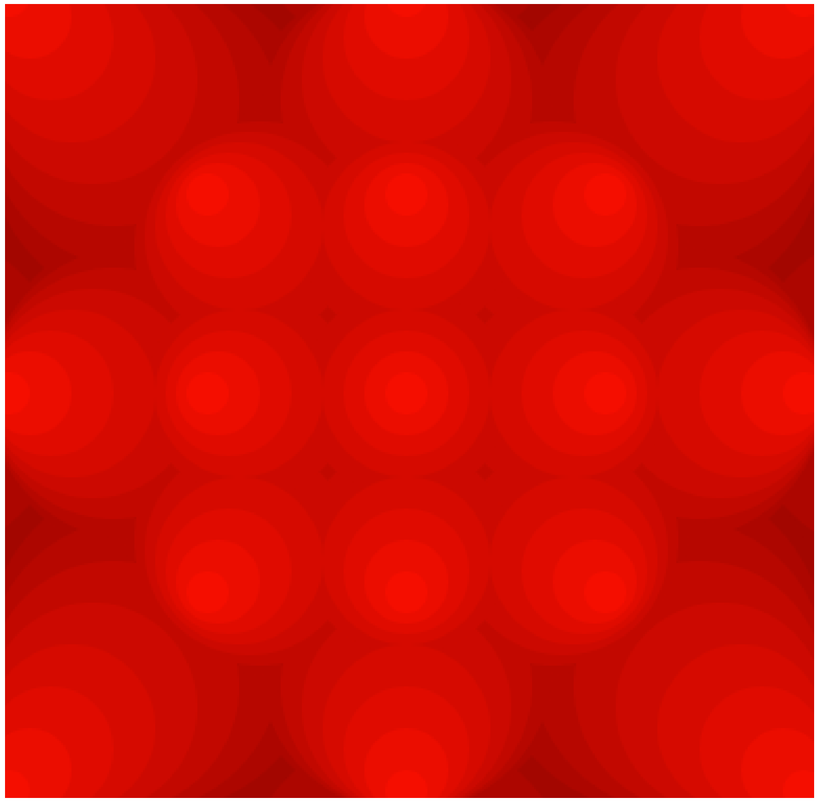






```
10 function draw() {
11   background(0)
12
13   for (var n of numbers) {
14     var size = map(n, 0, 20, 400, 0);
15     var offset = map(n, 0, 20, 0, 400);
16     var red = map(n, 0, 20, 50, 255);
17
18     fill(red, 0, 0);
19     circle(width / 2 + offset, height / 2 - offset, size);
20     circle(width / 2 + offset/2, height / 2 - offset/2, size);
21     circle(width / 2 + offset/4, height / 2 - offset/4, size);
```

17 Circles

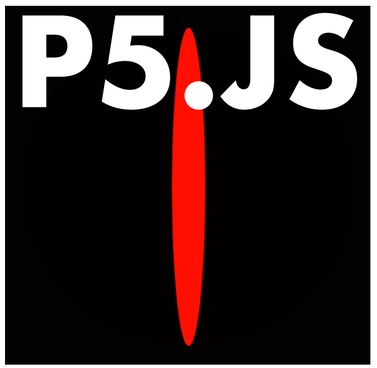
Mapping: 20 times
Size: 400px - 0
Offset: 0 - 400px
Colour: (50, 0, 0) - (255, 0, 0)

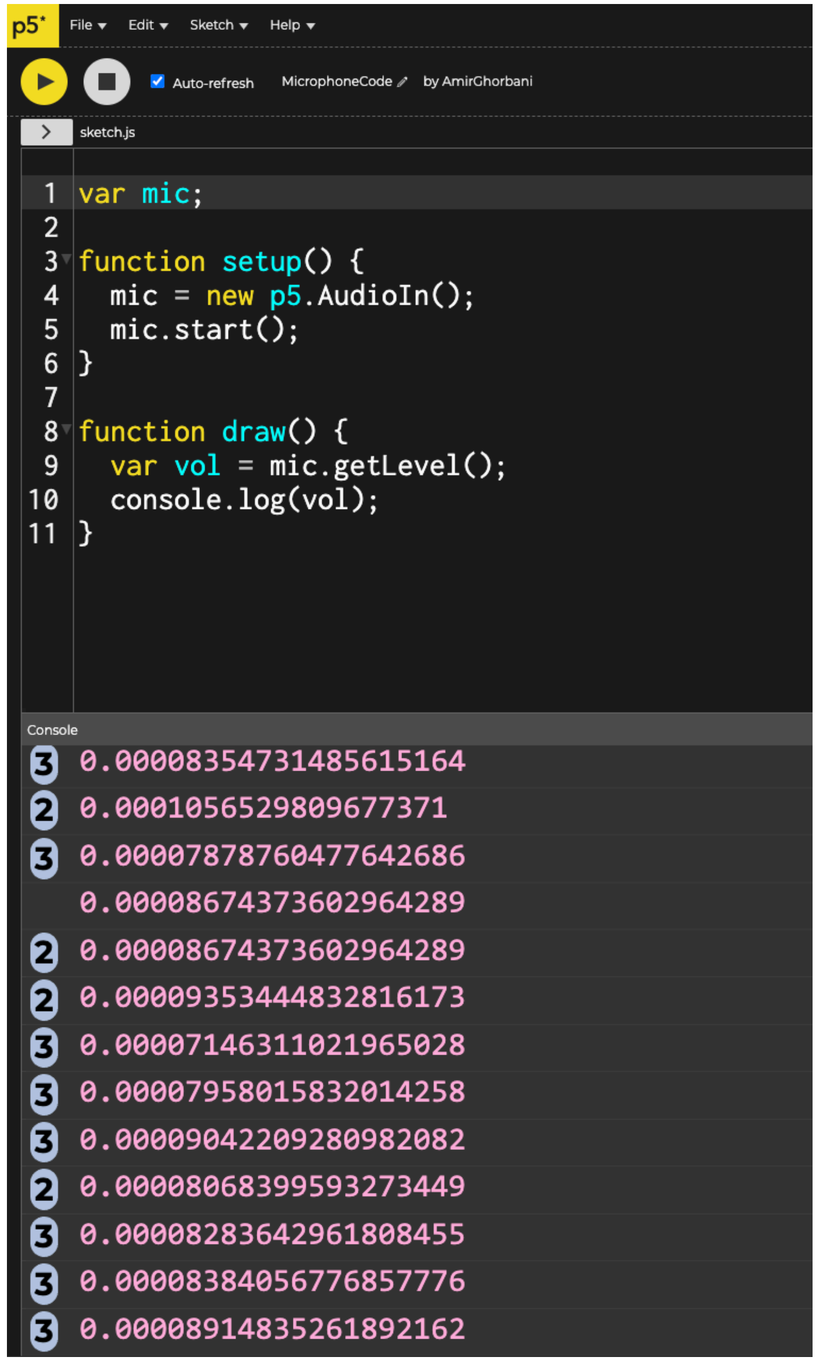
Change 400 to random(0, 400)
Random number between 0 and 400

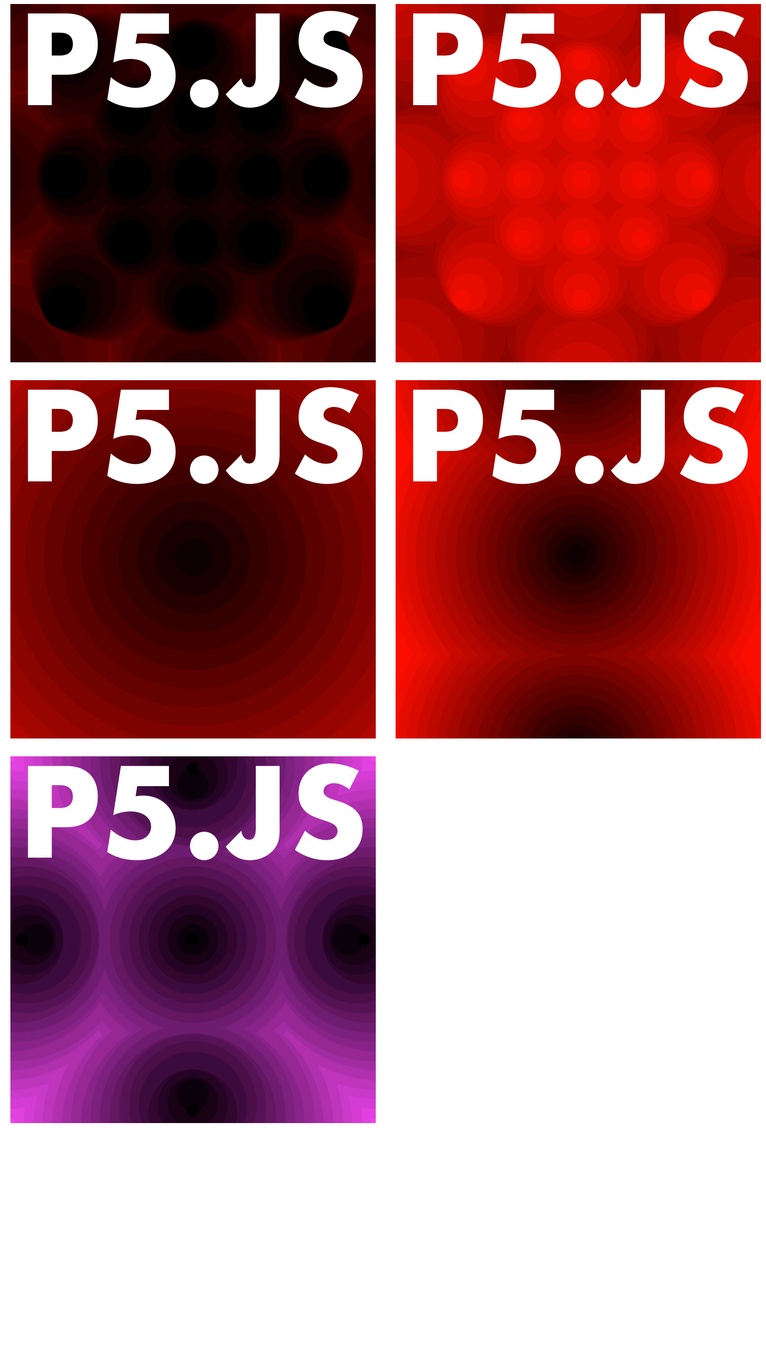
The generative graphics I made reminded me of speakers, so I asked myself why not create a digital display of a speaker.

Sound in p5.js

P5.JS Listen to the microphone



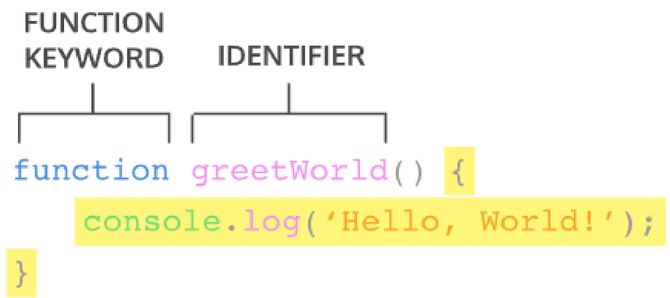




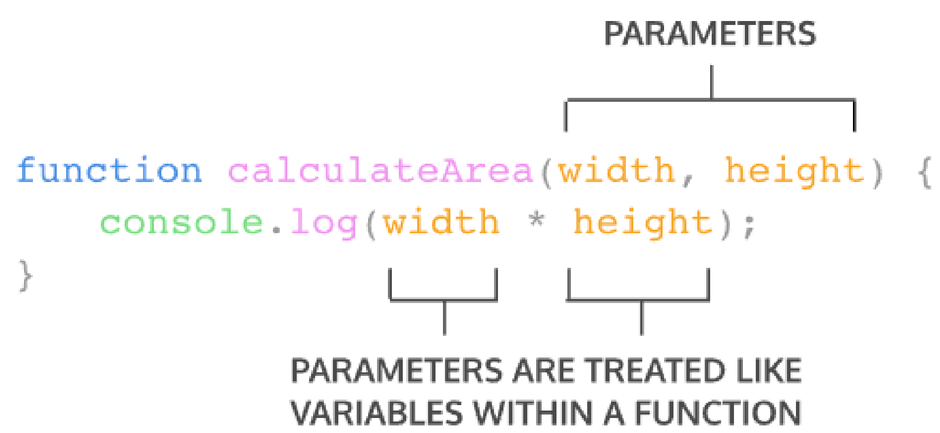
Function

What is Function:
A function is a reusable block of code that groups together a sequence of statements to perform a specific task.

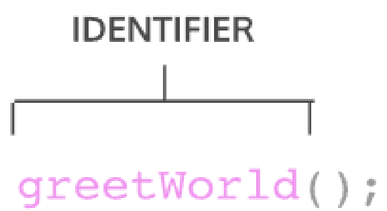
A function declaration:



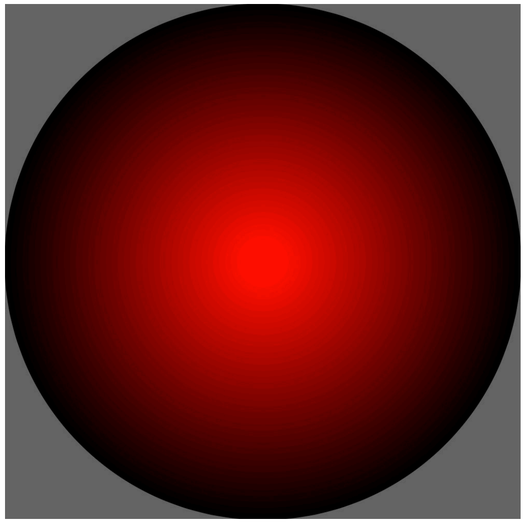
A **parameter** is a named variable inside a function's block which will be assigned the value of the argument passed in when the function is invoked:



To **call** a function in your code:

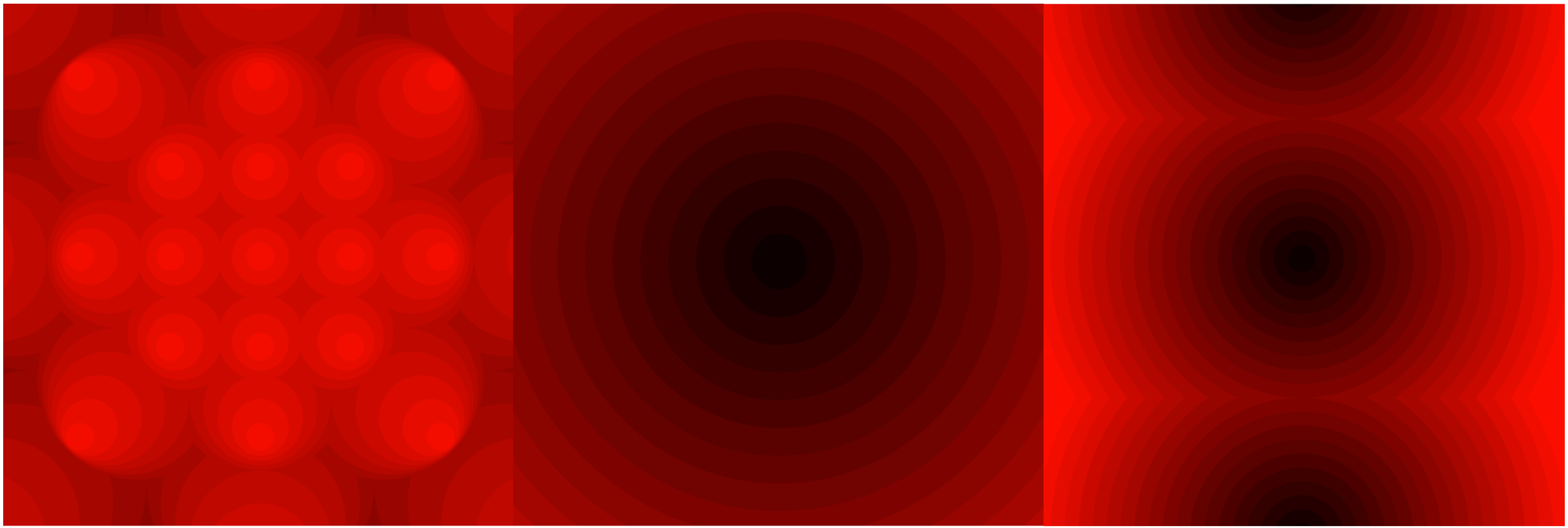


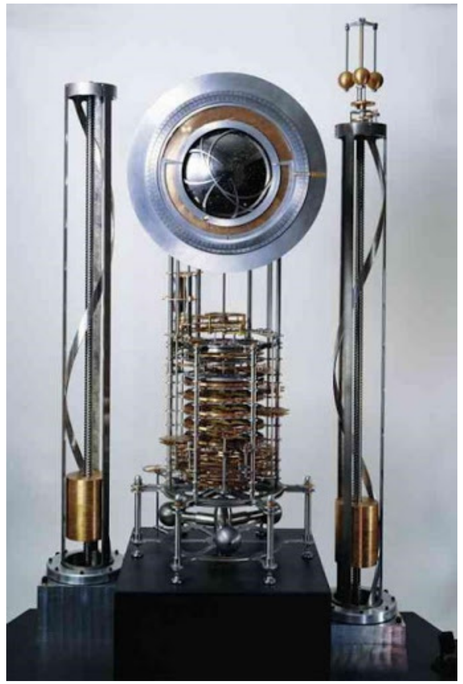
My Clock



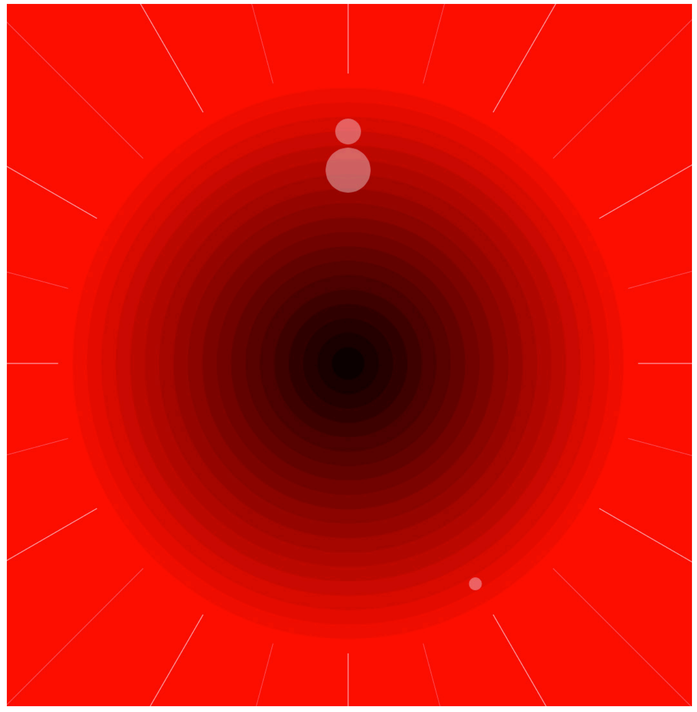
To come up with an initial design I started to generate simple visuals in p5.js only using circles.

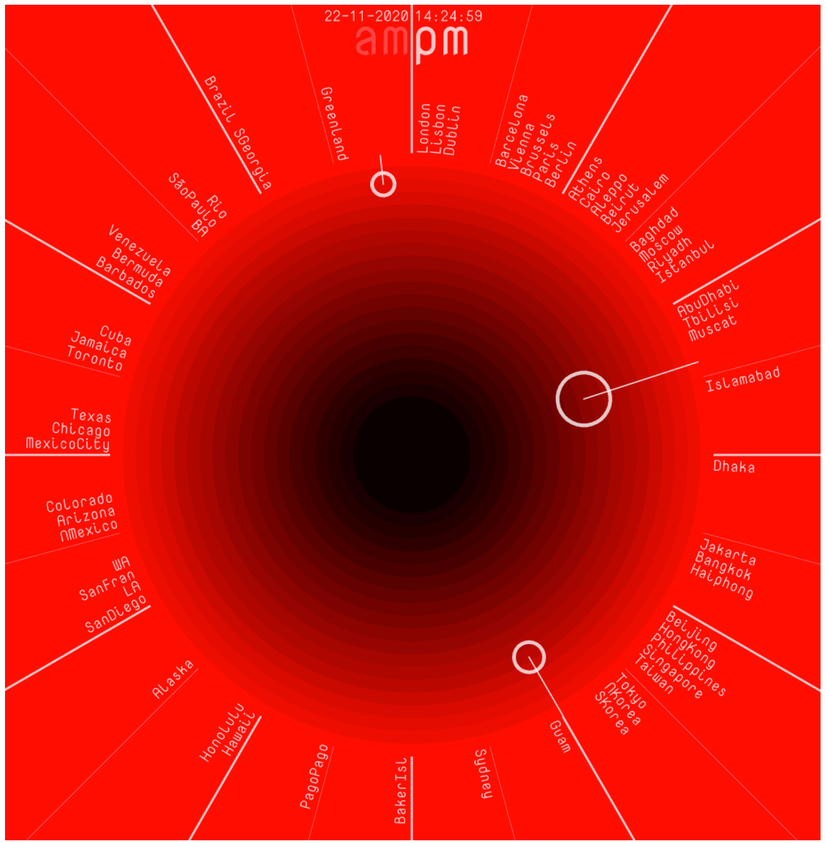
Different combinations and different numbers of circles.

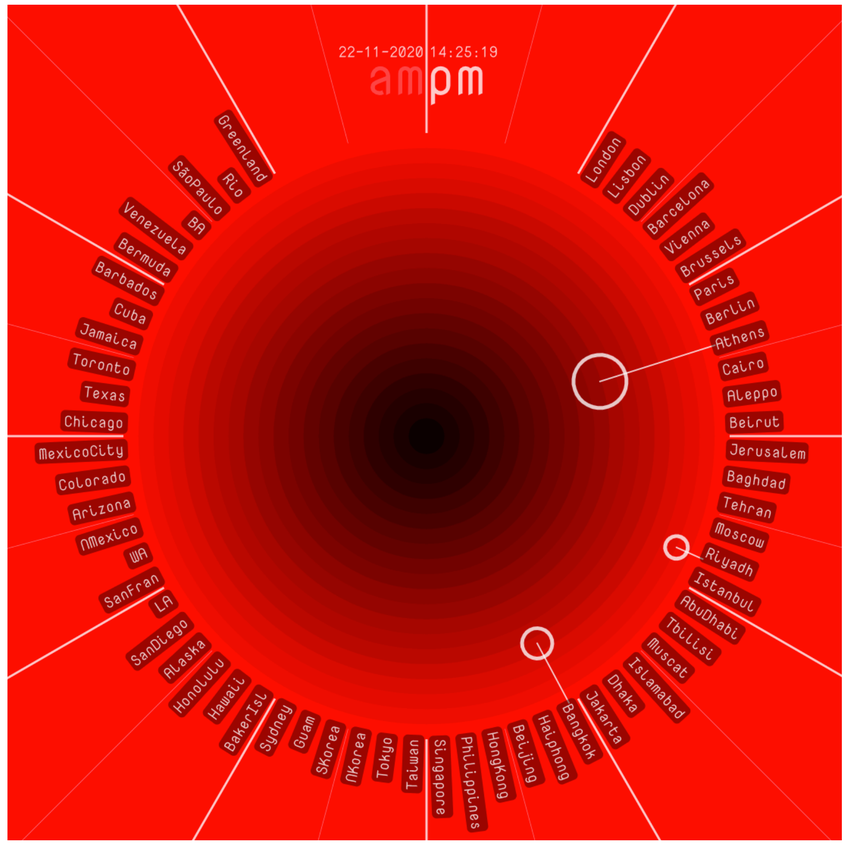




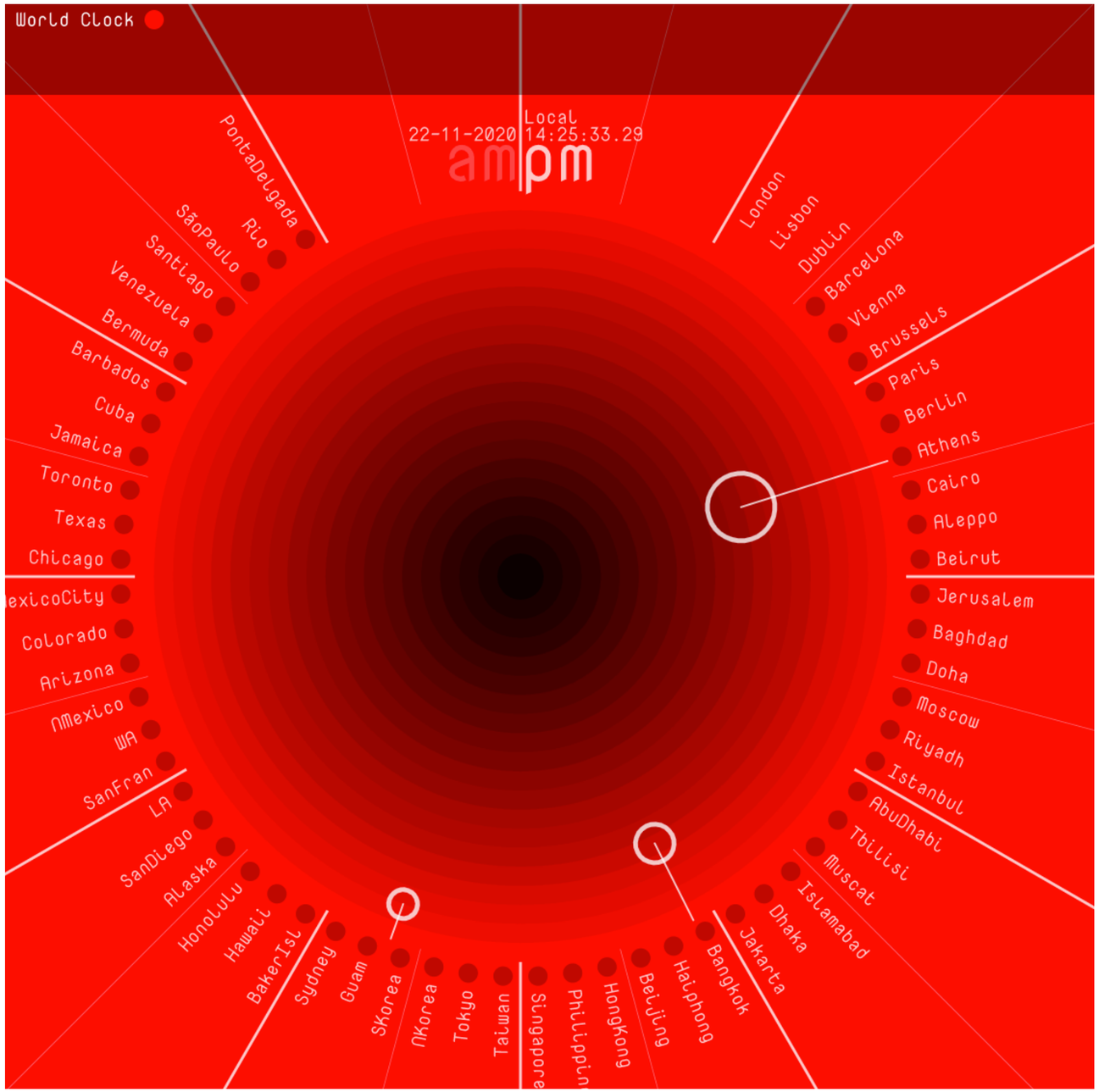
Using my initial generative visuals, Inspired by clock researches and the basic knowledge of sounds in p5.js, I came up with my first sound reactive design, inspired by speaker designs and 1000 year clock interface, in particular







I challenged myself to design a clock that shows different times for different time zones with each click.
It was a big coding challenge, from designing to program-ming. Creating many buttons and assigning them a new time.
I have decided to work on this clock as one of my personal projects.



I have developed my idea of a clock with different time zones. It works correctly only if you use it in London.

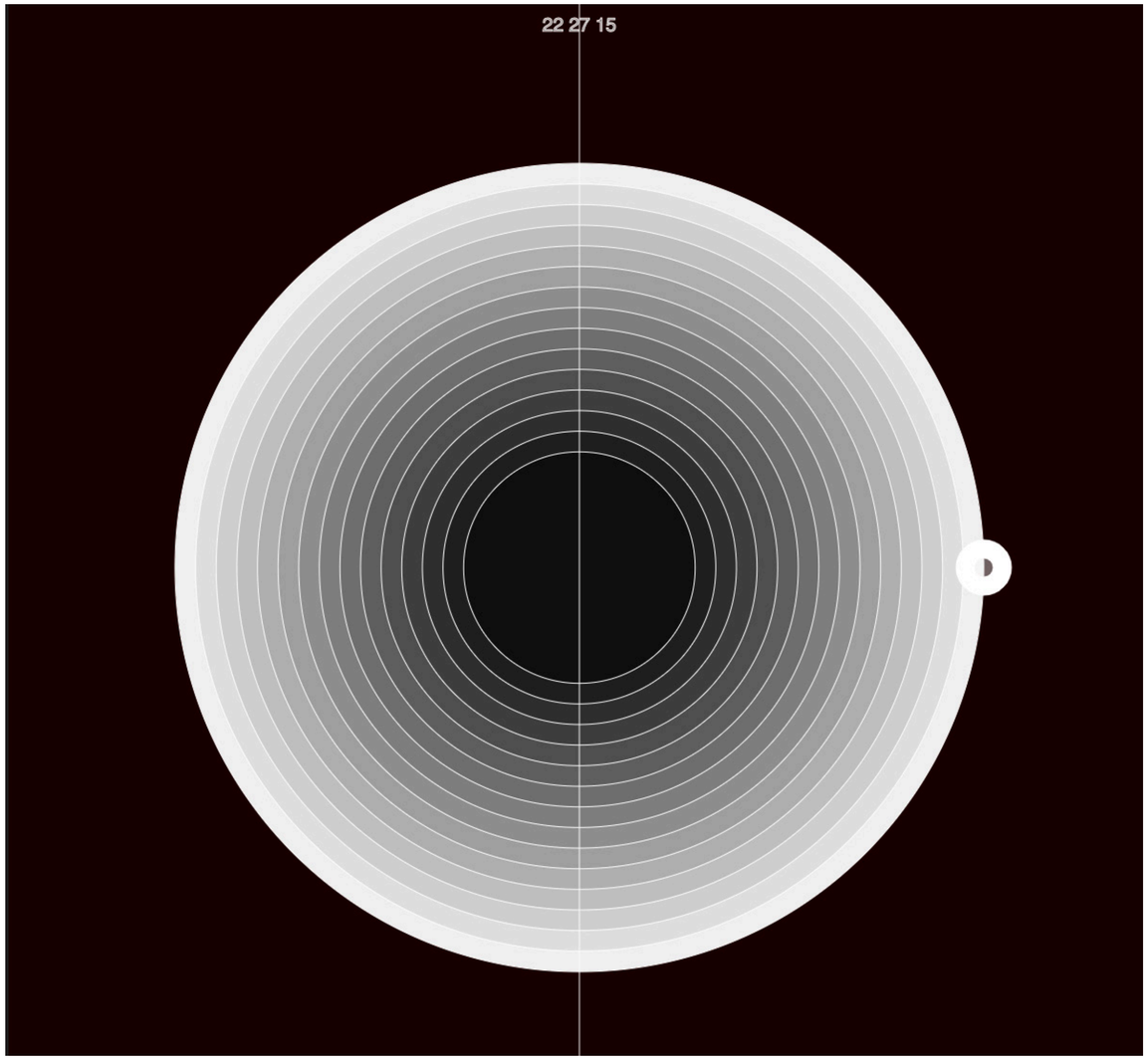
If you are living in london and click to see what time is in Berlin it shows the correct time:

London local time + 1

But

If you use the clock in Berlin and click on Berlin it show:

Berlin local time + 1



More elegant and minimal version of my time zone clock. Different colour modes.

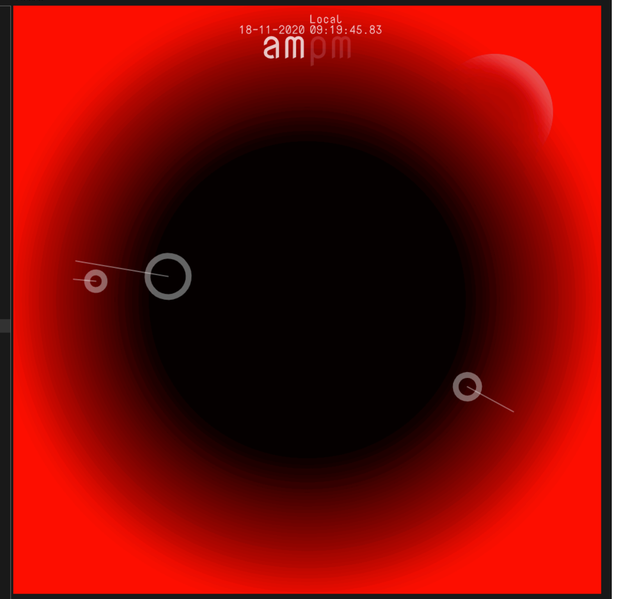
p5* File Edit Sketch Help

Auto-refresh Final01 - hms01 + Adding Text02 + sweepSec01 + click15 + Final03 + Colour picker02 + White02 + Final01 by AmirGhorbani

sketch.js Saved: 1 minute ago

```
let mic;
let myFont;
let lastSecond = -1
let baseMillis = 0

// Variables to create background
const numCircles = 50;
const numbers01 = _.range(numCircles);
const numLines = 24;
const numbers02 = _.range(numLines);
let red = 0
let green = 0
let blue = 0
let opac = 0
let colourChanger = true

// Load the font
function preload() {...}

// Day mode and night mode clock face
function myClockFaceColourPicker01() {...}
function myClockFaceColourPicker02() {...}

function fractionalSecond() {...}

function timeFunction() {...}

// am and pm
function myTopText(Opa01, Opa02) {...}

// Button + Click ----------------------------------------------------
// "Is the pointer inside the circle?"
function insideCircle(rotAng) {...}
// "Yes now thw pointer is inside the circle. Let change the circle."
function insideCircleChange(rotAng, red, green, blue, opa) {...}
// Click function
function mouseClicked() {...}
// -------------------------------------------------------------------

function setup() {...}

function draw() {...}
```

Preview



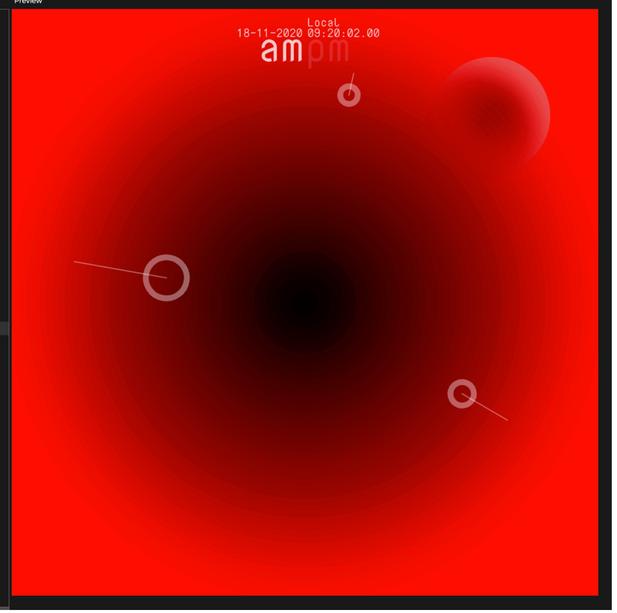
p5* File Edit Sketch Help

Auto-refresh Final01 - hms01 + Adding Text02 + sweepSec01 + click15 + Final03 + Colour picker02 + White02 + Final01 by AmirGhorbani

sketch.js Saved: 2 minutes ago

```
let mic;
let myFont;
let lastSecond = -1
let baseMillis = 0

// Variables to create background
const numCircles = 50;
const numbers01 = _.range(numCircles);
const numLines = 24;
const numbers02 = _.range(numLines);
let red = 0
let green = 0
let blue = 0
let opac = 0
let colourChanger = true

// Load the font
function preload() {...}

// Day mode and night mode clock face
function myClockFaceColourPicker01() {...}
function myClockFaceColourPicker02() {...}

function fractionalSecond() {...}

function timeFunction() {...}

// am and pm
function myTopText(Opa01, Opa02) {...}

// Button + Click ----------------------------------------------------
// "Is the pointer inside the circle?"
function insideCircle(rotAng) {...}
// "Yes now thw pointer is inside the circle. Let change the circle."
function insideCircleChange(rotAng, red, green, blue, opa) {...}
// Click function
function mouseClicked() {...}
// -------------------------------------------------------------------

function setup() {...}

function draw() {...}
```

Preview



Final Interface

Local
22-11-2020 14:23:43.95
am pm
Click

Local
22-11-2020 14:23:31.41
am pm
Click

code

vvv

The Clock’s sound reactive feature only works in p5.js editor.

https://editor.p5js.org/AmirGhorbani/sketches/Sh4Zlg_AI

Software: Clock

Is the code well laid out, structured and readable?
How does the sound reactive work?
How does this code facilitate the sweep second hand?
How is the gradient effect created?
Is the code well commented?

```
// Clock face variables
const numCircles = 40;
const numbers01 = _.range(numCircles);

let mic;

let myFont;

// MilliSecond variables
let lastSecond = -1
let baseMillis = 0

let red = 0
let green = 0
let blue = 0
let opac = 0

let colourChanger = true

// Mouse tracker variables
let x = 1
let y = 1

//////////////////////////////////////////////////////////////Functions//////////////////////////////////////////////////////////////
///
//////////////////////////////Functions//////////////////////////////Functions//////////////////////////////Functions////////
////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

// Load the font function ////////////////////////////////////////
function preload() {...}

// Day mode and night mode clock face ////////////////////////////
function myClockFace() {...}
// Night mode click face ////////////////////////////////////////
function myClickFace() {...}

function fractionalSecond() {...}

function timeFunction() {...}

// am and pm ////////////////////////////////////////////////////
function myTopText(Opa01, Opa02) {...}

// Button + Click ///////////////////////////////////////////////
// "Is the pointer inside the circle?"
function insideCircle(rotAng) {...}
// "Yes now the pointer is inside the circle (Click). Lets change the circle's property."
function insideCircleChange(rotAng, red, green, blue, opa) {...}
// Click function
function mouseClicked() {...}

//////////////Setup//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////Setup//////////////////////Setup//////////////////////Setup/////
//////////////Setup//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function setup() {...}

//////////////Draw//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////Draw//////////////////////Draw//////////////////////Draw////////
//////////////Draw//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

function draw() {...}
```

Variables

- Clock's face.
- Microphone.
- Font.
- MilliSecond.
- Red.
- Green.
- Blue.
- Alpha.
- Changing the colour.
- Mouse tracker.

Functions

- preload(): Font.
- myClockFace(): Day mode and night mode.
- myClickFace(): Night mode button.
- fractionalSecond().
- timeFunction().
- myTopText(Opa01, Opa02): For "am" and "pm" display.
- insideCircle(rotAng).
- insideCircleChange(rotAng, red, green, blue, opa).
- mouseClicked().

Setup

Draw

Variables

```
// Clock face variables
const numCircles = 40;
const numbers01 = _.range(numCircles);
```

Variables to create the clock's face

```
let mic;
```

```
let myFont;
```

Microphone and font (NovaMono-Regular.ttf)

```
// MilliSecond variables
let lastSecond = -1
let baseMillis = 0
```

```
let red = 0
let green = 0
let blue = 0
let opac = 0
```

```
let colourChanger = true
```

To Change the colour with every click

```
// Mouse tracker variables
let x = 1
let y = 1
```

How does the sound reactive work?

1 - To create a sound reactive element you need to define a variable. In my code the variable is mic.

```
var mic;
```

Inside the setup():
2 - Get audio from your computer's microphone and assign it to your variable.
3 - Turn the mic on with the start() method.

```
function setup() {
  mic = new p5.AudioIn();
  mic.start();
}
```

Inside the draw():
4 - When the mic is on, its volume can be measured with getLevel

```
function draw() {
  var vol = mic.getLevel();
}
```

5 - Use the vol

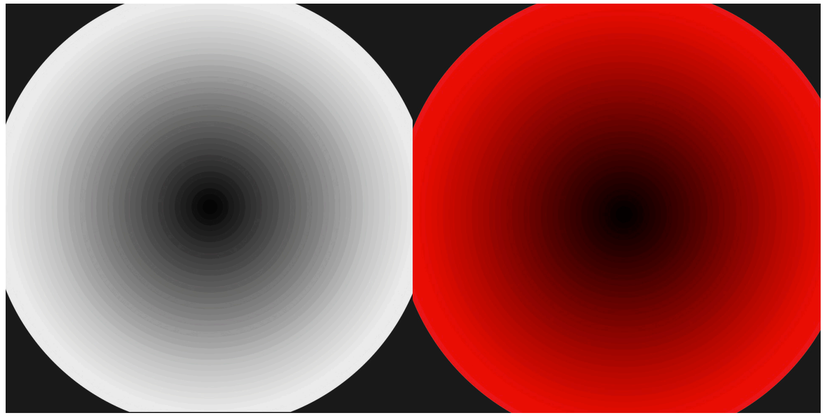
* This function requires you include the p5.sound library.

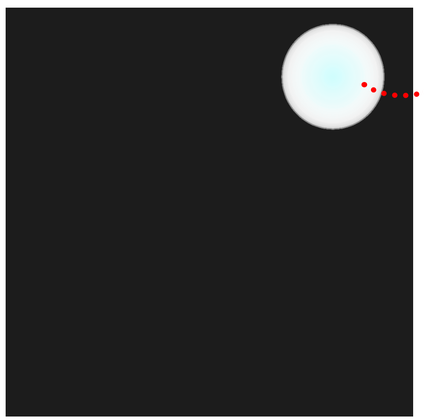
Function myFunction() {
  In detail;
}

```
// Load the font function ////////////////////////////////////////
function preload() {
  myFont = loadFont('NovaMono-Regular.ttf')
}
```

Reload the font

```
// Day mode and night mode clock face // When this function is called the
program will draw clock face which is a circle with gradient at the center of
the canvas ////////////////////////////
function myClockFace() {
  let vol = mic.getLevel();
  for (let n of numbers01) {
    let size = map(n, 0, numCircles, 1100, vol*5000);
    // Day mode // White mode ////////////////////////////////////////
    if (colourChanger) {
      red = map(n, 0, numCircles, random(250, 255), 0);
      green = map(n, 0, numCircles, random(250, 255), 0);
      blue = map(n, 0, numCircles, random(250, 255), 0);
      opac = map(n, 0, numCircles, 0, random(250, 255));
    } else { // Night mode // Red mode ////////////////////////////
      red = map(n, 0, numCircles, random(250, 255), 0);
      green = 0
      blue = 0
      opac = map(n, 0, numCircles, 0, random(250, 255));
    }
    fill(red, green, blue, opac);
    ellipse(0, 0, size, size);
  }
}
// Night mode click face // On night mode the button is illuminating with this
function // The function is not for the day mode ////////////////////////
/////////////
function myClickFace() {
  let vol = mic.getLevel();
  for (let n of numbers01) {
    let size = map(n, 0, numCircles, 250, vol * 400);
    // Night mode / Red mode ////////////////////////////////////////
    if (!colourChanger) {
      red = map(n, 0, numCircles, random(250, 255), 200);
      green = 255
      blue = 255
      opac = map(n, 0, numCircles, 0, random(50, 51));
      fill(red, green, blue, opac);
      ellipse(cos(45) * 450 - 20, -sin(45) * 450 + 20, size, size);
    }
  }
}
```





This function only works in night mode and it is illuminating the button.

```
const numCircles = 40;
const numbers01 = _.range(numCircles);
```

```
for (let n of numbers01) {

  let size = map(n, 0, numCircles, 1100, vol*5000);
  red = map(n, 0, numCircles, random(250, 255), 0);
  green = map(n, 0, numCircles, random(250, 255), 0);
  blue = map(n, 0, numCircles, random(250, 255), 0);
  opac = map(n, 0, numCircles, 0, random(250, 255));

  fill(red, green, blue, opac);
  ellipse(0, 0, size, size);
}
```

How is the gradient effect created?

1 - Low dash function to create 40 concentric circles.
2 - The program draw circles in this order:

1st circle:
fill(random(250, 255), random(250, 255), random(250, 255), 0);
ellipse(0, 0, 1100, 1100);

• • •

Last circle on top of all circles:
fill(0, 0, 0, random(250, 255));
ellipse(0, 0, vol*5000, vol*5000);

Function myFunction() {
  In detail;
}

```
function fractionalSecond() {
  if (second() != lastSecond) {
    baseMillis = millis();
    lastSecond = second();
  }
  var millIsWithinSec = millis() - baseMillis
  return second() + (millIsWithinSec / 1000)
}
```

Sweep Second function called inside the timeFunction

```
function timeFunction() {
  noFill();
  let endS = map(second(), 0, 60, 0, 360)
  let endM = map(minute(), 0, 60, 0, 360)
  let x1 = map(second(), 0, 60, 0, 6)
  let endH = map(hour() % 12, 0, 12, 0, 360)
  let x2 = map(minute(), 0, 60, 0, 30)
  let sweepSec = map(fractionalSecond(), 0, 60, 0, 360)
  push()
  rotate(-90);
  strokeWeight(10)
  if (colourChanger) {
    stroke(0, 100)
  } else {
    stroke(255, 100)
  }
  // Second ////////////////////////////////////////////////////////////
  push();
  rotate(sweepSec);
  circle(361, 0, 30)
  strokeWeight(2)
  line(361, 0, 899, 0)
  pop();
  // Minute ////////////////////////////////////////////////////////////
  push();
  rotate(endM + x1);
  circle(310, 0, 40)
  strokeWeight(2)
  line(310, 0, 899, 0)
  pop();
  // Hours ////////////////////////////////////////////////////////////
  push();
  rotate(endH + x2);
  circle(240, 0, 70)
  strokeWeight(2)
  line(240, 0, 899, 0)
  pop()
  pop()
}
```

How does this code facilitate the sweep second hand?

```
let lastSecond = -1
let baseMillis = 0

function fractionalSecond() {
    return millis() % 1000;
}

function fractionalSecond() {
  if (second() != lastSecond) {
    baseMillis = millis();
    lastSecond = second();
  }
  var millIsWithinSec = millis() - baseMillis
  return second() + (millIsWithinSec / 1000)
}
```

We want to know what was the last value for the second.
We assign -1, because we never have -1 value for the seconds and later we can change it.

Constantly count from 0 to 999

If the value of the second is not the value we saw the last time:

1. We remember the millisecond.
2. We assign the value of second to a variable

Constantly count from 0 to 999

Function myFunction() {
  In detail;
}

```
function myTopText(Opa01, Opa02) {
  if (colourChanger) {
    fill(0, Opa01)
    textAlign(LEFT)
    text("pm", -1, -410)
    fill(0, Opa02)
    textAlign(RIGHT)
    text("am", -2, -410)
  } else {
    fill(255, Opa01)
    textAlign(LEFT)
    text("pm", -1, -410)
    fill(255, Opa02)
    textAlign(RIGHT)
    text("am", -2, -410)
  }
}
```

This function controls the colour and opacity of texts "am" and "pm".
Black on day mode and white on night mode.
It will be called inside the draw function.

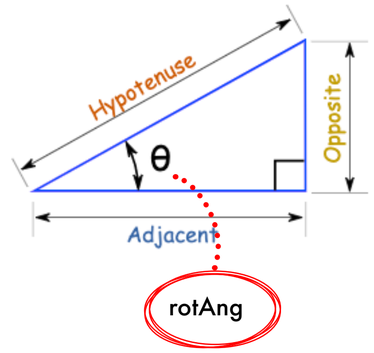
```
// Button + Click ////////////////////////////////////////////////////
// "Is the pointer inside the circle?"
function insideCircle(rotAng) {
  let d = dist(width / 2 + cos(rotAng) * 450, height / 2 - sin(rotAng) * 450,
mouseX, mouseY)
  if (d < 100) {
    return true
  } else {
    return false
  }
}
// "Yes now the pointer is inside the circle (Click). Lets change the circle's
property."
function insideCircleChange(rotAng, red, green, blue, opa) {
  let x = 200
  if (insideCircle(rotAng)) {
    x = 220
    fill(red, green, blue, 100)
  } else {
    fill(red, green, blue, opa)
  }
  push()
  rotate(-rotAng)
  circle(450, 0, x)
  pop()
}
// Click function
function mouseClicked() {
  if (insideCircle(45)) {
    colourChanger = !colourChanger
  }
}
```

This function controls if the pointer is inside the circle (button) - True of False?
rotAng value determines the location of the circle.

the pointer is inside the circle (Click).
Lets change the circle's property.

Click function

$$\sin\theta = \frac{\text{Opposite}}{\text{Hypotenuse}} \qquad \cos\theta = \frac{\text{Adjacent}}{\text{Hypotenuse}} \qquad \tan\theta = \frac{\text{Opposite}}{\text{Adjacent}}$$



Function setup() {
  In detail;
}

```
function setup() {
  createCanvas(1000, 1000);
  noStroke();
  angleMode(DEGREES)
  textFont(myFont)

  // Microphone code
  mic = new p5.AudioIn();
  mic.start();
}
```

General attributes of the canvas and drawing.

Microphone p5.js built in functions.

Function draw() {
In detail;
}

```
function draw() {
  // Background is changing with every click //////////////////////
  if (colourChanger) {
    background(255)
    fill(255,0,0)
  } else {
    background(255, 0, 0)
    fill(255)
  }

  // Easeing the mouse tracker ////////////////////////////////////
  x= x+(mouseX -x)/20
  y =y+(mouseY -y)/20
  ellipse(x, y, 40, 40);

  translate(width / 2, height / 2)

  insideCircleChange(45, 200, 200, 200, 100)
  myClickFace()
  myClockFace()
  timeFunction()

  // Bonus
  let vol = mic.getLevel();
  if (vol * 1000 > 100) {
    fill(255, 255)
    circle(0, 0, 2)
  }
```

The value of colourChange starts with true, so the background colour starts with red and then will toggle between white and red with every click.

Easing the mouse tracker

Translate all the drawings

Calling 4 functions:
1 - This function controls everything related to interaction with the button (the circle located on top right side of the canvas).
2 - The sound reactive button face working only on Night mode (Red mode)
3 - The sound reactive clock face (Day mode and night mode)
4 - This function controls everything related to time.

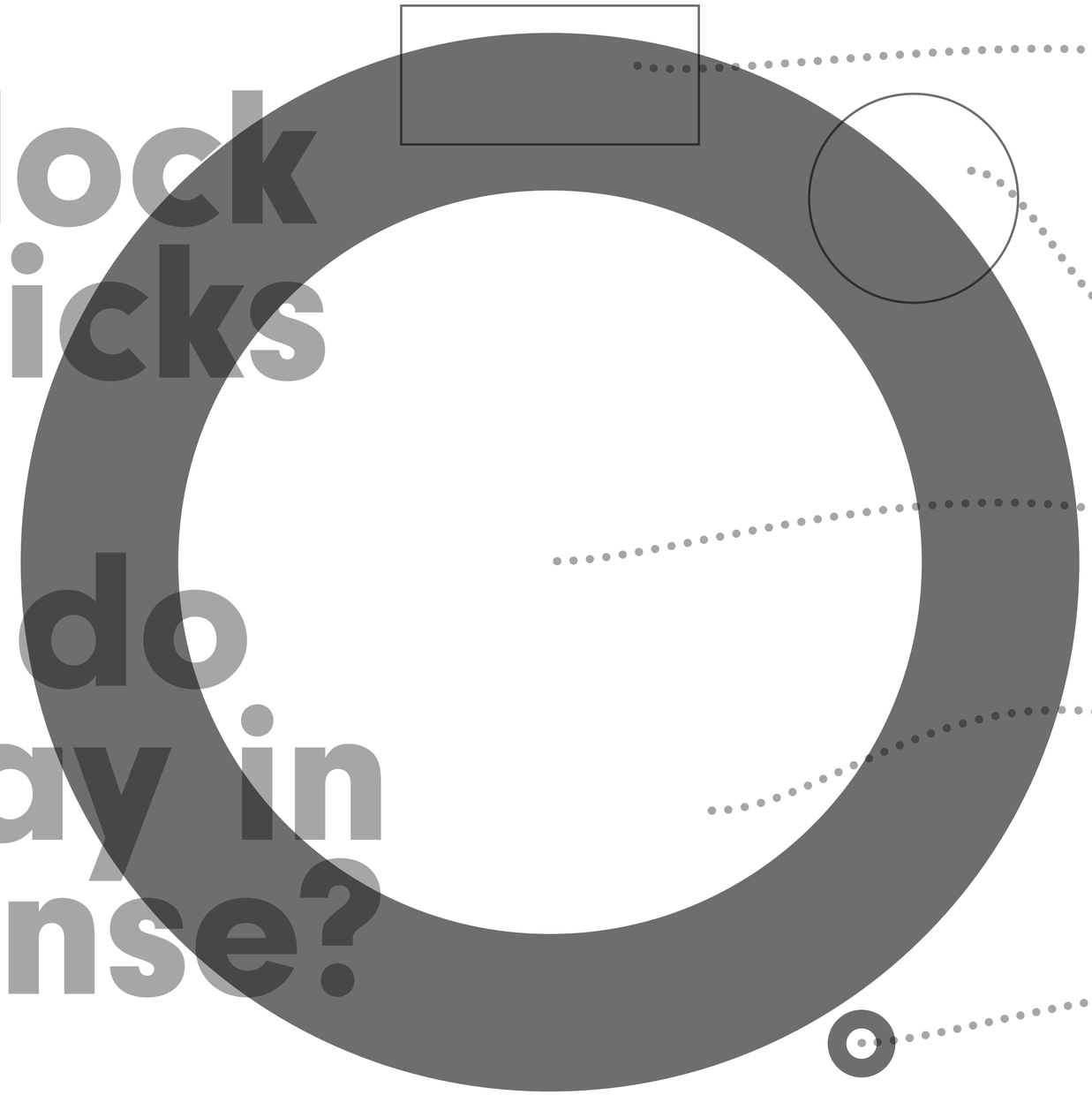
Bonus Part: It shows a tiny circle in the middle of canvas when the volume is going higher that a certain amount

```
  // Text AM and PM ///////////////////////////////////////////////
  // Date & time on top of the canvas
  textSize(70)
  if (hour() > 12 && hour() < 24) {
    myTopText(200, 40) // am and pm function ////////////////////////
  } else {
    myTopText(40, 200)
  }
  textSize(20)

  if (colourChanger) {
    fill(0, 200)
  } else {
    fill(255, 200)
  }
  textAlign(RIGHT)
  text(nf(day(), 2, 0) + '-' + nf(month(), 2, 0) + '-' + year(), -4, -452)
  textAlign(LEFT)
  text('Local', 4, -470)
  text(nf(hour(), 2, 0) + ':' + nf(minute(), 2, 0) + ':' + nf(fractionalSecond(), 2, 2), 4, -452)
  push()
  textAlign(CENTER)
  translate(360, -360)
  rotate(45)
  text('Click', 0, 0)
}
```

This part contains everything related to the text or digital clock on top of the canvas:
Also it calls the myTopText function

User Manual

The clock says ticks tocks. What do you say in response?



Local time and date

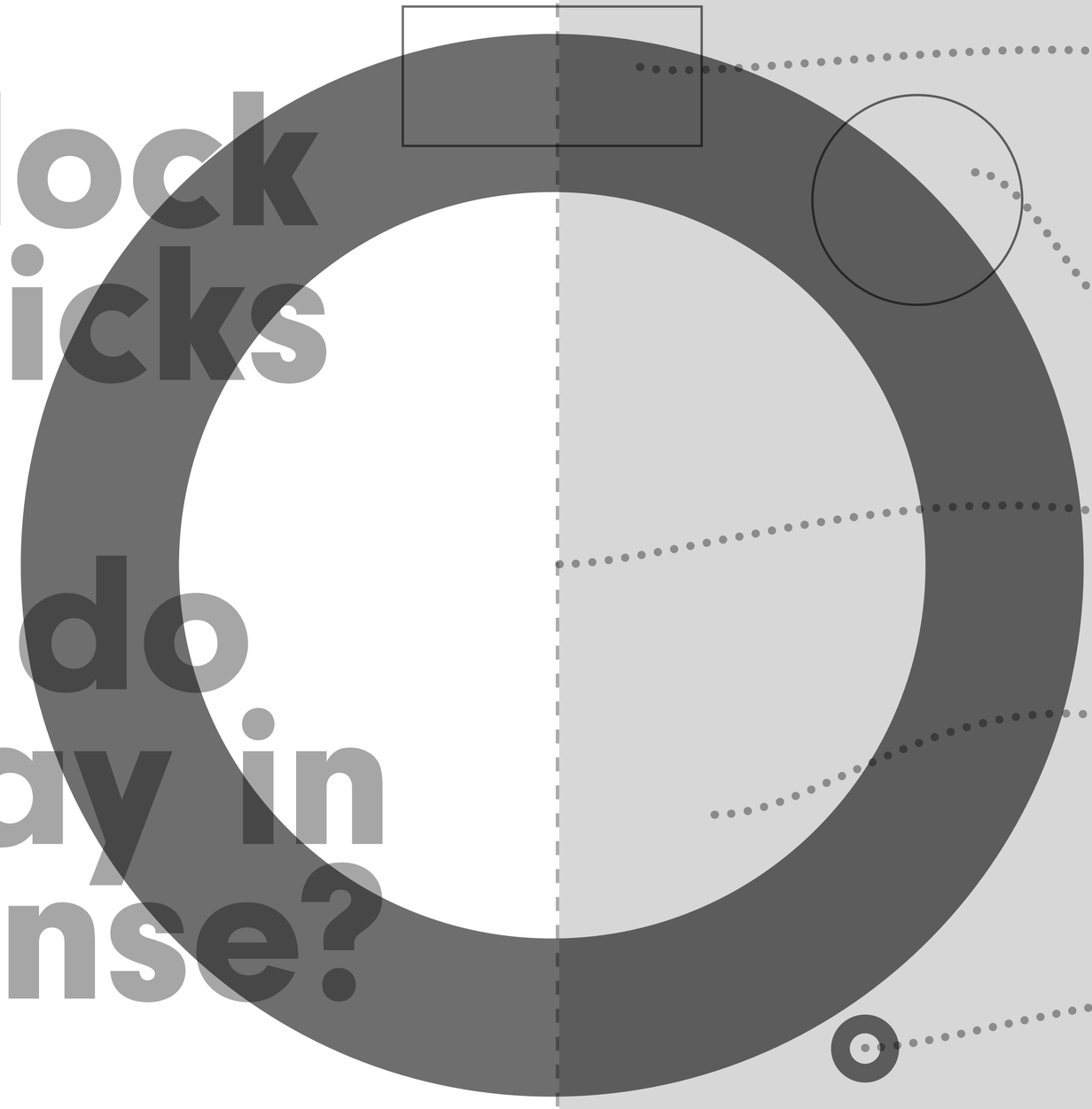
Click to change the mode
* Day and Night

Speak loader to see a surprise
* It is only a tiny white circle right at the centre

Speak to your clock in both modes.
* When no one is around

Mouse tracker ball

The clock says ticks tocks. What do you say in response?



Local time and date

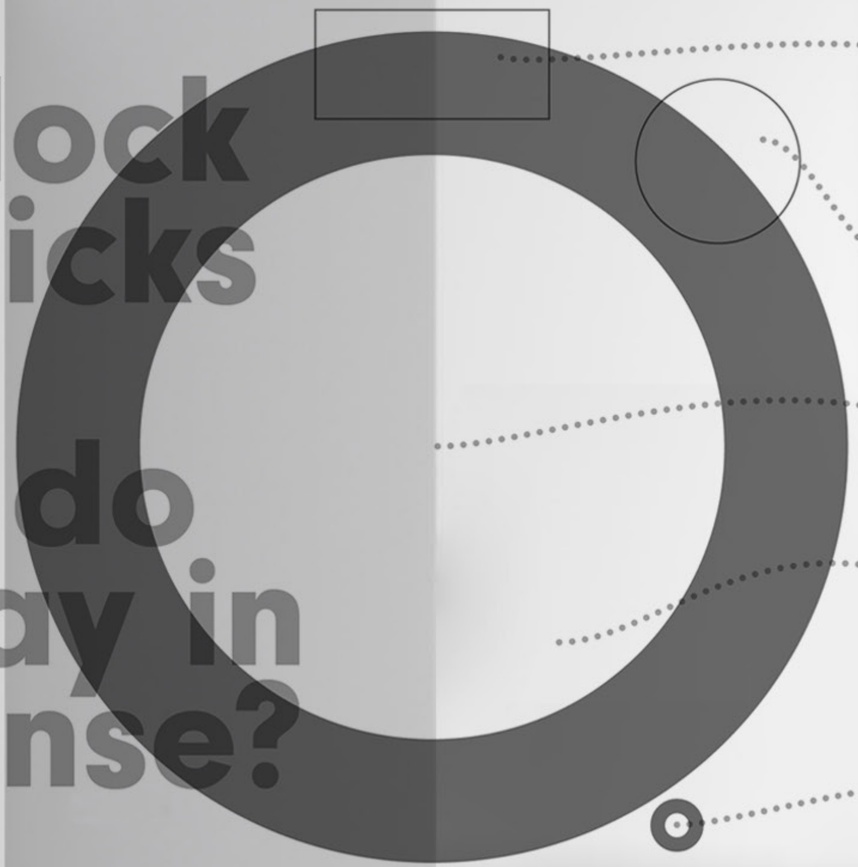
Click to change the mode
* Day and Night

Speak loader to see a surprise
* It is only a tiny white circle right at the centre

Speak to your clock in both modes.
* When no one is around

Mouse tracker ball

The clock says ticks tocks. What do you say in response?



Local time and date

Click to change the mode
* Day and Night

Speak loader to see a surprise
* It is only a tiny white circle right at the centre

Speak to your clock in both modes.
* When no one is around

Mouse tracker ball


30-11-2020
Local
11:58:57.56
am
pm
CLLrA


30-11-2020
Local
11:59:11.16
am
pm
CLLrA

Thank you

My Generative Clock

Amir Ghorbani

November 2020

Y1 USE18103 Coding, Figures, Visuals 20-21

Tutor: Nick Rothwell.