CART 253

Creative Computation 1

Email: l.wilkins@concordia.ca
Office Hours: Tuesday 12-1
Course Github: https://github.com/LeeRobot/CART253-F-22

What we'll be doing today

- Inspiration
- Time to work on project 2
- Learn about Arrays

Inspiration

- Jer Thorp: Just landed (2009) Avengers, Assembled (2012)

Upload your code to Github if you want to share!

- Add your link here and I'll share with the both classes!
- Github is a great way to store and share code
- There is version control so you can go back and see what you changed
- I recommend using the desktop client, here's a tutorial
- You can also just drag and drop your code into a repository

Second Project: Screens in Spaces, due week 8

- You may work in pairs (if you want)

- Create a **site-specific installation** using your computer screen and p5.js. Think about how people in these spaces interact and expect screens to behave in different contexts. Your piece does not have to be interactive, but it does have to evolve or change over time in some way. This can exist anywhere, but you should be able to show us the work either through video documentation and demo in class or, take us to the piece physically for critique. You can do this project in pairs or individually, but no bigger groups. You can use a makeymakey to simualte keybaord input.

- The piece must be interactive and/or change over time.

- Use at least 2 conditionals, 1 loop, and a function or array. We'l talk about those next week.

- Full details on Git and Moodle.

Second Project: Screens in Space, due week 8

- Some ideas:
  - Hallways
  - Classrooms
  - Lounge spaces
  - Outside
  - Lobby
  - Metro station
  - Restaurant/cafe
  - It can be somewhere far away or somewhere we can't visit (please record your installation / people interacting with it)
  - **How can your piece highlight, or contrast, or enhance the space its in? Can it blend in, can it make the viewer stop? How do people react to it? Can it take advantage of the architecture?**
  - **Think of projections, screens, tablets, laptops, screens etc.**

Second Project: Screens in Space, due week 8

- https://momentfactory.com/work/all/all/montreal-signe-ode-a-la-vie Projection mapping
- Putting p5.js on your phone https://www.youtube.com/watch?v=OyZNj7oMgek using screen cast or using your computer to host the code https://creative-coding.decontextualize.com/mobile/

```
let circleYOne = 10;
let circleYTwo = 20;
```

✗

```
let circleY = [10, 20];
```

✓

Declaired the same way as other variables

Variable Name

Square brackets hold the values of an array

items are comma separated

What if we want 2?

```
let circleY = [10, 20, 23, 3, 30, 26, 60, 80];
```

Or more!

Arrays let us store multiple variables in an indexed list. A good way know you need an array is if you are repeating variables, or have many of the same variables.

Arrays

- The number inside square brackets is the index, it indicates the position within the array.
- Arrays always start at 0
- Arrays can hold any type of variable (number, letter, word, boolean)
- You can usually identify an array because of square brackets myArray[0]

Arrays

Arrays

Declaring an array:

```
let myArray = [10, 20, 15];
```

Accessing an array:

myArray[0] is 10

myArray[1] is 20

myArray[2] is 15

Declaring an array:

```
let myArray = [10, 20, 15];
```

```
 for(let i = 0;i < 3; i++){
   print(myArray[I]);
}
```

Arrays

Cycle through an array

```
let myArray = [10, 20, 15];


  for(let i = 0;i < myArray.length; i++){
    print(myArray[I]);
}
```

Arrays

Array Push and Pop

```
  myArray.push(10);
```

Push adds a value to the END of the array. This makes myArray.length increase!

```
  myArray.pop();
```

Pop removes the last item in an array. This makes myArray.length decrease!

Cycle through an array

```
let myArray = [“dogs”, “cats”, “pigs”];


  for(let i = 0;i < myArray.length; i++){
    text(myArray[I], random(width), random(height));
}
```

Arrays

Code Together!

A sketch that has animated circles moving from top to bottom

Each circle is labeled with an animal name

When the user clicks, a circle is added. When the circle is added, it is spaced evenly along the X axis, not randomly.

With a user presses a key, a circle disappears, and the circles are re-arranged to distribute evenly along the X axis.

```
let circleY = 0;

function setup() {
  createCanvas(300, 300);
}

function draw() {
  background(50);

  circle(150, circleY, 25);

  circleY++;

  if (circleY > height) {
    circleY = 0;
  }
}
```

Back to our animation

```
let circleY = [];
function setup() {
  createCanvas(300, 300);
  for (let i = 0; i < 1; i++) {
    circleY[i] = random(height);
  }
}
function draw() {
  background(50);
  for (let i = 0; i < circleY.length; i++) {
    let circleX = width * i / circleY.length;
    circle(circleX, circleY[i], 25);
    circleY[i]++;
    if (circleY[i] > height) {
      circleY[i] = 0;
    }
  }
}
```

Arrays

Get Array from a CSV file

Video on visualizing CSV files

Arrays

Style & formatting

Full style guide here.

- Get rid of extra white space
- use TAB to indent functions, everything in side the function should be aligned.
- Curly brackets should be at the first line of the function, and alone at the end of the function.

```
function setup()    {
createCanvas(500,500);}
```

✗

```
function setup() {
  createCanvas(500,500);
}
```

✓