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

- Timers
- Consultations

Software Rube Goldberg Final week

- A rube goldberg machine uses one element to trigger another. Usually they are physical, but we will make a software version! Heres an example of a physical rube goldberg machine.

Software Rube Goldberg final week

- The goal of this project is to pass along a variable between programs to create a chain. Every person will get a number from someone else, use this number to create a piece of generative art, then pass along a new number to the next person. You must use MQTT to pass a variable.

Software Rube Goldberg final week

- Your piece can be ANYTHING: Audio, Visual, interactive, it can use a camera, it can use external data, particle systems, drawing, physical interactions, etc. Your work should be complex and show the skills we learned throughout the semester and things you've explored on your own.

Software Rube Goldberg final week

- Your project must:
  - Use **MQTT** protocol to send/receive messages
  - **Receive** a variable from another person
  - **Use** the variable as a factor in creating a visual, auditory, and/or interactive work.
  - **Change** the variable in some way
  - **Send** the variable successfully to the next person
  - **Successfully** complete this task during critique

Software Rube Goldberg final week

- Grading criteria:
  - Successfully receiving / changing / using / sending your variables
  - Using 1 external library for p5js in a meaningful capacity
  - Creativity and application
  - Complexity of code

Software Rube Goldberg due week 12

- Sign up: https://docs.google.com/spreadsheets/d/1CKbdBV9WpNAnygMrTLB8C4OHMg2SJD9eVuOk-nyH7v8/edit#gid=0

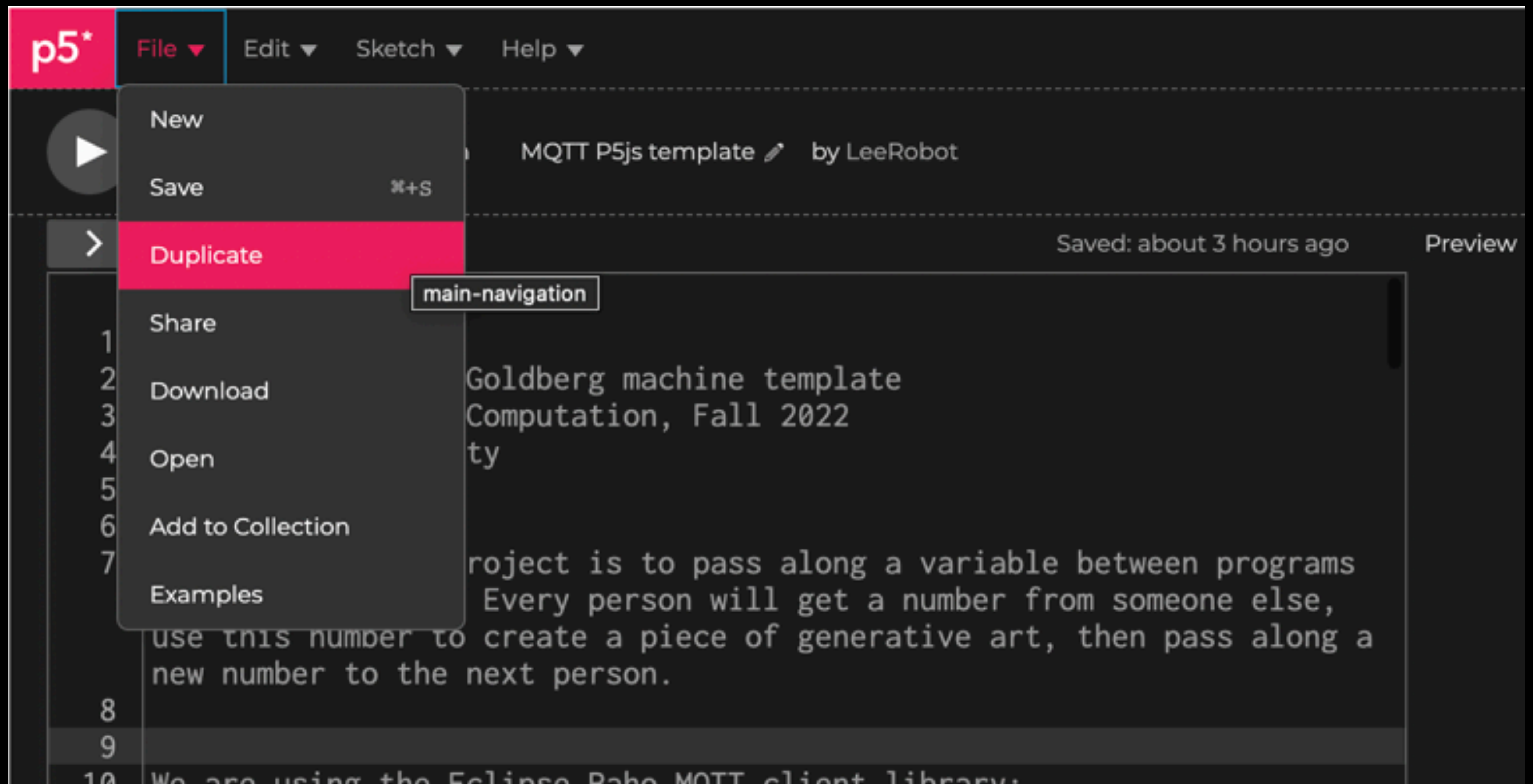
  Lee should be first, Enric should be last.

Template for project

- Find the template HERE https://editor.p5js.org/LeeRobot/sketches/RXA5yGasr

- Duplicate the template and save it!

- You can use the browser for this assignment, bu tyou can also use Atom if you want.



```
Goldberg machine template
Computation, Fall 2022
ty

roject is to pass along a variable between programs
 Every person will get a number from someone else,
use this number to create a piece of generative art, then pass along a
new number to the next person.

We are using the Eclipse Paho MQTT client library:
```

You can use a timer to keep track of intervals of time, or time elapsed since something, or count down to another event.

Timers

setinterval() can ue used to make a function occur at a regular period of time. More info here https://www.youtube.com/watch?v=EGQW0aUhDQM

setInterval

```
function setup(){
  createCanvas(500, 500);
  background(50);
 setInterval(randomCircle, 1000);
}

function randomCircle(){

  fill(random(255));

  ellipse(random(width), random(height), 100, 100);

}
```

This is the name of the function. You don't have to call it in setup, but if you want the interval to run at the beginning of your sketch its a good idea. You can set an interval at any point.
This is the function that is called at the interval
This is the duration between each time it is called in milliseconds

Set Interval

```
let myInterval;
function setup(){
  createCanvas(500, 500);
  background(50);
 myInteval = setInterval(randomCircle, 1000);
}

function randomCircle(){

  fill(random(255));

  ellipse(random(width), random(height), 100, 100);

}
```

Store the interval in a variable so we can reference it later.

This is the name of the function. You don't have to call it in setup, but if you want the interval to run at the beginning of your sketch its a good idea. You can set an interval at any point.
This is the function that is called at the interval
This is the duration between each time it is called in milliseconds

Set Interval

```
let myInterval;
function setup(){

  createCanvas(500, 500);
  background(50);
}
 function mousePressed() {
    myInterval = setInterval(randomCircle, 100);

 }
 function keyPressed(){
    clearInterval(myInterval);
 }
function randomCircle(){
  fill(random(255));
  ellipse(random(width), random(height), 100, 100);
}
```

Stop Interval

millis() returns the current number of milliseconds between the start of the sketch and now. You can use millis() to pinpoint a time in the future by saying:

```
endTime = millis() + 5000;
```

This takes the current time and ads 5 seconds (5000 milliseconds)

Millis()

```
let timer = false;
let duration = 5;

let endTime;

let myInterva;

function setup() {

  createCanvas(800, 400);

  background(50);

}

function mousePressed(){
  endTime = millis() + (duration*1000);

  timer = true;

  print("start timer");

  myInterval = setInterval(checkTimer);

}

function checkTimer(){

if(millis() > endTime){

    print("ended");

    timer = false;

    clearInterval(myInterval);

    background(50);

  } else {

    background(255, 0, 0);
```

Set a timer that is 5 seconds in the future of the click time
Set an interval to start checking if the timer is up
In the interval function, check if the current time is greater than the end time and if so, change the background, and set the flag to false.

Set a timer