CART 263

Creative Computation 2

Email: l.wilkins@concordia.ca
Office Hours: Tuesday 12-1
Course Github: https://github.com/LeeCyborg/CART263-W-23
TA: tricia.enns@gmail.com

What we'll be doing today

- A few camera examples
- Group forming and brain storming
- Camera tutorial

Camera

```
let cam;

function setup() {
  createCanvas(500, 500);
  cam = createCapture(VIDEO);
  cam.hide();
}

function draw() {
  /* Basic camera  */
  image(cam, 0, 0, width, width * cam.height / cam.width);
  filter(INVERT);
```

Make A filter

```
// A function to filter video by looking through all of the pixels
// and manipulating their value.
function videoFilter(){
  background(255);
  //load pixels of the camera feed
  cam.loadPixels();

  let counter = 0;
  // Every pixel divided into 4 elements in the array,
  // 0 is red, 1 is green, 2 is blue, 3 is green, 4 is alpha (opacity)
  for (let i = 0; i < cam.pixels.length; i += 4) {
    cam.pixels[i] = 0; // Get rid of all the red
    cam.pixels[i]+=counter; // Add red in as we  move down the image
    counter+=0.001; // increase counter by a very small amount
  }
  //update pixels and display them
  cam.updatePixels();
  // draw the camera
  image(cam, 0, 0, width, height);
}
```

Use input without camera

```
// A function to tell you what color is most present in the camera, without
// displaying the camera
  // 0 for red, 1 for green, 2 for blue
  // set thresh for 100 if no other input
function findColor(c, thresh=100){
  background(0);
  // set a counter to keep track of how much of the color there is
  let counter = 0;
  cam.loadPixels();
  // look at every 4th pixel and check it if it is greater than the threshhold.
  for (let i = 0; i < cam.pixels.length; i += 4) {
    if(cam.pixels[i+c]>thresh){
      counter++;
    }
  }
  // map it to the total pixels and draw a circle
  s = map(counter, 0, (width*height)*4, 0, width);
  fill(255);
  circle(width/2,height/2,s);
}
```

Playful Interaction
Teams of 2, 40%

```
### Overview
Create a playful interaction that expands beyond a single
computer screen and involves more than 1 person in some capacity.
This can mean:
- 2 screens connected remotely (MQTT, HTTP Request, any other
protocol you like).
- an interaction between an active data set and a player.
- a game or interaction between 2 players in the same space.
- a game or interaction between a player and a physical space
facilitated by a computer.
- anything else in the spirit of the assignment you can imagine.
```

•

Playful Interaction
Teams of 2, 40%

A game or playful interaction must have rules in some variety. This can mean the player can intentionally control and play with your creation to achieve a goal of some sort. Your creation may (but is not required to) have a points system, and should progress in some capacity as the user interacts.

You can use:
- Makey Makey.
- MQTT.
- Web sockets.
- bluetooth.
- Arduino.
- Camera / video / Kinect.
- Anything else you can imagine

•

Playful Interaction
Teams of 2, 40%

Grading and submission

Create a GitHub repository for this project shared by both group mates. Upload your playtest results in your readme file with photos, notes, and description of feedback you received during the play test by midnight of the playtest. Upload your final work by midnight of the due date.

20% Playtest (Preparedness, exploration, critique, documentation, uploaded day of play test.
20% Documentation (Readme file, screen shots, commented code, photo of physical interaction, instructions if needed).
20% Creativity and clarity of interaction. Is the interaction clear and interesting? Are players engaged?
20% Aesthetics and finish quality. Does it look polished?
20% Technical functionality and exploration. How does it work? Does it work ?! Did you learn something new, or brush up on existing skills?

Phase 1: Prototype & Play test
March 28/30th

We will host a play-testing session in class where we will all play each others' games/interactions and get feedback. Come to class prepared to have others play your work. This should include:
- A playable prototype (this can be rough! But it should indicate the interaction or concept)
- A mood board or sketches of final design
- A list of questions for play testers

Upload to Moodle and Github the day of the your play test by midnight with results.

Phase 2: Final playtest
April 11/13

Final results are due on the last day of the semester. Your end submission should include:
- A clear readme file that explains your work, has screen shots, photos, and instructions
- Well commented code
- Polished and finished aesthetic and interaction
- Clear and professional presentation of your work

Links

- https://makeymakey.com/
- https://www.youtube.com/watch?v=OyZNj7oMgek Running code on your phone or other device
- https://p5js.org/reference/#/p5/touches Touch on mobile devices

Device Movement

Acceleration

- deviceOrientation
- accelerationX
- accelerationY
- accelerationZ
- pAccelerationX
- pAccelerationY
- pAccelerationZ
- rotationX
- rotationY
- rotationZ
- pRotationX
- pRotationY
- pRotationZ
- turnAxis
- setMoveThreshold()