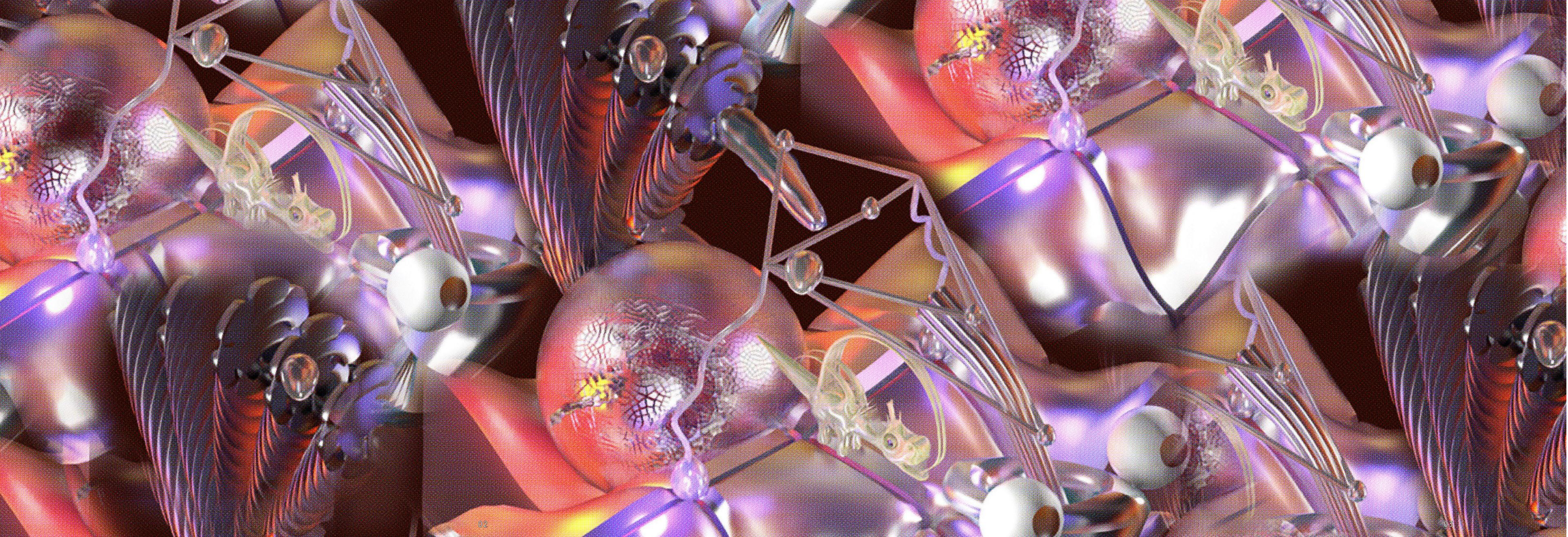


JENN LEUNG
DESIGN FOR PERFORMANCE AND INTERACTION
BARC0074 SKILLS PORTFOLIO



CONTENT

CINEMA 4D

FUTURE FOSSILS	04
WEEKLY TASKS	06

P5.JS

444.PHOTOBOOTH.444	08
WEEKLY TASKS	10

TOUCHDESIGNER

WHY ARE YOU CRYING?	12
SCREENSHOTS	14

FUTURE FOSSILS

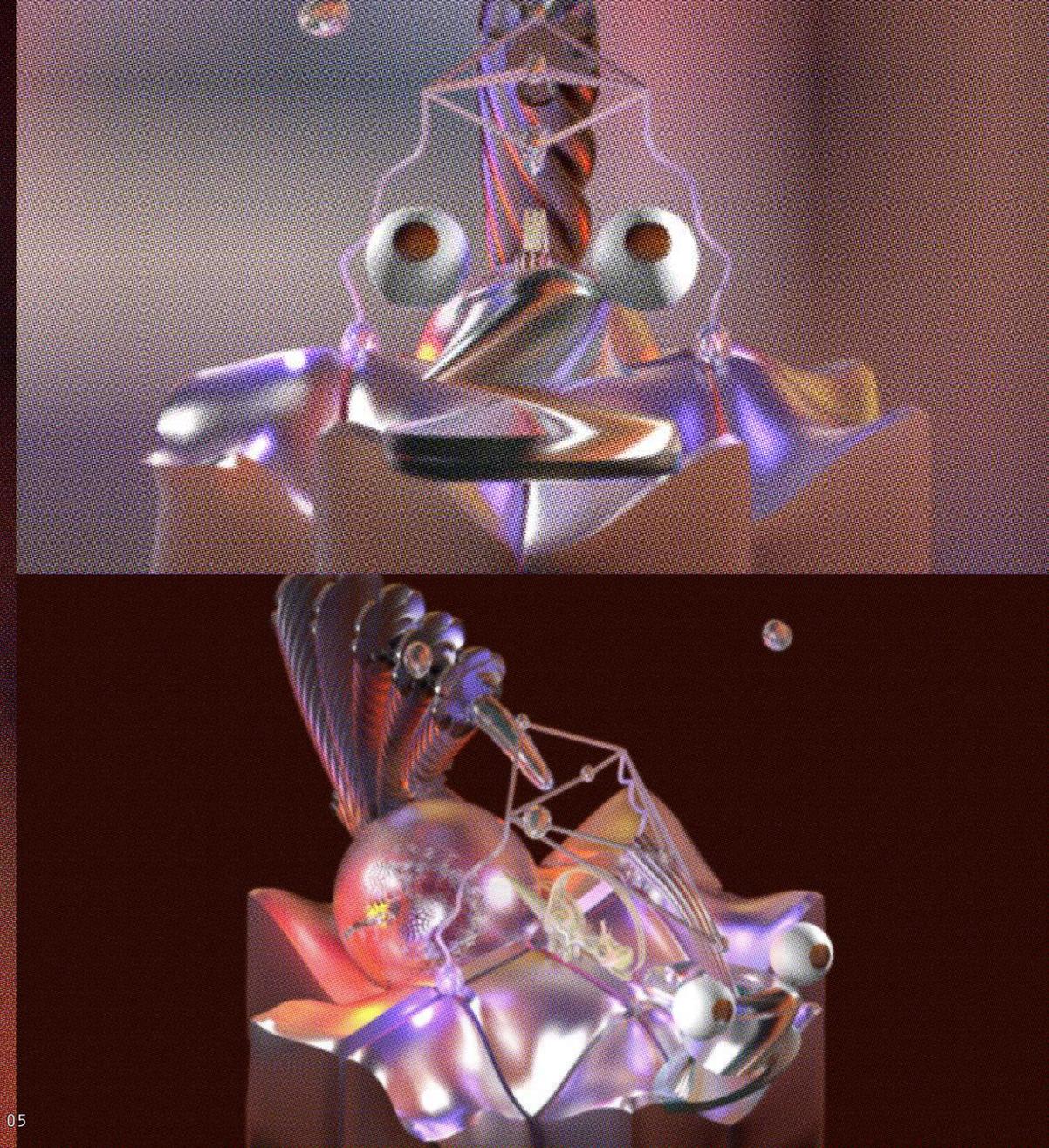
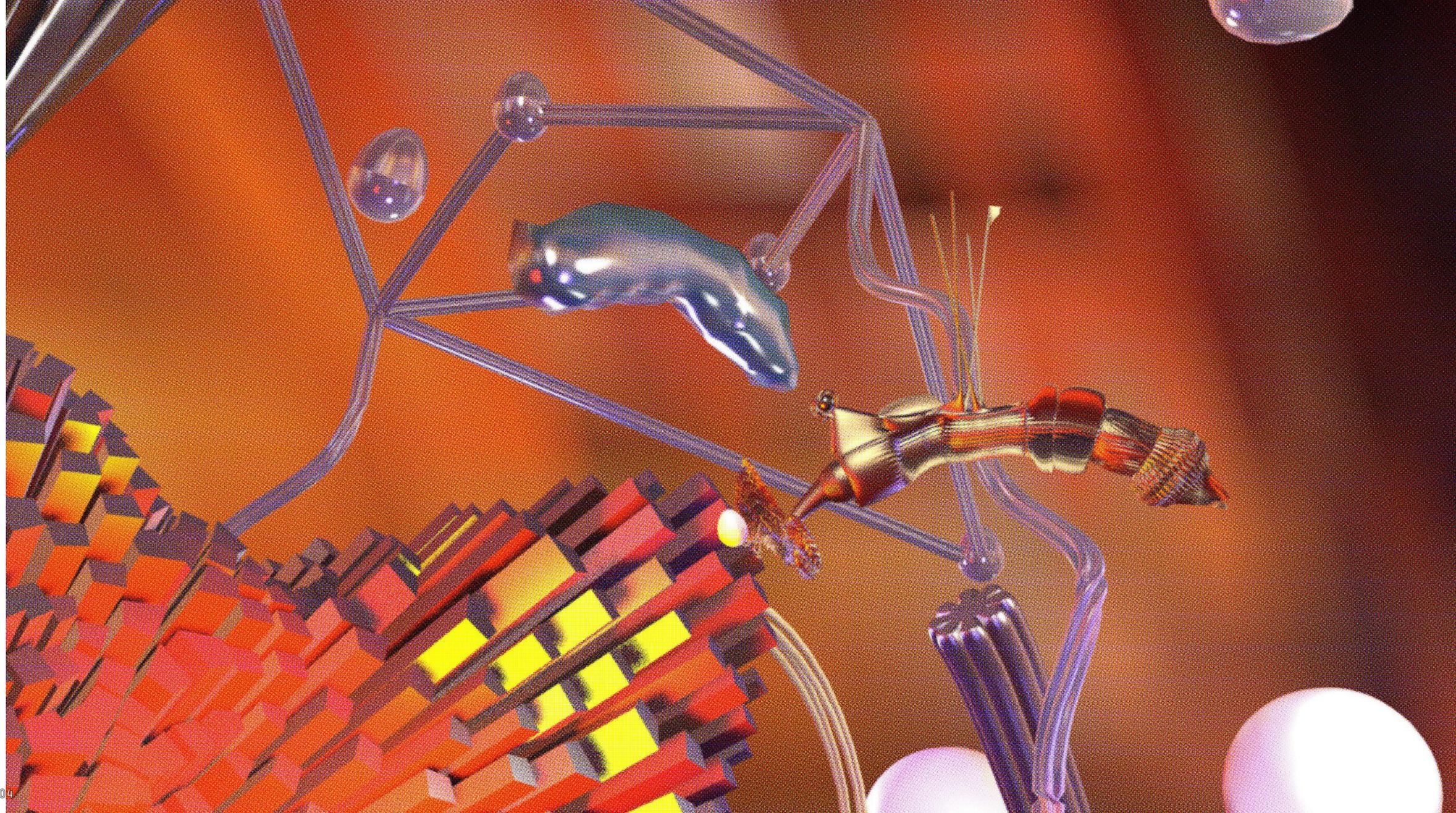
/// FUTURE FOSSILS IS A SPECULATIVE EVOLUTION PROJECT THAT REIMAGINES EXTINCT SPECIES IN A FICTIONAL SPACE

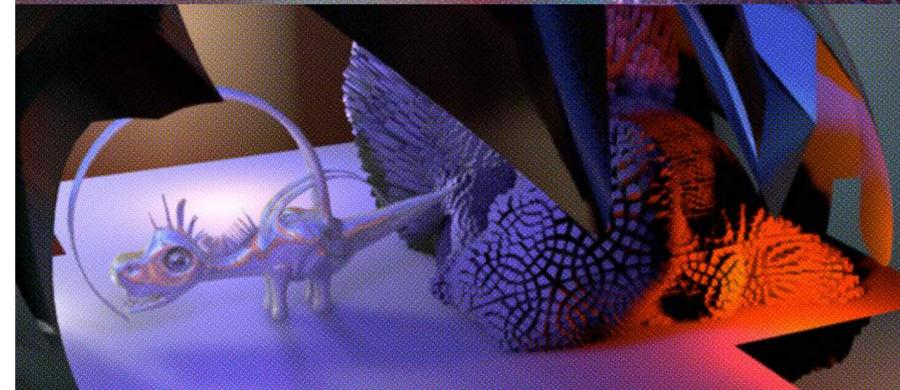
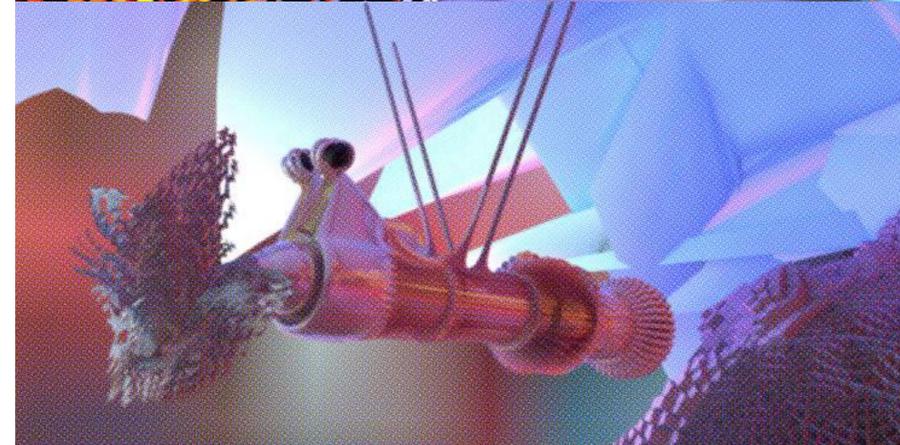
/// EXPLORING A NON-LINEAR ALTERNATE HISTORY/FUTURE HISTORY

/// SHAPES AND FORMS OF THIS PLAYGROUND ATTRACTION ARE AT ONCE GEOMETRIC AND ORGANIC, WHERE NAUFILOID, DINOSAUR, SEA LAMPREYS, BASALT COLUMNS AND OTHER GEOMETRIC STRUCTURES COLLIDE AND INTERACT WITH EACH OTHER ON THIS FROSTY/PASTEL/METALLIC PLOT.

/// ADAPTING 3D MODELING SKILLS AS A METHOD TO PRACTICE 'WORLDING', THIS PROJECT ENGINEERS SPATIAL COLLISIONS ACROSS INCONGRUENT TEMPORALITIES.

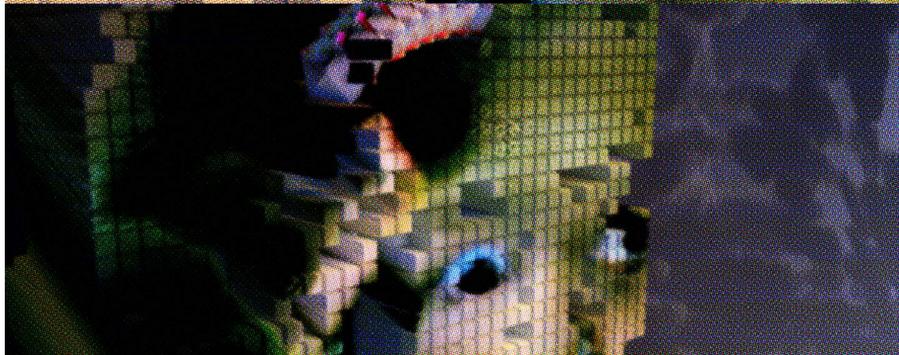
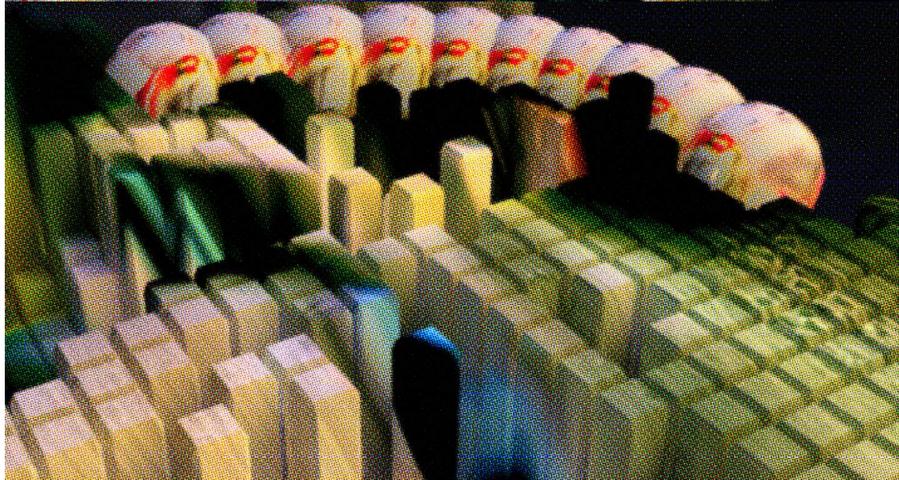
[VIDEO](#)





FUTURE FOSSILS:

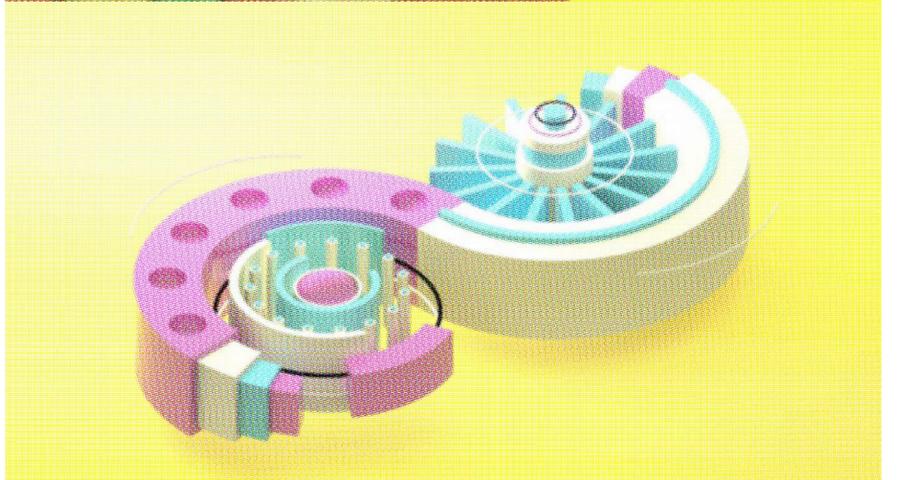
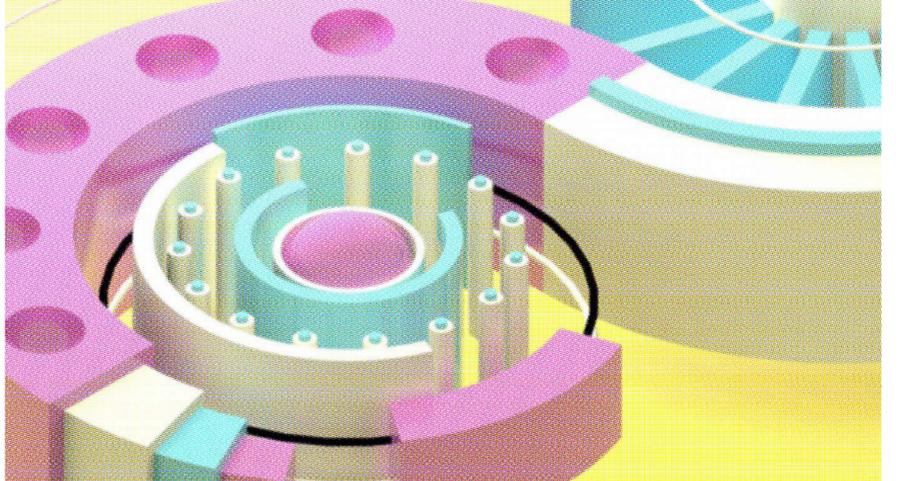
EARLY DEVELOPMENTS AND
ORGANIC MODELING



VOXELIZED:

6-SECOND ANIMATION WITH
MOGRAPH FIELDS

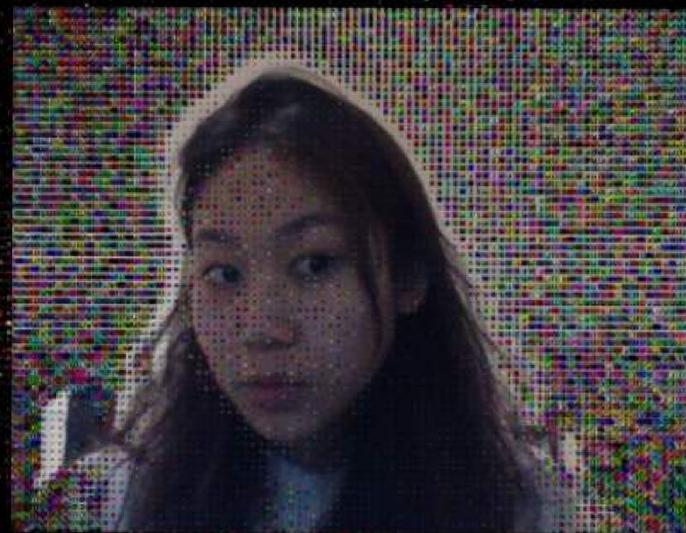
[VIDEO](#)



ASYMMETRY:

MODELING WITH
PRIMITIVE SHAPES

PHOTOBOTH..444



voxelcam

voxel

voxelInvert

voxelTexture

pixelSize: 4

pixelDepth: 4

noisecam

noisy

noiseInvert

noise Type

random

pixelSize: 4

pixelDepth: 2

bubblecam

bubble

bubbleInvert

bubbleMaterial

original

pixelSize: 4

pixelDepth: 7

distortcam

distort

distortInvert

pixelSize: 4

save picture



444.PHOTOBOTH.444

/// FINAL CODING PROJECT
/// INTERACTIVE CURIOSITIES

/// THIS PHOTOBOTH CONSISTS OF
FOUR CAMERA FILTERS:
/// VOXELCAM, BUBBLECAM, NOISECAM, DIS-
TORTCAM

/// THE INTERFACE ALLOWS USERS TO HAVE
CONTROL OVER DIFFERENT PARAMETERS TO
CREATE THEIR OWN UNIQUE FILTERS

/// CUSTOMIZABLE FILTERS CONTROLS:
FILTER ON/OFF, PIXELSIZE, PIXELDEPTH,
TEXTURE, MATERIAL, TYPE

/// OPTION TO "SAVE PICTURE"

/// ANGEL NUMBERS ARE REPEATING DIGITS
AND ECHO THE NATURE OF CODING IN FOR-
LOOPS/ NESTED LOOPS AND ITERATIONS

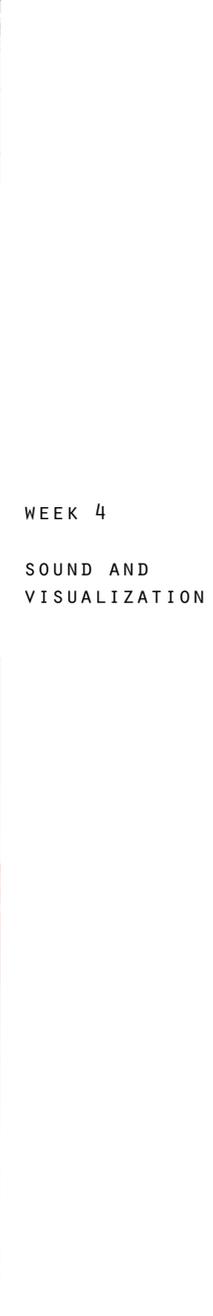
/// THESE FILTERS ARE ALL CREATED
THROUGH REPETITIONS, ECHOING THE THEME
OF GENERATION

[VIDEO](#)

```

if (mouseX>200 && mouseX <400){
  fill(180,100,100,50);
  ellipse(mouseX,mouseY,map(volu
  // abstract shape
  //moire shape!!!! was an accid
  fill(map(mouseX,200,400,0,360)
  beginShape();
  for (let i=70; i<290; i++){ //
    let x= 80* sin(i)+ 300;
    let y= 50* cos(i)+ 500;
    vertex(x,y);
    //mapping the upper vertices
    vertex(map(mouseX, 0,width,
  }
  endShape();
}
if (mouseX>400 && mouseX <600){
  fill(270,100,100,100);
  ellipse(mouseX,mouseY,map(volu
  // abstract shape
  //moire shape!!!! was an accid
  fill(map(mouseX,400,600,0,360)
  beginShape();
  for (let i=50; i<320; i++){ //
    let x= 60* sin(i)+ 500;
    let y= 50* cos(i)+ 500;
    vertex(x,y);
    //mapping the upper vertices
    vertex(map(mouseX, 0,width,
  }
  endShape();
}
if (mouseX>600 && mouseX <width){
  fill(360,100,100,100);
  ellipse(mouseX,mouseY,map(volu
  // abstract shape
  //moire shape!!!! was an accid
  fill(map(mouseX,600,800,0,360)
  beginShape();
  for (let i=120; i<260; i++){

```



WEEK 4
SOUND AND VISUALIZATION

```

let video;
let w=500;
let h=400;

function setup() {
  createCanvas(w, h);
  pixelDensity(1);
  video = createCapture(V
  video.size(w, h);
  background(0);
  video1= new VideoX(w,h)
  video.hide();
}

function draw() {
  video1.display();
}

class VideoX {
  constructor(w,h) {
    this.width = w;
    this.height = h;
  }

  display(){
    let x = 0;
    let k = 1000;
    video.loadPixels(
    copy(video, 0, 0,
    blendMode(DIFFERE
    copy(video, 0,0,
    x = x + 1;
    k= k- 1;

```



WEEK 3
CLASS AND OBJECTS

```

// draw
function draw(){

  let bgColor= map(mouseX, 0, 800, 0
  background(bgColor,20,100,100);
  let r = 30; // ellipse radius foll

  //loop to make tentacles display in
  for(j=0;j<6;j++){
    // question: can I use 'i' here
    let tentacleYPos = 50;
    displayTentacle(width/2-50+j*20,

  }

  r= 30* sin(frameCount*0.08);
  displayHead();

  ellipse(mouseX,mouseY,r,r);

  // conditional, mouse event interact
  if (mouseIsPressed){
    fill(180,30,100,60);
    ellipse(mouseX,mouseY, r*5, r*5);
    fill(0,0,100,80);
    ellipse(380,280,30,30);
    fill(0,0,100,80);
    ellipse(420,280,30,30);
    stroke(2);
    line(370,280,390,280);
    line(410,280,430,280);

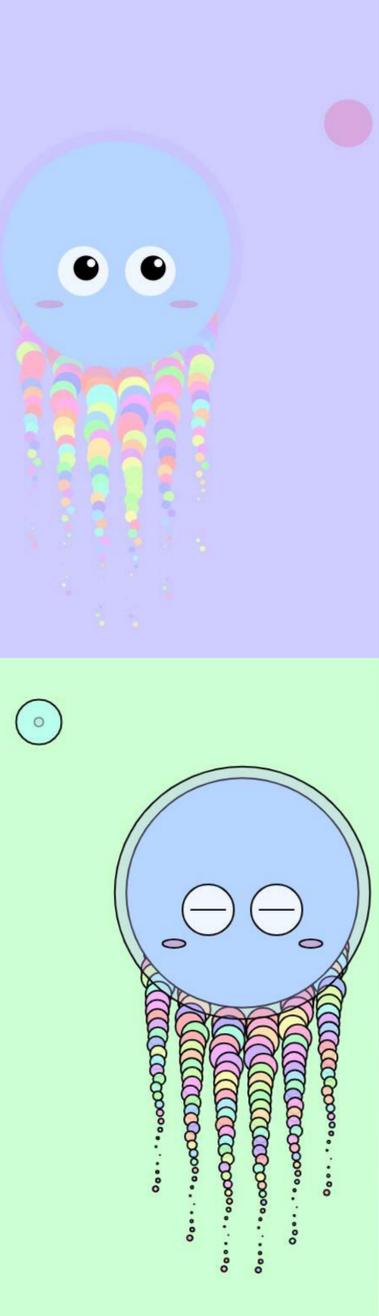
  }

  else{
    displayEye();
  }

}

function displayHead() {

```



WEEK 2
FOR-LOOPS AND PERLIN NOISE

```

function draw() {

  var ellipseColor = map
  var rectColor = map (mo

  stroke(ellipseX/2,mouseY
  strokeWeight(1);
  fill(ellipseColor, mouse
  ellipse(ellipseX, mouseY

  ellipseX-=2;
  ellipseScale *= 1.005;

  stroke(rectX/2,mouseY/2
  strokeWeight(1);
  fill(rectColor, mouseY/2
  rect(rectX, mouseY-15,r

  rectX+=2;
  rectScale *= 1.005;

}

function mousePressed() {
  redraw();
  blendMode(DIFFERENCE);
  background(0,0,100,100)

```



WEEK 1
ANIMATING WITH VARIABLES

WHY ARE YOU CRYING?

/// AN AUDIOVISUAL EXPERIMENT
EXPLORING OBJECT-ORIENTED
PLAYFULNESS

/// INSPIRED BY WONG KAR-WAI'S
CHUNG KING EXPRESS
AND THE CONVERSATIONS BETWEEN
ACTORS AND INANIMATE OBJECTS

/// THIS PROJECT REMAGINES THE
SECRET LIVES OF OBJECTS AND
PROVIDES A LENS INTO THE WAY
OBJECTS PERCEIVE THE WORLD:

/// FRAGMENTS OF SOUND AND VISION
ARE CAPTURED THROUGH COLOR
TRACKING AND MOTION TRACKING.

[VIDEO](#)

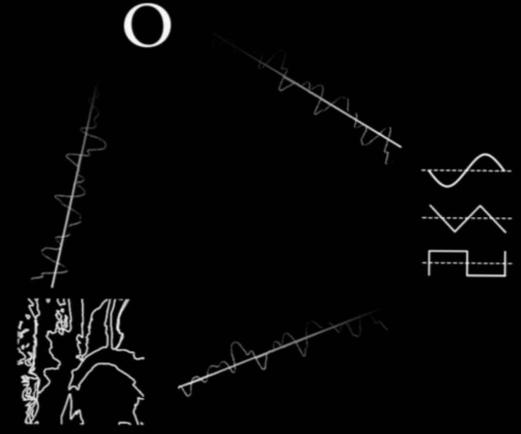


WEBCAM SCREENGRAB FROM TEAMS

COLOR TRACKING AND MOTION TRACKING

VISUALS CAPTURE TRAILS OF MOTION
LEFT BEHIND BY PHYSICAL OBJECTS

CHOREOGRAPHED BY SOUND WAVES



AUDIO SYNTHESIZER

SOUNDS GENERATED BY MOVEMENT COORDINATES

