Assignment 1:

A sequence of primitives

Design 240

Designing With Data

Etienne Naude

768485633 | enau831

Link to blog

https://awesome-swirles-f57a4f.netlify.app/blogs/processing.html

Link to MIRO

https://miro.com/app/board/o9J_l4D5A8o=/

Link to Github

https://github.com/etinaude/Des-240

Screenshots of my blog, miro board and my code are on the next 6 pages

Blog screenshots

ETIENNE NAUDE

Home Projects Blog About Me Contact

PROCESSING ANIMATION

FINAL GIF

DOWNLOAD GIF

FERTILITY RATE

In the latest census stats NZ found that New Zealand's fertility rate fell from 1.71 (2018) to 1.61 (2020) births per woman (BPW). The fertility rate of a country is a very broad indicator of its general health and well being. According to the demographic transition model, when a country improves its health, it reduces its death rate and decreases infant mortality rate. This then means that there is less of a need to replace the population and as such, the fertility rate also decreases. When the fertility rate drops to (or below) 2, a country is likely to be in stage 4 and nearing stage 5 of the demographic transition model.

This shows that New Zealand has relatively and very broadly speaking good health and well being. The fertility rate shows the average number of children born to a woman across their lifetime. If the fertility rate is at or below 2.0, then the population will drop as there are not enough babies to replace the population. If the fertility rate is slightly above 2.0 then the population will stay stable (it cannot be exactly 2.0 to account for early deaths). If the fertility rate is much higher than 2.0 then the population will increase.

OTHER FACTORS

Since the fertility rate is a very broad indicator, it doesn't take into account many considerations. For example, for many years, China had a very low fertility rate, not from good health care but rather from the One-child-policy. It also doesn't take into account population changes from immigration or skewed demographics.
Despite having many edge cases, fertility rate is still a useful metric to broadly look at a counties health. This is why I decided to create a visual representation of New Zealand's fertility rate.

INITIAL PROTOTYPES

I started by creating a range of initial prototypes to find the best way to approach the problem. From this, I received lots of feedback from peers, as the last three prototypes were the more popular ideas. This was because they were seen as more novel and more interesting to watch. I further developed a higher fidelity prototype based on the cellular automata idea.

CELLULAR AUTOMATA

THE RULE SET

The cellular automata I created was based on Conway's game of life. The game of life is a zero-player game that is meant to create a very minimal and basic simulation of populations. Because of this link, I thought it was ideal for showing how populations change over time based on their fertility rate.
I created the rule set of this automata to be based on the fertility rate of New Zealand with a margin of error of less than 1%. All my calculations and the rule set can be found on this spreadsheet.

AUTOMATA(D) NEW ZEALAND

Once I created the rule set, I made a hexagonal map of New Zealand to use as the grid for the game to be played on. This looked great but had a few flaws. Firstly it was reasonably

THE RULE SET

The cellular automata I created was based on Conway's game of life. The game of life is a zero-player game that is meant to create a very minimal and basic simulation of populations. Because of this link, I thought it was ideal for showing how populations change over time based on their fertility rate.
I created the rule set of this automata to be based on the fertility rate of New Zealand with a margin of error of less than 1%. All my calculations and the rule set can be found on this spreadsheet.

AUTOMATA(D) NEW ZEALAND

Once I created the rule set, I made a hexagonal map of New Zealand to use as the grid for the game to be played on. This looked great but had a few flaws. Firstly it was reasonably accurate in showing how populations change and adapt, growing and shrinking based on the amount of land they have. Unfortunately, it didn't accurately show what New Zealand's population is doing right now for longer than a single frame. Secondly, it is fairly difficult to understand the data being represented without an in-depth explanation, which means the data is being represented poorly. Since representing the statistic is the ultimate goal of the project, this is more important than aesthetics. On top of this, after receiving some feedback, it was pointed out to me that my design didn't meet the assignment parameters.

MEETING PARAMETERS

To meet the assignment parameters, I needed to create a 1000px*1000px gif using only black and white, which meant that representing live cells, dead cells, and ocean cells would be very difficult. When I edited the programme to meet the requirements, I didn't like the aesthetic, which meant that it now neither looked good nor was it easy to understand.

NEW AUTOMATA

From this, I decided to create an entirely new automata using the New Zealand Grid. This automata is much simpler, and it is easier to understand. It randomly selects a portion of the population and removes them. This is repeated to show generations. While this did meet the requirements of the assignment and looked good, it gave the statistic a negative spin. It looked as though the entire population of New Zealand was about to be extinct in a few generations. As I am meant to merely represent the statistic so it would be easy to understand and not add my own opinions about whether that statistic is good or bad, I found this to be represented too much bias.

FINAL REPRESENTATION

After receiving lots of feedback on my different automata, I decided to do a full pivot. I changed from representing on an countrywide scale to a individual scale.
The gif is split into five stages.

1. The older generation exists and is stable.
2. The first child is born.
3. The second child is born.
4. The older generation dies.
5. The new generation replaces the old.

This final representation shows just one generation of the fertility rate, which is good as it doesn't speculate as to how it will change or stay the same. Additionally, it is very clear to see the five stages of the animation and how the older generation is being replaced, but the resolution rate is not high enough to entirely replace the older generation.

This ended up being a much simpler idea to implement, looking good and representing the data well.

RESOURCES

CODE

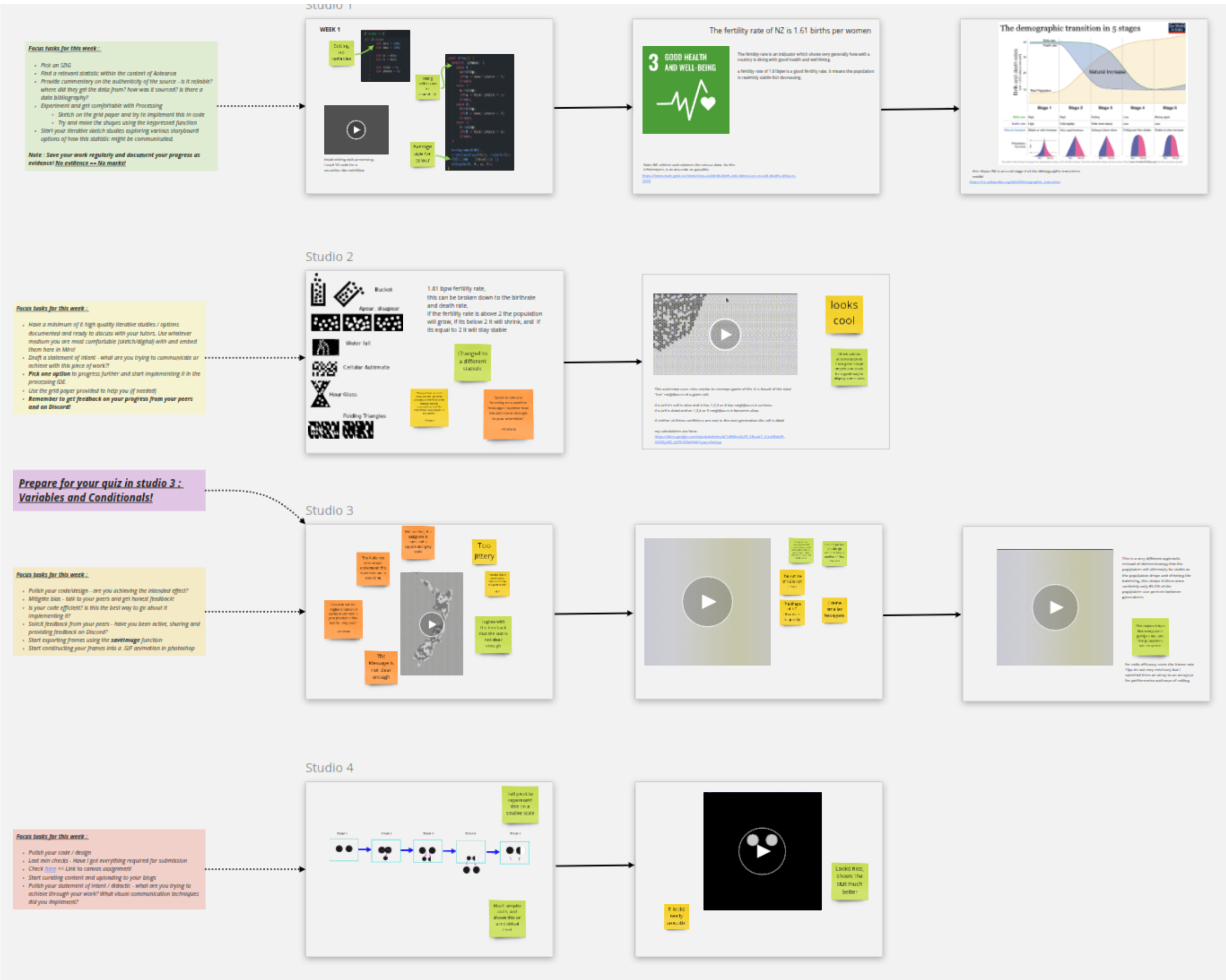
The code for all stages of this project can be found in my Github repo. The previous prototypes can be found in the commits; there are not many, so it should be relatively simple to find each version.

PLANNING AND PROGRESS

The weekly progress which I made and the feedback I received from peers and instructors can be found on my miro board. This also shows more detail about each of the designs.

Previous Post Next Post

Miro Screenshots



Code

Please just look at it on Github, its much better formatted
https://github.com/etinaude/Des-240/blob/main/a1/a1.pde

```
/**
 Des 240 Assignment 1, showing data through geometric primitives


 by Etienne Naude, 2021

 showing fertility rate of New Zealand - 1.61 bpw.
 http://www.stats.govt.nz/statistics/geos/geos-datasets/fertility-rate-nz/
*/

int stage = 0;



/**
 a class to storage the location of a human
*/
class Human {
 public float x, y, extent, percent;

 Human(int x, int y, int extent, int percent) {
   this.x = x;
   this.y = y;
   this.extent = extent;
   this.percent = percent;
 }

 void draw(){
   if(percent == 0) return;
   fill(255);
   circle(x, y, extent);

   if(percent < 100){
     fill(0);
```

```
    rect(x+extent*(percent-50)/100, (y-extent/2)-2, extent+10, extent+10);
   }
  }


 }


 // initalize the humans
 Human parentA = new Human(600, 400, 100, 100);
 Human parentB = new Human(400, 400, 100, 100);


 Human childA = new Human(500, 400, 50, 0);
 Human childB = new Human(500, 400, 50, 0);


 void setup() {
  size(1000, 1000);
  noStroke();
  frameRate(60);
 }


 void draw() {
  if(stage > 5) return;
  background(0);


  // draw each circle
  childA.draw();
  childB.draw();


  parentA.draw();
  parentB.draw();


  // animate based on stage of life
  switch (stage) {
    // reporductive stage
    case 0:
      comeTogether();
```

```
      break;
    // first child
    case 1:
      if(childA.y > 600) stage++;
      childA.percent = 100;
      moveAway();


      childA.x--;
      childA.y+=2;
      break;


    // reporductive stage
    case 2:
      comeTogether();
      break;
    // second child
    case 3:
      if(childB.y > 600) stage++;
      childB.percent = 61;
      moveAway();


      childB.x++;
      childB.y+=2;
      break;


    // death
    case 4:
      if(parentA.y > 1200) stage++;
      parentA.y+=3;
      parentB.y+=3;
      break;


    // growing up
    case 5:
      if(childA.extent > 100) stage++;
      childA.extent+=1;
```

```
          childB.extent+=1;


          childA.y-=3;

          childB.y-=3;


          childA.x-=1;

          childB.x+=1;

          break;

      }

     }


     void moveAway(){

      if(parentA.x >= 650) return;

      parentA.x++;

      parentB.x--;

     }


     void comeTogether(){

       if(parentA.x <= 500) stage++;

        parentA.x--;

        parentB.x++;

     }
```