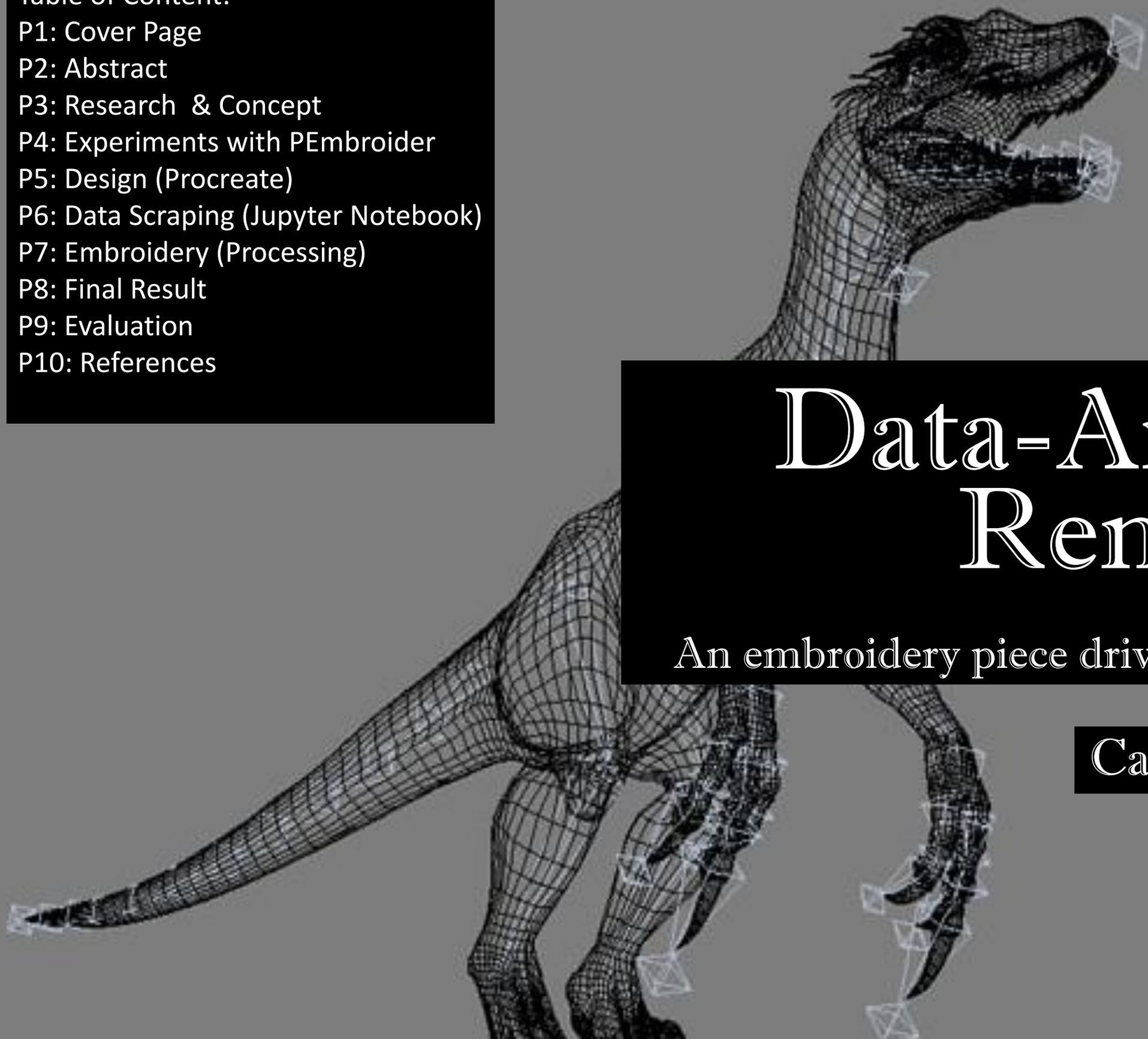
Table of Content:



Data-Animal Remnant

An embroidery piece driven by Big data

Canace CHEN

(Turbosquid, 2015)

Abstract

"Data-Animal Remnant" is an embroidery project driven by big data, investigating the convergence of digital life, de-extinction, and the transformative power of large-scale data analysis. Employing the processing.embroider library and PEmbroiderGraphics, the artwork showcases the potential of big data in creative expression.

The project incorporates two primary data sources: a comprehensive website featuring endangered animals worldwide and an API from Steam, a popular gaming platform. By extracting animal groups and quantifying endangered species from the first source, the project establishes connections between these groups and the gaming world represented on Steam.

Through the interplay of embroidery and big data, "Data-Animal Remnant" prompts contemplation on the evolving boundaries between animals and machines in the digital age. It challenges traditional notions of animals as solely physical beings, as digitalization extends beyond a mere recording of their physical existence (Adams, 2020).

Moreover, the project raises thought-provoking questions about the authenticity of digitized creatures. While advanced technologies enable the resurrection of extinct animals like dinosaurs in virtual worlds, "Data-Animal Remnant" questions whether these recreations truly embody the essence of the original species.

"Data-Animal Remnant" underscores the significance of big data in reshaping our understanding of animals and their relationship with the digital realm. By bridging embroidery and data-driven insights, the project invites viewers to reflect on the ethical considerations, blurred boundaries, and profound impacts of big data on our perception of the animal kingdom and the rapidly evolving landscape of technology.

Research & Concept

Research:

Technological progress has empowered ecologists to study the impact of human development on ecosystems using electronic tools. These tools enable the generation, storage, sharing, and analysis of large quantities of data, revolutionizing ecological research.

Previously, wildlife tracking relied on radio-tracking and offered limited data points. However, the introduction of GPS recorders, accelerometers and other bio-logging tools in the early 2000s ushered in the era of "big data" in wildlife tracking. This enabled researchers to gather extensive information on the movements of marine megafauna in the ocean (Grémillet et al., 2022).

Furthermore, a groundbreaking technique called environmental DNA (eDNA) metabarcoding has emerged. This approach utilizes DNA fragments present in the environment to identify individuals, species, and communities. By studying eDNA, researchers can track organisms, explore their distributions, and examine changes in biodiversity across different time periods and locations (Taberlet et al., 2018).

These technological advancements have significantly advanced ecological understanding, providing scientists with unparalleled access to vast amounts of data. They have opened up new avenues for research, facilitating comprehensive exploration and conservation of the natural world.

Concept:

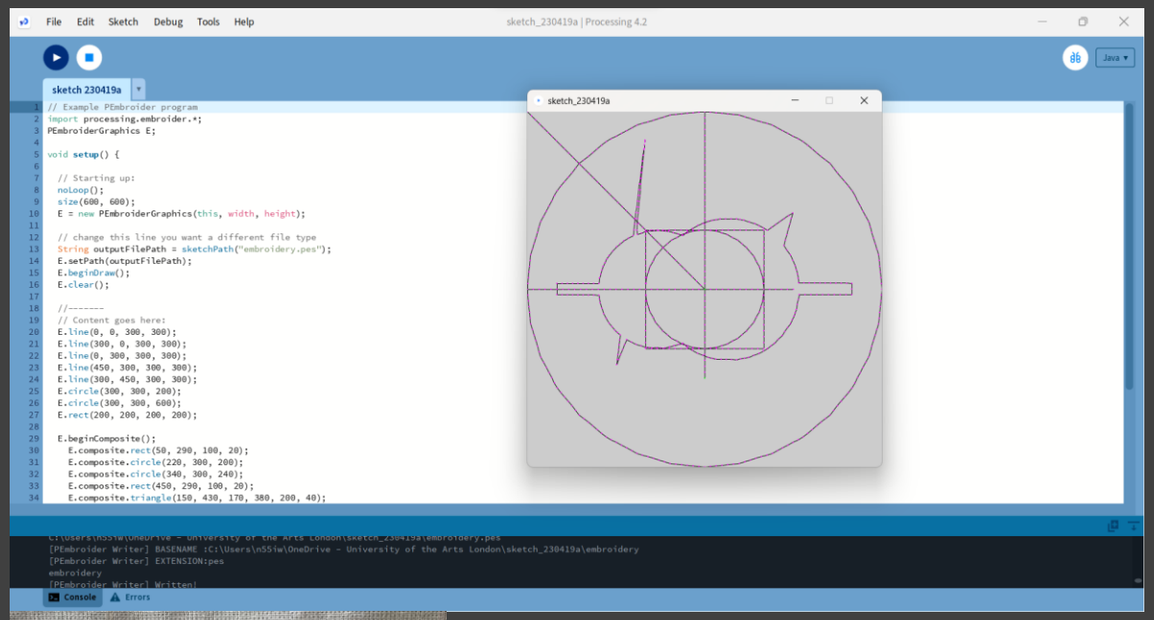
"Data-Animal Remnant" derives its name from the idea of using data remnants from different sources, including endangered animals and video games, to create an embroidered reflection on the coexistence of digital life and the potential for de-extinction.

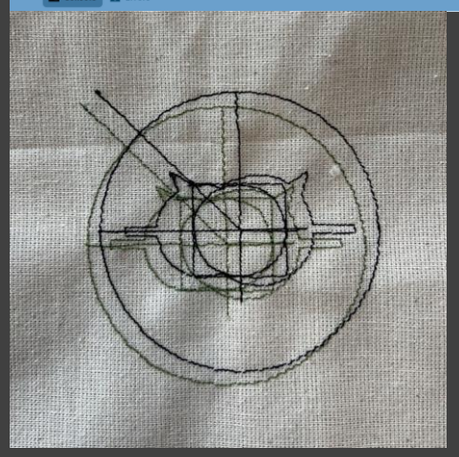
From the first data source, animal groups and their respective numbers of endangered animals were extracted. These numbers determined the density of the animals represented in the embroidery. Higher numbers resulted in lower density, highlighting the alarming disappearance of these creatures. The second data source, the number of games related to each animal group on Steam, determined the frequency of each animal's appearance in the embroidery. The more games associated with a specific animal group, the greater the representation of that animal.

The project delves into the concept of digital life and de-extinction, contemplating the authenticity and identity of digitally resurrected creatures. As technology continues to advance, animals are being digitized for purposes ranging from military applications to animal conservation. The project prompts us to question whether the virtual representation of an extinct animal truly captures its essence or merely manifests as a human-made artifact.

Experiments with PEmbroider

Experiment 1:

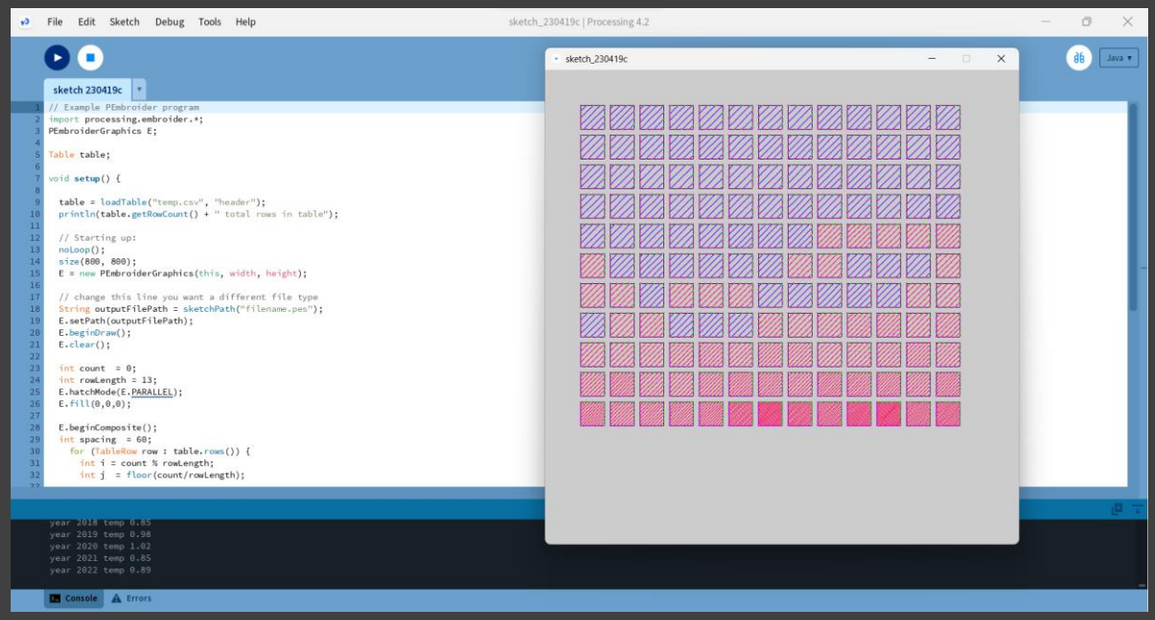




Embroidery Video: https://drive.google.com/file/d/1TjnaklAS9Zbn1Arr5RD9XChwxs5LajI9/view?usp=sharing

This was the first time using the PEmbroider Library. I experimented with different shapes. Some of the shapes are merged together by using E.beginComposite() and E.endComposite(). After drawing the pattern, E.visualize(true, true, false) would display it on screen. When the pattern is ready to be embroidered, use E.endDraw() to save the embroidery file. sketchPath(".pes") and E.setPath() are used to set the file path.

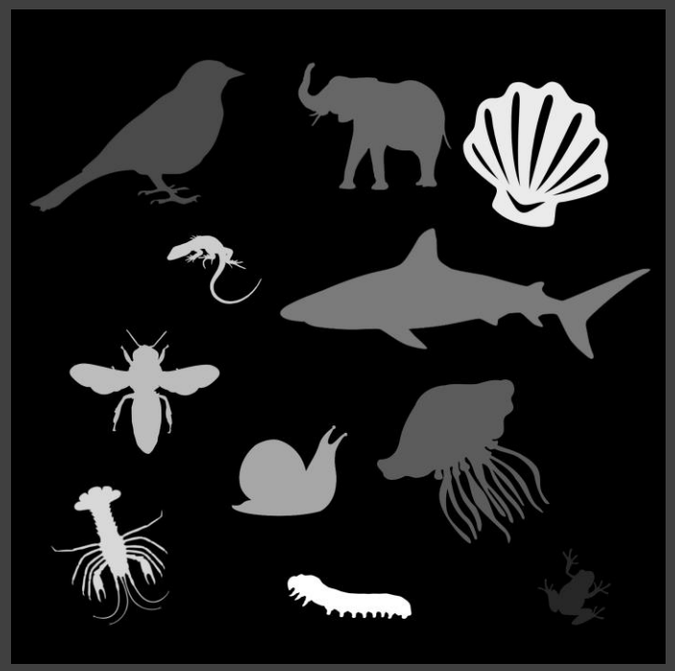
Experiment 2:



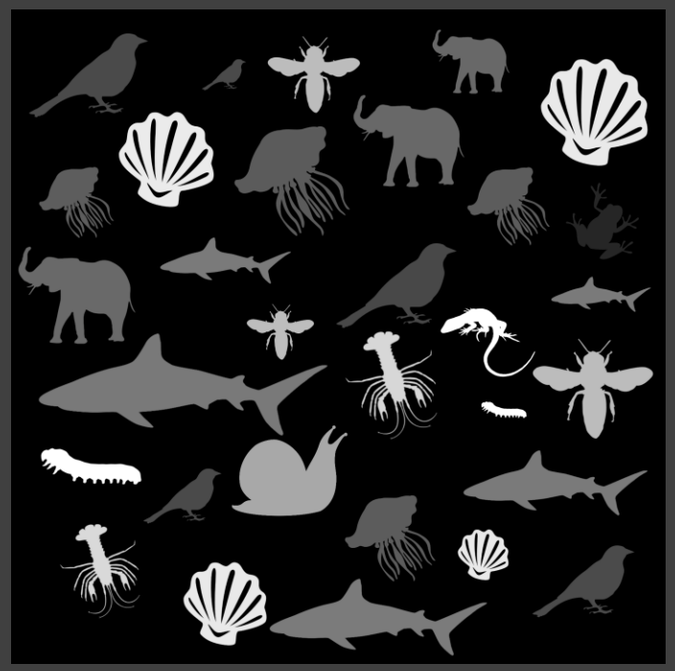
The second experiment was using the PEmbroider Library for data visualisation. The data used was from a .csv file containing smoothed and unsmoothed data on temperature from 1880 to 2022. The year and unsmoothed temperature columns were used. Each square represents a year, and its colour and density were determined by the temperature in that year. When temp < 0.0, the colour would be blue and if temp > 0.0 && temp <0.5, the colour would be red. In addition, the temp data was mapped from (-0.47, 1.02) to (10.0, 2.0) to affect the density of the stitching. This means as the temperature gets higher, the square would be more densely stitched. From the result pattern, it is obvious that the temperature has greatly increased from 1880 to 2020.

Design (Procreate)

Design 1:

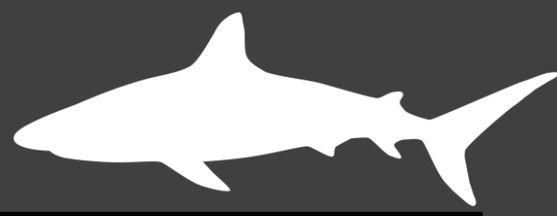


Design 2:

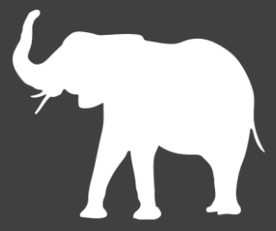


The 2nd design kept the use of the number of endangered animals in each animal group to control the density. However, it added multiple numbers of each animal. The number of each animal was determined by the number of games contains/about the animal groups on Steam, as these games has created ‘digital copies’ of them in the virtual world. Each animal will be put in a random position within the frame without overlapping. These changes gives the pattern a fuller and more aesthetic look.

As there will be two data sources used in the pattern, there will be two variables impacting the pattern’s look. The first design had the number of endangered animals in each animal group to control the density; larger number of endangered animals' results in lower density, to show these animals are disappearing. The number of games contains/about the animal groups on Steam to control the size of each animal, as they appear in the digital world more commonly.

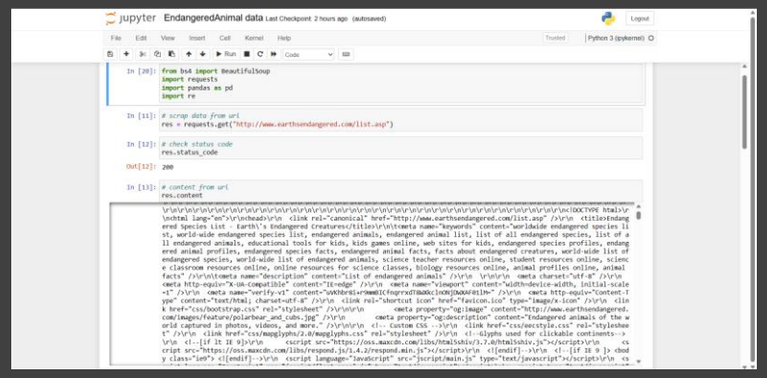


Animals design:
I drew all these animals for each animal group.

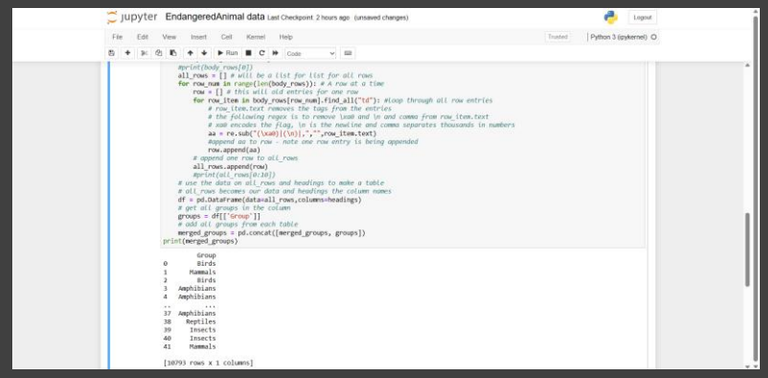


Data Scraping (Jupyter Notebook)

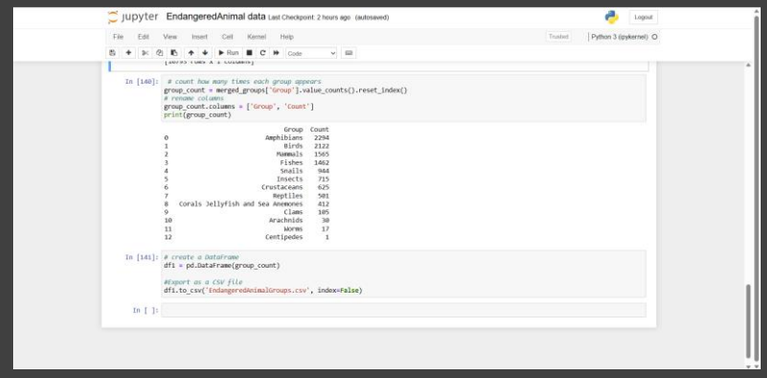
Data Source 1: Worldwide Endangered Animals (website)



The first data source is a website containing worldwide endangered animals. The data was scraped by using requests.get(web_url).

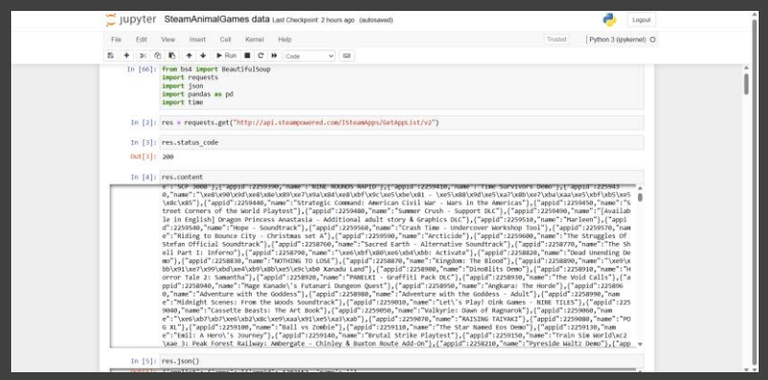


The heading and body rows are found by using BeautifulSoup find_all(tag). The website contains 26 tables (a-z), therefore a nested for loop was used to loop through all tables.

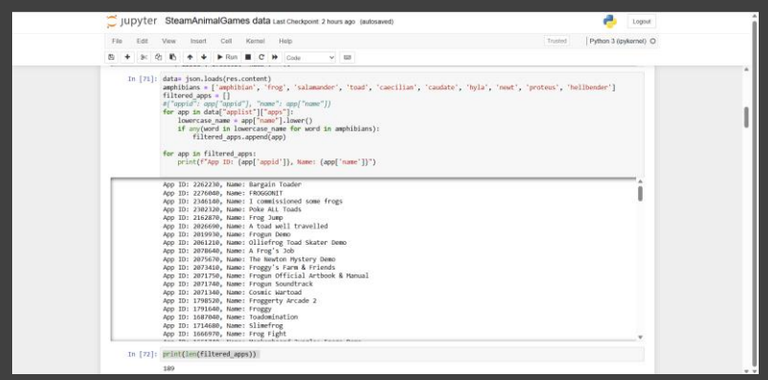


After getting only the Group column of all tables, a new table was created with the all the groups (Group) and the number of endangered animals it has (Count). The new table was exported as a .csv file using pandas DataFrame.

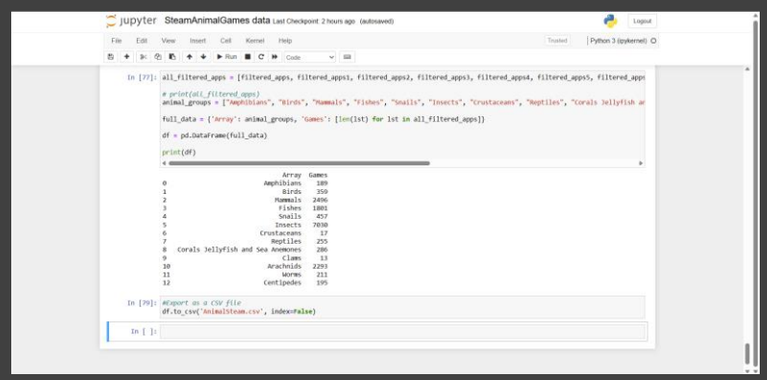
Data Source 2: Steam GetAppList (API)



The second data source is an official API from Steam and managed by Team Fortress. The API, GetAppList, contains a full list of every publicly facing program in the Steam store/library. The data was scraped by using requests.get(api_url).



The all the data was load as a Json file. To get game that relates to each group of animal, a list of 10 most related words to the animal group was created to check if the games' names contain any of these words. If so, append them to the filtered_apps list.



After finding all games relating to each animal groups, a new table was created with the all the groups (Group) and the number of games relate to it (Games). The new table was exported as a .csv file using pandas DataFrame.

Embroidery (Processing)

Load and use data:

```
table = loadTable("AnimalSteam.csv", "header");
imageCounts = new int[numImages];
//println(table.getRowCount() + " total rows in table")

//get data from the animal games on steam table
for (int i = 0; i < numImages; i++) {
  TableRow row = table.getRow(i);
  // use the number of games for each group as number o
  int value = row.getInt("Games");
  // map games value to 1 (appear once) to 6 (appear 6
  // as the range is too big and there are big gaps
  //(eg: between insects(7030) and mammals(2496, the cl
  imageCounts[i] = int(map(value, 13, 7030/3, 1, 6));
}
```

```
// Generate a list to store the indices of the images
ArrayList<Integer> imageIndices = new ArrayList<Integer>();

// Add image indices based on the desired appearance counts
for (int i = 0; i < imageCounts[0]; i++) {
  imageIndices.add(0);  // Animal 1
}
for (int i = 0; i < imageCounts[1]; i++) {
  imageIndices.add(1);  // Animal 2
}
for (int i = 0; i < imageCounts[2]; i++) {
  imageIndices.add(2);  // Animal 3
}
```

Load table 1 and map the values in the Games column from (13, 7030/3) to (1, 6). Create an ArrayList to store record the number of images. Use for loop to set the total number of each animal images to the value mapped from data.

```
table2 = loadTable("EndangeredAnimalGroups.csv", "heade
densities = new float[numImages];
//println(table.getRowCount() + " total rows in table")

//get data from the endangered animal groups table
for (int i = 0; i < numImages; i++) {
  TableRow row = table2.getRow(i);
  // use the count for each group as density (the more
  float value = row.getFloat("Count");
  // map count value to 7.0 (least dense) to 2.0 (most
  densities[i] = map(value, 2294.0, 1.0, 6.5, 2.0);
}
```

Load table 2 and map the values in the Counts column from (2294.0, 1.0) to (6.5, 2.0) and use mapped data as density.

Within screen and no overlap:

```
//create an ArrayList to store the postions of the anim
ArrayList<PVector> animals = new ArrayList<PVector>();
```

```
// Maximum number of attempts to find a non-overlapping
int maxAttempts = 50000;
```

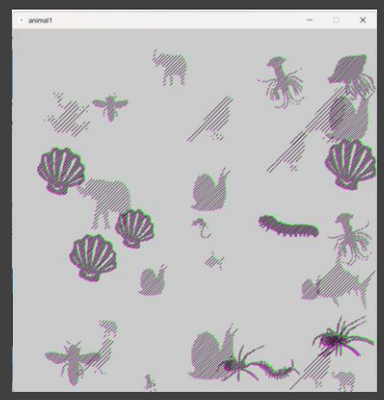
Create an ArrayList to record all position vector data. Set max attempt.

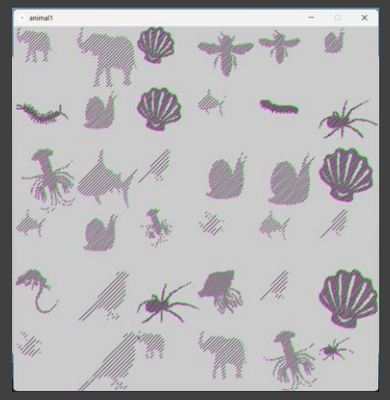
```
//try to find a position that does not overlap with oth
while (overlapping && attempts < maxAttempts) {
  // Generate random position within the canvas
  x = int(random(width));
  y = int(random(height));

  // Check if the new circle overlaps with any existing
  overlapping = checkOverlap(x, y, size);

  //if not overlapping
  if (!overlapping) {
    //keep animals in the frame
    x = constrain(x, size/2, width-size);
    y = constrain(y, size/2, height-size);
    //println(overlapping);
    //println(x + "+" + y);
    // Add the circle position to the list
    animals.add(new PVector(x, y));
    println(animals);
    // Draw the image
    E.image(imgs[i], x, y, size, size);

    break;
  }
  attempts++;
}
```

Use a while loop to try to find a non-overlapping position. If found, ensure the image stays in screen, add position to ArrayList and draw the image.

```
//check if the animals are overlapped
boolean checkOverlap(int x, int y, int size) {
    // Check if the new circle overlaps with any existi
    for (PVector animal : animals) {
      float distance = dist(x, y, animal.x, animal.y);
      if (distance < size) { // Minimum distance betwee
        return true; // Overlapping
      }
    }
    return false; // Non-overlapping
}
```

Check if images overlap.





The 1st pattern was from a discarded sketch. it draws the animals in random positions, but even with the checkOverLap function, the animals still overlaps. As the animals' sizes are random and the non-overlapping distance depends on the size of the closest animals, it is impossible to make sure they all maintain a perfect distance. Therefore, as show in the 2nd pattern, the animals are place in a 6*6 grid instead. This way it is easier to ensure the animals do not overlap.

```
    // Display the images in the grid
    int imageIndex = 0;  // Index of the current image in the
  for (int row = 0; row < gridSize; row++) {
    for (int col = 0; col < gridSize; col++) {
      int x = col * (width / gridSize) +5;  // X-coordinate o
      int y = row * (height / gridSize);  // Y-coordinate of
      int size = int(random(60, width / gridSize + 15)); //se

      // Get the current image index from the shuffled list
      int currentImageIndex = imageIndices.get(imageIndex);

      // Get the density for the current animal
      E.hatchSpacing(densities[currentImageIndex]);

      // Display the current animal
      E.image(imgs[currentImageIndex], x, y, size, size);

      // Move to the next image index
      imageIndex++;
    }
  }
```

Using nested for loop to place animals on a grid.

Final Result

Video:
https://drive.google.com/file/d/1FYMA1CJjdYaKFRKTr_BcS4UsIi9JIF8J/view?usp=sharing

Full code:
https://git.arts.ac.uk/21020295/Data-Animal-Remnant

Evaluation

Self-evaluation:

In this project, I merged embroidery, big data, and digital life. I transformed data into tangible art with processing and embroidery. By extracting information from a website on endangered animals and the Steam API for game data, I explored the relationship between animal conservation, digital existence, and human influence.

Through embroidery, I visualized the declining presence of endangered animals. Their density in the artwork reflected their dwindling numbers. Additionally, the number of games related to each animal group determined their representation in the embroidery, showcasing the concept of digital life and de-extinction.

"Data-Animal Remnant" integrated art, data, and philosophical inquiry, urging reflection on digital life's impact on our perception of reality. It emphasized the fragility of endangered species and invited contemplation on our role in conservation efforts and the blurred boundaries between the natural and the human-made.

Possible improvement:

1. The method used in finding all games on Steam related to each animal groups needs improvement in its accuracy. As words appeared in the name of games do not necessarily relate to its content.

2. While the concept of using embroidery to represent the density of endangered animals and their digital counterparts is intriguing, there is room for further refinement in the execution of the artwork itself. Consider exploring different mediums and technologies that can effectively convey the intended message and evoke a stronger emotional response from viewers.

1. The project touches upon thought-provoking questions about the authenticity of digital representations and the nature of digital life. To strengthen this aspect, consider expanding on the philosophical inquiry through additional research, references, or engagement with relevant theories.

References

1. Adams, W. (2020). *Digital Animals*. https://doi.org/10.17863/CAM.48779 (Accessed: 5 May 2023).
2. Alonso, N. M. (2021). PEmbroider: An Open Source Embroidery Library for Processing. Available at: https://www.youtube.com/watch?v=3yKEUhSFStc (Accessed: 19 May 2023).
3. Cameron, A. (2023). *Workshop: Generative Embroidery with PEmbroider.* Available at: Creating Digital Embro... | CCI Wiki (arts.ac.uk) (Accessed: 12 June 2023).
4. Cameron, A. (2022). *Creating Digital Embroidery Designs with Pembroider.* Available at: Workshop: Generative E... | CCI Wiki (arts.ac.uk) (Accessed: 12 June 2023).
5. Earth's Endangered Creatures (2023). Worldwide Endangered Animal List. Available at: http://www.earthsendangered.com/list.asp (Accessed: 1 Jun 2023).
6. Grémillet, D. et al. (2022). *Big data approaches to the spatial ecology and conservation of marine megafauna*. Available at: Big data approaches to the spatial ecology and conservation of marine megafauna | ICES Journal of Marine Science | Oxford Academic (oup.com) (Accessed: 20 May 2023).
7. Koech, K. E. (2020). *Web Scraping 1: Scraping Table Data*. Available at: https://towardsdatascience.com/web-scraping-scraping-table-data-1665b6b2271c (Accessed: 23 May 2023).
8. Python Pool (no date). What Causes "iopub data rate exceeded" Problem and How to Fix it. Available at: What Causes "iopub data rate exceeded" Problem and How to Fix it (pythonpool.com) (Accessed: 4 Jun 2023)
9. Taberlet, P. et al. (2018). *Environmental DNA: For Biodiversity Research and Monitoring*. Kettering: Oxford University Press.
10. Team Fortress (2021). *WebAPI/GetAppList.* Available at: WebAPI/GetAppList - Official TF2 Wiki | Official Team Fortress Wiki (Accessed: 28 May 2023).
11. Turbosquid (2015). *Jurassic Park Raptor 3d Model.* Available at: jurassic park raptor 3d model (turbosquid.com) (Accessed: 7 Jun 2023).