UNIVERSITY OF BIRMINGHAM

School of Computer Science
College of Engineering and Physical Sciences

Computer Science Project

Visualising Climate Change on Procedurally Generated Environments and Wildlife

Submitted in conformity with the requirements
for the degree of BSc Computer Science
School of Computer Science
University of Birmingham

Andreea-Ioana Bila
Student ID: 1784643
Supervisor: Dr Ian Kenny

April 2020

Andreea-Ioana Bila

Computer Science Project
Visualising Climate Change

Contents

1 Overview

1.1 Abstract

As the threats of climate change become ever more pressing, we find ourselves in need of tools to educate individuals and combat the abundant misinformation. Awareness is the first step towards eliciting action and change.

In this paper, I describe my attempt at tackling the aforementioned need for education by developing a system that simulates the effects of climate change on the environment. The aim of this tool is to provide a visualisation for the kind of changes we can expect to observe as a result of anthropogenic emissions. The system does not make predictions about future events, but rather it seeks to educate users on the harm we are causing to our environment, through projections based on extensive research. Exposure to such a system is expected to inform individuals on the severity of the climate emergency we are facing, and motivate them to start taking action to prevent future climate instability.

To achieve this, I create random environments using coherent noise and a temperature model based on latitude, terrain height and greenhouse gas concentration to generate terrain. I then add animals to habitats from the environment they can thrive in. Finally, I increase the greenhouse gas concentration, which triggers the rest of the environment to change. The outcome is a high-level visualisation of how global warming can transform any given environment and harm wildlife, based on the current available research on climate change.

"Change is the end result of all true learning."
– Leo Buscaglia

1.2 Acknowledgements

This project would have not been possible without the help and support of a few people.

To my Project Supervisor, Ian Kenny, for great advice and continuous guidance through my moments of revelation and lack there of, and for helping me not get distracted by the endless possibilities and lose track of my main goal.

To my friends, for constantly helping me test my system and suggesting improvements.

To my family, for endless support and reminding me to trust myself.

2 Introduction

Climate change is one of the next big challenges we will have to face as a society. After years of debate, hesitation and attempts to discredit the extensive proof and research accumulated surrounding the subject, the weight of evidence supporting the theory of global warming is more than sufficient to eliminate doubt and cause concern.

As misinformation spreads with ease, and individuals attempt to pass personal opinions and scepticism with the same weight as scientific fact, we find ourselves in need of ways to convey the message of urgency for action to mitigate climate change to the wider public. We require means to educate more groups of people on the matter and outline how the current state of affairs is causing harm to our environment.

Climate change is the phenomenon of disruption in normal, historical climate patterns that has started being observed within the past 50 years. Essentially, ever since the industrial revolution, greenhouse gas emissions have spiked up. Greenhouse gases are gases like carbon dioxide (CO_2), methane (CH4) and nitrous oxide (N_2O), which absorb heat radiation from land and ocean, and warm up the atmosphere. Measurements clearly show that both recently and historically, greenhouse gas concentration spikes are associated with spikes in temperature. This means that increasing anthropogenic emissions are warming up are planet continually. The Intergovernmental Panel on Climate Change (IPCC) projection on climate change suggest that the average world temperature will increase by more than 6°C by the end of this century (Pachauri & Reisinger 2007). While that might not seem like a big impact locally, a global temperature increase of 6°C would have tremendous repercussions on the environment, and respectively on our lifestyle and well being. Animals are also at risk to be affected by climate change. As their habitats become smaller and smaller as a result of climate instability, certain species face the risk of diminishing populations and even extinction. These are all alarming events, and the only way to prevent them is through global action and agreement. But, the first step towards making a change consists of informing as many individuals as possible. Given the technology and resources we have available nowadays, perhaps the best approach to educate individuals is to help them visualise the consequences we are likely to face because of climate change. This is the motivation that has inspired the development of this project.

In this paper, I outline the process I have undergone to develop a climate change visualisation system, from the early stages of research and literature review, to the system evaluation.

To build the system, I explored different procedural world generation techniques. In the end, I combined the prevalent technique of using coherent noise for terrain generation with my own model of temperature. I settled on this approach because coherent noise is a great tool to produce natural, organic textures and because this system has a particular focus on climate change, thus the temperature aspect was an essential contributor to the model. Using this approach, the environment is influenced not only by the noise, but by the temperature as well. Thus, when temperature changes, so does the environment. This is the essential functionality the entire system relies on.

After generating the environment, I discuss the process of introducing wildlife into the system. It is important to note that the environment is not the only aspect that is affected by climate change, and it was a valuable addition to reflect this in the system as well. Simple species representations are placed in the environment, in locations that they can thrive in. As circumstances in the environment change, the animals will be affected.

A timeline slider was decided on as the most suggestive and intuitive approach to show the transition the environment experiences as a result of increased greenhouse gas emissions. Thus, the user can control a slider and observe the gradual changes that take place in the environment. To clarify, the slider controls the passing of time over the generated environment. As time passes, greenhouse gas concentrations increase, to reflect what happens in the real world. As greenhouse gas increases, the greenhouse effect in the atmosphere is amplified: there is more greenhouse gas that captures heat radiation and warms the atmosphere. As a result, temperatures increase. Finally, as temperatures increase, because the model was designed with this in mind, the terrain adapts to the changing temperatures. As the terrain changes, it may become uninhabitable by the animals that populate it, so their population decreases.

The main factors that have shaped my project into what it is today have been the climate change literature I researched to expand my knowledge on the subject and other climate change visualisation systems I explored to gain an understanding of what kind of tools are available. Using these two resources as tools, I was able to identify what is missing from current existing systems, what my system needs to focus on, and how to correctly model the climate change effects. This is how I came to decide not to model these effects on a representation of the real world, but to use procedurally generated environments instead. The resulting tool is not prediction of the future, but a proof of concept based on parameters determined by our current circumstances. As parameters change, so will the outcome. If the global approach towards climate change does not change, these climate change effects might well occur sooner or later. However, if efforts are made to change our current emission patterns, we can prevent drastic events from taking place, or at least delay them until a better solution is discovered.

3 Background

3.1 Identifying demand

There is a lot of literature supporting the idea that visualising the effects of climate change can be a constructive way to raise awareness on the gravity of the situation we have put ourselves in. Many individuals process visual information better than written information.

Schroth et al. (2014) reinforce this statement through their research, which consisted of testing how an educational 3D game environment shifts individuals' perspective of climate change. Their findings confirm that interactive visual tools are effective in understanding climate change.

Nocke et al. (2008) show similar results after collecting data from questionnaires given to researchers that use visualisation techniques, tools and systems. They also underline the growing importance of presenting the data in a simplified, accessible manner for the general public.

O'Neill & Smith (2014) underline that indeed, climate change visualisations are becoming more widespread and a part of our daily discourse, but the way in which they are presented has the power to shape the politics and the worldview on climate change. This is why such tools must focus on providing a representation that is entirely based on research and must steer away from promoting unfounded material, speculations and being influenced by personal opinions.

Sheppard et al. (2008) suggest that while the urgency of climate change grows as an ever more popular subject among scientists, public awareness and drive to make changes are falling severely behind. A point that sets this paper apart from the previous ones is that the authors consider the ethical implications of using climate change visualisation as a tool for education and provoking action to mitigate climate change. They emphasise that, in doing so, we must not forget the uncertainties of such systems and the impossibility of modelling and predicting the future with unquestionable accuracy. The point of such systems is not to instigate fear, but rather to inform the user, using the knowledge we have so far, on the urgency of the matter and why action must be taken sooner rather then later. In a later paper with a specific focus on planning for climate change in a flood-prone community, Sheppard et al. (2010) agree with O'Neill & Smith (2014) that the means in which the information is presented is equally as important as raising awareness.

Throughout this literature, there is a consensus that there is a purpose and also demand for tools to visualise climate change, and that such tools must present factual, impartial information in an accessible manner.

3.2 Visualisations of Climate Change

Since it has been established that demand exists, the number of visualisations available comes as no surprise. Illustrations and graphics are widely popular as a mean to spread small but worrisome bits of information about climate change. These illustrations come in all shapes and sizes and refer to temperature, greenhouse gas, wildlife, sea levels, arctic ice and even food, and they portray what has already happened or predict what is yet to happen.

A popular wave of such illustrations came from a project by Ed Hawkins, *Warming stripes* (n.d.), which started with him publishing a series of visualisations for the increase in temperature over time in different regions, namely the United Kingdom, the United States of America and Canada. The images simply consisted of different coloured stripes, with each stripe representing a year, from as far back as 1772 to 2017, and its colour being determined by the average temperature of that year for the given location. While very simplistic, these visualisations were able to convey a strong message, reinforcing the evidence on climate change. This initiative led to the creation of the *#ShowYourStripes* (n.d.) tool, which follows the same concept. The tool allows you to select different regions and countries in the world and returns an image that shows the yearly increase in temperature for that location using the coloured stripes. Irrespective of the location you select, the result is undeniable - significant warming trends can be observed in recent years.

A vast collection of more such graphics and illustrations has been put together in *The Best Data Viz and Infographics on Climate Change Facts* (n.d.), providing condensed but meaningful educational material on most eminent concerns of climate change.

3.2.1 Interactive Systems

While static illustrations have the ability to transmit meaningful information, they are usually restricted in terms of amount of content, and they still rely on a significant amount of written content in order to communicate the intended information. This is what sets interactive systems apart, as they can allow the user to truly be immersed in the educational experience. The users learn through interacting with the systems without a need to rely on text content. They can observe changing visual elements as a result of their interaction with the system.

Climate Time Machine and Global Ice Viewer NASA's *Interactives – Climate Time Machine* (n.d.) is a tool that allows the user to visualise changes in temperature since 1884, increase of CO2 levels since 2002, changes in Arctic sea ice since 1979 and what a rise in sea level of up to 6 meters would look like, if the Greenland ice melted completely. The user is able to interact with the system and observe the progressive changes caused by climate instability. This tool has the ability to send a powerful message - the historical data clearly shows unprecedented circumstances occurring (more CO2, higher temperatures, less ice, increased sea levels) as a result of climate change. NASA also provides a second tool, *Global Ice Viewer* (n.d.), that focuses specifically on the effect of climate change on glaciers, sea ice and continental ice sheets. This tool provides more in-depth information than the *Interactives – Climate Time Machine* (n.d.) does on the subject of how worldwide ice has been affected by global warming. While the *Global Ice Viewer* (n.d.) still maintains a level of interactivity, it relies more on text content to achieve its goal of providing more thorough information. These tools rely on historical data rather than using predictions to send their message.

Climate Change Projections Another commendable interactive tool is *Climate Change Projections* (n.d.) an open source project by David Lemayian. This system takes a different approach at providing an understanding of increasing temperatures. The user can select any point on the world map and they will be able to see what other locations had that same temperature as the selected point in 2016, as well as what locations currently have the same temperature as the selected point will have in 2050 or 2070. It has an impactful effect, allowing users to see which locations are already very warm, and how they will become warmer over the years.

Climate Impact Lab *Climate Impact Lab* (n.d.) is another tool for observing changes in temperature. The tool is simple but effective, allowing the user to visualise how temperature has increased for any country on the world map from 1986 and how it is predicted to increase up until 2099.

Global Carbon Atlas *Global Carbon Atlas* (n.d.) is a tool that allows users to observe the changes in CO2 emissions from 1960 to 2018. The system offers a great amount of settings and types of visualisations that pack a lot of in-depth information. This is a powerful system, that can be used for raising awareness, but also for education and further research. The data is thoroughly explained and documented and can also be exported in a variety of formats for use in other projects.

Future Delta *Future Delta 2.0* (n.d.) is the interactive educational game that came as an outcome of the study conducted by Schroth et al. (2014). This system uses 3D visualisations and simulation tools to immerse the user and create an emotional response. The user observes the effects of climate change first-hand and must discover appropriate solutions to mitigate the predicted dangerous outcome. As previously mentioned, the results of the study confirm that such experiences are successful in raising awareness and concern on climate change matters.

Conclusion The common denominator for all these tools is that they represent scenarios for specific locations on the world map. That works perfectly when creating representations what has already happened using historical data, like *Interactives – Climate Time Machine* (n.d.), *Global Ice Viewer* (n.d.) and *Global Carbon Atlas* (n.d.) do. However, for systems that use predictions to educate users on only a subset of the effects of climate change, using the real world for the representation can prove counterproductive. While tools like *Climate Change Projections* (n.d.) and *Climate Impact*

Lab (n.d.) manage to obtain a powerful response right now, on the long term they might cause controversy. The fact is that creating a model that can accurately predict a process as complex as climate change is impossible. The literature clearly explains that climate change is not a one-size-fits-all event. Our world is an intricate system, and a tremendous amount of variables contribute to it. While researchers have proved that increasing the levels of greenhouse gas in our atmosphere does overall cause an increase in temperature, the precise outcome at any given location is still subject to the circumstances of that location. Lynas (2008) manages to explain this very clearly. In his first chapter, he talks about what we can expect if the average world temperature were to increase by 1°C. Extensive research supports that north western Europe could suffer from decreasing temperatures instead of increasing ones. A warm Atlantic current, the Atlantic Meridional Overturning Circulation (that includes the Gulf Stream), is responsible for warming the area, but if temperatures were to rise in the Norwegian and Greenland sea, this circulation could severely slow down, leaving northern Europe much colder than it is at the moment. The current has stopped in the past, after the end of the last ice age, bringing another wave of freezing cold and expanding glaciers. Back then, Norway was 7-9 °C colder than is it today. Another great example is that the Sahara Desert is predicted to turn green again, as a result of increased rainfall. However, *Climate Change Projections* (n.d.) and *Climate Impact Lab* (n.d.) both show north western Europe and the Sahara getting warmer in their visualisations, as they do all regions on the map. They are portraying a uniform temperature increase across the world, as a result of only considering the greenhouse gas and temperature aspects. This is a shortcoming of predictions: we can never reach perfect accuracy, because there is simply too much to consider. While these predictive tools are powerful at the moment, they might cause concern when they are unavoidably proven wrong, risking to even weaken trust in climate change. Instead, such systems should avoid making any promises of specific scenarios occurring in specific locations, and rather refocus their attention on creating an understanding of how severely the effects of climate change can disrupt the environment around us and consequently our lives.

These tools are readily available, but lots of other tools for modelling that are mentioned throughout pieces of literature are unavailable to the public, as their purpose is not to raise awareness, but to facilitate research. Therefore, visualising climate change is an idea that grows more popular, but it has not been over-saturated. We are still at a point where the tools available are scarce and restricted. Most tools focus on one specific aspect of climate change, or only use small amounts of historical data. This is a developing sub-field of climate change, and we can expect more complex, accessible systems for visualising climate change to emerge in the future.

3.3 Literature Review

To inform my project, I consulted multiple sources on the subject of climate change and world modelling.

3.3.1 Climate Change

Concerns around climate change have started emerging around the late 17th century, when Arrhenius (1896) suggested that greenhouse gases in the atmosphere influence the ground temperature. It was a long process of dealing with eager debate and doubt, but, ever since then, vast amounts of data have been collected to support the theory of climate change. Maslin (2014) underlines the importance of the principle of "weight of evidence", which implies that continuous data collection and research must be maintained in order to validate the theory of climate change. He kindly brings forth all this evidence, proving that it is indeed a heavily backed theory and there should be no room left for doubt. In the past, scientists believed that there were so many other natural factors that contributed to the resulting ground temperature that our small interferences were insignificant in comparison. This argument has since been invalidated by extensive gathered data that proves the opposite. This data comes in form of records of past and current temperatures and levels of CO2, that spike up ever since the beginning of industrialisation.

The discovery of the influence of greenhouse gases on temperature began as a by-product of a different research project, that was investigating how a decrease in atmospheric CO2 was a major cause for the ice ages. That set in motion what would become extensive research on both current and historical CO2 atmospheric concentrations. A clear testament to the impact of CO2 concentration on temperature is the study of past climate. Scientists have mapped the CO2 concentration and temperature over the past 800,000 years. This was possible through various sources, like marine sediments, ice cores (drilling to extract bubbles of air trapped inside them) from Antarctica, or tree rings. In fact, the records on climate go as far back as 50 million years. The evidence clearly shows a connection between CO2 concentration and temperature, as spikes and dips line up in the two graphs.

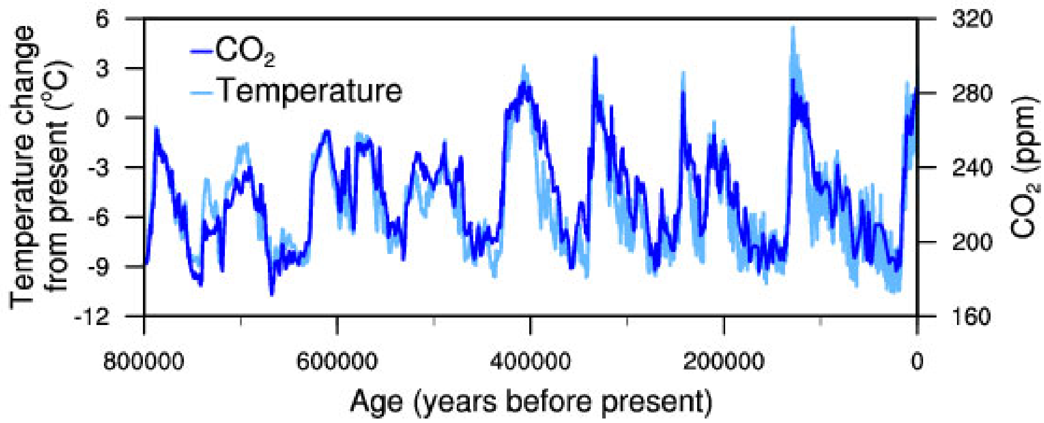


Figure 1*: *Temperature (light blue) and CO2 (dark blue) variations measured from ice cores (Jouzel et al. 2007†; Lüthi et al. 2008‡)*

In 1958, Keeling et al. (1976) first began keeping records of CO2 concentration in the atmosphere, by taking measurements in Antarctica and Mauna Loa, Hawaii using the latest modern technology. Measurements are still taken, and the graph curve that shows a consistent increase in CO2 concentrations over the years has become a popular icon in the climate change discourse.

Maslin (2014) manages to paint an even clearer picture of climate change by citing the long-term effects. Historically, our planet has been known to follow a cyclic pattern of glacial-interglacial climate, owing to changes in Earth's orbit around the sun. That means that it is possible to determine when the next ice age should occur, would there be no human interference involved. If that were the case, the next glacial period should begin sometime in the next 1000 years. However, a lot of the harm has already been done, and according to Archer & Ganopolski (2005), our meddling in the environment has already delayed this next ice age by 40,000 years. Even if we did our utter best to minimise our future emissions, the glacial period would still be delayed by 120,000. Furthermore, continuing CO2 emissions at our current rate, the ice age could be delayed by more than 500,000 years. That is not to say that an ice age would benefit our lifestyle, but purely to show how forceful our interference in the environment truly is.

According to the most recent reports by NOAA (National Oceanic and Atmospheric Administration), *Global Climate Report - Annual 2017* (n.d.) and *Assessing the Global Climate in 2019* (2020), not only has 2019 been the second warmest year since the beginning of record keeping in 1880, but all the past six year have been the six warmest years on record. That can hardly be classified as coincidence.

Lynas (2008) made use of the numerous papers and research available to put together a book about climate change, but with a different spin on it. Instead of writing a purely educational book about climate change, he had a different vision, creating what feels a lot like a dystopian novel, only it is about the real world. He paints the picture of what our world will look like as it warms up, degree by degree, up to 6°C, over the remainder of this century. This is based on the IPCC's projections that by the end of 2100, the temperature will increase by as low as 1.1°C if we severely reduce our emissions, or as much as 6.4°C in the highest emission scenario (Pachauri & Reisinger (2007)). These projections and Lynas's book have been pillars of this project, having a great influence on the design choices and decision making. Lynas focuses on describing the different events that are predicted to take place in different regions, constantly explaining historical data and quoting literature on these

*https://www.ncdc.noaa.gov/global-warming/temperature-change
†EPICA Dome C - 800KYr Deuterium Data and Temperature Estimates
‡EPICA Dome C - 800KYr CO2 Data

predictions. He succeeds in emphasising that temperature increase is not the only outcome we are bound to see from climate change, as previously mentioned about north-western Europe and the Sahara Desert.

3.3.2 Climate Change and Wildlife

Another emerging area of research in the climate change sector is how wildlife is affected by it. Many species have been severely affected or even met their end as a result of human interference in the environment (Braje & Erlandson (2013)), and many more are expected to meet the same fate due to anthropogenic meddling (Root et al. (2003)).

Radchuk et al. (2019) have investigated the ability of species to adapt to climate change. Their findings show that climate change has not affected morphological traits, meaning the appearance of species, but it has affected phenological* traits in some species, meaning their life cycle events that are influenced by climate. However, these adaptive behaviours are not as effective as the species normal biological responses that have been developed over millions of years. Using a theoretical model, Radchuk et al. (2019) concluded that failing to adapt sufficiently may have already started affecting the persistence of species.

Thomas et al. (2004) explored the risk of extinctions in regions that cover 20% of the terrestrial surface of the planet. Their projections show that 18% to 35% (depending on the severity of the climate scenario) of species in their sample regions will become extinct by 2050.

Johnston & Schmitz (1997) used simulations to observe how doubling the atmospheric CO_2 would impact 4 mammal species. Their findings show that the increase in temperature would have little to no effect on the animals, as they can withstand the heat. However, the fluctuations in vegetation as a result of rising temperatures can cause species to relocate and shrink. While temperature itself might not affect wildlife, changes in habitat and vegetation certainly will.

Swim & Bloodhart (2015) compared using an objective approach and an emphatic one on environmental campaigns about the harm caused to Polar Bear by climate change. Contrary to common warnings, their findings show that the objective approach offers no added benefit, and the emotional approach does not receive negative responses. Instead it motivates individuals, both environmentalists and non-environmentalist, to be more invested in and donate to environmental causes.

Giannini et al. (2012), Rasmont et al. (2015) and Le Conte & Navajas (2008) all investigate how bees (Brazilian bees, bumblebees and honeybees) are affected by climate change. They agree that the effects are already visible and will continue to intensify, through reduction of bee habitats, fluctuations in flower development and pollen and nectar production, and changes in bee psychology and behaviour.

Understanding the vulnerabilities of wildlife can help us comprehend how harmful anthropogenic emissions can be, and how this information can be used to educate individuals on climate change. While temperature increase may not directly affect us and animals, the events set in motion by temperature increase will. Further, it was established that portraying the harm that wildlife is subject to is effective in raising awareness and concern on climate change matters.

3.3.3 Environment Modelling through Procedural Generation

In order to show the effects of climate change, an environment is required to reflect these effects upon. For this purpose, there are two options: first, to use the real world for representation, or, second, to generate random worlds. As previously discussed, the real-world representation is the perfect approach when using data to reflect events or effects that already took place, but making an accurate prediction model that fits every region on Earth is impossible. Since the aim is not to predict the future, but educate individuals on climate change with the knowledge we have so far, we can do so using a general representation that can fit and adapt to any environment. That is why the idea of world generation is explored within this project. World generation is a popular area of software development, widely employed in simulation and game development. Using gradient noise for procedural terrain generation is a well-established technique.

Perlin (1985), (2002) studied the idea of developing naturalistic looking textures using stochastic functions. This is how Perlin noise emerged, and it is still used for that same purpose today. Perlin

*Phenology is a branch of science dealing with the relations between climate and periodic biological phenomena (such as bird migration or plant flowering) noise is a gradient or coherent type of noise. As opposed to pure noise, it has a logic to it. "Coherent noise means that for any two points in the space, the value changes smoothly as you move from one point to the other – that is, there are no discontinuities" - Matt Zucker[*]. Let us consider an image of noise, with each pixel a colour between black and white, determined by a noise value. With pure random noise, two neighbouring pixels have no connection to each other and can have completely unrelated values, for instance a black and a white pixel can be neighbours. With Perlin or any type of coherent noise, two neighbouring pixels will always maintain a clear relationship - the difference between their two noise values does not surpass a certain threshold, and a group of pixels will create a gradient. For this reason, coherent noise can be used to create natural, organic textures, including terrain maps. After Perlin noise, lots of other gradient noise variants appeared, including Perlin's (2001) Simplex noise, an algorithm that aimed to address the limitation of his previous work with Perlin noise. Archer compares a few different types of noise, including Perlin, Simplex and Value noise, highlighting their quality and performance and how Simplex noise attempts to fix the shortcomings of Perlin noise.

The first step in procedurally generating an environment is to create a noise map. A noise map can be created by sampling the preferred noise type function for each pixel. What that means is calculating a pixel sample and applying the noise function on that sample to generate the noise value for that pixel. Different noise functions will produce different noise maps given the same samples.

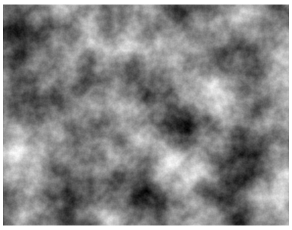


Figure 2: Perlin Noise

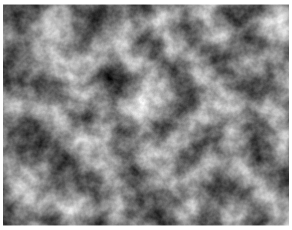


Figure 3: Simplex Noise

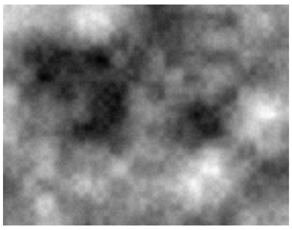


Figure 4: Value Noise

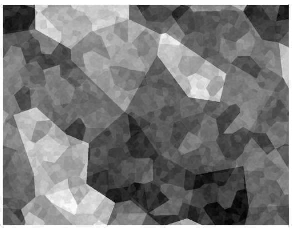


Figure 5: Cellular Noise

Figures 2 through 5 depict noise maps that were produced using the same samples on four different noise types: from left to right, Perlin, Simplex, Value and Cellular noise.

Sampling the noise Sampling the noise is a way of "shaping" the noise to meet your needs. To sample the noise, we need to set our preferences for the noise, namely:

- the map width and height, which determine the size of the environment we are modelling.
- the map seed, which acts like a unique identifier for the map. It is used for a pseudo-random number generator, which given the same seed, will always generate the same sequence of random values. Thus, using a certain seed will always create the same map.
- the noise scale, that determines how large or small the features in the noise will be. See Figures 6 through 9 of the same Perlin noise map with different scale noise.

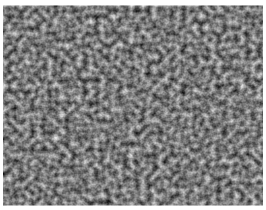


Figure 6: Scale 0.3

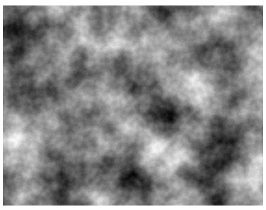


Figure 7: Scale 2.5

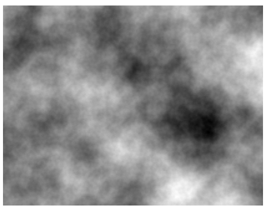


Figure 8: Scale 5


Figure 9: Scale 10

- the noise offset, which essentially allows us to move horizontally and vertically on the map. The width and the height of the map are merely bounds that determine how much of the map we see at once, but in theory a noise map can be infinitely explored using the x-axis and y-axis offset. Computationally, we are limited by number representations.

[*] The Perlin noise math FAQ by Matt Zucker

- the noise octaves, which determine the level of detail in the noise. The more octaves there are, the more detail there will be, but as each octave adds smaller details influence of those details over the overall noise shape decreases.

- the noise lacunarity, which determines how many more increasingly smaller details each octave will add.

- the noise persistence, which determines how much the details at each octave influence the overall shape of the map.

These parameters are used to create the noise map. To elaborate on how the map is shaped and the three factors that determine the shape of the noise, we will take a closer look at octaves, lacunarity and persistence.

As Lague explains it, a slice of coherent noise resembles a section of mountainous terrain. Amplitude and frequency are two features of noise. In a noise section, amplitude refers to the y-axis and frequency refers to the x-axis

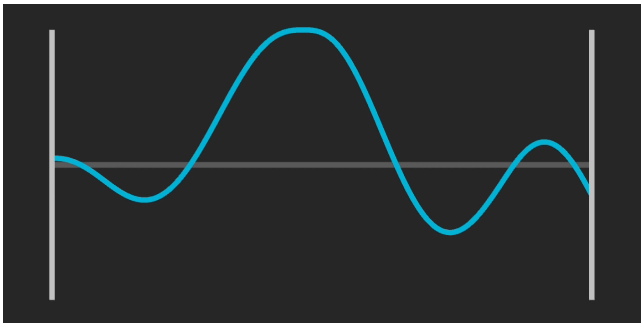


Figure 10: A coherent noise section*

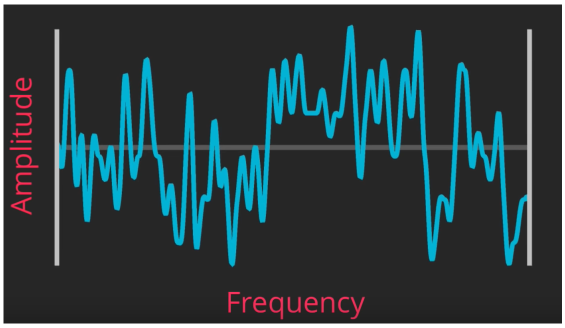


Figure 11: Amplitude and frequency*

However, this is section is too smooth to represent natural terrain. More detail needs to be added in, while still preserving the overall shape. To do this, we layer multiple levels of noise. Each layer is an **octave**. To control how each subsequent octave increases in detail (frequency), we use lacunarity. The frequency of each octave will be that of the lacunarity to the power of the octave index, $frequency = lacunarity^{octave}$. Lacunarity should have a value above 1, so that frequency increases with each octave. For instance, given a lacunarity of 2, the frequency of the first octave will be $frequency = lacunarity^0 = 2^0 = 1$, while the frequency of the second octave will be $frequency = 2^1 = 2$ and the third $frequency = 2^2 = 4$. Layering these octaves will produce a more detailed noise section, as depicted in Figure 12 in the red graph.

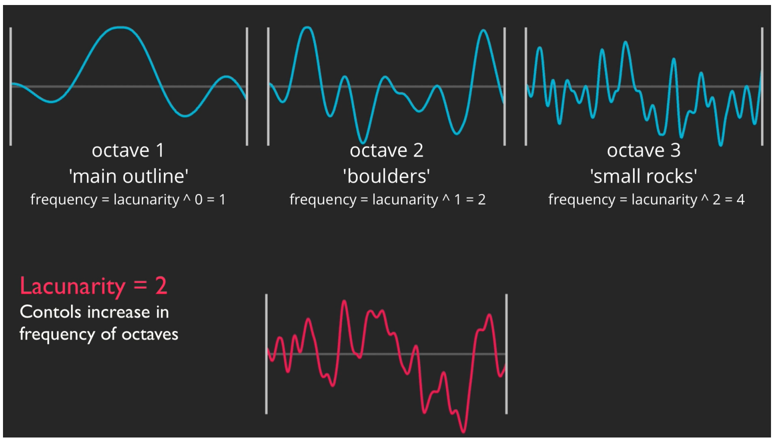


Figure 12: Frequency and lacunarity*

However, the section is still not passable as natural terrain, because it looks too rough. This is why we introduce persistence, which we use to control the amplitude of each octave. We want the amplitude to decrease with each octave, because as smaller detail (frequency) is included, it should influence the overall shape of the noise map less and less. Thus, amplitude will be a function of persistence,

*Sebatian Lague - Procedural Terrain Generation

$amplitude = persistence^{octave}$. To ensure the amplitude decreases with octaves, persistence must have a value between 0 and 1. For instance, given a persistence of 0.5, the amplitude of the first octave will be $amplitude = persistence^0 = 0.5^0 = 1$, while the frequency of the second octave will be $amplitude = 0.5^1 = 0.5$ and the third $amplitude = persistence^2 = 0.5^2 = 0.25$. Layering the octaves will create a more detailed noise section, but with smaller detail having less influence overall, as depicted in Figure 13.

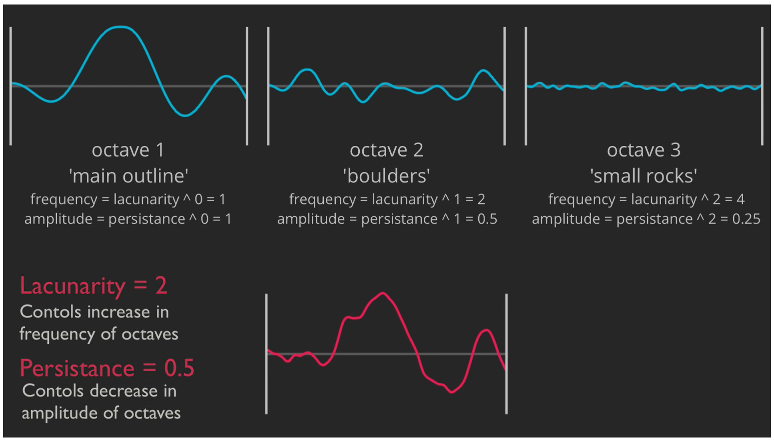


Figure 13: Amplitude and persistence*

This is the procedure used to generate natural-looking terrain maps. Calculating the frequency and amplitude of each octave is part of the sampling process, and "shaping" the coherent noise to the format you need. Figures 14 and 15 emphasise the difference between a noise map with one layer of noise (1 octave) and a noise map with 5 layers of noise, that implement the frequency and amplitude procedure. The same large features can be observed in both maps, but the layered noise one (Figure 15) has a more detailed, organic look.

Figure 14: Map with 1 octave

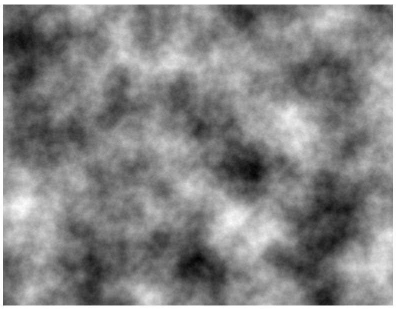


Figure 15: Map with 5 octaves

This is the first step in generating random, unique environments.

3.3.4 Modelling Temperature

To be able to represent climate change, a system must model temperature and how temperature impacts the environment. However, here is not one definitive function to calculate temperature at any given point in the world. Instead, temperature is a product of countless contributors, which makes it a difficult aspect to model. When modelling a system from the real world, an essential step is to scale down the problem. Instead of building a very complex system, one must consider which factors of that system are elementary for the problem they are trying to solve, and which are negligible. In this case, we know that temperature is influenced by altitude, pressure, latitude, local climate, precipitation, ocean currents, greenhouse gas and much more. Of these, pressure is non-essential for the purpose of simulating climate change, and it is not required for any other aspect of the system either. Local climate, precipitation and ocean currents are all complex systems to model and would not necessarily offer great substance to the purpose we want to achieve. However, altitude and latitude are features of any environment, real or modelled. For instance, given the first described step of generating a noise map for our unique random world, we already have access to both of these

*Sebatian Lague - Procedural Terrain Generation features. The altitude, that can also be referred to as the terrain height, at each pixel is given by the noise value - for instance, a white pixel can be the lowest point on the map and a black one the height point on the map, or vice-versa. The latitude of each pixel can be easily computed given the map height and the y coordinate of the pixel. Thus, altitude and latitude are elements that can be implemented with ease and used further for the purpose of modelling temperature. Apart from these, greenhouse gas is inherently relevant to the topic of climate change and needs representation, especially considering that the system needs not only to model temperature, but climate change as well. Greenhouse gas will be the "vessel" for representing climate change.

Latitude and Temperature In short, latitude is a geographic coordinate which represents how far a point on Earth is from the equator. Latitude is one of the main factors that influence temperature, because of the oblate spheroid shape of the Earth and the angle that the sun rays hit the surface at. As points move further away from the equator, they receive less sunlight and radiation, resulting in lower temperatures. (Sawe)

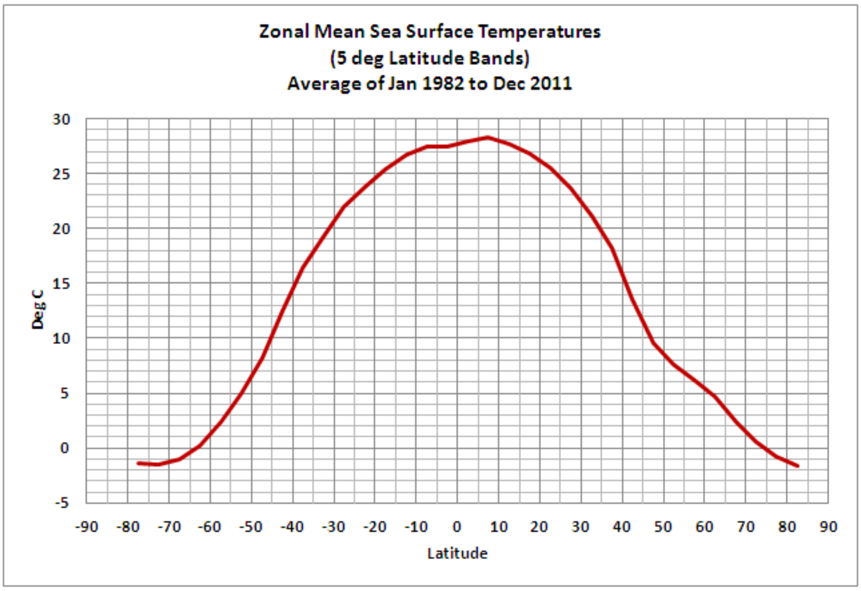


Figure 16: Latitude and Surface Temperature*

Figure 16 shows the relationship between latitude and sea surface temperature plotted as a graph. Given the graph curve, and our knowledge about temperatures, a function can be determined that calculates the sea surface temperature given the latitude. A cosine function creates a similar curve, but it needs slight adjustments to return temperatures in the range of 5°C to 20°C. Given that latitude values are in the range -90°and 90°, the function that meets these requirements for this range of latitude inputs is $7.5 \times \cos\left(\frac{x}{28}\right) + 12.5$. The resulting graph in Figure 17 is comparable to the one plotted using real-world data in Figure 16.

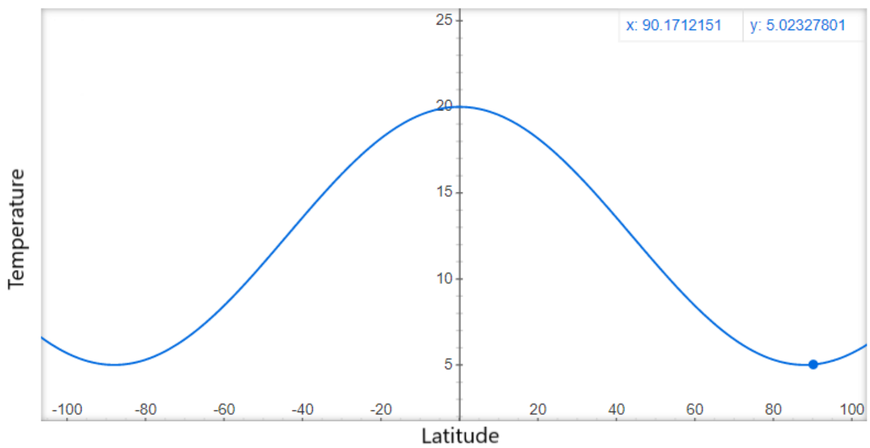


Figure 17: $7.5 \times \cos\left(\frac{x}{28}\right) + 12.5$

*Sea Surface Temperature Trends As A Function Of Latitude Bands By Pielke & Tisdale

Having a method to calculate the sea surface temperature given the latitude is essential for the next step in modelling temperature

Altitude and Temperature According to NASA's Earth Atmosphere Model Paper, the temperature at any altitude within the troposphere can be easily computed given the sea surface temperature. The troposphere is the lower zone of the atmosphere, that expands from Earth's surface to 11,000 meters. Within the troposphere, the temperature decreases linearly, approximately 6.5°C per 1000 meters. Given the surface temperature, which we established can be computed using the point latitude, and the altitude of a point, the temperature can be calculated using the formula $T = S - 0.00649 \times h$, where S is the sea surface temperature and h is the height.

Greenhouse Gas and Temperature Gases like carbon dioxide, methane and nitrous oxide which are present in our atmosphere are referred to as greenhouse gases because they cause a greenhouse effect. The land and the ocean absorb solar energy which warms them. As a result, they emit this warmth through long-wave infrared radiation ("heat" radiation). The greenhouse effect is the process through which the gases in our atmosphere partially absorb this radiation, warming the atmosphere (Maslin 2014, Planton 2013). However, the greenhouse effect is necessary, as temperatures would be much lower without it, and the average temperature in tropical regions would approach -10°C (Maslin 2014). Increasing greenhouse gas levels as a result of anthropogenic emissions cause an enhanced greenhouse effect, warming the temperature even further than normally. Thus, considering increased emissions of greenhouse gas over time, to accurately model temperature and climate change, the greenhouse gas induced temperature deviation must be taken into account.

3.4 Other Research in this Domain

As the topic of climate change becomes prevalent, more research emerges on niche topics related to it, providing insight into the intricacies of the subject that do not immediately come to mind.

A common theme among research surrounding simulations on climate change is analysing or predicting outcomes in particular regions, or focusing on a smaller problem set, restricted in some regard. This strategy can clearly prove helpful for particular affected regions that are in need of alarming, immediate planning to prevent future dangerous extreme events. It proves more difficult to find research that focuses on the broad picture, or projects that have the particular aim to inform the masses, rather than add to the collection of literature backing the theory of Climate Change.

Maslin (2014) indicates that climate change is not only expected to have environmental effects, but economic as well. According to Stern et al. (2007), the cost of climate change could go as far up as 20% of the world GDP (gross domestic product). However, the estimated cost, if action is taken as soon as possible, is much more bearable, around 2% of the world GDP.

If the environmental effects were insufficient to convey the message of urgency, these further snippets of life in our future with climate change should manage to drive the point home.

"If you think education is expensive, try estimating the cost of ignorance."
Howard Gardner

4 Designing the System

Considering all the information gathered so far, a clearer idea emerges of what kind of system needs to be built. To reiterate, the purpose of this project is to fill a gap in the field of climate change visualisation.

As it stands right now, there is a considerable lack of systems that represent the impact of climate change on wildlife. Thus, the aim of this project is to build a system that informs the user on the future consequences of climate change on the environment and wildlife concurrently, through the use of visualisations.

Further, as iterated, employing a emphatic approach by highlighting the harm that species suffer from climate change has proven successful in informing and motivating individuals to take action against climate change. Thus, this system will highlight the harm that animals suffer in the attempt to raise awareness.

Apart from the above, all available informative or educational interactive tools on climate change use real-world representations to make general predictions that only consider a small subset of climate change effects and scenarios, risking to inaccurately represent the consequences of climate change. This system will not be limited by the restrictive nature of modelling predictions on the real world. Instead, it will focus on conveying the effects of climate change in a general manner, on any given environment. Thus, instead of using a predefined map of the real world, random unique worlds will be generated and used as a vessel for representing climate change. This will not only remove the risk of making inaccurate predictions, but it will also render the system much more versatile. A straightforward system, that can generate unique environments with animals, temperature and elementary climate change effects, has great extensibility potential. By making the system open source, developers will be able to use as a pillar for personal or bigger projects to build upon. Thus, apart from the use case of educating individuals on climate change and raising awareness, the system will extend its audience towards developers and researchers.

The principle approach to achieving such a system is to simplify the problem as necessary, whilst conveying information as truthfully as possible. As previously iterated, it is impossible to accurately model real-world systems and their complexity, thus compromises and abstractions must be made. It is also impossible to predict the future, but efforts will be made to represents the knowledge we have so far on the future of climate change as precisely as possible. "Simplicity and precision" will resound as the motto of this project.

4.1 Requirements

Considering the detailed description above, a list of system requirements can be compiled. Broadly, the tool must be accessible, easy to navigate and offer an impartial portrayal of how climate change can impact any environment over a given period of time.

4.1.1 Functional Requirements

1. The system should provide the user with a visualisation of an environment.
   (a) The system should allow the user to interact with the environment.
   (b) The system should prompt the user with additional information about the segment of the environment they interacted with.
   (c) The system should allow the user to see the terrain map of the environment.
   (d) The system should allow the user to see the temperature map for the environment.
   (e) The system should allow the user to see the greenhouse gas map for the environment.
   (f) The system should allow the user to see a height map for the environment.
   (g) The system should allow the user to zoom in on a section of the map.
2. The system should provide the user with a menu interface.
   (a) The system should allow the user to change the map settings, including the type of noise, scale, offset, number of octaves, lacunarity and persistence.

   (b) The system should allow the user to create new, different environments.

3. The system should allow the user to add wildlife to any environment.
   (a) The system should allow the user to interact with the wildlife.
   (b) The system should provide the user with additional information about the animals they interact with.
   (c) The system should allow the user to display or hide the animals on the map.

4. The system should allow the user to manipulate time.
   (a) The system should update the state of the environment when the year is changed.
   (b) The system should update the state of the animals when the year is changed.
   (c) The system should prompt the user with additional information on the changes that occurred as a result of the year being updated.

4.1.2 Non-functional Requirements

1. Performance
   (a) The system should create a new map within 3 seconds of the user request.
   (b) The system should generate animals on a map within 3 seconds of the user request.

2. Reliability
   (a) The system should perform the same tasks in the same way on any machine, given the same parameters.
   (b) The system should not become unresponsive as a result of the settings a user sets.
   (c) The system should display unhandled exceptions in the console but cause no disruption to the user interface.

3. Accessibility
   (a) The system should be an application runnable on any machine with Java 8 installed.
   (b) The system should be open source and available to any person.

4. Modifiability
   (a) The system must be easily modifiable and extensible by one person.

5. Usability
   (a) The system must be easy to use and navigate.

4.2 Loose Coupling

A major goal for this system is to ensure its extensibility and versatility, such that it can be used not only as a learning tool, but as a building block for other similar projects as well. To achieve this, it is essential to minimise dependencies as much as possible and have a clear separation between components.

4.3 Sequence Diagram

The system has clear flow, that is best represented through a sequence diagram (see Figure 18) that describes the process of a user interacting with the system to observe the changes caused by climate change over time on an environment with animals.

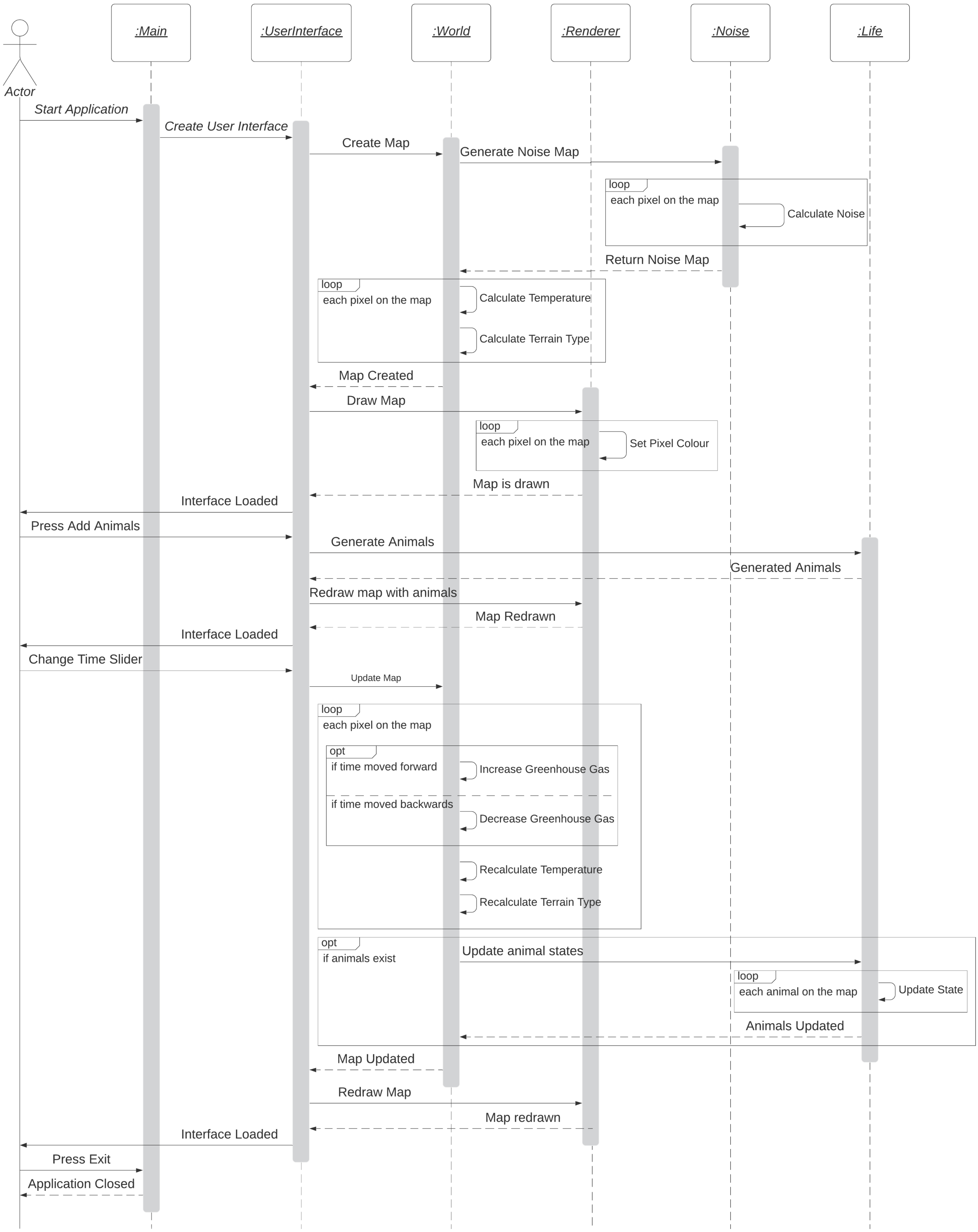


Figure 18: Flow of Events

5 Implementation

To achieve the design pattern detailed above, the system has been implemented using Java 8 and the integrated JavaFX library.

The system follows a clear, linear flow, where each step is reliant on the previous one. However, to ensure a smooth experience for the user, some operations are separated on different threads. Thus, the most intuitive approach to describe the system is to detail this flow.

5.1 Creating the User Interface

As the application starts, the application window is built. This process happens in the `UserInterface` class, which is one of the two classes that handle the graphical operations of the application. The entire graphical user interface is built using JavaFX components. Inside `UserInterface`, the `buildWindow()` method creates the interface, which consists of two main components: the menu, on the left hand-side, and the map view, on the right hand-side. The menu is further split into two sub-components: the user controls, and the information box.

5.1.1 The User Controls

The user controls represent the application settings (see Appendix A.1 and A.2). They allow the user to create new maps or change the existing one. The settings that are immediately available are those essential to the user (Appendix A.1):

- The `Auto Update` checkbox, which is selected by default and controls if the map automatically updates. This means that any changes to the settings will propagate to the map immediately, without having to press the Generate Map button.

- The seed, which acts as a map identifier and allows the user to change between maps. They can either input a specific seed, or generate a random one using the `Random` button, placed next to the seed input field.

- The `Map Type` drop-down list, which allows the user to switch between the height, temperature, terrain and greenhouse gas maps, which are available for all generated worlds.

- The `Generate Animals` button, which allows the user to add animals to any map. The decision to not include them by default stems from the wish to provide the user the flexibility to use the system in any way they prefer. There might be occasions when a user would prefer to exclusively focus on the changes in the environment, without regards to wildlife. The wildlife "layer" requires additional resources and time for the computations, that are simply redundant in situations where wildlife is not the object of focus. Thus, animals can be added in when needed. Even once they are generated, the user has the ability to hide them (through a checkbox that replaces the `Generate Animals` button once the button is pressed), if they want to focus their attention on something different.

- Possibly the most relevant menu option is the `Timeline` slider. This is how the system achieves its inherent purpose, of allowing the user to see the effects of climate change on the generated environments. By moving the slider, the user will be able to observe as the system updates the map to reflect the changes that take place in the environment as a result of the time change. The user can move forward or backward, between the years 2000 to 2100. This year range was decided on based on the IPCC's projections (Pachauri & Reisinger 2007) and the research compiled in Lynas' book, that predict that by the end of this century, the average world temperature will go up by as much as 6°Celsius. These are the projections the system aims to reflect (temperature increase from the year 2000 to 2100 in the range of 5 to 6°C), and these are the results that the system should return for maps with the default noise settings (noise type, scale, octaves, persistence, lacunarity).

- The `Map Setting` button, which toggles the advanced settings menu (see Appendix A.2). The advanced settings are non-essential to regular users that only aim to use the system as an educational tool for climate change. These settings allow the user to toggle the equator line on the map, or to manipulate the type, scale, offset, octaves, lacunarity and persistence of the noise. By default, they are set to the appropriate values that produce maps that resemble the real world. Thus, regular users are not required to interact with them for the educational experience. They are however very useful for development and for users that intend to use the system for purposes that surpass the initial system requirements.

- The `Generate Map (clean)` button, which generates a map with the current settings from the menu (seed, map type, noise type, scale, offset, octaves, lacunarity, persistence). There are two scenarios where a user would want to use this button: first, if they have the `Auto Update` checkbox unchecked, so they have to manually generate the map once they set their preferred settings, or second, if they want to refresh their current map to a "clean" state. The "clean" adjective refers to the fact that pressing this button will generate a map in its initial state from the year 2000 and without animals. Thus, the user can refresh the map to its initial state after experimenting with the `Timeline` slider and the animals.

5.1.2 The Information Box

The information box is a text box where the majority of textual information is presented to the user. Whenever a new map is created, basic details about the map are displayed within the information box. These are details that that are not immediately discernible simply by visualising the map, for instance the average world temperature and the world percentage of water and ice. Similarly, when animals are added to a map, the text box informs the user how many animals were generated. When a time change occurs, the box updates the user on the changes that took place: the temperature change, the water and ice percentages change, how many animals have changed state, what percentage of each species was lost since 2000. Finally, when the user interacts with any location on the map, details about that location are displayed in the information box. This component exists to provide the user with relevant statistics that might pique their interest and solidify the message sent by the visualisation.

5.1.3 The Map View

The map view represents the container where the maps are displayed. A particular focus for the map container was to ensure that users would be able to zoom and drag the map, in order to offer as much flexibility as possible in terms of interaction. For this reason, the JavaFX `ScrollPane` components was extended and modified to include the zooming and dragging functionality, using the approach described by Hári.

On top of the map display, a dialog overlay appears when the timeline slider is adjusted. The dialog presents the user with the most relevant information about the changes that took place, including the temperature change and animal state changes. The dialog can be dismissed by pressing the X button or by clicking anywhere on the map, outside the dialog itself.

5.2 Creating the Map

The `createMap()` method from `UserInterface` deals with setting the system in motion to generate a new map, and once it receives that information, it draws it inside the map container of the interface.

This method is called upon when the user presses the `Generate Map (clean)` button, or when they change the certain map settings (seed, noise type, scale, offset, octaves, persistence, lacunarity) if they have `Auto Update` on. However, the method is first called inside `buildWindow()`, once all the interface components are built (the menu, information box and map view), in order to generate the default map, with the default settings, that the users see when they first run the application.

`createMap()` gathers the settings from the menu and sends these to the `WorldThread`, which is a worker thread that deals with the calculations necessary to create the map. The `run()` method of `WorldThread` informs the JavaFX Interface thread to place the interface in the loading state, which consists of a loading animation overlay on the map container and disabling the menu settings, to avoid starting separate threads that depend on information that has not yet been computed or threads that overwrite each other's results because they are both creating two different maps at once.

While the interface is loading, the `WorldThread` calls the static method `calculateMap()` inside the `World` class. The `World` class deals with all the major computation relating to the environments. The `World` class has a collection of all the pixels on the map, the 2D-array `pixels` of type `Pixel[mapWidth][mapHeight]`. A `Pixel` object consists of all the information necessary for each pixel: latitude, terrain height, greenhouse gas, temperature and terrain type. The `calculateMap()` method gradually populates the `pixels` matrix, which is then used to draw the map. The very first step is to create a clean "canvas" for the map: set the initial sea level, set the year to 2000, set the water and ice percentages to 0, set the average temperature to 0 and clear the previous map if there was one. The next step is to lay the base for the environment, which, as previously discussed, is the noise map.

5.2.1 Generating the Noise map

As detailed in section 3.3.3 on Procedural World Generation, coherent noise is often used to create natural graphical textures, and especially terrain. In this case, the produced noise values are used as terrain height values to create a natural relief effect. The `calculateMap()` method calls the static method `generateNoiseMap()` of the `Noise` class twice: once to generate a height map, and once to generate a greenhouse gas concentration map. Since this system aims to model the effects of greenhouse gas increase in the atmosphere and not the factors that cause the increase in the first place, there was no source to extrapolate the greenhouse gas levels from. Thus, the decision has been made to use a second layer of coherent noise for the initial greenhouse gas concentration, such that the greenhouse gas distributions are not tied specifically to the terrain height. Thus, the `generateNoiseMap()` method is used to generate a terrain height map and a greenhouse gas concentration map.

The `generateNoiseMap()` method takes the map width and height and the noise seed, scale, offset, octaves, lacunarity and persistence. In order to ensure the application is as versatile as possible, it was decided to not limit the system to one particular type of coherent noise. For this reason, the system uses the FastNoise library by Peck (2020), which offers a variety of coherent noise functions. To use the library, we create an instance of the `FastNoise` class and set the preferred noise type using the method `SetNoiseType()`. The noise type preference is taken from the `Map Settings` menu. The different noise types available are: Perlin, Simplex, Value, Cubic, and their fractal equivalents. After setting the noise type, iteration over the map width and height begins, to simulate each pixel. For each pixel, the initial amplitude and frequency are set to 1. We then iterate over each octave, to add all the levels of detail to the pixel. For each octave, we calculate $sampleX = \frac{x - mapWidth/2}{scale \times frequency} + offsetX$ and $sampleY = \frac{y - mapHeight/2}{scale \times frequency} + offsetY$, which dictate the coordinates, scale, offset, amplitude and frequency the noise should have for this octave. We then call the `FastNoise GetNoise(sampleX, sampleY)` method, which calculates the noise for this octave based on the noise type that was set previously. It simply applies the specific functions used to calculate that type of noise. We add the generated noise value to our pixel `noiseHeight` and proceed to multiply the amplitude by the persistence, $amplitude = amplitude \times persistence$, to decrease the influence of the next octave over the overall shape of the noise, and the frequency by the lacunarity, $frequency = frequency \times lacunarity$, to increase the frequency of small details for the next octave. We repeat the same process for all octaves, for each individual pixel. Finally, we scale the noise values to ensure they are all in the range 0 to 1.

`calculateMap()` then calls `generateNoiseMap()` again, to create the greenhouse gas concentration map the same way the height map was created.

Thus, the random noise maps that make each world unique are created, and we can move forward to calculating the remaining details based on them. In `calculateMap()`, a nested for loop iterates over the map width and height, to signify generating each pixel. At each step, the pixel is created, and its noise height and greenhouse gas concentration are set from the two maps generated before. Then, the temperature of that pixel is calculated.

5.2.2 Calculating the Temperature

The `calculateMap()` method calls the `setTemperature()` method, still inside the `World` class, which takes the pixel coordinates as parameters. What follows is the three-step temperature calculation, as detailed in section 3.3.4 on Temperature Modelling.

The Sea Surface Temperature The pixel latitude is calculated from the y coordinate and the map height, using the formula $latitude = -\frac{y - \frac{mapHeight}{2}}{\frac{mapHeight}{2\times 90}}$, so that latitude is 90°and -90°at the poles and 0°at the map centre. Given the latitude, the sea surface temperature can be calculated for any pixel on the map. The sea surface temperature is calculated using the formula $7.5 \times \cos\left(\frac{latitude}{28}\right) + 12.5$ as describe is section 3.3.4, such that temperatures are warmest near the equator line and decrease gradually towards the poles.

The Height Temperature Next, the pixel noise height must be converted to a height in meters, so the temperature at any height on the map can be calculated using the formula describe by NASA, which requires the sea surface temperature and terrain height. To obtain the terrain height, the noise values are scaled from the range (0,1) to the range (-2500,2500). While the Earth certainly has terrain height outside that range, given the nature of coherent noise, the distribution of points of each height is somewhat equal, whereas in the real world, very few location have extreme heights, and the majority of the rest of the surface lies around similar, lower heights. Thus, setting the upper terrain height bound to 8850 meters, as that is the highest point on Earth, would have resulted in lots of locations with that height, and heights of 7000 to 3000 meters (which are rare occurrences in the real world), and much fewer locations with heights close to our global average altitude, which is of 800 meters. Thus, the compromise has been made to ensure a model overall more similar to Earth in terms of terrain height, but lacking the very high altitudes. Returning to the problem at hand, once the terrain height in meters is determined and stored in the pixel structure, the height temperature can be calculated with the formula $T = seaSurfaceTemperature - (0.00649 \times height)$. The formula is based around the assumption that within the troposphere, the temperature decreases by approximately 6.5°C for every 1000 meters in altitude, from the starting point which is the sea surface temperature at 0 meters altitude, which we have calculated previously. The height temperature is stored in the pixel, for use in future re-calculations of temperature after changes occur in greenhouse gas concentrations.

The Greenhouse Gas Temperature The final step in the temperature calculation is to consider the greenhouse gas concentration. Based on our knowledge that greenhouse gas absorbs "heat" radiation from the terrain and ocean which amplifies the temperature in the atmosphere, we can use the greenhouse gas concentration obtained through the second layer of coherent noise to deviate the height temperature. Thus, we obtain the final temperature by adding the greenhouse gas factor the height temperature. This has been modelled to reflect the projections of 6°C of temperature increase over 100 years. How greenhouse gas increases will be described in the 5.4 section on modelling the effects of climate change.

The Temperature Colour Once the final temperature is computed and stored in the pixel object, the `findPixelTemperatureColour()` method deals with determining the colour the pixel will have on the environment temperature map, which is also stored in the pixel object. `TemperatureSettings` is a class that holds a collection of predefined temperature values associated with colours. For instance, a temperature of 20°C is associated with a bright red, and a temperature of -3°C is associated with a dark blue. These predefined temperature settings are used to create gradient temperature maps. Thus, each pixel will have a temperature colour that is a gradient between the two temperature settings it lies between. For instance, a temperature of 18°C will lie between the 20°C setting, which is bright red, and the 16°C setting, which is orange. Its colour will be a gradient between red and orange, to reflect that it is warmer than 16°C, but cooler than 20°C. To achieve this for the current pixel, we iterate through the temperature settings collection until we find the two it lies between. Then, an intensity value is computed, which will be used in the `generateColour(intensity, colour1, colour2)` method of the `Renderer` class to reflect the distribution of the two colours setting in the final colour mix. To clarify, `generateColour()` receives two colours as parameters, the ones associated with the lower and upper bound temperature settings, and an intensity value. The intensity will determine how much of each colour should be added to the final colour of the current pixel. For instance, considering the previous example, the current pixel has temperature 18°C. The intensity is computed using the formula $intensity = \frac{(finalTemperature - lowerBound)}{(upperBound - lowerBound)}$, in this case $intensity = \frac{(18-16)}{(20-16)} = \frac{2}{4} = 0.5$, which means 18°C is equally close to 16°C and 20°C. The `generateColour()` method will receive 0.5 as the intensity and orange and red as the colours, which means it will return a colour that is a halfway mix between the orange and the red. If our temperature was 17°C, and thus the $intensity = \frac{(17-16)}{(20-16)} = \frac{1}{4} = 0.25$, the resulting colour would have been closer to the lower bound colour, which is orange. Thus, the final colour would have been 75% orange and 25% red.

5.2.3 Setting the Terrain

To reiterate, so far, for the current pixel, the noise height and greenhouse gas concentration were set, and the temperature was calculated in a three-step process. Returning to the application flow, inside the `calculateMap()` method of the `World` class, the `getTerrain()` method is called next, which deals with determining the terrain type of the pixel based on the terrain height and temperature previously calculated. This is achieved in a similar process as the one for setting the temperature colour.

An instance of the `TerrainType` class has a terrain name, a colour and an ID. The `TerrainSections` class contains a list of predefined terrain types that represent real terrain, like ice, water and grass. The `TerrainSections` class also has a `TreeMap` collection called `temperatureSections`. The type of the `temperatureSections` collection is `TreeMap<Integer, TreeMap<Float, Integer>>`. Each entry of the `temperatureSections` contains a temperature value associated with a second map, which we can refer to as a `heightSections`. An entry of a `heightSections` map represents a height, associated with a `TerrainType` ID. This `temperatureSections` collection is used to determine the `TerrainType` for the pixels. The temperature value of an entry in `temperatureSections` represents the upper bound of temperatures that can fit into that section. If the current pixel has a temperature lower than this bound, it will be part of this temperature section. For each temperature section (entry in `temperatureSections`) all possible terrain heights have a terrain type ID associated with them. This means that a pixel in this temperature section can find the terrain type ID associated with its height. For instance, if the current pixel had the temperature -6°C, it would it in the temperature section with the temperature bound -5°C. For all pixels with temperatures up to -5°C, the terrain type should be ice, for all terrain heights. Our pixel will then be assigned the ID of the ice terrain type. If the current pixel had the temperature 14°C, it would not fit the -5°C temperature section, so it would continue searching until it reaches a section with a bound above 14. Consider a temperature section with the bound of 15°C, which has a `heightSections` split as follows: up to the globally set `SEA_LEVEL`, the terrain is water; up to 20% above sea level, the terrain is grass; the remaining terrain to the top is rock. The pixel fits this section, so it begins searching for the correct terrain type inside the `heightSections` map. It will go through the entries until it finds a height bound that is bigger than the pixel's own height. Once a bound is found, the terrain ID associated with that bound is set to our pixel. It is essential that both `temperatureSections` and `heightSections` are of the type `TreeMap`, such that it is insured their entries will always be sorted by the keys. The order of entries is essential to this implementation, because if the sections are not in increasing order, so the pixels are compared to increasing temperatures (and heights) until they find the very first one that is bigger than their value. Otherwise, the pixels could be categorised within a temperature (or height) section that has a higher bound, but is not the closest one possible.

Thus, to set the terrain of the current pixel, we iterate over the `temperatureSections` map, until we find a temperature bound that is higher than the pixel temperature. Then, we iterate over the `heightSections` of that temperature section, until we find a height bound that is higher than the pixel height. We set the pixel terrain type to the `TerrainType` with that ID, and also save the terrain type immediately preceding the one we found. Following this, we repeat the same process as when setting the temperature colour, to determine the exact colour for the current pixel. The colour of the pixel will be a mix between the set terrain type and the one proceeding it in the `heightSections`. We calculate the intensity using the terrain height, to determine which section our terrain is closer to.

Once the terrain type is set, all the processing required for the pixel is finished. The same process is repeated for all pixels, until the entire map is generated.

5.2.4 Drawing the Map

With the map generated, `calculateMap()` returns to the `run()` method of the `WorldThread`, which informs the JavaFX Interface thread to stop the loading process and start the `drawMap()` method. `drawMap()` gets the map type to be drawn from the menu, and calls `Renderer.drawPixels()`. `Renderer.drawPixels()` handles the drawing of each different type of map by setting the colour of each pixel individually. For the height map, the colour is set to a grey value equal to the noise value of the pixel. For the temperature and terrain maps, the colour is set to the colours stored in the pixel object after the computations above. For the greenhouse gas concentration map, the colour is set using the `Renderer.generateColour()` method, to a colour between a light green for low concentration levels and bright red for high concentration levels.

After the `Renderer` draws each pixel,`drawMap()` adds event handlers to each pixel, such that whenever the user clicks on a pixel, all the information about that pixel will be printed to the information box for the user to see. The information includes: the coordinates, latitude, temperature, greehouse gas concentration,

Once the map is generated and drawn, the initial loading process of the application is complete. The user can now start interacting with the map and the menu options.

5.3 Generating the Animals

For the expected application usage, we assume that, after finding a map they want to observe climate change effects on, the user presses the `Generate Animals` button to add wildlife to the environment. This button press triggers the application to be put back in the loading state, while a worker thread is computing the processes inside `Life.generateLife()`. A list of available species is predefined inside the `Life` class. The species are instances of the classes that extend the abstract class `Animal`, in this case `PolarBear` and `Bee`. Animals have names, coordinates, images for rendering, a list of terrain types they can inhabit, the minimum and maximum terrain heights they can appear on, and a flag indicating whether they are alive or not. `Life.generateLife()` iterates over the map pixels. For each pixel, it iterates through the available animal species, and checks if the current pixel terrain is inhabitable by that type of animal and if the pixel terrain height is within the bounds that the animal can inhabit. If both conditions are passed, it means the environment is suitable for the animal. Once that is established, the probability for this animal to span on this pixel is calculated in the `Life.getAnimalProbability()` method. The function is designed such that the probability is highest for the animal to spawn when the terrain height of the pixel is equal to the middle point between the animal minimum and maximum terrain height. The probability decreases equivalently from that point, as the height increases or decreases. The highest probability an animal can have is determined by a `maxProb` field in the animal object, to ensure the map is not overloaded with too many animals, which would significantly slow down the system. For instance, in Figure 19 the middle point between the animal minimum and maximum terrain height is 0.5, thus the probability for the animal to spawn will be highest at that terrain height and decrease gradually from there.

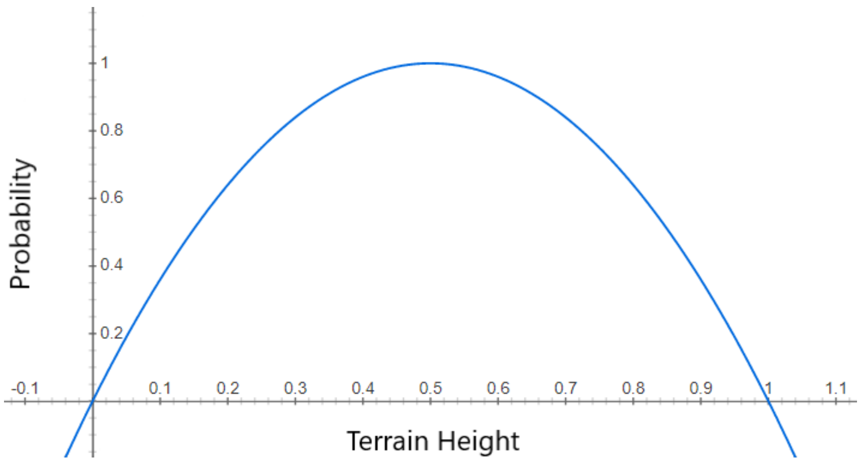


Figure 19: Animal spawn probability given the terrain height

After the probability is calculated, a random number generator is used to apply the probability. If the condition passes, an animal of that species is created on the pixel, and iteration moves to the next pixel. At the end, all the animals that were generated are added to the `animals` list stored in the `World` class.

Once the worker thread finishes generating the animals, the loading state is ended, and the map is redrawn by calling the `drawMap()` method. By default, `drawMap()` calls the `Renderer.drawAnimals()` method, which draws all the animals from the `animals` list on top of the map. Since until this moment the `animals` list was empty, no animals had been drawn. `Renderer.drawAnimals()` iterates over the animals in the list and displays the species' image, centred on the coordinates of the animal. If the animal is not alive when it is being drawn, a dark overlay is added on top of the animal image to suggest its state. Finally, an event handler is added to each animal image, such that when the user interacts with it a message is printed to the information box, which provides the user with general information about the animal, including its location, compatible terrains, current terrain and state.

5.4 Applying the Climate Change Effect on the Generated World

After adding the animals, we assume that the user will interact with the timeline slider, to observe the effects of climate change on the generated environment. Updating the slider will trigger the application to go into the loading state, while `TemperatureThread` deals with the computation required to update the environment. `TemperatureThread` calls the `updateMapCalculation()` method from the `World` class. This function covers both cases when the user moves the timeline forward or backward. The two operations are each other's inverses, meaning that moving forward and then backward should return the system to the same state as it was before moving forward and vice versa. `updateMapCalculation()` receives two parameters: a flag that indicates if time moved forward or backward, and an integer that indicates how many years the time changed by. The first step in updating the state of the environment is to update the sea level. A known effect of climate change is the sea level rising due to the Arctic ice cap and glaciers melting (*Global Ice Viewer* n.d., Maslin 2014). Thus, the sea level is updated depending on the direction the time moved in and how many years have passed. While this change in sea level is hardly noticeable* from a world-scale perspective, it can be observed when zooming in on map sections (see Figure 20), and it is important to reflect this event as it is an effect of climate change present in the real world.

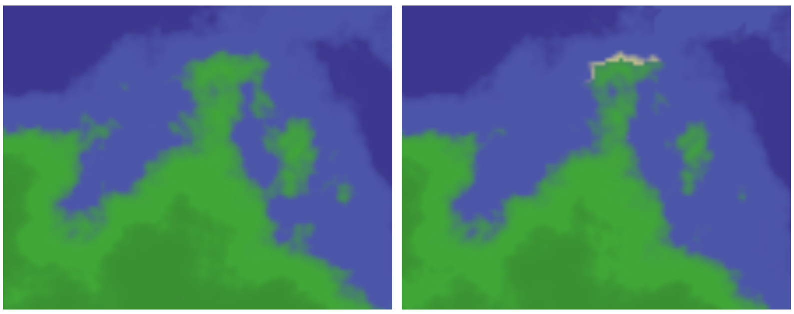


Figure 20: The sea level difference between 2000 (left) and 2100 (right)

After the sea level is set, iteration over all the pixels begins. For each pixel, the greenhouse gas concentration updates, depending on the direction time moved in. At each step, it increases or decreases by a factor of what it was before. This way, locations with higher gas concentrations have a more rapid growth, while locations with low concentration tend to increase emissions very slowly, which accurately reflects the events in the real world. Places with low emissions tend to stay that way, whereas places with higher greenhouse gas concentration tend to constantly increase their emissions (*Global Carbon Atlas* n.d.). Once greenhouse gas is updated, the temperature can be recalculated. The final step of the temperature calculation is reapplied, to reflect the changes in greenhouse gas concentration. Figure 21 depicts the change in the temperature patterns from 2000 to 2100 using the world temperature maps.

*According to NASA, if all the ice around the world melted, the sea level would rise by more than 60 meters

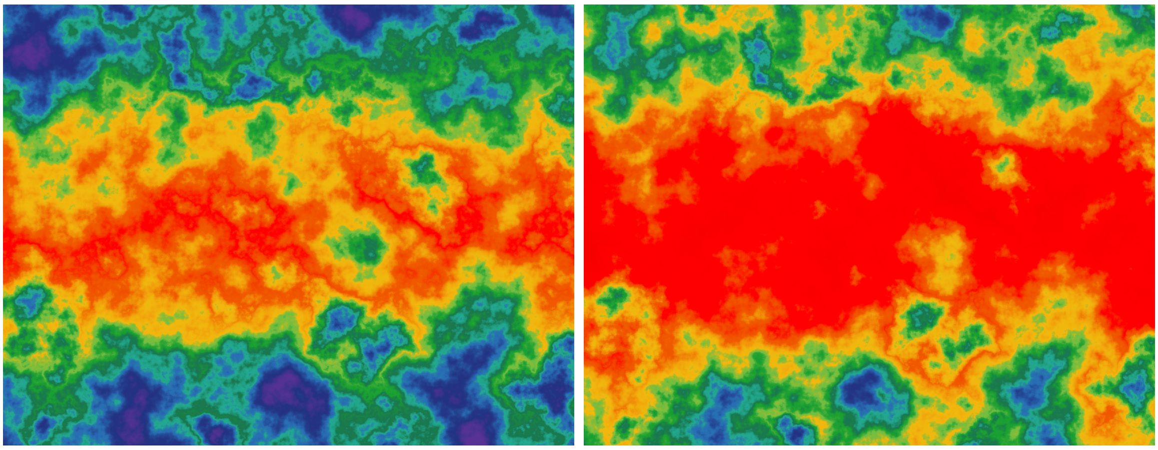


Figure 21: The temperature map difference between 2000 (left) and 2100 (right)

The sea level and temperature are both factors that influence the terrain, and since they have changed, so must the terrain. The terrain is recalculated as before. To finalises the environment update, the wildlife must be updated. We iterate over the list of animals in the environment and check whether the terrain the animal lies on is inhabitable by that species. If the terrain is not inhabitable by the animal, the animal dies. If the terrain is inhabitable, the animal is brought back to life. This is to ensure the operations of moving time forward and backward are inverses. Using this implementation, the user can observe how the size of an animal group decreases, even nearing extinction in the case of the polar bears, over time as its environment becomes less and less welcoming. Again, showing the harm that animals suffer as a result of climate change has been confirmed to be effective in motivating individuals to take action to prevent climate change (Swim & Bloodhart 2015). Figure 22 emphasises the changes that occurred to the terrain and wildlife from 2000 and 2100.

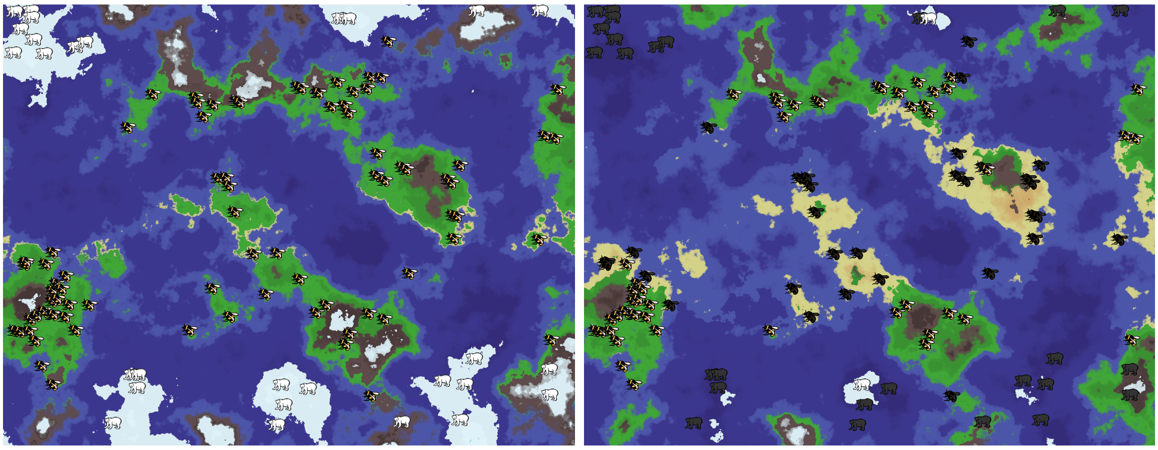


Figure 22: The terrain and wildlife state difference between 2000 (left) and 2100 (right)

Once the `TemperatureThread` completes the computations to update the map, the loading stops, and the map is redrawn. Finally, the dialog overlay is added on top of the map (see Figure 23). The dialog displays information collected throughout the computation process that if useful to the user, relating to the temperature increase and the state of the animals.

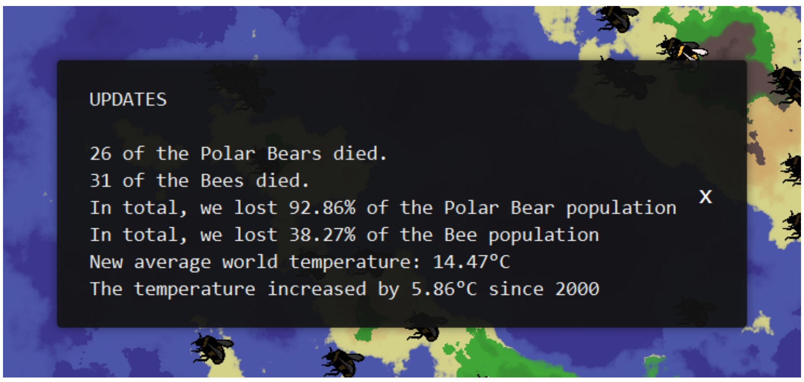


Figure 23: The information dialog

6 Project Evaluation

To reiterate, the aim of this project was to develop a tool that can generate random environments with animals, which can then be used to visualise some effects that climate change is predicted to have in the real the world over the remainder of this century. The tool aimed to be accessible and versatile, so that it could also be used to develop other more complex systems.

6.1 Testing

Considering the nature of the two main focuses of this project, world generation and climate change modelling, evaluating the system is not as intuitive as it can otherwise be in computation. There are no fixed parameters that can determine if a generated world is ”correct”, and there definitely are no parameters that can quantify the accuracy of modelling events 100 years into the future. Again, making predictions is not the aim of this system, hence the decision to use generated worlds for general modelling instead of the real world, for precise predictions. However, the goal is still to create a model that represents, as accurately as possible, the knowledge we have relating to climate change, and reflect the predictions researchers have made. For these reasons, evaluating the functionality of the system consisted of a substantial amount of empirical testing.

In terms of world generation, the best guide to determine the success of the system is our own experience and knowledge of the real world. The map settings sliders emerged out of the necessity to explore which parameters would achieve a result that best resembled real-world environments. Different combination of scale, octaves, lacunarity and persistence were tested until the current default settings were settled on. A second guide for the world generation was measuring the water and ice percentages of the environments. This process helped determine an appropriate sea level, which would result in water percentages that resemble the real world.

In terms of climate change modelling, the temperature was the primary factor that informed if the requirements have been met. The foregoing assumption was that the environments' average temperatures should increase by up to 6°C by 2100 (Pachauri & Reisinger 2007). This has guided the process of generating initial greenhouse gas concentrations, and the procedure used to increase the concentrations over time, such that the appropriate temperature increase would occur. To ensure that this requirement was consistently achieved, numerous random maps were sampled, and their temperature increase was verified. The results show that the temperature increase remains consistent, between the 5°C to 6°C margin. Results align across all the tested maps.

In general, the testing process of all the features of the visualisation consisted of experimenting with different parameters, functions or approaches, until satisfactory results that align with the expectations (maps that resemble our world, temperature increases of 6°C) were met.

6.2 Project Management

This system was built through an iterative and incremental development methodology. At the beginning of each week, I met with Ian, my supervisor, and we discussed the next iteration, based on what I achieved in the previous iteration and the overall requirements of the project. This was essential in helping me stay on track of the development and not lose focus of my final requirements. Each iteration aimed to incrementally add working features to the system.

6.3 Critical Analysis and Personal Development

Throughout the development of this project, I have solidified my interest in wold generation, modelling and visualisation. I was able to understand and employ popular techniques and algorithms that I previously assumed were beyond my reach.

While the overall requirements of the project have been met, there is undoubtedly room for improvement. In particular, a shortcoming of this system is that it exclusively models a uniform temperature increase as a result of climate change, whereas the supporting literature clearly states that while the overall world temperature would increase, different regions will suffer from different kinds of events. While some regions would get warmer, others would get cooler. This system does not cover the different types of reactions to climate change, but only focuses on temperature increase. The main reason for this is that temperature increases as a result of greenhouse gas emissions increasing, which can be modelled without much difficulty. However, the other effects are caused by other, more complex side effects of climate change, like changes in ocean currents or precipitations, which are inherently much more intricate systems, that are more challenging to model.

6.4 Future Work

I believe the next major focuses for this project would be to add more complexity to the climate change model and to the animals' behaviour.

6.4.1 Adding more layers to the model

As mentioned in section 6.3, the system could benefit from a more complex climate change model. In future iterations, I would like to explore concepts such as precipitation, wind and climate, and model how they would impact the effects of climate change.

In the past, I had considered randomising which regions of the map would get warmer and which would get cooler, and I would still be interested to explore this idea and observe the results.

Even further, I would like to explore the idea of anthropogenic biomes. As it stands right now, the system only considers natural habitats, without any human alterations. I would be interested in introducing human settlements and replacing the current random greenhouse gas concentration model with a model based around the development of these settlements. This would surpass the initial requirements of this project, as it was not intended to model what causes climate change. It would, however, add an interesting layer to the system, and perhaps accentuate the message of concern towards climate change, as users would be able to observe how anthropogenic actions harm the environment.

6.4.2 Complex Animal Behaviour

The animals currently in the system are simply a device to accentuate how serious the issue of climate change is, and they serve their purpose well. However, in future iterations I would like to explore the idea of animals trying to adapt to the changes, like looking for alternative habitats nearby, if their own has been compromised. At the moment, a Polar Bear will die if the pixel it lies on is no longer ice, even if the very next pixel to the right is still ice and it could relocate there. This is the type of behaviour I would like to explore in animals, to give them a better chance and see how that alters the effects of the climate change model.

6.5 Legacy of the Project

One of the main goals of this project has been to build a system that is accessible and can be used as a building block for other development projects. To ensure this, the source code has been thoroughly documented and overall designed with a clear, intuitive structure in mind. The project was made available on GitHub and will remain open source.

The reasoning behind this goal stemmed from my own experience, after struggling to find resources on the project topic that would help me throughout the development process. World generation is a popular topic, but clear explanations or examples to learn from are rare and difficult to come across. In terms of the climate change aspect, temperature modelling has been perhaps one of the biggest challenges of this project, as resources about how this can be achieved have been impossible to find. Gladly, I was able to extrapolate information about temperature from scientific sources and create a model that offers satisfying results. For these reasons, I felt compelled to make my results available online, for individuals who are interested in procedural generation and temperature and climate change modelling and find themselves in need of resources to support their learning or development.

7 Conclusion and Summary

Climate change is not a matter of debate, but a proven scientific fact. Vast amounts of data in the form of greenhouse gas measurement and temperature records support the theory. The fact remains that our ever-increasing anthropogenic emissions are bound to cost us, both literally and figuratively. The effects of climate change on the long term are expected to be not only damaging to the environment, but expensive to deal with as well. While the issue of climate change becomes more popular among scientist and researchers, the rest of the population is falling behind, and much more likely to be exposed to inaccurate information. We find ourselves at a point where, sooner rather than later we must take action to mitigate the effects of climate change. To be able to do that, efforts must be made to raise awareness and bring attention to the pressing matter of climate change. We require accessible tools that can effectively educate individuals on the risks we are facing as a result of excessive greenhouse gas emissions. This the problem this project addresses.

My contribution to addressing this problem was to build a system that can be used to visualise the results of climate. Throughout the development process, a second major goal of the project emerged: to create a versatile, accessible system that can be used for learning about world and climate change modelling, or for the development of more complex systems. To achieve these requirements, I employed popular software engineering techniques, like loose coupling of components, and separating the graphical component from the modelling component.

To avoid making inaccurate predictions of future events and focus on a general model of climate change, I opted to generate random environments to apply the climate change effects on rather than using a real-world representation. This decision also contributed to achieving the versatility requirement of the system, making it more flexible for any needed purposes. To generate random environments, I used coherent noise that creates natural textures, together with my three-step temperature calculation. Using these two, I generate the terrain that constitutes the environment. The user of the system can choose to add animals to this environment, and then proceed to observe the effects of climate change in action on the environment, by manipulating the timeline slider.

To create a system that models environments that resemble the real world and climate change effects that match the projection predicted by researchers, I had to rely substantially on empirical evaluation. It was a process of creating tool to facilitate the experimentation, like the map settings sliders, and constantly testing different approaches and functions to observe which one provides the most satisfying results.

The final system meets the outlined requirements. It provides users with the opportunity to generate trillions of unique environments (see Appendix A.3), and observe how they are impacted as a result of temperature increase caused by increasing greenhouse gas concentrations. Even more, they can get a sense of how animals can be affected by climate change. The system is easy to navigate and intuitive to use, and it is built with developers in mind. It is extensively documented and designed with an uncomplicated structure, such that the functionality can be easily grasped and extended upon. The project is open source, available on GitHub, and ready to be used for education, learning or development. Overall, the tool achieved what it set out to do.

While a visualisation tool will not solve the issue of climate change, it may bring a contribution to the course of events that ultimately does. After all, *"Change is the end result of all true learning."* – Leo Buscaglia.

References

Archer, D. & Ganopolski, A. (2005), 'A movable trigger: Fossil fuel CO2 and the onset of the next glaciation', *Geochemistry, Geophysics, Geosystems* **6**(5). Publisher: John Wiley & Sons, Ltd.
URL: *https://doi.org/10.1029/2004GC000891*

Archer, T. (n.d.), Procedurally generating terrain.

Arrhenius, S. (1896), 'XXXI. On the influence of carbonic acid in the air upon the temperature of the ground', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **41**(251), 237–276. Publisher: Taylor & Francis.
URL: *https://doi.org/10.1080/14786449608620846*

Assessing the Global Climate in 2019 (2020). Library Catalog: www.ncei.noaa.gov.
URL: *http://www.ncei.noaa.gov/news/global-climate-201912*

Braje, T. J. & Erlandson, J. M. (2013), 'Human acceleration of animal and plant extinctions: A Late Pleistocene, Holocene, and Anthropocene continuum', *When Humans Dominated the Earth: Archeological Perspectives on the Anthropocene* **4**, 14–23.
URL: *http://www.sciencedirect.com/science/article/pii/S2213305413000118*

Climate Change Projections (n.d.). Library Catalog: climatechange.codeforafrica.org.
URL: *https://climatechange.codeforafrica.org/map.html*

Climate Impact Lab (n.d.). Library Catalog: www.impactlab.org.
URL: *http://www.impactlab.org/map/usmeas=absoluteusyear=1981-2010gmeas=absolutegyear=1986-2005tab=global*

Future Delta 2.0 (n.d.).
URL: *https://futuredelta2.ca/*

Giannini, T. C., Acosta, A. L., Garófalo, C. A., Saraiva, A. M., Alves-dos Santos, I. & Imperatriz-Fonseca, V. L. (2012), 'Pollination services at risk: Bee habitats will decrease owing to climate change in Brazil', *Ecological Modelling* **244**, 127–131.
URL: *http://www.sciencedirect.com/science/article/pii/S0304380012003250*

Global Carbon Atlas (n.d.).
URL: *http://www.globalcarbonatlas.org/en/CO2-emissions*

Global Climate Report - Annual 2017 (n.d.).
URL: *https://www.ncdc.noaa.gov/sotc/global/201713*

Global Ice Viewer (n.d.). Library Catalog: climate.nasa.gov.
URL: *https://climate.nasa.gov/interactives/global-ice-viewer*

Hári, D. (n.d.), 'A scalable, pannable JavaFX ScrollPane'.
URL: *https://stackoverflow.com/a/44314455*

Interactives – Climate Time Machine (n.d.). Library Catalog: climate.nasa.gov.
URL: *https://climate.nasa.gov/interactives/climate-time-machine*

Johnston, K. & Schmitz, O. (1997), 'Wildlife and climate change: assessing the sensitivity of selected species to simulated doubling of atmospheric CO2', *Global Change Biology* **3**(6), 531–544. Publisher: John Wiley & Sons, Ltd.
URL: *https://doi.org/10.1046/j.1365-2486.1997.00093.x*

Keeling, C. D., Bacastow, R. B., Bainbridge, A. E., Ekdahl Jr., C. A., Guenther, P. R., Waterman, L. S. & Chin, J. F. S. (1976), 'Atmospheric carbon dioxide variations at Mauna Loa Observatory, Hawaii', *Tellus* **28**(6), 538–551. Publisher: Taylor & Francis.
URL: *https://doi.org/10.3402/tellusa.v28i6.11322*

Lague, S. (n.d.), 'Procedural Landmass Generation (E01: Introduction)'.
URL: *https://www.youtube.com/watch?v=wbpMiKiSKm8t=2s*

Le Conte, Y. & Navajas, M. (2008), 'Climate change: Impact on honey bee populations and diseases', *Revue scientifique et technique (International Office of Epizootics)* **27**, 485–97, 499.

Lynas, M. (2008), *Our future on a hotter planet*, HarperPerennial : [distributor] HarperCollins Distribution Services, London.

Maslin, M. (2014), *Climate Change: A Very Short Introduction*, OUP Oxford. Google-Books-ID: hmT-DBAAAQBAJ.

NASA (n.d.), 'Earth Atmosphere Model'.
URL: *https://www.grc.nasa.gov/WWW/K-12/airplane/atmosmet.html*

Nocke, T., STERZEL, T., Böttinger, M. & Wrobel, M. (2008), 'Visualization of Climate and Climate Change Data: An Overview', *in Ehlers et al. (Eds.) Digital Earth Summit on Geoinformatics 2008: Tools for Global Change Research (ISDE'08), Wichmann, Heidelberg, pp. 226-232, 2008* .

O'Neill, S. J. & Smith, N. (2014), 'Climate change and visual imagery', *WIREs Climate Change* **5**(1), 73–87. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/wcc.249.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1002/wcc.249*

Pachauri, R. & Reisinger, A. e. (2007), Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, Technical Report 4, IPCC, Geneva, Switzerland. Library Catalog: www.ipcc.ch.
URL: *https://www.ipcc.ch/report/ar4/syr/*

Peck, J. (2020), 'Auburns/FastNoise_java'. original-date: 2017-04-04T20:08:42Z.
URL: *https://github.com/Auburns/FastNoise$_java$*

Perlin, K. (1985), 'An image Synthesizer', *ACM SIGGRAPH Computer Graphics* . Publisher: ACM PUB27 New York, NY, USA.
URL: *https://dl.acm.org/doi/abs/10.1145/325165.325247*

Perlin, K. (2001), 'Noise hardware', *Real-Time Shading SIGGRAPH Course Notes* .

Perlin, K. (2002), Improving Noise, *in* 'Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques', SIGGRAPH '02, Association for Computing Machinery, New York, NY, USA, pp. 681–682. event-place: San Antonio, Texas.
URL: *https://doi.org/10.1145/566570.566636*

Pielke, R. A. & Tisdale, B. (n.d.), 'Sea Surface Temperature Trends As A Function Of Latitude Bands'.
URL: *https://pielkeclimatesci.wordpress.com/2012/07/11/sea-surface-temperature-trends-as-a-function-of-latitude-bands-by-roger-a-pielke-sr-and-bob-tisdale/*

Planton, S. (2013), IPCC, Annex III: Glossary. In: Climate Change 2013: The Physical Science Basis., *in* 'Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change [Stocker, T.F., D. Qin, G.-K. Plattner, M. Tignor, S.K. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex and P.M. Midgley (eds.)].', IPCC, Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA. Publisher: Citeseer.
URL: *https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5$_A$nnexIII$_F$INAL.pdf*

Radchuk, V., Reed, T., Teplitsky, C., van de Pol, M., Charmantier, A., Hassall, C., Adamík, P., Adriaensen, F., Ahola, M. P., Arcese, P., Miguel Avilés, J., Balbontin, J., Berg, K. S., Borras, A., Burthe, S., Clobert, J., Dehnhard, N., de Lope, F., Dhondt, A. A., Dingemanse, N. J., Doi, H., Eeva, T., Fickel, J., Filella, I., Fossøy, F., Goodenough, A. E., Hall, S. J. G., Hansson, B., Harris, M., Hasselquist, D., Hickler, T., Joshi, J., Kharouba, H., Martínez, J. G., Mihoub, J.-B., Mills, J. A., Molina-Morales, M., Moksnes, A., Ozgul, A., Parejo, D., Pilard, P., Poisbleau, M.,

Rousset, F., Rödel, M.-O., Scott, D., Senar, J. C., Stefanescu, C., Stokke, B. G., Kusano, T., Tarka, M., Tarwater, C. E., Thonicke, K., Thorley, J., Wilting, A., Tryjanowski, P., Merilä, J., Sheldon, B. C., Pape Møller, A., Matthysen, E., Janzen, F., Dobson, F. S., Visser, M. E., Beissinger, S. R., Courtiol, A. & Kramer-Schadt, S. (2019), 'Adaptive responses of animals to climate change are most likely insufficient', *Nature Communications* **10**(1), 3109.
URL: *https://doi.org/10.1038/s41467-019-10924-4*

Rasmont, P., Franzén, M., Lecocq, T., Harpke, A., Roberts, S., Biesmeijer, K., Castro, L., Björn, C., Dvořák, L., Fitzpatrick, , Gonseth, Y., Haubruge, E., Mahe, G., Manino, A., Michez, D., Neumayer, J., Ødegaard, F., Paukkunen, J., Pawlikowski, T. & Schweiger, O. (2015), *Climatic Risk and Distribution Atlas of European Bumblebees*, Vol. 10. Journal Abbreviation: BIORISK ? Biodiversity and Ecosystem Risk Assessment Pages: 246 Publication Title: BIORISK ? Biodiversity and Ecosystem Risk Assessment.

Root, T. L., Price, J. T., Hall, K. R., Schneider, S. H., Rosenzweig, C. & Pounds, J. A. (2003), 'Fingerprints of global warming on wild animals and plants', *Nature* **421**(6918), 57–60.
URL: *https://doi.org/10.1038/nature01333*

Sawe, B. E. (n.d.), 'What Is The Effect Of Latitude On Temperature?'.
URL: *Sawe, Benjamin Elisha. "What Is The Effect Of Latitude On Temperature?" WorldAtlas. https://www.worldatlas.com/articles/what-is-the-effect-of-latitude-on-temperature.html (accessed April 5, 2020).*

Schroth, O., Angel, J., Sheppard, S. & Dulic, A. (2014), 'Visual Climate Change Communication: From Iconography to Locally Framed 3D Visualization', *Environmental Communication* **8**(4), 413–432. Publisher: Routledge.
URL: *https://doi.org/10.1080/17524032.2014.906478*

Sheppard, S., Burch, S., Shaw, A. & Flanders, D. (2010), 'Planning for climate change in a flood-prone community: municipal barriers to policy action and the use of visualizations as decision-support tools', *Journal of Flood Risk Management* **3**(2), 126–139. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1753-318X.2010.01062.x.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1753-318X.2010.01062.x*

Sheppard, S., SHAW, A., Flanders, D. & Burch, S. (2008), 'Can Visualisation Save the World? - Lessons for Landscape Architects from Visualizing Local Climate Change', *Conference Proceedings, Digital Design in Landscape Architecture, 9th International Conf* .

#ShowYourStripes (n.d.).
URL: *https://showyourstripes.info/*

Stern, N., Stern, N. H. & Treasury, G. B. (2007), *The Economics of Climate Change: The Stern Review*, Cambridge University Press.

Swim, J. K. & Bloodhart, B. (2015), 'Portraying the Perils to Polar Bears: The Role of Empathic and Objective Perspective-taking Toward Animals in Climate Change Communication', *Environmental Communication* **9**(4), 446–468. Publisher: Routledge.
URL: *https://doi.org/10.1080/17524032.2014.987304*

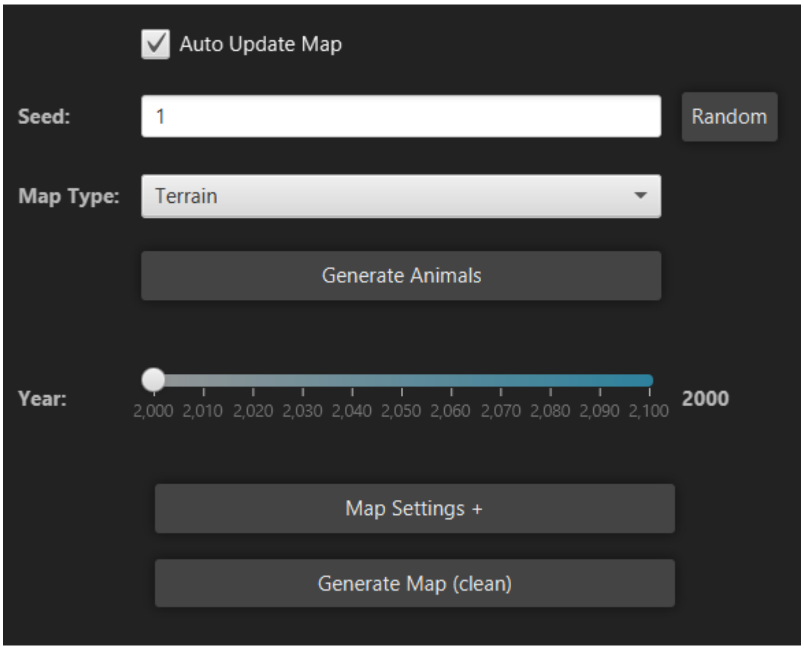
The Best Data Viz and Infographics on Climate Change Facts (n.d.). Library Catalog: visme.co.
URL: *https://visme.co/blog/climate-change-facts/*

Thomas, C. D., Cameron, A., Green, R. E., Bakkenes, M., Beaumont, L. J., Collingham, Y. C., Erasmus, B. F. N., de Siqueira, M. F., Grainger, A., Hannah, L., Hughes, L., Huntley, B., van Jaarsveld, A. S., Midgley, G. F., Miles, L., Ortega-Huerta, M. A., Townsend Peterson, A., Phillips, O. L. & Williams, S. E. (2004), 'Extinction risk from climate change', *Nature* **427**(6970), 145–148.
URL: *https://doi.org/10.1038/nature02121*

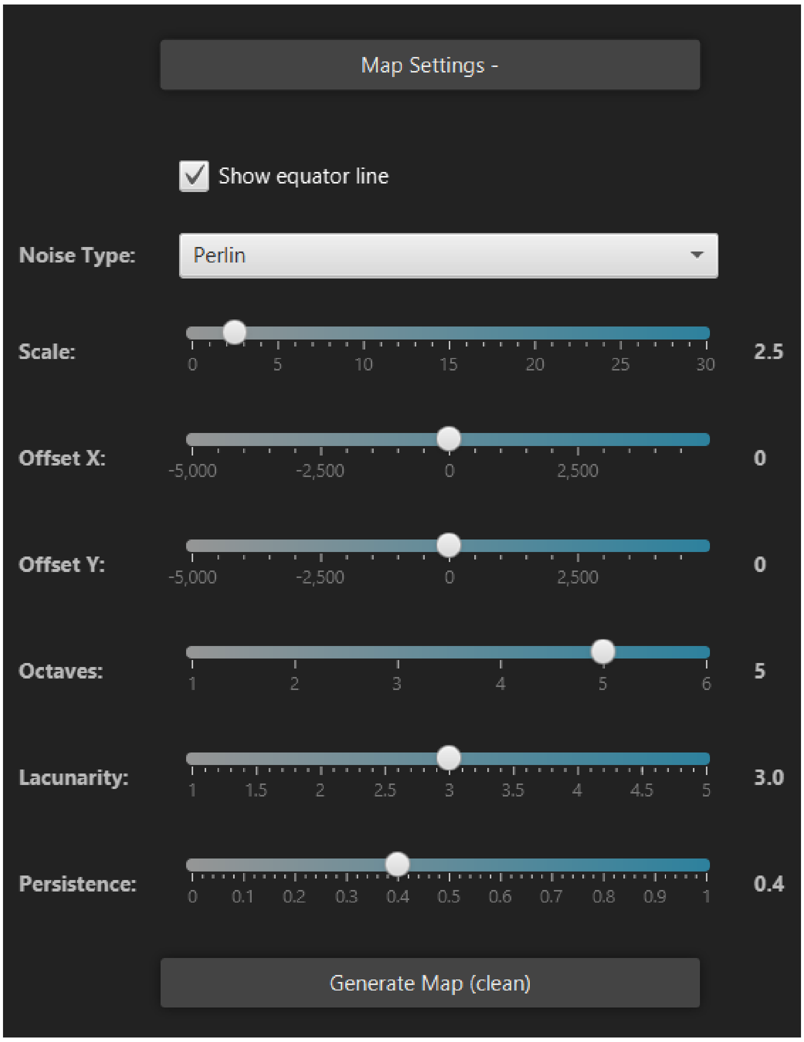
Warming stripes (n.d.). Library Catalog: www.climate-lab-book.ac.uk.
URL: *https://www.climate-lab-book.ac.uk/2018/warming-stripes/*

A Appendices

A.1 Appendix: Main Menu



A.2 Appendix: Map Settings Menu



A.3 Appendix: How many unique maps can the system generate?

While in theory, the number of possible unique procedurally generated maps using the techniques described in this paper is infinite, in computation we are restricted by number representations. We can, however, approximate the number of unique maps that this system can generate.

There are 8 factors that contribute to the creation of a map: seed, type of noise, scale, octaves, lacunarity, persistence, vertical offset and horizontal offset. Of these, we know that the values of octaves, lacunarity and persistence are essential to maintain the authentic, natural look of the map. They should always be kept the same, thus they cannot be considered for this calculation, as we want to count the number of unique yet realistic maps we can create. The scale is also redundant, as changing it does not produce new maps, but simply adjusts the scale of pre-existing ones. This means we are left with seed, type of noise, vertical offset and horizontal offset.

The seed is what distinguishes maps most of all, and it acts as their unique identifier. In the system, the seed is represented as an integer, and considering the integer definition in Java, it means there are approximately 4.29 billion possible map seeds available.

Moving on to the noise type, there are 8 available noise types in the system (Perlin, Perlin Fractal, Simplex, Simplex Fractal, Value, Value Fractal, Cubic, Cubic Fractal), but the fractal variants of noise are not significantly different from the non-fractal type, so they do not produce different enough maps. Of the remaining ones, Value noise and Cubic noise produce very similar noise maps, owning to similar noise functions. Thus, we can only consider 3 type of noise that produce entirely unique maps from each other. For each seed, we can create 3 unique maps using the noise type.

Finally, the offset is what, in theory, makes any singular map infinite, as you can explore different areas of the noise. In this system, the offset values are controlled through slider in the ranges -5000 to 5000. Increasing the slider range beyond that point would be redundant, because given the fact that the slider stays the same size, there are a limited number of steps you can make when moving the slider. Therefore, no matter if the bound is 5000, 50000 or 500000, it would not change the number of different maps you can create by "physically" moving the slider. Considering all this, moving a slider by 50 units (from 0 to 50), it has been observed the maps are sufficiently different to be considered unique. Thus, for a slider in the range -5000 to 5000, taken in steps of 50, you can create $\frac{10000}{50} = 200$ unique maps for each seed. There are two offset slider, therefore we can create 4000 (for each step on one slider, you can generate 200 more unique ones using the other slider, thus $200 \times 200 = 4000$ unique maps for each seed using the offset.

Considering the above, for each of the 4.29 billion seeds, there can be 3 unique maps created using the noise type, and for each of those, there can be 4000 unique maps generated using the offset. Therefore, the total number of unique maps that this system can procedurally generate is $4.29bn \times 3 \times 4000 =$ **51.5 trillion**.