Simulating human civilisation and the evolution of society based on factors such as culture, religion, and technology

James Benjamin Brimelow Gorman, 100505349

College of Engineering & Technology, University of Derby

100505349@unimail.derby.ac.uk

**Abstract:** LOREM IPSUM

**Keywords:** LOREM IPSUM

Contents

[1 Introduction 2](#_Toc87705767)

[2 Literature Review 2](#_Toc87705768)

[2.1 Dwarf Fortress 2](#_Toc87705769)

[2.1.1 World Generation 3](#_Toc87705770)

[2.2 Europa Universalis 4 3](#_Toc87705771)

[2.2.1 Provinces 3](#_Toc87705772)

[2.2.2 Artificial Intelligence 4](#_Toc87705773)

[2.2.3 Random New World 5](#_Toc87705774)

[2.3 Sid Meier’s Civilization 5 5](#_Toc87705775)

[2.3.1 Artificial Intelligence 5](#_Toc87705776)

[2.4 Iron Age 6](#_Toc87705777)

[2.4.1 World Generation 6](#_Toc87705778)

[2.4.2 Simulation & Artificial Intelligence 7](#_Toc87705779)

[2.4.3 Tile and National Names 7](#_Toc87705780)

[3 Artefact Design 8](#_Toc87705781)

[4 Bibliography 8](#_Toc87705782)

1. Introduction

LOREM IPSUM

1. Literature Review

The uniquity of this project comes in the form of its ability to put existing concepts such as world generation and time simulation together – software already exists to complete some of the objectives of this artefact, but little has been done to collate these ideas into one package. In order to perfect the individual components of the project, analysis into similar projects was completed.

* 1. Dwarf Fortress

The first similar system identified was the 2006 game Dwarf Fortress (Tarn Adams, 2006). This game, which began development in 2002, is set in a world populated by fantasy creatures such as elves, dwarves, and goblins. The objective of the player is to build a new colony for their chosen dwarven civilisation, trade with nearby populations and accumulate wealth. The game is notorious for its difficulty and attention to detail, particularly in its simulation of the world.

The game world is generated by the player at the beginning of the game and uses an entirely random map fit with its own history and civilisations, as well as records such as that of key peoples and artefacts. This, much like the artefact under development for this project, is done from the beginning of history, starting with an empty map that slowly becomes populated as time progresses. The impact of this approach to simulation allows the world to be truly defined by what occurred in its past: wars fought between nations in the past may affect the players ability to trade with others, artefacts of ancient history may be found by the player and can be stolen by enemies, the titular dwarves can even be seen engraving depictions of historical events or peoples in some cases.

This historical simulation system also has a direct impact on gameplay, a testament to the depth of the implementation. Some interesting examples of how the timeline of the world can affect gameplay are as follows:

* When starting a game, the player may choose their civilisation and what resources they will bring to their new colony. The resources available are defined by what the chosen civilisation is capable of – such as what metals they have available or what animals the nation has successfully domesticated.
* It is possible for a player to find themselves in a world that is missing a race, including as the playable dwarves, as a result of past events such as wars.
* The world is able to undergo changes in “age”, as defined by what peoples and monsters exist in the world at any given time. It is even possible for the world to enter a state in which all fantasy elements, such as dragons or monsters, have become extinct, which is referred to by the game as “The Age of Civilisation” (Dwarf Fortress Wiki, n.d.)

While this project includes many features applicable to this project, it is important to note the simulation serves as a backdrop to the main gameplay. Many liberties are taken to ensure the world is playable – a world without dwarves will still allow a player to start a colony and will simply spawn new people for the player to use. As such, while the game serves as an inspiration for this project, it has very little that can be used as material to work from. The following concepts highlighted are aspects which this project can gain insight from.

* + 1. World Generation

As described by the creator (Adams, 2009), the world generation method centres around elevation, with a location on the map being given a height and the sea level being defined as a result of the elevation of the world. This generated topographical map is then provided factors including temperature and rainfall – by which other factors such as river locations or biomes are built from. This approach to world generation is of particular interest, the concept of building a world by building layers of random content – elevation, temperature, rainfall – could be an appropriate method to be used by this project due to the relevance of these factors and their derivatives to the objective of the artefact.

The elevation generation seen in Dwarf Fortress makes use of a variant to the diamond-square algorithm (Adams, 2013) (Alain Fournier, 1982), the result of this being results that appear to be more continent-like than other algorithms in this field. Additionally, the developer of the game noted that the game did not use the more common Perlin noise method for world generation, citing it as looking “fluffy and rounded” (Adams, 2010). This assertion has implications for the direction of this project, as the diamond-square algorithm may serve as a better fit for the project than the original intended algorithm – Perlin noise.

* 1. Europa Universalis 4

Europa Universalis 4 is a game developed by Paradox Development Studio (Paradox Development Studio, 2013), which was released in August of 2013. The game presents a map of the real world as it emerges from the end of the medieval era, with gameplay spanning from the renaissance to the end of the age of revolutions. Unlike the aforementioned Dwarf Fortress, Europa Universalis 4 (Henceforth referred to as Eu4) places more emphasis on realism, with the map the game takes place on closely reflecting that of the real world in the time period a player chooses to start from. Additionally, the mechanics of the game strongly influence the human and computer players to act how a ruler in that time would have. This can be seen very clearly when reviewing the history of a game, in which certain real events will almost inevitably occur if the player does not intervene, such as the formation of the Russian empire, the uniting of the British Isles and the expansion of the Ottoman empire. Eu4 presents an interesting study into the field of simulating real world history, and examination into this media yields some interesting results.

* + 1. Provinces

First and foremost, the basic gameplay elements present within the game should be noted. The map in Eu4 is split into different “Provinces” – Polygonal areas representing the borders of a location, often defined by the existence of cities and population centres within the region. These provinces are the foundation on which the game is played – the nations a player represents own a certain number of provinces which represent the borders of the country, and the objective of the game can be said to be to build a large empire by taking as many provinces as possible, either through war or colonisation.

In addition to this, these provinces have a number of properties tied to them which determine how they interact with the gameplay. Some notable examples are the following (Paradox Development Studio, n.d.):

* Provincial development (Split between Taxation, Production and Manpower). This represents the infrastructure present in the area, higher developments corresponding to bigger cities which provide the player with more money, trade power and manpower. The development of a province also determines the value of the location in peace negotiations, the cost to the player to build an administration to the region as well as how the international community reacts to the annexation.
* Terrain. The environment of a province determines aspects such as the speed at which military units are able to travel through a location, the amount of supplies that can reach unit as well as the cost of developing the province. This provides a double-edged sword type scenario – a mountainous terrain is more difficult for an enemy to traverse and survive in, but the player will incur additional penalties for attempting to improve the infrastructure in the region.
* Culture. Each province has a culture that represents the peoples within it and occupying nations will find additional difficulty in attempting to maintain control of provinces with cultures that they do not support. Additionally, the player and the artificial intelligence is rewarded in various ways for occupying locations with their own national culture, incentivising the formation of borders based upon cultural boundaries.

The provincial system presented by Eu4 may be an aspect to consider in the design of this project. Splitting the world up in this way allows for the mapping of populations and population centres onto the model and may provide some advantages in organising the artificial intelligence of the simulation. As the system presented in Eu4 is designed to work on a pre-determined map, some adjustments need to be made to fit the random nature of the artefact.

* + 1. Artificial Intelligence

The Artificial Intelligence of Eu4 is largely classified data belonging to the development studio, however some things about its inner workings are known. As previously mentioned, the computer players will often prioritise the uniting of their culture, but this is additionally modified by aspects such as relations with foreign nations, the difficulty of taking provinces and the decision making of the rulers in charge of each nation.

A good example of how relations impact the decisions made by the AI can be seen in interactions between Spain (Known in the timeframe of the game as Castille) and Portugal. Both Spain and Portugal belong to the same “Iberian” culture group, and yet conflicts between them are minimised. This is partially due to the opinion modifier each country has on the other, citing historical alliances, meaning the two often prioritise other conquests rather than turning against one another.

In terms of rulers, computer-controlled nations will be designated a ruler “personality”, which is changed when a leader abdicates or dies. These personalities impact the actions nations will take during that leader’s reign, with militaristic rulers focusing on conquest and military expansion, and administrative rulers working instead to improve their economic condition. This creates a believable feeling of an evolving diplomatic landscape – players may find that once passive nations will suddenly become aggressive as a militarist takes charge, or former threats become minimised by a ruler less interested in war.

Particularly due to its ability to constantly change the diplomatic landscape of the world, ruler-based artificial intelligence is to be considered when determining the simulated history aspect of the development cycle. This system allows for great rises to power as well as falls due to negligence on the part of the ruler, which has been seen throughout real world history extensively.

* + 1. Random New World

The first purchasable update to Eu4, named “Conquest of Paradise” (Valve Corporation, 2014) , introduced a feature referred to as “Random New World”, advertised as the ability to generate an entirely new continent as a replacement for the Americas. This meant players would have the ability to explore a random map and build a new colonial empire as they explored new terrain. Upon release, this particular feature was met with mixed reception – partially as a result of the algorithm used for the random new world generation.

The generation mapping makes use of set-piece based method – in which the newly generated map is built from a number of premade assets put together to make a map. This method has seen successful usage in games like Spelunky HD (Mossmouth, 2013) and The Binding of Isaac (Valve Corporation, 2011), however this method reflects poorly in a geographic environment. The set pieces present in “Conquest of Paradise” are often individual islands, rather than the large landmass present in the real-world Americas, and veteran players will quickly come to recognise the set pieces as they appear – removing the charm of the pseudo-random geography. Additionally, as the sets are human made, they often reflect themes or shapes – such as the set piece that resembles the corporate logo of the Paradox Development Studio – which does not create a believable environment for the game to be played upon.

Unlike the other aspects mentioned in the discussion of this product, this random generation method should be avoided, as it does not suit the attempted goals of the random world generation feature of this project. As an extension to this, the use of any predetermined structures, such as the polygons of provinces, should be avoided unless necessary – as they represent many of the same issues discussed above.

* 1. Sid Meier’s Civilization 5

Another game that has some relevance to this project is the 2010 Firaxis game Sid Meier’s Civilization 5 (Firaxis, 2010). In this game, the player is prompted to choose a historical nation and leader, before being presented with a hexagonal world map for them to build a new nation upon. Other nations, led by civilisations and leaders that the player may have selected, exist on this map concurrently, and the nations of the world will compete to reach a number of arbitrary “Victory” conditions. This game is far more gameplay focused than titles like the previously mentioned Europa Universalis 4, forgoing much realism in favour of creating an entertaining product. This would largely make it inapplicable to this artefact, which focuses almost entirely on realism, aside from the inner workings of the AI within the game.

* + 1. Artificial Intelligence

Each leader in Sid Meier’s Civilization 5 has a number of integer values representing their focus towards different tasks – set on a scale of 1 to 10 (With one glaring exception) – this in turn influences the actions the civilization in question will take. For example – the nation of France, lead by Napoleon Bonaparte, has a “Loyalty” score of 3 – representing the real-life actions of the man, who was notable for violating various treaties and acting aggressively towards former allies. These values in turn then provide weighting to the decisions of the computer players, allowing them to act in pseudo-random ways which reflect the personalities assigned to them. This system allows for the nations of the world to act as if they have their own personalities and goals, and the actions of certain leaders can be predicted based on these hidden values.

This works as an extended implementation of the “personality” system present in Eu4. Whereas the Eu4 system works to provide a ruler a focus in areas such as diplomacy or economy, the Civilization approach instead provides weighting to specific areas – allowing for the characters within the game to have more of a character. This could see use in the artefact described in this document, with random rulers receiving random values that weight decision making. This, in addition, could permit the rulers to make mistakes – a ruler with a high weighting towards war could start conflicts with larger nations, potentially having significant consequences on the nation as a whole. The ability for a leader to make mistakes is one that should see implementation in the simulation, as misguided decisions have changed history countless times.

* 1. Iron Age

Iron Age (Gorman, 2018) was the working title of a previous attempt at a concept similar to that of this artefact conducted by myself in 2018. The software produced as a result of this endeavour was of amateur quality and did not meet the technical expectations it had set out to achieve, but yet managed to display two aspects relevant to the current artefact – World generation and time simulation. Despite the flaws of this project, some valuable research can be gained from investigation into the inner workings of this system.

* + 1. World Generation

Throughout development, Iron Age underwent no less than three different implementations of world generation algorithms – each uniquely designed to fit the project.

The first, referred to internally as the “Snake method”, involved a tile-based approach, in which a random index would be selected in a two-dimensional array as a “Snake”. This “Snake” block would then expand to an adjacent index, which would then repeat the process, expanding to another index location. Once this process was repeated a number of times defined by the parameters of the generator, each existing “Snake” block would generate 3 “land” tiles in each orthogonal direction, before becoming a “land” tile itself. This method allowed for the generation of long continents, as well as permitting the generation of features like inland seas and coves. However, in the end this method was not used due to its unpredictability and tendency to develop unrealistic geography.

The second method, the “Fill method”, was developed and rejected in less than a day. The concept was simple, populate a two-dimensional array with a number of tiles in random indexes with a number representing their elevation, then populate all tiles in a 3x3 area around said tile with a random modification to the elevation of its creator. Eventually this would produce a mountainous terrain map which would then be designated as sea level below a certain elevation, corresponding to the roughly 70:30 ratio of sea to land seen in real life. This system, while similar to those in use in similar projects, was not implemented in a way that benefitted from the strengths of elevation-based terrain – the nature of the expanding elevations meant that the landmasses produced were often of the same shape, and constructs like islands or inland seas simply did not appear.

Finally, the implemented “Continental method” was inspired by the real-world separation of Pangea into the modern continents. The model would produce a circle with a radius almost equivalent to the height of the map, which would be entirely filled with land tiles. The outermost tiles of the circle would then become “Active” tiles, which could move anywhere within a 3x3 area – switching the specified index to an active tile and replacing the previous tile with an ocean tile. These active tiles would then propagate simultaneously, creating random lines in the land, until they reached a point where there were no possible land locations to “jump” to in the 3x3 Area. Once all “Active” tiles had been removed, a flood fill algorithm would be utilised to identify each landmass and place it randomly in the ocean. This algorithm produced favourable results, producing random continents with believable proportions.

For this artefact, none of these methods should be used, as despite the relative success of the “Continental method” there were numerous flaws that could be alleviated by the use of other algorithms. Additionally, the continental method as implemented in Iron Age was particularly inefficient, using far more time and processing power than would typically be necessary for a world generation algorithm.

* + 1. Simulation & Artificial Intelligence

As discussed, Iron Age attempted to meet many of the same objectives as outlined in this artefact, including the development of nations across time. The simulation aspect of Iron Age was flawed in many ways, first and foremost in its scalability, a mistake which cannot be repeated in this artefact.

Iron Age implemented a system in which all nations generated a certain amount of “military units” per season, the amount of which would be modified by the technology of the owning civilisation, the number of tiles owned by the civilisation and the number of resources in the capital of the owning civilisation. These units would continuously generate as time progressed and would only be expended when the computer players attempted to take land from another nation. This resulted in an inevitable overflow in which a nation would exceed the bounds of the integer data allocated as their military size – an issue that was compounded by the already extensive use of memory by the software.

Additionally, the artificial intelligence implemented for Iron Age was very simplistic, and the only concerns of nations revolved around taking more land – with little regard for any properties of the land in question. This was sufficient for the small scale of Iron Age and its usage as an experiment into simulation but prevented the system from depicting a believable world. Additionally, the nations of the world fit a rigid decision-making structure – only declaring war if they had a numerical advantage over their competitor of above a certain constant value. This prevented the nations from taking risks, or developing any real form of personality, an aspect which will be emphasized in the development of this project.

* + 1. Tile and National Names

As a final note on Iron Age, there is one method that may see recreation for the purposes of this artefact, notably in the naming system for locations. Upon generating a world, the user would be prompted to select a method of name generation – either by random selection or random generation. Upon selecting random generation, the system would create a series of new location names based on existing names of cities.

This was done by first randomly selecting an index within the dataset containing the names of various cities, the first number of characters from this name would then be stored in memory. The algorithm would then find another random index, and compare the last stored character to the character in the same position in the character array (For example, if the memory stored the words “Lond”, a comparison against the city of “Leeds” would return true, as the fourth character is shared), after which the mechanism would pull a number of characters from this word to append to the original memory (In the listed example, this would produce “Londs”). This process would continue until the system reached a blank character or no names that fit the condition were found (after an iteration through the source material starting from the first failed comparison). The produced name would then be stored with an index representing the location it applied to.

While this system had some flaws, the results given were often believable location names that fit the simulation well, as such it is possible that this algorithm may be reworked to fit this artefact – though with better efficiency measures and potential quality checking systems implemented alongside it.

* Books such as guns germs and steel
* Bézier curve for Temperature

1. Artefact Design

As a result of the collated information and comparison in the literature review, the following design plans have been developed to specify how the system itself will function – in particular what features will exist and what algorithms will be used where appropriate.

* + 1. World Generation

The intended algorithm for the world generation was originally the Perlin noise method, but due to the results of the literature review the “Diamond-square algorithm” has been identified as a better fit for the goal of generating a landscape – and will therefore take the place of Perlin noise in this area of the system. This is not to say that Perlin noise will see no use in the project, as review into systems like dwarf fortress have identified the necessity of different “layers” of randomly generated data – it is not enough to simply create a landscape. The following “layers” will need to be produced in any single use of the map generator:

|  |  |  |
| --- | --- | --- |
| **Layer** | **Layer Name** | **Algorithm** |
| 0 | Elevation | Diamond Square |
| 1 | Temperature | Random gradient from the equator |
| 2 | Rainfall | Perlin Noise |
| 3 | Flora | Perlin Noise |

This in turn should make the following adjustments and additional values:

|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | **Name** | **Relevant Factors** | **Description** |
| 0 | Land/Sea Adjustments | Elevation | All values over the 70th percentile of elevation will be defined as land tiles, while all below will be defined as sea. All values relating to the sea locations will be discarded, and any following operations will only occur on the land tiles. |
| 1 | Mountain Adjustment | Elevation, Temperature | The top 20% of elevation values will have their temperature values lowered proportional to their deviation from the 4th quintile elevation. |
| 2 | Forestry | Flora, Rainfall | Locations are given a forestry level based on the overlap between above-average flora values and above-average rainfall values. |

All these collated factors will then be used to specify a biome type for each pixel, using a system by which the locations “fit” value to a biome type is a number between 0 and 1 that is calculated based on the weighting each biome assigns to a specific property. While exact values for weighting are not yet set, the rough impact each property will have on the “fit” value can be seen in the table below. For each property, the difference from the median value for the field is taken as the input and multiplied by the weight (specified here as either high or low) – therefore a negative difference will count as the opposite for the weighting (for example, a biome with negative forestry rating will return a negative number for a highly forested region, but will produce a positive value for a less-forested-than-average region)

|  |  |  |
| --- | --- | --- |
| **Biome** | **Value Weighting** | **Colour** |
| Temperate Forest | Forestry (High Positive Weight)  Temperature (Low Negative Weight) | Dark Green |
| Tropical Forest | Forestry (High Positive Weight)  Rainfall (Low Positive Weight)  Temperature (High Positive Weight) | Very Dark Green |
| Taiga | Forestry (High Positive Weight) Temperature (High Negative Weight) | Light Green |
| Grasslands | Forestry (Low Negative Weight)  Rainfall (High Positive Weight)  Elevation (Low Negative Weight) | Green |
| Savannah | Forestry (Low Negative Weight)  Temperature (Low Positive Weight)  Elevation (Low Negative Weight) | Orange |
| Tundra | Forestry (High Negative Weight)  Temperature (High Negative Weight) | Very Light Green |
| Desert | Forestry (High Negative Weight) Rainfall (High Negative Weight)  Temperature (High Positive Weight) | Yellow |
| Mountain Range | Elevation (High Positive Weight)  Forestry (High Negative Weight) | Grey |
| Forested Plateau | Elevation (High Positive Weight)  Forestry (High Positive Weight) | Green-Grey |
| Shrubland Plateau | Elevation (High Positive Weight)  Temperature (High Positive Weight)  Forestry (Low Positive Weight) | Orange-Grey |

* + 1. Provinces

Once all the geographical information is produced, the world generation algorithm can move towards the next stage of human history – the dispersion of humanity across the world. In this stage, the existing map will be divided into a number of polygons – each representing the geographical boundaries of a population of people, the rough borders at which the residents of a location live their lives.   
  
The first stage of this process is to divide the map into a number of equally sized squares, each of these squares that contain land is then split into two triangles, randomly selecting the direction of the division (from top left to bottom right or from top right to bottom left). Each of these locations is referred to as a “unit” and represents a fraction of a province. Each unit will then store the following information based on the average values of its containing pixels: Elevation, Temperature, Rainfall, Forestry and Biome. All units will then store their percentile Elevation, Temperature, Rainfall and Forestry when compared to the set of all units.

Talk about population density values

Combine multiple units into polygonal provinces – with each polygon attempting to reach a total population density (This means less dense locations will have bigger provinces, more dense locations will have smaller provinces)

1. Bibliography

Adams, T., 2009. *Dwarf Fortress Talk #4* [Interview] (3 November 2009).

Adams, T., 2010. *Nullpointer Dwarf Fortress Interview* [Interview] (1 July 2010).

Adams, T., 2013. *Dwarf Fortress Talk #21* [Interview] (23 May 2013).

Alain Fournier, D. F. C., 1982. Computer rendering of stochastic models. *Communications of the ACM,* 25(6), pp. 371-384.

Dwarf Fortress Wiki, n.d. *Dwarf Fortress Wiki - Calendar.* [Online]   
Available at: https://dwarffortresswiki.org/index.php/DF2014:Calendar

Firaxis, 2010. *Civilization 5 Home Page.* [Online]   
Available at: https://civilization.com/en-GB/civilization-5/

Gorman, J., 2018. *Github - Iron Age.* [Online]   
Available at: https://github.com/JaVonox/Iron\_Age

Lorenz, E., 1972. *Does the flap of a butterfly’s wings in Brazil set off a tornado in Texas?.* Cambridge, Massachusetts, s.n.

Mossmouth, 2013. *Spelunky World.* [Online]   
Available at: https://spelunkyworld.com/

Paradox Development Studio, 2013. *Paradox Plaza.* [Online]   
Available at: https://www.paradoxplaza.com/europa-universalis-iv/EUEU04GSK-MASTER.html

Paradox Development Studio, n.d. *Europa Universalis Wiki.* [Online]   
Available at: https://eu4.paradoxwikis.com/Province

Tarn Adams, Z. A., 2006. *Bay 12 Games.* [Online]   
Available at: http://www.bay12games.com/dwarves/

Valve Corporation, 2011. *Steampowered Store - The Binding of Isaac.* [Online]   
Available at: https://store.steampowered.com/app/113200/The\_Binding\_of\_Isaac/

Valve Corporation, 2014. *Steampowered Store - Europa Universalis IV Conquest of Paradise.* [Online]   
Available at: https://store.steampowered.com/app/241369/Expansion\_\_Europa\_Universalis\_IV\_Conquest\_of\_Paradise/