Atlas of Mystara

PIC2 - Master in Computer Science and Engineering
Instituto Superior Técnico, Universidade de Lisboa

Afonso de Cintra Vinhas Costa Leitão — 96722*
afonso.leitao@tecnico.ulisboa.pt

Advisor: Daniel Gonçalves

Abstract Mystara is a fictional world within tabletop role-playing games, characterized by diverse landscapes, cultures, and narratives. Various maps of Mystara exist, some official and others created by the Mystara community but they lack a unified representation. The "Atlas of Mystara" project aims to create an interactive map for the Mystara region, tapping into an extensive compendium of Mystara content, the Vaults of Pandius and the active Mystara community. Motivated by Mystara's rich history and diverse topography, the project aims to seamlessly blend these elements into a user-friendly, web-based application. The task of creating a unified map faces challenges, including standardizing symbols and addressing variations in representation across diverse maps while managing the scale of Mystara, which is comparable to our own planet. The completed Atlas of Mystara not only fulfills its primary objective of mapping the region but also signifies a pivotal step in advancing our comprehension of Mystara's complex geography and historical narratives. This collaborative effort not only benefits the Mystara community but also lays the groundwork for future contributions and explorations within the realm of fantasy cartography.

Keywords — Hexagons, Tiles, Maps, Web Maps, Interaction

*I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa (https://nape.tecnico.ulisboa.pt/en/apoio-ao-estudante/documentos-importantes/regulamentos-da-universidade-de-lisboa/).

Contents

1 Introduction

In the realm of Dungeons & Dragons [1], Mystara stands as a testament to the boundless imagination of fantasy enthusiasts and tabletop gamers alike. This thesis introduces a web app, the "Atlas of Mystara", which aims to transform diverse handcrafted maps into a cohesive, interactive digital experience. Uniting the works of talented cartographers, this project enables users to explore Mystara's realms, providing a dynamic fusion of fantasy, storytelling, and modern technology.

1.1 Motivation

Dungeons & Dragons (D&D) were the very first role-playing game in the world and will be 50 years old next year. Over time, it has gone through many versions (OD&D, D&D, AD&D). Although the rules were adjusted many times, one thing remains consistent: the existence of campaign settings, worlds in which the players' adventures take place. In this context, one of the very first settings was the Known World [2], later called Mystara. Originally tied to D&D, it was the setting many players in the early 80s adventured in, years before the appearance of more famous ones such as Dragonlance, weirder ones such as Dark Sun, Planescape, or the more mainstream Forgotten Realms.

In the case of the Mystara campaign setting, apart from all the info you got from rulebooks and adventure modules, a series of books, called Gazetteers, was released. These Gazetteers delved into the intricacies of individual countries, geographically, economically, and politically and were all accompanied with hex maps. Sometimes these hex maps changed a bit from product to product other times they were changed dramatically by cataclysmic events as in-game time moved on. But still, looking at one of these is like looking at a "living world".

Elevating the immersive experience for the Mystara community, the concept of a "living world" became particularly pronounced as the setting matured. After one of the cataclysmic events, "Wrath of the Immortals", this immersion reached new heights with the release of annual publications titled "Poor Wizard's Almanacs." These Almanacs meticulously documented all pertinent geo-economic-political developments within a game year (that corresponds to one real-world year).

Although the setting hasn't had official support since WotC [3] bought TSR [4], there is still, to this day, an active community of Mystaran fans. They have produced more Gazetteers, maps, adventures, etc. Which are all are accessible in the big hub for all Mystaran things, the Vaults of Pandius [5].

The focus of this thesis is addressing challenges regarding mapping within the Mystara universe. Notably, inconsistencies among maps, even within official publications, pose a significant problem. This incongruity motivates a community desire for dynamic maps that reflect changes over time, aligning with both publication timelines and in-game events. Another challenge lies in the diverse stylistic and graphic quality across different maps, requiring a standardized approach.

Thorfinn Tait's work in creating digital versions, particularly the Atlas of Mystara [6], serves as a foundational solution. Tait's georeferencing techniques address regional disparities, but gaps persist. These maps only cover a small part of the planet but other cartographers and players have created maps for other regions that are more or less accepted by the community. Building on Tait's approach, this thesis aims to contribute to refining and expanding the Atlas, systematically addressing problems of inconsistency and graphic diversity. Through collaboration with the Mystara community, the goal is to establish a standardized and evolving mapping framework for the Mystara world.

1.2 Work Objectives

This thesis aims to create a fully interactive map that integrates the entire region of Mystara using the available resources provided by many cartographers over the years. To start, the first logical step would be to index all the maps in the Vaults of Pandius mentioned above. As one of the biggest resources available regarding Mystara, if the application covers all the cases in this library, it would be safe to assume that it could correctly interpret and integrate most of the existing maps outside this repertoire.

After indexing, the next step is to parse every map and using various techniques such as image recognition and colour and pattern matching turn simple pictures into something identifiable by a computer.

A mesh of hexagons that transcend their bitmap counterparts and can be manipulated and changed at will. This is not trivial as there are a lot of cases to be considered and the process might not be entirely automatic, since some out-of-bounds and edge cases might need some manual adjustment.

The third and final step of the work is to aggregate and integrate the maps into a web map that's publicly accessible. The purpose of the Atlas is to be a collaborative project in the sense that users can upload their maps and, after a bit of adjusting, see their work added to the World of Mystara.

1.3 Document Structure

We will start with an in-depth analysis of the state of the art, covering the most relevant papers, documents, and applications related to our objectives. Covering the history of board games, tilling systems, and fantasy maps, as well as several attempts at creating applications that present Mystara's world in the palm of the user's hands.

After this brief introduction of the state of the art, a solution will be delineated. Starting with the system's desired architecture and workflow and ending with the introduction of specifics such as which data structures and coordinate systems to use.

Finally an evaluation method for this thesis will be proposed and an expected work schedule will be presented.

2 Related Work

In this section of the document, we will examine the history and legacy of board games, tiling, and coordinate systems as well as explore existing technologies and resources relevant to this thesis.

2.1 War Games and Hexmaps

The Royal Game of Ur (Figure 1) is the oldest playable board game in the world, with its earliest representation being dated to c.2620 BCE. This 4,600-year-old game from Egypt is, for all intents and purposes, the first instance of a board game that presents a square grid of tiles to its players. Nowadays, the square grid is used in most commonly known board games like chess, checkers, scrabble, and Sudoku, but there are other ways of implementing a grid for players to use.

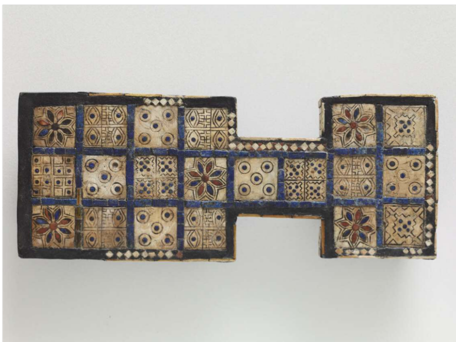

Figure 1: Royal Game of Ur [7]

Agon (Figure 2) (also called Queen's Guards or Royal Guards), first published in 1842, is perhaps the earliest recorded board game using hexagonal cells. The hexagonal grid style then started to be used in certain genres of board games, such as war games, with the second edition of Gettysburg in 1961 and The Settlers of Catan in 1995. Interestingly, games like chess and checkers also have a hexagonal variation which was invented around this time.

Figure 2: Agon [8]

In board games where players have access to maps, it's common to reference directions using cardinal directions (N, S, E, W), and this collides with the idea of hex maps. On a hex map, southwest is not southwest, or at least not without any adjustments. Instead, it is 30 degrees south of west, which is about 15 degrees off (Figure 3). This can have a great impact on a player's experience. For instance, in a Dungeons and Dragons campaign, if players travel a week north, a week east, and then travel back southwest, they're not going to get back to where they started. To work around this issue, it is common to mention the hexes themselves when moving; "I travel a hex east, then a hex northeast, then a hex north" rather than to pretend you don't have access to the convenience of hexes on your map.

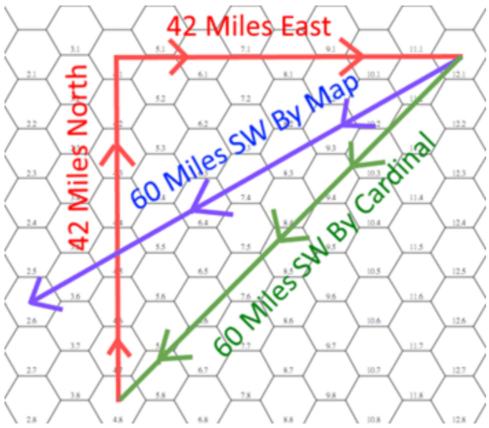

Figure 3: Hexagon-based tile system [9]

There have also been attempts at creating a triangular-based tile system for board games. One example of this is Blokus Trigon (Figure 4), a board game invented in 2006, which is, at its core, Blokus (2000) with a triangular grid instead of a square grid. Here each piece consists of triangles joined by their edges, but you join your pieces together by their corners, with the restriction that each piece must touch another of the same colour on a corner but not on an edge. This means that only two of the pieces that touch at a given intersection may be the same color; while in the triangular version, you have 6 spaces around each intersection and thus can fit three pieces of the same color. In this way, it manages to take advantage of both the triangular and hexagonal properties of the triangular grid.

Figure 4: Blokus Trigon [10]

More complex grid layouts exist (Figure 5) (like the Cairo Tiling or Octagonal Tiling) but won't be discussed in this report since they are not as relevant to the project.

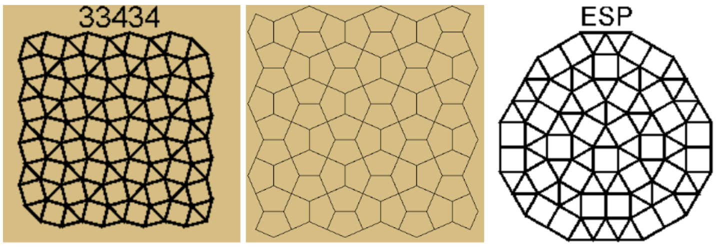

Figure 5: Complex Tilling Systems

To figure out which shape works best for our web map's tiling system, we need to compare its advantages and disadvantages.

2.2 Tilling and Hexagons

Tiling is covering a surface with a pattern of polygons. Some polygons, like pentagons, heptagons, and octagons aren't preferred for tilling as they leave gaps between themselves. These problems could be solved by using 2 kinds of polygons but this increases the complexity of the tilling system, leaving essentially three shapes. Triangles, squares and hexagons. In the case of square tiles, the squares could be infinitely divided into smaller squares and cover an infinitely large plane with infinite parallel lines. The advantages of the square grid are that it maps nicely onto Cartesian coordinates, and the axes of movement are orthogonal to one another, meaning you can move freely on one axis without affecting your position on the other axis. However, on a square board, move horizontally or vertically one space and you've moved one space, but move diagonally and the distance is the square root of two spaces (Figure 7). This means diagonals warp the distance between squares. The advantages of hex grids are that each of its 6 neighbors is connected to all adjacent cells through their edges; hexes never connect through vertices alone and so the distance between the centers of adjacent cells is always the same (Figure 6), which simplifies distance calculations.

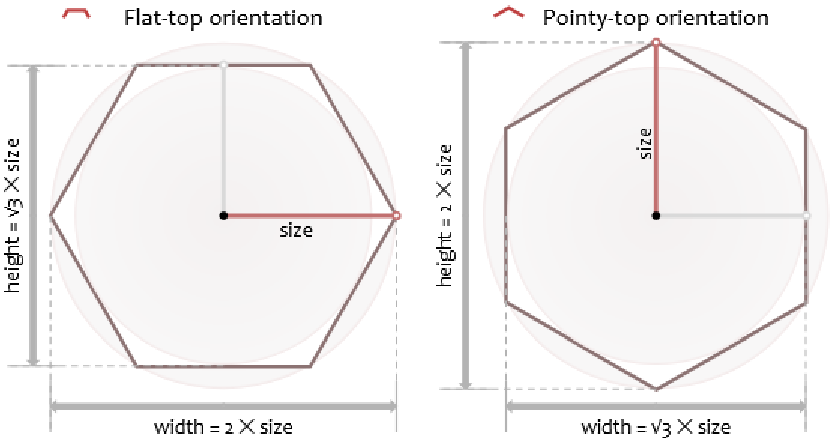

Figure 6: Hexagon orientations [11]

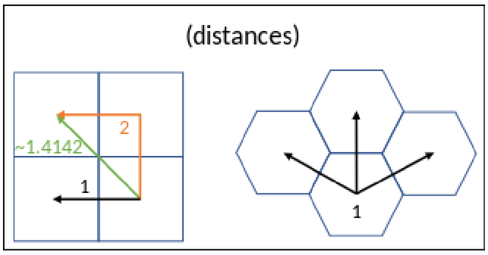

Figure 7: Hexagon orientations [12]

Triangular maps have none of the advantages of hex maps or square grids, or to put it another way, they have all the disadvantages of both.

- Like a hex map, a triangle map does not map nicely onto Cartesian coordinates.
- Like a hex map, the movement axes are not orthogonal. The movement axes aren't even perpendicular to the cell edges.
- Like a square grid, the centers of adjacent cells are different distances apart depending on how they're connected. But unlike a square grid with two different distances (through the edges and the vertices), a triangular grid has three.

2.3 Fantasy Maps

Fantasy maps serve as fundamental tools in tabletop gaming, notably in games like Dungeons & Dragons, the concept of a hex map is intertwined with the concept of the game itself. Each hexagon on the map represents a distinct area with a specific type of terrain, ranging from forests and mountains to deserts and plains. These hexes are essential for spatial navigation and encounter management. To enhance clarity, symbols or glyphs could be added to symbolize cities, landmarks, or other points of interest within each hex. The maps also account for other details such as rivers, roads, and coastlines, contributing to an objective visualization of the fantasy world that approaches the level of detail we see in real-world maps. Whether guiding players through winding paths or depicting the strategic placement of cities, these hex maps provide a concise and structured means of representing expansive and immersive game settings. However, creating a comprehensive world atlas based on fantasy maps, like the "Atlas of Mystara," has faced challenges. Several attempts, with varying limitations, have been made in this endeavor. Some focus on only a small part of the world map or lack true interactivity and control over specific regions. These attempts, although not perfect, offer helpful insights into our objectives, guiding our efforts to develop an interactive fantasy hex map that addresses the community's demands.

2.3.1 42 Rolls of duck tape

42 Rolls of duck tape [13] is a blog created by Lance Duncan, a Geography student at the University of California Northridge in 2017 and it details his work on his capstone project. Its goal was to build a web map and analyze the benefits and drawbacks of such an application, specifically concerning fantasy and RPG maps.

This project's focus is on the Grand Duchy of Karameikos, a small region of Mystara. While limited, Duncan's work can be seen as a proof of concept for our Atlas of Mystara. The country covers an area of approximately 22,260 square miles, an area slightly larger than modern Croatia. Mystara is 6,190 miles in diameter (a 3,095 radius). The total surface area of the planet's outer world, minus the polar openings, is just over 105 million square miles. 60% of which are covered by water, meaning our project would have about 4716 times the magnitude of Duncan's project.

The web map has a couple of limitations besides the area it covers. Firstly, it's very slow (the web page takes about 40 seconds to load), unresponsive, and the panning is not seamless. The perspective is warped (Figure 9) and the map has some visual bugs.

Figure 8: Warped perspective of Lance Duncan's map

Unfortunately, this project does not solve some of our main concerns such as how to handle zoom. It's also important to note that in the hex map (Figure 9), the hexes themselves are not interactive. Instead, Duncan made it so that the icons he placed on top of the map were clickable and the rest of the map is simply a very big bitmap with hexagons.

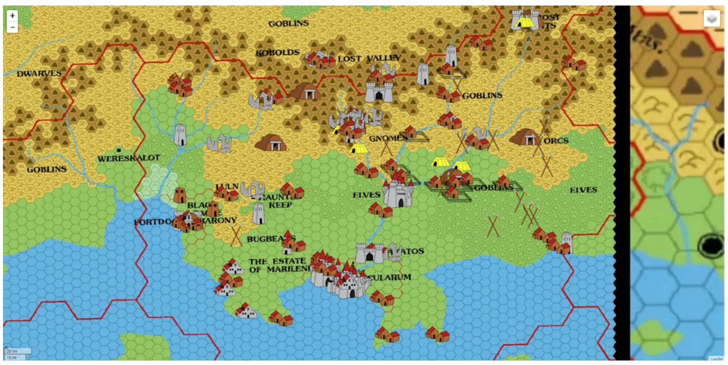

Figure 9: Zoomed-in map of the Grand Duchy of Karameikos

2.3.2 Il Mondo di Mystara

Il Mondo di Mystara [14] is an interactive hex map that divides Mystara's world into big regions (Figure 10). We can zoom in by clicking on a given region and the map updates with the region's map. This solves the issue of subdividing hexagons when zooming by replacing the world map with a region map without any transitions.

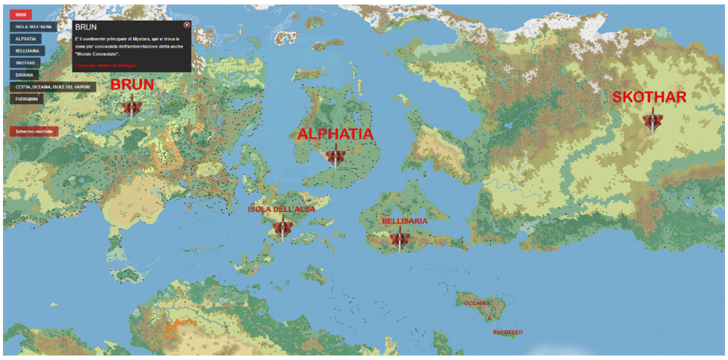

Figure 10: Il Mondo di Mystara [14]

The region's map is also an interactible hex map but this time, instead of zooming, interacting with the map opens a book (Figure 11) covering the region's geography and history.

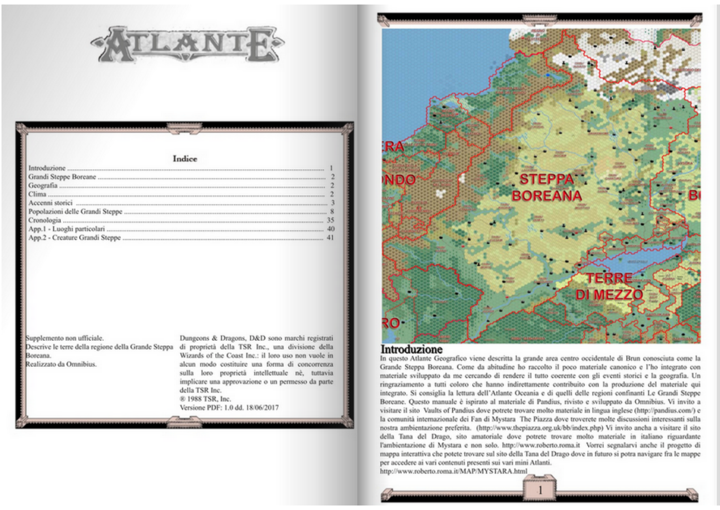

Figure 11: Information regarding a specific region in the map [14]

2.3.3 Mystara map by dkourmyshov

This Mystara map by dkourmyshov [15] uses a Robinson Cylindrical Projection to represent the world of Mystara. This map (Figure 12) is interactible. There are some dots spread out over the map which users can click on to access information regarding that specific place.

Figure 12: Mystara map by dkourmyshov [15]

This project's main limitation is the lack of granularity. It also presents a single bitmap image with interactible dots but the hexagons themselves occupy a very large area and since the zoom function doesn't subdivide the hexagons, a lot of important information regarding the region's topography is lost.

2.3.4 The Mystara's Map Navigator

The Mystara's Map Navigator [16] is a simple map navigator for a small part of the Mystara world. It's very limited in the sense that the map is not interactible and it displays only either the entirety of the region (Figure 13) or a small predetermined portion of it. Navigation is used to move between portions of the map using 4 arrows (each for each of the cardinal points) that change from one region to the next.

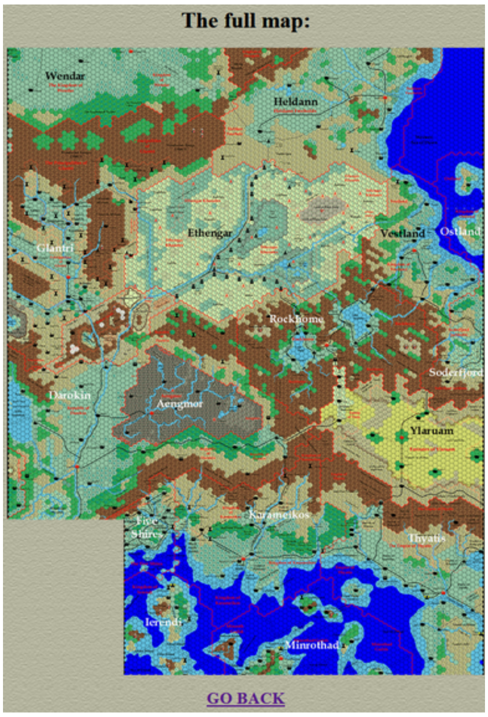

Figure 13: The Mystara's Map Navigator [16]

2.4 Mystara and the Vaults of Pandius

Mystara, much like our world, has gone through several eras. From the Time of the Elemental Lords to the current year, Mystara has suffered a lot of world-changing events that mark the end of an era. These epochs in Mystara's history play a vital role in shaping the cartography of the world. These distinct periods, characterized by significant events such as magical phenomena, cataclysms, and the rise and fall of civilizations, exert a profound influence on the topography. Notably, the early epochs witnessed the impact of Immortals (powerful and eternal beings who have transcended mortal limitations), resulting in the creation of mystical landmarks and enchanted regions. Cataclysmic events have further contributed to the formation of new geographical features, causing substantial alterations to the landscape. Consequently, Mystara's cartography serves as a visual archive of its historical epochs, offering a dynamic portrayal of the world's evolution and establishing a tangible connection between narrative developments and the physical layout within the D&D universe.

Even though it is similar to our planet, Mystara's topography diverges from Earth in several notable ways (Figure 14). One distinctive feature is the Hollow World, an interior space within the planet that harbors entire civilizations. Instead of Earth's polar ice caps, Mystara features entrances to the Hollow World at both poles. Additionally, Mystara incorporates more geological anomalies such as floating continents known as the Skyshield. While both worlds share a spherical shape, these unique features make the Mystara globe challenging to map.

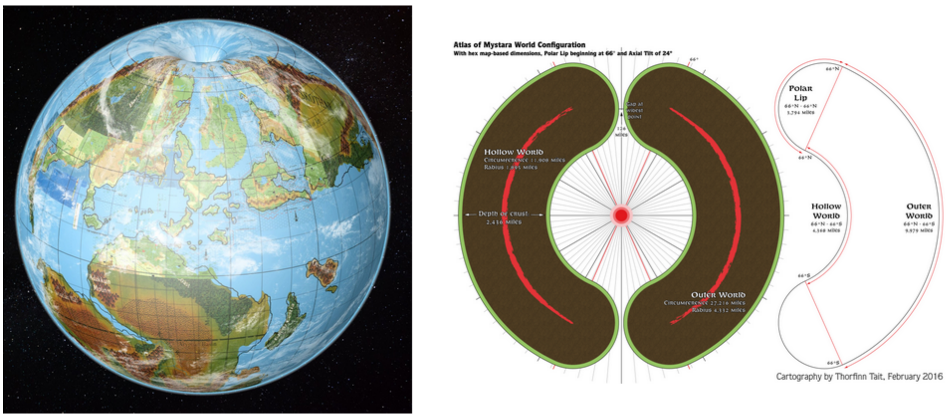

Figure 14: Mystara's shape [6]

When exploring the intricacies of Mystara's history and topography, it is imperative to mention the Vaults of Pandius [5]. Functioning as a comprehensive online compendium, the Vaults of Pandius serve as an invaluable resource for those seeking an in-depth understanding of Mystara's lore and cartography. This digital repository encapsulates the epochs, events, and geographical nuances that define the world.

The Vaults of Pandius have organically evolved, becoming a dynamic repository of knowledge with active contributions from the Mystara community akin to a living archive. The expansive insights found within the Vaults are not only the result of meticulous curation but also a testament to the engagement of the community and a landmark to the level of detail this world has to offer.

2.5 Hex Map Tools and Libraries

Some attempts have been made to create an Atlas of Mystara, but more work exists in the realm of hex maps. In this section, we will be discussing other relevant hexagon-based applications as their efforts to create a user-friendly intuitive experience for map navigation and editing will be greatly appreciated for our work.

2.5.1 HEXTML

HEXTML [17] is a hex map editor specifically made for D&D campaigns. It has many interesting features that would be very useful to a Dungeon Master (DM) such as connecting maps via a sub-maps menu, adding secret notes only visible to the DM, and allowing for collaboration both for help building the map itself and for other players to join in and play the campaign directly on the website.

As for editing the hexmaps, it has tiles of various colours each with different names, allowing users to add their tile set if they choose to do so. It has a stamp system that lets you stamp out tiles with other tiles and/or symbols. It also allows users to override the background colors of the hexes by using the color menu. It's possible to manually add rivers, roads, and borders to all hexes. The map editor also has a fog of war feature that can hide certain tiles from the players until the DM desires to do so.

The hexes themselves can have information in them. They each have three menus where it's possible to input text which is then saved in the hex itself.

The map also has the option of using three different grid shapes. Two of them are hexagon grids with the hexagons aligned in different ways and the last one is a square grid. There is a feature to export the map to an image. I've made a very simple map (Figure 15) that covers most of the features of this editor (presented below) and found out the map-to-image is a bit flawed as it cuts the right outer edge of the map. The exported map doesn't include the borders, the rivers, and the roads I've added and the hexes have a thicker border than in the editor.

Figure 15: Image exported from HEXTML

2.5.2 Molotov Cockatiel Hex Map Maker

Molotov Cockatiel [18] presents a simpler approach to hex map making. It asks the user to prompt out the map size and generates it for him letting the user paint over the generated map with solid color-only tiles. This tool can export the map to SVG and an "array of hexes". This "array of hexes" meaning a list of lists containing numbers from one to eight which were associated with each of the possible hexes the editor used (with one being the clear tile and eight being the custom tile), meaning there was no difference between two different custom tiles and the amount of information it saves is very little.

2.5.3 Worldographer / Hexographer II

I tested Worldographer [19] on a free version. This editor (Figure 16) presents a full-fledged application with a lot more features and detail than the ones we have discussed so far. It has the feature of generating its maps with options for generating rivers, roads, nations, and more. It's possible to fully customize any given tile with numerous options for transparency, texture, color, etc.

In this editor, hexes have two numbers associated with them, one for the row and the other for the column, making it easy to locate a hex by its coordinates.

Figure 16: Worldographer's main interface [19]

Zooming in and out of a very large map is one of the biggest challenges we'll face when making the Atlas. This editor solves this problem by subdividing or grouping hexagons according to a zoom level that goes from "village" to "kingdom". Refining hexagons is not trivial as there are lots of methods to do so. In the paper Hexagonal Connectivity Maps for Digital Earth [20], the authors describe six types of hexagonal refinement (Figure 17). 1-to-3, 1-to-4, and 1-to-7 refinement in both their centroid-aligned (c-refinements), represented in orange and vertex-aligned (v-refinements) represented in red. Centroid-aligned refinements produce refined cells sharing centroids with coarse cells while in vertex-aligned refinements the refined cells' vertices are shared with the centroid of coarse cells. Worldographer uses centroid-aligned refinement methods for zooming with varying degrees of refinement which the user provides in the form of zoom levels.

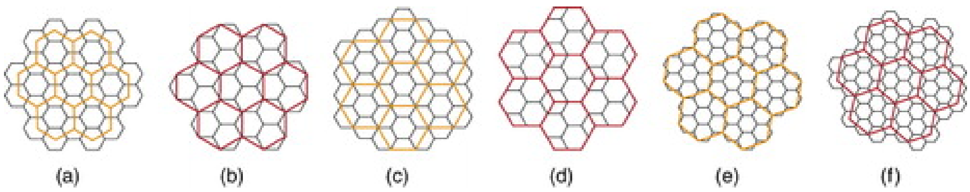

Figure 17: Different types of refinement for hexagons [20]

As for zooming in further, Worldographer has (similarly to the first example) map links that increase the level of detail. These maps are often used as town or village maps (Figure 18). These aren't made from premade tile sets but are essentially a picture with a uniform hex grid on top.

Figure 18: Worldographer submap example

2.5.4 Digitally representing the Earth

The H3 library [21], originally developed by Uber, has emerged as one of the leading options for discrete global grids. While other grid or grid-like systems are in use, including Google's S2, Quadkeys, and more, H3 is one of the few that uses hexagons as the grid's standard unit of measure (Figure 19).

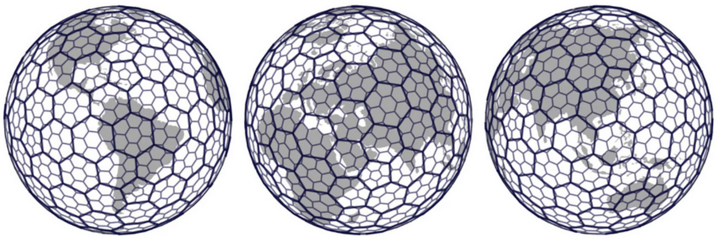

Figure 19: Representation of H3 grid on a planet scale [21]

It achieves this by creating a hierarchy of grids based on resolution levels. Each hexagon is part of one out of 16 possible resolutions from 0 (the size of a small continent) to 15 (less than a square meter). Each hexagon then has associated with it a parent and children hexagons, each one resolution level away from their own. When zooming in, H3 uses its centroid-based refinement method (Figure 17) with a specific angle so that subdivided hexagons occupy roughly the same area as its parent (Figure 20).

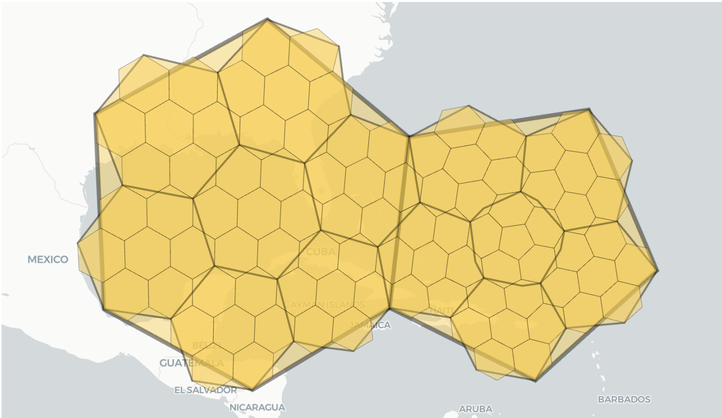

Figure 20: H3's hexagon refinement method [22]

It's important to note that, using the library, not all hexagons have the same size and shape (although in actuality they amount to the same area); this is to accommodate for the dymaxion projection used by the library which introduces some distortion. This projection places the world map onto the surface of an icosahedron (a convex polyhedron with 20 faces), which can be unfolded and flattened to two dimensions. This projection is used to limit shape and size distortion that is added after mapping the polyhedrons to spheres (Figure 21). Additionally, all the vertices of the projection (where distortion is greatest) are over ocean areas, further enhancing its accuracy. Other shapes like cubes (used by Google's S2) could be used but these introduce more distortion than an icosahedron when mapping to a sphere.

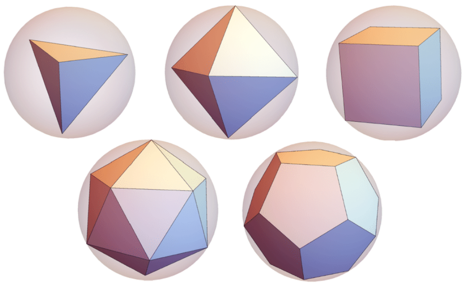

Figure 21: Mapping 3D shapes to spheres [23]

More attempts at digitally representing the Earth exist; some of which are covered in the paper A Survey of Digital Earth [24]. In this paper, a collection of Discrete Global GridSystems (DGGSs) is presented and analyzed. Amongst these lies the Icosahedral Snyder Equal Area Aperture 3 Hexagon (ISEA3H) (Figure 22) which is very similar to Uber's approach to this problem. It uses a 1-to-3 centroid refinement method (Figure 17) over an icosahedron. This DGGSs is used in a library similar to H3 called geogrid which can be used to create maps in JavaScript. The ISEA3H approach is widely used and mentioned in the literature, with several attempts at improving the algorithm with different coordinate systems. With ISEA3H, Quadrilateral 2-Dimensional Integer (Q2DI) indexing is typically used. Q2DI partitions the Earth's surface into 12 quad tiles and has (i, j) coordinates on a certain quad for each cell at each level.

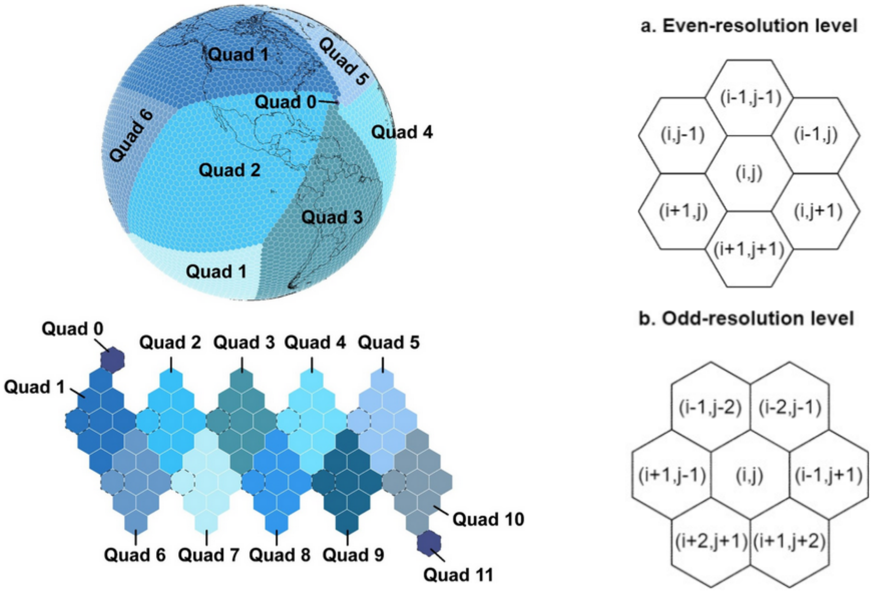

Figure 22: Earth's division in quads based on ISEA3H [24]

In this case, a double-axis coordinate system is used but more coordinate systems exist, each with their advantages and disadvantages.

2.5.5 Coordinate Systems

The main coordinate systems used for hexagonal grids are offset coordinates (Figure 23), cube coordinates (Figure 24), and axial coordinates (Figure 24). Sometimes, a spiral coordinate system is used to map hexagons but these work best on maps with a fixed size which although could technically be used in our atlas (since the size of the Mystara doesn't change) could lead to complications later on while subdividing the hex maps due to the complexity of this coordinate system.

The most used intuitive method is the offset coordinate system (Figure 23) (used in the Q2DI indexing method mentioned above) offsets every other column or row. This is the system most used for square grids but applied to hexagonal ones. It's also possible to use the "true" coordinates of the hexagons' center using a variation of the indexing of this system.

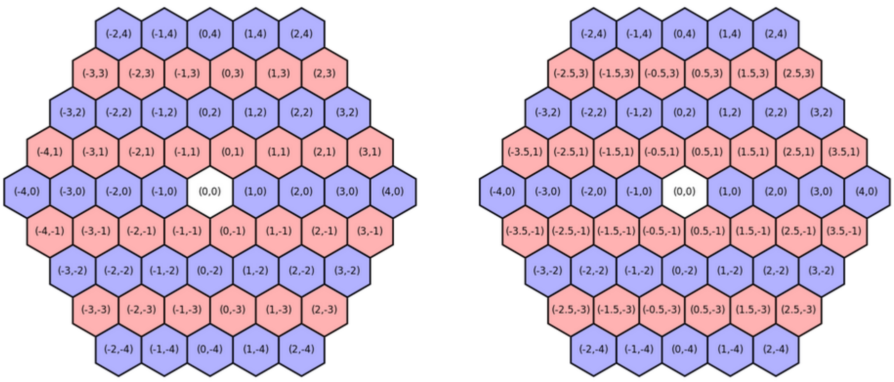

Figure 23: Offset coordinate system for hexagons [25]

Instead of using two orthogonal axes like in the offset system, we can use three, (x, y, z) given the hexagon's nature, thus using a cube coordinate system (Figure 24).

The cube coordinates are a reasonable choice for a hex grid coordinate system. The constraint is that x + y + z = 0 so every algorithm must preserve that. The constraint also ensures that there's a canonical coordinate for each hex.

The axial coordinate system (Figure 24), sometimes called "trapezoidal", "oblique" or "skewed", is similar to the cube system except the z coordinate is not used. Since we have a constraint x + y + z = 0, we can calculate z = -x-y when we need it. The axial/cube system allows us to add, subtract, multiply, and divide easily with hex coordinates. The offset coordinate systems do not allow this, and that's part of what makes algorithms simpler with axial/cube coordinates.

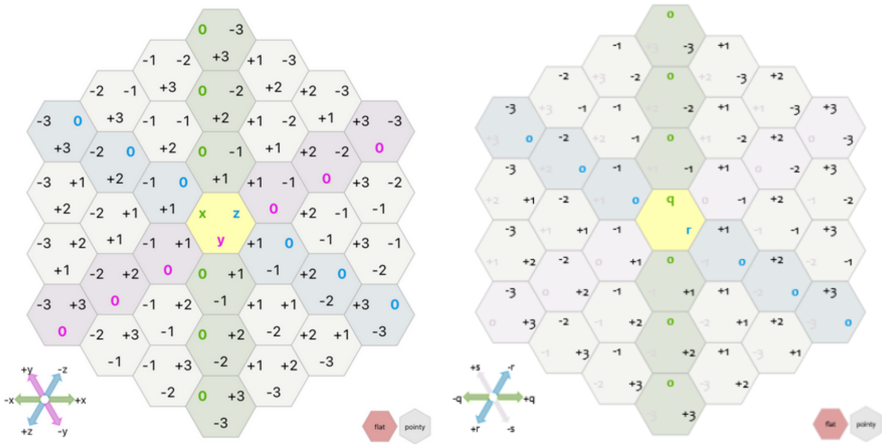

Figure 24: Axial and cube coordinate systems for hexagons [11]

These coordinate systems have been further researched in other papers such as Implementing Hexmap Generation Framework using Cube Coordinate System in Unity3D [26], which compares the cube with the offset system and proceeds with the ladder since it is more straightforward to reason and explain. The cube system, while having a more complex way of defining each hex, is more elegant and better fitted for hex grids, since it represents the laws (e.g directions and relative positioning) of hexagonal grids more naturally, which in turn simplifies most of the algorithms for operations defined on hexagonal grids.

3 Solution

In the following section, a solution to the problem will be presented. Referencing the system architecture 32 and the used coordinate systems as well as data structures and other relevant topics. A GitHub repository [27] of "Proof of Concept" programs has been made available with some demonstrations of the principles mentioned here.

3.1 System Behaviour

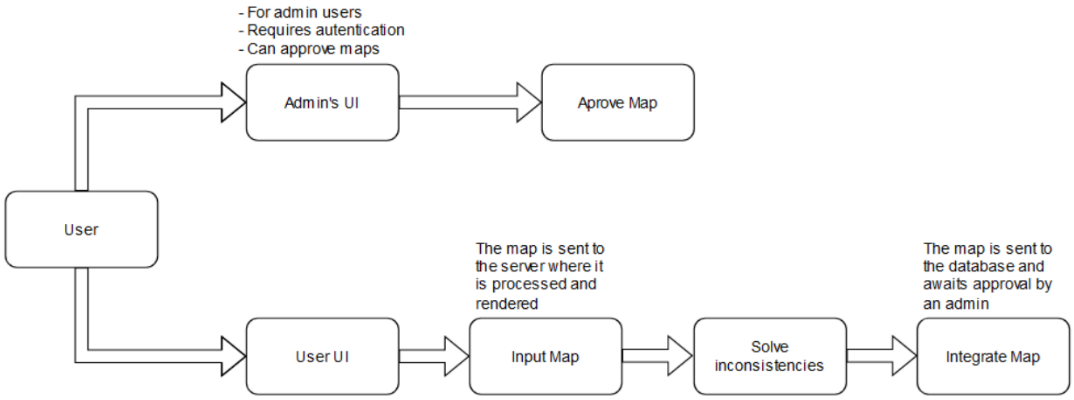

Figure 25: System Behaviour

The system can have essentially two kinds of users with access to different operations (Figure 25). Regular users don't require credentials to access the application's main functionalities but can only observe and upload new maps to the web server.

Admin Users need authentication to log in and can approve maps uploaded by regular users. When a user decides to upload a new map to our system it first needs to be processed by our algorithm, essentially turning a simple bitmap image into an interactive part of our Atlas. The processing system uses image recognition and the user's help to differentiate the tiles in the uploaded map and integrate it with the existing maps. Users have the option to add their custom tilesets which will be added (after approval) to the database. These tilesets will be used to identify tiles in maps. After all the tweaks are made, the user helps integrate the map with the Atlas by placing it in the appropriate part of the world. Maps are then sent to the server and saved in a database, awaiting approval by an admin.

Maps uploaded by regular users are flagged as "not approved" maps in the database and as such aren't displayed to other regular users when using the application. These maps are moved to the "approved maps" pool once approved by an admin. This ensures the Atlas' consistency. An admin can choose to accept or deny the addition of the map to the database and can also edit the said map if he so desires.

The map rendering tool takes all the maps from the "approved maps" pool in the database, combines them into a big fully interactive map, and displays it to the user. This map allows for zooming, panning, and interacting with its tiles to reveal information on the selected area.

3.2 Map Importing Pipeline

The process of turning a bitmap image into a fully interactive tile of hexagons is not trivial and therefore it's necessary to describe the processing pipeline for pre-processing the said image 26.

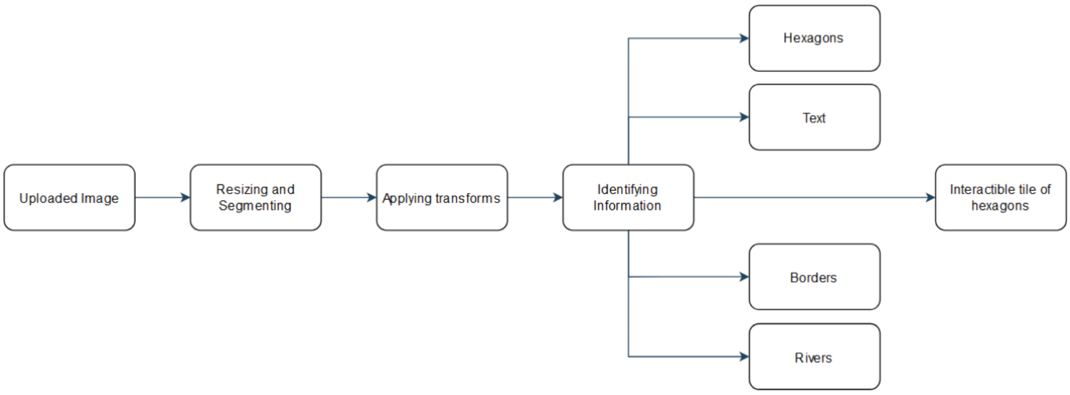

Figure 26: Map Importing Pipeline

The uploaded image first goes through a resizing and segmentation process where the original image is divided into smaller ones to facilitate hexagon identification. This step is needed due to the resolution of some maps that when segmented are easier to process. If the image contains a legend, it will be identified in this step, compared to the ones in our database, and added, if it refers to a new tile set. An algorithm for identifying different hexagons in a legend and associating their image to the label in the legend already exists in the GitHub repository (PoC/Label Processor/LabelProcessorTesseract.py). After this step, several transforms are applied to the image which helps in identifying contours and color contrasts between the hexagons.

In the "Identifying Information" section of the pipeline, information is extracted from the image to a format that we can use to create interaction with the hexagons. It identifies different kinds of hexagons. Take for instance the following map of Karameikos made by Thorfinn Tait 27 at a resolution level of 6 miles per hex.

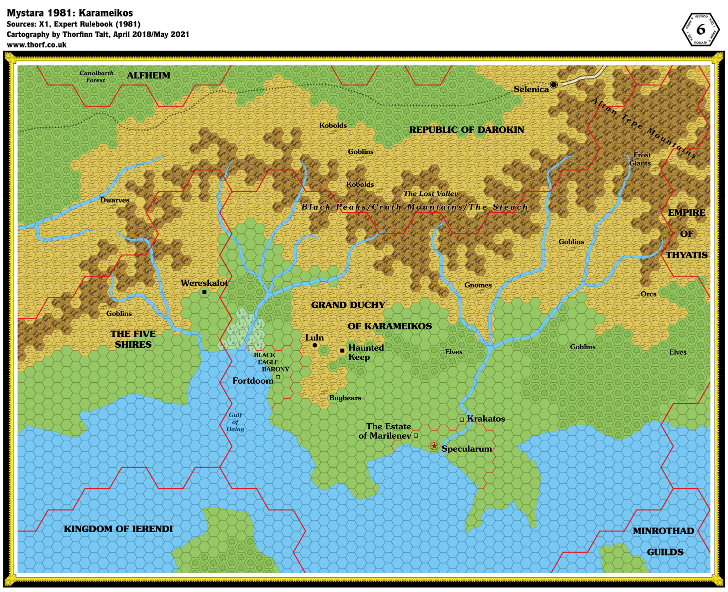

Figure 27: Karameikos, 6 miles per hex by Thorfinn Tait [6]

If input, the map would first go through a resizing process (Figure 28). Since the map does not contain a legend, it won't process it and will instead compare each hexagon found with the ones in our database.

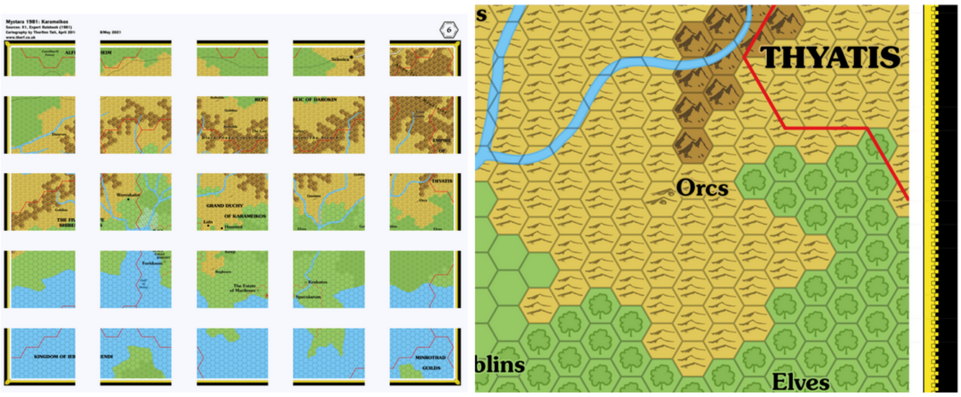

Figure 28: Karameikos' resizing process

For each section, the algorithm will then try to identify existing text, borders, rivers, and types of hexagons after applying some transforms. In figure 29 we can see this information being identified. Finally, we use all this extracted information to build a uniform grid of intractable hexagons using our preferred tilling system. One relevant aspect of this extracted information is the concept of layers. Mainly these layers consist of a baselayer, a layer with all hexagons, a layer with coastlines, and a layer for glyphs and text. These layers will be important to accurately render the map to the user.

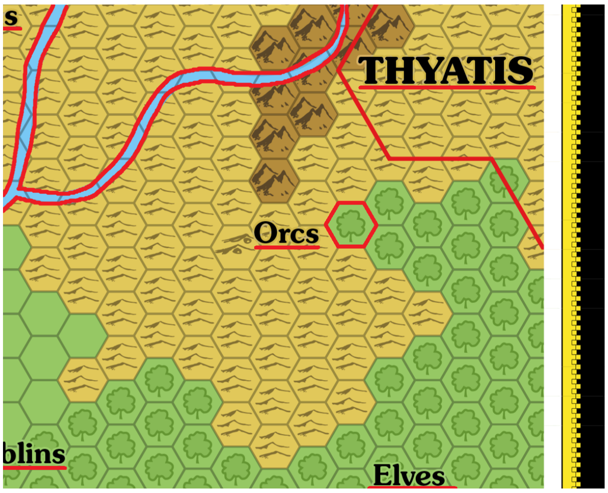

Figure 29: Identifying information

3.3 Relevant Maps

The Vaults of Pandius are the biggest known resource of Mystara content. As such it will be used as a staple of what is a relevant map. The Atlas of Mystara should therefore allow for the integration of maps similar to the ones in the Vault. Currently, 431 maps were found on the Vaults' main page using a Python indexer (File in repository: PoC/Indexer/indexer.py). Of the indexed maps, the most relevant are the ones made by the cartographer Thorfinn Tait, one of the most renowned cartographers of Mystara. Most other cartographers follow his norms and techniques so covering all of Thorfinn's maps will mean reasonable completude over most maps. There are however other relevant cartographers whose contribution can't be neglected such as Thibault Sarlat and Dennis Kauth. Fortunately, Thorfinn as well as many other cartographers have made their legends, containing the respective author's tile set, publicly available [28] making this a suitable base for our tests.

Thorfinn Tait's maps usually follow the same principles and use Tait's tileset (some examples of Tait's maps are in the GitHub repository at PoC/Database/ThorfMaps). The biggest issue with these maps is the coastlines and other contours that don't follow the hexagons' border and instead reside inside the hexagon itself. This is a problem that exists in many maps 30 that deal with water bodies in general as well as territorial borders that divide cities, countries, or continents. The Atlas of Mystara should be able to account for these details and transcribe the map correctly to its intractable counterpart.

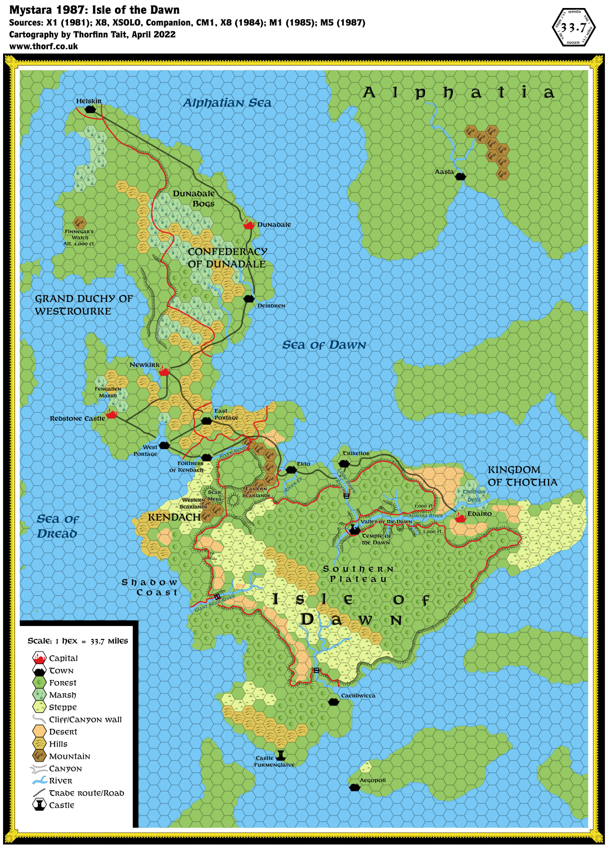

Figure 30: Isle of the Dawn by Thorfinn Tait [6]

As previously mentioned, different cartographers use different tile sets that are often shared amongst all maps of said cartographer (Figure 31). The Atlas should account for this by accepting maps with different tile sets and integrating them. This unification will be based on the type of hexagon in question. This means all hexagons of a specific type will look the same in the Atlas even though the original image's tile was from a different tileset. This approach also means users could be able to change the base tileset of the Atlas and use their own if it covers all of the types of hexagonal tiles that exist.

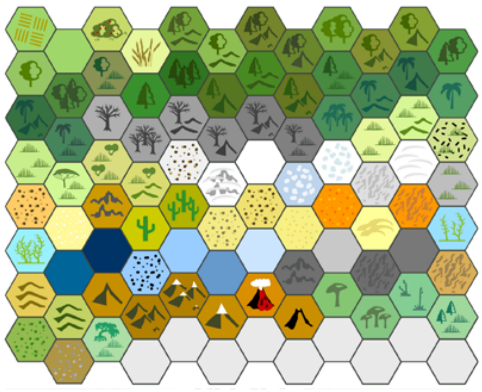

Figure 31: Different hexagonal tiles

It's also important to note the existence of resolution levels for maps in the Vaults. Like with real maps, Mystara's maps usually follow a resolution level to scale the specific map to the world's proportions. In the image 27, we see the annotation of "6 miles per hex" that gives us this information. All maps made by Thorfinn Tait, as most maps in the Vaults, share this property which will be crucial to then integrate these maps in the Atlas. Maps can have several resolution levels. The smallest maps can have resolutions of less 1 mile per hex (example in the repository: PoC/Database/ThorfMaps/cm5-wilderness-0.167.png) and the biggest can surpass resolutions of 100 miles per hex (example in the repository: PoC/Database/ThorfMaps/m1-norwold-150.png). In the final application, our Atlas of Mystara will display all these maps integrated as one with the possibility of showing all maps of the same region, even those with different resolution levels as if it consisted only of a single map.

3.4 Backend

The Atlas' backend will be responsible for handling the web server's requests and all the required operations for map handling (adding, rendering, processing, and approving the maps). It will be written in Python as this problem's solution could be abstracted into a library for hexmaps that will be publicly available for future projects. As such the library used to manipulate the hex maps will be homebrew since there are not many options for the job in most programming languages that suit our needs. The application will present a basic client-server architecture with a database that stores all necessary information about our Atlas such as our hexagonal tiles' size, type, and position (Figure 32).

Figure 32: Client-server architecture of the app

Hexagons will be mapped using an existing coordinate system, as discussed before. Since we are going to use non-rectangular maps that can be rotated, the offset system is not ideal so we will use the axial or cube coordinate system. The difference again is that in the axial system, there's the omission of one axis that can be calculated using the others ($z = -x - y$).

Besides this coordinate system, it is also important to keep in mind the latitude and longitude of hexagons. Some maps aren't perfectly integrated but should still be present in our atlas. Having a separate coordinate system that is absolute will help render these maps in their appropriate positions.

As we have seen before, hexagons are objectively the best shape to use for flat maps where the distances between tiles are relevant. This is not the case for maps drawn over spheres since it is impossible to cover a sphere using only hexagons. The angles of a hexagon do not add up to 360 degrees, which is a requirement to form a closed, curved surface like a sphere. This is possible with other shapes like pentagons that do add up to 360 degrees, allowing them to form a closed surface. Even though this is the case, there are workarounds to this problem like we have seen before with, for example, Uber's H3 library. Hexagons will still be used to pay homage to the fantasy map setting and to facilitate the integration of the existing maps in the Vaults of Pandius and other relevant map repositories.

3.5 Data Structure

Defining the relevant data of the hexagon structure is not trivial and depends on the chosen implementation. Various fields can and will be added later to improve performance, solve inconsistencies, and facilitate operations. For now, a basic structure that covers most existing cases in the Vaults of Pandius can be stipulated using the available maps such as the ones mentioned in section 3.3 Relevant Maps. The structure's attributes are presented below.

- Map ID
- Hexagon ID
- Relative position in the current map

  i.e: [42,73]
- Absolute position in the world (latitude and longitude)
- Type

  i.e: "Jungle", "Mountain", "Unkowkn"
- Layer

  i.e: "Hex", "Coastline", "Glyph"
- Orientation

  i.e: "Flat" (0º), "Pointy" (30º)
- Size
- Borders
- Content (Links to information regarding the specific region the hexagon is in)
- Date of creation
- Date of last known update

To find out the ideal data structure for transferring the hexagon tiles, several serializations were attempted using a basic hypothetical JSON file as a model. The JSON file and the tests are available in the GitHub repository (data/DataStructures/DataStructureTest.py). According to the paper "A Literature Review on Device-to-Device Data Exchange Formats for IoT Applications", the most pertinent data interchange formats used are, JSON, BSON, XML, CSV, Protobuf, and YAML. A simple test was conducted that translated a mock JSON file to all these formats and compared their size. The results are presented below.

- JSON Size: 394 bytes
- XML Size: 514 bytes
- YAML Size: 329 bytes
- CSV Size: 202 bytes

- ProtoBuf Size: 146 bytes
- MessagePack Size: 290 bytes
- BSON Size: 410 bytes

One of the most promising formats in this test was ProtoBuf, a format originally developed by Google for use in their internal systems. This format's main downside is that it is not easily readable by humans and is therefore not ideal for debugging. Nevertheless, in our case, it might prove to be very useful as it presents a similar data structure to a JSON but with a very low total size since it conveys binary data. ProtoBuf, as well as the other formats, can also be simply transformed into a JSON for debugging if needed. This format can then be translated into SQL and integrated into our database for ease of access after being received.

4 Evaluation

The Atlas of Mystara application will be essentially divided into 3 parts each with its own purpose and evaluation method.

The automatic parsing of the maps will be evaluated on its completeness and accuracy. This tool should be able to parse most maps with the least possible amount of user input. For this part of the Atlas, we are aiming at 95% of the maps in the Atlas (which corresponds to more than 400 maps) to have less than 5% of the hexes corrected by the user due to uncertainty. These hexes will be flagged by the algorithm so the process of testing this part of the Atlas will be automatic.

The second part of the tool will be related to the correction of the inputted maps and integration with the rest of the Atlas. This is a process that will require the user's input and as such will be evaluated via user tests. Users will be asked to compare the given map with the one outputted by the parsing algorithm and solve the existing inconsistencies. They will then be asked to add their map to the correct position relative to the existing maps and submit them for approval. The admin's interface will also need testing. An admin should verify if the submitted map is valid and is correctly integrated into the Atlas. If not, this should be corrected and then finally integrated into the Atlas for all users to see.

After the map is fully integrated and approved by an admin, the last part of the application will be tested. Consulting and navigating the map. This component of the Atlas will also be evaluated via user tests. Users will be asked to find information on specific regions of the map by navigating it and selecting the appropriate location.

5 Work Schedule

This thesis follows a defined work schedule depicted in the accompanying Gantt chart 33. The chart outlines stages such as application development, user testing, and dissertation writing. In the initial development phase, an automatic parsing method for maps will be created. Simultaneously, work will commence on an alpha version of the map editing tool. Following this, the alpha version of the map integration tool will be developed. Subsequent tasks encompass the final evaluation of the beta version, the completion of the map importing tool, and the refinement of both the alpha and beta versions of the application. Proceeding into the user testing phase, the final version of the application will be introduced, followed by the start of the testing phase. The final segment of the project involves writing the dissertation which after a final revision by the advisor, will lead to the delivery of the completed dissertation and extended abstract, thus concluding the work.

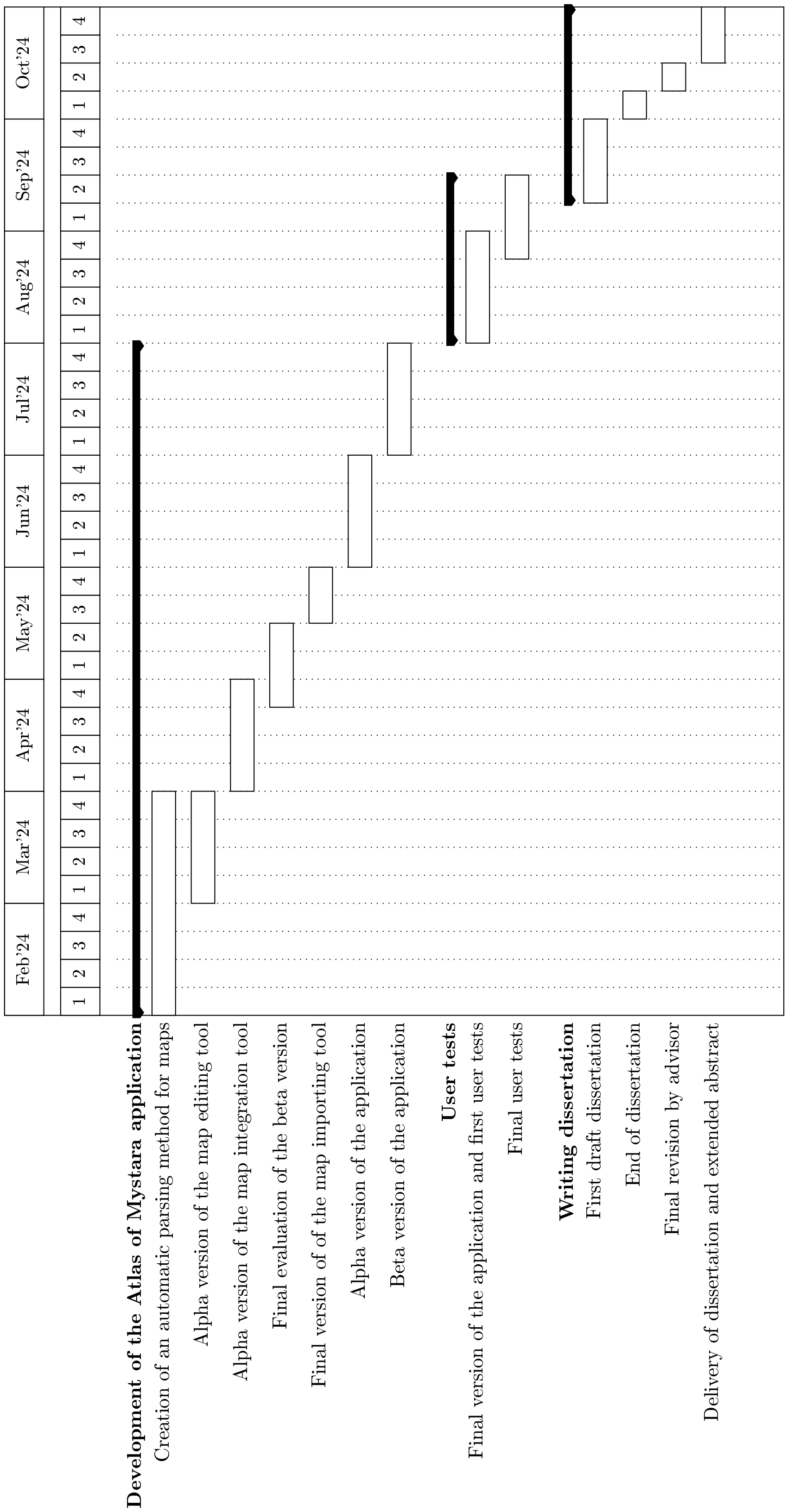

Figure 33: Planned Schedule

Bibliography

[1] D. . Dragons, https://dnd.wizards.com/, accessed: 2023-12-31.

[2] "Known world," http://www.pandius.com/knwnwrld.html, accessed: 2023-12-7.

[3] "Wizzards of the coast," https://company.wizards.com/en, accessed: 2023-12-20.

[4] "Tactical studies rules (tsr)," https://en.wikipedia.org/wiki/TSR,_Inc., accessed: 2023-12-27.

[5] "Vaults of pandius," http://pandius.com/, accessed: 2023-12-28.

[6] "Atlas of mystara," https://mystara.thorfmaps.com/, accessed: 2023-01-06.

[7] "The royal game of ur," https://www.britishmuseum.org/collection/object/W_1928-1009-378, accessed: 2023-10-25.

[8] "Agon," http://www.cyningstan.com/game/126/agon, accessed: 2023-10-25.

[9] "Wait... why exactly do we use hex maps?" https://gnomestew.com/wait-why-exactly-do-we-use-hex-maps/, accessed: 2023-10-25.

[10] "Blokus trigon board game review and rules," https://www.geekyhobbies.com/blokus-trigon-board-game-review-and-rules/, accessed: 2023-10-25.

[11] "Hexagonal grids," https://www.redblobgames.com/grids/hexagons/, accessed: 2023-10-30.

[12] "Hex map," https://en.wikipedia.org/wiki/Hex_map, accessed: 2023-10-30.

[13] "42 rolls of duck tape," https://42ducktape.blogspot.com/, accessed: 2023-11-27.

[14] "Ill mondo di mystara," http://www.roberto.roma.it/map/MYSTARA.html, accessed: 2023-12-24.

[15] "Mystara map by dkoumyshov," http://dkourmyshov.github.io/mystaramap/, accessed: 2023-12-30.

[16] "Mystara map navigator," http://pandius.com/navigator/index.htm, accessed: 2023-12-22.

[17] "Hextml," https://hextml.playest.net/, accessed: 2023-12-20.

[18] "Molotov cockatiel hex map maker," https://molotovcockatiel.com/hex-map-maker/, accessed: 2023-12-20.

[19] "Worldographer," https://worldographer.com/, accessed: 2023-11-5.

[20] E. H. Ali Mahdavi-Amiri and F. Samavati, "Hexagonal connectivity maps for digital earth," *International Journal of Digital Earth*, vol. 8, no. 9, pp. 750–769, 2015. [Online]. Available: https://doi.org/10.1080/17538947.2014.927597

[21] "H3: Uber's hexagonal hierarchical spatial index," https://www.uber.com/en-PT/blog/h3/, accessed: 2023-11-30.

[22] "Tables of cell statistics across resolutions," https://h3geo.org/docs/core-library/restable/, accessed: 2023-11-30.

[23] A. Tavakoli and N. Gisin, "The platonic solids and fundamental tests of quantum mechanics," *Quantum*, vol. 4, p. 293, 07 2020.

[24] A. Mahdavi-Amiri, T. Alderson, and F. Samavati, "A survey of digital earth," *Computers Graphics*, vol. 53, 09 2015.

[25] "Hex grids and cube coordinates," https://backdrifting.net/post/064_hex_grids, accessed: 2023-11-31.

[26] T. Martyniuk, "Implementing hexmap generation framework using cube coordinate system in unity3d," *Computers Graphics*, 2021. [Online]. Available: https://ekmair.ukma.edu.ua/handle/123456789/22404

[27] "Atlasofmystara: Github repository," https://github.com/acvcleitao/AtlasOfMystara, accessed: 2023-01-09.

[28] "Legends," https://mystara.thorfmaps.com/tag/legends/, accessed: 2024-01-07.