Navigare necesse; vivere non est necesse*

* Sailing is precise; living is not precise

Paulo Roma Cavalcanti
Department of Computer Science
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil
roma@dcc.ufrj.br

***Abstract*—This work describes Navigare, an application that employs Computer Graphics to present Geomatics concepts to the general public. Its interface is composed of a 3D globe and a 2D map, both of which are synchronized to display the same location. A set of predefined historical sites is presented, and users can select two locations to trace a loxodrome or orthodrome path between them or calculate their distances. The application also provides information about each historical site, including its name, country, longitude, latitude, and a brief description of its significance. A Timeline of historical figures is also available, allowing users to explore the lives and contributions of notable individuals throughout history.**

***Index Terms*—Geomatics, Computer Graphics, Mercator, Loxodromes, Orthodromes, Geodesics**

I. Introduction [1]

This work describes Navigare, an application that employs Computer Graphics to present Geomatics concepts to the general public. Its interface is composed of a 3D globe and a 2D map, both of which are synchronized to display the same location. A set of predefined historical sites is presented, and users can select two locations to trace a loxodrome or orthodrome path between them or calculate their distances. The application also provides information about each historical site, including its name, country, longitude, latitude, and a brief description of its significance. Navigare is open source and offers the following functionalities:

1) Equirectangular and Mercator projection viewer using lighting combined with texture mapping written in Vanilla JavaScript and WebGL.
2) Auto Rotation: intrinsic (around local axes) or extrinsic (around global axes); around X-axis, Y-axis (parallel), Z-axis or q-meridian.
3) Arcball Paradigm: natural rotation using the mouse or touch and necessary to display the historical tooltips (globe should stay still).
4) Loxodrome and orthodrome tracer: Draw paths on the map and globe when the user selects two points (by clicking either on the map or globe) to get their GCS coordinates (Geographic Coordinate System) [2] (longitude and latitude). The bearing angle (azimuth) is measured from the previous to the current location. Loxodrome checkbox must be checked in the interface.
5) Around the World in 500 Historical Figures: presents a summary of each location visited by using the arrow keys or clicking on the bottom right or left of the globe canvas (to advance or recede).
6) Dial Earth: type a pair of latitude and longitude coordinates and press enter to go to the selected destination on the map and globe; type a string with more than three characters to get a list of matched sites. Additionally, press the [≈Site] button to go to the closest historical site.
7) Mobile Friendly: responsive interface design for running on smartphones and tablets.

WebGL [3] is a JavaScript API for rendering interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins. It is based on OpenGL ES 2.0 and provides a low-level interface for rendering graphics using the GPU. The Graphics Processing Unit is a specialized processor designed to handle complex mathematical calculations required for rendering graphics. Basically, it is a device for drawing triangles in parallel (4 billion triangles per second on a modern GPU) by executing a vertex shader for each vertex and a fragment shader for each pixel.

A shader is a program writen in GLSL (OpenGL Shading Language) [4] that runs on the GPU and is responsible for determining how each pixel on the screen should be colored based on the geometry, lighting, and texture of the scene. A chart map is a texture (an image) that can

[1] According to Plutarch, the Roman General Pompey in 56 BC when informed about a storm, while sailing to Sicily to transport vital grain to a starving Rome, said to his sailors, “Navigare necesse; vivere non est necesse”, which translates to english as “Sailing is necessary; living is not necessary”. In 1914, Portuguese poet Fernando Pessoa [1] wrote a poem and used the verse “Navegar é preciso; viver não é preciso.” However, in Portuguese, the word “preciso” means “necessary” but also “precise”. Therefore, the translation could also be “Sailing is precise; living is not precise,” which points to the necessity of some mathematical knowledge to reach a target location by sea. At last, the song “Os Argonautas” was written in 1969 by Brazilian composer Caetano Veloso.

be applied to a 3D model of the earth to create a realistic representation of the planet.

Texture coordinates can be set in each model or sampled at each pixel in the fragment shader. We can also approximate a sphere by subdividing a convex regular polyhedron and solving Mipmapping artifact issues by using Tarini's method [5], in this case. These artifacts show up due to the discontinuity in the seam when crossing the line with 0 radians on one side and 2π on the other. Some triangles may have edges that cross this line, causing the wrong mipmap level 0 to be chosen.

II. Before satellites and the GPS

During the Age of Discoveries (1418-1620) [6], Portuguese cosmographer Pedro Nunes was the first mathematician to propose in 1537 the notion of a loxodrome [7], or course line routes that intersect all meridians with a constant angle defined by a unique compass direction, thus determining a spiral curve headed towards the poles. However, he did not address the problem of determining the azimuth angle for going from a source location to a destination using a map. The Flemish cartographer Gerardus Mercator has given the answer thirty two years later in 1569 [8].

To locate a ship at sea it is necessary to know its latitude and longitude with a good degree of accuracy. Two tools were used to determine the latitude: the astrolabe and the sextant. The astrolabe measured the altitude of the sun at noon or the altitude of a star at night. The sextant was used to measure the angle between a celestial object and the horizon. Both tools allowed sailors to determine their latitude with reasonable accuracy. However, determining longitude remained a challenge until the development of accurate marine chronometers in the 18th century. The navigation sextant invented in 1731 independently by John Hadley [9] and Thomas Godfrey [10] was used to measure the angle between a celestial object and the horizon. It was a crucial tool for determining latitude at sea, allowing sailors to calculate their position north or south of the equator. By measuring the angle of the sun at noon or the angle of a star at night, navigators could determine their latitude with reasonable accuracy.

The first marine timekeeper to address the longitude problem at sea was John Harrison's H1 [11], completed in 1735. It used a pair of linked balance bars and temperature-compensating parts to keep accurate time on a rocking ship. It allowed sailors to compare local time with home port time (Greenwich Mean Time) to find exact east-west positions.

Prior to the invention of the marine chronometer, sailors relied on dead reckoning and celestial navigation to estimate their longitude. As a consequence, disasters like the Scilly Naval Disaster [12] and diseases like scurvy [13] were common among sailors during long voyages.

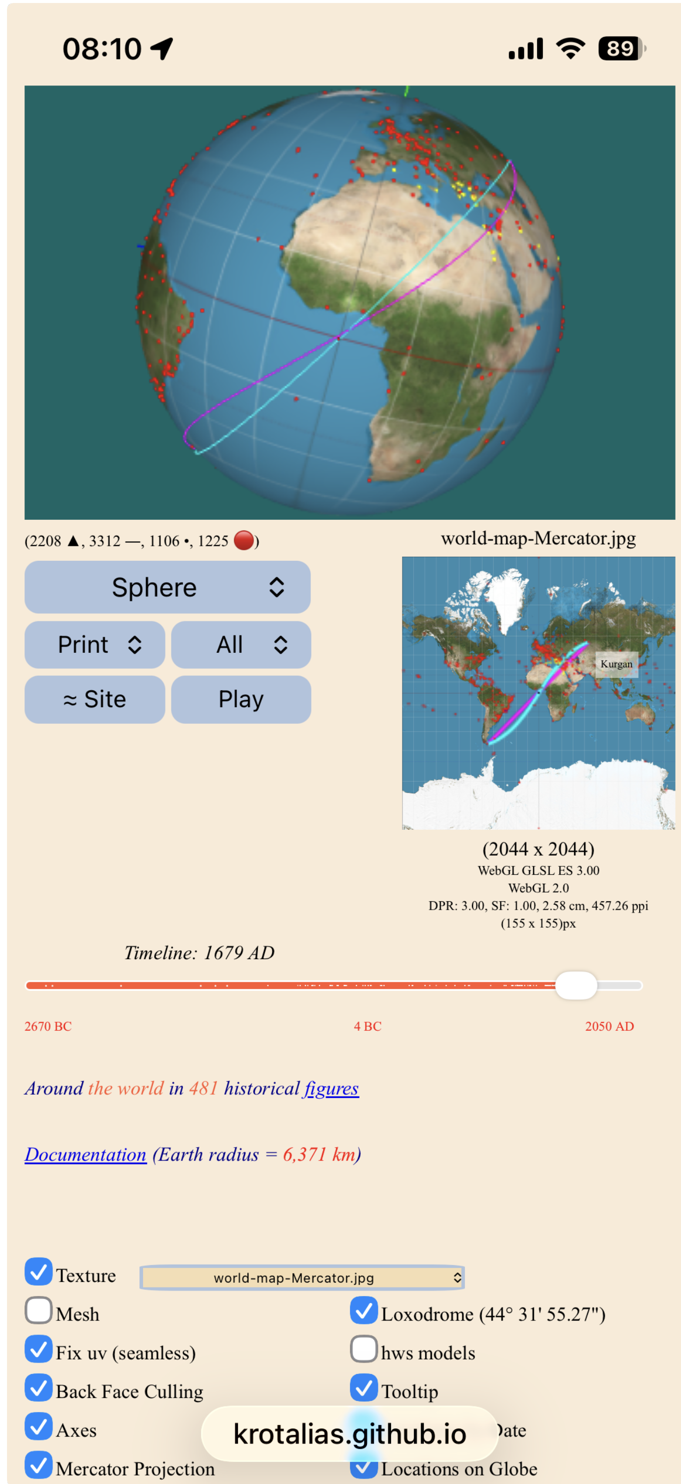


Figure 1. Cape Horn to Kurgan on iPhone 13 (2021) with 6.1" screen (Azimuth: 44.53°- 17,106 km)

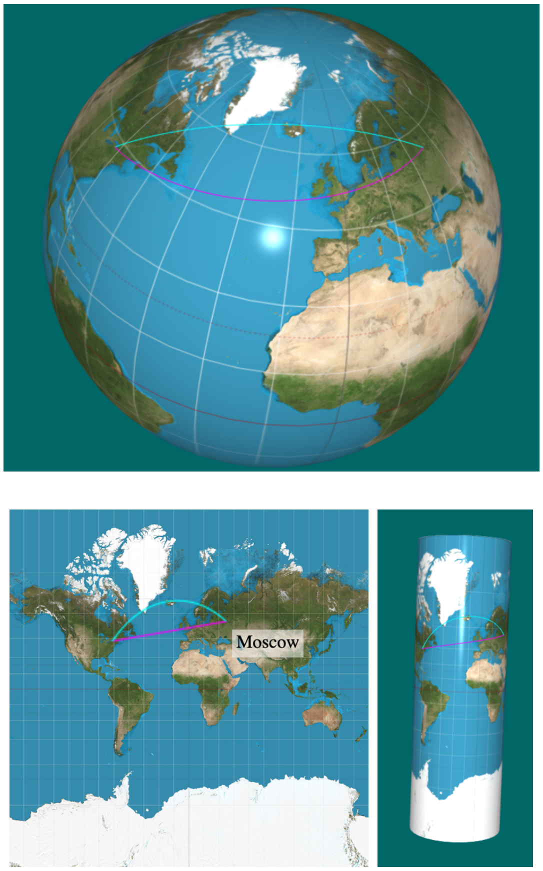


Figure 2. Syracuse - Moscow (Azimuth: 80.19°- 7,396 km)

III. Computer Graphics for Tracing Courses

To display a globe on the screen, its initial position should take into account the obliquity of the earth (23.44°) and illumination parameters, such as the Phong highlight [14] position. In Navigare, the highlight projects onto the equator line if the user has not interacted using the Arcball [15] [16]. If PHP is running on the HTTP server, then any image file in directory textures will be available in the menu. Otherwise, sorry GitHub pages, only the images listed in the HTML file.

Chart maps are images laid onto a sphere as textures and should have an aspect ratio of 2:1 for equirectangular projections or 1:1 (squared) for Mercator projections. As long as a parametrization from $\Re^2$ to $\Re^3$ is known for a surface, a texture can be georeferenced [17]. This is the case of the quadrics, and Navigare also supports cones and cylinders.

Maps are transformations from 3D space to 2D space, and they can preserve areas (equal-area maps) or angles (conformal maps). In terms of projection invariance, loxodromes appear as straight lines on conformal map projections. Therefore, the success of the Mercator projection lies in its ability to preserve angles, making it ideal for navigation (directions on the map match the directions on the compass). However, it distorts areas, especially near the poles, where landmasses appear much larger than they are in reality. Meridian and parallel scales are the same, meaning that distances along a parallel or meridian (in fact, in all directions) are equally stretched by a factor of $\sec(\phi) = 1/\cos(\phi)$, where $\phi \in [-85.051129°, 85.051129°]$ is its latitude.

At latitude ϕ, horizontal distances on the chart are magnified by $\sec(\phi)$ to make the meridians parallel. So locally, the vertical distances should be magnified by $\sec(\phi)$ also to make the projection angle-preserving [18] (Figure 3).

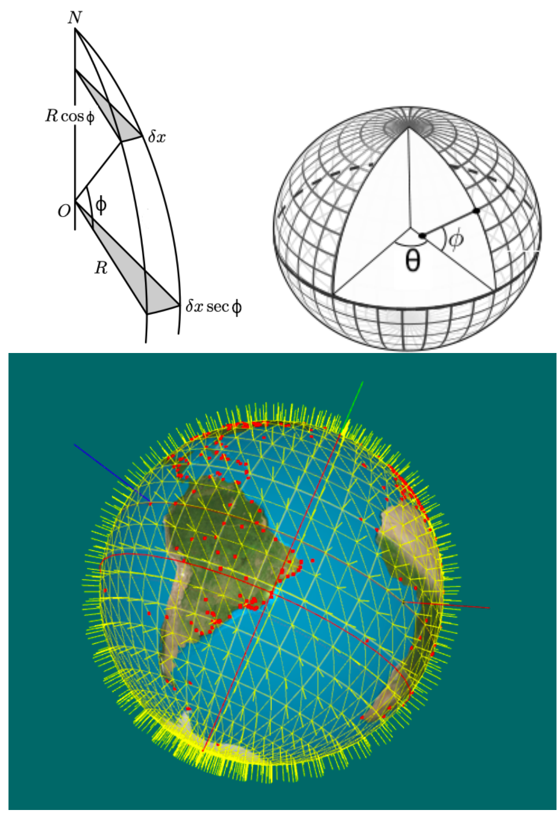


Figure 3. Mercator vertical distance.

ϕ = latitude
θ = longitude
R = radius of the sphere
$R_\oplus$ = radius of the earth
$C_\oplus$ = circumference of the earth
$\delta\theta = \delta x$
$R\cos(\phi)/R = \delta x/d \implies d = \delta x/\cos(\phi) = \delta x \ \sec(\phi)$

Mercator chart [19]

- $(x, y) = M(R, \theta, \phi)$
- $x = R\ \theta, -\pi \leq \theta \leq \pi$
- $y = R\int_0^\phi \sec(\phi)d\phi = R\ \ln[\tan(\pi/4 + \phi/2)]$, $-\pi/2 \leq \phi \leq \pi/2$
- For a square Mercator chart, $-\pi \leq y \leq \pi \implies \phi \in [-85.051129°, 85.051129°]$

- $\phi = 2\ \tan^{-1}(e^y) - \pi/2, -\pi \le y \le \pi$
  $\rightarrow -85.051129° \le \phi \le 85.051129°$

Meridional Parts [20]

- Meridional parts quantify vertical chart lengths by scaling them relative to horizontal minute units at the equator
- 1 Nautical Mile = one minute of longitude at the equator
- 1 nm $= \frac{2\pi\ R_\oplus}{21600}$ km $= \frac{40030}{360 \times 60}$ km $= 1.853\,240\,7$ km (1852 m or 6076.12 ft)
- $C_\oplus = 21600$ nm ($C_\oplus/2 = 10800$ nm)
- 1 radian on earth $\approx (10800/\pi)$ nm
- $MP = (10800/\pi) \times \ln[\tan(\pi/4 + \phi/2)]$ minutes of arc length (not using the spheroid shape of the earth)
- $MP = toDegrees(toMercator(toRadian(\phi))) \times 60$
- The knot is a unit of speed equal to one nautical mile per hour, exactly 1.852 $\mathrm{km\,h^{-1}}$ (1.150 78 $\mathrm{m\,h^{-1}}$)
- Captains in the 1700s tossed ropes overboard with knots spaced out at 50 ft and used a sand glass that measured half of a minute to approximate the speed of a ship in nautical miles per hour
- $50\frac{ft}{0.5min} = 100\frac{ft}{1min} \times 1\frac{nm}{6076ft} \times 60\frac{min}{h} = 0.9875$
  $\approx 1\frac{nm}{h}$

Loxodromes [21]

- $\phi =$ latitude
- $\theta =$ longitude
- $\alpha =$ bearing angle (measured clockwise in degrees from north)
- $\tan(90° - \alpha) = \cot(\alpha) = 1/\tan(\alpha) = dx/dy$
- $dx = R\ \cos(\phi) d\theta$
- $dy = R\ d\phi$
- $\tan(\alpha) = dx/dy = \cos(\phi) d\theta / d\phi$
- $d\theta/d\phi = \tan(\alpha)\sec(\phi)$
- $\int \sec(\phi) d\phi = 1/\tan(\alpha) \int d\theta$
  $\implies \tan(\alpha) \int \sec(\phi) d\phi = (\theta - \theta_0)$
- $(\theta - \theta_0) = \int_{\phi_0}^{\phi} \tan(\alpha)\sec(\phi) d\phi$
- $\theta(\phi) = \theta_0 + \tan(\alpha)\ \ \ln[\tan(\pi/4 + \phi/2)/\tan(\pi/4 + \phi_0/2)] = \theta_0 + \tan(\alpha) \times diffMercator(\phi, \phi_0)$
- $\theta(\phi) = longitudeOnLoxodrome(\phi_0, \phi, \alpha)$

The Web Mercator projection, on the other hand, is a variant of the Mercator projection, which is widely used in web mapping applications. It was designed to work well with the Web Mercator coordinate system, which is based on the WGS 84 datum [22]. The projection is neither strictly ellipsoidal nor strictly spherical, and it uses spherical development of ellipsoidal coordinates. The underlying geographic coordinates are defined using the WGS 84 ellipsoidal model of the Earth's surface but are projected as if defined on a sphere. Misinterpreting Web Mercator for the standard Mercator during coordinate conversion can lead to deviations as much as 43 km on the map [23] (21 km on the ground) [24]:

```
const Tromsø = gpsCoordinates["Tromsø"]; // Norway
const { latitude: lat, longitude: lon } = Tromsø;
const rlat = toRadian(lat);
const psi1 = toMercator(rlat, true);
const psi2 = toMercator(rlat);
const N_ellipsoid = earthMajorAxis * psi1;
const N_sphere = earthMajorAxis * psi2;
const ndiff = Math.abs(N_ellipsoid - N_sphere);
console.log(`Northing difference (${lat}): ${ndiff.toFixed(2)} km`);
// Northing difference (69.6517): 40.11 km

const latGlobe = toDegrees(toSpherical(psi1));
const locGlobe = { latitude: latGlobe, longitude: lon };
const gdiff = haversine(loc, locGlobe).km;
console.log(`Ground difference (${lat}): ${gdiff.toFixed(2)} km`);
// Ground difference (69.6517): 13.97 km
```

Nonetheless, all formulae implemented in Navigare consider the globe as a perfect sphere. The only exceptions are functions calculateLoxodromeDistanceWGS84 and toMercator.

It is impressive how Gerardus Mercator was able to create such a projection [8] in a time (1569) when there was no calculus (integrals, derivatives — Leibniz in 1674 [25] and Newton in 1666 [26]) or even logarithm tables (John Napier, 1614) [27]. This is an example of the need for calculus to solve a navigation technological issue that had an enormous economic impact.

The first application of calculus to solve this map's meridional spacing was by English mathematician Edward Wright in 1599 [28] [29]. Wright's geometric proof conceptually utilized the integral of the secant function. At the time, this closed-form solution was half-century away from being discovered. The actual calculus formulation and logarithmic proofs for Mercator's scaling only emerged in the mid-to-late 1600s, primarily through the works of James Gregory [30] and Isaac Barrow [31], who provided in 1670 the first intelligible, closed-form mathematical proof for the integral of the secant function, resolving the long-standing problem of the spacing of latitudes for the Mercator map projection.

Mercator texture coordinates can be set in a model directly or in the shader that samples texture coordinates for each pixel. Since a unit sphere fits in the WebGL NDC space, it is possible to go into each fragment from:

- cartesian → spherical (equirectangular) → Mercator
- (x, y, z) → (long, lat) → (x, y)
- sample texture at (x, y)

The equirectangular projection, also known as the equidistant cylindrical projection (e.g., plate carrée), is a way of representing the Earth's surface on a flat plane, and maps longitude and latitude lines into straight, evenly spaced lines, essentially flattening the sphere into a rectangle.

Its meridian scale is 1 meaning that the distance along lines of longitude remains the same across the map, while its parallel scale varies with latitude, which means that the distance along lines of latitude is stretched by a factor of $\sec(\phi)$. Ptolemy claims [32] that Marinus of Tyre invented the projection in the first century (AD 100). The projection is neither equal area nor conformal. In particular, the plate carrée (flat square) has become a standard for global raster datasets. Loxodromes are no longer straight lines on the equirectangular projection, but possess a sigmoid shape [33], rendering them unusable for navigation purposes (Figure 4).

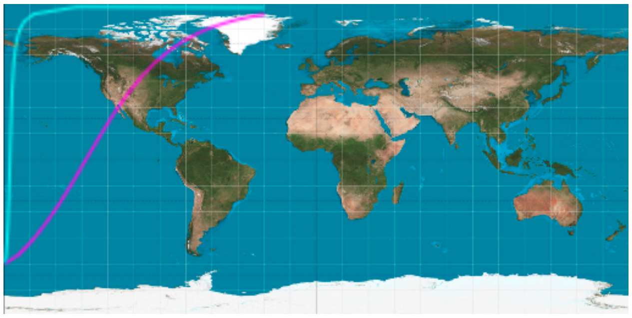
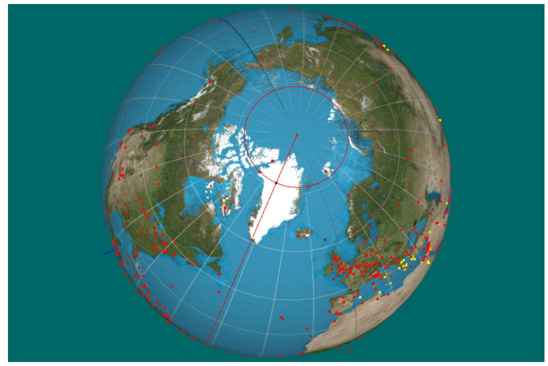


Figure 4. Loxodrome (magenta) is a sigmoid (S-shaped curve) in the equirectangular projection

All projection types involve some kind of distortion, which is not a problem per se. The real problem is being tricked by the shape of the chart. The Mercator projection, for instance, stretches land sizes near the poles while keeping compass directions. Country size changes is a problem for school classrooms if students do not understand what causes it. To circumvent the map distortion confusion, the flat chart should always be paired with a synced 3D globe, as can be seen in Figure 5. The interface we propose to tackle this problem presents a 3D view of the globe all the time so the real sizes are easily seen. Every time a user clicks on the chart, the globe rotates to make the corresponding point on the globe face the viewer. This is accomplished by keeping a forward vector from the origin to the center of the projected sphere, which is the position of the Phong highlight. The position vector from the origin to the clicked point is then just rotated to become the new forward vector. However, the north (Y-axis) also rotates, and the globe can be positioned in a non-conventional way, for example, "upside-down." A second rotation around the new forward vector can reset the north vector appropriately. It is amazing what we can do with a few cross and dot product operations.



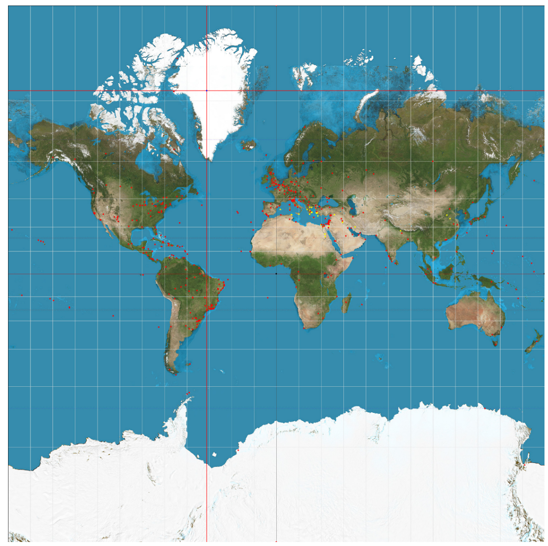


Figure 5. Greenland real size and Mercator country size distortion

IV. The Navigare application Around The World in 500 historical figures

When a child has to study history, he/she is generally not able to visualize the actual location of an event. For instance, where were the locations of Thrace, Anatolia, Troy, the Parthian Empire, the Inca Empire, and Rapa Nui?

Therefore, it is important to present in a graphical way, the connection between historical events in time and space, and Navigare is an adequate application for accomplishing this goal. However, how Navigare is implemented is not the main point. For not using any npm packages or bundlers, Navigare sticks only with HTML, CSS, JavaScript, and WebGL, plus some packages, including three.js [34].

The concepts applied in the implementation rely on Linear Algebra for spatial transformations [35], Computer Graphics for rendering theory and pipelines [36], and Web Development frameworks to display and run the interactive visual content in a browser. However, Computer Graphics has become a commodity, much like database management; everyone utilizes it, yet very few people are interested in understanding the underlying mechanics. This trend poses a challenge for educators, as it becomes increasingly important to inspire curiosity and a deeper appreciation for the artistic and technical aspects of graphics programming. By focusing on foundational principles, we encourage students to explore beyond the surface and engage with the creative possibilities that computer graphics offers.

Of course, everything could have been implemented using a package such as three.js only, although OrbitControls does not provide the flexibility needed to build the interface. Nonetheless, exploring alternative libraries and frameworks could provide additional tools and features that enhance the user experience and streamline development. Ultimately, the goal is to create an environment where students feel empowered to experiment and innovate, pushing the boundaries of what they can achieve in graphics programming.

Navigare used no AI and each event was researched one by one about its history for dating the occurrences associated with a site. All sites are hold in a JSON file called “locations.json” and it is possible to cycle through the historical figures by country, when choosing a two letter ISO country code in the interface. For example, selecting BR will display all the figures from Brazil, while CA will show those from Canada (Figure 6) and US from the United States of America. The sorting of dates uses the day, month and year of the last date of the last entry in the remarkable list field.

E.g., “Battle of Britain, 10 July 1940 - 31 October 1940” is sorted using “10 July 1940”.

```
"Moffat": {
    "country": "× Scotland (United Kingdom)",
    "remarkable": [
        "Hugh Dowding (Air Chief Marshal), 1882-1970",
        "Battle of Britain, 10 July 1940 - 31 October 1940"
    ],
    "latitude": 55.333814,
    "longitude": -3.444884
},
"San Francisco": {
    "country": "× California (United States)",
    "remarkable": [
        "Golden Gate Bridge (1.6 km), 5 January 1933 - 27 May 1937",
        "William Halsey Jr. (Fleet Admiral), 1882-1959",
        "Doolittle Raid departure, 2 April 1942"
    ],
    "latitude": 37.77493,
    "longitude": -122.41942
},
```

Dates should be preceded by a comma in the remarkable list field to be considered for sorting (no error checking is done). Examples:

, 324	(1 0 324)
, 657 BC	(1 0 −657)
, 330−1453	(1 0 330)
, 10 July 1940 − 31 October 1940	(10 6 1940)
, 2 April 1942	(2 3 1942)

V. Picking Locations

Navigare displays the location of a city as the intersection of its line of latitude (parallel) and line of longitude (meridian).

When a user picks a point in the chart map, using the mouse or any pointer device, Navigare displays its location on the chart map and on the globe by executing the following steps:

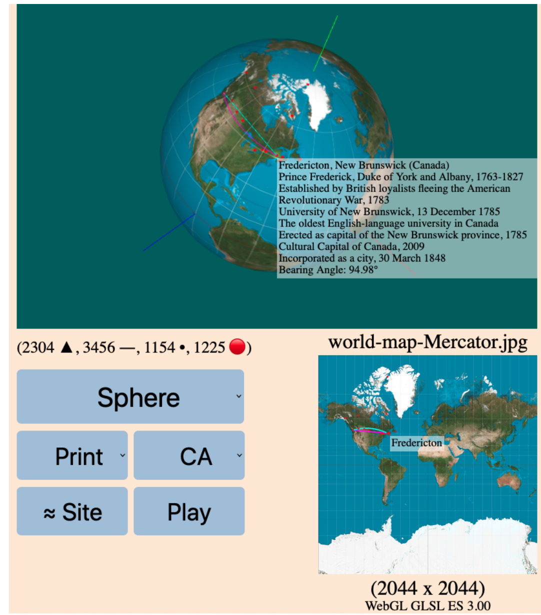


Figure 6. Vancouver to Fredericton (Azimuth: 94.98°- 4,149 km)

1) Convert the screen coordinates of the mouse pointer into texture coordinates (u, v), then into GCS coordinates (longitude, latitude) using the currentLocation, and optionally transforming them to Mercator coordinates.
2) Draw meridian and parallel lines: use the lineTo() from HTML5 by placing an element <canvas> on top of the element <img>.
   - This is simple to accomplish by nesting the <canvas> element with position absolute in a <div> element with position relative and the same size as the image element.
   - The <canvas> element should have a higher z-index than the image element and ignore pointer events.
3) Define a pointerdown event handler to set the currentLocation as the gpsCoordinates ”Unknown” and draw the lines by calling drawLinesOnImage.

When a user picks a point directly onto the globe’s surface, it is necessary to implement a 3D pick in this case by casting a ray, unprojecting it, and finding its closest (first) intersection (relative to the viewer) with the polygonal surface of the model:

1) Shoot the ray from the mouse position and intersect it against the surface of an implicit sphere by solving a second-degree equation [37].
   - Alternatively, intersect the ray against each face of the polygonal surface by testing if the ray intersects the plane of a face and then checking if the intersection point is inside the corresponding triangle [38].
2) Select a position on the globe by clicking or pointing

directly in the WebGL canvas.

3) Display the GCS coordinates of the pointer: If the globe is spinning, keep track of the cursor position and then shoot a ray through this position to find its intersection on the globe.
4) Use the Phong highlight position on the globe as a cursor for touch devices, since they do not have a cursor position: Get its initial position in world coordinates (0, 0, 1) - or in GCS coordinates (-90°, 0°) - and transforms it into screen coordinates.
5) Exhibit the current location name in 3D: Transform its GCS coordinates into screen coordinates (pixels) and use the position HTML properties “top” and “left” of the WebGL <canvas> element (plus a tooltip element) to display the text.

VI. Conclusions

Since elementary school, people learn Geography through planar maps [39]. Nonetheless, the earth is an oblate spheroid, and shortest distances are geodesics [40] and not straight lines on a piece of paper. Therefore, it is important to realize the difference between distance measurements on the plane and on a sphere.

The interface we propose presents a three-dimensional view of the earth to enhance the perception of the spherical shape of our planet. Furthermore, we discussed the effect of projections in shape distortion [41] and the need of mathematical integration for the development of the Mercator projection and its importance for navigation in the past. The ability of associating historical events with their geographical locations sorted by dates is a powerful mechanism for learning and understanding history in a more engaging and interactive way.

The code is open source and hopefully it will be useful for teachers and students alike to enhance their map literacy. Navigare was implemented in a peculiar way, which took advantage of JavaScript’s dynamic typing and the flexibility of WebGL to create an interactive and engaging learning experience. The interface runs on any modern web browser, and it is responsive to different screen sizes and resolutions thus allowing users to access the application from various devices, including tablets and smartphones. In certain cases this may be the only way to access the application, and it is important to make it as accessible as possible.

References

[1] F. Pessoa, Navegar é preciso, Publicado postumamente ou em Obras Completas, frase célebre: ”Navegar é preciso; viver não é preciso” (1914).
URL https://dominiopublico.mec.gov.br/download/texto/jp000001.pdf

[2] IBM, Geographic coordinate system (2026).
URL https://www.ibm.com/docs/en/informix-servers/15.0.x?topic=data-geographic-coordinate-system

[3] K. Matsuda, R. Lea, WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL, OpenGL Technical Library, Addison-Wesley, 2013.
URL https://sites.google.com/site/webglbook/home?authuser=0

[4] J. Kessenich, R. Rost, The OpenGL Shading Language, Version 4.60.8, Khronos Group, accessed: 2026-08-11 (August 2023).
URL https://registry.khronos.org/OpenGL/specs/gl/GLSLangSpec.4.60.html

[5] M. Tarini, Cylindrical and toroidal parameterizations without vertex seams, Journal of Graphics Tools 16 (3) (2012) 144–150.
URL http://vcg-legacy.isti.cnr.it/Publications/2012/Tar12

[6] D. Arnold, The Age of Discovery, 1400–1600, Psychology Press, London, 2002.

[7] P. Nunes, Tratado da sphera: com a theorica do sol & da lua. E ho primeiro liuro da Geographia de Claudio Ptolomeo Alexa[n]drino., Germão Galharde, Lisboa, 1537.
URL https://purl.pt/14445/1/index.html#/1/html

[8] G. Mercator, Nova et aucta orbis terrae descriptio ad usum navigantium emendate accommodata, world map in 18 sheets introducing the Mercator projection (1569).
URL https://acm5.blogs.rice.edu/page-f/

[9] J. Hadley, An account of a catadioptrick telescope, made by John Hadley, esq; F. R. S. with the description of a machine contriv’d by him for the applying it to use, Philosophical Transactions of the Royal Society of London 32 (376) (1723) 303–312.
URL https://archive.org/details/philtrans07963102/page/n9/mode/2up

[10] W. B. Ashworth, Scientist of the day: John hadley, The Linda Hall: Science Library and Arboretum (2023).
URL https://www.lindahall.org/about/news/scientist-of-the-day/john-hadley/

[11] R. T. Gould, The marine chronometer, its history and development (1923).
URL https://www.rmg.co.uk/collections/objects/rmgc-object-79139

[12] K. Drew, The british shipwreck that changed the world (2021).
URL https://www.bbc.com/travel/article/20210729-the-british-shipwreck-that-changed-the-world

[13] J. Talmage, D. Maneice, Anson’s voyage, Naval History (August 2023).
URL https://www.usni.org/magazines/naval-history-magazine/2023/august/ansons-voyage

[14] B. T. Phong, Illumination for computer generated pictures, Communications of the ACM 18 (6) (1975) 311–317.
URL https://dl.acm.org/doi/epdf/10.1145/360825.360839

[15] K. Shoemake, Arcball: A user interface for specifying three-dimensional orientation using a mouse, in: Proceedings of Graphics Interface ’92, Canadian Human-Computer Communications Society, 1992, pp. 151–156.
URL https://graphicsinterface.org/wp-content/uploads/gi1992-18.pdf

[16] K. Shoemake, Arcball rotation control, in: Graphics Gems IV, Academic Press, 1994, pp. 175–192.
URL https://theswissbay.ch/pdf/Gentoomen%20Library/Game%20Development/Programming/Graphics%20Gems%204.pdf#page=188

[17] Atractor, Parametrization of the loxodrome, Online (2013).
URL https://www.atractor.pt/mat/loxodromica/saber_parametrizacao1-_en.html

[18] J. M. Spivey, Thomas harriot and the mercator map (1993).
URL https://spivey.oriel.ox.ac.uk/corner/Thomas_Harriot_and_the_Mercator_Map

[19] K. Vezie, Mercator’s projection: A comparative analysis of rhumb lines and great circles, Whitman College (2016).
URL https://www.whitman.edu/Documents/Academics/Mathematics/2016/Vezie.pdf

[20] N. Bowditch, The American Practical Navigator, 2024th Edition, no. 9 in Pub. No., National Geospatial-Intelligence Agency, Bethesda, MD, 2024.
URL https://www.dco.uscg.mil/Portals/9/NMC/pdfs/examinations/bowditch_Vol_2_2019.pdf

[21] C. Veness, Calculate distance, bearing and more between latitude/longitude points, Online, movable Type Scripts (2026).
URL https://www.movable-type.co.uk/scripts/latlong.html

[22] N. G.-I. Agency, Department of defense world geodetic system 1984: Its definition and relationships with local geodetic systems, Technical Report NGA.STND.0036_1.0.0_WGS84, National Geospatial-Intelligence Agency (NGA), current standard

update for the Global Positioning System (GPS) (July 2014).
URL https://nsgreg.nga.mil/doc/view?i=4085

[23] E. Stefanakis, Web mercator and raster tile maps: Two cornerstones of online map service providers, Cartographica: The International Journal for Geographic Information and Geovisualization 52 (2) (2017) 92–108. doi:10.5623/cig2017-203.
URL https://doi.org/10.5623/cig2017-203

[24] EPSG:3857, Wgs 84 / pseudo-mercator, maptyler (2020).
URL https://epsg.io/3857

[25] G. W. Leibniz, Nova methodus pro maximis et minimis, itemque tangentibus, quae nec fractas nec irrationales quantitates moratur, et singulare pro illis calculi genus, Acta Eruditorum (1684) 467–473.
URL https://www.rbhm.org.br/index.php/RBHM/article/view/377/340

[26] S. I. Newton, The october 1666 tract on fluxions, in: D. T. Whiteside (Ed.), The Mathematical Papers of Isaac Newton, Volume I: 1664–1666, Cambridge University Press, 1967, pp. 400–448.
URL https://www.newtonproject.ox.ac.uk/view/texts/normalized/NATP00100

[27] J. Napier, Mirifici Logarithmorum Canonis Descriptio, Andrew Hart, Edinburgh, 1614.
URL https://archive.org/details/mirificilogarit00napi

[28] E. Wright, Errors in Navigation, Printed by Valentine Simmes for Ed. Agas, London, 1599.
URL https://archive.org/details/bim_early-english-books-1475-1640_errors-in-navigation_wright-edward_1599/mode/1up

[29] E. Wright, Certaine Errors in Navigation, Detected and Corrected, Printed by Felix Kingston, London, 1610.
URL https://archive.org/details/bim_early-english-books-1475-1640_certaine-errors-in-navig_wright-edward_1610/page/n5/mode/2up

[30] J. Gregory, Geometriae Pars Universalis, Museo Galileo, 1668.
URL https://archive.org/details/gregory_universalis

[31] I. Barrow, Lectiones Geometricae, Gulielmi Godbid, Londini, 1670, english translation edited by J. M. Child: *The Geometrical Lectures of Isaac Barrow*, Open Court Publishing Co., 1916.
URL https://archive.org/details/geometricallectu00barruoft/mode/2up

[32] C. Ptolemy, Ptolemy's Geography: An Annotated Translation of the Theoretical Chapters, Princeton University Press, Princeton, NJ, 2000.
URL https://uwm.edu/libraries/2026/04/23/1478-ptolemys-geography/

[33] M. Lapaine, A new derivation of the formula for the length of a loxodrome arc on a sphere using cylindrical projections, ISPRS International Journal of Geo-Information 14 (4) (2025) 137. doi:10.33390/ijgi14040137.
URL https://www.mdpi.com/2220-9964/14/4/137

[34] R. Cabello, three.js (2010).
URL https://threejs.org/

[35] J. Hefferon, Linear Algebra, 4th Edition, 2021, available online as a free textbook.
URL https://hefferon.net/linearalgebra/

[36] D. J. Eck, Introduction to Computer Graphics, version 1.4 Edition, Hobart and William Smith Colleges, 2023, available online as a free textbook.
URL https://math.hws.edu/graphicsbook/

[37] E. Gullotti, Line - sphere intersection (2007).
URL https://www.illusioncatalyst.com/notes_files/mathematics/line_sphere_intersection.php

[38] P. C. P. Carvalho, P. R. Cavalcanti, Point in polyhedron testing using spherical polygons, in: A. W. Paeth (Ed.), Graphics Gems V, Academic Press, 1995, pp. 42–49.
URL https://theswissbay.ch/pdf/Gentoomen%20Library/Game%20Development/Programming/Graphics%20Gems%205.pdf#page=56

[39] D. The, The mathematics of maps (2017).
URL https://math.uit.no/ansatte/dennis/MoMS2017-Lec3.pdf

[40] H. Busemann, The Geometry of Geodesics, Pure and Applied Mathematics, Academic Press, 1955.
URL https://ia801900.us.archive.org/25/items/in.ernet.dli.2015.134526/2015.134526.The-Geometry-Of-Geodesics_text.pdf

[41] K. Drew, The true size of... (2026).
URL https://thetruesize.com/