

TRPINAV: A SYSTEM FOR TERRAIN, ROUTE, POINTS OF INTEREST AND
NAVIGATION

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Lana Hodzic

June 2016

© 2016
Lana Hodzic
ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: TRPINav: A System for Terrain, Route,
Points of Interest and Navigation

AUTHOR: Lana Hodzic

DATE SUBMITTED: June 2016

COMMITTEE CHAIR: Zoë Wood, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Aaron Keen, Ph.D.
Professor of Computer Science

COMMITTEE MEMBER: Shinjiro Sueda, Ph.D.
Assistant Professor of Computer Science

ABSTRACT

TRPINav: A System for Terrain, Route, Points of Interest and Navigation

Lana Hodzic

Maps are as fundamental to society as language and the written word. Maps are an abstraction of reality that help people to better understand the world around them. Route maps, specifically, depict a path from one place to another. Many route maps generated today contain excessive information making it difficult for people to process visually. Previous work has been done in generating routes, searching for nearby points of interest, and path generalization.

This thesis proposes a system, TRPINav, that will integrate terrain data, route data, points of interest, and navigation in a cartographic visualization. This prototype navigation application includes a system designed to unify data from various sources, such as Yelp, Google Maps, and Digital Elevation Models (DEMs) into a single coordinate frame for useful navigation and reference. This system presents a model to handle data acquisition and unification of these data sources and render them in a simplified and pleasing representation to the user. The prototype includes simple rendering techniques and a user study to validate which features are appealing and useful to the user. Analysis shows that users are generally pleased with features of the system, and that the system can help answer some geographic questions. The results of this analysis are used to determine future iterations of the system.

ACKNOWLEDGMENTS

A big thank you to my family for supporting me in everything I did and helping me get to where I am today. To my advisor Dr. Zoë Wood for always believing in me and pushing me to fulfill my potential.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	viii
CHAPTER	
1 Introduction	1
2 Background	3
2.1 Cartography	3
2.2 Geometry Clipmaps	6
2.3 Geometry Clipmaps Using Vertex Textures	8
3 Related Works	10
3.1 LineDrive	12
3.2 Destination Maps	14
3.3 Points of Interest	16
4 Implementation	19
4.1 Data Gathering	20
4.1.1 KML Data	20
4.1.2 ASC DEM Data	22
4.1.3 Points of Interest Data	24
4.1.3.1 Yelp API	24
4.1.3.2 Google Places API	26
4.1.4 Route Data	28
4.1.4.1 Google Maps Elevation API	31
4.2 Data Processing	32
4.2.1 Data Conversion	33
4.3 Data Rendering	34
4.3.1 Map Outline	35
4.3.2 Route Generation	36
4.3.3 Points of Interest	37
4.4 Terrain Generation	39
4.5 Top-down View	44

4.6	Navigation View	45
4.7	Performance	46
5	Validation	48
5.1	Comparison of Maps	49
5.1.1	TRPINav Map Critiques	49
5.1.1.1	Noticed Features	51
5.1.1.2	Liked Features	52
5.1.1.3	Disliked Features	54
5.1.1.4	Missing Features	56
5.1.2	Google Map Critiques	57
5.1.2.1	Noticed Features	57
5.1.2.2	Liked Features	59
5.1.2.3	Disliked Features	61
5.1.2.4	Missing Features	61
5.1.3	Three Map Comparison	62
5.2	Real-Time Testing of the System	72
5.2.1	Difficulties	73
5.2.2	Top Down View and Navigation View Comparison	74
5.2.3	User Preferences	76
6	Conclusion and Future Work	81
6.1	User Settings	81
6.2	Points of Interest	82
6.3	Path Generation and Simplification	84
6.4	Terrain Generation	84
6.5	Information Processing	86
6.6	Port to Mobile or Webpage	86
6.7	Scaling of the Map	87
	BIBLIOGRAPHY	89
	APPENDICES	
A	Top Down View Evaluation	92
B	Real-Time User Evaluation	100

LIST OF FIGURES

Figure	Page
2.1	An example on the different ways to organize elements within a map found in [9] 4
2.2	An example of representing placement of information based on importance found in [9] 4
2.3	Example of basic elements found on a map [5] 5
2.4	Representing scale using statement of equality, a representative fraction, or a bar graph respectively [9] 5
2.5	Example of the geometry clipmaps implementation in the TRPINav system 6
2.6	An example from Geometry Clipmaps [19]. Distribution of the active, clip, and desired regions 7
2.7	From GPU Gems [22], example of toroidal access and update 9
3.1	Example of the map use cube to explain characteristics for the three main visualization goals [23] 11
3.2	LineDrive Final Result 13
3.3	Destination Maps, Seattle in Treasure Map Style 15
3.4	Destination Maps, Chicago in Sketchy Style 15
3.5	Navigation view of Google Maps allowing users to choose quick points of interest 17
3.6	Google Map in navigation view overlaid with coffee shops 17
4.1	TRPINav System Overview 19
4.2	TRPINav System Overview 20
4.3	Example of the rendered KML data 21
4.4	Example of the California ASC file 23
4.5	Final rendering of the DEM data in the system 24
4.6	One of the businesses returned by Yelp Search API for Donut Shops in San Luis Obispo, CA 25
4.7	Final rendering of Yelp data within the system 26
4.8	An example of a park returned by the Google Places API 27

4.9	Final rendering of parks within the system	27
4.10	Directions from New York to New Jersey JSON response	29
4.11	Overview Polyline Encoded String	30
4.12	Geocoding JSON Response Object	31
4.13	Elevation API JSON Response Object for a path between two points with three samples	32
4.14	Some of the values used to convert from WGS84 latitude and longi- tude to Cartesian	34
4.15	Image of the United States in mesh lines generated in ionEngine . .	35
4.16	Image of the United States in mesh lines and solid mesh background generated in ionEngine	36
4.17	Coffee and Donut shops that can be found along the route rendered as sprites	37
4.18	Parks that can be found along the route rendered as sprites	38
4.19	Parks that can be found along the route rendered as sprites	39
4.20	The left picture uses the actual camera, the right picture uses the fake camera. Highest resolution geometry is closest to the camera in the left picture, and further away in the right image	41
4.21	Most commonly found terrain color scale, can be toggled on with number '1' key if color scheme changed	42
4.22	Greyscale terrain color, toggled on by pressing the number '2' key .	42
4.23	Two-tone terrain color, toggled on by pressing the number '3' key .	43
4.24	Top-down view of a path from San Luis Obispo, CA to Bellingham, WA	44
4.25	Frenet frame, Spline, and Path Example	46
5.1	TRPINav Map used in survey comparison	50
5.2	Google Map used in survey comparison	50
5.3	Hand drawn map used in survey comparison	50
5.4	Distribution of features that stood out to users at first glance of the TRPINav Map top down view	52
5.5	Distribution of features that users liked in the TRPINav Map top down view	54
5.6	Distribution of features that users disliked in the TRPINav Map top down view	55

5.7	Distribution of features that users noticed first in the Google Map .	59
5.8	Distribution of features that users liked in the Google Map	60
5.9	Distribution of responses for determining the trip that looks the most fun	63
5.10	Distribution of responses for determining clearest route	64
5.11	Distribution of responses for finding the closest park	65
5.12	Distribution of responses for deciding where to stop on a drive . . .	66
5.13	Distribution of responses for the location of San Francisco	67
5.14	Distribution of responses for finding the California/Nevada border .	68
5.15	Distribution of responses determining cities close to destination . .	69
5.16	Distribution of responses for which map best used for directions . .	70
5.17	Distribution of responses for which map best used for pit stop . . .	71
5.18	Distribution of users' preference to show or hide road names	73
5.19	Distribution of users' preference to show or hide road names	74
5.20	Usefulness of points of interest in the top down view	75
5.21	Usefulness of points of interest in the navigation view	75
5.22	User sentiment on usefulness of road names in the top down view .	76
5.23	User sentiment on usefulness of road names in the navigation view .	76
5.24	User preference on camera height	77
5.25	The three options users were given in the Google Form, which can be found in Appendix B	78
5.26	Distribution of user preference for terrain color scheme	78
5.27	The three options users were given in the Google Form, which can be found in Appendix B	79
5.28	Distribution of user preference for coffee icon	79
5.29	The three options users were given in the Google Form, which can be found in Appendix B	80
5.30	Distribution of user preference for donut icon	80

Chapter 1

INTRODUCTION

Route maps are one of the most common forms of maps today. Unfortunately, they can be cluttered with extraneous information that the user does not need for the purpose of route exploration [1]. Extraneous information can include any elements or features of a map that do not enhance a user's understanding of it [9]. Computer visualizations are an emerging field in cartography due to the many technological developments and improvements over the years [23]. Cartographic visualizations are used to explore, analyze, synthesize and present data. Maps, such as Google Maps, are a form of cartographic visualization, as they synthesize and present geographic information to users. As users begin to expand their use of maps, different maps need to be created for different types of use cases [23].

Automatic generation of maps causes very exact, computerized maps [1]. While exact directions can be important, it is difficult to lay out all of the data in the exact form we see in the real world. Maps are meant to be “symbolic abstractions - generalizations or representations - of reality” [9]. Choosing how to represent the real world is important. Choosing which features to show is also important. Features within a map are important to the way that users solve their geographic problems. Maps can provide concrete answers to questions such as: Where is this? How do I get there? Where can I find this? [23]. These are the types of questions a cartographic visualization system can help answer.

Today, many maps contain route generation and point of interest searches as separate functionalities. The two, however, can be integrated to provide directions to a user and provide ideas for rest stops, interesting activities, and a breath of fresh air on longer drives. Instead of having to do extra searches at different locations, a

system could provide a number of points of interest along the route. This would allow users to explore their surroundings even more and enhance their driving experience. In addition to giving users more information about activities, geographic information is important to helping users understand their surroundings. Many maps have topographic information or options for viewing elevation information, but these are not three-dimensional interactive maps. Having a system that integrates elevation for a path from start to end, in an immersive experience, can help the user feel more comfortable with driving or entice them to explore one area more than others.

This thesis proposes a system, TRPINav, that will integrate terrain data, route data, points of interest, and navigation in an interactive three-dimensional computer rendered cartographic visualization. This prototype navigation application includes a system designed to unify data from various sources, such as Yelp, Google Maps, and Digital Elevation Models (DEMs) into a single coordinate frame for useful navigation and reference. This system presents a model to handle data acquisition and unification of these data sources and render them in a simplified and pleasing representation to the user. This thesis includes a prototype with simple rendering techniques and a user study to validate which features are appealing and useful to the user. Analysis shows that users are generally pleased with features of the system, and that the system can help answer some geographic questions. The results of this analysis are used to determine future iterations of the system.

The contributions of this thesis are:

- A system unifying terrain data, route data, and points of interest data
- A 3D interactive cartographic visualization representing the unified data
- A user study revealing features that are necessary for understanding path traversal and for locating points of interest

Chapter 2

BACKGROUND

This thesis project builds on top of basic cartographic principles and uses graphics implementations discussed in this chapter.

2.1 Cartography

“Cartography is very much a process of abstraction in which features of the real world are generalized or simplified to meet the demands of the theme and audience” [9]. It is important to consider which details are truly necessary to the understanding of a map. The amount of detail on a map is dependent, in part, on the scale of the map. Consider a map of San Luis Obispo county. At the county scale, a useful map would typically contain major roads and larger cities, but not every single road. Imagine zooming in further to San Luis Obispo city. More details can be included at the city scale such as smaller roadways, rivers, landmarks, etc. When considering details to show, it is often good to follow a “less is more” methodology, because too much detail or a complex layout can become a barrier to communication [9].

By viewing cartography as a form of communication, maps become symbolic abstractions or representations of the real world [9]. There are two important decisions to make when designing a map: how much should the depiction of the real world be simplified in the representation, and how should relationships between entities be symbolized? While the qualities of a “good” map are ambiguous, the perceived quality of a map often relates to how well it communicates with the intended audience. This communication can be greatly affected by aesthetic qualities. Basic aesthetic elements of a map include form, layout, composition, scale, and the proportion of

content. Figure 2.1 shows an example of a few different possible layouts for a map. The layouts illustrated in the picture are shown best to worst from left to right. The middle and the rightmost image obscure the map and make it difficult for the user to see the map.



Figure 2.1: An example on the different ways to organize elements within a map found in [9]

Depending on importance, location of information or elements within a map varies. Figure 2.2 shows the distribution of information based on importance [9].

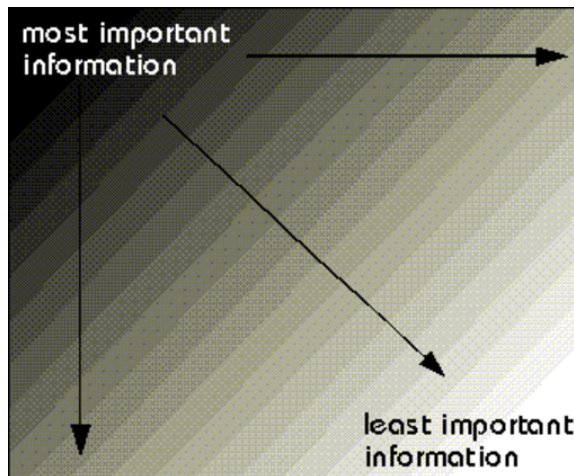


Figure 2.2: An example of representing placement of information based on importance found in [9]

The layout of a map determines the scale or proportion of elements within the map. The composition determines which features are displayed. Regardless of layout and composition choices, there are some elements of a map which are necessary: a title, scale, legend, projection used, and information about map data sources. Figure 2.3 provides an example of a map with some of the basic elements from a chapter in

[5] called “Cartographic Principles”. These basic elements are critical to the user’s comprehension of the map.

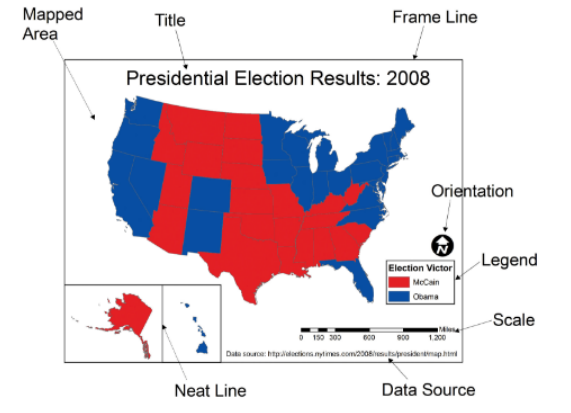


Figure 2.3: Example of basic elements found on a map [5]

A map is not about just transferring information to the user, but also “enhancing the map user’s understanding of reality” [9]. The basic elements illustrated above give the map user more context and information with which to process the map. Elements on the map can be represented in a number of different ways. For instance, the scale of a map can be communicated either graphically or numerically. Figure 2.4 shows two ways of representing the scale. A legend or symbol table is both a written and graphical communication of symbols found within the map. This can be important to understanding a map.

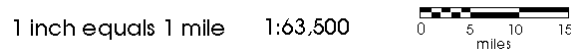


Figure 2.4: Representing scale using statement of equality, a representative fraction, or a bar graph respectively [9]

In a well-designed map, each element should provide meaning. Aesthetic properties of elements (such as size, color, etc.) should be chosen to optimally convey the element’s purpose [5]. This thesis uses these cartographic ideas and principles when producing the rendered map product.

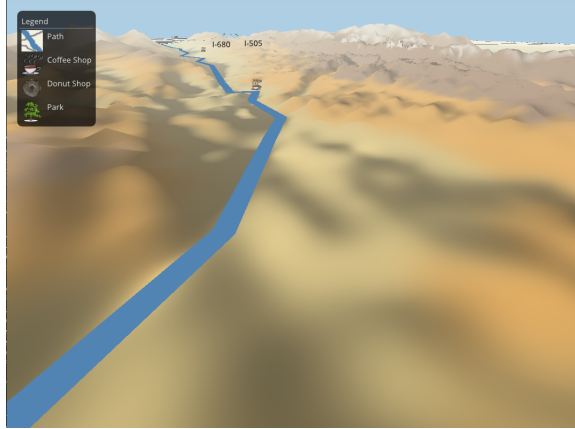


Figure 2.5: Example of the geometry clipmaps implementation in the TRPINav system

2.2 Geometry Clipmaps

In order to render terrain for the 3D views in this project, geometry clipmaps were used. This technology is important due to the large amount of terrain information and path length we propose to support in this system. Figure 2.5 shows a rendering of the terrain within the TRPINav system.

The geometry clipmap is introduced, by Losasso and Hoppe, as a “novel approach to level-of-detail (LOD) control in terrain rendering” [19]. This implementation of terrain rendering uses the cache to store terrain in grids centered around the camera (or, as the paper calls it, “the viewer”). The grids represent “filtered versions of the terrain at power-of-two resolutions, and are stored as vertex buffers in video memory” [19]. With help from the nested layers of terrain, LOD is determined by the distance, in world space, from the camera position. The geometry clipmap caches a terrain pyramid using a set of m levels. Each level contains a grid of vertices. Efficient incremental updates are possible due to toroidal array access, meaning there is wraparound addressing. When the variable used to access into a grid exceeds the width or height, that variable is reset to zero. Each level is made up of a set of rectangular regions. The clip region is the extent in world space of the n by n grid

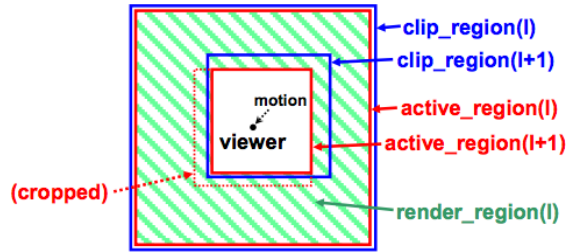


Figure 2.6: An example from Geometry Clipmaps [19]. Distribution of the active, clip, and desired regions

at a specific level. The active region is the extent of the area to be rendered and is a square of size n by n centered at the viewer (camera). As the camera moves, the clip region of each level is shifted to match the desired active region (which is centered at the camera location) in order to update the clipmap. The render region is the region of terrain in between the current active region and the next active region. Figure 2.6 shows an example of these regions from the paper. Each clipmap level contains an associated texture image, the normal map, which is used for shading the terrain. The normal map is computed from the geometry whenever the clipmap is updated. Like the vertex buffers, the textures are accessed toroidally for quicker updates.

For each frame, the implementation “determines the desired active regions, updates the geometry clipmap, and crops the active regions to the clip regions and renders” [19]. The desired region is view-dependent, meaning it relies on the viewer or camera position. For each level, the desired active region is a square whose size is $n * \text{gridspacing}$ by $n * \text{gridspacing}$ centered at the location of the viewpoint. The desired clipmap region is always re-centered around the camera. As the camera moves, the clip regions also move accordingly. Due to their implementation of toroidal access, old data is not copied when shifting a level. Only newly exposed regions are filled with height and color values. Rendering the terrain happens by clipping the active regions to their respective clip regions (the extent to which the geometry will be generated). The sharpest terrain is rendered in the grid centered around the camera, moving

away from the camera into higher levels of the pyramid the terrain becomes coarser, less refined. The 2D toroidal access requires CPU recomputation of vertex indices every frame, which can become CPU intensive. Hardware mipmapping is disabled and the authors chose to perform LOD on the texture for each level based on the same regions that were applied to the geometry. This way the texture LOD is based on viewer distance. View-frustum culling is implemented for geometry clipmaps and “reduces the rendering load by a factor of three for a 90 degree field of view” [19]. The implementation of geometry clipmaps used by this thesis follows the grid and layers scheme and toroidal access of data for faster updates [8].

2.3 Geometry Clipmaps Using Vertex Textures

Another implementation of terrain generation and LOD control in terrain using vertex textures is proposed in GPU Gems [22]. The original implementation uses vertex buffers, but not all graphics application software interfaces (APIs) and graphics processing units (GPUs) have the ability to modify vertex buffers, so central processing unit (CPU) intervention is sometimes necessary. The use of vertex textures alleviates this CPU intervention. In the original version, each level has a grid of geometric samples. For this implementation, the geometry of the samples is split into two parts - “the (x,y) coordinates are stored as constant vertex data, and the z coordinate is stored...as an elevation map” [22]. There is an n by n elevation map texture defined for each clipmap level. The textures are updated as the clipmap levels shift with the camera. The vertices obtain their elevations by sampling the elevation maps as a vertex texture - meaning that it is done in the vertex shader and per vertex as opposed to per pixel. Using textures allows “direct processing using the GPU rasterization pipeline,” which greatly improves performance [22]. Instead of cropping levels of terrain that were not fully updated (as the original implementation does), this GPU

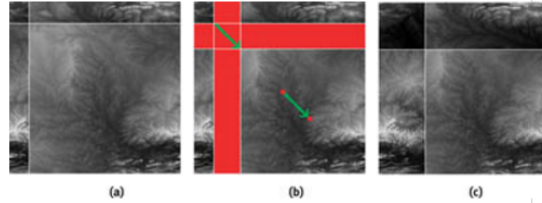


Figure 2.7: From GPU Gems [22], example of toroidal access and update

implementation ensures that a level is either fully updated or sets it as inactive so that only fully active regions are rendered. The number of active regions is based on the height of the viewpoint over the terrain because as the height of the camera over the terrain increases there is less of a need for dense or very refined terrain below the camera. As the camera moves, each clipmap area translates within the pyramid to stay centered at the camera. Accesses into the textures are done toroidally, following the original geometry clipmaps implementation. Figure 2.7 shows an example from GPU Gems showcasing how toroidal access is used in updating the different regions.

The normal map is updated by using the elevation map at the same level in the terrain pyramid and computing the normal as the cross product of two grid-aligned tangent vectors. This process is done in a shader. The shader also does a texture lookup to gather the normal from the corresponding coarser level. The elevation map and toroidal access aspects of this implementation are a part of the geometry clipmaps implementation used by this thesis.

Chapter 3

RELATED WORKS

The “need for maps has increased enormously in all parts of the world, as a consequence of.....the growing number of geospatial relationships, greater human mobility, and more physical planning problems brought about by a more intensive use of land and water” [23]. Maps are made and used now more than ever before due to technological advancements. Today, cartographers are not necessarily producing the maps, but they are moving into more “preparatory and advisory roles,” in which they utilize their expertise and knowledge in helping to develop the software generating the “cartographic models” [23]. One of the most important pieces of information that cartographers can help share is their understanding of map use by conducting “map use research” [23]. MacEachren defines visualizations as “a new field in cartography, made possible by technological developments, and mostly taking place in the environment of geographic information systems (GIS)” [23]. Kraak and Ormeling broaden the definition by stating that a visualization can be used to “explore unknown and often raw data, to analyse or manipulate known data, and to present or communicate knowledge of spatial information” [23]. There are four map use goals defined by MacEachren and Kraak. These four goals are to explore, analyze, synthesize, and present. The paper depicts an image of a “map use cube” shown in Figure 3.1, which has visualization determining characteristics for the “three main situations for visualizing data in a GIS: present, analyze and explore” [23]. Later, MacEachren and Kraak discuss a concept called geovisualization “which integrates methods and procedures of scientific visualization, cartography, image processing, information visualization, exploratory data analysis and geographic information science” [23].

Van Elzakker, specifically focuses on the exploratory cartography and researching

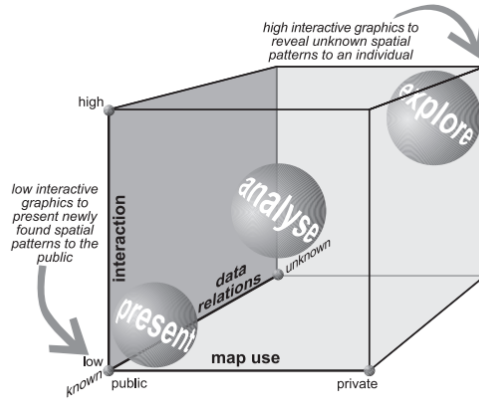


Figure 3.1: Example of the map use cube to explain characteristics for the three main visualization goals [23]

map use in such a visualization. “Exploratory cartography for geographic problem-solving may be regarded as a cognitive process in which maps are used as tools for discovery” [23]. Van Elzakker then states that true cartographic exploration is possible if maps can be “generated from raw geographic data in demand-driven ways” by the users with the help of software tools [23]. Map use research has been around since the 1950s. However, research and progress have come to a stand still [23]. This type of research is still important today. For exploratory cartography, it is important to conduct this type of research because it is important that the map can either answer a specific geographic question or meet an objective of the user [23]. Map use research in the field of exploratory cartography is limited, which means that there are still a lot of unanswered questions such as why users generate a map, how they generate the map, and when they generate the map for “geographic problem-solving in an exploratory cartography environment” [23]. The research conducted by van Elzakker aims at learning about the what, why, how and when questions regarding peoples’ geographic problem-solving techniques for the ability to improve upon tools for exploratory cartographic visualization.

This thesis aims at creating a system for exploratory cartographic visualization. This project falls under exploratory cartographic visualization because of the aim to

bring together raw data from different sources to generate a visualization that will provide users with answers to questions such as: How do I get from Los Angeles to San Francisco? Where can I stop for coffee in the middle of the route? Are there any interesting activities to do in the destination area? There is also an exploratory aspect of the project because Digital Elevation Models are used to showcase the terrain of the route. The hope is to entice users to want to explore the area through the visualization to gain geospatial knowledge. The map use research ideas in this research project are used in determining types of validation questions for this thesis as will be explained in the Validation chapter.

3.1 LineDrive

LineDrive was an application built as a system for automatically generating and rendering route maps [1]. Route maps have become a common form of graphic communication and thus it is important that they be portrayed correctly [1]. People use maps for directions, finding nearby restaurants, coffee shops and gas stations.

When designing a map, cartographic generalization techniques such as distortion, simplification and abstraction are used to bring out the most important aspects of a map [1]. Different types of maps will aim to emphasize different features, and thus design choices change. Computer-generated maps tend to be more exact and tend to contain more information, which makes them more difficult to process [1]. One problem with automatically generating route maps is that the systems in place do not differentiate between important and extraneous information, which prevents the use of generalizations that can be applied in hand drawn maps [1]. Extraneous information can include anything from names and locations of cities to parks or landmarks and roads far away from the route. This extraneous information can make it difficult to find the information necessary to navigate along the path. These types of “distortion

and abstraction techniques” are generally considered to be within the realm of non-photorealistic rendering. In LineDrive, however, they are applied to cartography. The LineDrive paper states its two major contributions as the generalization techniques used on route generation, and an automatic system for route map generation. The three types of generalization their implementation uses are: length generalization, angle generalization, and shape generalization.

Although LineDrive was innovative in its approach to mapping, the final results were gave almost no geographical context to the user. The user knew the start and end locations and was able to determine navigation easily by looking at the resulting map. However, travel within a state looked the same as cross-nation traversal. Figure 3.2 is an example of a final rendering of the LineDrive application. While only the bare minimum is shown for the purpose of making a more generalized version of a path and making it simpler for the end user to process the information provided, geographical context is important to understanding path information.

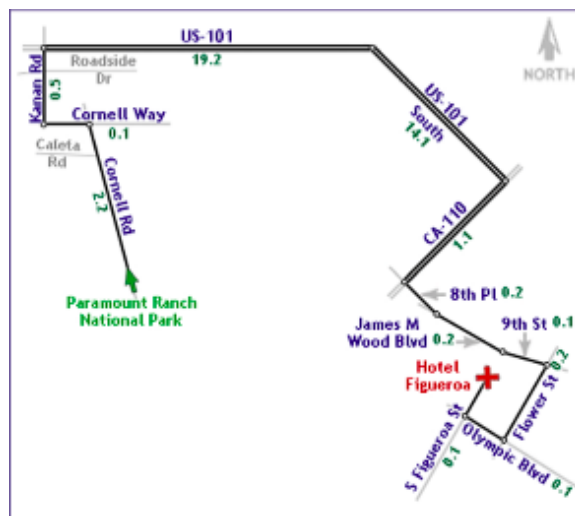


Figure 3.2: LineDrive Final Result

This thesis aims to build a system that will showcase only the necessary elements of a route, points of interest along the route, geographic and geospatial elements to

give the user more context. Similar to LineDrive, this system provides the user with only the essential information. LineDrive was such a great application because of its ability to convey to users directions with nothing but a path. LineDrive uses multiple algorithms for determining placement of road names along a route. This ensures that roads are always clearly labeled for the user. LineDrive also provides distance markers for each road. TRPINav does provide road names, but the placement is not as well-developed. Instead, TRPINav puts more focus on adding geographical and geospatial information into the map for user reference. State and county borders are helpful to users by giving them an idea of distance without explicitly stating it.

3.2 Destination Maps

Destination maps are designed for navigational purposes, to help users reach a desired location [17]. The Destination Maps system proposed novel algorithms for establishing and rendering road networks. They have many different layouts for the maps ranging from informal to formal. Destination Maps writers argue that mapmakers are faced with two challenges - road selection and road layout. They state that they present the first system for automatically generated maps that deals with road selection and road layout challenges. The system lays out the entire map network in three steps by: generalizing the selected roads, manipulating position, scale and orientation of the roads for legibility, and adding geographical context. [17]. The final look of the map is that of a hand-drawn map.

“Well-designed destination maps emphasize roads that facilitate navigation, while minimizing extraneous details” [17]. There are five design principles used during the creation of Destination Maps: hierarchical navigation; complete traversable routes; intersections at decision points; geometry and topology of roads; simplification of highways and interchanges. These principles allow the system to create images shown

in Figure 3.3 and Figure 3.4.

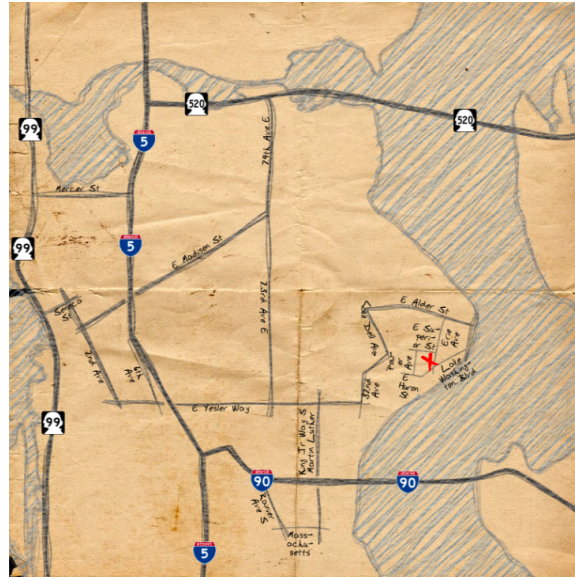


Figure 3.3: Destination Maps, Seattle in Treasure Map Style



Figure 3.4: Destination Maps, Chicago in Sketchy Style

Destination Maps was a great application for its generation of road networks. All major roads chosen for a route were clearly labeled and visible to the user. Adding in geographical context also allowed users to differentiate between regions on the map.

This thesis aims to expand the amount of context on the map. The TRPINav system includes elevation information which will allow the user to see the terrain they will be driving through, so that they better know what to expect during the drive. With generation of terrain, the user will be able to see bodies of water, much like the Destination maps, but the user will also be able to see more mountainous regions along their route.

3.3 Points of Interest

Yelp is a tool that was created to help people find businesses such as restaurants, coffee shops, clothing stores in different locations [25]. Yelp is able to locate specific types of businesses in the area close to where the user is. A map can be shown of the location so that the user can then use Apple Maps, Google Maps or their preferred choice of navigational tool. Currently there is really no way of finding local businesses or points of interest along a path, or ahead of where you are. The user must perform an extra search to determine if there are any restaurants, gas stations, national parks, etc in any of the cities that they will be travelling through. As a user this can be an inconvenience.

Google also has a tool for determining businesses near a user. These locations are anywhere within some radius around the user. Recently Google Maps for Android and iOS has been updated with a points of interest search allowing users to look for coffee shops, gas stations, restaurants and grocery stores [11]. Users are also able to search for other types of points of interest, those mentioned are the quick look up options.

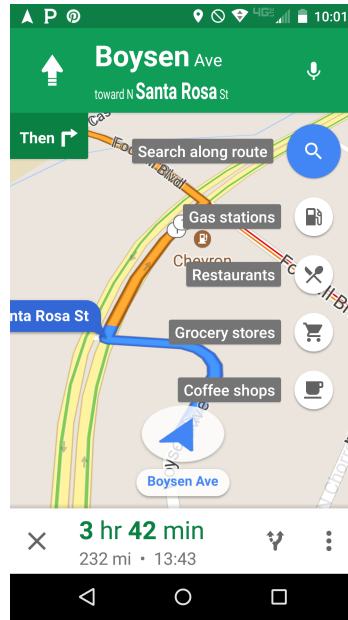


Figure 3.5: Navigation view of Google Maps allowing users to choose quick points of interest

Looking at the maps in Figure 3.5 and Figure 3.6, it can be seen that there is a lot of extraneous information that can be difficult for users to process.

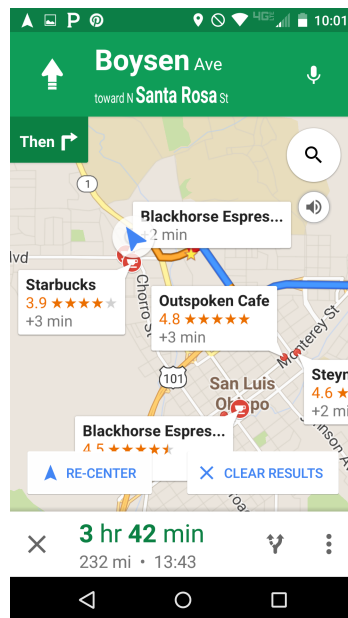


Figure 3.6: Google Map in navigation view overlaid with coffee shops

Garmin also has a global positioning system (GPS) navigation system that provides users with the ability to show gas stations and other points of interest along the route. To do so, users need to download specific types of points of interest maps from the website for their specific device and then download those maps to the device [20]. This process can take some time and requires users a number of extra steps.

This thesis aims at giving users only the most important information to their path in combination with points of interest and 3D terrain data. By doing so, the goal is to allow users to explore their surroundings without overloading users and making it difficult for them to process the map.

Chapter 4

IMPLEMENTATION

TRPINav is a system created to unify information found for terrain, routes and points of interest. This interactive system is intended to provide the user with route information in an aesthetically pleasing and intuitive manner. The system has a top-down view and route following view, shown in Figure 4.2, incorporating points of interest, from Yelp and Google, and elevation data from Google Earth Engine. The system is designed in such a way that data is gathered, processed and then rendered to the screen. This system design is outlined in Figure 4.1. Implementation of this project waTs done on a Mac OS X, using C++ and a graphics engine that uses OpenGL under the hood [8].

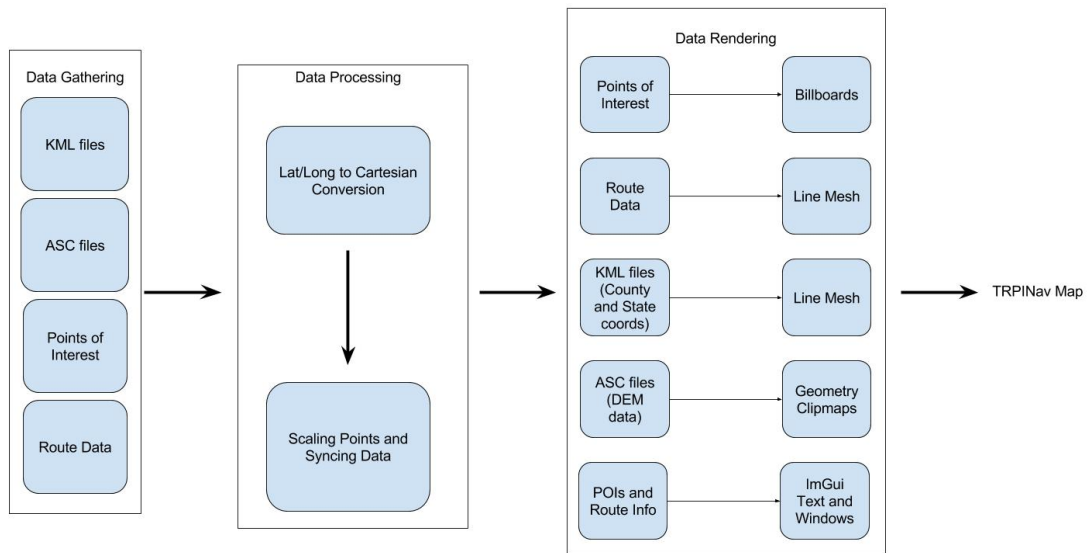


Figure 4.1: TRPINav System Overview

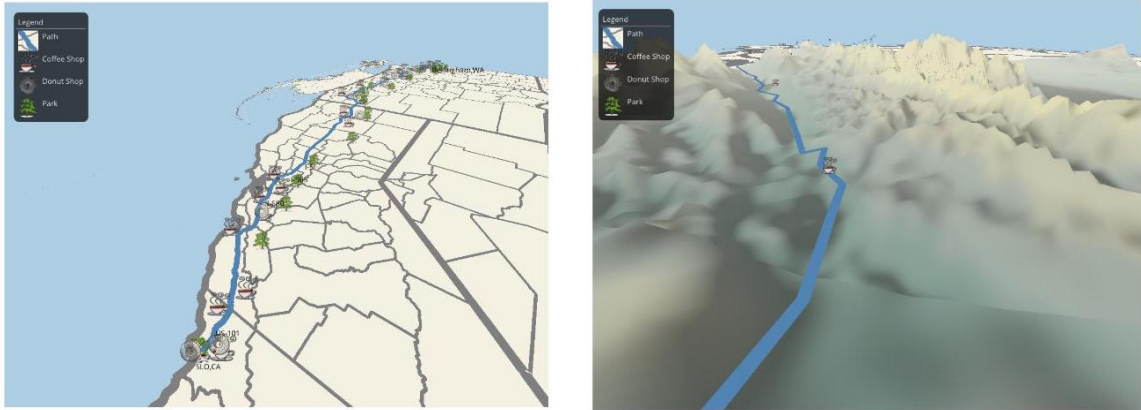


Figure 4.2: TRPINav System Overview

4.1 Data Gathering

The first step in the system is data gathering. This step involves obtaining all of the information the system needs in order to render a map. This information includes geospatial and topographical data describing the world, the path from source to destination, and interesting and helpful stops along the way. The majority of the data gathered is in latitude, longitude and altitude format. These values can come from many different types of geodetic datums. The datum used in Google and for this thesis project is WGS84. Each datum describes the x,y,z-axes, size, orientation and origin of the world differently.

4.1.1 KML Data

A critical part of the TRPINav system is a visual representation of regional, state, and national lines. These lines give context to the user and the path and provide the basis for the entire map. In order to generate a mesh visualization, the system requires input describing these boundaries in the form of Google KML (Keyhole Markup Language) files. These files contain information about latitude, longitude and

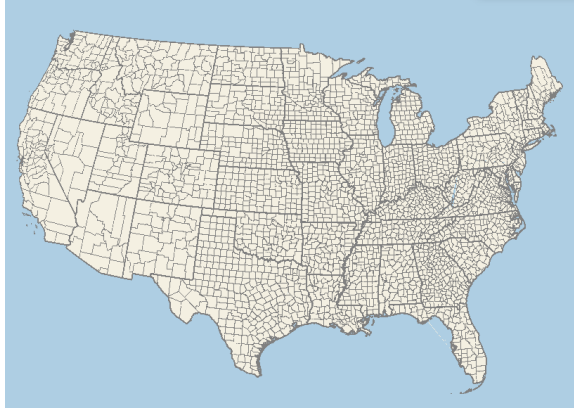


Figure 4.3: Example of the rendered KML data

sometimes altitude. These values are found in the file as `<coordinates>` tags that are a part of `<LinearRing>`, `<outerBoundaryIs>`, `<Polygon>` and `<MultiGeometry>` elements. For the purpose of this system, there are two such files that are parsed for information. The first is a KML file containing the outlines of each state. The second is a KML file containing outlines of each county in each state. Figure 4.3 shows an example of the states KML file.

The state file contains `<Placemark>` elements where each `<Placemark>` `</Placemark>` pair within the file represents a different state. There is some information about the data for each state and then the `<MultiGeometry>` elements begin meaning that there is more than one `<Polygon>` or other type of Geometry that can be found for the `<Placemark>`. Within the `<MultiGeometry>` element there can be found `<Polygon>`, `<outerBoundaryIs>` and `<LinearRing>` elements. The coordinates for each state are found within the `<LinearRing>` `</LinearRing>` pair. The county file contains `<Placemark>` `</Placemark>` pairs that represent each county. Each county is a single `<Polygon>` `</Polygon>` pair with `<outerBoundaryIs>` and `<LinearRing>` elements inside. Much like the state file, the coordinates are found within the `LinearRing` element. To parse these files a library called `libkml` provides an interface to loop through all the `Placemark` elements in the file and gather coordinate

information.

Each state is represented by a C++ `std::vector` of type `Point` containing all of the coordinates found in the `LinearRing` element. The `Point` object contains latitude, longitude, and elevation information for each coordinate in the file. All states, vectors of type `Point`, are gathered into another vector. State counties are also represented in this manner. The points are now gathered and ready for processing.

4.1.2 ASC DEM Data

TRPINav renders three-dimensional geometries via a terrain mesh in the navigation view. Providing elevation information will hopefully draw a user's attention to their surroundings and make them want to explore. To produce the mesh object, TRPINav requires elevation information in the form of Digital Elevation Model (DEM) files. Google Earth Engine has many different types of National Elevation Datasets (NED) available for downloading. This thesis uses geophysical terrain data from Google Earth Engine. The format chosen for the terrain data is an ASCII grid format that contains all of the elevations at some resolution in a grid representing the area. The file contains a header with information about the grid size (rows and columns), the latitude and longitude of the lower left corner of the grid, and the size of each cell. This information is used in combination to index into the grid at a specific latitude and longitude.

DEM files can be found at many different resolutions. This thesis uses a National Elevation Dataset (NED), which is "seamless" terrain data for the United States that is derived from DEMs. There are multiple resolutions for this type of data: 1 arc second, 1/3 arc second, and 1/9 arc second. Those resolutions map to 30 meters, 10 meters, and 3 meters respectively, meaning that coverage of the area ranges from complete data to sparse data [10]. The resolutions mentioned for each DEM refer

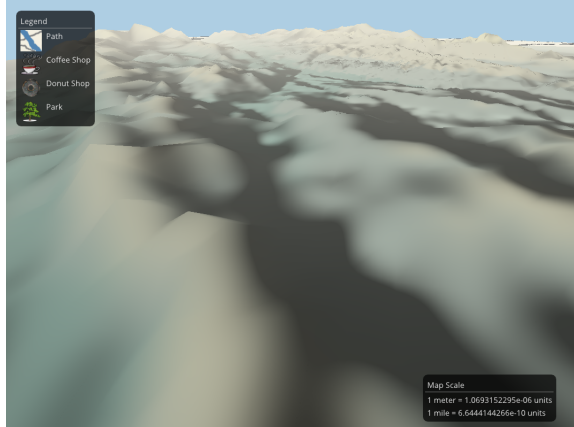


Figure 4.5: Final rendering of the DEM data in the system

with the grid data from the file. Access of data is done using information from the header. Figure 4.5 shows a final rendering of the DEM data in the system.

4.1.3 Points of Interest Data

4.1.3.1 Yelp API

This thesis uses Yelp to gain information about local businesses along the path. We felt that finding coffee and food would be important to users on a long drive. Eventually users will have the option to determine which points of interest they want, but for the purpose of this prototype we felt that coffee and donuts would encompass the needs of most people. At approximately every five data points along the path, there is a query to the Yelp Search API to determine if there are any coffee shops or donut shops within a five mile radius. The query to the API involves setting a search limit, search term (coffee or donuts in this case), and a radius. A Yelp-provided OAuth1.0 API key is required for performing searches with the Yelp API. The OAuth1.0 details are handled by a library called liboauthcpp. The system provides an API URL which liboauthcpp then uses to authenticate with the Yelp API. This method returns an API token and user key. The query to the API involves

using an HTTP request, so a library called libcurl is used to perform any API calls. A curl handler is set up with a callback function to return the size of the result of the API call. If the curl call failed, an error is printed to the screen. Otherwise the data is saved into a variable for parsing.

```
"businesses": [
  {
    "categories": [
      [
        "Donuts",
        "donuts"
      ],
      [
        "Bagels",
        "bagels"
      ]
    ],
    "display_phone": "+1-805-782-9766",
    "distance": 1622.4026583058778,
    "id": "slo-donut-company-san-luis-obispo",
    "image_url": "https://s3-media3.fl.yelpcdn.com/bphoto/Uc8NG_aRSLmKs9bMW9E6EA/ms.jpg",
    "is_claimed": true,
    "is_closed": false,
    "location": {
      "address": [
        "793 E Foothill Blvd",
        "Unit F"
      ],
      "city": "San Luis Obispo",
      "coordinate": {
        "latitude": 35.2935834865723,
        "longitude": -120.672041138042
      },
      "country_code": "US",
      "display_address": [
        "793 E Foothill Blvd",
        "Unit F",
        "San Luis Obispo, CA 93405"
      ],
      "geo_accuracy": 9.5,
      "postal_code": "93405",
      "state_code": "CA"
    },
  },
]
```

Figure 4.6: One of the businesses returned by Yelp Search API for Donut Shops in San Luis Obispo, CA

The data from the Yelp API call is returned as a JavaScript Object Notation (JSON) object. This thesis utilizes a library called libjsoncpp to parse JSON objects into native C++ data structures. Figure 4.6 shows an example of the JSON object returned by the API call. Figure 4.7 shows the final rendering of only Yelp data

within the system.

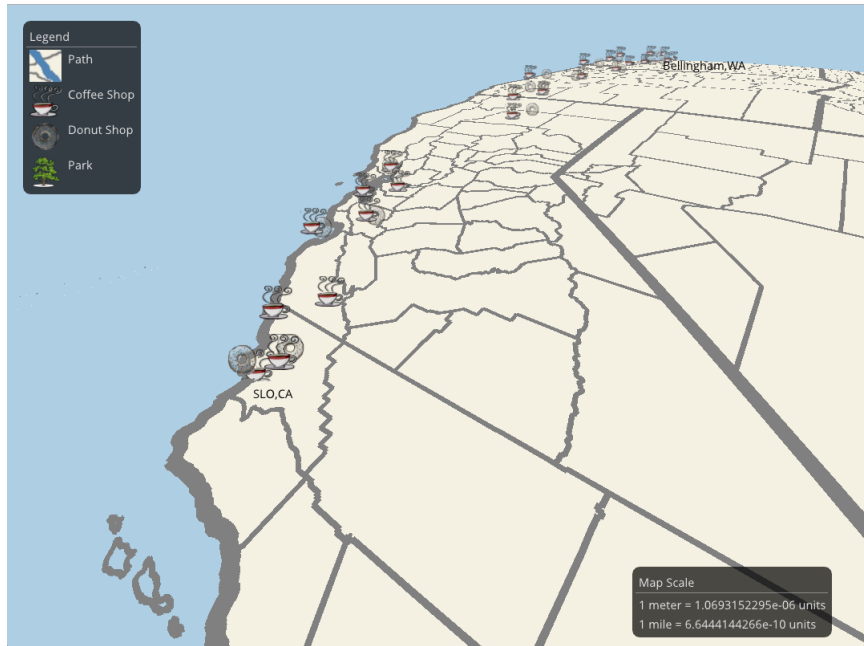


Figure 4.7: Final rendering of Yelp data within the system

4.1.3.2 Google Places API

The Google Places API is used to gather information about parks along the route. Parks are a great place to stop on long drives to get fresh air, stretch legs, or eat a packed lunch during a long drive. We hoped that by adding parks users would be even more interested in exploring the surrounding areas.

The Google Places nearby search is used to find national and state parks within a radius around a location. The URL used for the call accepts parameters for a location in latitude and longitude coordinates, the search radius, type of Point of Interest (specified in a list of accepted types), and keywords. If the call succeeds, it returns a JSON object that is parsed to get information about the location, name, and vicinity of the park. The vicinity for the nearby API call is an unformatted address of the area. The vicinity includes the street number and street name and the city

information. Figure 4.8 provides an example response from the Google Places API. Figure 4.9 shows the final rendering of only Google Places data within the system.

```
{
  "html_attributions" : [],
  "results" : [
    {
      "geometry" : {
        "location" : {
          "lat" : 42.2062219,
          "lng" : -122.714733
        }
      },
      "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/generic_business-71.png",
      "id" : "5797ff25892c7a094123b4010582df924ae036ec",
      "name" : "Crater Lake National Park Trust",
      "opening_hours" : {
        "open_now" : false,
        "weekday_text" : []
      },
      "place_id" : "ChIJq0bmIkVzz1QR-h84ZkHYg4U",
      "scope" : "GOOGLE",
      "types" : [ "park", "point_of_interest", "establishment" ],
      "vicinity" : "100 East Main Street, Medford"
    }
  ],
  "status" : "OK"
}
```

Figure 4.8: An example of a park returned by the Google Places API

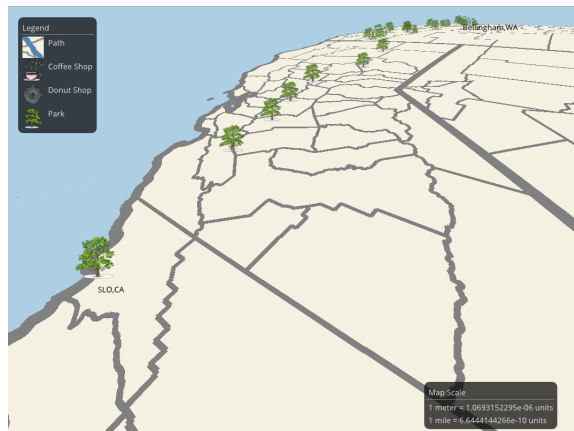


Figure 4.9: Final rendering of parks within the system

Data for all parks, coffee shops, and donut shops is stored in a map where the key is the type of point of interest and value is a vector of POI objects of that type. This provides faster access to all parks, coffee shops, and donut shops when rendering the points of interest or showing a list of the points of interest to the user. Information

such as name, location, address, and type are stored in the POI object. The name and address are displayed to the user through a menu list on the screen, and the location is used for placement during rendering.

4.1.4 Route Data

This thesis utilizes multiple Google APIs for path generation and road labeling. This provides the user with the necessary information to get to their destination. Similar to points of interest, calls to the Google Maps APIs are done through the use of libcurl. The Google Directions API “is a service that calculates directions between locations using an HTTP request” [6]. The source and destination locations are sent as either latitude, longitude coordinates or as strings in an Address, City, State format. If the Address is omitted, Google does an approximation on the location within the city that the user chooses. The user has two choices for response format. This thesis uses the JavaScript Object Notation (JSON) format. An example of this response is shown in Figure 4.10.

```

"geocoded_waypoints" : [
  {
    "geocoder_status" : "OK",
    "place_id" : "ChIJyYfhZ79ZwokRMtXcL6CYxkA",
    "types" : [ "premise" ]
  },
  {
    "geocoder_status" : "OK",
    "partial_match" : true,
    "place_id" : "ChIJ8YWMWnz4wokRCOVf1CcJCbY",
    "types" : [ "street_address" ]
  }
],
"routes" : [
  {
    "bounds" : {
      "northeast" : {
        "lat" : 40.8171321,
        "lng" : -73.99449150000001
      },
      "southwest" : {
        "lat" : 40.7416627,
        "lng" : -74.0728354
      }
    },
    "copyrights" : "Map data ©2015 Google",
    "legs" : [
      {
        "distance" : {
          "text" : "9.7 mi",
          "value" : 15653
        },
        "duration" : {
          "text" : "25 mins",
          "value" : 1480
        },
        "end_address" : "1 MetLife Stadium Dr, East Rutherford, NJ 07073, USA",
        "end_location" : {
          "lat" : 40.814505,
          "lng" : -74.07272910000002
        },
        "start_address" : "75 Ninth Ave, New York, NY 10011, USA",
        "start_location" : {
          "lat" : 40.7428759,
          "lng" : -74.00584719999999
        },

```

Figure 4.10: Directions from New York to New Jersey JSON response

There is a field in the response named `overview_polyline`. This is a base64 encoded string of all of the points along the route represented in latitude and longitude.

```

"overview_polyline" : {
  "points" : "_rtwFpgubMtBuGVNFPfAr@TNIVkB|FsBpG_C_BqA}@mGeEwH_FsDcCaGwDkNiJsJmG_SoMyByAzA}EP
i@nA{Dp@qBtDdCLh@vCT_@nAKLMB0AIG}@{Aq@eAWWOMIEWEY?UDYJi@\\]^w@|AcC|Ew@bBy@hCg@dBQ`AU~A_ChH
_D|J_AzCc@rAwXl|@yLt_@k@hCSrBATABp@LlALp@FPZ~@Tf@Z^X^jAx@PXRT rCxB~@`@lA^`@HVLv@j@n@h
@`@Rb@dL@G|@wVQTU^w@PaA@gAKcAMa@Q}c@i@}@g@iDoAeDaAeDq@qC}i@Cg@eAT[L[Vk@h@SZQ^_@bAeJ`XoAbEm
B~GqAvDkDzIsFvNcAvCcFrRkBNhCE`Pq@~Bi@rAk@hA}@hA_@f@cAfAoDjDiAv@iB|@a@XoAf@iBj@yA`@aB\\{Dz@y
Cl@yA\\mE`A}Aj@qCjAuC~AkBnA_@ZcBxAoApA{BhCoBtCuJrOkAxAkBzCeH-KKAlBeMrR{I|M};CtEyBjD}EpHmLbQ{
BdDiBvCUBMJy@l@g@VeAZs@HiABuKT_BEiBUaEkAcA}oCeAmAi@iCsAyCkBgEaDcCuBa@}kHgkDyCQ000G_@Fk@h@Y
fI_DxAi@vAge^`@h@l@h@dLev@Ef@0d@m@fAy@nBmAxCc@dAALAHQf@Qv@G^Et@n@Fr@Z`BHLABnAp@Cj@Q"
},

```

Figure 4.11: Overview Polyline Encoded String

This string is then decoded as defined by the Google API documentation. All of the points for the route are stored in a vector of type Point, for use later in the data gathering and data processing stages. An example of the encoded string can be seen in Figure 4.11.

The Google Maps Geocoding API is used to do reverse geocoding, which is the “process of converting geographic coordinates into a human-readable address” [7]. This API is used to get street information every five points along the path. An example of the JSON object returned by the API call is found in Figure 4.12. A libcurl call is made to the API with a URL containing the Google Developer Key and the point to reverse geocode in latitude and longitude. The results received from the call are parsed through the address_components to search for a component of type “route.” If it is found, the route name is saved to one of the path points from the Google Directions API.


```

{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "1600",
          "short_name" : "1600",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Amphitheatre Pkwy",
          "short_name" : "Amphitheatre Pkwy",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Mountain View",
          "short_name" : "Mountain View",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "Santa Clara County",
          "short_name" : "Santa Clara County",
          "types" : [ "administrative_area_level_2", "political" ]
        },
        {
          "long_name" : "California",
          "short_name" : "CA",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "United States",
          "short_name" : "US",
          "types" : [ "country", "political" ]
        },
        {
          "long_name" : "94043",
          "short_name" : "94043",
          "types" : [ "postal_code" ]
        }
      ]
    }
  ],

```

Figure 4.12: Geocoding JSON Response Object

4.1.4.1 Google Maps Elevation API

Google’s Elevation API “provides a simple interface to query locations on the earth for elevation data,” including “sampled elevation data along paths, allowing you to calculate elevation changes along routes”. There are two approaches to using the API - as a set of one or more locations, or as a series of connected points along a path. Both approaches use latitude and longitude coordinates. Output format is specified in the URL, this thesis uses the JSON format. An example of the response is shown in Figure 4.13.

```

{
  "results" : [
    {
      "elevation" : 4411.941894531250,
      "location" : {
        "lat" : 36.5785810,
        "lng" : -118.2919940
      },
      "resolution" : 19.08790397644043
    },
    {
      "elevation" : 1381.861694335938,
      "location" : {
        "lat" : 36.41150289067028,
        "lng" : -117.5602607523847
      },
      "resolution" : 19.08790397644043
    },
    {
      "elevation" : -84.61699676513672,
      "location" : {
        "lat" : 36.239980,
        "lng" : -116.831710
      },
      "resolution" : 19.08790397644043
    }
  ],
  "status" : "OK"
}

```

Figure 4.13: Elevation API JSON Response Object for a path between two points with three samples

Originally the Elevation API was used to incorporate elevation to the route positions. However, since the DEM files could be pre-processed, elevation information is sourced solely from the DEM. Code is still in place to use the Elevation API should there be a need for it in the future.

4.2 Data Processing

Once all of the data has been gathered, it is processed. This data processing stage consists of data conversion from latitude and longitude to Cartesian coordinates, then data scaling.

4.2.1 Data Conversion

Input data is provided in latitude and longitude format and must be converted to Cartesian prior to rendering. Converting to Cartesian coordinates is not trivial. There are different datums used for latitude and longitude points, each needing a different conversion algorithm. Another difficulty is that the Earth is not a perfect sphere, it is more of an ellipsoid so the minor axis and major axis must be taken into consideration. For the purpose of this thesis, all latitude and longitude points are of datum WGS84.

All of the vertices used within the map are converted from WGS84 Latitude Longitude coordinates into Cartesian coordinates. An algorithm using the radius of the earth, the eccentricity of the ellipsoid that is the earth, and the transverse radius of the curvature is used to transform the points. An example of these values can be found in Figure 4.14. Once those are computed all of the vertices are normalized to a range $[-1, 1]$. Typically in small-scale graphics environments a $[-1, 1]$ vertex range is sufficient for rendering objects to a screen. This scale for rendering is used due to legacy code, and is not the best chosen scale for this type of application. The vertices were scaled to a much smaller range to make sure data was being unified properly. The Future Work chapter will talk about scaling the system better as a whole and some of the challenges with simply changing the scale factor.

$$e^2 = (a^2 - b^2) / a^2$$

$$v = a / \sqrt{(1 - e^2 \cdot \sin^2 \varphi)}$$

$$x = (v+h) \cdot \cos \varphi \cdot \cos \lambda$$

$$y = (v+h) \cdot \cos \varphi \cdot \sin \lambda$$

$$z = ((1-e^2) \cdot v + h) \cdot \sin \varphi$$

```

/**
 * Ellipsoid parameters; major axis (a), minor axis (b), and flattening (f) for each ellipsoid.
 */
LatLon.ellipsoid = {
  WGS84:      { a: 6378137,      b: 6356752.31425, f: 1/298.257223563 },
  GRS80:      { a: 6378137,      b: 6356752.31414, f: 1/298.257222101 },
  Airy1830:   { a: 6377563.396,  b: 6356256.909,  f: 1/299.3249646  },
  AiryModified: { a: 6377340.189, b: 6356034.448,  f: 1/299.3249646  },
  Intl1924:   { a: 6378388,      b: 6356911.946,  f: 1/297          },
  Bessel1841: { a: 6377397.155,  b: 6356078.963,  f: 1/299.152815351 },
};

```

Figure 4.14: Some of the values used to convert from WGS84 latitude and longitude to Cartesian

To scale the data, the max extent is determined. The extent is determined by finding the max x, y, z value and finding the min x, y, z value and then subtracting the min from the max. The max extent is then $(1 / \max(\text{extent.x}, \text{extent.y}, \text{extent.z}))$. Once the max extent is determined all points are scaled by that max extent.

Once all the vertices have been converted and scaled, they are redistributed into new data structures of type CartPt. CartPt is a class holding the x, y, z values of a vertex. At this point in the system, the data is gathered, transformed and ready for use.

4.3 Data Rendering

Once the data has been gathered and processed, the TRPINav system generates and renders visualizations of the data. This thesis utilizes a graphics engine called ionEngine. State management and the graphics pipeline for OpenGL is handled by

ionEngine, which simplifies design and implementation. As a consequence, immediate mode OpenGL is not available for use and older calls with modes such as `GL_LINES` or `GL_PLYGON` are not implemented.

4.3.1 Map Outline

The first attempt at rendering the data into lines and polygons used OpenGL immediate mode, with `GL_LINES`. Due to limitation with ionEngine, and the inability to define line width using `GL_LINES`, lines are represented by line meshes composed of triangles. Figure 4.15 shows an example of the state and county lines for the United States.

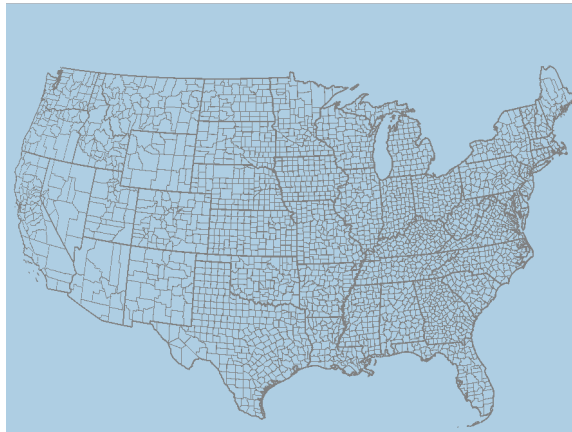


Figure 4.15: Image of the United States in mesh lines generated in ionEngine

As can be seen in Figure 4.15, the map is transparent due to only representing the United States with line geometry. In immediate mode, this was solved using `GL_POLYGON` to generate a solid polygon in the shape of the United States. However, `GL_POLYGON` is deprecated, so instead we chose to generate object (OBJ) files representing the different states. Mesh generation and writing to OBJ files is done through ionEngine. The algorithm for mesh generation runs in $O(n^2)$, so the number of vertices that it could generate a mesh for at one time became limited quickly.

Splitting up the state KML file became the best option, so there are multiple OBJ files that represent the entire solid area of the map. Placing mesh objects proved challenging. The general vertex scaling algorithm used throughout the application was not adequate due to the variance in the extent. The solution was to use the maximum extent from the application to scale the vertices of the mesh objects to the correct range and then write them to the file. Figure 4.16 shows an example of the map outlines with the state meshes in place.

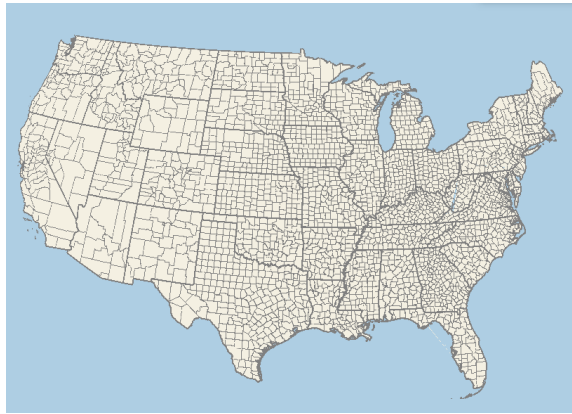


Figure 4.16: Image of the United States in mesh lines and solid mesh background generated in ionEngine

4.3.2 Route Generation

The path generation is very similar to that of the state outlines. The vertices are sent to a function, which generates a mesh that represents a line from the vertices. The path is lifted off of the map as it includes elevation information from the DEM files. During the gathering and processing stages, multiple DEM files are loaded through the extent of the route to retrieve the elevation information. During the processing stage, after conversion, the elevations for each point were scaled by the map scale and added to the y-coordinate of the path point to elevate the path. This elevated the path on top of the terrain during the navigation view.

The color chosen for the path is one that stands out against the darker outlines

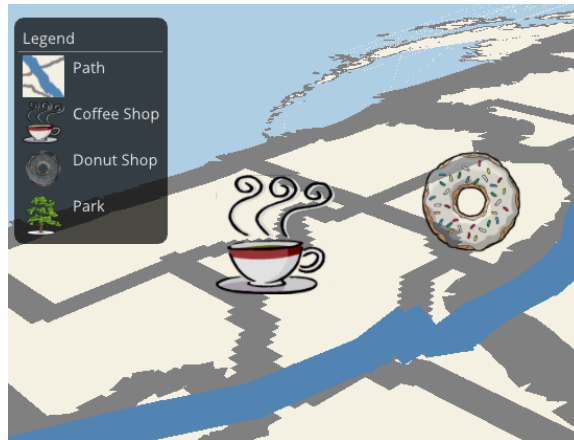


Figure 4.17: Coffee and Donut shops that can be found along the route rendered as sprites

and light background of the map. The width of the path changes depending on the view of the map. For a top-down view, the width of the path lies in between the width of the state and county lines. This is to ensure differentiation of the path from other lines within the map. For the navigation view, the width is a factor of ten smaller, because what is important is the elevation information and seeing the path on top of the terrain. The original width is too wide in this view.

4.3.3 Points of Interest

All of the parks and businesses that are pulled through Yelp and Google Places are visualized as billboarded sprites. Sprites are two-dimensional images representing objects in the scene. These images are always drawn as a rectangle and thus the “sprite usually encompasses transparent areas so it provides the illusion of a non-rectangular drawing” [18]. Image information is held in a texture that defines a grid of texels, which are the “smallest unit that can be stored by the GPU” [18]. Billboards are “texture[s] that [are] mapped to a special plane that is always perpendicular to the camera axis” [18]. This means that no matter which way the camera rotates or moves, the images are still facing the camera.



Figure 4.18: Parks that can be found along the route rendered as sprites

For each type of point of interest, a texture is created with an image to represent it. The images are in bitmap image (BMP) file format. This format includes a file header, color table, and image data. The image data is represented as a grid of pixels, each containing RGBA values. In the fragment shader, the texture is indexed into using texture coordinates for the current position to determine what color to set the pixels to. In the vertex shader, each vertex is transformed into world space using a projection matrix, view matrix and model matrix. To billboard an image, only the rows and columns pertaining to the plane chosen for the billboard are applied to the vertex. This will keep the image on that plane and continually facing the camera. Donut shops are denoted with a donut, coffee shops with a coffee mug and parks with a tree. Figures 4.17 and 4.18 showcase the billboarded sprite renderings.

One of the challenges that surrounded generating the sprites, as can be seen in Figure 4.19, is determining proper placement. Initially all sprites were drawn at their exact locations. However, this implementation produced areas with tight clusters of coffee and donut shops that caused overlapping and z-fighting. Overlapping occurs when too many sprites exist close together, or the sprites are too large. This also has to do with the scale of the sprites with respect to the scale of the map. If there

are three or more coffee shops on a street, given the scale of the map, the sprites all become intertwined and it is difficult to see any information. Z-fighting occurs when multiple objects have similar values in the z-buffer during rendering and runtime of the application. Z-fighting results in an interesting rendering of the objects where neither is in front and the two fight for placement in the same position in the world. To solve this, sprite locations are slightly altered before being added to a billboard object for rendering, and sprites which do not exceed a minimum distance from other sprites are excluded. This provides the user with knowledge of features in an area, but does not overload the user with an over-concentrated cluster on the map. Users are provided with a list of points of interest split up by type in a menu. This menu allows users to find the address and name of each point of interest.



Figure 4.19: Parks that can be found along the route rendered as sprites

4.4 Terrain Generation

Elevation data is used during the navigation view to generate a mesh that looks like the physical terrain. The DEM files are used to create a heightmap that is indexed into to find height elevations at different latitude and longitude locations. The implementation used for the terrain generation is a combination of two geometry

clipmaps implementations. The terrain is represented by a grid with multiple layers. These layers are rings of different sizes surrounding the innermost grid. The nearby terrain (terrain closest to the camera) is represented with a grid, and then a ring of terrain at half its resolution is placed around it, and rings are continually added. There is a linear increase in vertices that gives an exponential increase in visible area, which allows for high scalability. Also, a heightmap texture is used along with toroidal updates so that changing the view by a small amount only requires a small update to the heightmap. This implementation also reduces redundancy in values being thrown out of the heightmap to be replaced by others.

The interface to the clipmaps implementation within ionEngine is relatively simple because the implementation is also in latitude and longitude coordinates. Based on the location of the camera, the clipmaps object will ask for heights at a set of latitude longitude points for each layer. The position of the camera is retrieved from the camera object. The position of the camera needs to be projected down to the earth's surface for proper latitude and longitude coordinates. This is done by creating a vector from the center of the world (which is the origin at 0, 0, 0) to the current position of the camera. The current position of the camera is actually manipulated to be seem further in front of the actual camera. This is done to create more terrain further from the camera. Then the origin is moved by the radius of the Earth in meters multiplied by the scale of the map along the vector to get to a position on the surface of the Earth. This position on the surface of the Earth is then converted from Cartesian coordinates into latitude and longitude coordinates, at which point the object can ask for height values. To accentuate the elevations for the user, the heights at each latitude and longitude are exaggerated by a factor of seven.

One problem that is a consequence of faking a camera in front of the actual camera is a level-of-detail problem. Since the level-of-detail (LOD) is at the highest resolution about the position of the camera, faking a camera means that any fine LOD is seen

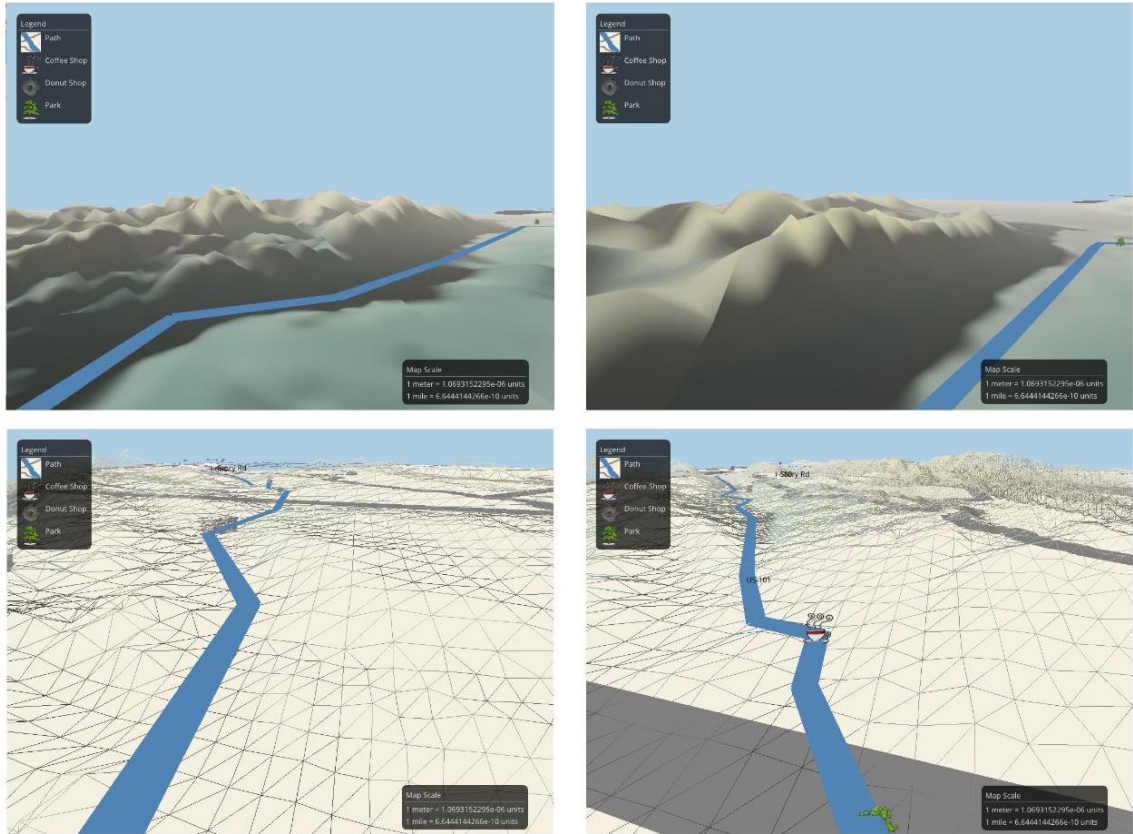


Figure 4.20: The left picture uses the actual camera, the right picture uses the fake camera. Highest resolution geometry is closest to the camera in the left picture, and further away in the right image

further along the path and the terrain closer to the actual camera becomes more coarse. This distinction can be seen in Figure 4.20.

Hypsometric shading is “assigning colours solely by elevation,” and is most common in “small-scale topographic maps” [15]. This type of coloring scheme is normally applied as a simple gradient or in different layers to represent subsets of data or levels of elevations. The terrain coloring scheme ranges from a layered coloring scheme to a gradient coloring scheme. The first coloring mode chosen is a typical blue, green, tan, brown, and white color scheme where white represents the tallest mountain and blue represents sea level. The second coloring mode is grey scale, so as the height increases the color nears the white point. The third coloring mode chosen is a two-tone scale

with grey-blue tones and tan to orange tones. Figures 4.21, 4.22, 4.23 show examples of the color schemes on terrain of differing heights.

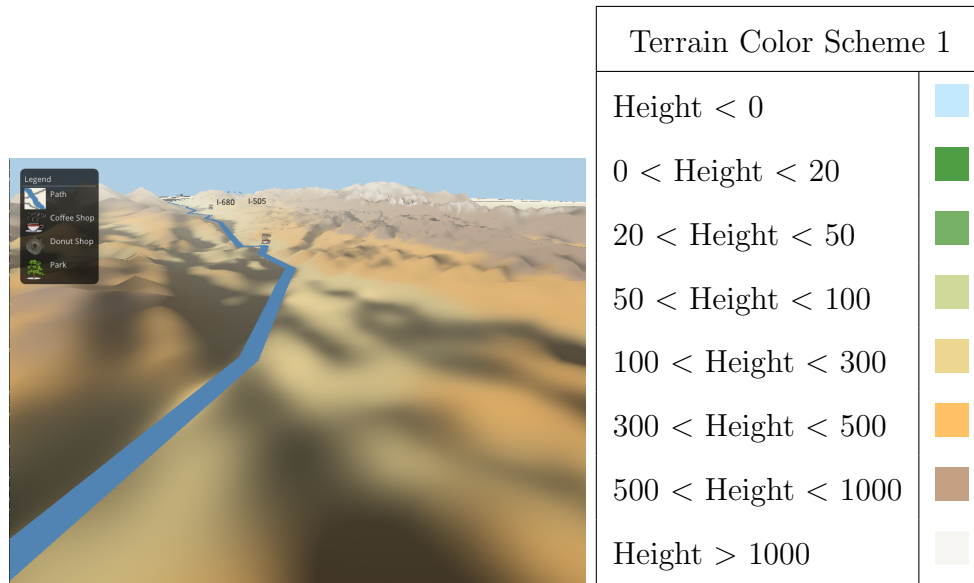


Figure 4.21: Most commonly found terrain color scale, can be toggled on with number ‘1’ key if color scheme changed

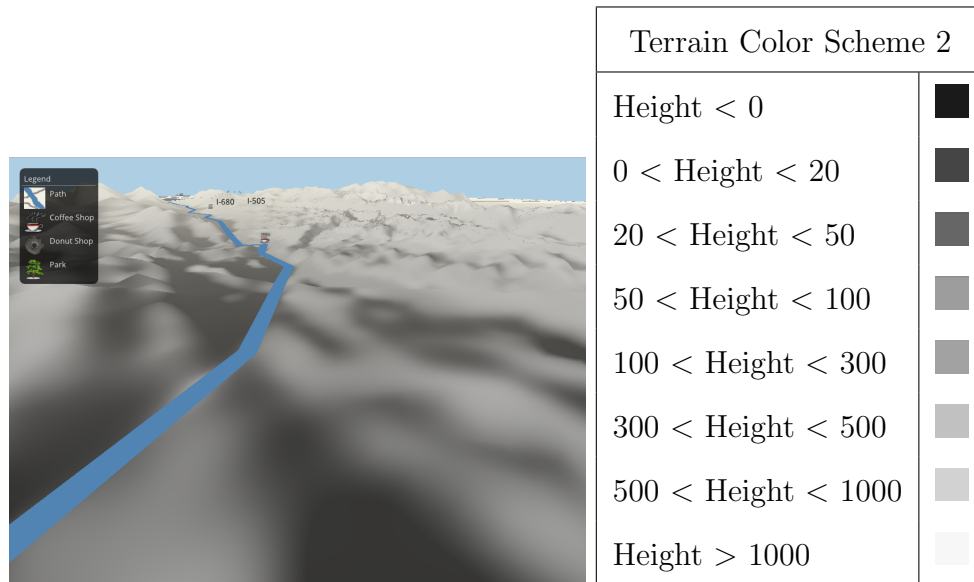


Figure 4.22: Greyscale terrain color, toggled on by pressing the number ‘2’ key

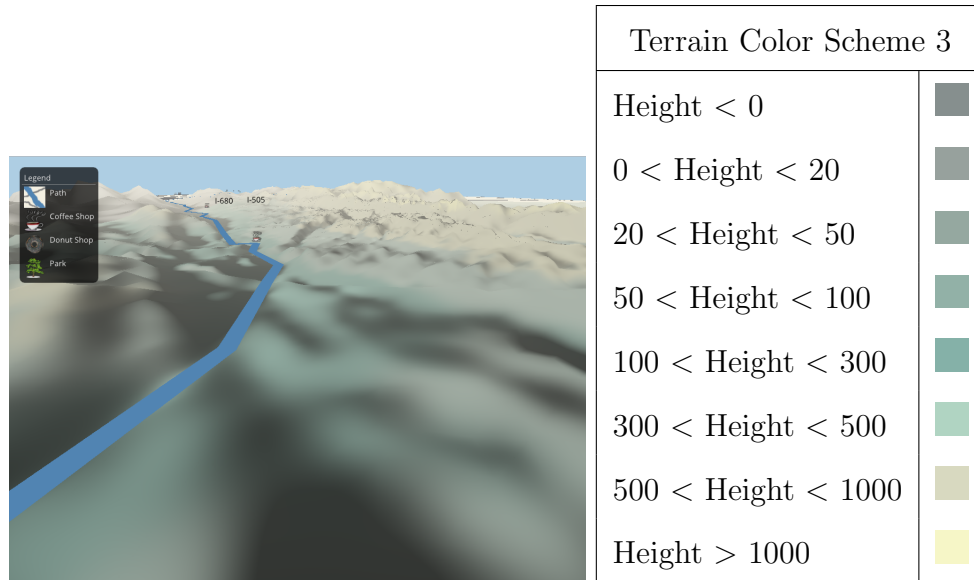


Figure 4.23: Two-tone terrain color, toggled on by pressing the number ‘3’ key

Since the geometry clipmaps terrain is added on top of the drawn map, there is a distinct line between the terrain mesh and the map. One way to fix this was to lower the camera down, closer to the ground. This produced an illusion of more terrain further off in distance. In addition, a fog layer was added to the geometry clipmaps so that as the distance from the camera increases the clipmaps terrain will blend in with the underlying map color. The fog is added in the fragment shader for the geometry clipmaps. It uses an exponential function to determine the thickness of the fog based on the distance from the camera. To determine the color of the current pixel an interpolation is done with the fog color and the pixel color based on the texture map using the fog amount value. This interpolation is done with `mix(vec3, vec3, float)`, a built-in OpenGL function that interpolates between two values.

4.5 Top-down View

The top-down view incorporates the points of interest, the route information and the KML data that provides the state outlines and map mesh background.

The top-down view of the map is the initial view when the map is rendered to the screen. Instead of the traditional perpendicular to the screen or parallel to the screen view that most maps use as top-down, due to the curvature of the Earth the top-down view is slanted at an angle. The angle allows the user to notice the curvature of the map. The top-down controls include the typical WASD controls found in many computer games. In this scheme, W will zoom in, S will zoom out, A will strafe left, and D will strafe right. In addition, using the up and down arrow keys allows the user to move up and down the path. Figure 4.24 shows an example of a top-down view with a path from San Luis Obispo, CA to Bellingham, WA.

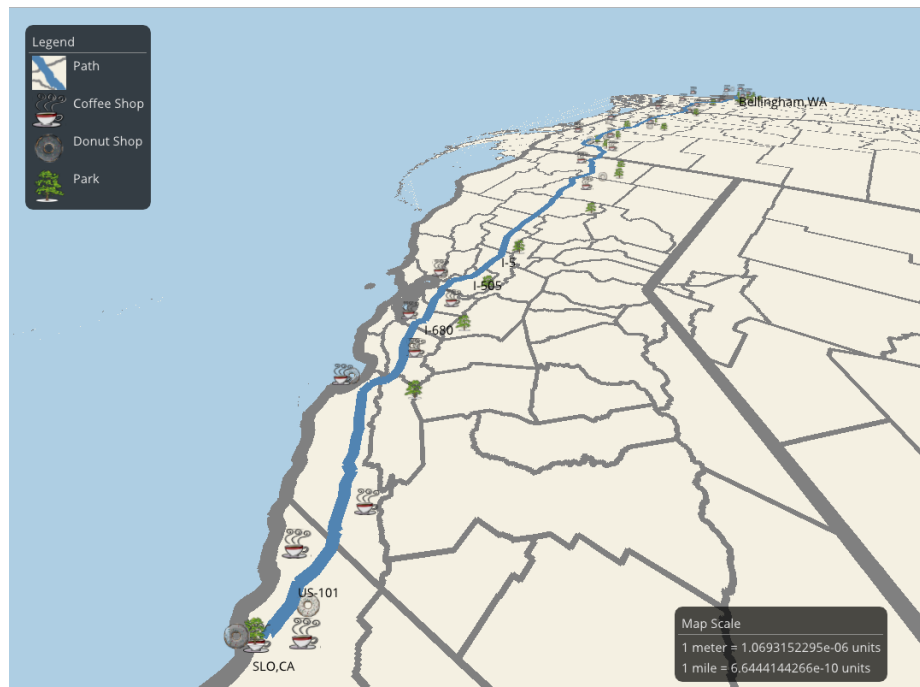


Figure 4.24: Top-down view of a path from San Luis Obispo, CA to Bellingham, WA

4.6 Navigation View

An interesting aspect for a map is to be able to navigate through the terrain and visually get an idea of where and how to get from one location to another. In order to do this, a separate camera from the top-down view was created for the purpose of path following. Initially a Catmull-Rom spline was implemented. One of the biggest issues with this implementation was the jerky movements as the camera followed quick turns and movements along the path. Determining the proper up vector and look at position for the camera became a challenge.

To solve the jerky movements caused by using a Catmull-Rom spline, a B-spline is used instead. The B-spline creates a C2 continuous curve, while a Catmull-Rom spline creates only C1 continuous curves [2]. This means that B-splines create smoother curves than the Catmull-Rom spline [2]. This provided some relief to the swift movements, but fast changes from one look at position to another was still a problem. To solve this, interpolation between the camera position, up vector and look at position was added. This produces a smoother transition between the current position and the previous position. Figure 4.25 shows an example of the spline, a Frenet frame (showcasing the positive x, y, and z vectors), and the path below it on an older version of the application.

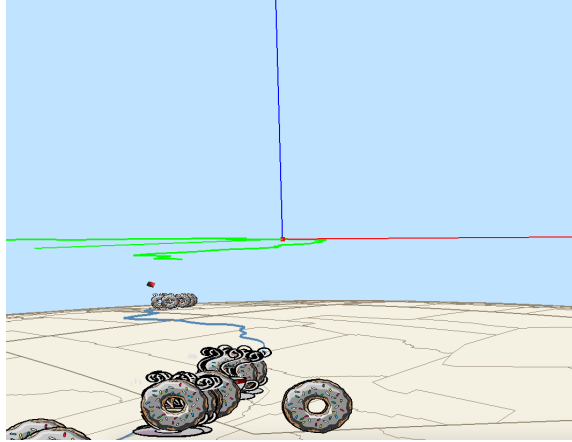


Figure 4.25: Frenet frame, Spline, and Path Example

Due to the curvature of the earth and the positioning of the United States within the graphics world, the up position is not simply a unit vector in the x, y, or z direction. The up vector changes as the position of the camera changes within the world. As a naive implementation, a vector is created from the world origin where x, y, and z are zero to the current position of the camera and then normalized. The lookAt position is the location that the camera is pointed towards. As the camera moves along the path, the lookAt position will change, as it cannot stay in one place to truly follow the path. In order to produce the correct lookAt position, the tangent and position of the camera are used. The final lookAt position is the position of the camera moved some scale along the tangent vector and a small scale down the normal to lower the gaze onto the route.

4.7 Performance

Performance is critical for a mapping application. Users expect maps to load quickly. However, the system gathers data from multiple large source files, and the initial, unoptimized implementation took from two minutes to two and a half minutes for loading of data and start up of the map. This was unacceptable.

To relieve the main thread from heavy file parsing, all of the file parsing was offloaded onto multiple threads. Each aspect of the data gathering was moved to a separate thread and thread results are accumulated prior to data use in the processing stage. After making this optimization, the time from source and destination input to complete rendering was reduced to approximately 30 to 50 seconds.

Frames per second (FPS) are used in graphics to indicate how well an application is performing. Even better is using the Frame Time [14]. For this system, the top-down view has a Frame Time of about 50 ms and runs at about 20 FPS. For the navigation view the Frame Time is about 52.1 ms and the system runs at about 19 FPS during traversal and 16 FPS when the next DEM file is loaded in.

The system transfers data to the GPU every single frame. For the top-down view, this does not really make sense as the data is not moving or changing between frames. So this is something to consider in future versions. In the navigation view the terrain is made up of six different layers, each layer consisting of a 64 by 64 vertex grid. All of the geometry is rendered every single frame, there is currently no view frustum culling implemented, so even though terrain is not seen behind the camera, the geometry is being transferred to the GPU and rendered. This adds to the transfer bottleneck that is common in graphics. Between county lines, state lines, route, and points of interest there are about 300,000 vertices transferred to the GPU each frame. In addition to the terrain geometry itself, each DEM file contains anywhere from one million to four million elevation values of type double. The number of polygons drawn each frame in the top-down view is 6,095,799. The number of polygons drawn each frame in the navigation view is 6,439,095. This is a very large number of polygons, especially to be drawing each frame.

Improvements on performance are discussed in the Future Work Chapter.

Chapter 5

VALIDATION

This chapter talks about the validation framework for this thesis. Validation for this system contains two parts. The first part of the validation includes a comparison study involving the TRPINav system map, a Google map, and a hand drawn artistic representation of the United States with a blue path overlaid on it. The goal of this first part is to collect user feedback on what features and characteristics stood out to them, which features were liked and which were not. This will help improve future implementations of the TRPINav system to be better tailored to users and their goals for the map. The second part of the validation is real-time user testing of the system. The goal of this part of the validation is to see how well the visualization does in helping users answer geographic questions and whether the representation and layout of the data drives exploration.

Overall, users were pleased with the TRPINav system and wanted more features and functionality. The main results of the user studies are assembled in the following list of highlights:

- Hand drawn style drew users to landmarks and attractions
- Terrain promotes exploration
- Users expected more interaction with points of interest
- Travel distance and travel time are important to users
- Organization and layout are important to users
- County lines were found to be too prominent

- Users had difficulty determining their actual location
- Positioning of text caused poor readability

5.1 Comparison of Maps

This part of the validation asked users to fill out a Google Form, which can be seen in Appendix A, containing questions that compared different types of maps. Figures 5.1, 5.2, 5.3 show the three maps compared during the second section of the survey. The first section of the form displayed an image of the TRPINav system map with a path from San Luis Obispo, CA to Seattle, WA and asked users about features that stood out to them, features they felt were missing, and what they liked or disliked about the image. The second section of the form displayed an image of a Google Map with a path from San Francisco, CA to Portland, OR asking the same set of questions as the previous section. The goal in these two sections was to determine which features or characteristics of a map draw the users' attention and whether these characteristics are liked or disliked by the users. The third section of the form displayed a TRPINav map, a Google Map, and a hand drawn representation of the United States. Each has a path from somewhere in California into the Oregon or Washington regions. The goal of this section was to determine what the use case for each map was. The section was written in a way that the users were asked a series of "If I asked this, which map would you refer to" type questions. This would force the user to investigate their use cases for each map. Thirty-nine people volunteered their time and took the survey online.

5.1.1 TRPINav Map Critiques

The first section of the Google Form took users through a series of questions to determine which features stood out to them at first glance, which features they liked,

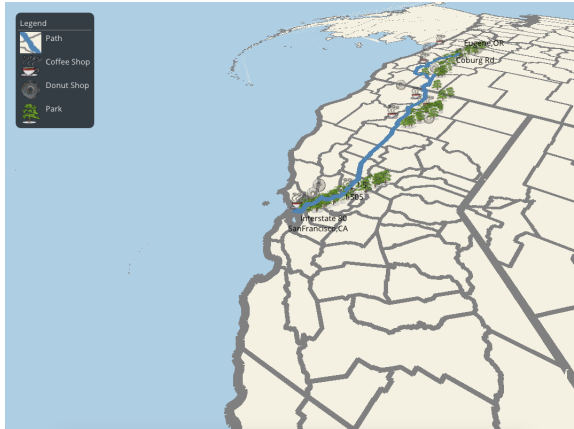


Figure 5.1: TRPINav Map used in survey comparison

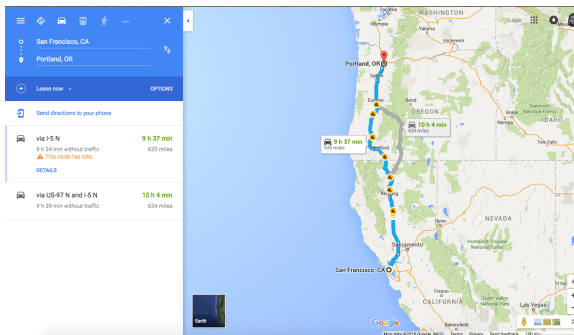


Figure 5.2: Google Map used in survey comparison



Figure 5.3: Hand drawn map used in survey comparison

which features they disliked, and if any features were missing from the system. One thing to note is that the users were looking at an image of the system map and therefore had no control over zoom or their preferred view on the map, and this could affect their responses.

5.1.1.1 Noticed Features

Aspects of the map rendering that stood out most to people can be categorized into points of interest, the angle of the map, state, county, and path lines, road information and colors. Figure 5.4 shows the distribution of responses based on the categories mentioned. Out of those categories, 38.5% of users noticed the angle of the map first. It is not a typical top down view as seen in most mapping applications. One person reacted to the angle by saying it is “classy and interesting,” also stating that “it looks unique.” Most people, however, were surprised by the angle and felt that it was odd or weird.

The next most popular response is points of interest, 25.6% of users noticed this feature first. Many commented on the number of parks, or noted that they could see the icons but it took them some time to realize what the tree icon represented. One user noticed coffee stops immediately and was excited for the addition of this type of point of interest. Overall, though, people commented on the concentration of icons.

Next, state, county and path lines were noticeable. Overall people stated that the lines were clearly visible, and the organization of the map into states and counties made sense. Additional feedback received about the listed features is:

- Two users did not understand what the lines inside of the states represented and found them confusing
- One person commented that there were just too many county lines

- Five users first noticed the colors of the map
- Many people first the green of the tree icons, as it really stood out against the coloring of the map
- One person made a comment about the blue and green contrast, stating that the color of the path really stood out
- Two users first noticed labeled roads first - Shamrock Rd and Highway 101 were noticed

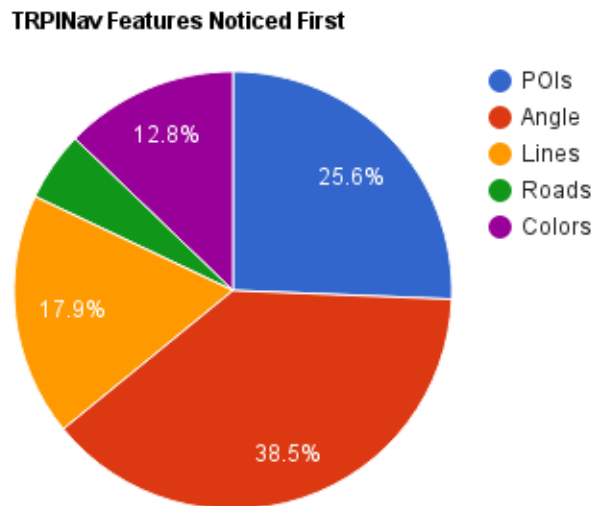


Figure 5.4: Distribution of features that stood out to users at first glance of the TRPINav Map top down view

5.1.1.2 Liked Features

Aspects of the map that people liked can be categorized into points of interest, the legend, the angle, path visibility, and county lines. Figure 5.5 shows the distribution of responses based on the categories mentioned. The most liked feature was the

county lines; 28.9% of users like this feature. One person stated that they felt the county lines were “pretty neat.” The other responses simply stated “county lines.”

The next most liked feature was the points of interest; 26.3% of users liked this feature. Four people mentioned specific points of interest or icons that they liked, while the rest referred to them as pit stops, points of interest, or simply icons. One person stated that they liked the icons because “each is unique, so easier to identify.” One person also specifically stated liking the coffee shops and donut shops because those are of interest to them when they travel. This is an important distinction to make when reviewing the feedback, some people did like the idea of the points of interest, however, the ones displayed in the map were not to their personal taste.

The next feature was the legend; 15.8% of users liked this feature. People called the legend clean, unconventional, and that it helped clarify destinations on the map. This is a good sign, because the legend helped users to better understand the map and the features within the map.

Visibility of the path received positive feedback from 13.2% of users. One user stated that the “map draws the eye to the path so it is easier to see the main subject of interest.” Three people mentioned the ability to see the whole path, or see the path from start to end.

The view angle of the map received positive feedback from 10.5% of users. People liked that the entire coastline was visible.

Other feedback received from users includes:

- One comment with “looks like a fun roadtrip!”
- One comment on the feature-less water. They said that they liked that the water is not textured.
- Two comments about the 3D aspect of the map as something users liked as

opposed to a completely flat top down view.

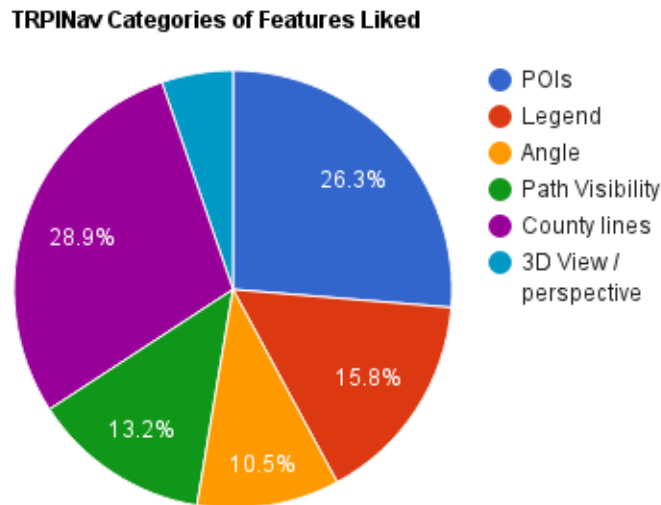


Figure 5.5: Distribution of features that users liked in the TRPINav Map top down view

5.1.1.3 Disliked Features

The responses to which features of the map were disliked could be placed into categories as well. These categories are point of interest placement, text, angle, county lines, and no response. As can be seen in Figure 5.6, people who responded with none tied with point of interest placement and angle as features that were not liked. The point of interest placement category is derived from responses dealing with clutter and poor visibility of icons.

The most disliked feature was the text; 25% of users disliked this feature. People said it was hard to read the text, that the text was cluttered, or that the font was not clear enough. This feedback is important to future implementations of the system, as text is a form of communication within the map and its legibility is important to

comprehension of the map.

County lines received 22.5% of negative feedback. Users did not understand what the county lines were. Since the feature did not make sense to them, they did not like it. The rest of the users either did not like the county lines at all, or felt that the lines could be less prominent.

- Two people said the lines should be lighter in color and three said that the width of the lines should decrease
- One user noted that without the county lines the map would be rather empty, so the system should include a topographic view instead
- One person also mentioned that the county lines were distracting

From the responses, it seems that making county lines less prominent on the map could make them less distracting and less intrusive to users.

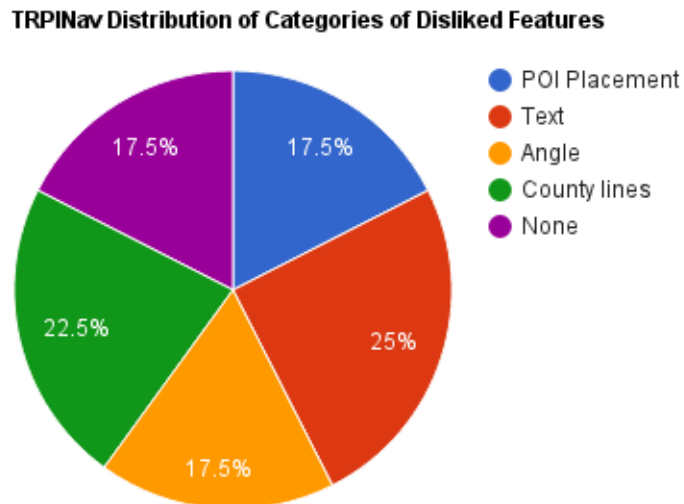


Figure 5.6: Distribution of features that users disliked in the TRPINav Map top down view

5.1.1.4 Missing Features

The responses for missing features can be categorized into points of interest, route options, level of detail, and information helpful to the route.

For points of interest, people wanted more options and a few specifically wrote showing gas stations, restaurants, hotels, or popular attractions. This is expanded on in the Future Work chapter. Points of interest are already a feature of the map, but allowing a more user-driven feature provides more personalization of the route and the trip to the user. People also wanted roads leading to the points of interest. This may cause extra clutter in this type of top down view, but in the navigational view, this is an interesting feature to add.

When it comes to route options, a few people mentioned wanting alternate route options the way Google Maps does it. The TRPINav system currently takes the first route out of the available routes returned from the Directions API, and it should continue to do so, but having some UI feature to show alternate routes and allow the user to choose the one and rendering the new route could be of use to the user.

For level of detail, users wanted more details pertaining to overall whereabouts on the path. Major city names and state names were a common response. People also wanted other roads, for navigational purposes. Information helpful to the route is total travel time, total distance for the route, a compass to understand where north is on the map, and a scale. One user also mentioned wanting topography on the map, which gives the user more information about the areas through which the path traverses. Overall the missing features truly encompass the objective of the map and the Future Work chapter expands on these topics.

With the responses given for this section of the Google Form, it is seen that users enjoy the major aspects of the system and want more information to better help them

understand the map.

5.1.2 Google Map Critiques

The second section of the Google Form took users through a series of questions to determine which features stood out to them at first glance, which features they liked, which features they disliked, and if any features were missing from the system. One thing to note is that since this was an image, people could not zoom or find the level of detail that they would want to see in a map, this could affect the way they respond to the questions. Although users are more familiar with Google Maps, determining which features are important to them is the best way to improve upon the TRPINav system.

5.1.2.1 Noticed Features

The responses for this question were categorized into traffic alerts, path(s), angle, directions, colors, Google Map, organization, travel time, distance. Figure 5.7 shows the distribution of responses among these categories. Travel time and distance could have been grouped together, however, more people noted the travel time than they did the distance, which makes time more important to users.

The most noticeable feature at first glance was that it was a Google Map; 22.7% of users noticed this feature first. Noticing the map as a Google Map means familiarity with the layout and the system.

The next most notable features were traffic alerts and the multiple route options, each receiving 18.2% of responses. Although eight people noticed the traffic alerts, one or two people mentioned finding them distracting. This array of responses shows how people hope to use the map. People that find the traffic alerts useful may be more concerned about the total travel time, ways to avoid those areas, etc., while

people who find them distracting or unnecessary are more concerned with the trip itself.

Travel time received 11.4% of user responses. People liked that they could see how long the trip would take, even as a rough estimate. This type of information is useful to users who plan to drive the route, as they can plan stops, or departure time based on this information.

The Google Map colors were preferred to the TRPINav colors, receiving 9.1% of user responses. This is interesting because the TRPINav system colors are based off of the Google Map.

Next, the angle, directions, and organization of the map were noticed. The organization of the map has to do with the level of detail at the zoom level of the image, and layout of the map. Users liked the simplicity of the map. Users that noted the angle said that it was a typical top down view, something that they were used to. The directions in Google Maps are shown in a side panel, which drew the user's attention.

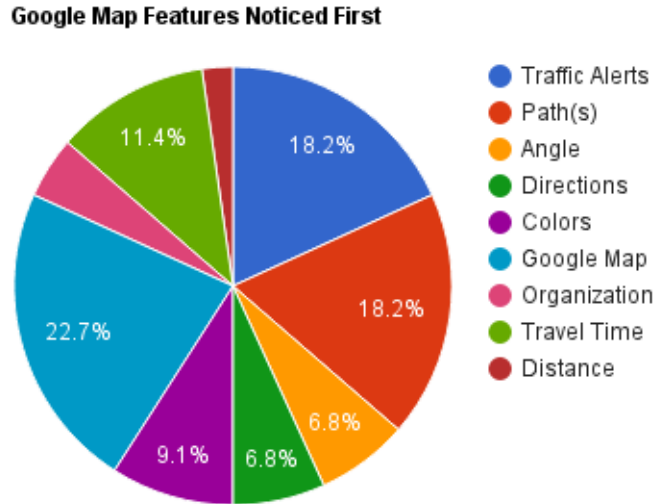


Figure 5.7: Distribution of features that users noticed first in the Google Map

5.1.2.2 Liked Features

For this question, the responses were split into path visibility, city and road names, simplicity and organization, traffic alerts, and travel information. Figure 5.8 shows the distribution of responses within these categories.

Receiving 25.8%, the most liked feature is the simplicity and organization of the map. People liked the layout of states, the coloring of the underlying map, readability of all features on the map. Part of the simplicity is the level of detail differentiation at each zoom level.

Next people really liked the travel information which includes travel time, travel distance, and alternate routes. This feature was liked by 24.2% of users. The responses were very similar to the ones provided in the previous question. Users liked that the information was clearly labeled on the map. Because of clear labeling, users

felt they could quickly find this information and quickly learn more about the route.

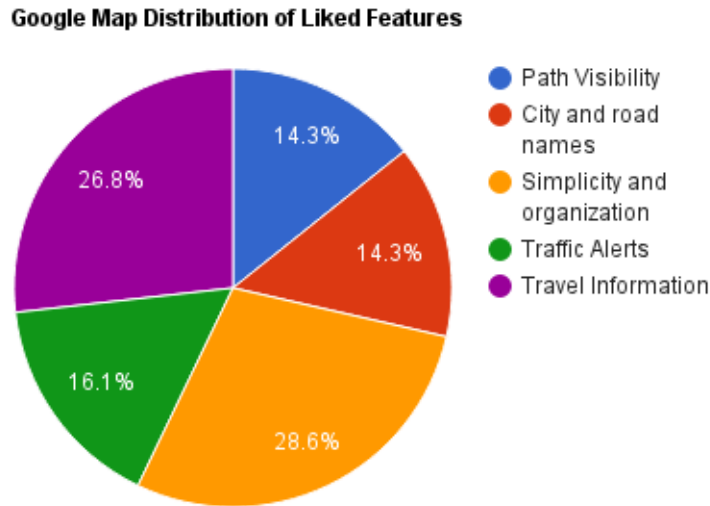


Figure 5.8: Distribution of features that users liked in the Google Map

What this section and the previous section show is that people like simplicity and cleanliness in their map. Giving less information and allowing people to zoom in to increase the level of detail shown in the map is a more pleasant experience for users than being shown a lot of information at once. All elements of the map also need to be legible. Users here noticed city, state, and road labeling because they could easily read it as compared with the TRPINav system. Also, people like information regarding their route - whether this means giving users alternate routes, travel time, travel distance - as it gives them the information they need to plan their trip. For the TRPINav system it would be important to associate each zoom level with a new level of detail, allowing users to more easily process all of the data on the screen. Also, looking into placing text directly onto the surface of the map as opposed to hovering the text over areas may make it more readable.

5.1.2.3 Disliked Features

The number of responses for this question decreased significantly. The majority of responses was no, none, etc. However, it was interesting to find that some people did not like the construction markers along the path. They felt that for the length of the trip, the markers were unnecessary or even “overkill.” One person mentioned that the path itself was too small and not enough details were shown. Another user mentioned that the optional paths were unnecessary, that they would rather be given one set route to follow. The final comment had to do with having more context and information at the zoom level of the image. The user felt it was unhelpful to have to zoom in for more cities along the path, that they would rather have that information already visible when the route is generated.

To solve this type of problem the TRPINav system can look into providing more details at each zoom level, but doing so with careful placement and size of each element to avoid clutter of the screen. This could be done by only showing one point of interest in a given location when zoomed out and increasing the number as the zoom increases. Also, determining point of interest necessity based on the total distance of the path.

5.1.2.4 Missing Features

The number of responses for this question also decreased significantly from the first two, however, there were still a number of responses. The categories that the responses fit into are points of interest and rest stops, and a compass for navigation. One user also mentioned that they felt the map was empty, but were not sure how to fill the space. People that mentioned points of interest and rest stops noted gas stations and actual rest stops that can be seen off of the highways. Two people mentioned having settings where they could program in planned stops, have default points of interest

for any route they generate, or having points of interest for longer routes in general. One user wanted more information when it came to the traffic alerts and waypoints that the Google Maps sometimes show. Overall, the main missing feature is points of interest.

This section of the survey shows that people do like the idea of having points of interest along their route and would like to see many different kinds of points of interest. The range of preferred points of interest means creating a system that has user-driven points of interest along the route. Users should have the option of the type places they see pop up along their route. This is something discussed in the next chapter.

5.1.3 Three Map Comparison

In this section of the Google Form, users were asked a series of questions pertaining to map use case. Which map would they refer to for a specific need. This analysis looks into the type of features that users want for maps that involve solving geographical problems including finding specific locations, finding borders, finding pit stops, and finding directions.

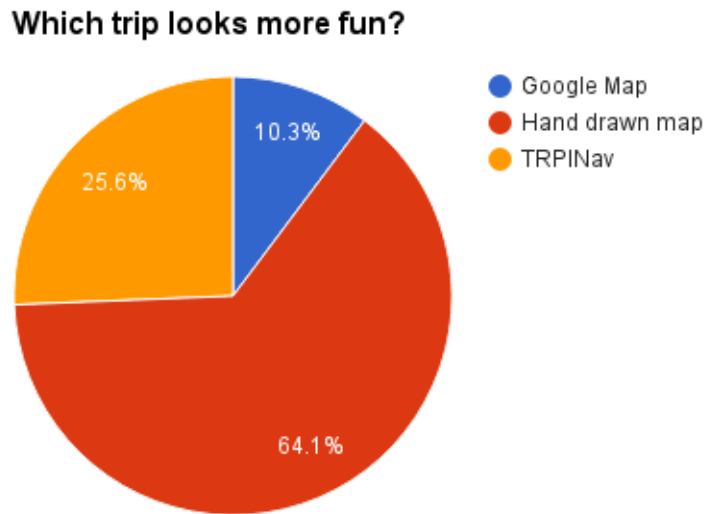


Figure 5.9: Distribution of responses for determining the trip that looks the most fun

Figure 5.9 shows the distribution of response to which trip looks the most fun? Users in general chose the hand drawn map over the TRPINav Map and the Google Map. Users felt that the style of the features gave it a storybook, carefree vibe. Also, the style made it feel more adventurous, one user noted that it was the hand drawn aspect that added the adventure feeling. Two people noted that the hand drawn map reminded them of a Lord of the Rings map or a treasure map. The scenery, landmarks and sight seeing places on the map made them want to explore more. Two users noticed the mountains on the map, and that's what made them feel like the trip was more fun. The style of landmarks and the hand drawn aspect of the map is something that TRPINav can incorporate into rendering of system features to entice users to want to explore more.

Which map gives a clearer route from start to end?

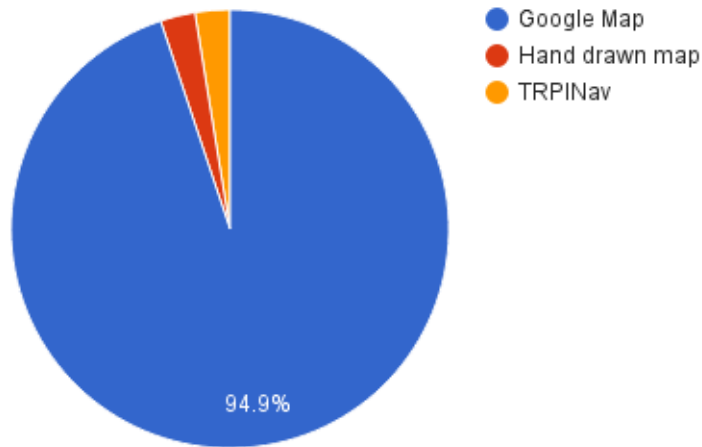


Figure 5.10: Distribution of responses for determining clearest route

When asked which route was the clearest, 94.9% of people responded with Google Maps. Figure 5.10 shows the distribution of responses. People chose the Google Map because of their trust in the system and they felt that it was more precise. One user noted that the scale was more precise. People liked the bold path, cities as landmarks, time estimate and that the highways were clearly marked. Seeing a hand drawn version and the TRPINav style made people feel that the paths were less precise. Part of this is not knowing the TRPINav system and not having trust with the system. Giving users more details about the travel route could help them feel more comfortable with the TRPINav route.

If I ask where the closest park is to the start of your trip, which map would you refer to?

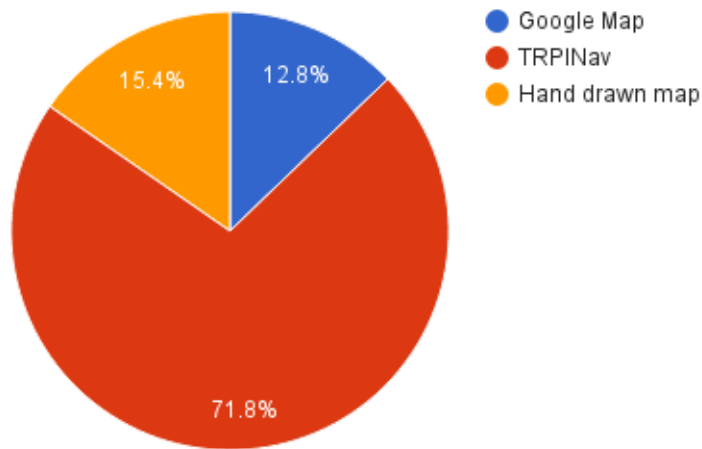


Figure 5.11: Distribution of responses for finding the closest park

When users were asked to find the closest park to their starting location on the route, 71.8% of people chose the TRPNav map. Figure 5.11 shows the distribution of responses. A few users felt that this question was a leading question because the TRPNav system explicitly shows parks on the map with a tree icon. However, users that chose the Google Map stated that they could see the green regions on the map, but weren't sure if they were parks or not. For the purpose of this evaluation this question is going to be disregarded because a few users noted that this was a leading question.

**If I ask where would you stop on your drive,
which map would you refer to?**

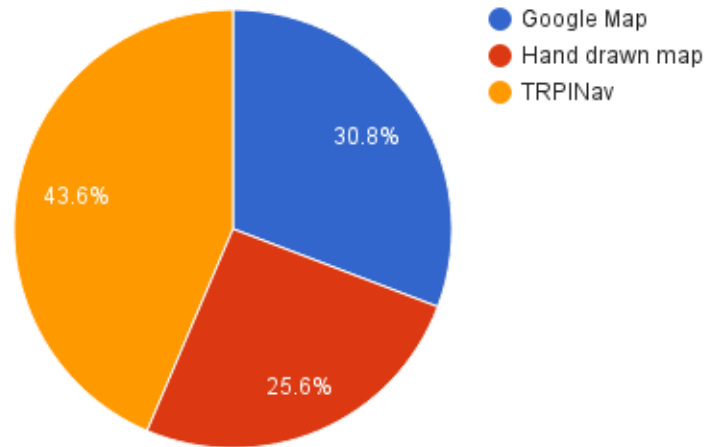


Figure 5.12: Distribution of responses for deciding where to stop on a drive

When users were asked which map they would use to find a place to stop along the drive, 43.6% of users chose the TRPINav map. Figure 5.12 shows the distribution of responses. Users overwhelmingly responded saying that the TRPINav map already has the points of interest so they would not have to do an extra search. A few people did note that they might still perform a Google Search. The users that chose the Google Map stated that they felt more familiar with the system and trusted it to tell them where they could stop. Looking into how to help users build trust with a system is something that would be necessary for future use of the TRPINav system.

If I ask where is San Francisco with relation to the route, which map would you refer to?

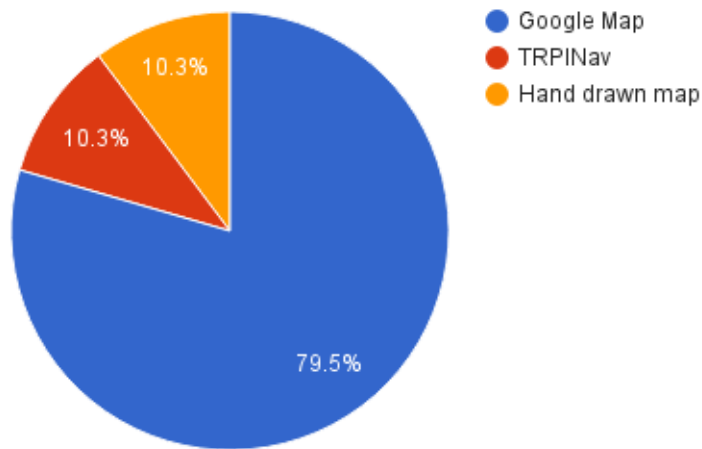


Figure 5.13: Distribution of responses for the location of San Francisco

When asked which map users would refer to in order to find San Francisco, 79.5% of users chose the Google Map. Figure 5.13 shows the distribution of responses. Users felt that San Francisco was clearly labeled on the Google Map, so they could quickly find it. For the hand drawn and TRPINav maps, users said they could determine the location based on the shape of California or the coastline, but that would require knowledge of San Francisco's location. Results show that for questions involving finding specific locations, users really want and need clearly labeled maps.

**If I ask where is the California-Nevada border,
which map would you refer to?**

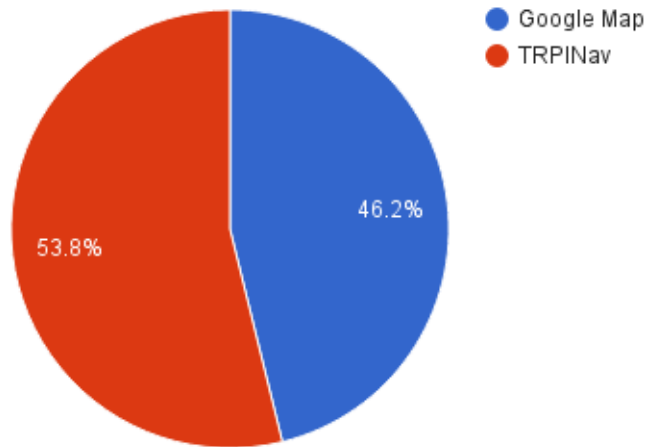


Figure 5.14: Distribution of responses for finding the California/Nevada border

When users were asked which map they would use to find the California/Nevada border, the answers were split between the Google Map and the TRPINav map. Despite the split, 53.6% of users chose the TRPINav map. Figure 5.14 shows the distribution of responses. The TRPINav map was said to have borders that were clearly defined so it was easier to make the distinction between the two states. One user did note that while they could find the border quickly, they had to have known the location of California and Nevada to find them on the map. Again, for finding and locating places on a map users need clearly defined regions and clear labels.

If I ask what cities are close to your destination, which map would you refer to?

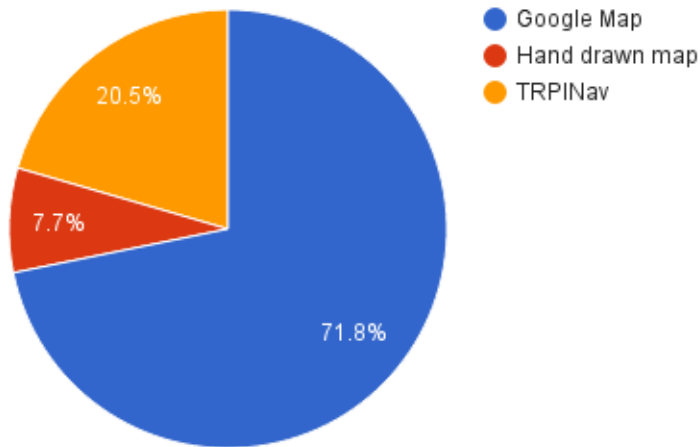


Figure 5.15: Distribution of responses determining cities close to destination

When asked which map they would refer to in order to find cities closest to the destination of the route, 71.8% of users chose the Google Map. Figure 5.15 shows the distribution of responses. The Google Map has labels of cities throughout the map and it has clear labels. While the hand drawn map does have some city labeling throughout, users felt it was not clear or apparent enough to them. The TRPINav system currently only shows the start and end cities on the route, so users felt they could see those labels, but otherwise did not have enough context.

Out of the choices, which map would you use for determining directions?

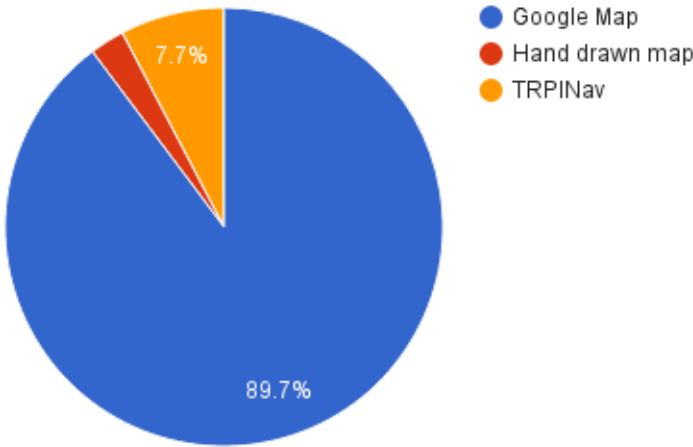


Figure 5.16: Distribution of responses for which map best used for directions

When asked which map they felt was best used for directions, 89.7% of users chose the Google Map. Figure 5.16 shows the distribution of responses. Users liked the clarity of the instructions provided in the side bar, the labeled cities and roads, and they had a pre-existing trust and familiarity with the system. To improve upon TRPINav, a scale, compass and clearer labeling along the route would be necessary to show users that the route is as precise as a Google Map route.

Out of the choices, which map would you use for determining pit stops?

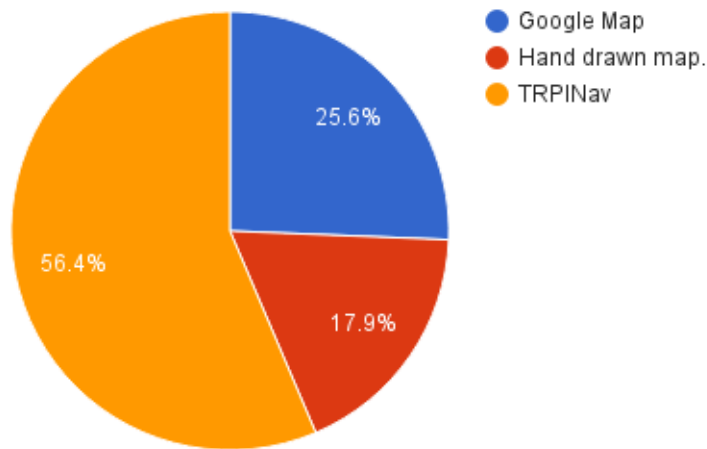


Figure 5.17: Distribution of responses for which map best used for pit stop

When asked which map users would use for determining pit stops, 56.4% of users responded with the TRPNav map. Figure 5.17 shows the distribution of responses. People liked that the points of interest were integrated into the TRPNav system requiring less work on their part. Having the points of interest integrated into the system also made users feel like they would be more inclined to stop along the drive. People who chose the hand drawn map liked the types of attractions on the map. Again, people felt more comfortable with the Google system as a whole and felt that they might still search through Google to determine a point of interest, if they could.

This section of the Google Form took users through a series of geographic questions. When it comes to locating places on the map, the TRPNav system needs improvement on clarity and readability. When it comes to regionalizing or recognizing areas on the map the TRPNav has clear borders, so users are able to differentiate between regions. However, clearer labels would also provide users with extra support

and confidence in the map. Users were able to identify objects (points of interest) on the map fairly clearly. Overall the TRPINav system could improve on readability, which includes organization of data within the map, more labels, and better placement of labels. Also based on the first two sections of the Google Form, users are interested in the points of interest aspect of the map as it seems to be something that they feel the Google Map lacks.

5.2 Real-Time Testing of the System

This part of the validation involved fifteen volunteers that took time to use the system to answer questions. This evaluation gave the user one to two minutes to get acquainted with the system and ask any questions about the system. After this point, the users began a survey in a Google Form that took them through a series of questions asking them to perform tasks within the system. Appendix B contains the Google Form with all the questions the users had to answer. The goal in this section was to see which types of questions users had trouble answering with the system, which features needed to be improved on to help users answer those questions, and what style of features users preferred within the system. Style included image style for sprites and color scheme for the terrain.

Analysis looks into what users had difficulties with, however, the system had features that users liked. Users liked the points of interest aspect of the system and wanted even more options. Users really enjoyed the terrain in the navigation view. They felt it was helpful to understanding the route and their surroundings more. Users also appreciated being able to pause the navigation to explore areas more. Overall, the system was able to entice users to explore their surrounding areas, give users interesting stops along the route, and provide users with capabilities of catering the map to their needs (changing speed of traversal, camera height, and color scheme).

5.2.1 Difficulties

Users had difficulty answering questions about location of specific places. In the top down view of the map, users were asked to find a park halfway along the route. Only two users had the same response, the rest were all different parks. While all the parks were generally around the halfway point, the array of answers shows that the system does not help the users answer location-specific questions. More information is needed for the user to be able to determine a park at the halfway point. When asked how to improve upon this aspect of the map users mainly responded with having interactive icons.



Figure 5.18: Distribution of users' preference to show or hide road names

Users want to hover over or click on the icon and have the address show up. Another interesting idea was to connect the list of points of interest within the map to the icons, so if a user were to click on an icon it would be highlighted in the list. One user also mentioned that clicking on a point of interest could result in a pin on the map. This pin could be chosen for a re-route, or at least the address would be visible to the user. Looking at Figure 5.18, the usefulness of road names also had a range of answers. However, despite this range of responses, as seen in Figure 5.19, most users would prefer to have road names in view. Based on the initial survey results, if labeling were clearer, more people may find the road names useful.

Would you be more likely to toggle the road names on or off in this type of view?



Figure 5.19: Distribution of users’ preference to show or hide road names

In the navigation view, users had trouble determining the highest and lowest elevation locations. While users could generally describe the area they were in - North, closer to Washington, on I-5, leaving San Luis Obispo - they explained that they were not comfortable with the vagueness of their answer and hoped they could actually pinpoint specific locations. Users felt they needed more city information along the route to be able to determine their location. Two users suggested having a mini map screen in the bottom right corner during navigation to show the top down view with city information. Another suggestion was showing the percentage of the path traversed so as to give some indication of length of the route or distance left. This ties into responses from the first survey results, where users wanted to see route distance and a time estimate in order to feel more comfortable with the route.

5.2.2 Top Down View and Navigation View Comparison

Comparing Figures 5.20 and 5.21, it is interesting to notice that users generally seemed to like the points of interest in the navigation view more than the top down view. Users noted that they liked the pause feature in the navigation view as it allowed them to explore their surroundings more and notice points of interest more easily. Having more information about surroundings through the navigation view gave users more context for the points of interest. Taking this into consideration, more points of

interest could be rendered in the navigation view in contrast to showing everything in both the top down and navigation views.

On a scale of 1-5, how useful did you feel the points of interest were in this view of the map?

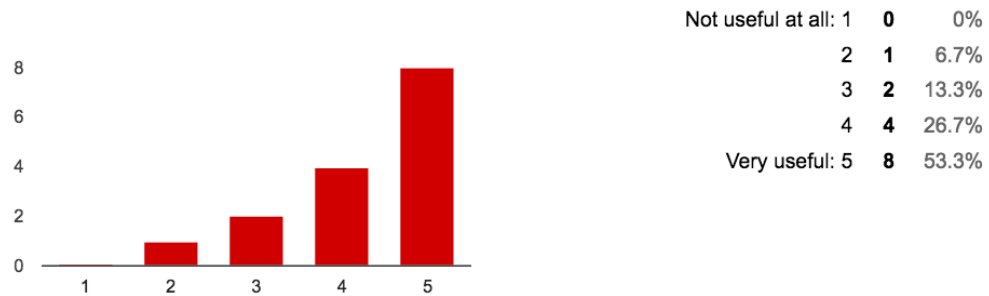


Figure 5.20: Usefulness of points of interest in the top down view

On a scale of 1-5, how useful did you find points of interest in the navigational view?

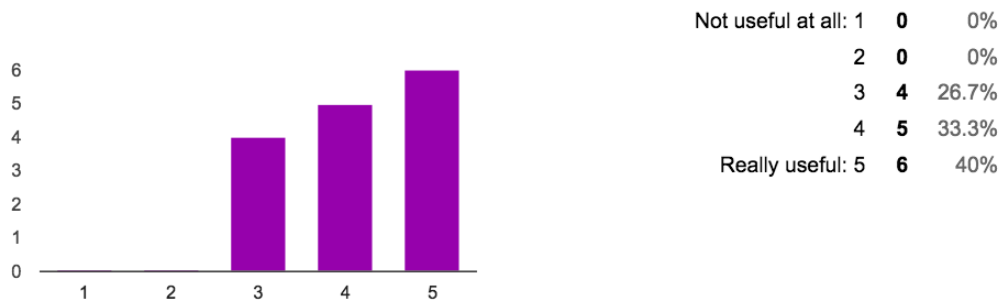


Figure 5.21: Usefulness of points of interest in the navigation view

Comparing Figure 5.23 to Figure 5.22 shows that the distribution was geared more towards the navigation view, with one outlier. Overall users felt that having the highway and road names in the navigation view gave them some context to their location in the world. However, a few users commented saying that they would prefer to have more road information, preferably more than major highways.

On a scale of 1-5, how useful did you feel the road names were for the route?

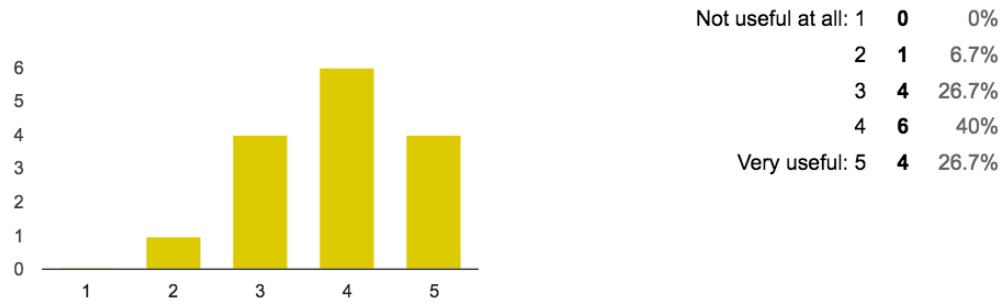


Figure 5.22: User sentiment on usefulness of road names in the top down view

On a scale of 1-5, how useful did you find the road names in the navigational view?

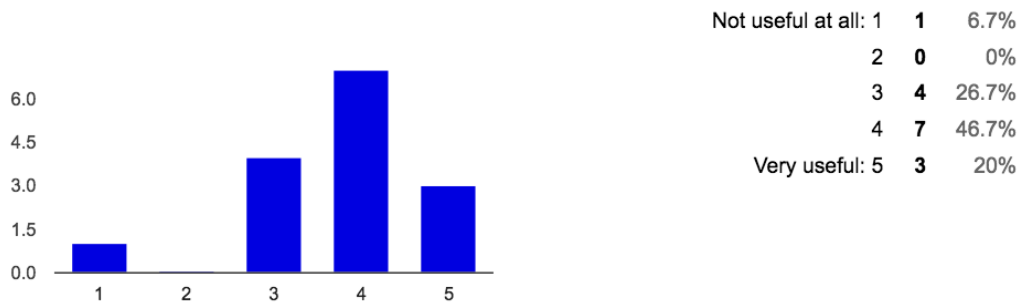


Figure 5.23: User sentiment on usefulness of road names in the navigation view

5.2.3 User Preferences

Users were given the ability to move the camera up and down while the navigation traversal was in pause mode. This would allow users to see if they preferred to be closer to the route and the terrain, or higher up above it and looking down and over the whole map. Figure 5.24 shows the distribution of answers. Most people actually preferred a higher camera that overlooked the terrain and route. Users liked that they could look ahead and see what was coming, especially for the points of interest. Users that liked a lower camera said that it really allowed them to see the modulation in the terrain. Those users liked that it also kept them closer to the actual route. What

the responses show is two different map use cases. Exploratory use of the navigation view could move the camera closer to the ground, while pure navigation could keep the camera higher up to allow users to see what was coming up along the route.

User Preference of Higher or Lower Camera in Navigation View

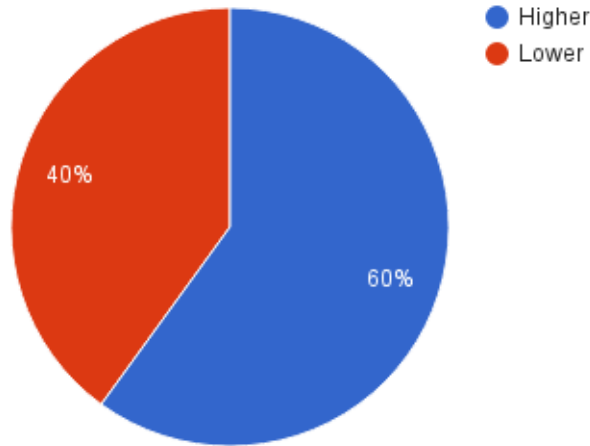


Figure 5.24: User preference on camera height

Users also had the opportunity to determine which style icon they preferred and which color scheme they preferred for the terrain. For the terrain, users were asked which color scheme was more aesthetically pleasing to them and which one provided them with better elevation differentiation. Figure 5.25 shows the options users were given and Figure 5.26 shows the questions and responses. Users felt that Options one and three were best for understanding geography, while Option three was more aesthetically pleasing than Option one. Interestingly enough the greyscale color scheme was not helpful to users nor was it aesthetically pleasing. This is an important consideration for future implementations of the system. With future implementations of the system, further testing could be performed splitting up users into three groups

(one for each color scheme) and seeing how the color scheme affects their ability to answer geographic questions. This information could then be used to either set a color scheme for the entire application or allow users to change the view - one solely for presenting elevation information, the other for general purpose views.

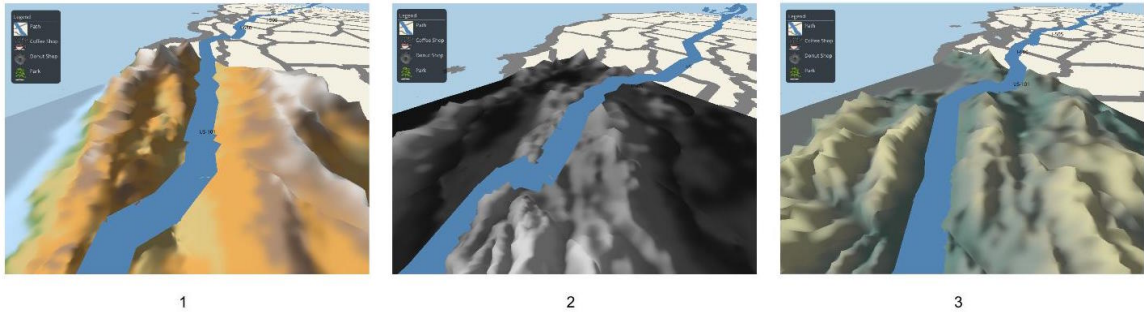
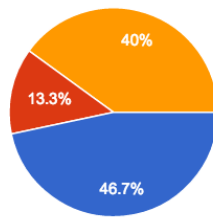


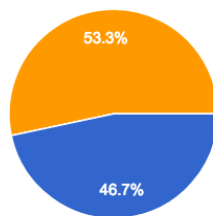
Figure 5.25: The three options users were given in the Google Form, which can be found in Appendix B

Out of the above choices, which color scheme helps your better understand the geography better?



Option 1	7	46.7%
Option 2	2	13.3%
Option 3	6	40%

Out of the above choices, which color scheme is more aesthetically pleasing?



Option 1	7	46.7%
Option 2	0	0%
Option 3	8	53.3%

Figure 5.26: Distribution of user preference for terrain color scheme

Users were provided coffee icon options shown in Figure 5.27, and Figure 5.28 shows the distribution of tastes.

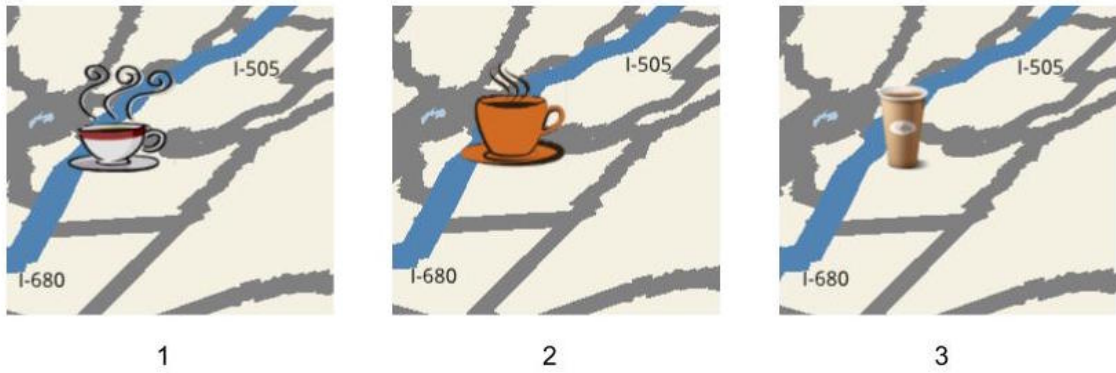
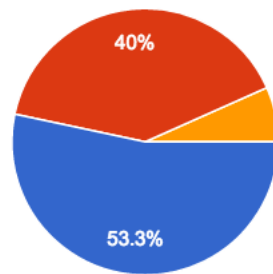


Figure 5.27: The three options users were given in the Google Form, which can be found in Appendix B

Of the three options for coffee, which style icon do you like best?



Option 1	8	53.3%
Option 2	6	40%
Option 3	1	6.7%

Figure 5.28: Distribution of user preference for coffee icon

Users preferred the more hand drawn coffee mug. This result correlates to the first survey results about the hand drawn map feeling more like an adventure and feeling fun to users. The hand drawn aspect really draws the users in.

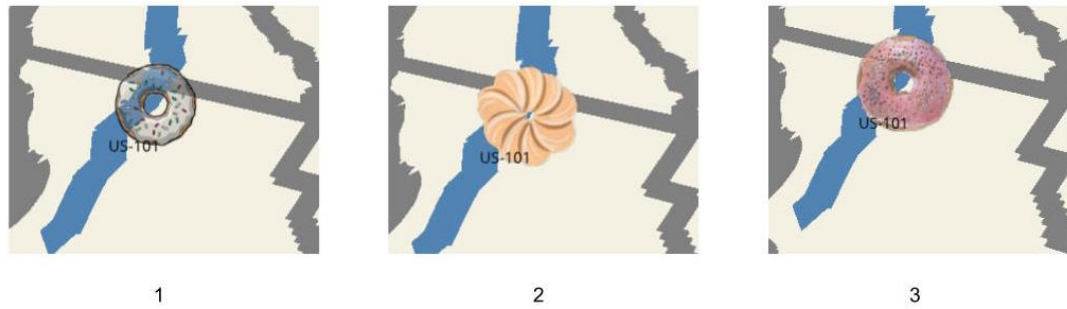
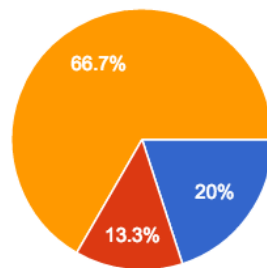


Figure 5.29: The three options users were given in the Google Form, which can be found in Appendix B

Of the three options for donuts, which style icon do you like best?



Option 1	3	20%
Option 2	2	13.3%
Option 3	10	66.7%

Figure 5.30: Distribution of user preference for donut icon

Interestingly though, the donut decision was much different. Figure 5.29 shows the options that users were given and Figure 5.30 shows the distribution of responses. People overwhelmingly preferred the third option, which was a more realistic image of a donut. It is possible that the donut choices were not varied enough. Option two was much more artistic than the others; however, it does not have the same line features that the coffee mug has.

Chapter 6

CONCLUSION AND FUTURE WORK

The TRPINav system is built as an exploratory cartographic visualization to entice users to find points of interest and to explore areas along a path. The system unifies data from many different sources and renders an interactive map. Contributions of this thesis are:

- A system unifying terrain data, route data, and points of interest data
- A 3D interactive cartographic visualization representing the unified data
- A user study revealing features that are necessary for understanding path traversal and for locating points of interest

Through user studies we found that features necessary for understanding path traversal include:

- Clearly labeled directions, major cities, and major roads
- Clearly labeled travel time and travel distance

We found that users needed more interactivity for determining location of points of interest. This chapter discusses future work for the TRPINav system, pulling ideas from user feedback.

6.1 User Settings

Currently the TRPINav system has an immense amount of hard-coded settings, which include the types of points of interest shown on the map and the elevation information

available for navigation use. Currently the user really only has control of start and end location, and even that is limited to the west coast to view the geometry clipmaps terrain. A future goal would be to allow rendering of the map without a path or points of interest and allow the user to input the type of information they would like to see. Once the user has entered start and end locations, they can input any or no points of interest they are looking for along the route. The system would then go through the data gathering, data processing, and data rendering stages for points of interest and directions. Other settings may include different views of the map, for instance, giving topographical views instead of showing all the county break ups. Allow the user to open a settings menu and choose which view they would like to see.

6.2 Points of Interest

There are a few aspects of points of interest that can be improved upon.

First, the TRPINav system currently does a pre-fetch of all of the points of interest, holding their information in memory. Pre-fetching the points of interest adds to the amount of time it takes for the system to start up. This can be annoying to the user and does not allow for user or demand-driven points of interest. Instead, the system could wait until a user has inputted types of points of interest to begin fetching the information. Another way to do this is on a per need basis during navigation. This means that points of interest are queried for as the camera moves along the path in the navigation view, so that only points of interest close to the user at a given time are shown on the map. This can increase the amount of points shown and help the user explore one area more.

Second, a better algorithm needs to be implemented for rendering the points of interest. This can mean representing multiple points of interest of the same type in a given area with one symbol. A method like that would require determining a radius or

distance as small enough to group point of interests to around a point. Another way to represent multiple points of interest is with the size of the sprite. Depending on the number of coffee shops in an area, the size of the sprite will increase or decrease. This gives the user a visual of many or few. Another representation of the concentration of coffee shops could be a number and an icon to give exact values to the user.

Third, the system currently gives no directions to points of interest, nor can users determine where one coffee shop is with respect to others. In the menu of points of interest, the addresses are listed, but the users cannot connect those addresses to the points of interest shown on the map. One way to fix this is to make the list interactive, such that when a user clicks on an address in the list the camera moves along the path to that specific point of interest. At this point the user can be given the option to get directions to that location, or continue on their current path. The icons could be clickable as well, so that users do not have to search for a specific one in the list. If users are in the navigation view, they can pause the navigation and two things can happen: hovering over the sprite shows a small message box with the address, or clicking on the sprite adds directions to that point of interest.

Also important is telling users where the data is coming from. Having a small acknowledgement of data sources at the bottom of the map will give users insight into where the points of interest, directions and terrain data is coming from. This adds credibility to the map and allows users to build trust within the system. This is important, because from the user studies we found that users had more trust and familiarity with the Google Map and thus chose it for directions even though the TRIPNav system uses Google for directions.

6.3 Path Generation and Simplification

Although the route generation uses the Google Directions API, the path could still be simplified and generalized to more clearly show turns and direction changes. The techniques used in LineDrive and Destination Maps could be re-implemented for better navigation and path traversal. LineDrive's features include road names and turning angles to allow a driver to understand where they are turning and which way they are turning. Destination Maps added in a road network to showcase major roads near the path. These aspects are important to users and their understanding of their location along a path. Being able to add city names along the path is another beneficial feature.

In addition to determining a clearer route to show users, the rendering aspect of the lines could be improved upon as well. The current implementation creates a rectangle mesh for each vertex along the path and then overlays the rectangles as necessary to try to create a smooth line. While this works, the lines still have odd angles and edges that protrude from the path at curves and turns. Instead of using rectangles, the lines could be represented by a more triangle-dense mesh.

6.4 Terrain Generation

Currently the DEM files have to be loaded on a per need basis, and not all DEM information is available easily. For the current system, multiple files must be downloaded from Google Earth Engine to make up the entire United States.

One solution to this would be to create a database for elevation information, such that generating the clipmaps terrain becomes a single query to the database for a chunk of data within a certain latitude and longitude range. This data can be incrementally updated, instead of flushing out the data in memory and loading a

whole new set of data.

If file loading is still the only option, then coming up with a better file loading algorithm is necessary. Currently the files are listed in ascending latitude order along the west coast. While this implementation works for some routes, it does not work for all routes. The system currently has a fixed ordering of loading in the files. It would be best if based on the latitude and longitude needed, the files could be searched for and loaded. Searching for the files could cause performance issues as the number of files in the system increases. One way to improve on this is to create a list of the files needed for path traversal and order them by latitude or longitude from start to end along the path.

Another aspect of elevation data that would be interesting to look into is the terrain generation. Currently a camera position is faked to be able to see more terrain from the actual camera view. The most detailed terrain is created about the faked camera because that is what is used as the viewer that the terrain is centered about. For the purpose of this map, it makes sense that the majority of the terrain is in front of the camera, and not centered around the camera. Realistically most users will want to look at what is ahead of them instead of what is behind them. Looking into an implementation where the terrain is not centered about the camera, but begins at the camera position and moves outward would be best for the system.

Currently there is no view frustum culling for terrain generation, so any geometry not visible by the user is still being rendered to the screen. This reduces performance in the navigation view. By adding view frustum culling the visualization would be much more interactive for the user.

6.5 Information Processing

As mentioned with the points of interest, all of the data is pre-processed or pre-fetched. While information like the KML data defining state and county lines would need to be pre-processed, data such as the directions, points of interest and elevation data could be done during runtime or on a per need basis. Currently in the system, the path is elevated off of the map from the very beginning. In the navigation view, the user can see the elevated path in the distance and it looks odd. It would be better if the user could see elevation changes as they happened in the terrain.

In addition, the map of the United States could be rendered to the screen for the user prior to deciding a start and end location, thus eliminating some of the performance hits taken due to pre-processing. The user would then choose source and destination locations.

Much of the data gathering and data processing involves for-loops and while loops, which lend themselves to optimizations. Cuda and OpenMP pragmas could also be used to parallelize and vectorize the implementation. A profiler would need to be used to determine the slowest parts of the implementation to see if they could be parallelized. Also the profiler would help analyze and see if performance improvements of offloading to the GPU are worth the overhead of starting up the kernel.

6.6 Port to Mobile or Webpage

Since the current system is a stand-alone application, it would be interesting to have it ported to a webpage or mobile application. Some issues that would arise with this are CPU and GPU intensive operations such as the geometry clipmaps terrain generation, or the amount of data that needs to be stored within the application to draw the map. By porting to mobile devices, users could really get the chance to

explore areas in a more meaningful manner. Points of interest along a route would also become more interesting, instead of having to plan them out before starting the drive, they could become spur of the moment decisions influenced by the system.

Once the system is ported to mobile, it would be really interesting to start integrating GPS capabilities to convert the navigation view into true navigation mode. The camera would follow as the person drove along the route. This would allow users to use the app in a more exploratory and mobile setting.

One major consideration is that network connectivity is important to gathering all of the route data and points of interest data. Future versions will need to look into what features could still be available if there were no network connection available. Also, looking into network failures and how to deal any loss of data within the system.

6.7 Scaling of the Map

As mentioned in the Implementation chapter, the scaling of the map is currently in the range from $[-1,1]$. This is a very small scale and does not give a lot of meaning to the developers or the users. The map should be scaled down to millimeters or nanometers such that the visualization has a more meaningful scale. One of the difficulties of simply changing the scale, is that all of the rendering is done in such a way to fit with the current scale of the map. The current offsets have no actual meaning and would need to be changed to fit the new scale. Also, the line widths would need to change to properly render with the larger scale. The biggest difficulty with this change would be updating the entire data rendering stage of the system to scale properly. Though this would be tedious and difficult, having a meaningful scale is worth the effort.

In addition to changing the scale, making the scale more meaningful to users in the lower right hand corner is important as well. Users need to be able to map a mile in the

map world to the real world. Currently, the system provides a statement of equality, but representing the scale as a bar graph is simpler for the user to understand.

BIBLIOGRAPHY

- [1] M. Agrawala and C. Stolte. Rendering effective route maps: improving usability through generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 241–249. ACM, 2001.
- [2] T. Akenine-Möller, E. Haines, and N. Hoffman. *Real-time rendering*. CRC Press, 2008.
- [3] P. Benard, A. Hertzmann, and M. Kass. Computing Smooth Surface Contours with Accurate Topology. *ACM Transactions on Graphics*, 33(2), 2014.
- [4] C. A. Brewer. Color use guidelines for data representation. In *Proceedings of the Section on Statistical Graphics, American Statistical Association*, pages 55–60, 1999.
- [5] J. E. Campbell and M. Shin. Geographic information system basics. 2012.
- [6] G. Developers. Google maps directions developer’s guide. <https://developers.google.com/maps/documentation/directions/intro>, 2016.
- [7] G. Developers. Google maps geocoding developer’s guide. <https://developers.google.com/maps/documentation/geocoding/intro>, 2016.
- [8] I. Dunn. ionEngine. <https://bitbucket.org/iondune/ionengine>, 2009–2016.
- [9] K. E. Foote and S. Crum. *Cartographic Communication*, 1995. <http://www.colorado.edu/geography/gcraft/notes/cartocom/section2.html>.
- [10] D. Gesch, M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler. The national elevation dataset. *Photogrammetric engineering and remote sensing*, 68(1):5–32, 2002.

- [11] Google. Google Maps v9.2.
<https://developers.google.com/maps/documentation/>, 2016.
- [12] L. Guelke. Cartographic communication and geographic understanding. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 14(1):129–145, 1977.
- [13] D. Huffman. Remembering LineDrive.
<https://somethingaboutmaps.wordpress.com/2011/03/08/remembering-linedrive/>.
- [14] T. K. G. Inc. OpenGL Software Documentation Kit. website, 1995-2015.
<https://www.opengl.org/sdk/docs>.
- [15] B. Jenny and L. Hurni. Swiss-style colour relief shading modulated by elevation and by exposure to illumination. *The Cartographic Journal*, 43(3):198–207, 2006.
- [16] E. Kalogerakis, D. Nowrouzezahrai, S. Breslav, and A. Hertzmann. Learning Hatchin for Pen-and-Ink Illustration of Surfaces. *ACM Transactions on Graphics*, 31(1), January 2012.
- [17] J. Kopf, M. Agrawala, D. Barger, D. Salesin, and M. Cohen. Automatic generation of destination maps. In *ACM Transactions on Graphics (TOG)*, volume 29, page 158. ACM, 2010.
- [18] A. S. Lobão, B. Evangelista, J. A. Leal de Farias, and R. Grootjans. 2d graphics, audio, and input basics. *Beginning XNA 3.0 Game Programming: From Novice to Professional*, pages 15–38, 2009.
- [19] F. Losasso and H. Hoppe. Geometry clipmaps: terrain rendering using nested regular grids. *ACM Transactions on Graphics (TOG)*, 23(3):769–776, 2004.

- [20] J. D. O’Grady. *The Garmin Nuvi Pocket Guide*. Peachpit Press, 2009.
- [21] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. Flow map layout. 2005.
- [22] M. Pharr and R. Fernando. *Gpu gems 2: programming techniques for high-performance graphics and general-purpose computation*. Addison-Wesley Professional, 2005.
- [23] C. P. van Elzakker. *The use of maps in the exploration of geographic data*. Utrecht University, 2004.
- [24] A. van Goethem, A. Reimer, B. Speckmann, and J. Wood. Stenomaps: Shorthand for Shapes. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), December 2014.
- [25] I. Yelp. Yelp 2.0.
<https://www.yelp.com/developers/documentation/v2/overview>, 2004–2016.

APPENDICES

Appendix A

TOP DOWN VIEW EVALUATION

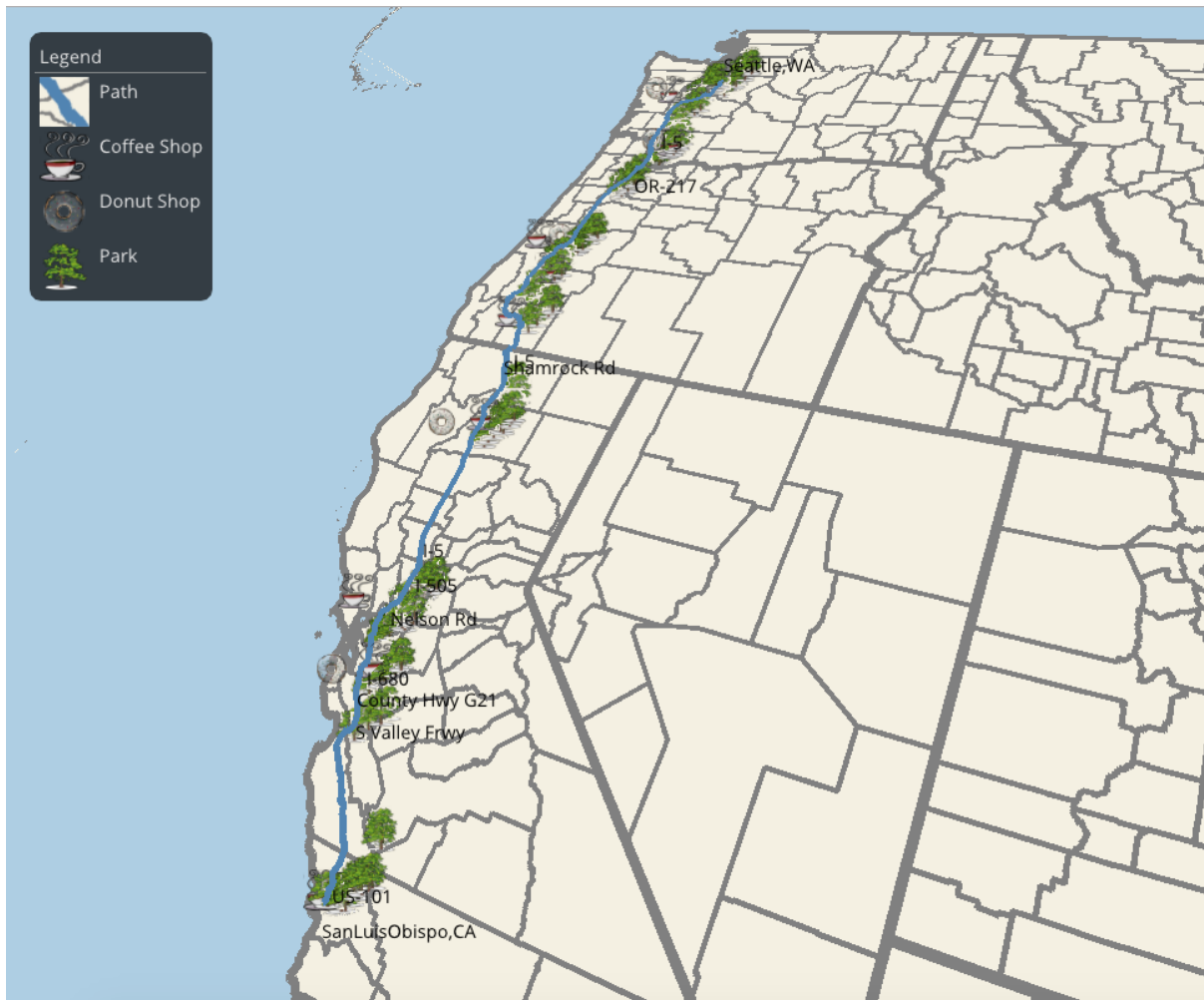
Top-Down Map View Comparison Evaluation

Please answer all of the required questions, the more feedback and more thorough your response the better! Thank you!

* Required

TRPINav Map

Please feel free to zoom in the page to view the image more clearly.



1. What did you see or notice first about the map? *

.....

2. Are there any features of the map that you like? Explain. *

.....

.....

.....

.....

.....

3. Are there any features of the map that you dislike? Explain. *

.....

4. Are there any features you feel are missing? If so, please explain. *

.....

.....

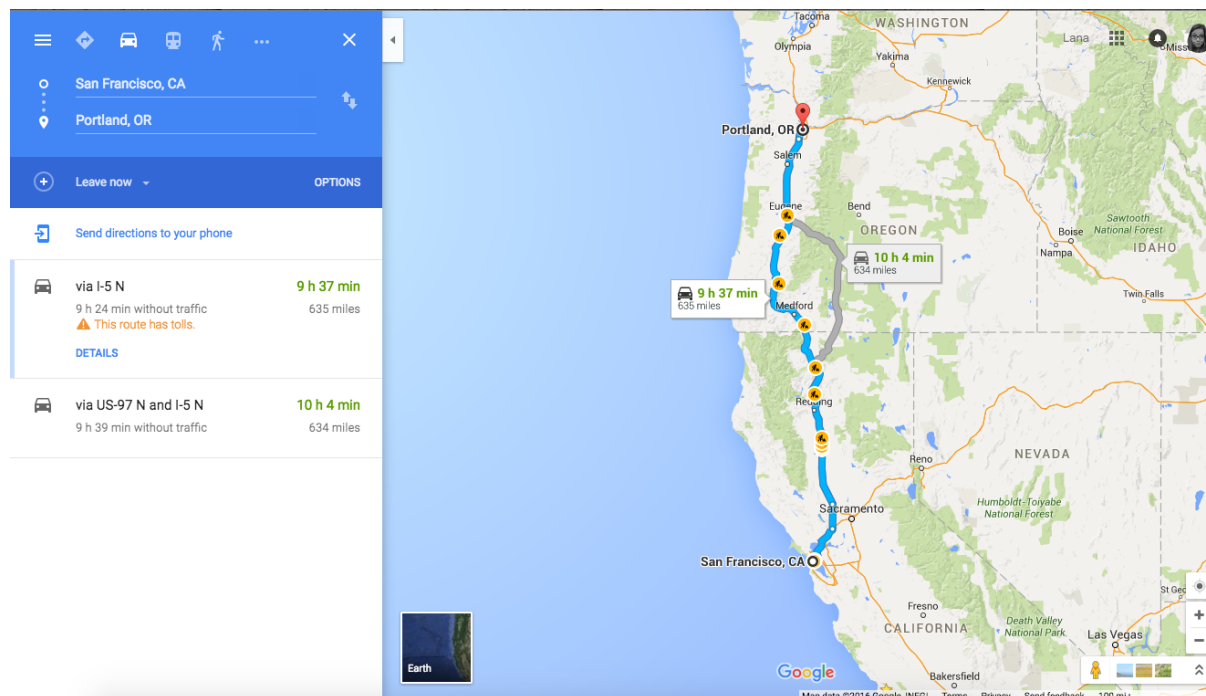
.....

.....

.....

Google Map

Please feel free to zoom the page in to view the image more clearly.



5. What did you see or notice first about the map? *

..... 94

6. Are there any features you like? Explain. *

.....
.....
.....
.....
.....

7. Are there any features you dislike? Explain. *

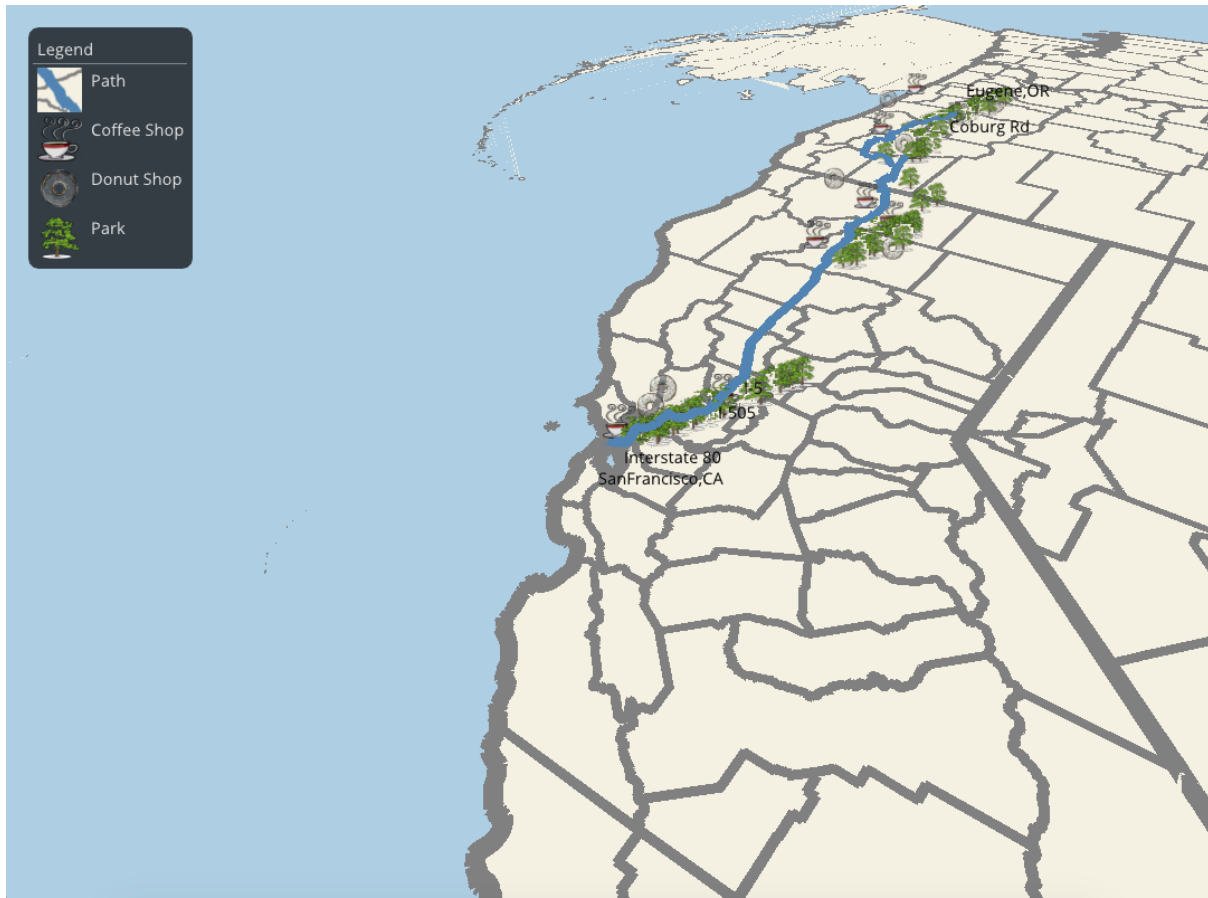
.....

8. Are there any features you feel are missing? If so, please explain. *

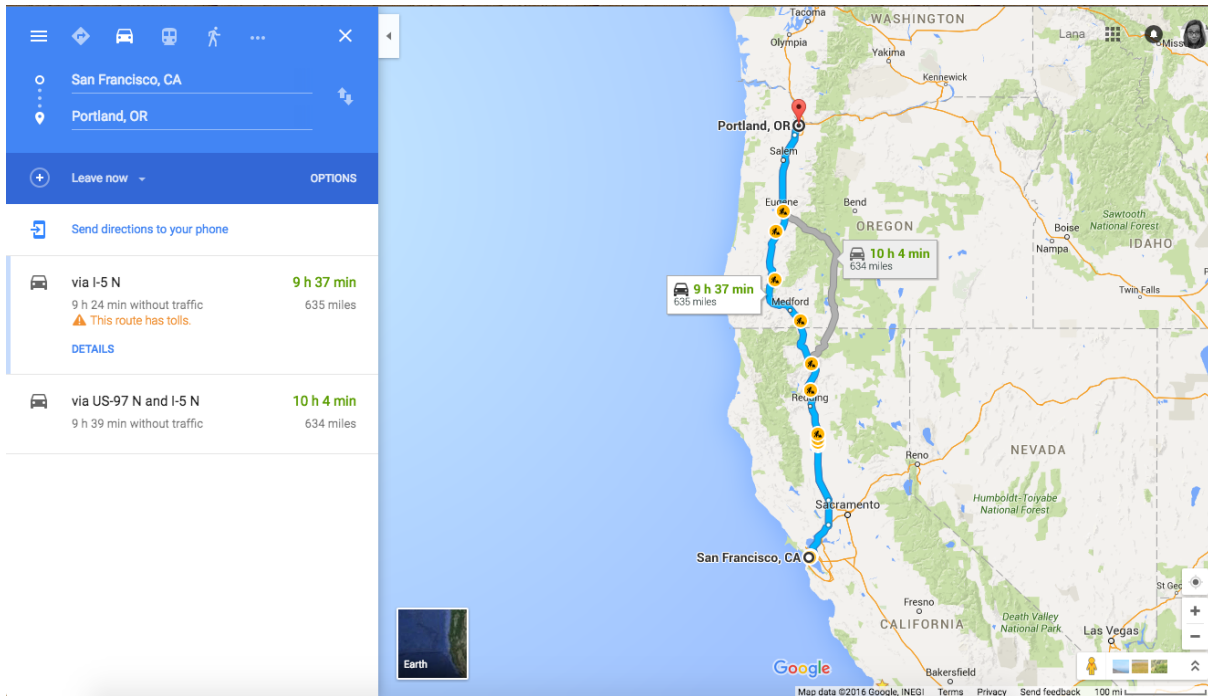
.....
.....
.....
.....
.....

Multiple Map Comparison

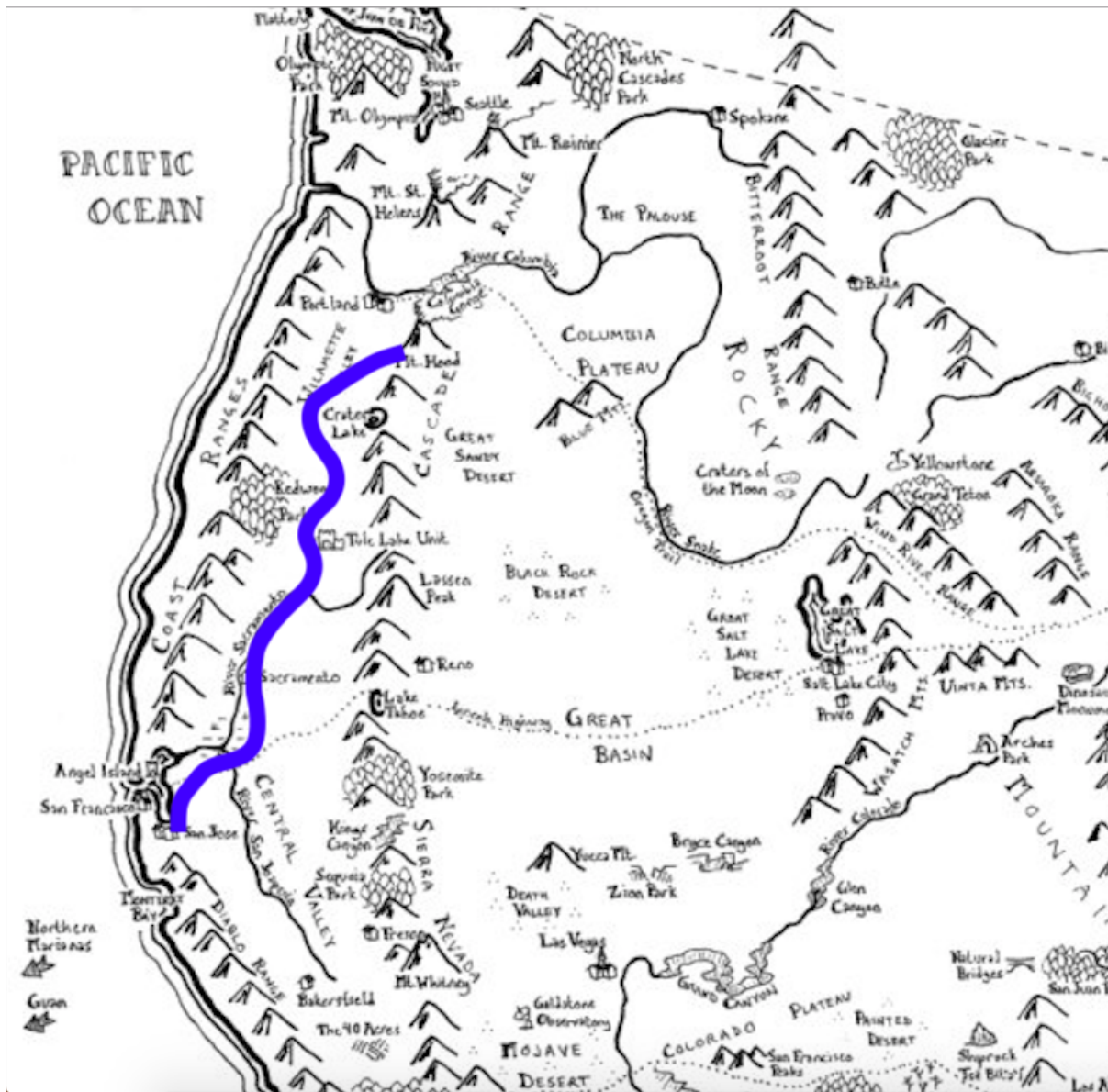
TRPINav Map



Google Map



Hand Drawn Map



9. Which trip looks more fun? *

Mark only one oval.

- TRPINav
- Google Map
- Hand drawn map

10. Please explain your choice of map for the previous question. *

.....

11. Which map gives a clearer route from start to end? *

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

12. Please explain your choice of map for the previous question. *

.....

13. If I ask, where the closest park is to the start of your trip, which map would you refer to? *

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

14. Please explain your choice of map for the previous question. *

.....

15. If I ask, where would you stop on your drive, which map would you refer to? *

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

16. Please explain your choice of map for the previous question. *

.....

17. If I ask, where is San Francisco with relation to the route, which map would you refer to? *

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

18. Please explain your choice of map for the previous question. *

.....

19. **If I ask, where is the California-Nevada border, which map would you refer to? ***

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

20. **Please explain your choice of map for the previous question. ***

.....

21. **If I ask, what cities are close to your destination, which map would you refer to? ***

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

22. **Please explain your choice of map for the previous question. ***

.....

23. **Out of the choices, which map would you use for determining directions? ***

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map

24. **Please explain your choice of map for the previous question. ***

.....

25. **Out of the choices, which map would you use for determining pit stops? ***

Mark only one oval.

- TRPNav
 Google Map
 Hand drawn map.

26. **Please explain your choice of map for the previous question. ***

.....

Appendix B

REAL-TIME USER EVALUATION

TRPINav Real-Time User Evaluation

Please answer all of the required questions and leave any feedback you would like in the comments section.

* Required

TRPINav Top-Down View Evaluation

1. Pick a park halfway through the route to picnic at. Which park is it? *

.....

2. If you had a hard time finding this, what would have made your search simpler?

.....

3. On a scale of 1-5, how confident were you in your ability to find the start and end location on the map? *

Mark only one oval.

	1	2	3	4	5	
Zero Confidence	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Full Confidence

4. What do the grey lines represent? *

.....

5. What does the blue line represent? *

.....

6. Did you find the legend helpful? *

Mark only one oval.

- Yes
 No

7. If you replied 'no' to the last question, were you aware that there was a legend?

Mark only one oval.

- Yes
 No

8. If you replied 'no' to the last question, is there a better placement for the legend?

.....

9. On a scale of 1-5, how comfortable do you feel with your understanding of the map? *

Mark only one oval.

	1	2	3	4	5	
Zero Understanding	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Full Understanding

10. Were you able to determine what the icons represented? *

Mark only one oval.

- Yes
- No

11. On a scale of 1-5, how useful did you feel the points of interest were in this view of the map? *

Mark only one oval.

	1	2	3	4	5	
Not useful at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very useful

12. On a scale of 1-5, how useful did you feel the road names were for the route? *

Mark only one oval.

	1	2	3	4	5	
Not useful at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very useful

13. Would you be more likely to toggle the road names on or off in this type of view? *

Mark only one oval.

- Toggle on
- Toggle off

14. Are there any aspects of the map you like? Explain. *

.....

15. Are there any aspects of the map you did not like? Explain. *

.....

.....

.....

.....

.....

16. Overall, how likely are you to use this map? *

Mark only one oval.

	1	2	3	4	5	
Not likely at all	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Extremely likely

TRPINav Navigation View Evaluation

Route from San Luis Obispo, CA to Bellingham , WA. Please use the map to navigation. Hit the 'H' key to review the key strokes if you forget.

17. Which part of the route looked most mountainous? *

.....

18. At what part of the route was elevation the lowest? *

.....

19. If you had a hard time answering the previous question, what could have helped you to better understand the terrain?

.....

20. Is the ability to pause and explore the area with terrain helpful? *

Mark only one oval.

Yes

No

21. Please explain your answer to the previous question. *

.....

22. On a scale of 1-5, how useful did you find points of interest in the navigational view? *

Mark only one oval.

1 2 3 4 5

Not useful at all Really useful

23. If your answer to the previous question is below 3, what would have made the points of interest more useful?

.....

24. On a scale of 1-5, how useful did you find the road names in the navigational view? *

Mark only one oval.

1 2 3 4 5

Not useful at all Very useful

25. If your answer to the previous question is below 3, what would have made them more useful?

.....

26. Did you prefer using a lower to the ground or higher off the ground camera? Why? *

.....

27. Does the terrain make you want to explore areas of the route more? Please explain. *

.....

28. Were you able to navigate to areas you thought were interesting? *

Mark only one oval.

Yes

No

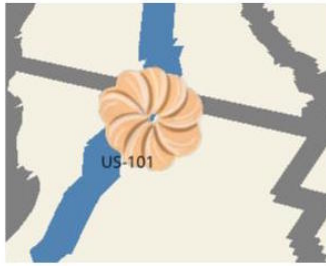
29. If your response to the above question is 'no', please explain why.

.....

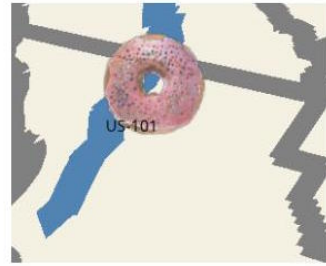
TRPINav Style



1



2



3

30. Of the three options for donuts, which style icon do you like best? *

Mark only one oval.

- Option 1
- Option 2
- Option 3



1



2

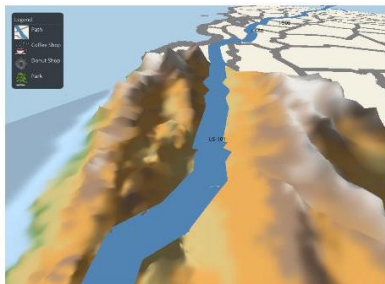


3

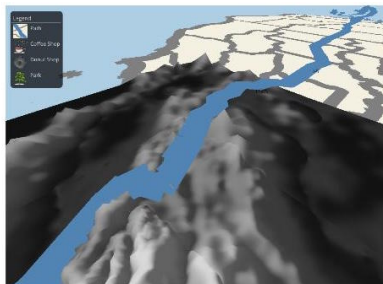
31. Of the three options for coffee, which style icon do you like best? *

Mark only one oval.

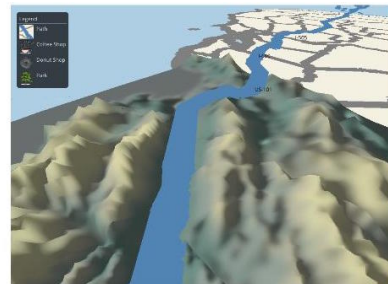
- Option 1
- Option 2
- Option 3



1



2



3

32. **Out of the above choices, which color scheme helps your better understand the geography better? ***

Mark only one oval.

- Option 1
- Option 2
- Option 3

33. **Out of the above choices, which color scheme is more aesthetically pleasing? ***

Mark only one oval.

- Option 1
- Option 2
- Option 3

General Feedback

34. **If you have any feedback you would like to give about the system, please do.**

.....

.....

.....

.....

.....

