Using Spatial Augmented Reality
for Collaborative Data Visualization in *Visible Disparity*

by

Niloofar Khoda Bakhsh
Stephen Leber
Pierre Sagarminaga

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Arts

in

Multimedia

and the Designated Emphasis

in

Interaction Design

in the

Graduate Division

of the

California State University, East Bay

Committee in charge:

Professor Gwyan Rhabyt, Chair
Assistant Professor Ian Pollock, Graduate Program Coordinator
Instructor Jason Anaya
Instructor Tana Lehr

Spring 2019

CALIFORNIA STATE UNIVERSITY, EAST BAY

MULTIMEDIA MASTER'S DEGREE

INTERACTION DESIGN PROGRAM

Spring 2019

Using Spatial Augmented Reality
for Collaborative Data Visualization in *Visible Disparity*

Niloofar Khoda Bakhsh

Stephen Leber

Pierre Sagarminaga

FACULTY COMMITTEE MEMBER APPROVAL

Ian Pollock	______________________________	Date: __________
Gwyan Rhabyt	______________________________	Date: __________
Tana Lehr	______________________________	Date: __________
Jason Anaya	______________________________	Date: __________

Abstract

Cell-phones and computers and their interactive surfaces have become common technological platforms for people to view data visualizations. Though these interfaces have had great improvements over the years, they provide only a singular experience and this is not entirely suitable or appropriate for groups of people wishing to analyze and discuss their data together in the same space. Visible Disparity is a group effort to design a collaborative space for exploring and discussing geovisualizations with the goal of freeing users from the confines of digital screens. The development team behind this project utilized a Human Centered Design approach to create an experience where multiple people can control the visibility of the datasets by moving Augmented Reality marker cards on the surface of a round table. The content's focus is on the increasing economic disparity within the city of Oakland, California, including data on rent and access to resources such as Bay Area transportation and public schools. The team used the Processing programming language to code the visualizations and the Spatial Augmented Reality(SAR) environment. The foundation of the project was built using Unfolding maps, a coding library specialized for geovisualizations, and PapARt, an augmented reality library that handles the detection of user interactions. PapARt functions with a camera and a projector housed in a pendant lamp that tracks and responded to users' feedback. The findings from user testing sessions with different groups of participants confirmed that SAR holds great potential for bringing people together for the sake of collaboration and conversations over data. Data visualizations in a SAR space can become part of educational environments that practice project-based learning to engage students in learning processes.

Keywords: Spatial Augmented Reality, Collaborative Data Visualization, Tangible User Interface, Human-Centered Design, Geovisualization

Contents

List of Figures

1. Original Concept

The world around us is filled with data, often rendered in different visual formats such as tables, charts, and maps. Over the years, the use of data visualizations to help identify new patterns and knowledge has become widespread. Some of these visualizations can provide interactivity, allowing users to zoom in/out, hover over information, and filter their search parameters on their cell phones and computer screens. However, this solo experience is not ideal for a team setting where groups are required to analyze and discuss data. Our project, called Visible Disparity, aims to transform this experience into a collaborative one where more than one person can view and interact with data.

Collaboration within this project offers users the freedom to investigate different datasets while discussing the subject matter with others. We theorized that this collaboration process can help users perceive and learn about data more effectively, especially when viewing complex or otherwise prosaic datasets. Therefore, it became our objective to free users from the restrictions of screens and encourage human to human interactions instead of human to computer interactions.

Our project was built on the principles and methodologies of Human Centered Design (HCD). IDEO, a well-known design firm in Palo Alto, California, defines HCD as “a creative approach to problem-solving that starts with people and ends with innovative solutions that are tailor-made to suit their needs”.[1] This means observing users, understanding their needs, designing for them, and testing solutions while allowing user insights to direct the iteration process.

In our efforts to create an HCD project that encourages collaboration and conversation, we investigated a cutting-edge technology called Spatial Augmented Reality (SAR). SAR augments the real

---

[1] "IDEO's Human Centered Design Process." UserTesting Blog. Accessed May 10, 2019. https://www.usertesting.com/blog/how-ideo-uses-customer-insights-to-design-innovative-products-users-love/.

world by displaying interactive digital information onto physical objects. Cameras and projectors work together to build a SAR space. Cameras track users' interactions. Projectors display relevant information based on those interactions. We decided this technology fit the goal of our project as it does not limit users to devices such as cell phones or Head Mounted Displays.[2] Since all users in a SAR space are able to view and interact with the same information together, collaboration and conversations are likely to occur. For this project, we re-examined how SAR might be utilized by people to discuss and interact with tangible, map-based data in an open and collaborative environment.

To help guide our investigations and showcase the capabilities of SAR technology, the subject of economic inequality was selected for our research. According to the Economic Analysis and Policy Division (EAPD) of the United Nations, there are two perspectives on what economic inequality refers to. One is concerned with unequal access to opportunities such as employment and education. The second is associated with the material dimensions of human well-being, including unequal income levels and health status.[3] We found this project would be a good opportunity to visualize datasets that might reflect this economic disparity within the location of Oakland, California. We therefore designated our project name as Visible Disparity.

The project Visible Disparity was developed over the course of fifteen weeks during which we created a table-top SAR experience. The project is comprised of a projector and a camera housed inside a lamp, a computer, and a round table. A total of six Augmented Reality marker[4] cards were set on the

---

[2] Head Mount Display (HMD) refers to a type of display device, worn on the head that is meant for creating an immersive experience for the users. For more information: https://www.techopedia.com/definition/2342/head-mounted-display-hmd.

[3] Helena Afonso, Marcelo LaFleur and Diana Alarcón. "Concepts of Inequality." *Development Strategy and Policy Analysis Unit in the Development Policy and Analysis Division of UN/DESA.* October 21, 2015. Accessed May 10, 2019. https://www.un.org/en/development/desa/policy/wess/wess_dev_issues/dsp_policy_01.pdf.

[4] Augmented Reality marker are visual cues which trigger the display of virtual information.

table next to a projected map of the City of Oakland. When a card is placed on the map a data visualization appears. By removing and adding different cards, users can control the data they are viewing. The datasets being used for this project covered information on rent and access to resources such as public transportation and schools. To create the project we used the Processing[5] programming language. We also utilized two coding libraries, one called PapARt[6], an augmented reality application, and another called Unfolding Maps[7] a library specialized for creating map-based data visualizations.

[5] "Processing.Org." Processing.Org. Accessed May 10 2019. https://processing.org/

[6] "PapARt: Paper Augmented Reality Toolkit – A Processing Library." Inria Aerial Image Labeling Dataset. Accessed March 08, 2019. https://project.inria.fr/papart/.

[7] "Unfolding Maps." Unfolding Maps: Unfolding Is a Library to Create Interactive Maps and Geovisualizations in Processing and Java. Accessed 10 May 2019. http://unfoldingmaps.org/.

2. Prior Art and Context

Rethinking user interactions for SAR is a topic many HCD researchers and artists have explored over the last few years. This exploration outside of traditional user interfaces has helped guide Visible Disparity. We took inspiration from Dynamicland[8] a non-profit organization based in Oakland, California founded by Bret Victor in 2014. The project consists of an entire building that operates as a walk-in computer. People can collaborate inside it by creating or interacting with premade SAR programs. Projectors track paper markers that the programs are printed on with data projections displayed around them. A user can then manipulate the projections by moving their associated tracker. (fig. 1) This is made possible by the unique programing language called RealTalk OS.[9] Dynamicland was founded around the idea of bringing the user's experience out of a small isolated computer screen and into the real world.

The team behind Dynamicland attempts to give users better interactions between themselves and the digital content. For Visible Disparity, we also use SAR programs to help run our augmented environment, although our code remains within Processing. Dynamicland exists within a permanent location-based collaborative space. In comparison, Visible Disparity relies on a portable setup that can be easily transferred between different locations. There are also content differences. Dynamicland would like to help others learn to code and their setup is used to make the interactions of the learning process fun. We share similar goals for Visible Disparity although our focus is more on the possibilities that SAR can bring to a user's experience with concern to geovisualizations[10].

---

[8] "Dynamicland." Dynamicland. Accessed March 08, 2019. https://dynamicland.org/.

[9] Realtalk is the operating system behind the authoring environment at Dynamicland.

[10] Geovisulaizations mean using a set of protocols to assist in analyzing geospatial data and applied to interactive visualization

In terms of technology, a similar project to Visible Disparity is *SyMAPse: Design and Evaluation of an Augmented Reality Map*.[11] *SyMAPse* is a SAR map developed and led by Julia Chatain at Inria, the French national institute for computer science and applied mathematics in 2015. (fig. 2) It is based on the long-standing practice of annotating on map directions and labeling places of interest. Through her project, Chatain argues that the existing interactive maps are mostly limited to standard interactions that are commonly displayed on digital screens and input devices such as mice, keyboards, and/or multi-touch inputs found on smartphones and tablets.

SyMAPse involves projecting a virtual map onto a physical piece of paper using the Processing libraries, PapARt and Unfolding. The researchers behind *SyMAPse* had previously explored and compared three different input modalities[12] for this SAR map: touch-based, object-based and spatial. These modalities are commonly used for map functions with three interaction techniques. These interactions are zooming, panning and changing of the base map. Chatain found that users expected the touch-based interactions to react with just as much preciseness as with the zooming and panning as with a mutual capacitive screen such as a touch screen on a mobile phone. The projects Visible Disparity and *SyMAPse* make use of the same technology. However, *SyMAPse* is a single-user experience that allows for adding annotations on the map, whereas Visible Disparity takes advantage of the same technology to create a collaborative platform in which all users can equally contribute to the experience.

[11] Julia Chatain. SyMAPse: Augmented Interactive Maps for Subjective Expression. Human-Computer Interaction. 2015. Accessed March 08, 2019. https://hal.inria.fr/hal-01191429v2/document.

[12] A modality is a single independent channel of sensory input/output between a human and computer or a computer and human.

SketchSynth[13] is another project involving interactivity and a SAR environment. *SketchSynth* was created by Billy Keyes at Carnegie Mellon University in 2012. The project is able to create tangible user interfaces from sketches of circles, rectangles, and lines on a physical piece of paper. (fig. 3) The project adds digital interaction onto non-digital interfaces. This gives the user the ability to create their own buttons, switches, and sliders. After the shapes are drawn, the program renders an interface and sends data to anything that is configured to receive it.

This particular data is created from Open Sound Control (OSC)[14], a protocol that provides a platform for communication between computers, devices, and other multimedia devices that use networking technology. The program is limited to only three input types: momentary buttons, toggle switches, and sliders. These three inputs cover most possibilities that a control panel can have. Sliders cover most linear or rotary sliders such as knobs. Switches cover many buttons that act like toggles.

The hardware setup uses a projector and a web camera that aligns to the four corners of a projected rectangle on a piece of paper. The webcam first detects edges in drawn controls. Then it looks for the contours of shapes and uses the ratios of these shapes to determine the type of control to be used. The program does not detect when a hand is touching the paper, however, it can detect the hand and improvises accordingly in real time. The creator, Keyes, was inspired by the many complex control panels that he encountered as a child and by the imaginary controllers that he used to make from drawings on cardboard.

[13] Interactive Art and Computational Design Spring 2016. Accessed March 09, 2019. http://golancourses.net/2012spring/05/13/billy-keyes-final-project-sketchsynth/.

[14] The Open Sound Control is a programming protocol specialized for networking audio capable applications.

Another project that also creates interactions for SAR environments is *OmniTouch*.[15] *OmniTouch* is a body-worn projection/sensing system created by Chris Harrison at Microsoft Research in 2011. (fig. 4) It enables the person wearing the device to create a graphical, interactive, multitouch tangible user interface on any surface that is available to them including their own skin. This system affords for the manipulation of those interfaces so it could provide interaction similar to that of a smartphone. By projecting this interface onto such surfaces, no calibration is required because of how the system is worn on the user's body, the proximity of the camera, and the tracking of the finger for touch input.

SketchSynth and *OmniTouch* are both valuable resources for the research and development of SAR and their user-testing findings. A couple of key differences distinguish them from our project Visible Disparity. Besides the fact that *OmniTouch* is wearable technology, the main goal is to give the user the opportunity to create interactions. Visible Disparity is a project that creates such interactions for the user and facilitates the users' experience in investigating data.

Wage Islands[16] (fig. 5) is an interactive installation by artist Ekene Ijeoma that looks into the wage and housing inequality in New York City. The artwork features a 3D topographic map of NYC installed in a tank, filled with dark blue water. The map consists of topical elevations based on the median rents from $271 to $4,001. Over 500 pieces have been laser cut and put together to form these sculptures. Outside the tank, there is a display and a button that shows the hourly wage, $8.75, the city's minimum wage at the time.

---

[15] "OmniTouch: Wearable Multitouch Interaction Everywhere." Chris Harrison | Rethinking the Progress Bar, © Chris Harrison, www.chrisharrison.net/index.php/Research/OmniTouch. Accessed June 1st, 2018.

[16] Ekene Ijeoma.. "Wage Islands: Immigrants Is a Sculpture Which Submerges a Topographic Map of NYC Underwater to Visualize Where Low-wage Immigrant Workers Can Afford to Rent." Ekene Ijeoma. Accessed March 09, 2019. https://studioijeoma.com/#.

Upon first viewing, this 3D map is almost entirely hidden under water. Holding down the button slowly raises the sculpture up, representing the increases in the wage shown on the display. Pressing the button lifts the map from the water, presenting the topographical representation of neighborhoods from people who earn different amounts of money and where they can afford to live. As seen, the hourly wage raises and more areas of the map are revealed. When the button is not pushed, the map slowly lowers back to its original state below the water and the display resets to show the minimum wage. This interactive artwork displays the contrast between the resources that low-income individuals have access to versus high-income individuals, essentially picturing the reality of housing affordability. *Wage Islands* demonstrates how an artist challenges the way viewers perceive and interact with data visualizations. The unfamiliar form of this moving artwork and its subtle interactivity work together in a piece meant to engage the viewer on an empathetic level.

The *Anti-Eviction Mapping Project*[17] (fig. 6), or *AEMP*, is a website that documents projects related to the experiences of displaced residents in California. The *AEMP* is a collaborative multimedia data mapping project that started in 2013. It primarily focuses on residents and housing data from California. The website shows stories of displacement, gentrification, violence, and other topics impacting California residents. These visualizations of data overlaid with maps of California are meant to convey the impact of the housing crisis in ways that written data simply cannot. A quick look at the programmed maps and recorded video stories gives a viewer a clear and daunting sense of the current state of the housing and eviction crisis. Their list of data maps includes real data visualizations of evictions, demographic maps, surveillance, development, and narrative spaces.

[17] Anti-Eviction Mapping Project. "Anti-Eviction Mapping Project." Anti-Eviction Mapping Project. Accessed March 09, 2019. https://www.antievictionmap.com/.

The Narratives map section shows a bird's eye view of San Francisco with dots placed around the map. Each dot represents a person's location and story listened to through a SoundCloud file attached to the map. Hundreds of dots start to convey a clear picture of the number of residents' similar experiences. Real housing data is used as well as reports on crime and police shootings by category of race. Housing and rent boards, though not available in certain parts of California, were accessed to help map out the various data visualization displays, online charts, interactive videos, and recorded interviews. *AEMP* is trying to make these overlooked stories of displacement visible. It ultimately aims to combat the issues they are trying to map.[18]

[18] Johnson, Sydney. "The Art of Data-telling." East Bay Express. March 01, 2019. Accessed March 09, 2019. https://www.eastbayexpress.com/oakland/the-art-of-data-telling/Content?oid=4747483.

3. Project Development and Methods

3.1 Technology

Hardware

A good amount of research went into choosing the hardware for the project. We had decided to base our project off of PapARt, an open-source software development kit for SAR. We investigated images of the hardware that the team behind this technology, RealityTech, had posted on their webpage. (fig. 7) We wanted to see if we could identify any manufacturer or model names that were visible on the hardware. Through this investigation, we identified some of the pieces that they were using for their SAR device, Nectar.[19] We planned to purchase the exact same hardware so we could eliminate any unnecessary compatibility issues with the library.

We were able to identify the exact projection device used for the Nectar SAR hardware platform. This was an ASUS P3B short throw LED projector. (fig. 8) We also found out the type of computer that was being used, which was a small form factor[20] Intel NUC computer. The webcam was not identified so we chose to go with whatever webcams were readily available at our disposal. We decided before settling on the final hardware for our project that we would run a few tests on various models of projectors, computers, and webcams.

We tested two projectors, an LG Minibeam ultra short throw LED projector borrowed from the graduate program. The other projector we tested was the ASUS projector which was purchased for the project. The LG projector was used initially for the first prototype which was made out of cardboard and

[19] "Nectar Platform." RealityTech - Spatial Augmented Reality, RealityTech. Accessed May 10 2019. http://rea.lity.tech/en/natar.html.

[20] Form factor refers to a product's size, basic design and functionality.

served for the rough testing of the PapARt SAR technology. However, this projector was bulky and did not have the crisp, high-quality image we were looking for.

We moved on to the ASUS projector and after a few tests, it was decided that it would serve as the projector for the project. It was smaller in size compared to the LG Minibeam, was much cheaper, and had better resolution on surfaces when positioned at short distances. After settling on the ASUS projector, we began the testing phase of it with PapARt the results of which can be found in the results section.

Some of the other struggles we encountered in our technology setup process included choosing which computer and operating system were the best for development as well as the computer form factor. We tested two operating systems, macOS and Ubuntu, to see which one had better tools for installing the many applications and dependencies that our code libraries needed.

Testing revealed that the macOS operating system was the easiest for installing and configuring the PapARt library. Nearly every software package was installed and configured via the command-line interface, Terminal. The Ubuntu operating system required more configuration for the Java dependencies. The webcam was having problems being detected by the PapARt library. Unlike Ubuntu, macOS was able to detect the webcam easily and through this, AR markers could be identified immediately.

We tested two webcams and their various placements on our prototypes to make sure that the projection space was being captured proportionately. Calibrating the zoom and focus of the cameras was important so that AR markers could be easily detected from different heights. The height distance from the surface of the table mattered to one webcam, but not the other. This was due to the frames per second rate (fps) of each camera. The first webcam, a Logitech C920 had a maximum resolution of 30fps in 1080p. The one we chose, a Logitech Brio, had a max of 30fps in 4K resolution. (fig. 9)

After experiencing the high functionality of the PapARt library with the webcam, we chose macOS Sierra for the final operating system. Next, we needed an appropriate computer form factor for the hardware structure. Portability was an ideal goal as much as performance. We tested different Apple computers through four successive prototypes.

Initially, a MacBook Pro was used for the first prototype to test the PapARt library. We then moved on to an iMac for our second prototype to test the performance. After a few weeks with the iMac, we switched to a Mac Mini for portability. After some feedback about the performance of the Mac Mini, we used a Mac Pro for our third prototype. This was the prototype that we worked with the most.

After having a functioning version of our project on the Mac Pro and choosing all of the necessary hardware, we transferred all of the work back to the Mac Mini. This became our ultimate computer for our fourth and final prototype. We then worked on the camera-projector alignment and optimal distance between the projector, webcam, and surface. Placing the projector and webcam at ~53 inches (134.62 centimeters) from the table surface was an adequate height for the projected images to be detected easily while maintaining high-quality resolution.

The final projector and webcam we chose went inside of an Ikea Melodi pendant lamp. (fig. 10) A Bogen ball head secured the projector through its tripod socket. The ball head was held in place by an all thread rod that passed through a hole on the top of the lamp which was then secured with a hex nut. The webcam was taped onto the top of the projector with gaffer tape. The lamp was then installed as a ceiling fixture. The final measurements allowed the hardware to disappear from the users' line of sight so that the SAR on the table surface could be the main focus of the project.

Software

The technology and hardware used for the project were dependent on open-source software. We used Processing 3, an open-source graphical library and integrated development environment[21] (IDE) based on the Java language. Two open source libraries for Processing were used. Together, these libraries helped create a SAR user interface that displayed map-based data visualizations. These libraries served as the base code that ran and manipulated the input and output devices for the hardware.

The search for an open source SAR software development kit[22] (SDK) began in the Spring of 2018. The search for an open source SDK specialized for SAR was not an easy task. Through our boolean searches[23], we discovered that the term "Spatial Augmented Reality" is not commonly used outside of academic research. By the end of Spring 2018, we had found an open source library for Processing called PapARt for the Processing IDE. During the summer of 2018, we downloaded PapARt 1.2 and explored its contents. Within the Processing file structure, the PapARt library had to be stored within the libraries folder. We also downloaded and experimented with the examples folder.

The PapARt documentation recommended hardware including a short-throw projector, a web camera, and a depth camera. We gathered this hardware and configured it for the SAR library. In the early phases of our project research and development, we had successfully configured the hardware and software required to run most of the basic example sketches that it offered. The depth camera was abandoned after we changed the scope of our project.

---

[21] IDE refers to a visual coding environment for developers.

[22] A Software Development Kit is a set of software tools specialized for development on specific application platforms.

[23] A type of search that allows users to combine keywords with operators such as AND, NOT, and OR to further produce more relevant results.

Processing 3.3.7 was used to establish the coding environment that employed the two open-source libraries. PapARt, which stands for Paper Augmented Reality Toolkit, required additional Processing libraries that handled the math, networking, 3D image processing, camera tracking, depth perception, marker detection, and video streaming. The additional libraries we added were Jama, PCA Transform, PeasyCam, OSCP5, Processing Video, Toxiclibs, SVGExtended, Processing TUIO, Skatolo, GuiModes, Reflections, and Libfreenect. The final library for the PapARt environment was JavaCV which embedded some other pre-compiled libraries such as OpenCV, FFMPEG, and ARToolKitPlus which was used for tracking the AR markers. PapARt 1.2 is used to display SAR images onto physical surfaces. It can track ARToolKit markers, small objects, and colors. It displays the AR images through a video projector. This system was used as the basis for the main software and also serves as the medium for displaying the map-based data visualizations.

Unfolding 0.9.9, another Processing library, provides an application programming interface[24] (API) for designers to quickly create and customize geovisualizations for the purpose of enabling interactive maps that afford for basic interactions. Such interactions include zooming and panning. Data visualizations can be created using geo-positioned markers to display data on a tile-based map sourced from map providers or Web Map Services (WMS)[25] such as OpenStreetMap.

The software behind Visible Disparity is installed and operated on a 2014 3rd generation Apple Mac Mini running the macOS Sierra version 10.12.6 operating system. The physical hardware for the structure uses one input and one output. The input is a Logitech 4K Pro webcam for capturing the video feed and detecting AR markers. The data collected from this camera is then uploaded to the computer. The computer streams the 3D positions found by the tracked AR marker to a projector. The projector is

[24] API is a set of instructions used by developers to call down data from another source.

[25] WMS is a protocol for providing geospatial data.

an ASUS P3B with WXGA[26] resolution. It serves as the output for displaying the images on the surface. The computer processes the information from the AR markers in real time and the projector displays multimedia content on the surface of a table. This allows for the creation of a user interface that affords for an interactive, multi-user experience through collaborative SAR.

3.2 Content

To help convey the capabilities of Visible Disparity, we focused on the area of Oakland, California and how it has been impacted economically. For this, we needed datasets that included information on median rent, public transportation, and school locations. The data we used spans the years 2008-2017. Initially, we wanted to map data over time but narrowed down our parameters later on in the project. Any datasets that were collected had to be map-based so that Processing could place the data exactly where it needed to go. This meant using GIS[27] data.

We searched several websites including the US Census Bureau. The Census remained a consistent reference for many of the sources we continued to find. Because we were implementing SAR, the data that was collected appeared on the projected map using AR markers. Each of the markers needed to be tied to a specific dataset that could be mapped over the area of Oakland. The datasets needed to be in a JSON[28] file format or more specifically a GeoJSON[29] file.

[26] Wide Extended Graphics Array has a display resolution of 1366x768 pixels and a wide screen aspect ratio of 16:9.

[27] GIS refers to analyzing data in visualized geographical layers for spatial visualization

[28] Java Script Object Notation is a file format based on Javascript that uses human-readable text that works with data with no complicated parsing and translations.

[29] A JSON file with longitude and latitude coordinates.

This made the datasets easier to work with when assigning GIS data to different places on our map. For other datasets, we simply needed to convert one GIS file type to another. It was convenient to use an online GIS file converter called MyGeodata Converter[30], which permitted us to transform any map based file format into a GeoJSON format. This function provided we had longitude and latitude coordinates.

The initial idea of the project required access to data on rent prices in Oakland sorted by neighborhood and other factors such as the location of public transportation, schools, and liquor stores. Data sets collected needed to be topically different and occupy the same location within the same time periods. A few promising datasets were found covering these topics, however, these datasets tended to lack longitude and latitude coordinates.

We started to bring available geospatial datasets into the project to have an earlier start in coding the visualizations. Other factors affecting housing prices were discussed, such as having access to resources such as public transportation. Generally speaking, apartments closer to a BART (Bay Area Rapid Transit) station have higher rent[31].

Data on BART Station locations were used for testing the mapping capabilities of Unfolding. When this file was coded into a simple example sketch named “SimpleMarkerApp”, it displayed gray circle markers over a map of Oakland. The circles corresponded to the locations of BART stations in Oakland. We tested additional datasets using the same coding structure.

---

[30] “MyGeodata Converter.” MyGeodata Converter. Accessed May 10, 2019. https://mygeodata.cloud/converter/

[31] "Homes for Sale, MLS-based Real Estate." Estately. Accessed April 13, 2019. https://www.estately.com/.

We then found GeoJSON data for school locations created by the Alameda County Data Sharing Initiative website[32]. This site provided an online view of a map from the Oakland area with many circular point markers placed over the map indicating the location of established schools. School locations in the Alameda County could be downloaded from the website in a variety of formats including GeoJSON.

During our research, we discovered that certain datasets found on websites were missing crucial information such as the geographical coordinates. To work around this, we had to map this relevant data to our map of Oakland. When we could not find a neighborhood, we sometimes could find a Census Tract[33] area instead. Census Tracts have latitude and longitude data so we could map the center points of each Census Tract and then apply our found datasets to those locations. Due to the time constraints of our project this method was never fully applied.

Zillow is a well known real estate website that also owns another real estate finder service called Hotpads[34]. Though an excellent source for homebuyers we could not download the actual data from the website per Zillow's instructions that no API would be available for developers. We discovered a way around this by manually copying each residential address of the high and low end properties, pasting them into Google Maps, exporting them as KML files, and then converting those into GeoJSON files.

GeoJSON files continued to be the ideal file format for our code and made it easy to plot data points onto our map. This work-around process also became useful for the Data USA[35] source where we

---

[32] "DataArc." DataArc. Accessed April 13, 2019. https://www.dataarc.ws/.

[33] An neighborhood sized area designated by the US Census Bureau for collecting population data.

[34] "HotPads." HotPads - Go Ahead. Rent around. Apartments and Houses for Rent. Accessed May 10, 2019. https://hotpads.com/.

[35] "Data USA." Data USA. Accessed May 10, 2019. https://datausa.io/profile/geo/oakland-ca/#housing.

found that we needed to include one extra step by matching datasets with their Census Tract area via an online source called ArcGIS.

3.3 History of Development

The process of designing geovisualizations started with creating a map of Oakland in Processing. Unfolding's inbuilt classes made it convenient to select a portion of the map to be displayed by adding the latitude and longitude coordinates of the city. After that, we restricted the map zooming and panning functions as these features were not important to the project. Next, we imported and read the GeoJSON files in Processing and created markers for each data point. When mentioning the term "Marker" in the Unfolding library, we are referencing the point, lines or polygon shapes representative of geospatial data. These types of markers appear in the default color of gray on top of the map. It was important to be able to display different datasets overlaid on top of each other. The library's default colors, however, did not work well for this design and so we had to customize the colors in the code.

We programmed a Java class to set different colors, opacity, shapes and outlines for point markers. There is an existing class in Unfolding called SimplePointMarker. Our JAVA class overwrites the default class in order to change the colors and shapes of the markers. (fig. 11) This class was later called when creating the point markers in void setup()[36]. (fig. 12)

Customizing polygon markers is different from point markers. We discovered that unlike point markers, creating a separate class was not required. After reading through the Unfolding API, we found a method called setColor(), which sets the hue for polygon shapes. Another method was found called setHighlightColor. This changed the polygon colors when active (hovered over by mouse). We coded a custom function called changeColor() after learning this. Inside this function, the aforementioned

[36] A function reserved for initializing code at the beginning of a program

methods were called and new color values were set for the polygons. This function was then called in void draw()[37]. (fig. 13).

After creating the visualizations for all the datasets we brought them into the project SAR space. This meant integrating Unfolding with PapARt. Creating a project that takes advantage of both of their features along with their built-in functions and methods was difficult to accomplish. To create a PapARt app, the Processing sketch required at least two files. The first was the main sketch which consists of general information such as libraries, global variables[38], void setup() and void draw(). The second file is the customized PapARt app where visualizations could be programmed to be displayed when an AR marker was detected. We extended existing classes in PapARt such as "PaperScreen" and "TableScreen" in order to use the AR detection features. This meant that the app was based on an existing class, but with some customized features that were coded in. In this file, the AR marker that triggers the visualizations is defined with Unfolding functions. Simply put, the pseudocode[39] reads:

If "specific AR marker" is detected on the table's surface, send the specific data visualization to the projector, display it on the table's surface

This structure was based on the "Two_Circles" sketch copied from an example that was included with PapARt. In the code, each circle was assigned an AR marker such as "A4-default.svg" shown in the example below:

public void settings() {

---

[37] A function reserved for running code continuously throughout a program

[38] A variable that can be accessed and found throughout a program's code

[39] Pseudocode uses an informal description of normal programming language that is intended for human reading rather than machine reading.

```
  setDrawingSize(210, 297);
  loadMarkerBoard(Papart.markerFolder + "A4-default.svg", 215, 279);
  setDrawOnPaper();
 }
```

Further code was copied to run the active drawing of the code:

```
public void drawOnPaper() {
  //background(40, 40, 40);
  fill(0,0,255);
  ellipse(50, 50, 50, 50);
    }
}
```

By using the same logic and organization, the next step was to bring Unfolding data visualizations into PapARt. However, in attempting to integrate this map library, we could not figure out why visualizations appeared on the surface regardless of an AR marker being present. The code shown below is what was being drawn when the Processing sketch was run:

```
public void drawOnPaper() {
  noStroke();
  smooth();
  noFill();
  markers = MapUtils.createSimpleMarkers(features);
  map.addMarkers(markers);
    }
```

We created conditions in the code for the map to appear only when the camera detected the AR marker:

```
public void drawOnPaper() {
    if(markerboard.isTrackedBy(camera)){
        markers = MapUtils.createSimpleMarkers(features);
        map.addMarkers(markers);
    } else {
        println("not working");
    }
}
```

Further attempts resulted in persistent coding errors. "NullPointerException" remained one of the consistent errors. Despite our attempts to resolve it, this error has been regarded in many forums online to be one of the most common and easily fixable errors.[40]

Around this time, we were given access to the latest version of PapARt. In this version, the creator, Jeremy Lavviole had fixed the errors of the previous version. We investigated the examples provided in the new library. One of these examples was an application called "Light". (fig. 14) We noticed that this is one of the few sketches that uses multiple AR markers and runs without any errors. Using the same concept and code structure we started our next attempt in merging Unfolding and PapARt. We were able to assign one of our data visualizations to an AR marker. However, the visualization did not disappear even if the AR marker was no longer present. One workaround for this

[40] "Processing Forum - Why do I get NullPointerException?" Processing Forum - Why do I get NullPointerException? Accessed May 10, 2019. https://forum.processing.org/two/discussion/8071/why-do-i-get-a-nullpointerexception

problem was to have an AR marker that clears all data points from the map. This solution functioned, however, we needed alternative options that could provide a better user experience.

So, we analyzed the code further and figured out this issue derives from Unfolding classes, not PapARt. The function clearMarkers() was not programmed to accept the name of the specific dataset and it removed all data points all at once. We started our research for another function or class in Unfolding that allowed for disabling visualizations of a specific dataset. Unfortunately, no solution was found. We contacted the creator of Unfolding, Till Nagel, and with his help, we were able to identify an existing class in Unfolding called MarkerManager. The methods and booleans inside this class allowed us to enable or disable each dataset independent of others.

We then created another sketch in Processing in which we imported all the datasets and used the aforementioned methods with the keys on the keyboard. Pressing the "s" key would display data on schools. Pressing it once again made the school data disappear. (fig. 15) After understanding how the methods worked with our datasets, we updated our PapARt code. The assigned AR markers worked perfectly with Unfolding methods and our final sketch was ready for refinement.

4. User Testing

4.1 The Users

SAR is a fairly new concept in the emerging field of Augmented Reality. So understanding users' intent to interact with something that is physically not there was crucial for us. We needed to understand how SAR creates a space that encourages users to learn collaboratively together. We researched the affordances[41] that a tangible user interface could provide and then we began testing.

Before inviting users to test our project there were a few points we needed to consider. First, testing sessions had to be conducted with a group of two to six people. This range was based on the average space required for one person to stand around the table. In a few cases, we tested some features with only one user. In the later user tests we invited groups of users once they became available. Our project stands apart from the standard interfaces that data visualizations are commonly displayed on such as paper lists or computer monitors. The interface could be unfamiliar to people who are not in the business or technology industries. The second important factor for us to consider was to observe how people from non-technological backgrounds reacted to the project. The final consideration was user's educational levels. For a college level audience, we decided that economic disparity could be an easy to understand subject of interest so we tested this theory to confirm.

4.2 Methods

The first method we applied in user testing involved explaining the project to users prior to the actual testing session. This gave them the context of tangible user interfaces and SAR before interacting with any part of the project. It was important to explain the limitations of the system to inform the users

[41] Affordance refers to the possible interactions that users perceives for an object based on their goals and past experiences. In this project, the cards on the table are considered affordances.

of features that were still in the development stage and were not responsive to their interactions. One advantage of this method was that it reduced testing session times when users reported that they had time constraints prior to participating.

The second method we used involved allowing users to explore the project without providing context or any prior explanation of how the interface works. This produced an observation period where users worked out how to interact with what they saw on the table. This approach typically meant that the users had more time to be involved in the testing and could provide more qualitative data to the research.

We used both these methods depending on what stage of user testing we were at. During the first and second testing sessions the project was not in a shape to stand alone and further explanations were required. However, in the second and last user testing sessions, we conducted the SAR experience without giving any context to the users. Users who had no prior knowledge of the project helped us to better evaluate our results.

4.3 Evaluation Metrics

Our evaluation metrics changed slightly for each testing session, but some were used for assessment after all testing sessions were conducted. We measured users' engagement time which is a common metric in usability testings. The time users are willing to spend in an experience is usually associated with how enjoyable the experience is for them. We documented by writing notes or video recording so we were able to look back and approximate the learning curve and engagement time for each round of user testing.

We needed to evaluate the efficiency of our project affordances. Since the project has intentionally limited instructions, the users had to work out the functionality of cards on their own. Our job during testing sessions was to track users behavior and note whether users reach out to touch or grasp the AR marker cards. We wanted to see if they placed the cards on the designated surface and whether they kept placing and removing the cards on the map to activate and deactivate visualizations. We also needed to know how difficult or easy a task it was for users to understand how the project interaction worked.

One of the main goals of our project was to create a collaborative space for exploring data visualizations. Conversations between the users was another significant factor that could indicate whether the space we had created provided for such interactions. For this, we paid close attention to whether users discussed the data correlations and patterns with one another. To accurately measure our success in our goal, it was important to consider whether users knew each other or were strangers prior to user testing sessions.

As well as observation, we followed up user testing with an additional feedback interview. We asked questions to evaluate our users' general impression of our project. We wanted to see if users could determine, based on what they had experienced, what our project was about. What did they like about the project? What could we do to improve their experience? We documented and analyzed the answers to these questions after each user testing session.

We changed some of the criteria for each user testing session based on what we required feedback on. In the second user testing session, we asked users to provide feedback on their experience of standing at a round table versus a rectangular one. In the same session, we provided two different technologies for testing, our original technology PapARt, and our backup technology, OpenCV. We asked questions about each piece of technology in the feedback discussions. These specialized questions were one-time evaluating points that only applied to that particular round of user testing and were not meant to be repeated for the next successive rounds. This same process was relevant for the follow-up questions that we asked based on some specific reactions from users. For instance, during the third round of user testing, we noticed that users held the AR markers cards up in the air instead of placing them on the table despite the small instruction text we used that read “Place cards on the map.” (fig. 16) We kept a note of this reaction and in the interview session afterwards we asked the users what caused this reaction. Users felt the cards were light so they thought they could just simply hold them above the projection surface.

4.4 Testing Session Structure

Here, we briefly go over how each testing session was structured and what was tested. The results of our findings will follow in the results section of this paper.

The first session was held in the second week of March 2019 where we presented a few examples from the PapARt library to the users. We also provided some sketches and paper prototypes of a slider with data visualizations. The slider was a straight line drawn on a piece of paper. An eraser was put on top as the slider handle. We used a few hand-drawn sketches of data visualizations and these were basically lines and dots on pieces of paper. Once the users moved the repurposed eraser on the paper slider, we manually placed different pieces of paper on the map to complete the visualization. Prior to the beginning of the experience, we explained to the users that our job, placing data visualizations on the map, would be replaced with the program that we were coding. At that point in the project, our focus was on the economic disparity in Oakland, so we raised questions about that subject matter. Although the slider effect was later removed from the project, due to technology and time constraints, we still received valuable feedback from our users on tangible interfaces and the interactivity of the piece.

Between the first and the second testing sessions held in the third week of April, the project had undergone significant changes. We had moved on from a rectangular table to a round table and had started working on our backup technology while still continuing our original technology solution, PapARt. We also created four new datasets for this user testing session and set up two tables. We used a round table that presented our backup solution and the rectangular table presented our original technology. Both were tested and followed by feedback sessions.

We held our final user testing session in the first week of May. During the two-week period between sessions, the project had come closer to its final version. We had managed to run our original technology, added more datasets, and made some aesthetic decisions for the look of the final version of the project. This was the last and most official user testing session so it was important to keep track of all the different aspects of the project that were being tested. For this, we recorded two videos of each group of testers. One, a top-view angle by the project's camera that recorded user's hands and the table surface. Another video recorded a medium-shot of users and our project structure. Both of these recordings were watched several times later in the evaluation process.

5. Results

In this section we present results on different areas of our project including using SAR for collaborative spaces, technology, datasets, user testing, and the final project.

5.1 Using SAR for Collaborative Spaces

We found the case for using SAR was justified once serious user testing got underway and the technology we were working with was figured out and properly implemented. The final project setup consisted of a ceiling lamp that housed the projector and the camera. The ceiling lamp was set up approximately ~52.5 inches above a round table. The rest of the hardware, such as the computer, was placed out of site of the users. The table and tracker cards were left out in the open and remained clearly visible to all participants. This interaction space leaves plenty of room for people to move around the table and interact with the AR marker cards without being impeded or distracted by the technology.

Some small text asking users to place the cards on the map was placed around the border of the projection. This meant users, upon first engaging with the project, had little else to go on in terms of directions than to pick up the cards and see what happens. This was the proposed affordance of the markers. We hypothesized that users would be intuitive enough to pick up the cards as there were no other devices for them to use. They only had to approach the table space and move the cards to understand how the project worked.

The final version of the markers were flat, disk-shaped, and 4.2 inches in diameter and laser cut from plastic. This was a comfortable hand-sized, AR marker. Their rapid, real time data displaying response rate also made them satisfactory controls for users.

The shared space meant users had to stand in close proximity to each other while moving datasets with the markers. Actively comparing the datasets in real-time in front of them seemed to prompt discussion as to what the comparisons they displayed might mean. We were pleased to find that many of the things we had hypothesized, such as automatic discussions about the data arising from user interaction, were confirmed once we ran the project.

5.2 Technology

One of the objectives of our project was to create a data visualization template that other open source developers or users could take advantage of. This template could be implemented through a SAR device using similar hardware and open source software. Ideation, rapid prototyping, usability testing, and other design methodologies also contributed to the final version of the project.

The hardware that went into building the prototypes for demonstration purposes was in research and development for the good part of fifteen weeks. Figuring out what model of computer and camera was needed for getting the SAR part to work remained an ongoing challenge. Through most of the project development we continued to use an Apple Mac Pro computer.

The operating systems of the Apple computers we used, macOS Sierra and High Sierra, demonstrated efficiency at solving basic software needs. The Mac Mini computer was ultimately chosen for the final setup due to its small size and port availability. It allowed us to plug in the necessary components including the camera, projector, keyboard, mouse, and monitor. This configuration gave us the freedom to test out the project structure and the extended plug-n-play devices. This was especially important when configuring the PapARt application which needed access to the physical hardware.

After settling on the ASUS P3B projector we ran a few tests to determine the best configuration of the projection. Testing revealed that the projector’s mode of display as well as height placement mattered in terms of how the image from the computer was being viewed on the surface of the table. We tested the available modes of projection and found that the default projection mode, “Front Table,” looked accurate if a person was standing behind the projector and looking forward. This was the only mode that worked correctly with the PapARt library, specifically the “Light” example. This correctly translated the x and y axis of the colored circles when an AR marker was being moved. Other modes of the projector, such as “Front Ceiling,” reverse translated the movement of these circles so when the AR marker was being moved to the left, the circle on the table moved right.

Calibration for the projector and camera was essential. We decided to make use of what professional equipment the Multimedia Program offered would work better for securing our project structure. We acquired four C-Stand gobo arms and three gobo heads[42]. The projector’s tripod socket was necessary to mount it onto the head of a Benbo ball tripod so that it was positioned directly above the table. (fig. 17)

The calibration process on the computer began after setting up the physical structure for the hardware. Since PapARt was already programmed to handle the projector-camera calibration we ran a few Processing sketches to assist with the calibration process. The first sketch, called “PCConfiguration,” allowed us to test and configure the hardware correctly. Our first test was to see if the projection was being displayed correctly along with the right resolution and size dimensions. The second test was to configure the web camera in order to make sure that the camera would turn on and start capturing.

Another PapARt example sketch that we examined very closely was called “Lapin.” This projected a 3D model of a rabbit when an AR marker was present. (fig. 18) This was the marker to projection functionality we wanted to incorporate into our application. We were able to successfully

---

[42] A setup that assists in holding lighting instruments

integrate the two Processing libraries PapARt and Unfolding so that when an AR maker was present our program would display geographical features superimposed over a map of Oakland. Integrating meant we could create map-based data visualizations in SAR.

Each of the many examples the PapARt library contained gave us a sense of how real-time human interactions with physical pieces of paper made AR objects move in three-dimensional space. The Processing IDE first needed to detect the marker positions for the virtual objects to move to the correct location on the table. We found out that the response and accuracy of the virtual objects were highly dependant on the speed of the AR marker movement. If the marker moved too fast then the virtual object would not move.

After exploring the various interactions we could use for the project we decided to try getting different data visualizations to appear after specific AR markers were placed on a tabletop. Placing multiple markers would mean overlaying multiple geospatial visualizations on the map. To make this work, we explored the PapARt API thoroughly to get a better understanding of the in-built classes and functions[43].

5.3 Datasets

Visible Disparity runs off of a specific set of pre-researched datasets that had to be imported into the project code after being searched for online. In other cases, we had to take data, convert it, and shape the files ourselves. All datasets were fact-checked via the US Census Bureau website, which we discovered formed the backbone of most the sources of the datasets we found.

[43] Java templates used to create and define object, object data types, as well as methods Functions in Java programming refer to code that executes particular tasks

From the BART.gov website we successfully downloaded and plotted our first dataset to be seen and tested. Compared to our other datasets, these are fixed points that do not usually change over time and so they became, as discovered through user testing, information to compare to the other datasets. We were able to find GIS files that mapped all the school points in Alameda County via the ACGOV website. We found when compared to our other datasets that these locations were spread out evenly across all of Oakland and like the BART data does not usually change over time.

Housing data turned out to be the easiest to map as it is inherently related to geographical locations. The Zillow real estate website, Hotpads, contained enough information on low and high-end apartments to make a visual comparison. By using our work-around method via Google Maps and GeoData Converter, we were able to identify, download, and plot the address points for each living space. These two datasets were interesting enough together and even more so when compared in various combinations to the others.

Our workaround method made us realize that we could search Google Maps for almost anything, grab the addresses, download them, convert them, and plot them. This was the same method used for when we collected liquor store locations to see how it might compare with the other datasets.

All datasets worked once we brought them into the project code. Our method of copying addresses and putting them through Google Maps to create our own GeoJSON files was efficient enough for us to render several more. So, as long as we had an address or GPS coordinates, creating map based files proved to be no challenge.

5.4 User Testing

The three user testing sessions we held helped us identify unforeseen problems with the project and improve upon them. During the second round of testing sessions, we presented two project setups.

One was a rectangular table and the other was a round table. We did not mention to the users that this dual setup was intentional. During the testing period, users were only interested in the projected map and visualizations. In the feedback discussion after testing, we asked them which setup they preferred. All users preferred the round table. This confirmed our theory that the round table creates a more inviting environment. From that moment on we used the round table setup.

During the same feedback session we invited users to ask any questions they had about the project. We discovered users were more interested to learn about the technology and how a SAR space actually works. We understood that this might be in a part due to the fact that some of our users were incoming Multimedia graduate students a year behind us and so were curious to learn more about the technology for their own edification. However, if this project was to be presented in a gallery or classroom space, we would not necessarily have control over who interacts with it. It was at this point that we decided to create another setup for the projector and the camera with the objective of hiding the devices and the cables. This way, users' attention would remain focused on the project content.

As previously mentioned, during the last testing session we noted some users held the AR marker cards up in the air instead of placing them on the table. This was something we had not anticipated. According to users' feedback, this was due apparently to the lightness of the markers. We ended up reprinted the targets and pasting them onto laser cut plastic to give them a little more weight.

When making design decisions over the project color scheme, we soon realized that we had to change our limited color palette. Prior to the third user testing session, all colors were reassigned new hues: brown, red, purple, green, cyan and dark blue. This was to help overlapping data points stand out on the map.

In the first two minutes of the third user testing session, we noticed both groups thought that they were playing a game at first. A user in the first group even asked whether this was the case before picking up the cards and dealing one AR marker to each user. This was another interesting and unanticipated reaction. From this session we posited that our colorful set of AR marker cards had inadvertently linked users to memories of card playing games. Colors had initially been chosen merely to distinguish them from one another when we programmed and tested them. This interaction proved the color choice to be an unexpected success because it made the experience more enjoyable for users.

Another significant piece of feedback we received concerned our Oakland map. Unfolding Maps works with different map providers or Web Map Services. Each WMS has a different look and feeling with various color schemes and level of details. The map provider we used at the time of testing was EsriProvider.WorldGrayCanvas(). We selected this provider because of its monotone gray color scheme that helped bring users attention to the colorful datasets. This map did not include the names of the cities. During the testing sessions, a couple of users explained that they did not know which city they were looking at or that it took long to figure that out. Since we were limited in the number of map providers available through Unfolding Maps, and the Esri maps suited the project interface well, we decided to add the name of the city, “Oakland” in the lower left area of the map. This worked well as it helped clarify which location the users were looking at as soon as the experience started.

5.5 Final Project/Prototype

The final project and prototype of Visible Disparity consisted of a round table, an opaque lampshade housing the projector and webcam, and an Apple Mac Mini computer. The computer is turned on, the Processing sketch is loaded, and users can begin interacting with the trackable markers. The lamp, housing the tech, was hung from a ceiling and the computer controlling the project was left to the side. The cards were vibrant and recognizable as interactive objects and the datasets began to tell different stories when compared to each other in various ways. The final project also succeeded in encouraging conversations through SAR.

6. Discussion

6.1 Use of Spatial Augmented Reality to create Collaborative Spaces

When researching use cases of SAR technology we found out that its potentials had not yet been explored in the realm of data visualizations. Data visualizations are significantly used as a way to increase awareness amongst people about different subject matter. We found this as a good opportunity to explore how we can integrate interactive data visualizations into a SAR space. The main goal of the project was to bring people into a collaborative space where they can discuss ideas around data visualizations.

Our hypothesis stated that SAR could be the technological solution in creating such spaces. We proposed that discussing data with others could improve understanding and can encourage learning. So, we aimed to remove mobile phones and computers screens from the experience of viewing and interacting with data to encourage face to face conversations and interactions with others. Could this technology be a solution to the issue of digitally isolating experiences? What happens when people interact with tangible objects, like paper, to make changes to a digital image? How can this new technology setup affect the well-being of their experience?

Over the course of a semester, we found answers to all of these questions. Our SAR project was tested several times by different groups of users. Since the very first testing session, we could see that the project had been successful in bringing users together. This was in part due to using a round table. The use of a round table was a small decision that we made in the development stage of the project but proved to have a great impact on the project outcome. Testing confirmed that users felt more welcome to contribute to the experience in comparison to when they had to take sides by standing at a rectangular table.

Another successful factor was the use of vibrant colors and color coding of AR marker cards which at first was intended to allow data points to stand out when overlapped. In testing sessions, we noticed that this feature also reminded users of card games where each player is assigned a different color. This added to the gamification aspect of the project and made the experience of viewing data fun and more engaging. This could especially be of great importance when thinking about future use cases of this technology. For instance, in an elementary classroom, a teacher would be able to import relevant datasets about the subject being taught into the program, run the program and show visualizations that are generated by the application where students would then be gathered around the table and prompted to view and find patterns in the data. To measure how such technology could improve students' learning, additional user testing with students of different age groups and with relevant datasets would be required. However, our research and testing results manifest the great potential of the application.

During one of our user testing sessions, we noticed that some conversations amongst the users were not about the data but just the project technology itself. This was due to SAR being uncommon amongst them. Such conversations were interesting to hear as they led us to rethink how the technology setup should look like and how we might improve it.

After prototyping with various materials and different arrangement of devices, we assembled our fourth and final prototype. What made this prototype stand out compared to the previous ones was that it made the projector and camera mostly invisible to users. This new setup helped our users focus their attention on the data presented in front of them rather than having them raise questions about how the hardware structure was made.

6.2 Development Learning Outcomes

Creating a Minimum Viable Product

One of the biggest lessons learned in the process of creating this project was understanding what makes for a successful Minimum Viable Product (MVP)[44]. MVP refers to creating just enough features to satisfy an early customer base. For Visible Disparity, this meant creating a space where users would simply activate/deactivate different data visualizations where patterns that datasets could be visualized would prompt discussions amongst users. Our journey towards this MVP had its ups and downs. Here, we would like to go through how we tailored our project to a manageable scope and deliver something that was not only accepted but well received.

The project started with the interest to help users understand and discuss the housing crisis in the Bay Area in a SAR setting. Initially, data visualizations were planned to be just one segment of the project. In the Spring of 2018, we proposed that the project would involve other content such as interviews with individuals struggling with high rents in the Bay Area and with professionals in economics to get their perspective.

However, a few months into the project development, we found ourselves overwhelmed by the project scope. Researching such complex issues along with understanding how a SAR space would be created seemed like a bigger project than what our team of three could accomplish in fifteen weeks. We began to think about other use cases that such a project could have so we decided to shift our focus on the technology and its potential for the project.

[44] Read more about MVP here: https://www.interaction-design.org/literature/article/minimum-viable-product-mvp-and-design-balancing-risk-to-gain-reward

This is when we started to think about the bigger opportunities that this technology could create if it were to be used as a tool for educational purposes. Rather than abandoning our content, we simply shifted our interest to SAR while also making the project more manageable by narrowing down the research parameters. We continued our research on housing around the Bay Area but narrowed it down to the City of Oakland. It seemed to be the ideal location since across the Bay, the highest rents in the country can be found in San Francisco and yet Oakland comes in at sixth place in the exact same report.[45] This data on high cost housing visualized side by side with low cost housing would be something that could be informative in SAR yet we would have needed relevant datasets collected for this.

We decided to concentrate solely on creating data visualizations from existing datasets. This made sense for the project as it helped us use resources created by experts while demonstrating the possibilities of SAR technology. Tailoring the project was not limited to the content. In order the finish the project, we also needed to remove some features from our interface such as the function to rotate and translate the projected map.

While in the development stage, we observed that the features we were coding such as zooming and panning would have made our project more interactive but seemed to slow down the progress of which we needed to finish the most basic functionality of the code. Similarly, we also noticed that every time we wanted to add or remove certain features, the user experience design had to be reviewed and altered when needed. This took more time to plan, code, and design. However, we learned that such revisions were valuable and made us aware of the time constraints that we had with the project. The revisions that we made to the final code were essential to achieve the best results.

[45] "IDEO's Human Centered Design Process." UserTesting Blog. May 09, 2019. Accessed May 10, 2019. https://www.usertesting.com/blog/how-ideo-uses-customer-insights-to-design-innovative-products-users-love/.

Main Takeaways

Working with the two complex Processing libraries, Unfolding and PapART, gave us the skills to understand the Java language a little more in depth. We had to learn the most basic Java syntax so that the integration between libraries was possible. Because everything in Java is associated with classes and objects, understanding how to use the attributes and methods was important for the functionality of our application. We also learned that contacting the creators for assistance with the code that they had documented was also helpful. Both the creators of Unfolding and PapARt were very helpful in pointing us in the right direction with how to use their libraries and other technical issues that we were having.

Over the course of a semester, we experienced how to use multiple tools and techniques for rapid prototyping. The use of rapid prototyping allowed us to make use of many different configurations of the hardware and how the different devices would fit within the structure. The many iterations that we made with our hardware and structure were essential in figuring out the exact measurements for placement of the camera and projector from the surface of the table.

Our struggles with configuring the camera and projector to find the right calibration ended when we decided to upgrade the webcam. During our first prototype, we were using a webcam that was capable of 1080p. This webcam, along with a 1080p DLP projector was accurate in detecting AR markers. However, when we exchanged the DLP projector with an ultra short throw projector for portability in our second prototype, the AR markers were not being detected.

We experimented with various heights and placements of the camera around the projection space as well as different lighting conditions. After going through these various configurations, we decided to try a higher resolution webcam. We purchased an Ultra HD 4k webcam that was capable of High Dynamic Range (HDR) which captured high resolution images in many lighting environments. When testing this webcam with our code for detecting AR markers, we instantly noticed that these markers now had a high detection rate under medium to high lighting. We learned that when testing out a technology, it is best to experiment with various devices of different specifications and be prepared for hardware incompatibilities.

6.3 Future Improvements

As mentioned earlier, the project's technical development took longer than planned. Within the restricted time frame, we completed the project with the datasets we had on hand. This was comprised of geospatial data on Oakland including rent, BART stations, school locations, neighborhoods, etc. If we were to continue this project, we would have spent some additional time to research relevant datasets such as median household income. This could involve interviewing educators from different fields of studies such as social justice or economics, to ask what type of data they would find relevant to be displayed in the project.

This project could be used to assist an instructor in creating data visualizations that are needed for collaborative project based learning. The instructor could take our project to the classroom and test it with the students. This could be important to measure the learning outcome of using SAR with the help of the instructor and observe the level of engagement among the students.

We could also spend more time improving the technology and including touch based interactions in response to the users hand movements and possibly adding more interactions with AR markers to create more possibilities in which users can investigate data. Panning through the map, zooming into the details, hovering over each point/polygon marker to view more information and rotating the map are some interactions that were removed in the development process early on. Given a longer time to work on this project, our team may have continued to work on these features.

Creating our own customized AR markers was another challenge that we were not able to overcome. The PapARt example that our project is based on uses AR markers from the ARToolkitplus library which are black and white fiducial[46] markers. (fig. 19) In terms of functionality, these AR markers have a high detection rate and are easy to use because they are readily available for downloading online. In order to replace the fiducial AR markers with customized ones, we would have needed to modify the base code which would have made the AR detection function in the code stop working as it was written for ARToolkitplus. However, it is ideal to replace the fiducial AR markers with highly textured images that represent a meaningful connection to the data that is being displayed. For instance an image of houses that represent expensive or affordable housing.

[46] A fiducial is an easily detected feature that serves as a point of reference in proximity to an object targeted for tracking

7. Dissemination

Visible Disparity was promoted via our Instagram page, *visible_disparity*, which was created at the beginning of the project. It has helped us to keep track of our accomplishments as well as generate some interest with social media followers. Our research website, visibledisparity.com, served two purposes. During the process of working on the project, we updated the website weekly to summarize our progress and the feedback we received from our graduate committee. After project completion, we updated the website with a new page that gives an overview of Visible Disparity in the form of a user experience case study.

Center for Student Research Poster Symposium, April 17, 2019

The team received a scholarship from the Center of Student Research (CSR) at California State University, East Bay. The requirements for becoming a scholar meant submitting written responses to questions asked by the CSR through a Google form and attending monthly CSR workshops, lectures, and events. One of these events was the CSR poster symposium, held on April 17th, 2019. During this event we presented an academic poster(fig. 20) explaining our research question, development, and current results. It was also an opportunity to let students and professors in different departments see what the CSUEB Multimedia department was working on. Our poster was viewed by others who might not have been familiar with the concepts and terms used in the Art Department and our research. We had to provide context for our work and explain clearly the importance of Visible Disparity. We also had to display and write about our interpretations of findings.

Spring Presentation, May 16th, 2019

Visible Disparity will be exhibited at the 2019 Multimedia Spring Presentation. In this presentation we will present a video trailer of our project, and then showcase the final prototype at the University Art Gallery on Wednesday, May 15th and Thursday, May 16th.

Maker Faire Bay Area, May 17-19, 2019

After the Spring 2019 presentation, we will have the opportunity to present our work at the Bay Area Maker Faire in San Mateo, CA. The Maker Faire is a great place to showcase creative technology and get into contact with other creative makers. This will be a great opportunity to see our work with a broader audience and collect their feedback.

8. Conclusion

Project Visible Disparity uses both data visualizations and SAR to bring people to collaborate in a real-world space. It takes geographical data in code form and outputs it as visualizations. Looking at data in this setting was expected to be clearer for users' understanding, especially if they are given the power to interact with data collaboratively in a group setting.

SAR encourages this freedom of interaction using AR markers, which further occupy the same collaborative space as tools for interaction. Rather than using a singular computer or touch screen, projecting into a physical space allows for data to be displayed on any flat surface. The setup was intended to free users from the isolating experiences of screens. The affordance of a projector allows users to setup and display their own data visualizations anywhere in a shared environment where collaboration with others is possible.

Implementing a round table creates a more inviting environment where users can feel that they are participating equally with others present near them even when they are unacquainted with. Placing the hardware that is used for the SAR technology out of view allows for users to focus their attention on the data visualizations and discuss viewpoints between the different datasets instead of the technology that makes it happen.

Guiding users with instructions on how to interact with the SAR space allowed for a simple way to introduce users on how to first interact with the interface. Using different colored AR markers created a playful way to visualize data and easily differentiate the many data points that could be seen on a map. This also made it easier for users to discuss correlations between data and find new ways to collaborate to see different visualizations.

The goal for Visible Disparity is to provide a template that anyone wanting to visualize data in a SAR environment can implement. Using affordable hardware and open source software, we propose that this setup can introduce new ways to interact with data that allows for a more open collaborative space without the distraction and interference of digital screens. The creation of this SAR setup could create a better educational environment where everyone would be able to see the same information being presented.

Understanding how the structure is to be built is important for this SAR environment to work. We presented how the physical hardware is to be used and configured in order for the successful calibration of the projector and camera. We explained what type of media is to be presented through this SAR medium and what types of datasets are to be easily used to present the visualizations. Through the research and development of our application in Processing, we proposed how to integrate the PapARt and Unfolding Maps in order to create geographical markers to be visualized as shapes on a map.

We also presented the types of interactions that would be incorporated into the SAR data visualizations. The possibilities of adding more interactions such as panning, zooming and hovering over certain point/polygon markers to view more data are viable with the better understanding of the Java language. Learning the API of both libraries would allow for introducing many more features that could enhance the data visualizations. Creating SAR for interacting with data visualizations creates for a more collaborative and engaging space for all.

9. Recommendations

9.1 Creating SAR Spaces

As of today, there are two primary ways to utilize SAR spaces. The first is by taking advantage of existing commercial products such as the Lightform interactive projection system and the Pico Light tangible augmented reality system. This provides the advantage of having a source that has already done most of the programming and configuration as well as the research. The second option is to use open-source programs/libraries such as PapARt and Paper Programs. The latter option allows more flexibility in creating custom programs and is more cost friendly. Therefore, we recommend that designers and developers take advantage of them especially if they want to get a hands-on sense of how SAR works. This means either building on top of the existing libraries or simply using the programs for their own creations. Despite the limitations and insufficient documentation in some cases, the effort is worth it. After all, it is by testing out such programs that the creators get the chance to improve their products. Moreover, AR is likely to become an inseparable part of the future and as designers and/or developers by creating programs that do not require pricey headsets or setups we can make this technology accessible for a larger group of people.

9.2 A Tool for Collaborative Data Visualizations

Our research and user testing results point towards SAR having great potential for creating collaborative spaces. We believe that this technology could be used in educational settings including classrooms or workshops or any other environment where project-based learning is a key factor. A school could use the set up to provide a new interface for their students to collaborate around, rather than having their focus being isolated towards one screen. This would be especially useful in a classroom setting where the teacher gets to change the data based on the subject matter while still using the same technology to engage students.

With that being said, we encourage that designers think of Visible Disparity as only one possible way of using such technology and start to transform this application into a tool where different datasets can be imported and presented in a user friendly setting. We envision a future where users can upload their own datasets to the program and then data visualizations are created based on their input. The setup of Visible Disparity is specialized for group interaction. It can be utilized for groups to import, analyze, display, and interact with data. This setup is applicable for teams that have to work together to analyze information such as geographical data and require the interactive ability to manipulate the data in a collaborative setting. A group of people can engage in subject matter directly based on what data is fed into the application.

Bibliography

"5 Common Myths About Virtual Reality - And Why They Aren't True." Techopedia. Accessed May 10, 2019. https://www.techopedia.com/.

"A Geo Game Based on Open Data: Learn Neighborhoods of Cities... and so Much More." Click That 'hood. Accessed May 10, 2019. http://click-that-hood.com/.

"Bay Area Home Prices by Transit Stop." Estately. Accessed May 10, 2019. https://www.estately.com/bay-area-home-affordability-transit-stop.

"Build Software Better, Together." GitHub. Accessed May 10, 2019. https://github.com/.

"Dynamicland." Dynamicland. Accessed March 08, 2019. https://dynamicland.org/.

"Federal Communications Commission." FCC Area API. Accessed May 11, 2019. https://geo.fcc.gov/api/census/#!/area/get_area.

"Geographic Information System (GIS)." Geographic Information System (GIS) - Data & Statistics (CA Dept of Education). Accessed May 10, 2019. https://www.cde.ca.gov/ds/gi/.

"Geographic Information System (GIS)." Geographic Information System (GIS) - Data & Statistics (CA Dept of Education). November 4, 2018. Accessed April 10, 2019. https://www.cde.ca.gov/ds/gi/.

"Lexicon of Geospatial Terminology." Geospatial Metadata - Federal Geographic Data Committee. 2008. Accessed March 08, 2019. https://www.fgdc.gov/policyandplanning/a-16/lexicon-of-geospatial-terminology#G.

"Methodology." The Lean Startup | Methodology. Accessed May 10, 2019. http://theleanstartup.com/principles.

"Methodology." The Lean Startup | Methodology. Accessed May 11, 2019. http://theleanstartup.com/principles.

"Search Results for SDK, Format Sustainability." The Library of Congress. Accessed May 10, 2019. https://www.loc.gov/search/?q=SDK&fa=subject:format sustainability.

"Sustainability of Digital Formats: Planning for Library of Congress Collections." CSV, Comma Separated Values (RFC 4180). July 27, 2017. Accessed April 10, 2019. https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml.

"The Tech Terms Computer Dictionary." The Tech Terms Computer Dictionary. 2019. Accessed April 10, 2019. https://techterms.com/.

Anti-Eviction Mapping Project. "Anti-Eviction Mapping Project." Anti-Eviction Mapping Project. Accessed March 09, 2019. https://www.antievictionmap.com/.

Ekene Ijeoma.. "Wage Islands: Immigrants Is a Sculpture Which Submerges a Topographic Map of NYC Underwater to Visualize Where Low-wage Immigrant Workers Can Afford to Rent." Ekene Ijeoma. https://studioijeoma.com/#. Accessed March 09, 2019

Foundation, Processing. "Processing.org." Processing.org. Accessed May 10, 2019. https://processing.org/.

Fry, Ben. "Processing 2.x and 3.x Forum." Processing 2.0 Forum. Accessed May 10, 2019. https://forum.processing.org/two/discussion/8071/why-do-i-get-a-nullpointerexception.

JavaScript Functions. Accessed May 10, 2019. https://www.w3schools.com/js/js_functions.asp.

Johnson, Sydney. "The Art of Data-telling." East Bay Express. March 01, 2019. Accessed March 09, 2019. https://www.eastbayexpress.com/oakland/the-art-of-data-telling/Content?oid=4747483.

Julia Chatain. SyMAPse: Augmented Interactive Maps for Subjective Expression.

Human-Computer Interaction [cs.HC]. 2015. Ffhal-01191429v2 Accessed March 08, 2019

Nwobi, Kachy, Eliza Wallace, and PolicyMap Team. "PolicyMap." PolicyMap. April 10, 2019. Accessed May 10, 2019. https://www.policymap.com/.

Pensacola News Journal. "Data Central Home." Pensacola News Journal. Accessed May 10, 2019. https://data.pnj.com/.

RealityTech - Spatial Augmented Reality. Accessed May 10, 2019. http://rea.lity.tech/en/nectar.html.

Sudhakaran, Posted BySareesh. "The Anatomy of a Light Stand (Part Two)." Wolfcrow. Accessed May 10, 2019. https://wolfcrow.com/the-anatomy-of-a-light-stand-part-two/.

US Census Bureau. "Census.gov." Census.gov. Accessed May 10, 2019. https://www.census.gov/.

US Department of Commerce, and Noaa. "What We Do." What We Do | NGS INFO - About Us | National Geodetic Survey. June 5, 2015. Accessed May 10, 2019. https://www.ngs.noaa.gov/INFO/WhatWeDo.shtml.

World Leaders in Research-Based User Experience. "UX Training, Consulting, & Research." Nielsen Norman Group. 2019. Accessed March 08, 2019. https://www.nngroup.com/.

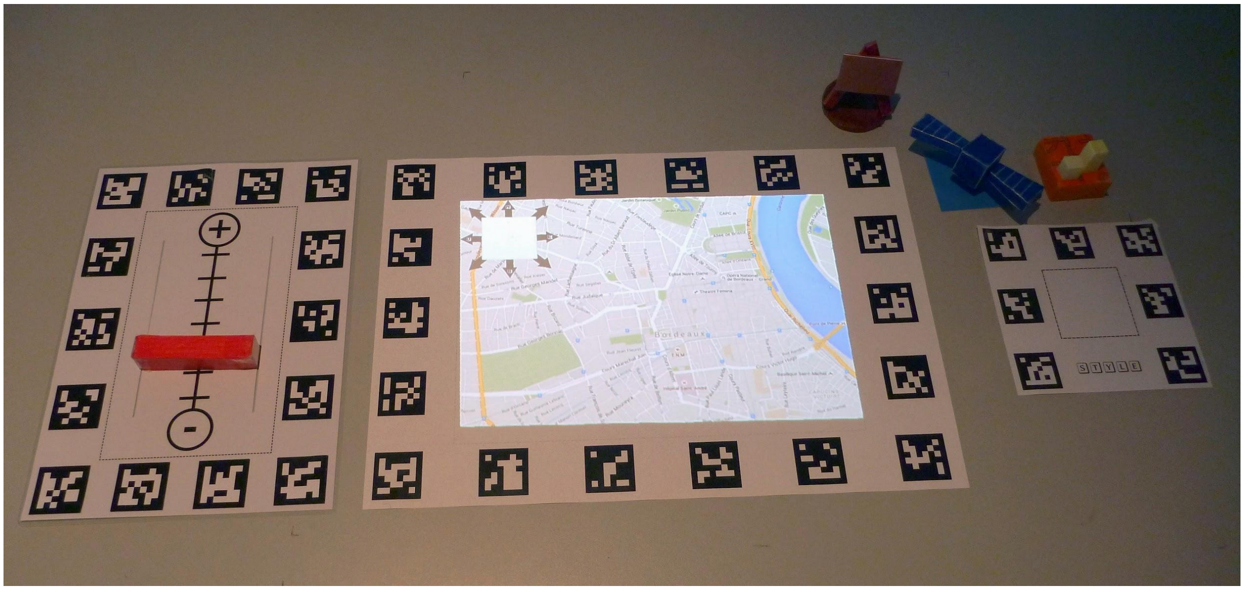
"OmniTouch: Wearable Multitouch Interaction Everywhere." Chris Harrison | Rethinking the Progress Bar, © Chris Harrison, www.chrisharrison.net/index.php/Research/OmniTouch.\

10. Appendix

Figure 1

Dynamicland, *Untitled*. 2016, mixed media, Oakland, California. Source: Kosminsky, Eli, Oakland, 2018, Digital Image. Available from: https://concord.org/blog/dynamicland-a-new-direction-for-immersive-simulations/ (accessed March 8, 2019)

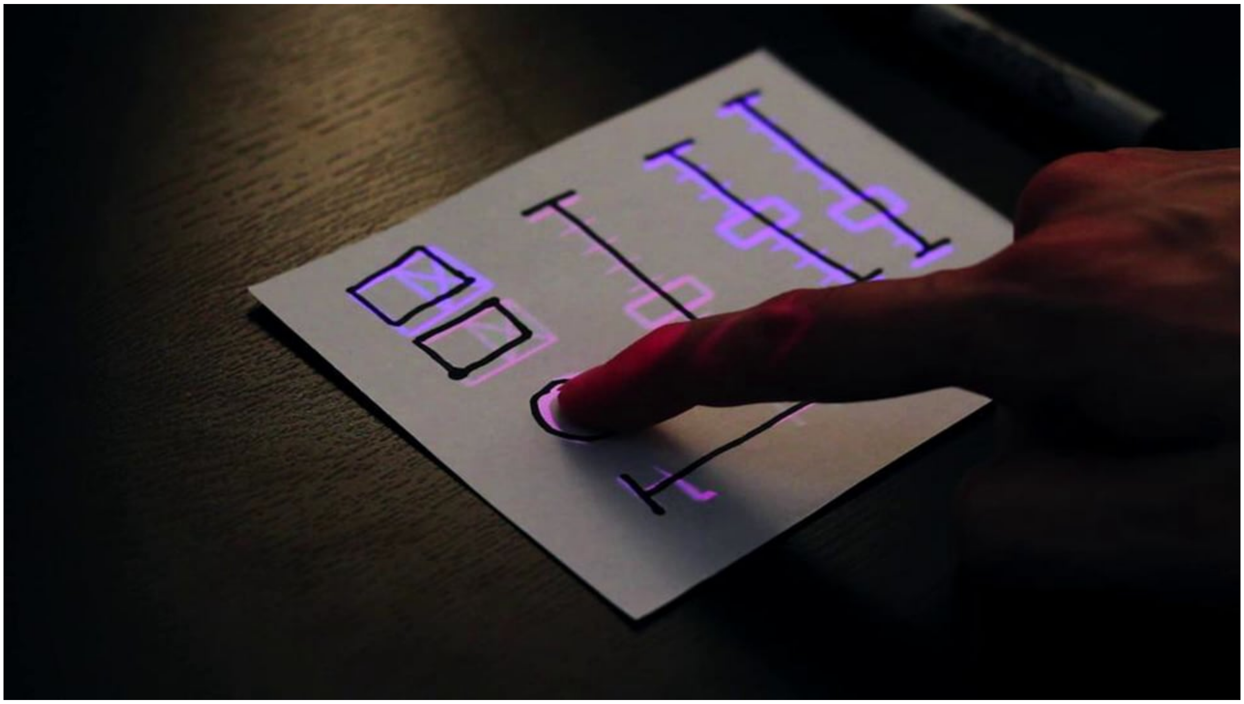
Figure 2



SyMAPse, *Untitled*. 2018, mixed media. Source: Chantain, Julia, Digital Image. Available from:

http://people.bordeaux.inria.fr/jchatain/ (accessed March 8, 2019)

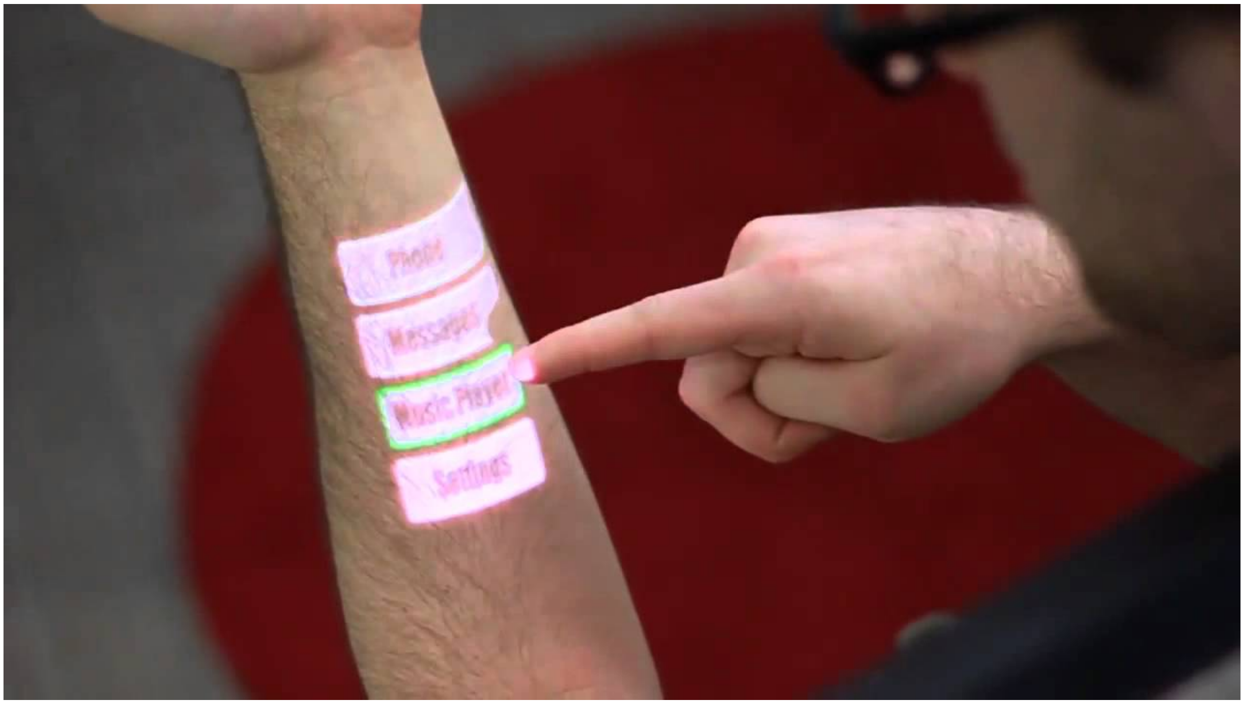
Figure 3



Keyes, Billy, *SketchSynth*. 2012, Venmo video, 1:48. Available from: https://vimeo.com/42053193

(accessed March 8, 2019)

Figure 4



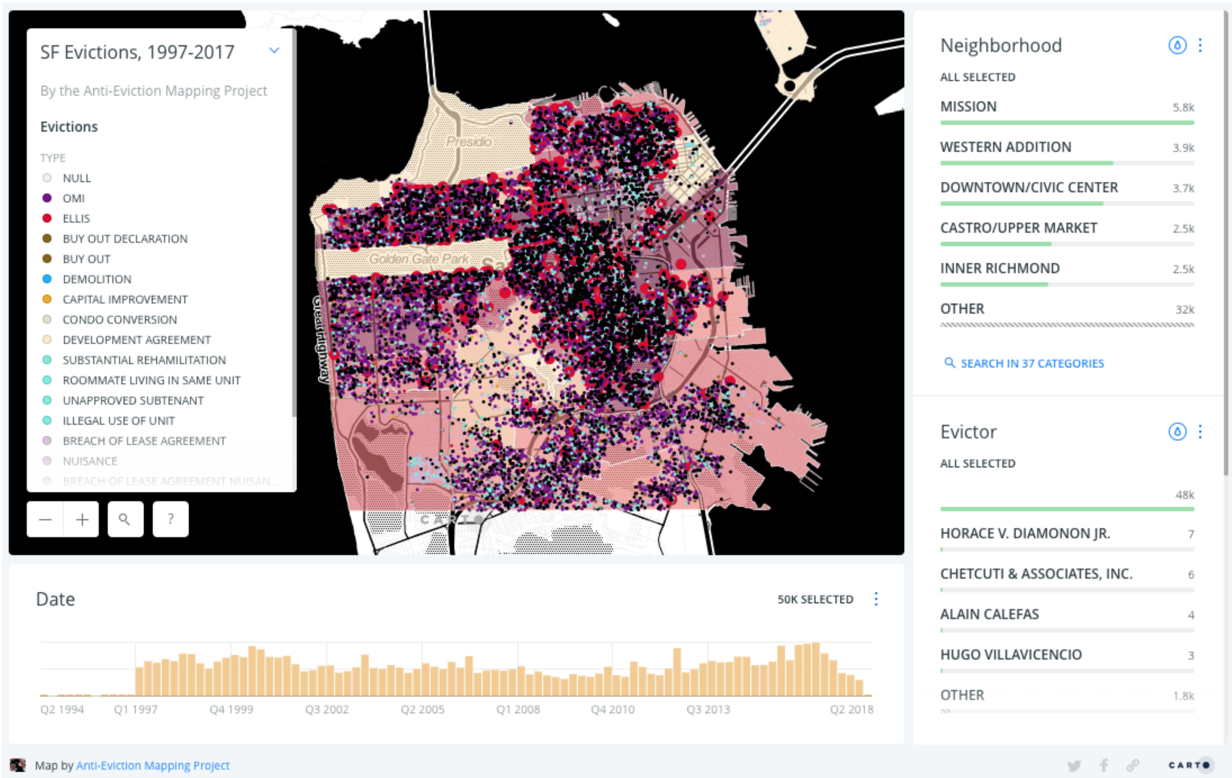
Harrison, Chris. *OmniTouch - Demo Video - ACM UIST 2011*, 2011, Youtube video, 1:48. Available from:

https://vimeo.com/42053193 (accessed March 8, 2019)

Figure 5

Ijeoma, Ekene, *Wage Islands*. 2017, mixed media. studioijeoma.com. https://studioijeoma.com/Wage-Islands (accessed March 8, 2019)

Figure 6



Anti-Eviction Mapping Project, *All San Francisco Eviction Notices, 1997-2018*. antievictionmap.com.

https://www.antievictionmap.com/#/sf-evictions/ (accessed March 8, 2019)

Figure 7

Nectar Platform, *RealityTech - Spatial Augmented Reality.* http://rea.lity.tech. http://rea.lity.tech/en/natar.html (accessed May 10 2019)

Figure 8

Visible Disparity, *ASUS P3B Projector*, May 2019.

Figure 9

Visible Disparity, *Logitech Brio*, May 2019.

Figure 10

Visible Disparity, *Lamp Structure,* March 2019.

Figure 11

bart_stations | pointStyle.java

```java
import processing.core.PGraphics;
import de.fhpotsdam.unfolding.geo.Location;
import de.fhpotsdam.unfolding.marker.SimplePointMarker;

public class pointStyle extends SimplePointMarker {

  public pointStyle(Location location) {
    super(location);
  }

  public void draw(PGraphics pg, float x, float y) {
    pg.pushStyle();
    pg.noStroke();
    pg.fill(245, 98, 30, 100);
    pg.ellipse(x, y, 20, 20);
    pg.popStyle();
  }
}
```

Visible Disparity, *Java class for customizing point markers in Unfolding,* May 2019.

Figure 12

bart_stations | pointStyle.java

```
public void setup() {
  size(800, 600, P2D);
  noStroke();

  smooth();
  map = new UnfoldingMap(this); // you can specify the map provider ir
  MapUtils.createDefaultEventDispatcher(this, map);
  map.zoomAndPanTo(oaklandLocation, 12); //sets default to Oakland by
  float maxPanningDistance = 1; //defines maximum panning distance val
  map.setPanningRestriction(oaklandLocation, maxPanningDistance);//res
  map.setZoomRange(12, 15);//restircts zooming

  features = GeoJSONReader.loadData(this, "BART_Stations.geojson");
  MarkerFactory markerFactory = new MarkerFactory();
  markerFactory.setPointClass(pointStyle.class);
  markers = markerFactory.createMarkers(features);
  map.addMarkers(markers);
}
```

Visible Disparity, *Referring to the custom Java class for point markers in void setup(),* May 2019.

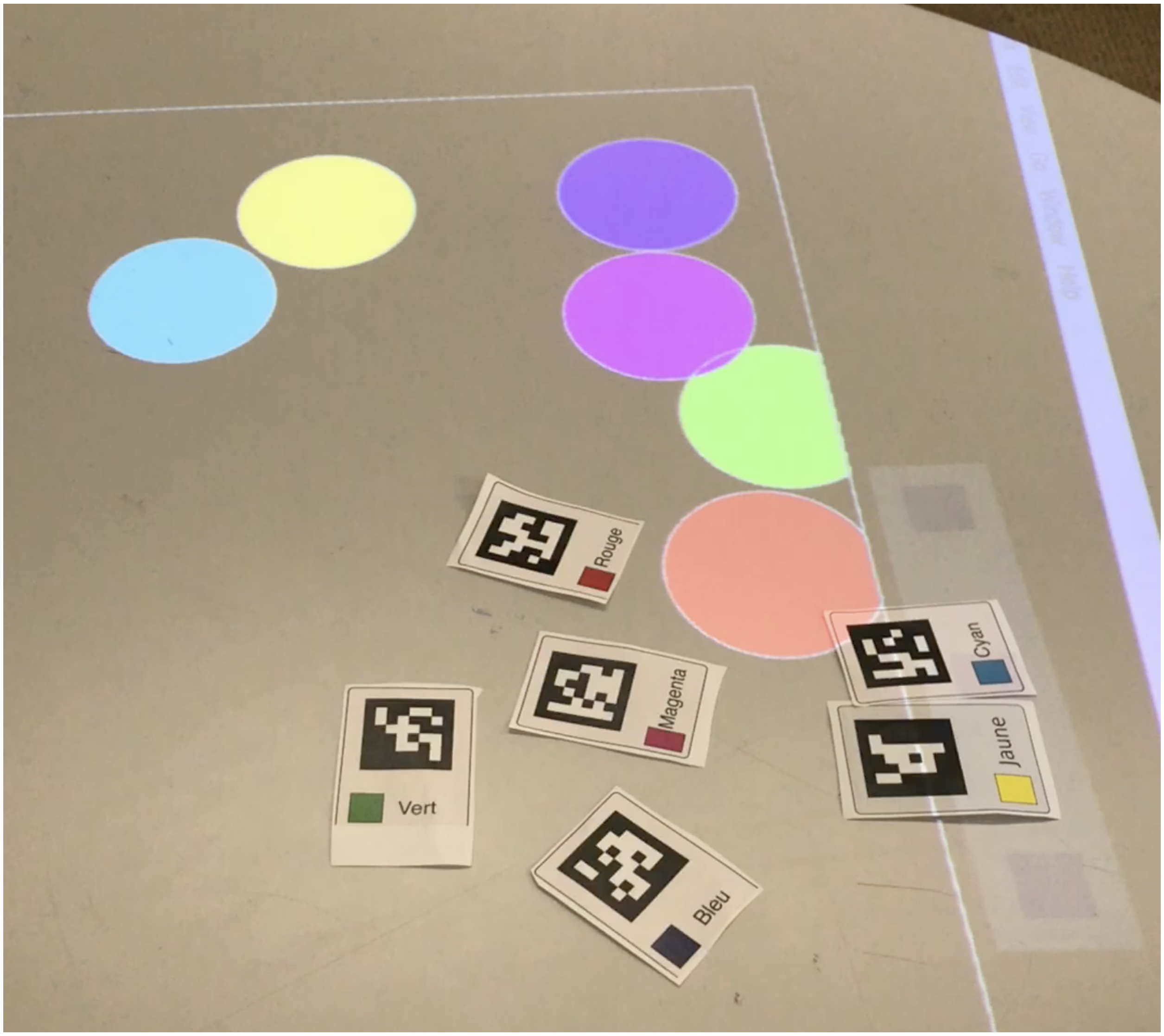
Figure 13

```
void draw() {
  background(0);
  map.draw();
  //Location location = map.getLocation(mouseX, mouseY);
  //fill(0);
  //text(location.getLat() + ", " + location.getLon(), mouseX, mouseY);
  println(frameRate);
  changeColor();
}

void changeColor() {
  for (Marker marker : map.getMarkers()) {
    marker.setColor(color(0, 60, 100, 100));
    marker.setStrokeColor(color(255, 255, 255));
    marker.setHighlightColor(color(0, 100, 0, 200));
    marker.setHighlightStrokeColor(color(255, 255, 255));
    marker.setStrokeWeight(1);
  }
}
```

Visible Disparity, *Creating and calling a custom function to change polygon markers colors,* May 2019.

Figure 14



Visible Disparity, *Light Sketch,* May 2019.

Figure 15

```
public void keyPressed() {
  if (key == 'o') {
    showOak = true;
    clear = false;
    //showOak = !showOak;
    println("showOak  is " + showOak);
  }
  if (key == 'b') {
    showBart = true;
    clear = false;
    //showBart = !showBart;
    println("showBart  is " + showBart);
  }
  if (key == 's') {
    showSchool = true;
    clear = false;
    //showBart = !showBart;
    println("showSchool  is " + showBart);
  }
  if (key == 'c') {
    clear = true;
    showOak = false;
    showBart = false;
    println("clear is " + clear);
  }
}
```

Visible Disparity, *KeyPressed conditional statements that made visualizations appear/disappear,* May 2019.

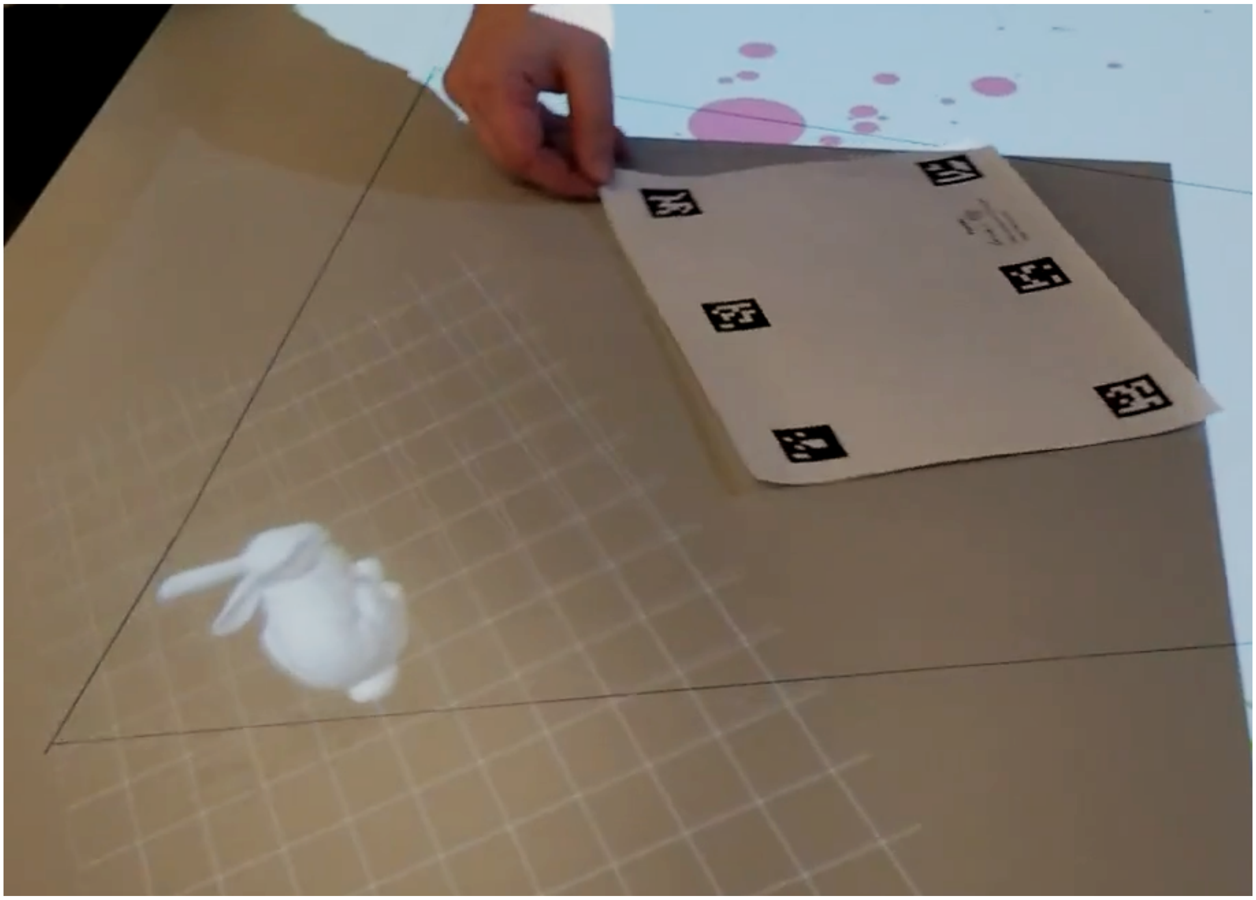
Figure 16

Visible Disparity, *Project Instruction,* May 2019.

Fig 17

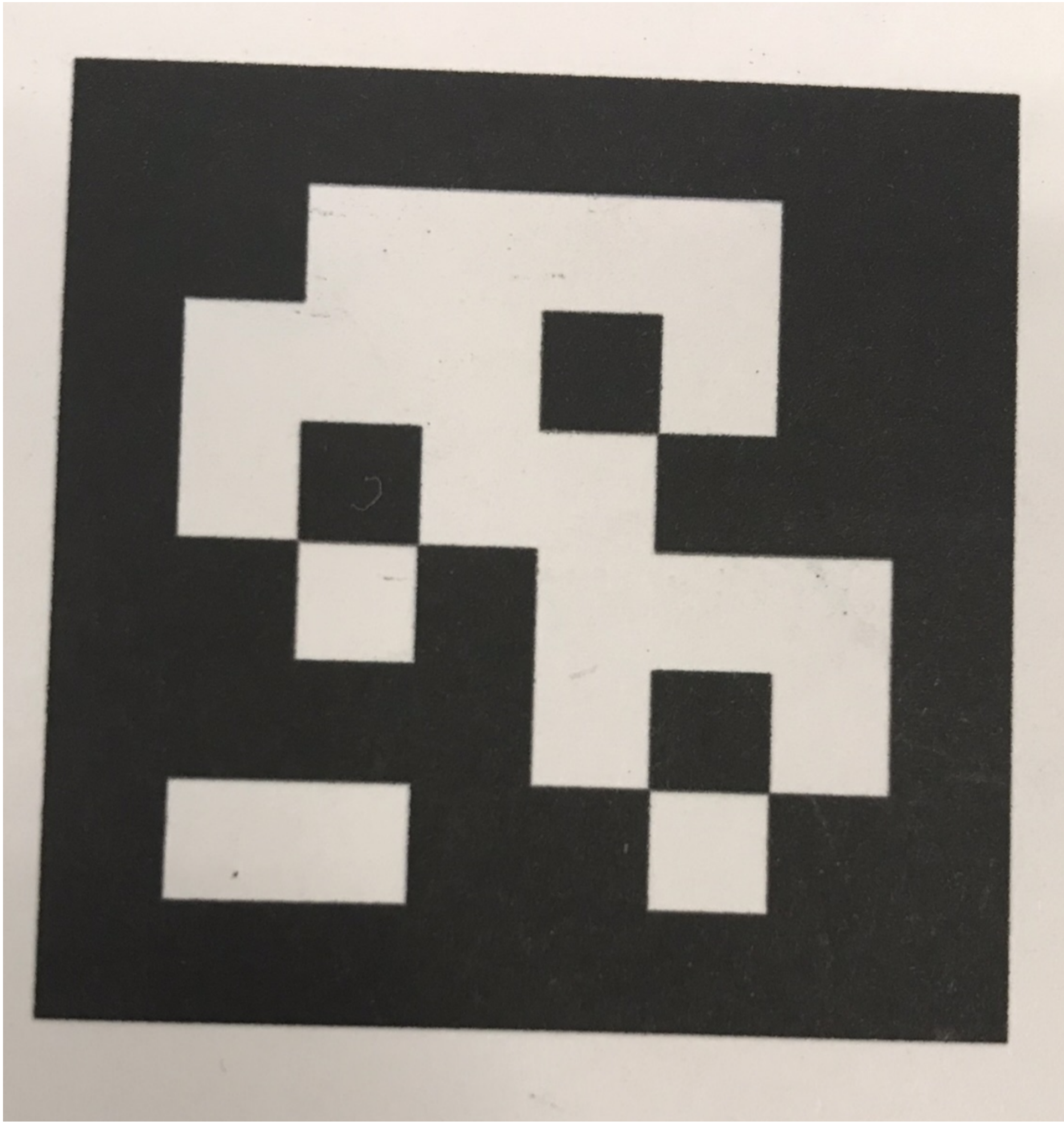
Visible Disparity, *Benbo Tripod Head,* May 2019.

Figure 18



Visible Disparity, *PapARt Lapin Example,* April 2019.

Figure 19



Visible Disparity, *ARToolkit Fiducial Marker,* May 2019.

Figure 20

Visible Disparity, *CSR Poster Symposium,* May 2019.