Project Aurora: An Examination Of How Notifications Should Be Handled In A Virtual 3D Environment

KARTHIK PALUSA* and MORGAN DAVIS*, Colorado State University, USA

CCS Concepts: • **Virtual Reality**; • **Virtual Environments**; • **User Notifications**; • **Human Computer Interaction**;

ACM Reference Format:
Karthik Palusa and Morgan Davis. 2022. Project Aurora: An Examination Of How Notifications Should Be Handled In A Virtual 3D Environment. 1, 1 (May 2022), 12 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Notifications are a crucial part of any system's user experience. As defined by Lauren Goode, in her piece for Wired, "Notifications are, at the most basic level, a method of alerting people to some piece of information, often with some element of urgency" Goode [2]. In the early days of computers, notifications came in the form of icon flags and the occasional audible alerts. Usage and how often you were alerted was limited to how often you strictly interacted with the computer, with no simple solution to notify users when away from their personal computer.

However, in 2003, Blackberry introduced email notifications to your phone. This ushered in an era of true instant communication, as now even when you were physically away from your computer, you could still receive email notifications. Innovations addressing notifications would go stale until 2009 when Apple introduced its Apple Push Notification service (APNs) with iOS 3.0. APNs worked through maintaining a continuous connection to the phone and Apple's notification servers. When a 3rd party service wanted to send a notification, they would send the data to Apple's servers, which then would relay that information back to the user. Google would follow by releasing its own similar service known as Google Cloud to Device Messaging (C2DM). In the 12 years since the introduction of push notifications, both leading mobile platforms (iOS and Android) have implemented and improved how push notifications are handled, from how and when they're communicated to also how they're displayed.

Nonetheless, there was still room for improvement. Interactive notifications came in 2014, which enabled another layer of interaction such as quick replying to a message or email. This additional layer added another element of interaction in the UI, as now more operations to handle a notification were added, such as swiping right for quick actions and left for notification dismissal.

Push notifications are also used in macOS and Windows, however, both systems follow less intrusive notification centers; which display collections of notifications in a hidden side panel. While mobile platforms focus on touch interfaces to manage notifications, both PC OSs use a mouse and keyboard as the main form of user interaction.

*Both authors contributed equally to this research.

Authors' address: Karthik Palusa, kpalusa@rams.colostate.edu; Morgan Davis, medavis1@rams.colostate.edu, Colorado State University , Fort Collins, Colorado, USA, 80521.

XXXX-XXXX/2022/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

While touchscreens enabled a 1:1 relationship with a gesture mapped to an action, PCs require another layer of interaction due to the gestures provided by the mouse, alongside the physical clicks from the mouse as well. While it does add additional complexity, PC notifications can offer more actions due to the more precise control of gestures and interactions.

With the introduction of the Meta Quest 2 (FKA Oculus Quest 2) in 2020, we saw the first mass-market, affordable, commercial VR headset. Following the release of the Quest 2 headset, we saw Facebook (Meta) pivot its business towards virtual reality and the “Metaverse”. The metaverse is a virtual 3D world that is focused on social interactions and connections. While gaming was the first wave ushering us into the metaverse, Meta has promises of a platform with many use cases. With this new wave of software interaction, notifications are once again back into their infancy. There currently isn’t a tried and tested way to handle notifications in a virtual environment.

While Meta hasn’t reached the platform size of either Apple’s iOS or Alphabet’s Android, a proper notification system is still required for the platform to reach max adoption. Currently, Oculus’s management of notifications is practically nonexistent, as most applications don’t have a proper suite to manage push notifications. Our goal with Project Aurora is to create a productivity-focused dashboard that handles email and calendar notifications, while also letting users customize how various notifications are presented and handled. Through the usage of an application developed for the Meta Quest 2, Project Aurora’s research will be focused on how to achieve the best user experience regarding notification handling within a 3D virtual environment.

2 RELATED WORK

Push notifications have had close to 20 years of innovation followed by research. Previous research involved understanding how to deliver notifications from the real world into a virtual one Rzayev et al. [5]. Users would have to exit the virtual environment (e.g taking off the headset) to interact with their phone or computer after receiving a notification. In a study conducted by Rufat Rzayev, Sven Mayer, Christian Krauter, and Niels Henze, they investigated how presenting digital notifications compared in three different virtual environments with three different tasks and four different placements of the notifications Rzayev et al. [5]. The study found that different types of notifications should be displayed in different placements. For example, urgent notifications should be displayed using a “head-up display” placement because this placement is the most noticeable Rzayev et al. [5].

Another study done in 2018 for the CHI Conference on Human Factors in Computing Systems, proposed that VR applications should have notifications that are displayed in a way that fits the virtual environment experienced by the user Zenner et al. [9]. This means having notifications that show up for example by messenger on a horse if the environment has a medieval setting Zenner et al. [9]. The goal behind this experiment is to provide a future framework for developers to be able to deliver notifications in a more immersive and less intrusive manner Zenner et al. [9].

Another study looked at delivering both physical and/or virtual notifications to inform users of head-mounted displays what their surroundings were in the real world Ghosh et al. [1]. They analyzed using multi-modal notifications for common interruptions users would face when immersed in a virtual environment Ghosh et al. [1]. In conclusion of the study, they found that visual notifications alone had poor noticeability and that was possibly “due to the highly visual nature of VR” Ghosh et al. [1]. When they incorporated haptics they were able to get better responses, but haptics without clear distention were difficult to understand Ghosh et al. [1]. These studies provide a great amount of information to take into consideration when designing how we would like to deliver notification in a virtual environment.

3 DIGITAL NOTIFICATIONS IN VR STUDY

To complete this project, we developed a program that displays notifications in a 3D virtual environment. The users will then be prompted with notifications from different areas of the environment; the upper right-hand corner, lower right-hand corner, upper left-hand corner, and the bottom left-hand corner. To make sure all inputs are equally favored after each corner notification is handled, the user would have to reset their view back to the center. The actions in the corners and center are button inputs, which would require the user pays attention to the content while also giving us an idea of how users interact with the controller versus the mouse. We are using the center button as a way to signal the "core content" that a user may be working on or viewing. Once the user is focused on the center, similar to how we interact with notification content on our screens, the user then would be shown a notification in another corner.

The amount of time it takes for a user to interact with each quadrant's notifications from the press of the center button will be tracked. This data will be collected 3 times and will be averaged to represent the data set. The other part of our experiment will stem from a simpler program on the computer. To create an understanding of how users interact with their computers, we built a simple web page that handles a test very similar to our VR experiment, however this time we used a mouse to track the interaction. Results were taken similarly, as we used the computer data set as a control variable, relative to our VR experiment, due to users being more comfortable with a mouse.

The data collected, once analyzed, will allow us to determine what is the best way to deliver notifications in a virtual reality environment.

All project work and deliverables will be divided evenly amongst both members. We will manage progress, tasks, and deliverables using Notion, to ensure all information is easily accessible.

3.1 Notification Design

Project Aurora will be an application designed for the Oculus Quest 2 and Quest built using the Unity game engine. Our Minimum Viable Product (MVP) will be a minimal productivity dashboard that has a configurable UI. The core reason the UI will be configurable is to make it easier for us to place notifications in various areas to test the most optimal notification placement.

Our measured metric will be the response time of the user interacting with the notification within the various notification locations. Using the two images below as a mockup, we can see how the UI can vary depending on the notification's banner location.

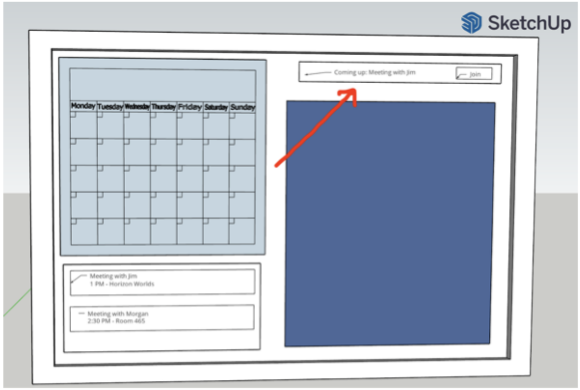


Image A: Upper right-hand notification

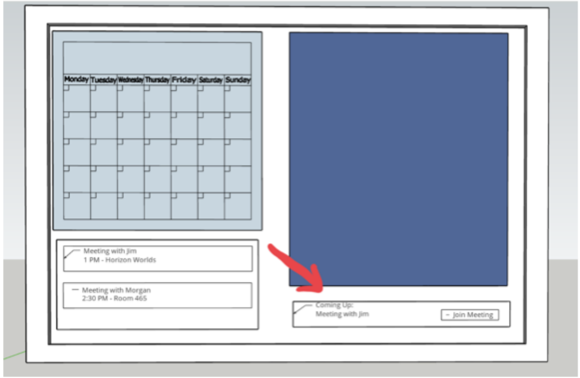


Image B: Lower right-hand notification

4 EXPERIMENT DESIGN

In the planning stages of Project Aurora, our first question was how do we want to handle information and data. While notifications as a component share a similar design, the content and information density also were hypothesized to play an important role Rzayev et al. [4], Seinfeld et al. [6], Woodward et al. [8].

While the locations of where notification content and actions would be located would remain constant, we wanted to ensure that the data we gathered would be able to represent a range, from single line to multiline notifications. Ultimately, we decided to streamline the notification component to focus on the interaction and response time.

In the design of our component, we kept the relative rectangular shape found in most operating systems as the modular notification component, however, due to the fact that interactions of a notification differ from platform to platform, we made the entire component a touch target. Making the entire object interactable ensured that we could ensure that both macOS and Windows users would have a fair chance on their interaction times.

Once we got the notification component designed, we had to focus on how we wanted to present the experiment environment. For our environment, we wanted to create a virtual environment that would be similar to a clean desktop. While VR enables users to interact with their environment in more ways than just a simple 2D interface, content like notifications are unique as they stay in the same location(s), no matter the content you are working with. We used this model as a background and created a minimal environment that was open for movement but also limited in what you could. As we merged these ideas, we then focused on understanding how notification alerts can be presented in a respectful way, complimenting the experience, not deteriorating it Imamov et al. [3], Wiehr et al. [7]. Ultimately, we created an environment that is similar to a classroom whiteboard or smartboard. As originally envisioned in our proposal, we saw VR as a platform rather than an application, thus we centralized the notifications to just the scaling of the canvas board. While this would allow us to tailor a more experiment-focused environment, it would also give users a sense of familiarity.

5 EXPERIMENT RESULTS

This experiment was done using a within-subject design. There were 10 participants that completed a set of three trials in two different experiments. Before either the experiments were conducted each participant was asked what their OS preference was as well as where they would prefer notifications to be presented in VR. Majority of participants preferred MacOS and all participants preferred notifications to display in the top right of their field of view in VR.

The first experiment was a web based experiment based off of the VR experiment. The objective of this experiment was to collect reaction times to notifications displayed in the web browser. The task was not to try and bet a specific time, but to see what placement of a notification they responded to the quickest. A participant would first click the start in the center of the web-page (Figure 1), once they have done that a timer would start and a notification would appear in either the corner of the web-page (Figure 2).

Fig. 1. The start screen of the web based experiment.

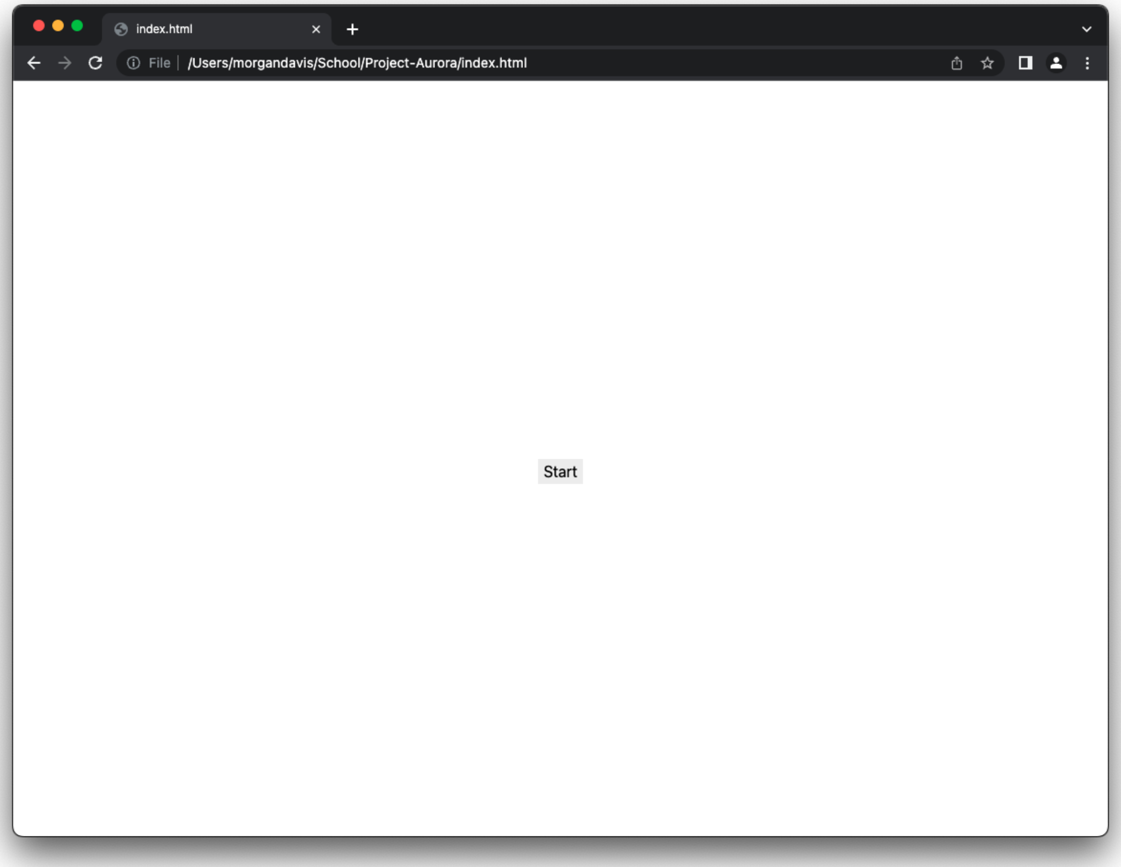
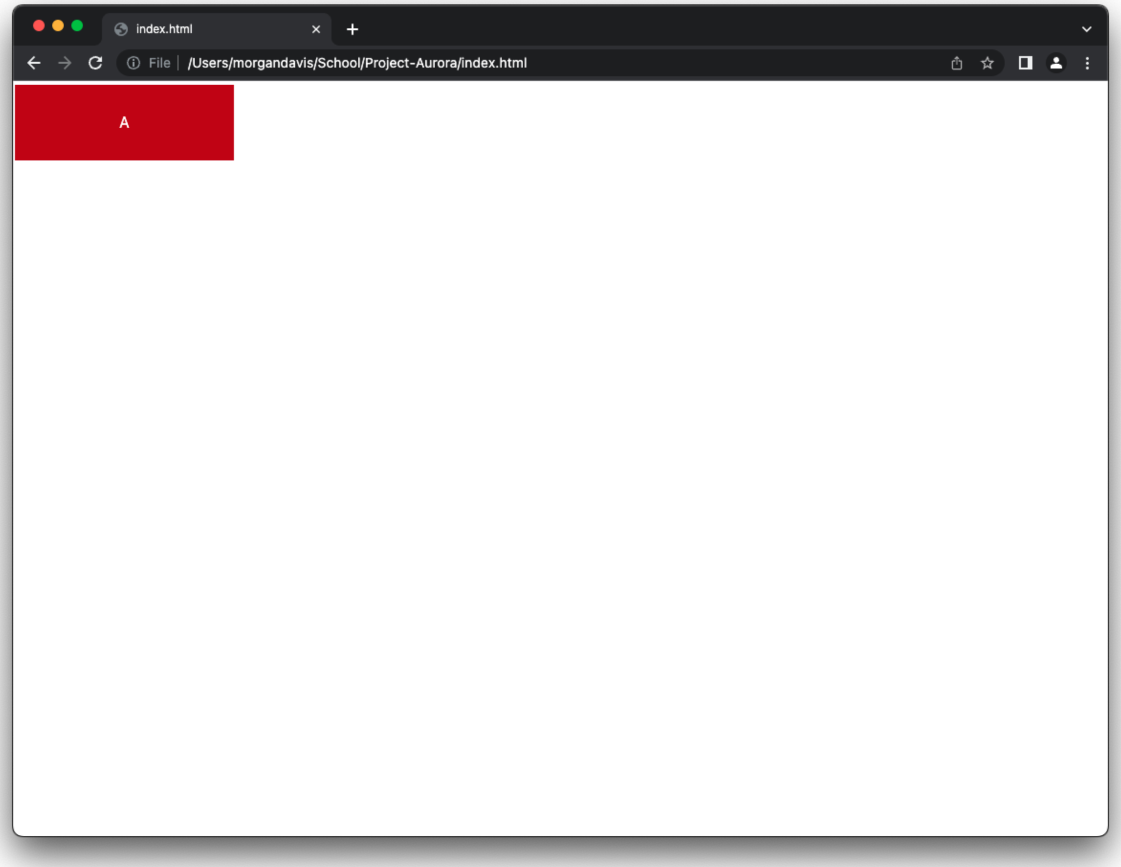
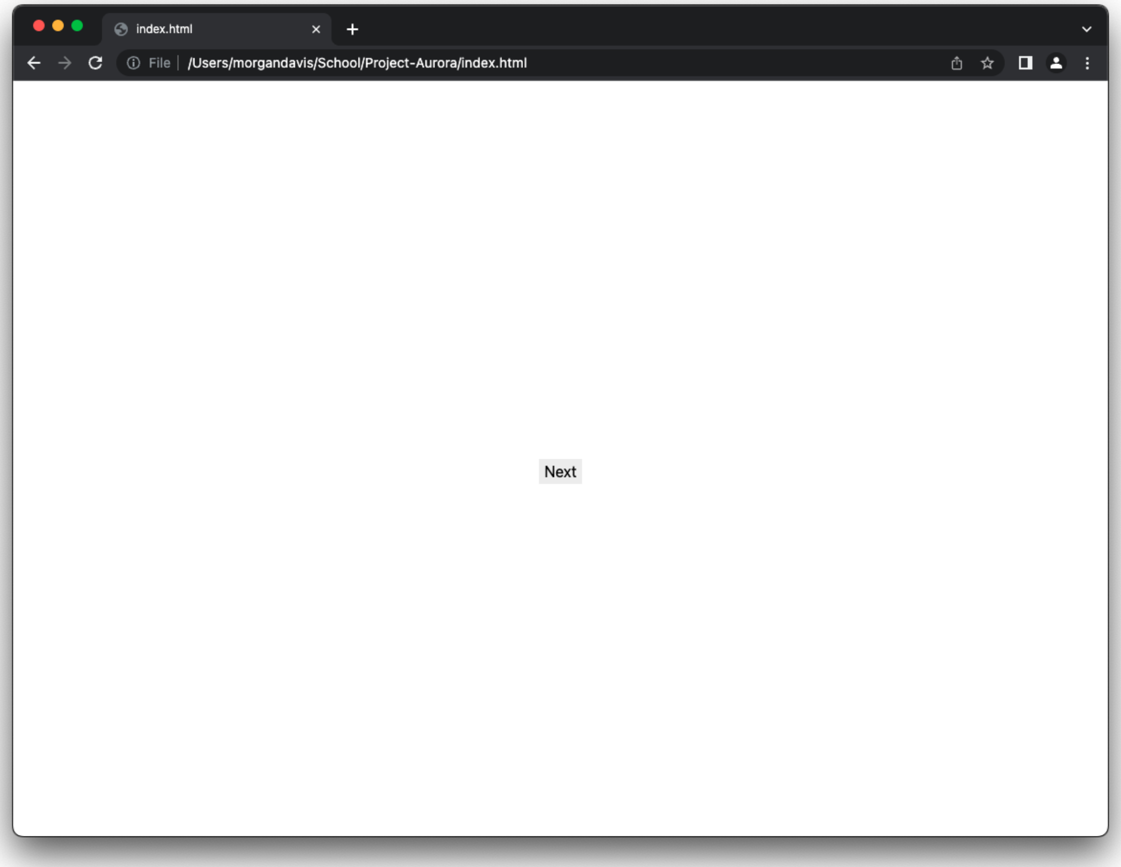


Fig. 2. An example of a notification that would appear after the start button is clicked



Once the notification was dismissed a button would reappear in the center of the web-page and say next (Figure 3). When the participant clicked next another timer would start and another notification would appear. There were four notifications that would appear in the same pattern each trial.

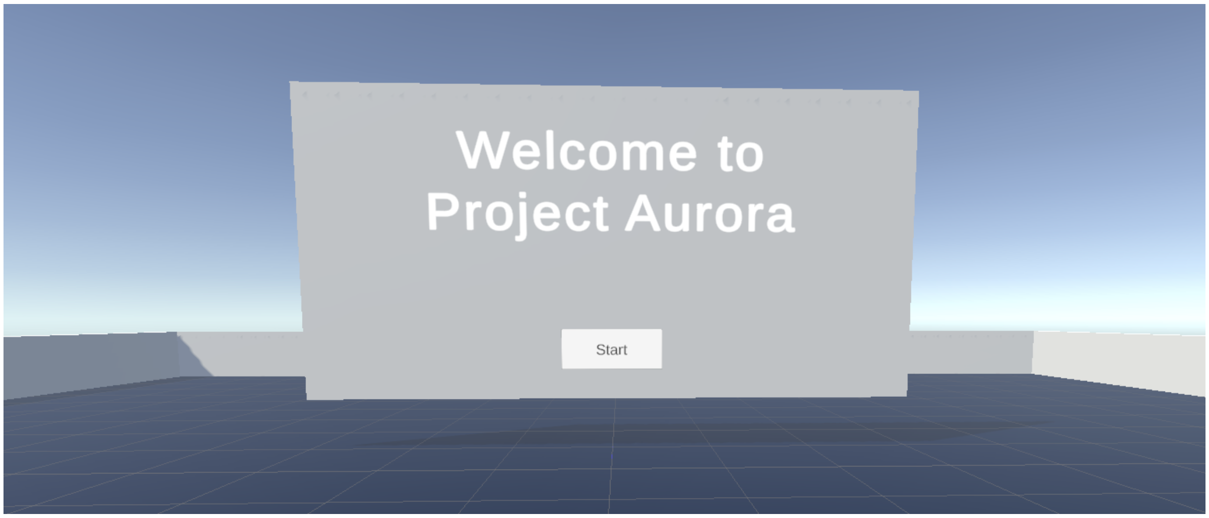
Fig. 3. The next button the user would click to push the next notification.



Once a participant completed a trial their times would be collected and then they would test again. After three trials the participants times for the various notification placements were then averaged to determine what placement they reacted to the quickest.

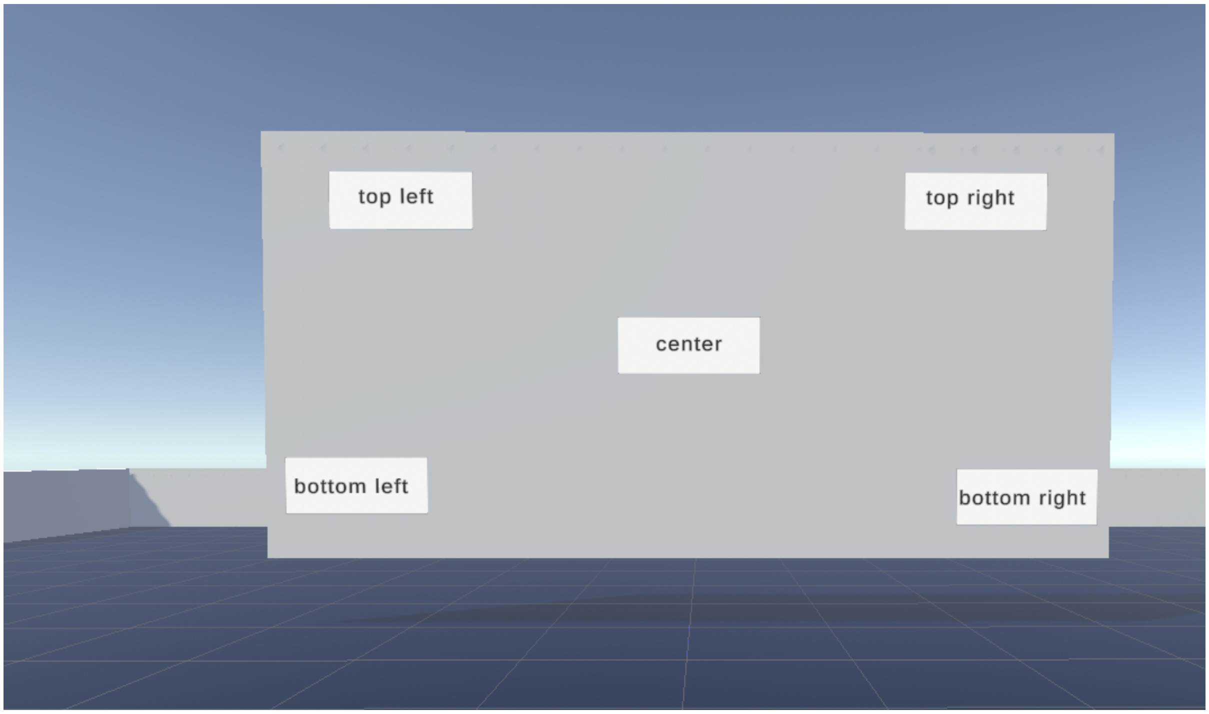
The second experiment was a VR based experiment. Using the Oculus Quest 2 participants would immerse themselves into our virtual world to complete a similar task to the web based experiment with the main difference being that users would use the controllers to dismiss the notifications instead of a mouse. Another key difference was that the notifications appeared on a canvas board in the virtual world in order to keep a distraction free environment for the users to focus on the task at hand.

Fig. 4. The start canvas in the virtual world.



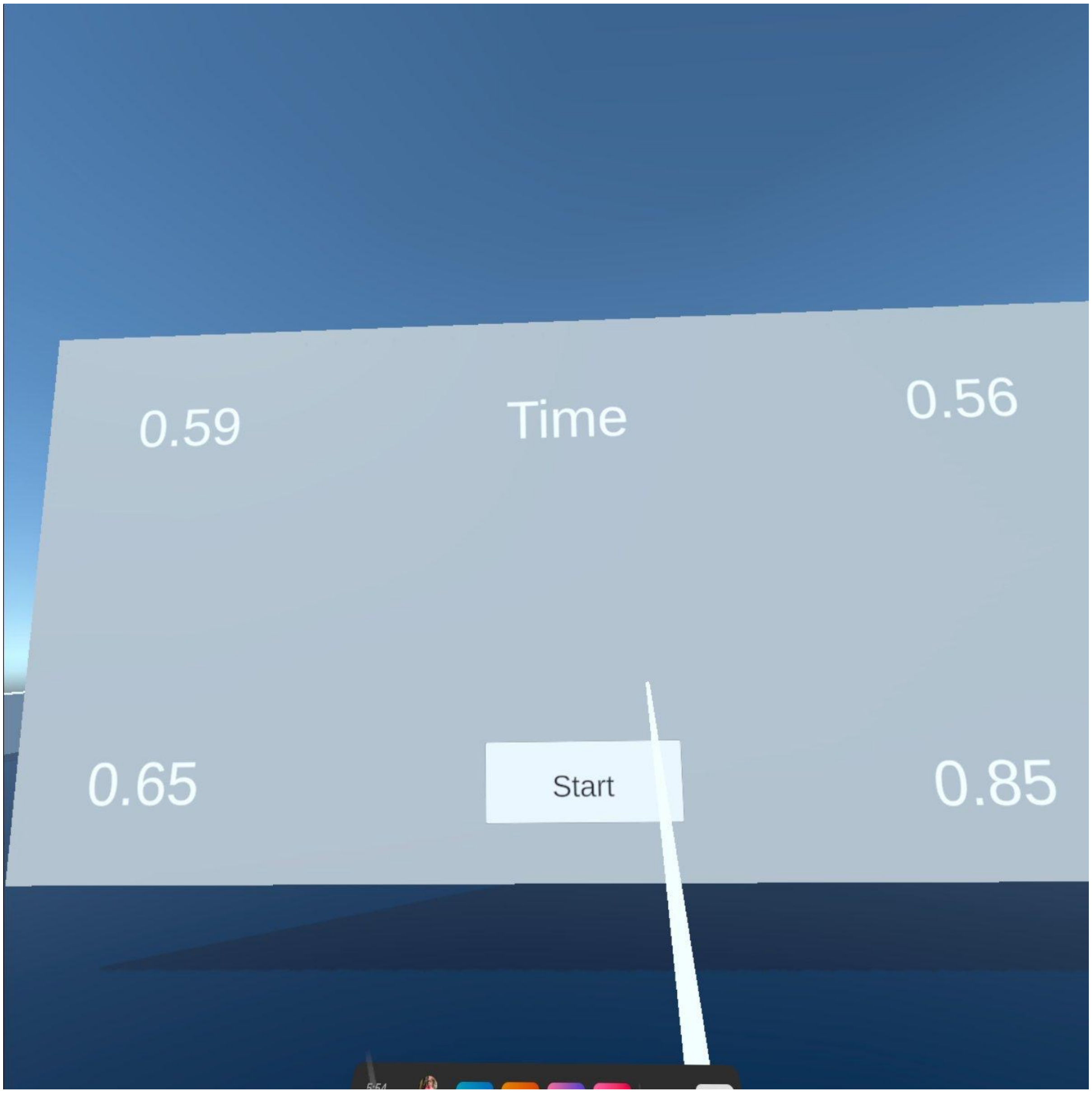
A participant would click the start button and the timer would begin once the notification appeared. Once they interacted with the notification a button labeled center would activate the next notification.

Fig. 5. Placement of each notification on the canvas (All notifications showing for demonstration purposes).



After all four notifications are dismissed the canvas would then display times in each corner (Figure 6).

Fig. 6. The result times displayed after a trial



0.59

Time

0.56

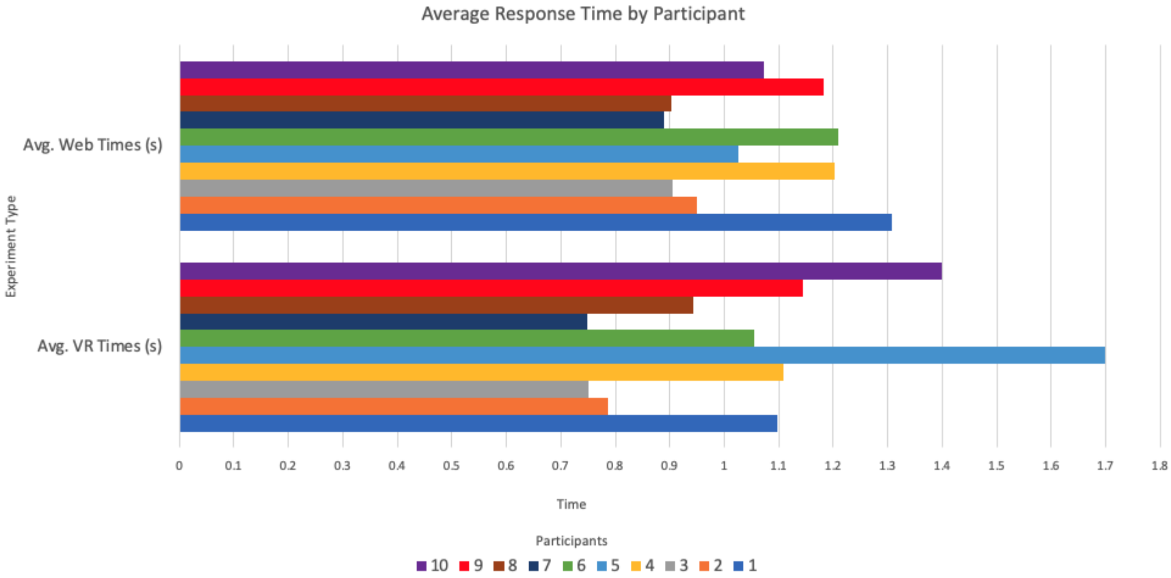
0.65

Start

0.85

These reaction times were then averaged after the three trials to determine the participants quickest reaction to each notification placement.

Fig. 7. A bar graph displaying the average response times to all four notifications for each participant

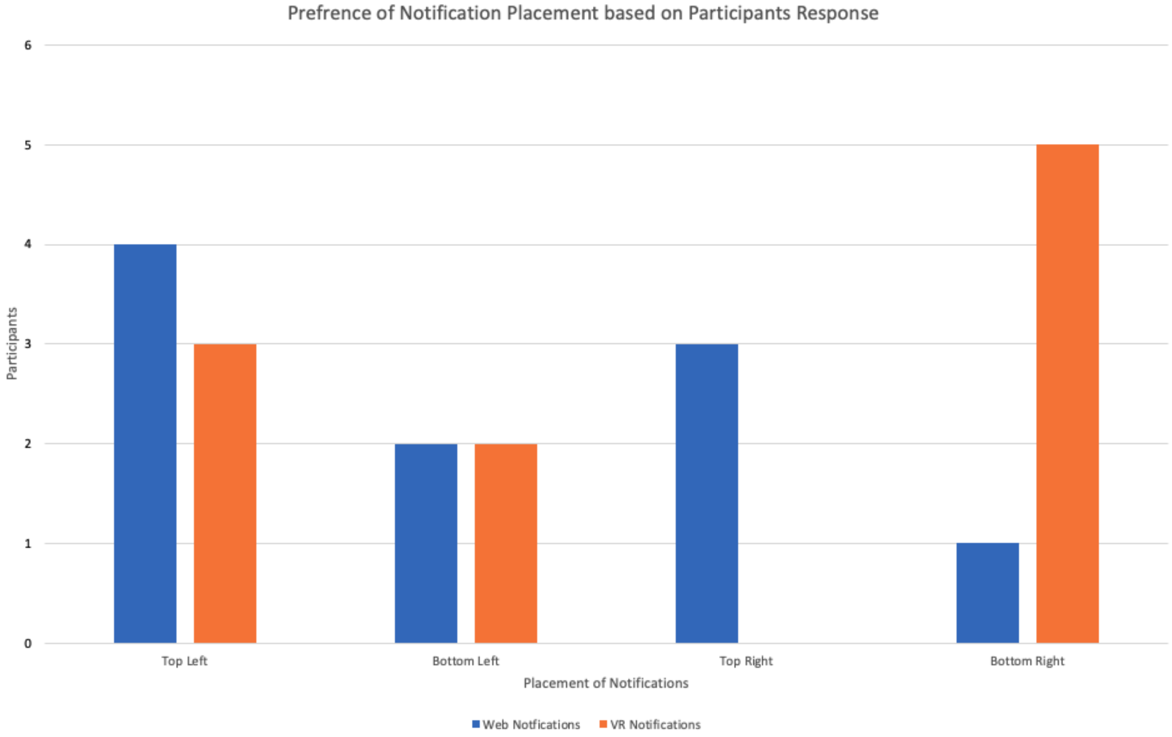


6 CONCLUSION

Through the experimentation completed, Project Aurora was able to build a base for understanding the relationships between the preference in desktop OS and how the elements from that platform transfer to another platform. While at first, we had hypothesized that there would be a correlation between what operating system users used, and how that would be translated similarly in VR, our findings showed another story.

With our trials, our biggest takeaway was that a user's preference does not translate to it being the most efficient route (Figure 7). While all users in our study had stated that they preferred notifications in the top-right corner, almost all users did not have the best result with that corner.

Fig. 8. Bar graph showing Notification preference based on each participants quickest response time to the placement of each notification



The outlier data point in our data set is the only value where the user's prior OS preference (Windows), translated to them responding the quickest to the bottom-right corner, which is where Windows deploys notifications. However, this data point had participant how he prefers the top right corner handling notifications. Throughout the time trials, we saw the same data points get faster, but maintain similar deltas been values.

While our hypothesis did not come out in our favor, we conclude that while prior operating system usage does not translate core user interface details from platform to platform, we do think that preference is a strong element in how user interaction should be handled. This is because our data showcased how users were quick to adapt to changes, just within the trials they completed, and how even though the preferred route was not the fastest, users shared they felt their preference for the top-right corner was because it was the most comfortable and natural to use. We hope that with this data, future VR, AR, and XR developers can take our data and continue researching more about how information should be represented in a new environment, especially by comparing the values between one's preference against their performance metrics.

REFERENCES

[1] Sarthak Ghosh, Lauren Winston, Nishant Panchal, Philippe Kimura-Thollander, Jeff Hotnog, Douglas Cheong, Gabriel Reyes, and Gregory D. Abowd. 2018. NotifiVR: Exploring Interruptions and Notifications in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1447–1456. https://doi.org/10.1109/TVCG.2018.2793698

[2] Lauren Goode. 2019. *A Brief History of Smartphone Notifications*. Retrieved February 15, 2022 from https://www.wired.com/story/history-of-notifications/

[3] Samat Imamov, Daniel Monzel, and Wallace S. Lages. 2020. Where to display? How Interface Position Affects Comfort and Task Switching Time on Glanceable Interfaces. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 851–858. https://doi.org/10.1109/VR46266.2020.00110

[4] Rufat Rzayev, Susanne Korbely, Milena Maul, Alina Schark, Valentin Schwind, and Niels Henze. 2020. *Effects of Position and Alignment of Notifications on AR Glasses during Social Interaction.* Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3419249.3420095

[5] Rufat Rzayev, Sven Mayer, Christian Krauter, and Niels Henze. 2019. Notification in VR: The Effect of Notification Placement, Task and Environment. (10 2019), 199–211. https://doi.org/10.1145/3311350.3347190

[6] Sofia Seinfeld, Tiare Feuchtner, Johannes Pinzek, and Jörg Müller. 2022. Impact of Information Placement and User Representations in VR on Performance and Embodiment. *IEEE Transactions on Visualization and Computer Graphics* 28, 3 (2022), 1545–1556. https://doi.org/10.1109/TVCG.2020.3021342

[7] Frederik Wiehr, Alexandra Voit, Dominik Weber, Sven Gehring, Christoph Witte, Daniel Kärcher, Niels Henze, and Antonio Krüger. 2016. Challenges in Designing and Implementing Adaptive Ambient Notification Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (Heidelberg, Germany) *(UbiComp '16)*. Association for Computing Machinery, New York, NY, USA, 1578–1583. https://doi.org/10.1145/2968219.2968552

[8] Julia Woodward, Jesse Smith, Isaac Wang, Sofia Cuenca, and Jaime Ruiz. 2020. *Examining the Presentation of Information in Augmented Reality Headsets for Situational Awareness.* Association for Computing Machinery, New York, NY, USA. https://doi-org.ezproxy2.library.colostate.edu/10.1145/3399715.3399846

[9] André Zenner, Marco Speicher, Sören Klingner, Donald Degraen, Florian Daiber, and Antonio Krüger. 2018. Immersive Notification Framework: Adaptive 'I&'; Plausible Notifications in Virtual Reality. (2018). https://doi.org/10.1145/3170427.3188505