A Study on Graphics Quality in Smartphone Games

by

Shady E. Glenn

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science (Honors Scholar)

Presented May 25, 2012
Commencement June 2012

AN ABSTRACT OF THE THESIS OF

Shady E. Glenn for the degree of Honors Baccalaureate of Science in Computer Science presented on May 25, 2012. Title: A Study on Graphics Quality in Smartphone Games.

Abstract approved:

____

Mike Bailey

As of February 2012, approximately 46% of American adults own a smartphone. The graphics quality of these devices gets better each year. However, they still have many more limitations in graphics processing and storage space than desktop computers. This means that applications on these devices should focus on optimizing their file sizes and graphics quality in order to maximize the number of devices that can run and store them. Unfortunately, there is no defined metric for graphics resolution on smartphones. This thesis explores what users believe to be the minimum acceptable graphics quality in smartphone games and graphics applications. By using a testing program we designed in OpenGL, we were able to find at what point in an image's degradation users found it graphically unappealing and found the app unacceptable. Participants gauged four images that degraded over time. For our two high frequency images, participants found the minimum acceptable graphics quality to occur at 43 pixels per inch (ppi), while in low frequency images they found minimum acceptable graphics quality to occur at around 31 ppi, with the average minimum being 37 ppi.

Key Words: graphics, smartphones, image resolution, gaming, computer

Corresponding e-mail address: shadyeglenn@gmail.com

A Study on Graphics Quality in Smartphone Games

by

Shady E. Glenn

A PROJECT

submitted to

Oregon State University

University Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science (Honors Scholar)

Presented May 25, 2012
Commencement June 2012

Honors Baccalaureate of Science in Computer Science project of Shady E. Glenn presented on May 25, 2012.

APPROVED:

Mentor, representing Computer Science

Committee Member, representing Computer Science

Committee Member, representing Computer Science

Chair, Department of Computer Science

Dean, University Honors College

I understand that my project will become part of the permanent collection of Oregon State University, University Honors College. My signature below authorizes release of my project to any reader upon request.

Shady E. Glenn, Author

ACKNOWLEDGMENT

The author would like to thank several people. I would like to thank my parents for supporting and encouraging me these last 23 years. I would also like to thank my fiancée, Valerie for keeping me motivated and for participating in "thesis time" with me. Last but not least, I would like to thank Mike for helping me focus my ideas to something testable and for being supportive during the entire thesis process.

TABLE OF CONTENTS

LIST OF FIGURES

A Study on Graphics Quality in Smartphone Games

INTRODUCTION

The field of visual communications has been intriguing humankind from the time we were painting cave walls. With the rise of computers we have been striving to display clearer and better images on our screens. Unfortunately, not all screens are created equally and many people use smartphones (46% of American adults own one [Pew Internet 2012]), which have limited graphics processing power. Smartphones have issues with limited space and processing power or excessive cost. These limitations make the goal of constantly present and excellent picture quality inaccessible to the common smartphone user. Since not everyone will be able to have the best picture quality currently available, where do we draw the line? The average smartphone user does not have the latest and greatest smartphone, but has a midrange Android smartphone [comScore 2012] with a 4” screen of about 200dpi [Android Developers 2012a]. Given a smartphones limited space and processing power, our goal was to find the lowest level of graphics resolution that users would find acceptable in a smartphone game. This would bring acceptable graphics to most users.

In the following pages we explore what resolutions users found to be acceptable and where the line between acceptable and unacceptable image resolution lies. In order to do this, we defined the problem and explored what is currently published involving this problem. We present the design of the data gathering tool and the data collected, show its analysis and the conclusions we came to, and then look at what was learned and how this study could be improved if repeated.

PROBLEM DESCRIPTION

Each day smartphones are becoming better and closer to desktop computers, with respect to processing power and display quality, but they still have significant limitations. The two we are concerned about are storage space and graphics processing power. For the purpose of this paper we will be referring to the Android smartphone platform, since Android, as of March 12, 2012 holds a 51% share of all smartphone subscribers in the United States (comScore). Many smartphones have a set capacity with no expansion slots or only support limited memory expansion. For the purpose of this paper, I will refer to a high-end smartphone that is popular, the Samsung Galaxy S2. The Samsung Galaxy S2, according to Samsung's website, comes with either 16GB or 32GB of storage built in with an expandable MicroSD slot up to 32GB [Samsung 2012]. That gives us a total of 64GB possible, which depending on the user, may be acceptable. Someone who streams their media online from sources such as YouTube and Netflix for video and maybe Pandora for music would not need a significant amount of storage on his or her smartphone.

However, some users like to have media available when they have limited service, so we can consider several movies or TV shows on their smartphone as well as a few thousand songs. Assuming a standard song size of 6MB and 7,000 songs with 15 movies of size 500MB and 30 shows at 200MB we get (6MB*7,000 + 500MB*15 + 200MB*30) 55,500MB or 55.5GB. Using the same smartphone as above, we begin with 64GB and after we have added our media we have (64GB – 55.5GB) 8.5GB left, which still does not account for the Android Operating System space used. The goal of this use case is to illustrate that minimizing the size of smartphone applications (apps) is good practice in order to help users worry less about smartphone storage space.

The other reason we have for decreasing app size is that not everyone has access to a wireless connection and could worry about data plans and how much data they consume. So an app developer should focus on making their apps a reasonable size to maximize the amount of users that can use them. Also, some of the lower-end smartphones have less powerful graphics processors that could cause the game to run at unbearably slow frame-rates. One example of this is when I tried to run Angry Birds on my Samsung Moment. It took over two minutes to load the home page and made the game unplayable on my smartphone.

On the other hand, there is the problem of smartphone graphics apps that have been optimized for space too well, and have become graphically unappealing, causing users to stop using the app. Our goal was to find the sweet spot where we can optimize image resolution (reducing size) while keeping the fidelity users expect (keeping the app's graphics appealing). This "sweet spot" could be used to set a standard minimum resolution for graphics quality in smartphones apps, such as games, where there would be a significant amount of images that are generated and this could help define the minimum resolution needed.

PREVIOUS WORK

Many people have noted the limitations of smartphones and pointed out that developers need to take these into account when developing for these devices. However they only note that one should make sure to optimize image size in order to speed up computation and not overload the system, they never note any standards or give any guidelines.

In a paper written in 2008 titled “The State of the Art in Mobile Graphics Research” the same limitations of smartphones I have noted above are listed and they look at methods of compression of textures in order to save space and reduce the quantity of data sent over a bus [Capin 2008 Pg. 76]. The paper notes that different types of compression can be lossier than others, but does not give a standard of what is good.

In Romain Marucchi-Foino’s 2012 book "Game and Graphics Programming for IOS and Android with OpenGL ES 2.0," he makes a note about texture optimization, stating that “Portable-device GPUs have a limited amount of video memory compared to desktop cards. Always remind your artists to keep their texture resolution as small as they can to save as much memory as possible" [Marucchi-Foino 2012 Pg. 84]. So once again the need for optimization is there, but there is no clear definition of what a correct size or optimized size is.

By this problem being noted, but not solved, there was an opportunity for research to find a minimum graphics quality acceptable to users that would create a guideline for app creators.

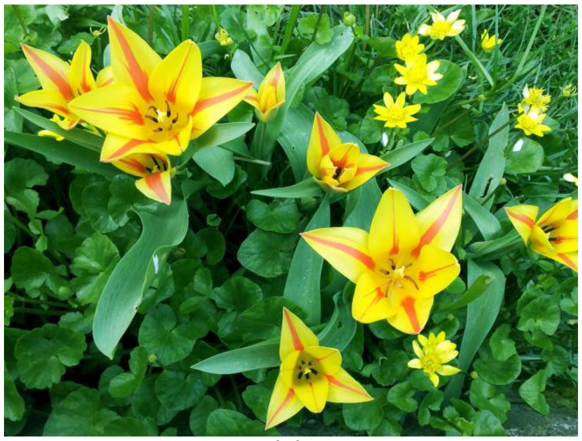
An important part of the testing process was being able to make sure each participant was able to see the images clearly so that the only degradation they would be noticing would be that which we were creating. So we needed to make sure that the users would be far enough away from the screen so they would be unable to discern single pixels of our LCD display. From the paper "Capability of the Human Visual System" we found that "for a human eye with a pupil diameter of 2.5 mm and light with a wavelength of 555 nm (the wavelength at which photopic, or cone-mediatiated, vision is most sensitive), the maximum resolution would be . . . about 0.92 arc-minutes" [Curry 2003]. We used this number to calculate at what distance a human with perfect eyesight would be able to see individual pixels on the screen. This number helped us calibrate our Snellen chart, which we used to determine the distance at which each participant should stand.

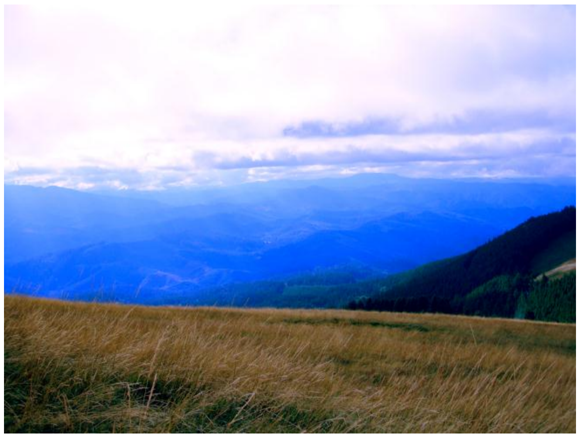
MATERIALS AND METHODS

OpenGL Data Gathering Program

In order to find out what the lowest resolution users find to be acceptable, we created a data gathering program in OpenGL and had participants look at a series of images whose image quality slowly degraded and had the participants make keyboard input when they thought the resolution was the lowest acceptable. We had four different images that participants analyzed, each one having a different main scene color as well as having a different frequency of elements.

(a)


(b)


(c)

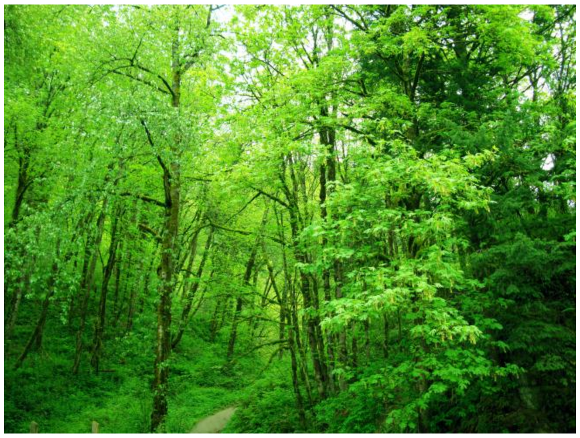

(d)

Figure 1. ***(a)*** **Scene 1: high frequency in ice chunks.** ***(b)*** **Scene 2: low frequency in flowers.** ***(c)*** **Scene 3: semi-high frequency in grass, low frequency in blue valley.** ***(d)*** **Scene 4: high frequency in leaves. All images are the property of Shady Glenn.**

All of these images were ones I had taken. I converted them to bitmaps, which I then read in using the imageloader.cpp file from the OpenGL "Textures" tutorial on www.videotutorialsrock.com. For each image read in, we created a 2D texture from the output of the imageloader.cpp file with the glTexImage2D command and applied that texture to the front face of a quad that is the size of the screen. We displayed the scene at its full quality initially and then reduced quality by 5% of the full quality every second. The way we did this was by mapping the texture to a section of the quad instead of the entire quad, then we took a snapshot of that section using the command glReadPixels with GL_UNPACK_ALIGNMENT and GL_PACK_ALIGNMENT set to one, meaning that the pixels we read in from the screen being read in one at a time. Once we had this snapshot of pixels we created a 2D texture from it and mapped this new texture to the full quad. Figure 2 shows the method where we take the black rectangle, scale it down to one of the rainbow colored rectangles (5% intervals), take a snapshot of the scaled down picture on the screen, make a texture of it, and then stretch that new texture back to the black rectangle, thus reducing quality. This scaling procedure of decreasing quality by 5% happened every second, so the program moved

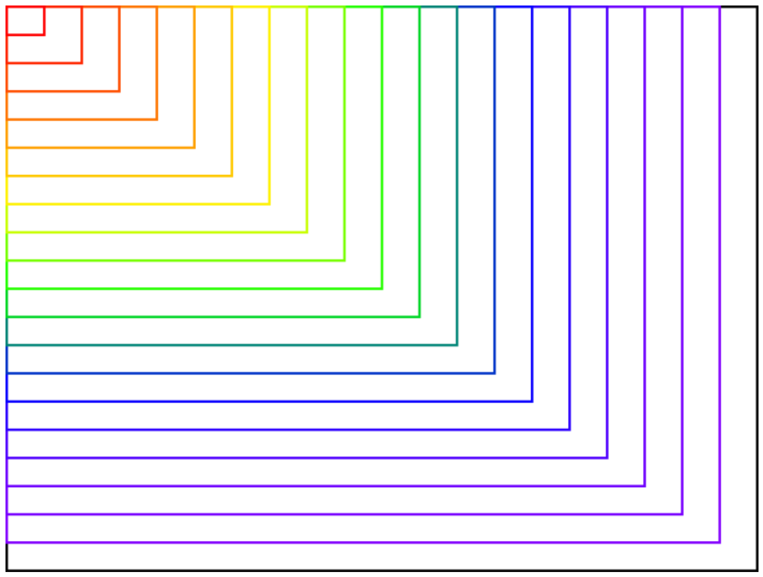


Figure 2. Visualization of image-resizing technique that begins at full image resolution and scales down by 5% each second until it reaches a minimum size of 5% complete resolution.

one rectangle down, starting at black, each second until it reached red, 5% total image quality, where the program would stop with this scaling and move to the next scene.

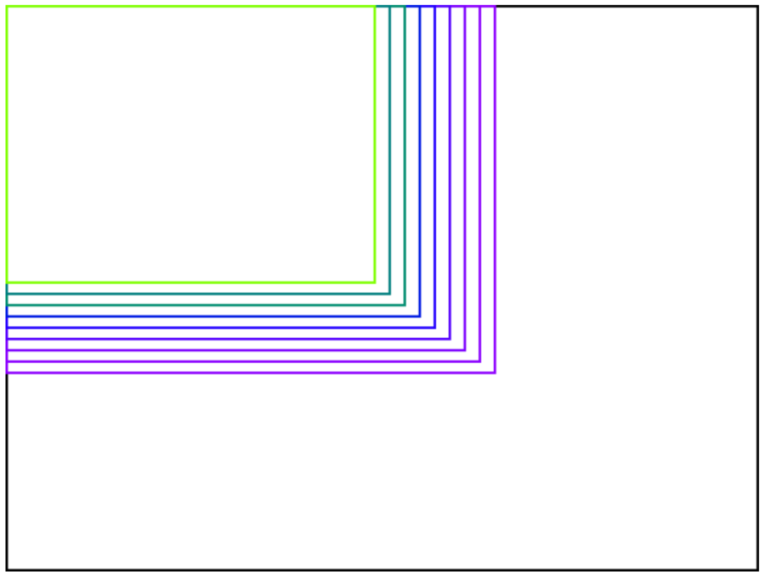


Figure 3. Demonstration of image-resizing technique that occurs during the third viewing of a scene.

The participants saw each scene go through this degradation process three times. On the second and third showing of a scene, the program moved to the next scene once the participants had denoted that they found the sweet spot. On the third run of a scene, the program looked at the first two sweet spots the participants selected, picked the least degraded one, increased quality to 10% better than that location and began the third run degradation at that point. During this third run of a scene, the image was degraded 2% per second. This allowed us to get a more precise location of where the participant's sweet spot was for that scene.

In order to find out what participants found to be the lowest acceptable resolution, we needed to define a few things. We looked at two different types of resolutions: pixel resolution and spatial resolution. Pixel resolution is the actual width and height of an image measured in pixels. For testing, we used a screen with a pixel resolution of 1280x800 (1280 pixels wide by 800 pixels tall). On this screen we ran our testing program which had a pixel resolution of 900x600, so it took up a majority of the screen we were using. Spatial resolution is, in its simplest terms, how clear an image appears. The units that we will be using to measure spatial resolution are pixels per inch (ppi). The spatial resolution of our test images was 72ppi.

We rendered a set of four test images to the screen and slowly decreased their spatial resolution while keeping their pixel resolution constant. The participant gave input by using the arrow keys. We had the participant tap the right key at the beginning of each scene a couple of times to designate good quality, and then they would hit the left key, denoting the location where they began finding image quality unacceptable. The reason for having them hit the right key (really any “arrow” key, the program accepts the main arrow keys, the numpad arrow keys, W, A, S, D, or E, S, D, F all as arrow keys) is to figure out what the good quality key is, and then once they hit a different key, that is the bad quality key. When we detect a switch from good quality to bad quality, that is the sweet spot, the location where the participant no longer finds image quality acceptable. The reason we had a good quality key and a bad quality key is so that we could detect that change point with certainty. It also allowed our participants to skip the rest of the scene in the second and third viewings in order to save them time as well as the unpleasant experience of watching the image degrade.

The main data we gathered came from the third run of each scene. We did this because the third scene had the most precise data. During the first run, the participant got used to the image and saw how far it would actually degrade. In the second run the participant decided on a rough location of the sweet spot. In the third run, the participant specified a more precise location based on the first two passes.

Gathering Accurate and Consistent Data

The pixel resolution on the test laptop was 1280 px by 800 px and the screen size was stated at 15.4 inches, but after measuring we found it to be 15.3125 inches. In order to find the distance between the pixels, we needed to get the number of pixels per inch or PPI. We got PPI by comparing the hypotenuse of the 1280x800 rectangle with the 15.3125" screen size measurement which is measured on the diagonal. So sqrt($1280^2 + 800^2$) = 1509.44. To get ppi, we just take the pixel screen size and divide it by the screen size in inches, so 1509.44 px / 15.3125" ≈ 98.58 px/in. Now that we have pixels per inch we want to get the dot pitch, which is the distance between pixels. A pixel is made up of red, green, and blue dots that are in a triangle shape, one of each color. Dot pitch is the distance between these triads and is measured in millimeters per pixel, so we needed to invert our px/in to 1(px/in) and multiply that by mm/in to get px/mm. There are 25.4 mm in an inch, so we used the mm/in value of 25.4/1. We inverted our 98.58 px and multiplied it by 25.4 to get (1/98.58)*25.4 ≈ 0.26 mm/px or dot pitch.

In order to have a visually appealing graphic, participants should not be able to distinguish single pixels, but should have the pixels blend together to make a smooth image. To find this distance, we will use the number for the maximum visual acuity of humans, which is about 0.92 arc-minutes [Curry 2003]. From the above calculation, we know that the dot pitch is about 0.26 mm/px meaning that with a little trigonometry we can figure out ate what distance a viewer with perfect vision should be. We have a small triangle with the angle being 0.92 arc-minutes, the opposite side being the dot pitch (since the participant will be viewing at a 90 degree angle), and the adjacent side being the distance between the participant's eyes and the monitor. We express this equation in this manner: tan(angle) = opposite/adjacent, so tan(0.92 arc-minutes) = .258mm / x → x = .258 mm / tan(0.92 arc-minutes) = x -> x = 962.8 mm, which is equivalent to 37.91."

If we were dealing with humans with the best possible vision, then they would need to sit about 38" away from the monitor. This is an unrealistic number, since most corrective lenses and the vision of general people differs so greatly, with the accepted standard of vision being 20/20, while those with excellent vision can have 20/10 or even better vision. Since most humans have less than half the vision of our perfect vision number, we can safely assume that somewhere around 15" away from the monitor should be about where most people should sit. This will allow them to see the screen clearly, as well as not have a distorted image from being too close that they can see the pixels, or so far away that their below perfect vision hampers the images clarity or makes the detail difficult to discern.

However we needed some metric that would allow all our participants to be able to see the screen with equal clarity. We decided to use a Snellen chart, which is a visual acuity measuring chart (where the 20/20 measuring system comes from) and have each participant focus on the clarity of a specific letter [Watt 2003]. We used a scalable vector graphic (SVG) version of the chart, which allowed the graphic to be drawn at whatever resolution asked of it and keep its quality at any resolution, since it is a set of drawing instructions instead of an array of pixels. We scaled down a Snellen chart svg file from Wikipedia to a size of 385 px wide and 479.87 px tall using a free SVG editing program called Inkscape [Wikimedia Foundation 2011]. We then had the participants view the svg file displayed in the Chrome web browser, since it initially rendered svg files to be at the size specified in the file, instead of a browser preset. We had participants position themselves relative to this chart, specifically the "F" in line 9. We had participants move away from the monitor until the "F" appeared to look more like a lowercase "R" or an uppercase "P" and then move forward until they could discern the second line of the "F" and tell that it was an "F" instead of an "R" or "P." This distance kept participants from getting too close to the screen to discern pixels, kept them close enough to the screen to clearly see the graphics, and solved the problem of getting everyone in a position where they all could view the graphics at approximately the same quality.

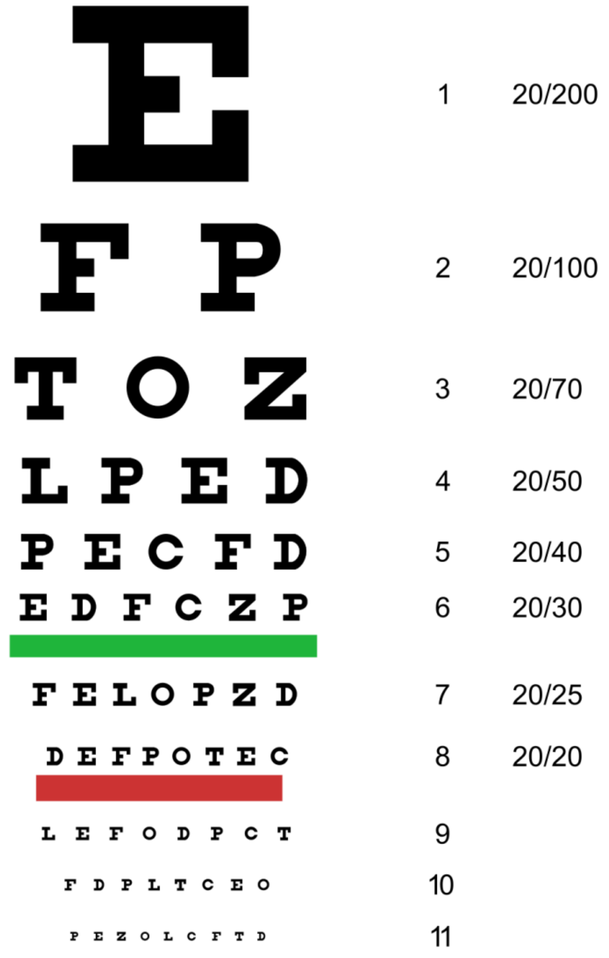


Figure 4. Snellen chart used for testing participant's visual acuity.

Institutional Review Board

For this study we had to go through the Institutional Review Board (IRB). There were several tests we had to take in order to become IRB certified as researchers. We also had to get our protocol, consent process, and the justification and benefits of our study approved. For the main protocol document, we needed to state our qualifications and give a basic description of the research to be done. We had to decide on our target population as well as the method with which we would interact with our participants in order to gather our data. We gave the chronological outline of what a participant would go through for the research, as well as providing the speech we would give participants warning them about the risks and stating the benefits of the study. We stated how we would make sure the participants were able to understand what they were agreeing to and that they did indeed consent to participate. We went through an explanation of how we were going to go about recruiting for the study. Finally, we had to explain how we were going to keep any personal data safe and how we would make sure to keep it available for at least three years. Once we received approval from the IRB, we began testing using the method above.

RESULTS

We realize our demographic for this study was limited and we had a small number of participants. Most of our participants were 18-24 year old males. However, males of this age do represent a decent part of the gaming population, so we still were able to make useful conclusions, though they apply mostly to this group.

All of the results we looked at were based on the third run of each scene. We made this decision since the goal of the first two runs of each scene were to get the participant used to how the scene degraded and modified the beginning resolution of the third run, making it so the participant would get to the sweet spot faster. Also on the third run the image degradation occurred at 2% every second instead of 5% every second, giving better precision on where the sweet spot actually is.

Table 1. Sweet spot location averages, standard deviations, PPIs.

	Scene 1	Scene 2	Scene 3	Scene 4
Degradation Level of Sweet Spot	56.9%	70.4%	66.6%	56.0%
Standard Deviation	13.7	11	10.4	13.7
Minimum Participant Found Pixels Per Inch	42.5ppi	29.2ppi	32.9ppi	43.4ppi

PPI calculations are modeled after the total scene PPI calculation at end of results section.

Between all of the four scenes the average sweet spot occurred at 62.5%. This value means that the participants found the sweet spot to be at 62.5% degradation of the original images (which have an original width of 2000px and height of 1500px). The participants found the minimum accepted graphics quality level to be at 337.5 px by 225 px of the original size (we calculated this by taking 100 – the sweet spot %, multiplied it by .01 to get a fraction, and then multiplied that by the original image size. For width we got (100 - 62.5)*.01 * 900 giving us 337.5 px and for height we got (100 - 62.5)*.01 * 600 giving us 225 px. This value is less than half the quality of the original image.

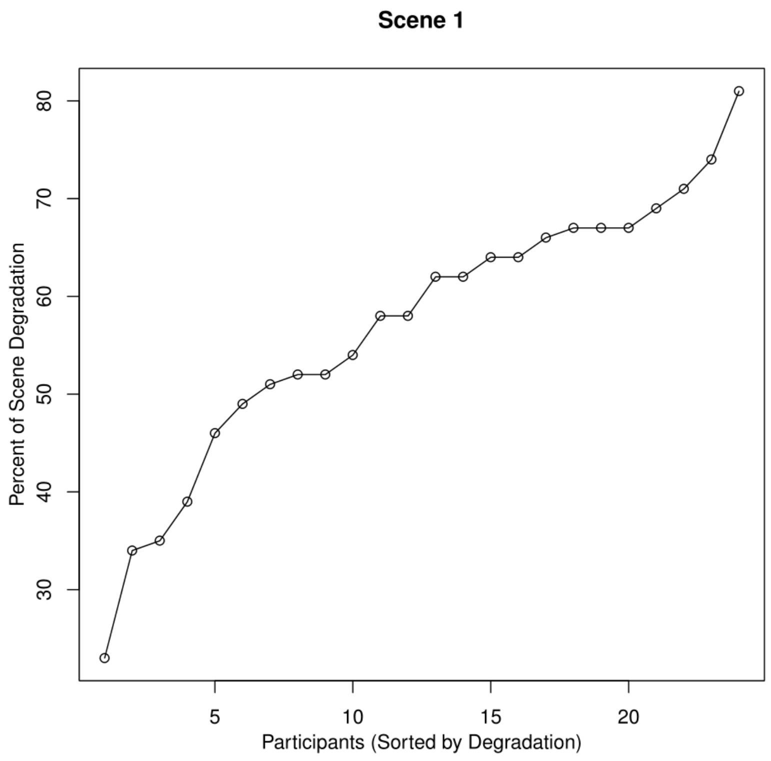


Figure 5. Scene 1 data, ordered by lowest degradation to highest.

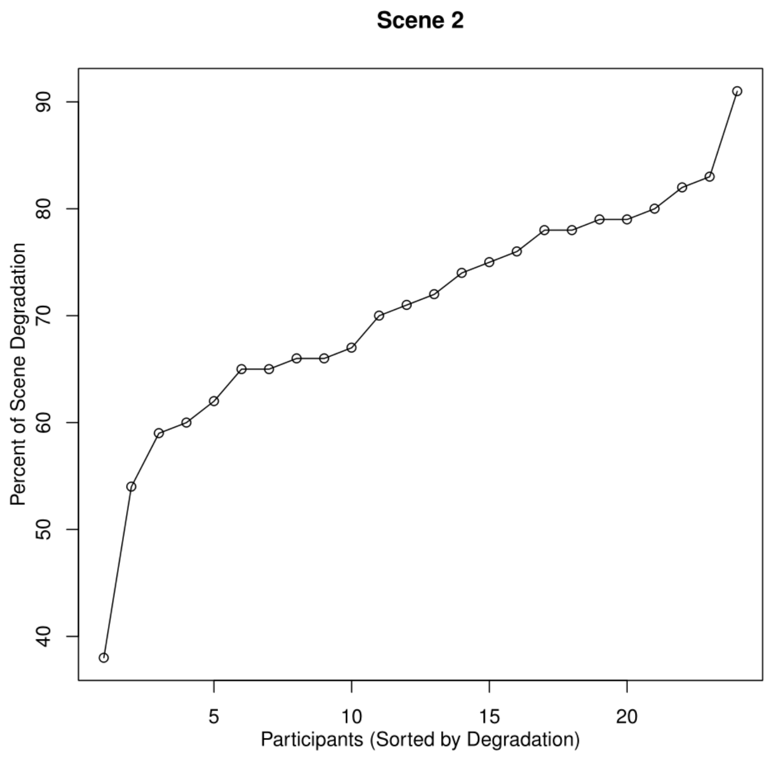


Figure 6. Scene 2 data, ordered by lowest degradation to highest.

<u>Scene 1</u> was the ice chunk scene. Most participants found the sweet spot to be between 45% and 70%. This scene has a high frequency of ice chunks, which it is easy to tell when the quality changes, due to the many shifting chunks of ice, but some participants could find it acceptable, since the high frequency items blend together.

<u>Scene 2</u> was the tulip scene, and most participants found the sweet spot to be between 60% and 80%, meaning that the scene could degrade over halfway before participants found it unacceptable.

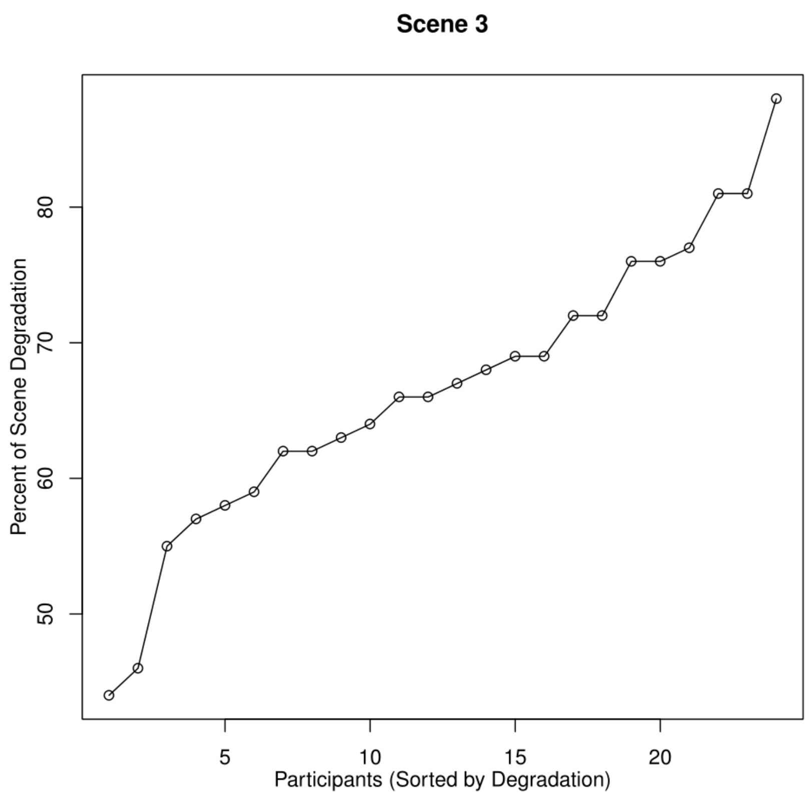


Figure 7. Scene 3 data, ordered by lowest degradation to highest.

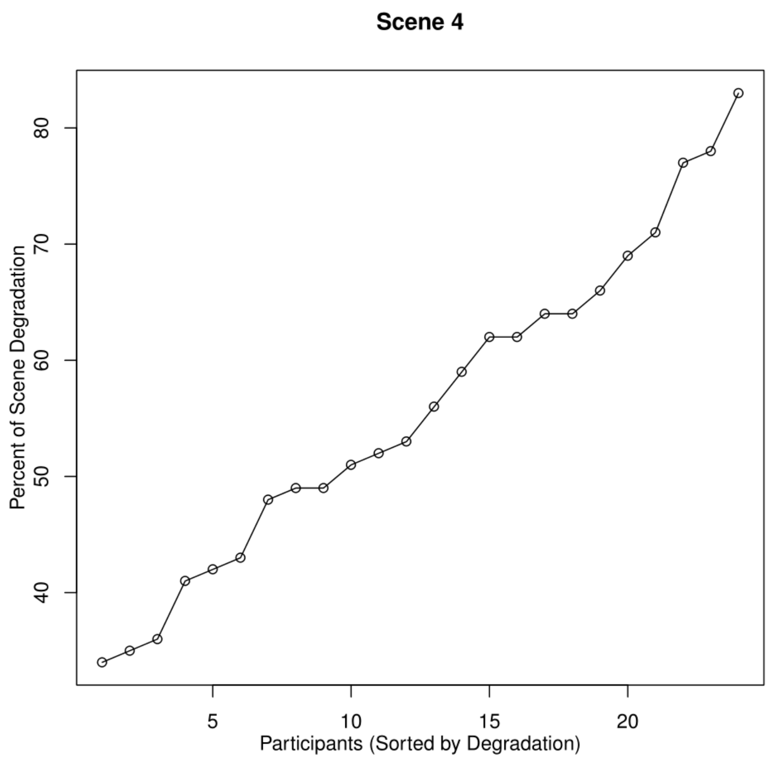


Figure 8. Scene 4 data, ordered by lowest degradation to highest.

Scene 3 was the hill scene overlooking the Willamette Valley. Most participants found the sweet spot to be between 55% and 80%. This is most likely due to the high frequency grass blending instead of pixelating significantly as well as the main focus of the image being the blue of the valley, which does not have sharp edges.

Scene 4 was the deciduous tree scene and has high frequency leaves. Participants found the sweet spot to be between 35% and 70%. This is most likely because the leaves when degraded do not blend well, but get pixilated. However depending on where the participants were looking, they may not have focused on the leaves so much as the path.

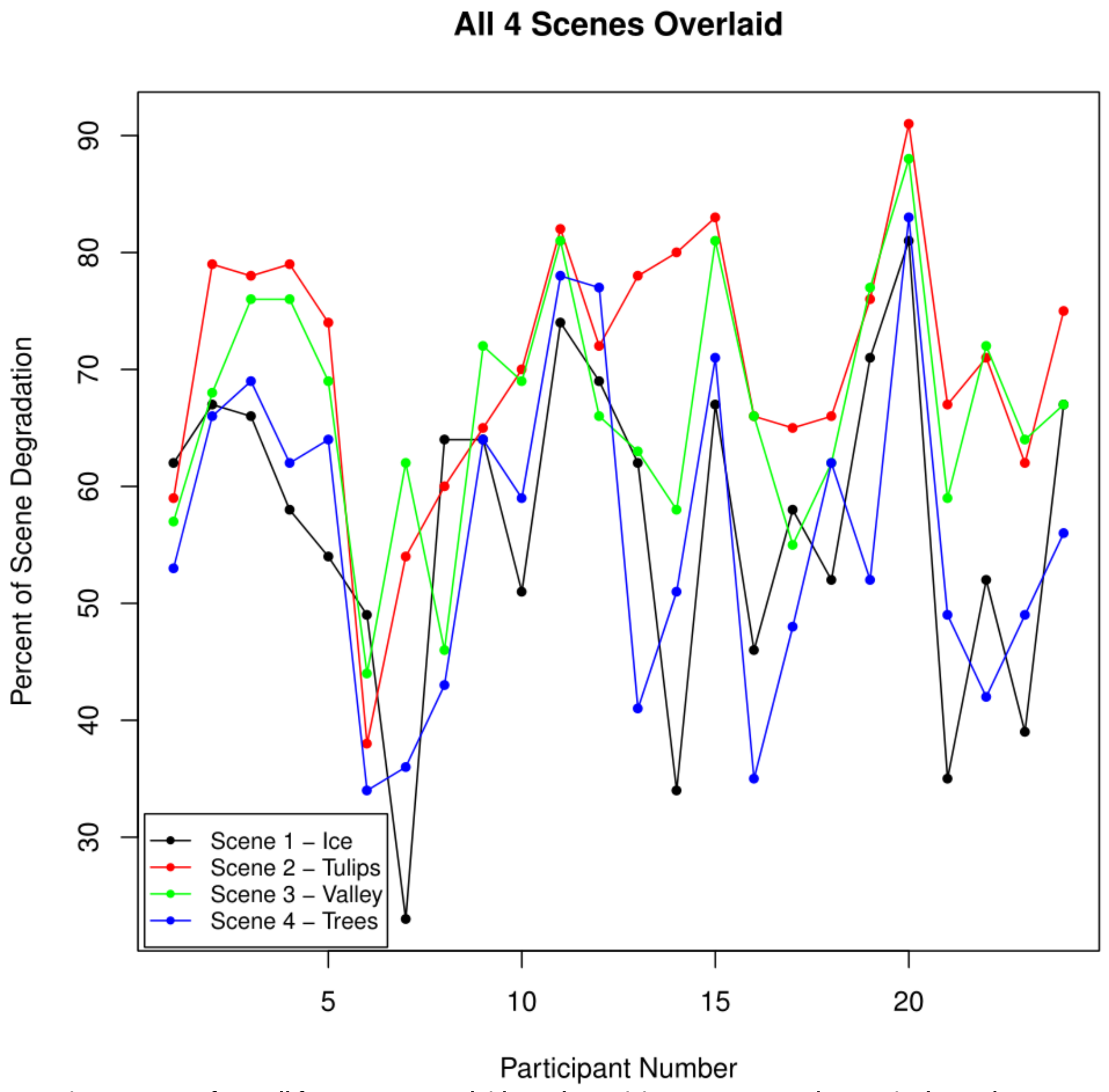


Figure 9. Data from all four scenes overlaid. Each participant corresponds to a single x value.

This figure shows how individual participant's data matches up between the scenes. For instance, participant 1 had a 9% range for all of the scenes, while participant 14 had a 46% range. Participant 14's large range came from his interpretation of the sweet spot in scenes 1 and 2. In scene 1 he said that the sweet spot was at 34%, suggesting that he was focusing on the changes in the high frequency ice chunks, while in scene 2 he was most likely focusing on one of the tulips instead of the grass. The tulips are quite large and of uniform color, so the degradations take quite some time to become visibly distorted. Our demographic was primarily male, with 21 out of 24 participants being male. Fourteen of these participants were in the age range of 18-22; the other seven were in the range 26-43. There did not appear to be any difference between male and female, but more testing would be needed to confirm this. Age did not seem to play any noticeable role in selection of the sweet spot, but a larger data set would be needed to confirm this.

In order to calculate the exact pixels per inch that participants found to be minimally acceptable we need to use our numbers from above for some math. On our test screen we had a resolution of 98.58 px/in, so we wanted to get the px/in of our 337.5 px by 225 px degraded image, but in the original 900 px by 600 px (converted to inches) frame participants viewed the images in. We did this by taking the hypotenuse of the degraded image in pixels and dividing it by the hypotenuse of the original scene size in inches on our test monitor. We got that the degraded image's hypotenuse is sqrt(337.5^2 +225^2) = 405.62 px. For the hypotenuse of the original scene size in inches we needed to get the hypotenuse and multiply that by our test screen resolution which was 98.58 px/in (since we want inches we will invert it, so 1 in/95.58 px), so sqrt(900^2 + 600^2) * (1 in/95.58 px) = 10.98 inches. To get pixels per inch we took our 405.62 pixels / 10.98 inches to get approximately 37 pixels per inch, which is significantly less ppi than our screen spatial resolution.

CONCLUSIONS

We realize we had a limited demographic and a small number of participants for this study. If we had more time, we would have liked to have extended our research to consider a more varied population. However, college age males do represent a decent part of the gaming population, so our conclusions will have application within this group.

For scenes with high frequency, we found that participants had a higher standard for minimum acceptable graphics quality, with the ice chunks scene having 42.5 ppi and the forest scene having 43.4 ppi.

For scenes with low frequency, we found that participants had a lower standard for minimum acceptable graphics quality, with the flower scene having 29.1 ppi and the valley scene having 32.9 ppi.

Between all of the scenes, the average minimum accepted graphics quality was approximately 37 ppi.

WHAT I LEARNED

There were two main aspects that would have significantly helped if we would have done them differently. First was participant input. The idea was to have a participant sit at a computer and give input by typing on a keyboard. Unfortunately when we were positioning participants by their visual acuity, we ran into instances where participants had 20/30 or 20/10 vision. We assumed people would have a range between 20/20 or 20/25, so we made the comfortable sitting distance from the screen be at 20/25 which was about fifteen inches from the screen. So people with 20/30 were close to the screen, which was still manageable, but people with 20/10 vision (something around one fifth of the study!) had to stand about five feet from the monitor. Consequently I ended up being the one inputting their key inputs when they said “now” or something similar. This was not the optimal solution, it would have been better if the participants had a wireless device they could use for input.

I had assumed at the beginning of the study that participants would be hitting more keys than they actually did. This was why I had them hit the right arrow key a bunch in the beginning to denote good and then hit the left for when it was no longer acceptable. However this method was more confusing to participants. Instead what I should have done is have the participants only give input when they perceived the sweet spot. Something like a presentation advancer, with only one button, would have been significantly less confusing to participants and avoided the distance problem.

Another issue we ran into is the way that the participants interpreted what good graphics on a smartphone game meant. Most participants were looking at aesthetics and how appealing the picture looked, which was what we were expecting participants to do. However, some participants looked at the functionality of the game and how visibly important elements showed up. One example was in scene 4, where one participant measured the sweet spot by how visible the path was, since in games you generally need to physically move through a space and a path is the logical way to move. We could solve this issue by saying what type of detail we are looking for, or instead of asking about graphics quality in a smartphone game, just asking about a smartphone application.

One interesting cause for the varying levels of quality decisions participants made was what they assumed was acceptable. Some participants commented that they had been playing a lot of Minecraft recently (game focusing on building things out of blocks with blocky and low levels of graphics), so were able to tolerate a significant amount of degradation and pixelation. On the other hand, there were the graphics elitists who have powerful gaming rigs and won't tolerate any pixelation, so they noted the sweet spot once they saw pixelation.

If we were to go through the IRB again for a human research group, we would make sure to give a larger window for paperwork to make it through the IRB before we plan to begin researching. A period of about twelve weeks would have been better for us, since we had to make a couple revisions to our protocol that the IRB suggested. We would also be less specific in our descriptions, since some descriptions limited us. Instead of saying the participant will view scenes sitting, with back straight, say that the participant will view the scenes from an appropriate distance and location.

One thing that would really have helped is if we had more time to gather more as well as varied data. In order to do this, we should have given the IRB more time to process our application as well as making sure that they had received our application and that it was still in queue for processing. Also it would have been really nice if we had more time for data analysis. Given more time, we could have done several different types of data analysis, including a Fourier analysis of the images in order to better detect frequency of different elements in the image.

BIBLIOGRAPHY

Android Developers 2012a: "Screen Sizes and Densities." *Android Developers*. Android, 1 May 2012. Web. 08 May 2012. <http://developer.android.com/resources/dashboard/screens.html>.

Android Developers 2012b: "Platform Versions." *Android Developers*. Android, 1 May 2012. Web. 08 May 2012. <http://developer.android.com/resources/dashboard/platform-versions.html>.

Capin 2008: Capin, Tolga, Kari Pulli, and Tomas Akenine-Möller. "The State of the Art in Mobile Graphics Research." *IEEE Computer Graphics and Applications* 28.4 (2008): 74-84. Print.

comScore 2012: "comScore Reports March 2012 U.S. Mobile Subscriber Market Share." *ComScore.com*. ComScore, 2012. Web. 08 May 2012. <http://www.comscore.com/Press_Events/Press_Releases/2012/5/comScore_Reports_March_2012_U.S._Mobile_Subscriber_Market_Share>.

Curry 2003: Curry, David G., Gary Martinsen, and Darrel G. Hopper. "Capability of the Human Visual System." *Cockpit Displays X (Proceedings of SPIE)*. Ed. Darrel G. Hopper. Vol. 5080. Bellingham, WA: SPIE, 2003. 58-69. Print. Security, Cockpit, and Future Displays II.

Glenn 2010: Scene 1, Broughty Ferry, Scotland. Personal photograph by author. 2010. <https://docs.google.com/open?id=0B6MDoVFy8NGTVV9ycEFHTXZMQ1U>.

Glenn 2011a: Scene 3, Mary's Peak, Oregon. Personal photograph by author. 2011. <https://docs.google.com/open?id=0B6MDoVFy8NGTdFFiVDdwdFZHSmc>.

Glenn 2011b: Scene 4, Latourell Falls, Oregon. Personal photograph by author. 2011. <https://docs.google.com/open?id=0B6MDoVFy8NGTeVdVMmdZZ0tuOGM>.

Glenn 2012: Scene 2, Corvallis, Oregon. Personal photograph by author. 2012. <https://docs.google.com/open?id=0B6MDoVFy8NGTX2dxMWtGX1IxdEE>.

Marucchi-Foino 2012: Marucchi-Foino, Romain. *Game and Graphics Programming for IOS and Android with OpenGL ES 2.0*. Chichester: Wiley, 2012. Print.

Pew Internet 2012: "Nearly Half of American Adults Are Smartphone owners." *Pewinternet.org*. Pew Internet & American Life Project, 1 Mar. 2012. Web. 12 May 2012. <http://pewinternet.org/Reports/2012/Smartphone-Update-2012/Findings.aspx>.

Samsung 2012: "Samsung GALAXY S II." *Samsung.com*. Samsung. Web. 08 May 2012. <http://www.samsung.com/global/microsite/galaxys2/html/specification.html>.

Watt 2003: Watt, Wendy Strouse. "How Visual Acuity Is Measured." *Mdsupport.org*. MD Support, Oct. 2003. Web. 20 May 2012. <http://www.mdsupport.org/library/ acuity.html>.

Wikimedia Foundation 2011: "File:Snellen Chart.svg." *Wikipedia*. Wikimedia Foundation, 27 Nov. 2011. Web. 14 May 2012. <http://en.wikipedia.org/wiki/ File:Snellen_chart.svg>.