Choir Maker

Senior Integration Project

Reagan Duggins

1 May 2016

INTRODUCTION

In Christian education, it is not uncommon for people to consider disciplines like computer science to be less Christian--or more “worldly”--than disciplines like theology or Biblical studies. Although this perspective may have surface appeal because of the explicit references to God in theology and the Bible, it ultimately reflects a non-Biblical worldview. It wrongfully “limits” God to certain spheres of reality and human activity and excludes Him from others. But the truth is that computer science is just as “Christian” as any other area of human endeavor not devoted to sin and corruption. By its very nature, computer science reflects God’s glory by beautifully revealing a wondrous blend of creativity and order. And as if that were not enough, it has a double role. It can also serve as a practical tool for assisting people in reflecting and glorifying God in other areas of life. My Senior Integration Project--Choir Maker--is one example or one illustration of this two-fold “Christian nature” of computer science.

THE ORIGINS OF CHOIR MAKER

The idea behind Choir Maker was born about a year ago in a conversation with Dr. Scott Finch, a music professor at Covenant College. He was talking with a friend of mine about the amount of work that happens behind the scenes in order to present a choir performance. He noted that most people do not realize the sheer amount of man-hours that go into such a production. When the typical person sees a performance, they assume the choir practiced the songs a few times and then sang them to the audience.   Most in that same audience usually don’t realize or think about what it takes to organize the event, to work through the red tape to make the event happen, to recruit people, to get everyone to practice, to arrange for facilities and equipment, and to generally make sure everything is prepared effectively for the performance. But what even fewer people recognize is the work that goes into such seemingly simple tasks as determining where choir members should sit and stand during the performance. That is where Choir Maker comes in.

Before I was in a choir, I knew that choir seating was not completely random. But beyond that, about the only thing I knew about the seating of choir members was that the ladies were typically on each side and the men were in the middle. I had no clue, and many audience members have no clue, about the degree of detail and work that can go into determining the exact seating position of each choir member. In truth, a high quality choir director devotes serious time and attention toward placing each choir member intentionally into a specific position. Dr. Finch, for example, literally clears his desk, writes each choir member's name, height, and other pertinent information on a separate sticky note, and then moves the sticky notes around on his desk to create a "sticky note seating chart." Each person's sticky note from the seating chart is then placed on the seat where that particular choir member is supposed to sit.

Of course, not all choir directors prepare their seating chart in the same way that Dr. Finch does. I am aware of seating charts that are prepared from several sheets of paper marked with squares and then combined. I am aware of seating charts prepared by a team of teaching assistants working together for hours. The exact methods vary, but all must prepare seating charts. At least they must if they seek to present a choir in excellence. The idea behind Choir Maker is to make that seating chart development process easier, faster, and better. By doing so, it it saves time for a director and others, reduces the burden on the choir leadership, and reduces the frustration and stress that can often accompany preparation for a choir performance. That in turn enhances the overall experience of the performance.

One of the most important factors in developing a choir seating chart is each person's voice type. The choir director will typically want people with the same voice type grouped or seated together, but there may be times when they want them more scattered for a particular purpose. Regardless, seating based on voice type is very intentional. The initial level of voice-type classification is soprano, alto, tenor and bass. But even that typically breaks down into soprano 1, soprano 2, alto 1, alto 2, tenor 1, tenor 2, bass 1 and bass 2. All of these differences or different classifications need to be considered when preparing a good seating chart.

Likewise, a choir member's height is very important to the analysis of where a particular person should be positioned in reference to his or her fellow choir members. Generally, taller members need to be positioned behind shorter members for several reasons. First, placing a taller member in front of a shorter member can block some of the sound from the shorter member, making it harder to hear the shorter member's voice. This in turn negatively impacts the overall sound quality of the performance. Second, positioning the shorter person in back makes it difficult or even impossible for the audience to see the shorter person. While this may seem a minor issue at first, it can have negative effects on the quality of the performance. If the audience includes family members or friends of the choir singer, it can be very important for an audience member to be able to see his relative or friend in the choir.

Third, positioning choir members by height is an important component of the visual experience of a well-done choir performance. A choir performance is not all about sound. Unlike listening to the radio or a CD, where all is auditory, attending a concert or choir performance in person is a visual experience as well as auditory. And that visual experience can include things like the facial expressions, or sometimes even dance moves, of the choir members. This feature is more important with some types of music than others, but the facial expressions often tell part of the story and emotion that the performance is designed to convey. If some choir members are hidden because of taller members standing in front of them, the missing facial expressions can cause the performance to lose a significant part of its impact. And in addition to all this, some of the visual experience is as simple as achieving symmetry that is more aesthetically pleasing than a more scattered or random appearance.

Even beyond what the audience sees and hears, ignoring height has negative consequences within the choir itself. If one of the shorter choir members has spent hours and days preparing for a performance, only to be completely blocked by a taller member, it is not hard to imagine the detriment to morale that this would cause. In the end, much of the beauty of music relates to its unity--unity of purpose, unity of voices, and even unity of appearance. Accounting for height when developing a choir seating chart helps create and display that unity--through auditory harmony, visual harmony, and even emotional harmony.

While singers' heights and vocal range or vocal types are typically the most important components of a choir seating arrangement, other factors can come into play as well. Assuming the choir is not acapella, a complete seating chart should also place the instrumentalists. And placement of instrumentalists might also impact placement of singers. Another factor that could be accounted for in a choir seating chart is simply the strength or weaknesses of particular singers. For example, if a choir included a number of people with pitch problems, the director will not want those particular choir members grouped together. If they were all seated next to or near each other, the result could be compared to the traditional children's game of "telephone." In the telephone game, the first person whispers something to the second person who attempts to whisper the same thing to the third person who attempts to do the same for the next person and so on. By the time the whisper gets to the last person at the end of the line, the statement has inevitably changed multiple times and is a far cry from what the first person told the second person. Perhaps the second person's whisper to the third was only slightly off, and perhaps the third person's whisper to the fourth was only slightly off of that, but a series of small differences can add up to a huge difference. The same dynamic can exist in choirs. Thus, the ideal choir seating chart will factor in individuals' strengths and weaknesses when placing the singers into position.

Singers are affected by those singing around them. The particularly skilled singers can still hit the right pitch even if next to someone who cannot, but one who has pitch problems needs to hear the "right" sound next to him in order to increase the likelihood of hitting the right note himself. But if someone with a pitch problem is hearing his neighbor miss the note, that will often cause him to miss the note as well, but not only that, he will often miss the note a little more. The cumulative effect that a group of pitch-problem singers have upon each other can become quite significant and thereby create a rather noticeable problem with the overall performance. In fact, the result can be many different “songs” or “tunes” being sung at the same moment in time. On the other hand, if the director has strategically positioned the pitch-problem singers so that none of them are next to each other, they will typically be pulled in the right direction and their individual performances will be enhanced by the more skilled singers around them. As a result, there is unlikely to be a noticeable problem with the choir performance as a whole. This further illustrates the importance of the director’s seating chart.

It is my hope that at this point, a question has occurred to the reader, or at least a feeling that something seems strange about the previous couple of paragraphs. Especially if one has spent time in a more successful choir, it is at about this point that they may be wondering why there are people with intonation issues in a choir. After all, is not a choir generally made up of talented singers? This is the beauty of Choir Maker. As a free, open source project, this project is uniquely able to help choirs that cannot afford the “best” training or equipment. This program is not built to cater to the choir of Juilliard or the Mormon Tabernacle Choir. The target recipient of this project is the choir director for a school or church down the street who may not have $1,000 to spend on a choir seating program or even on new equipment. It is for choirs at churches like New City East Lake, where less than 10% of the population has gotten any sort of college degree, and more than 50% have never been to college.[1] Where the average household income is only $26,000.[2] For this type of community, even having a choir is an amazing blessing. It is choirs like these that this program hopes to serve. Just as our heavenly choir will contain singers from every nation, tribe, people, and language, so should our earthly choirs not exclude people because of lack of skill or training. That is the ideal that Choir Maker hopes to embody.

Of course, there are sometimes other special needs that may not have to do with singing capability. These factors can range from accommodating physical disabilities of singers to planning for a particular singer to--as a part of the performance--step out of the group as a whole and sing a solo, make a comment about a song, or otherwise temporarily perform some type of individualized function rather than singing as part of a group. Logistically, it is best for this particular choir member to be positioned in a place where his emergence from the group is barely noticeable or at least does not create a disruption.

In the end, a choir seating chart and its effects are generally not noticed by the observing public, but it truly is an important component of a choir concert performance. And the creation of the chart can be a rather burdensome, frustrating, and time-consuming process. Directors or their assistants can spend hours, or even days developing an appropriate seating chart for a particular choir event. This process can be greatly simplified and move much more quickly, and with much less frustration, with

[1] "Point2 Homes." East Lake Demographics & Statistics. http://www.point2homes.com/US/Neighborhood/TN/Chattanooga/East-Lake-Demographics.html.

[2] Ibid..

the help of a computer generated choir seating chart. And that is the idea behind Choir Maker. I chose to create Choir Maker in order to assist and enhance choir performances by developing a program that processes data about individual choir members and about the choir director's goals and then generates a seating chart better and more quickly than the "sticky note" approach or any other manual approach to developing the chart. It will also give choirs without access to excess amounts of money a chance to use a product as good as, if not better than, the $1,000 counterparts.

Now that we have discussed the motivations, purpose, and driving ideals of the Choir Maker project, there is another aspect that is absolutely imperative to discuss: how does this relate to the Christian faith? We have touched on this briefly above, but this is something that requires more than a brief mention, and as such, will be discussed below.

THE INTEGRATION OF FAITH AND THE PROJECT

How does Choir Maker relate to the Christian faith? This is a very important question that has been present throughout the development of this project. After all, as Paul the Apostle tells us in Colossians 3, we are to do everything as if we were working for God Himself. As I was thinking about the implications that the Christian faith would have on this project, I was able to boil everything down to two main categories. Computers give us a great chance to assist people in ways that other professions cannot. And, the study of computer science gives us a unique window into the mind of our Creator, allowing us to glorify Him by creating. Thus, in computer science we are able to glorify God by creating and by serving--two actions that the Bible shows as integral to who God is. It is these two aspects of God's nature that inspired the Choir Maker project and have guided its progress so far.

First, creating. Before anything else in the Bible, the first thing we learn about God is that He created everything, that He is The Creator. A significant portion of the first book of the Bible is dedicated solely to the details of how, what, and when God created which things. It is made clear that He created, not on a whim, but because it is part of who He is. After each creation, we hear God say that what He created was good, or even very good, and through that we see Him enjoying the act of creation. He is an artist who creates, then steps back and rejoices because the piece of art He was working on is good. It stands to reason then, that as bearers of the image of God Himself, we too should take joy in creation.

So, just as the painter has his canvas, or in some cases the roof of a cathedral, the programmer has his text editor. Just as the sculptor has a variety of tools to help him sculpt, the programmer has a variety of languages, libraries, IDEs, and the like to assist him. More explicitly, programming is an art in the same way that painting or sculpting is an art, and just as the artist paints beautiful paintings, or sculpts amazing sculptures, so the programmer can write truly beautiful programs. Donald Knuth explains this idea of programming as an art in one of his online publications:

> When I speak about computer programming as an art, I am thinking primarily of it as an art *form*, in an aesthetic sense. The chief goal of my work as educator and author is to help people learn how to write *beautiful programs*. . . . My feeling is that when we prepare a program, it can be like composing poetry or music; as Andrei Ershov has said , programming can

> give us both intellectual and emotional satisfaction, because it is a real achievement to master complexity and to establish a system of consistent rules.
>
> Furthermore when we read other people's programs, we can recognize some of them as genuine works of art. . . . The possibility of writing beautiful programs, even in assembly language, is what got me hooked on programming in the first place.
>
> Some programs are elegant, some are exquisite, some are sparkling. My claim is that it is possible to write *grand* programs, *noble* programs, truly *magnificent* ones![3]

So we see that programming is an art, and as with any art, we as Christians should find joy in programming in the same way that a sculptor finds joy in a beautiful sculpture or our Creator found joy in creating us.

Simply by writing programs we are, in some measure, imitating our Lord. As Scott Hoelsema so artfully puts it in his paper *Food Pantry Manager*:

> God is the Creator, and humans, created in His image, cannot help but feel the drive to create. Though there are a number of outlets for this creative work, few so strikingly mirror God's acts of creation as computer programming. . . . Programming establishes not just the work, but the laws that govern it. Perhaps it would be fitting to say that programming, though faintly, is the best analogue we know to God's mode of creation – speaking.[4]

Just like when we paint beautiful paintings, or write amazing stories, when we write a program, we are imitating our Creator’s creativity.

Now that we have seen how creation is a vital part of our Creator’s nature, and that as programmers we have an amazing opportunity to imitate Him in this way, we should discuss another vital aspect of God that programmers have unique opportunities

[3] "Knuth: Computer Programming as an Art." Knuth: Computer Programming as an Art. December, 1974. http://www.paulgraham.com/knuth.html.

[4] Hoelsema, Scott. Food Pantry Manager. Covenant College. Covenant.edu. April 13, 2015. https://huss.covenant.edu/sip/pdf/2015_Hoelsema_Scott.pdf.

to imitate: service. Just as the Son of Man came to serve, not to be served, we are also called to serve others. It is as Jesus says, if He came to serve, and we are His servants, how much more are we to serve?[5]

Of course, serving is not unique to computer science. No matter our vocation we have the ability to serve others. In fact, we are called to serve, even if we do not have a vocation. As Christians, service is part of who we are, it is a driving force behind our decisions and actions. Computer science is no exception to this. But each vocation, each area of life has its own unique ways to serve, and special opportunities that other vocations do not have. One thing that is special about computer science is that, due to the amazing versatility that computers allow, we have a special role in serving those who are serving others. Computer scientists can be sort of support line, giving the people on the front lines better tools so that they can do their jobs more effectively. But before we explore this idea in more depth, it will be helpful to deal with a common misconception in Christian communities.

Earlier in this discussion, a viewpoint that many Christians hold was briefly mentioned. That is, the belief that theology or Biblical studies are somehow more "Christian" than disciplines like computer science. This viewpoint is extremely common among Christians, especially in Christian educational circles. Though it is rarely stated, or even realized, explicitly, this thought pattern often expresses itself in subtle ways. It is not uncommon for students in the sciences to feel some amount of guilt that they are not preparing to go into "the mission field" with their profession (though this operates on

[5] Mark 10: 43-45; John 13: 14-15

the assumption that the workplace is not a mission field of its own.) As we mentioned earlier, this is a tantalizing view to hold, since it focuses greatly on the direct discussion of God that are intrinsic to theology and Biblical studies, but upon further analysis we see that it is actually a very non-Christian worldview. Paul the Apostle summarizes why this is not a good way to think in his first letter to the Corinthians:

> Now if the foot should say, “Because I am not a hand, I do not belong to the body,” it would not for that reason stop being part of the body. And if the ear should say, “Because I am not an eye, I do not belong to the body,” it would not for that reason stop being part of the body. If the whole body were an eye, where would the sense of hearing be? If the whole body were an ear, where would the sense of smell be? But in fact God has placed the parts in the body, every one of them, just as he wanted them to be. If they were all one part, where would the body be? As it is, there are many parts, but one body.[6]

It is clear from this passage that as Christians, we are not supposed to consider our worth or “Christianness” based on our skills. Just as an entire body of feet would be useless, we cannot all be missionaries. We need each other, and we need the variety of work that a variety of professions provide. Just as the hands and feet need the eyes and ears, and vice versa, missionaries benefit greatly from the capabilities that computers provide. These capabilities can come in many forms. From speeding up the tasks necessary for everyday survival so that the missionary has more time to preach and teach, to preventing the sudden loss of large amounts of data via backup files and storage, to helping speed up the process of Bible translation, the technology resulting from the study of computers has had a

[6] 1 Corinthians 12

profound effect on modern missions work.[7]

Choir Maker attempts to work along this vein by making the jobs of choir directors much easier, allowing them to spend their time imitating the creative spirit of our Lord. As we discussed above, there is much more to a choir performance than the audience is typically aware of, and many of the hours of preparatory work go to simple, tedious tasks. These tasks, along with being an enormous time sink, add to the already stressful job of arranging, training, and preparing a choir for performance. The more time goes into things like picking which singer sits where, the less time goes into creating and practicing beautiful music. Over time, as these tasks and the amount of time allotted to them add up, the director is likely to become more and more stressed which in turn causes tension to build in the choir, who then inevitably shares this tension with the audience. Obviously, this is not an ideal performance condition. Ideally, the choir and director should be excited and happy about the performance and prepared to share the music with the audience.

Because Choir Maker is a free, open source project, it also has a unique opportunity to assist groups with lower income. There is proprietary software available, but it can be very expensive. This makes it extremely unfeasible for small churches in poor communities to acquire them, forcing their choir director to spend many hours doing work that could be reduced to only a few minutes with the proper tools. With Choir Maker itself, and potentially many derivative projects, these churches will be able

[7] Leibee, Weston. "Technology on the Mission Field." Charity Christian Missions. October, 2007. http://charitychristianfellowship.org/sites/default/files/mission-newsletter-archives/2007-10-technology.pdf.

to tailor-make Choir Maker to fit their individual needs, and that is if there is not already a version that fits their needs.

WHAT I LEARNED THROUGH CHOIR MAKER

My journey with the project

We have discussed how Choir Maker came to be and the Christian rationale behind it, and in some detail we have discussed its functionality, but it is also important to take note of the lessons that the development of this project taught me. I have learned a lot through this project, and some of these lessons will shape the way I view and approach programming or even life. Before we discuss what lessons I learned, it will be helpful to know where they came from, and in what circumstances these lessons were formed.

This whole project came to be when my professor, Dr. Scott Finch, stated somewhat offhandedly that he would like a computer program to help him figure out where to seat his choir students. He described the methods for choir seating chart generation that he had seen used during his time in graduate school. Choir directors in his graduate school would generally delegate the work to five or six teacher's assistants who would essentially work together using the "sticky note method," with each one responsible for a certain section of the choir.

This process, he said, could easily take upwards of five or six hours, resulting in anywhere from 25 to 36 man-hours of work, simply to figure out where the choir members are supposed to sit. It is not hard to imagine that if the choir director must do all of this work by himself, that is a huge chunk of time that he has to spend in tedium before he can even begin the other work he has to do to run the choir. In light of this, Dr. Finch mused that it would be amazingly helpful if a computer could do this for him. Though he said this offhandedly, as part of conversation, the thought stayed with me. So the next week I discussed with him the possibility of making that program a reality and nailed down some of the details about how exactly such a program would work.

Now that I understood how choirs are organized, it was time to decide what the end goal of the project should be. There were several features that Dr. Finch mentioned that would be helpful for the program to have. One would be a drag-and-drop interface, so that the user could drag a singer from a list of singers and drop them into a specific seat in the choir, or move a singer from one seat to another. Another goal of the project would be to have a system for saving and loading choirs for later use. Another helpful feature would be the capability to switch between several different choir formations simply and easily. Also, as a more long term goal, the project should be expandable, so that someday it could be used for orchestras as well as choirs. And of course, the program should also be easy to use for someone who does not have very much experience with computers. All of these are good goals for a program, but as I came to realize later, this project was quickly becoming more than one person could handle in a single semester. Nonetheless, there was now a goal for the project to move toward.

With an understanding of how to organize a choir and an end goal in mind, all that was left was to write the program. In the beginning, Java seemed like a logical choice of language to write in. I am most familiar with Java, and with the Swing libraries a Graphical User Interface (GUI) would not be too difficult to implement. Soon after I began programming, however, I made my first mistake. Being excited about all the useful things I was planning on having the GUI do, I started building the GUI before a framework was in place. Instead of developing the sorting and storage algorithms and data structures first, and then building the user interface so that it would support effective use of the program, I attempted to build a user interface with all the bells and whistles before I even had a program to use it with.

As time went on, my attempts to match up a GUI with the vague idea of how the actual program would work led to more and more complicated GUI schemes. I tried using the NetBeans Integrated Development Environment (IDE) in combination with Eclipse, another IDE, to alternate between generating the look and style of the GUI with NetBeans, and then programming the details with Eclipse. Unfortunately, this method ended up producing very confusing code, so I switched from Java to Ruby, a language I had become familiar with during an internship the previous year. I found a GUI development library for Ruby called Shoes, and began developing the GUI with that. Though Shoes provided a simpler set of instructions for GUI development, I was soon reminded that a user interface, no matter how effective and user friendly it is, is useless without a program to work with. So after some discussion with my professors, I realized it was time for a paradigm shift in the project, and after some deliberation I decided to make the project open source (which will be discussed in more detail below.)

Now that the project was going to be open source, it became apparent that instead of focusing on the look and feel of the user interface, I should create a foundation for others to build on. It should be simple enough that other programmers would be able to learn to use it, but robust enough that it could be modified and grown to achieve all of our long term goals, and potentially even more. This is the project as it stands currently.

Fairly simple, it reads the name, height, special needs, and other pertinent information about each choir member from a .csv file, which is easily created from a Microsoft Excel file by using the "save as" function. Then it sorts the choir members based on their vocal range (i.e. Soprano 1, Soprano 2, Bass 1, Bass 2, etc..) Once that is done, the user picks what formation they want for the choir and the width of the back row of each section, and the program places the singers into seats so that taller people are in the back and shorter people are in the front. Now that the choir has been arranged and seated, it saves to a text file in a format that is easy to read, and should help to visualize the generated choir.

Open Source Production

There are three main factors that convinced me to change the project to open source. Each one works well with the open source paradigm and fits well with the guiding purpose of the Choir Maker project. These factors are that open source development allows the project to grow from many different intellects instead of just my own; it allows for modification of the project to fit particular needs; and it allows me to share this product with others with no monetary cost to them.

The first factor is important to the project because, though I have a reasonable amount of knowledge about both computing and setting up choirs, there are many things about both that I do not know. One good example of this is how to know what choir formations the program should support. As I learned from Dr. Finch, there are entire books devoted to different formations the choir can stand in and what each one does to the sound. There is no way that a program written by only one person could do justice to such a complex field, so being able to utilize the knowledge and wisdom of multiple people would add a lot to the usefulness of this project.[8]

Because the entire purpose of this project is to minimize the amount of time each choir director spends on tedious tasks, it is extremely important that the project is able to be customized depending on the needs of the individual director. There are only so many features that a single person can add to a program, but if there is an entire group working on it, then there is unlikely to be someone whose needs the program cannot be attuned to.[9]

One of the biggest problems with choir generating software as it exists so far is how expensive it is to get. According to Dr. Finch, most of the choir generators already on the market cost upwards of $1,000. And while this may be fine for larger schools or choirs, for any smaller school and most churches, this price is simply not feasible.

[8] Wadhwa, Vivec. "Benefits of Crowdsourcing." The Wallstreet Journal. October 29, 2014. www.wsj.com.

[9] "Gartner: Open Source Software Gives Businesses Flexibility." ComputerWeekly. February 09, 2011. http://www.computerweekly.com/news/1280095102/Gartner-open-source-software-gives-businesses-flexibility.

Thus, for this project to help those who need it the most, it has to be of little or no cost. Open source production allows for this, without sacrificing the effectiveness of the program.

Of course, open source development is not without its own risks. Even a quick Google search can show that there are any number of pros and cons for open source software, but the one that is most threatening to this project is that for open source to work, people have to want to develop the product. If nobody knows about it, or if nobody ends up wanting to work on it, then Choir Maker may not last very long.

With all these things in mind, the advantages of open source development of this project seemed to outweigh the disadvantages. It is my hope that people will see this project and think about their local church, a school nearby, or even a friend or family member whose days--or ministry--could be made much simpler with a program like this.

What I learned

The journey of creating Choir Maker was a huge learning experience for me. Throughout the project, there were several particular lessons that I learned, whether from mistakes, or simply from lack of knowledge, that have shaped the way I view programming as a discipline. Several of these may overlap somewhat, but that just goes to show the importance of each.

The first, and perhaps the most influential lesson I learned regarding the way I think about programming is best summed up in two sayings, “How do you eat an elephant? One bite at a time,” and “a journey of a thousand miles begins with a single step.” In the beginning of my work on the project, all I could see was the end goal. This caused me to waste a lot of time and effort trying to build good user interfaces for idealized programs instead of using the resources I had to build an effective program with potential for addition of a GUI and a plethora of other features in the future.

Another lesson I learned during this project is not to be scared to ask for help. There were many times throughout the process when asking for help would have saved both time and effort. With the input of others I may have been able to sidestep the fixation with creating a beautiful GUI. It is also possible that with the insight of others, I may have been able to add more features, or even simply make the existing features more robust. There is only so much that one person can accomplish by himself, and this project taught me a lot about what my own limits are.

A more practical lesson that I learned during my time writing and developing Choir Maker was that programming often takes much longer than is originally estimated. Of course, this difficulty is not unique to me. It is the source of many discussions and jokes within the programming community.[10] One website even has a chart detailing why various time estimates given by programmers will inevitably be incorrect.[11] In addition, time estimation was a significant topic throughout my class on Software Engineering. Because I knew it would likely be a very time consuming project, I began work within a

[10] "The Art of AgileSM." James Shore: The Art of Agile Development: Estimating. April 23, 2010. http://www.jamesshore.com/Agile-Book/estimating.html.; "How to Respond When You Are Asked for an Estimate?" Development Process. September 11, 2011. http://programmers.stackexchange.com/questions/648/how-to-respond-when-you-are-asked-for-an-estimate.; "Posts Containing 'estimate Time'." StackExchange. http://programmers.stackexchange.com/search?q=estimate time.

[11] "Programmer Time Translation Table." Passion for Coding. June 7, 2012. https://coding.abel.nu/2012/06/programmer-time-translation-table/.

couple days of my first meeting with Dr. Finch. But even though I started almost immediately, I ran into many problems for which I had no contingency plan. Through the complications that followed, I learned the importance of being able to accurately estimate the amount of time that a given project will take. It is not an easy task, but it is something that I will put much more effort into in the future.

As part of the previous lesson, I also learned the importance of planning. Throughout the project I spent some time planning, but never very thoroughly. If I had developed more thorough plans in the beginning, and perhaps even come up with a sort of outline or development plan for the project, then the actual production would likely have gone more smoothly.

Finally, the most important lesson that I learned throughout the adventure of producing Choir Maker was humility. Every time I felt like I was getting close to a beautiful product, something would get in the way. Whether it was a well-hidden bug, or simply that the structure for the program that I had in place was ineffective for what I wanted to accomplish, or even that I overestimated my familiarity with a particular language, this project served as a continual reminder that I am only one man. And no matter how good of a programmer I become, I will always have need of others, and I will always have need of God.

Bibliography

"Gartner: Open Source Software Gives Businesses Flexibility." ComputerWeekly. February 09, 2011. http://www.computerweekly.com/news/1280095102/Gartner-open-source-software-gives-businesses-flexibility.

Hoelsema, Scott. Food Pantry Manager. Covenant College. Covenant.edu. April 13, 2015. https://huss.covenant.edu/sip/pdf/2015_Hoelsema_Scott.pdf.

"The Art of AgileSM." James Shore: The Art of Agile Development: Estimating. April 23, 2010. http://www.jamesshore.com/Agile-Book/estimating.html.

"Knuth: Computer Programming as an Art." Knuth: Computer Programming as an Art. December, 1974. http://www.paulgraham.com/knuth.html.

Leibee, Weston. "Technology on the Mission Field." Charity Christian Missions. October, 2007 http://charitychristianfellowship.org/sites/default/files/mission-newsletter-archives/2007-10-technology.pdf.

"Programmer Time Translation Table." Passion for Coding. June 7, 2012. https://coding.abel.nu/2012/06/programmer-time-translation-table/.

"Posts Containing 'estimate Time'." StackExchange. http://programmers.stackexchange.com/search?q=estimate time.

Wadhwa, Vivec. "Benefits of Crowdsourcing." The Wallstreet Journal. October 29, 2014. www.wsj.com.

"How to Respond When You Are Asked for an Estimate?" Development Process. September 11, 2011. http://programmers.stackexchange.com/questions/648/how-to-respond-when-you-are-asked-for-an-estimate.

"Point2 Homes." East Lake Demographics & Statistics. http://www.point2homes.com/US/Neighborhood/TN/Chattanooga/East-Lake-Demographics.html.

THE HOLY BIBLE, NEW INTERNATIONAL VERSION®, NIV® Copyright © 1973, 1978, 1984, 2011 by Biblica, Inc.™ Used by permission. All rights reserved worldwide.