Music Software Design IQP

Brian Seney
Anthony Sessa
Beth Hankel
Michael Metzler

Advised by Professor Brian Moriarty

Table of Contents

d. *Field Artist*

VIII. Research

a. Methods of research

b. Results

c. Interpretation / Findings

IX. Design Document

a. Title

b. Tagline

c. Summary

d. Assessing the Market

e. Walkthrough

f. Use Description

g. Marketing

h. Implementation

X. Conclusion

a. Discussion of problem statement

b. Discussion of project goals

c. Recommendations

XI. References

XII. Production Details

XIII. Authorship, Acknowledgements

Section I: Abstract

This project was formed to analyze the problems of current drill designing software and determine the specifications for an optimal software solution. Through research into current products and contact with users, three areas of design weakness were identified: usability, realism, and capabilities to help marching students learn. We believe our proposed software design, *PowerDrill*, offers significant improvements to these categories.

Section II: Executive Summary

2a) Introduction

Marching band is a difficult and rewarding activity. Elite bands can rehearse for months to perfect a field show with percussion, flag twirling, and dance routines. These shows are often thematic and contain several musical numbers that must be memorized for performances. Marching band is an activity just as physically demanding as some sports, and in the process, members learn skills such as teamwork, leadership, and self-discipline.

2b) Problem Statement

In the creation of drill, or the written nomenclature for marching shows, drill writers usually use computer programs to complete the challenging job with ease. Several software tools exist on the market today that allow music educators and drill writers to plan marching formations in a clean and efficient way. Unfortunately, current drill designing software programs often fail to provide adequate assistance to the user. The instructors, writers, and amateurs who use these programs are not necessarily computer savvy, and drill writing software programs have a high learning curve, creating frustration for unsophisticated users. Writing drill itself is a very difficult process -- the writer must take into account spacing, collisions, instrumentation, visual design, and emotional impact. Modern drill design software can currently alert the creator to the first two problems, but they do nothing to offer suggestions to improve what is being created. These software packages fail to help the writer organize their ideas in a manner that keeps the show entertaining yet takes into consideration the ability of the members. Furthermore, programs that aim to make rehearsals more efficient are almost non-existent. The software that currently exists has some of these features, but no one drill writing program excels.

2c) Research Methods

Research was conducted to become fluent in drill designing software programs, modern marching bands, and the history of drill writing. Numerous books were consulted, including *The Student's Guide to Marching* by Chris Previc, *The Dynamic Marching Band* by Wayne Marksworth, and *Drill and Ceremonies* by the United States Department of Defense. The marching periodical *Halftime Magazine* was also referenced to attain insight into current marching band topics. Surveys relevant to our topic were circulated to marching band students, directors, and drill writers, and social networking increased the distribution of these surveys, for which we received over one hundred responses. The results from the surveys gave us valuable insight to what marching students and drill writers look for in a polished drill-designing program. In addition, several drill writing software programs were downloaded and evaluated to determine where improvements might be made.

2d) Design Overview

After reviewing the failings of current drill writing software, we determined there are three areas that can be improved upon: usability, realism, and capability to help students learn. We believe our proposed software design offers significant improvement in these categories. The design has many features designed to make the program easy to use, such as intuitive tools and a helpful tutorial to get the user started. The program also functions as a field show simulator, which animates drill sets, displays collisions, and provides an informative interface for users to quickly view problem areas such as collisions, large moves, etc. The program has the ability to synchronize the drill animation to a music file, which gives students a feel for the final product. There are also robust printing tools, including a utility for the printing of student positions, or “dot books”, for each member of the band.

2e) Conclusion

PowerDrill is designed to address the issues and problems modern drill writing programs face. We believe the ideas presented in this paper offer a strong foundation for a powerful, intuitive, useful drill writing software program.

Section III: Project Goals

Our Music Software Design IQP was created to fill what we believe is a current gap in the way drill is written and shared with performers and band leaders. Some of the members of our IQP group have performed in marching bands and have been frustrated with the way the drill is represented when they receive the physical copy of their drill. New software is constantly being developed, but certain aspects are often missing in programs that could be rectified with research into the field and contact with the people who use drill-designing software in some way.

The project goal was to write a complete design document for a piece of drill writing software by asking people who are directly involved with marching band what features it should have. The resulting program would be a software suite, including components for drill writers, directors, and members. Most of these features would take advantage of advances in computer technology, such as the ability to connect the drill to an audio file.

Section IV: Introduction

"What is life itself but a perpetual march?" - John Philip Sousa (84)

4a) What is a Marching Band?

The phrase "marching band" evokes different images for different people. Some might imagine a company of musicians marching in a parade to the rhythm of a Sousa march. Some picture an assembly of men dressed in Scottish kilts, playing fifes, drums and bagpipes, or a group of highly-excited students, playing a fight song to cheer their football team to victory. These activities, though varied, all fall under the broad category of "marching band." Several main types of marching band have evolved, which are described below (Marksworth 8). These categories can, and often do, overlap; for example, a pep band, which plays at football games, may also perform a field show.

- Friday-Night Football Band – These marching bands (sometimes called "pep bands") support their team at sporting events such as football games.

- Parade Band – These marching bands perform mainly in parades, in both competitive and non-competitive settings. Instrumentation can vary from trumpets and trombones to fifes and bagpipes, depending on the style of band.

- Show Band – These bands perform "field shows," a performance on a football field (usually during halftime) in which the members march and play at the same time. Show bands vary widely in marching style, and dancing is often involved. Big Ten schools - large Midwestern colleges in the United States with large, well-funded band programs -

often have large, extravagant show bands to support their athletic teams.

- Music Style – These bands perform field shows as their primary focus, often attending competitions solely for field shows. Their marching style is designed to facilitate playing while moving, and they rehearse often to perfect their music and formations. This paper will focus on the challenges and solutions of writing field shows as it pertains to music style bands, but much of what is stated will also apply to the other types of marching bands.

4b) What, Specifically, is a Field Show?

A field show is a performance that encompasses marching and playing instruments. At the beginning of a performance, the marchers take the field and find their first location. When the drum major, or conductor, gives the signal to begin, the members begin to play and move to their next locations at marked times in the music. Wayne Bailey, in *The Complete Marching Band Resource Manual,* states that: "Marching band shows have evolved into eight- to twelve-minute mini-stage shows complete with elaborate props and staging, dancing, and often singing" (3). These field shows can be very difficult to learn. Each member in the band must memorize between fifty locations for smaller-scale shows, to over two hundred locations for very competitive bands and drum corps. Modern field shows can be very similar to musical theatre productions, only with instruments and on a much larger stage.

4c) Why do People Participate In Marching Band?

Marching band is a very difficult, albeit rewarding activity. Physical demands on marching members can be very high. For wind players, the activity can be as demanding as running, but with restricted breathing. Marching band rehearsals can also be tedious. On average, every week of rehearsal only adds about a minute of performance time to the show. In addition, the activity also requires a high degree of musical competency, which takes years of practice. Despite all the difficulties, students still excitedly flock to the activity. *The Dynamic Marching Band* lists numerous benefits of being in a marching organization, such as, "performance opportunities, musical development, teamwork, leadership, family atmosphere, sense of belonging, self-esteem, stress management, self-discipline, excellence, physical fitness, etc…" (Marksworth 9). Marching band benefits not only musical development, but also personal development. Like many other performing arts, people enjoy perfecting their show for themselves and others. As stated by the great bandmaster John Philip Sousa, "The musician is actuated by a power beyond himself, so fleeting and intangible that it cannot be explained and I can only call it inspiration" (122).

4d.) <u>How-To of Marching Band</u>

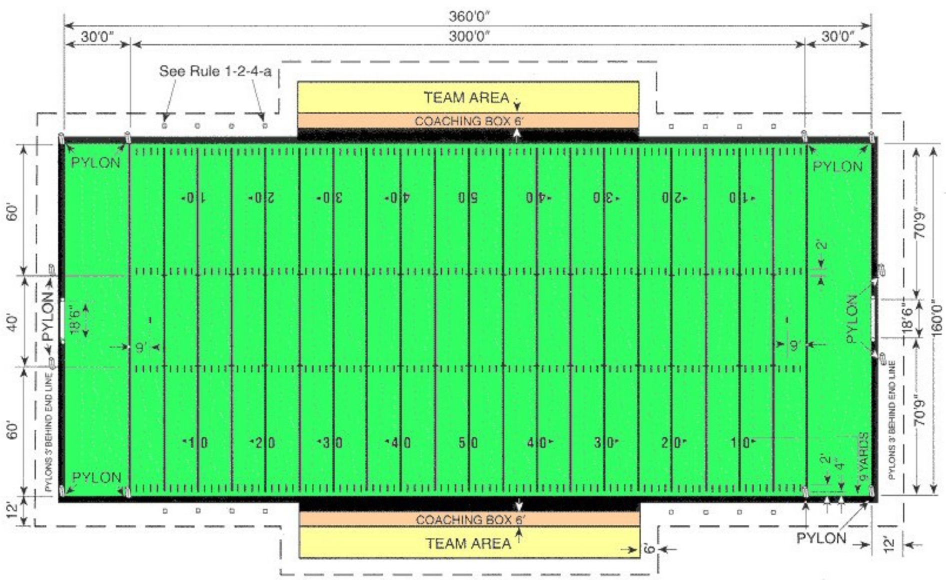
Components

- Marcher: The band's marching member is the core element of the marching band, as he or she bands together with his or her fellow marchers and creates the field show images and shapes. To achieve the precise look of the show, the marcher must use a dot book, or a print out of the two-dimensional top projection of each stopping point in the show, known as a set, which is annotated with each marcher's specific spot relative to the side lines and the yard lines. Each transition is memorized by the marcher, and each set is aligned with the proper musical point. As each move is practiced, less emphasis is put on lining up to physical markers as the need to be accurate relative to the surrounding marchers, making the show seamless, crisp, and impressive.

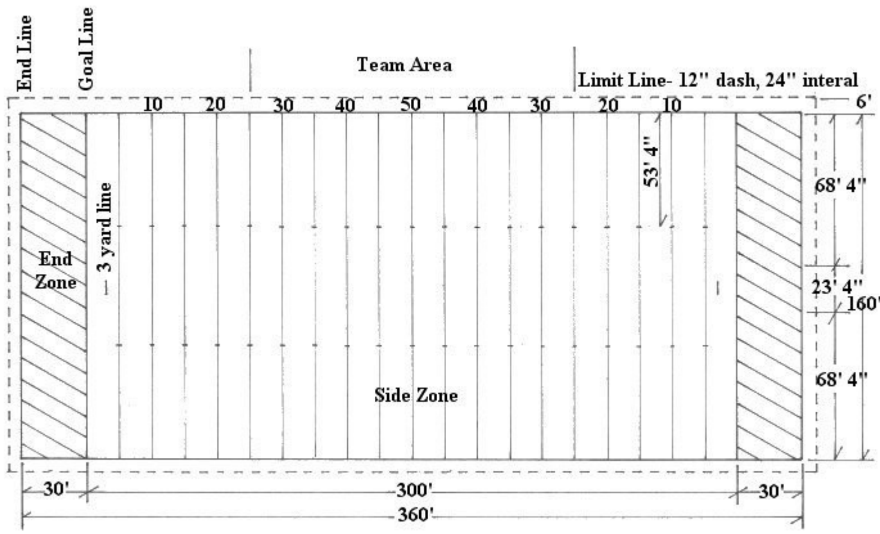
- Drum Major: To conduct a band is to command a well-tuned machine - each part must be guided together, and any lapse in concentration may lead to disaster. The drum major may be the instructor of the band or an assistant to the director. Like traditional band or orchestra conductors, the drum major conducts the band with either a baton, a mace, or just his or her hands, and communicates dynamics in the music through changes in conducting size and motion. For accents and movements, whistles and/or voice commands are used, directing the band which way to move or what action to take, based on the individual command or beat. A jam block, a percussion wood block-like instrument made of plastic, is similarly used during practices for both tempo and commands. The jam block can be easily heard over a band ensemble, making it the perfect tool for synchronizing tempo. Utilizing his or her tools effectively is the mark of a well-trained, confident commander of a marching band.


- Field Staff: Even the most powerful major needs help sometimes. The field staff is often comprised of collegiate music education students or marching enthusiasts, hired by the director, that assist the marching band in rehearsals and performance. Knowing what to look for while being freed of field constraints, the field staff is essential for being an outside analyst of the band. The field staff assists the drum major by making sure the marching members are marked off correctly in reference to the field lines, the sets look accurate, and the sound quality stays consistent in different audience locations. During performances, the staff positions and retrieves color guard equipment and prepares the field for the band, which can include moving football benches and setting up the drum major platform. The field staff can also double-check drum major instructions and provide a second opinion for collision avoidance and practical maneuvering solutions for solid shows.

- Field: Typical marching band shows occur during the halftime break of football games, defining the venue on which all the work occurs - the football field. As shown in these diagrams from AthleticFieldMarker.com, an American college football field measures 120 yards long with field-wide lines every five yards, further subdivided by yard on the sidelines, and 53.33 yards wide with two lines of short hash marks 40 feet apart perpendicular to the yard lines.



High school fields have wider hash marks, marked 53.33 feet away from each other.



Bands will usually be contained between the 20 yard markers and the second set of hash marks, facing the home team stands, but it is not uncommon for a band to span beyond, depending on size. In addition, it is not unusual for a band to leave the field itself and play as close to the audience as possible[6]. For proper band usage, the field may be marked with vertical yard markers as members face towards the sidelines and therefore cannot look down at the field numbers. In addition, the field must be equipped with a bench or platform for both easy drum major view of the band and member view of the drum major.


WPI
40
50
40

Types of Marching

1. Roll Step

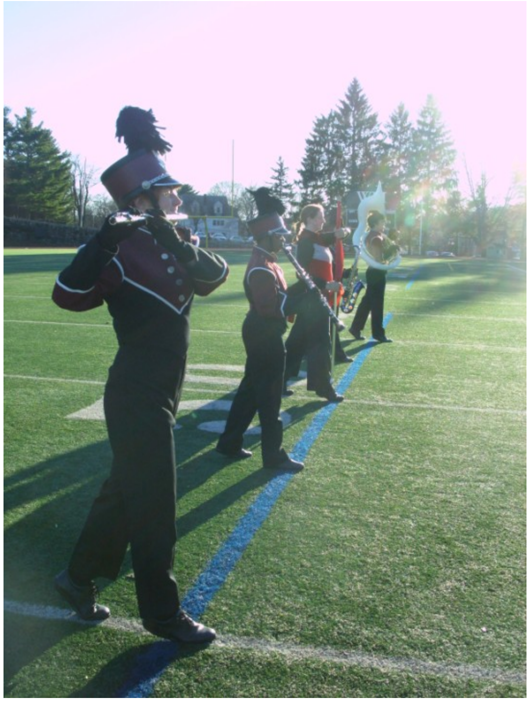
Roll step marching dictates marchers to not raise their knees and to roll their feet in a heel to toe fashion, keeping the upper body completely still. This method of marching emphasizes the fluidity of the ensemble. This marching step is more fashioned for smaller bands as it is less physically demanding.

2. High Step

High knee height is key for high step marching, and the movement requires the toe be the last part off the ground and the first to hit. This method is a favorite of Big Ten and traditional marching bands as it is more flashy and decorative than roll step marching.

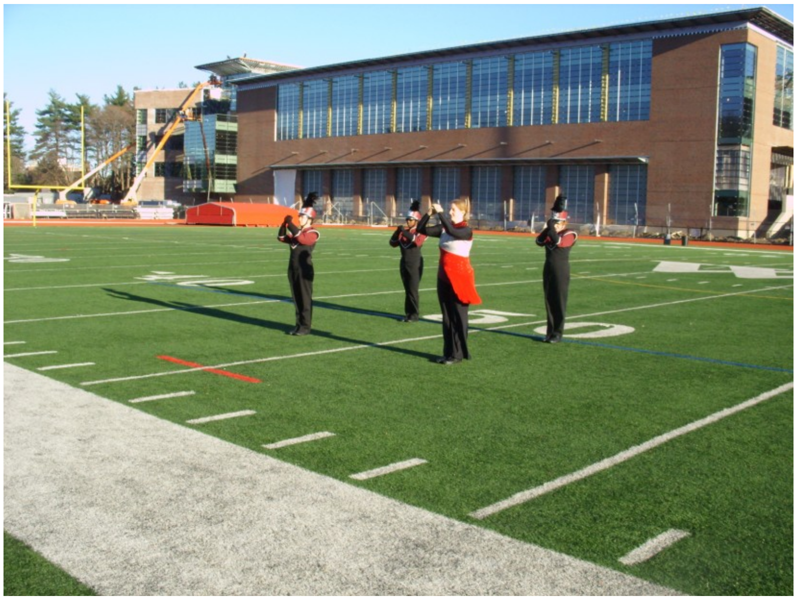
4. Step Size

Uniform step size is crucial for an accurate presentation and a good-looking field show. Standardly, step size will be eight strides every five yards, or 22.5 inches, especially for parade routes and band movements off-field. In performing a show, step size will be varied, but marchers will want to move between set points without variation of length.



Basic Shapes and Movements

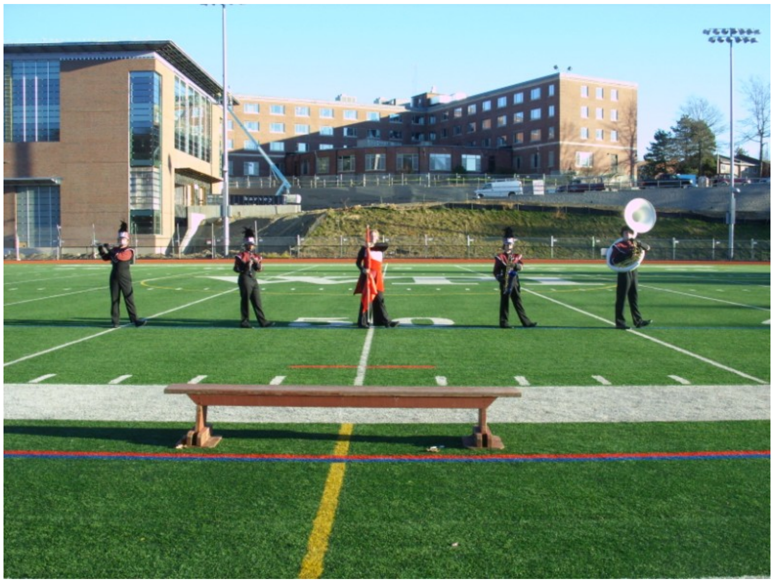
- Block: A block is a formation consisting of several even lines of marching members, lined up in a box. Uneven member numbers fall in to the back of the block. This set up is used for warm up and parade movements due to its simple shape.



- Line: Lines are the most basic, yet demanding, shapes in a marching show. Members must be constantly aware of others to keep lines straight and to keep spacing normalized. The commands “Dress Right” and “Dress Left” are used to address discrepancies when in a horizontal line, but lines become much more difficult to straighten when angled. Members must practice achieving sets accurately to avoid skewed lines.

- Arc: An arc is a curved line formation that can be curved sharply or shallowly, depending on the set. For a successful arc, the performers must be aware of positioning as to not become a point creating a steep angle or a point making a flat side, breaking the uniformity of the shape.

- Shift/Flank: To maneuver around a field without running out of space, marchers must be able to rotate in shifts, flanks, and turns. When shifting, marchers turn their lower body in the direction they want to move while keeping their upper body parallel to the sidelines. Flanking is a complete body turn done in one beat and is used left often in shows, as the instrument bells are no longer facing the audience. Finally, turns are a four beat ninety degree rotation, done by shifting the feet in forty-five degree increments.

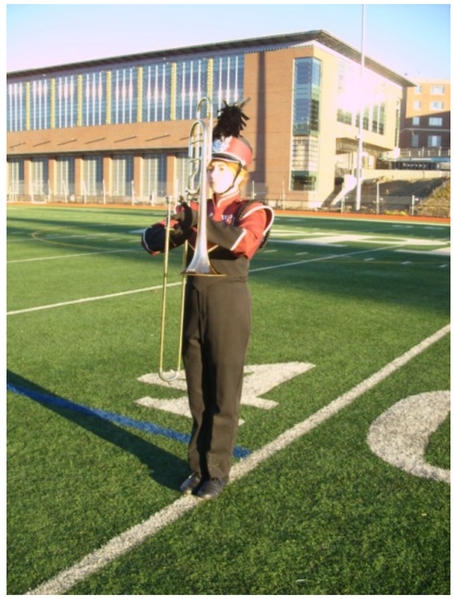


Common Problems

- Tuba: The largest brass instruments are the tuba and its fiberglass marching counterpart, the sousaphone. Both need special consideration within a marching show, as they have a large size radius and have less visibility and maneuverability around other instruments.

- Bass Drum: In the drum line, the bass drums create the low-frequency foundation for the rest of the band. Bass drums have a large range of sizes, from 14" to 32" in the example of drum company Pearl[6], and 28" is standard for a one-bass band. This huge instrument must be taken into account when drill-writing as much space is needed. Typically, bass drummers march with the drum heads parallel to the stands for maximum sound projection, and when combined with the height of the drum, a vision problem is created. Depending on which way the drum line is facing, drill writers must take into account which marchers cross paths with the drums and plan accordingly.

- Trombones: Trombones are physically different than most instruments, utilizing a slide to achieve notes instead of pressing buttons or covering holes. In such way, the slides must be considered for sets.



  - Juxtaposition: Trombone players, when playing music, have the risk of hitting other marchers in the back of the head if placed too closely behind another marching member. However, a trombone player may be able to tilt the instrument up or down to compensate for members in immediate threat of collision.

- Positions: A trombone has seven positions of increasing length, moving down a half step in pitch at each mark, extending up to four feet beyond the marcher. Thus, the music must be considered when placing a trombone player behind any other member.

Section V: History of Drill Writing

Armies around the world have participated in some form of drill for centuries. In the past, drill was not used for the entertainment purposes that it is known for today; its primary purpose was to prepare troops for battle. The "marches" that the troops performed were similar to tactical maneuvers used on the battlefield. They allowed the commanders to move their forces from one point to another with relative ease. By doing this, maximum firepower and efficient battle formations were achieved.

The origin of these practices can be traced to the ancient Romans. (U.S. Marines Corps.) Drill was used to train the Roman soldiers until their movements were second nature, so each would know the whole battle formation at any given time. Through the use of drill, the Romans were able to form moving walls of spears and swords that could defeat any opponent. The soldiers were able to move in unison, and as a result of their training, the Roman Empire was able to conquer a large part of the ancient world.

Drill writing also played a large role in the American Revolutionary War. (U.S. Marines Corps.) The leaders of the thirteen colonies were presented with the problems of not only establishing a government separate from Great Britain but also getting a mobilized army organized in such a way that would cinch victory. From 1775 to 1778, America's army consisted of nothing more than groups of civilians fighting "Indian-style" against highly trained and disciplined English Redcoats. Lack of funds, clothing, rations, and equipment plagued many American forces. As a result, there was little organization, control, teamwork or discipline. With the help of Benjamin Franklin, the American ambassador to France, General George Washington recruited the help of Baron Friedrich von Steuben, a Prussian officer and former employee of Frederick the Great. Upon arriving and seeing the disarray of the American forces, von Steuben spent his nights writing drill and his days teaching a company of about 120 soldiers, selected from thousands. These individuals quickly learned to respond to commands without any hesitation. The drills instilled a sense of alertness, attention to detail, and urgency in these men. As the group mastered the art of drill, confidence in themselves and their weapons began to grow. Eventually, the troop of soldiers departed, passing on their knowledge to the rest of the army. The overall effectiveness and efficiency of the American army grew immensely. To ensure the continuation of this growth, von Steuben wrote the first field manual for the army in 1779, commonly referred to as the Blue Book. The drill that was written back in 1775 was used for almost ninety years until the American Civil War. The same sense of alertness, attention to detail, and urgency has lived on into the present day military as well. (U.S. Marines Corps.)

Section VI: Background

6a) The Difficulties of Learning Drill

Drill takes many repetitions to learn properly. Large, exposed, linear forms such as blocks or lines take special care, as a single member out of place makes the entire form look sloppy. In softer, curvilinear forms, members must still keep exact spacing, or undesirable "bunches" form. These difficulties are exacerbated by the fact that members must stay at "attention" and cannot look side to side to check the form. Despite these difficulties, a skillful drill designer can create drill that is both easy to learn and impressive to watch.

6b) The Difficulties of Designing Drill

Drill designers have a lot of their mind when creating formations. Sections need to be kept together if possible; this increases confidence and keeps the sound concise and strong. Staging also needs to be considered. For example, if the percussion section is in the back of the field during a "drum break," when only the percussion section is playing, the impact will be diminished.

The band should also be kept as close as possible to eliminate phasing. Phasing, the undesirable "shearing" of musical phrases owing to the speed of sound, can become a problem if the band is spread out. The speed of sound is about 375 yards per second. It is not uncommon for larger bands to have members at both of the 10 yard lines - 80 yards apart. At that distance, it takes more than a fifth of a second for the sound to travel from one end to the other. At an average marching tempo, 120 beats per minute, a fifth of a second delay puts the players almost half a beat behind, which can be very disconcerting to the audience.

Certain sections of the band also have special considerations; for example, marching tubas generally cannot see to their left because of the size of their instrument, and color guard flag bearers need to be kept a certain distance away from others to prevent dangerous collisions. In addition, drill designers need to ensure step sizes are not too large, as shorter marchers will be unable to cover the distance. A competent drill writer must keep in mind all of these considerations, as well as create forms that are visually appealing, artistic, and synchronized with the music.

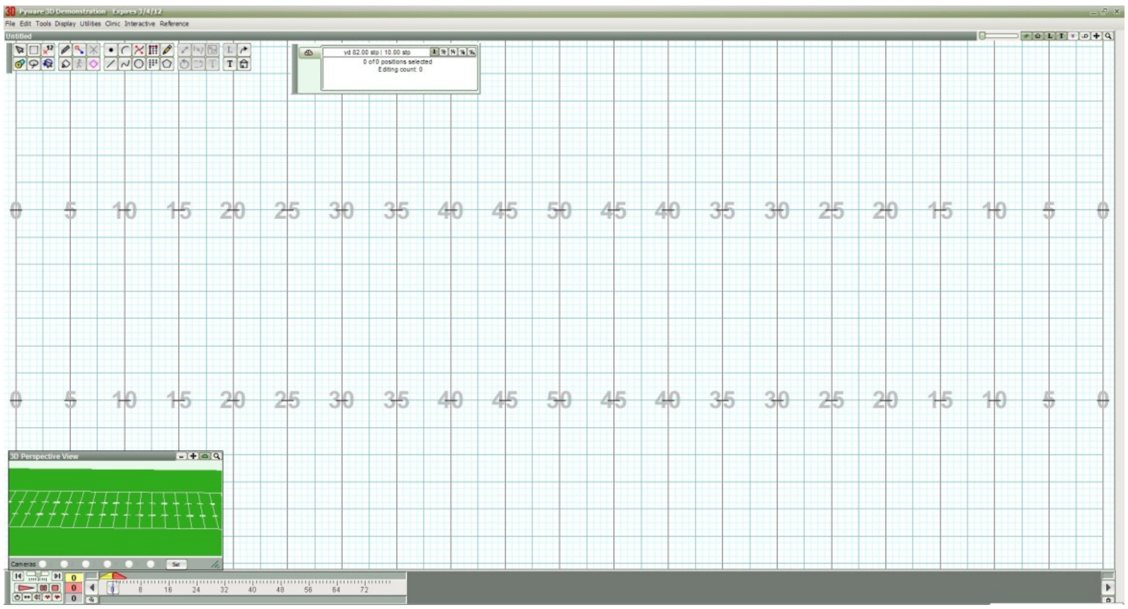
6c) A Potential Solution

The constraints drill designers must conform to are easily computable by software. For example, it is trivial to write a program that will determine no two members occupy the same space at the same time. In this way, a computer program working together with a human drill writer can greatly facilitate the creation of marching formations. The drill designer would focus on aesthetics, staging, and presentation, and the program would automatically determine collisions, phasing issues, and other concerns. A drill writing computer program can also assist in student learning of drill. A program could show a 2D or 3D animation of the completed field show based on the coordinates that would be 100% precise. This animation could then be viewed by students to impart an overall idea of the flow of the drill. Different camera angles could also be utilized to display the show from behind a given marcher. Other potential learning benefits to drill writing software include a dot book printer, as well as synchronizing formations to a music track to create a full audio-visual experience. A full investigation of the potential benefits of drill writing software can be found in the Design Document section of this paper.

Section VII: Evaluation of Existing Software

7a) *Pyware 3D*

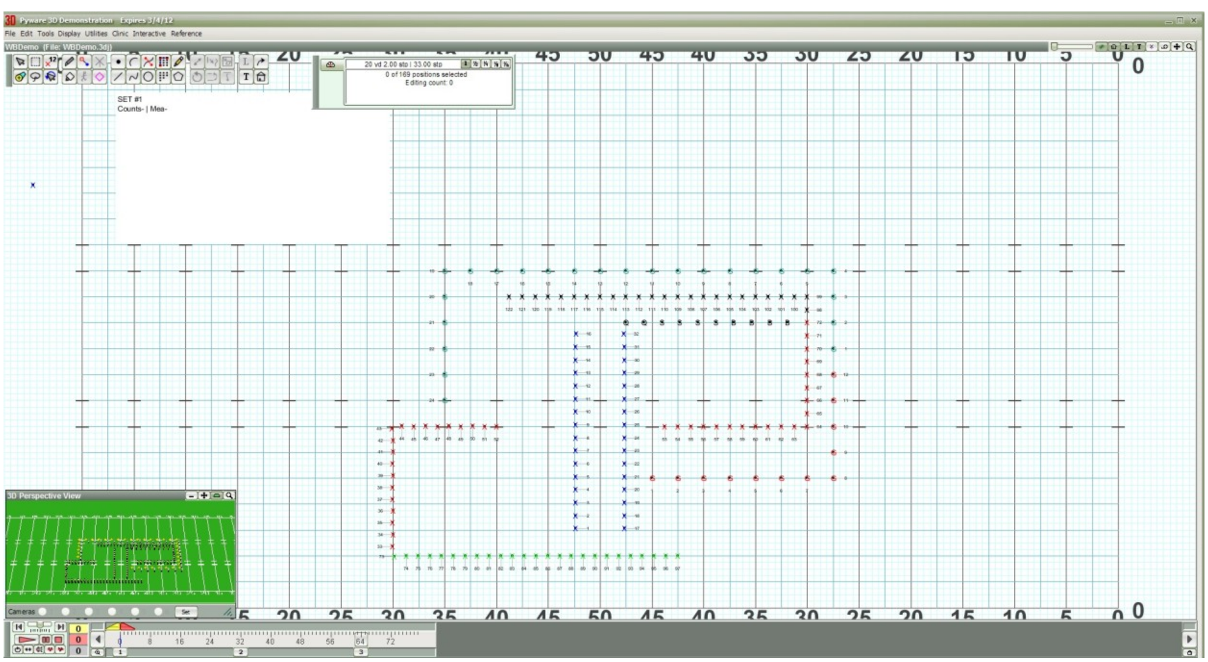
--Pygraphics, Starting at $349



Pyware 3D is the leading drill writing software on the current market. In 1990, Pygraphics produced *Music Writer 1.0*, a music composition program comparable to music writing programs such as *Finale* and *Encore* (Cummings). While its composing software was adequate at the time, it failed to make an impact on the market. Pygraphics carried on to create *Pyware 3D*, a drill designing software program. In its first iteration, *Pyware 3D* was simple, with emphasis placed upon a need to print "dot books", as shown in the second version's update manual, as well as an online file manager for simplified communication between drill writer and the performers (Pygraphics). *Pyware 3D* is now in its sixth version.

History

Pygraphics improves its programs with each update, including their highly prized Perspective View, a three-dimensional representation of the field with drill animation. Having a history of being very friendly to their customers since their first program, *Music Writer 1.0* (Cummings), Pygraphics is famous for their attention to their users, clearly shown through their extensive list of updates to the program, many of which were mentioned in the distributed surveys. Additionally, Pygraphics offers *Performer's Practice Tools (PPT)*, a set of programs aimed to help students and performers learn drill faster and more easily. *PPT* allows for specific member analysis of the drill as well as a game-like leaderboard for best drill practice scores on a *Dance Dance Revolution*-like arrow mat. In its most recent generation, Pygraphics has tackled the challenge of a changing technological landscape by introducing *Music Writer Touch*, an innovative music writing program designed for touchscreens and interactive whiteboards (IWBs). As tablets and IWBs become more prevalent in music education and performance, Pygraphics seized the opportunity to establish themselves as the premiere touchscreen software. For now, Pygraphics seems to only be focused on music notation for touchscreen implements, but the move towards interactive screen drill writing is an obvious avenue to be pursued. While technology is changing, Pygraphics will continue to innovate with the help of their customers.

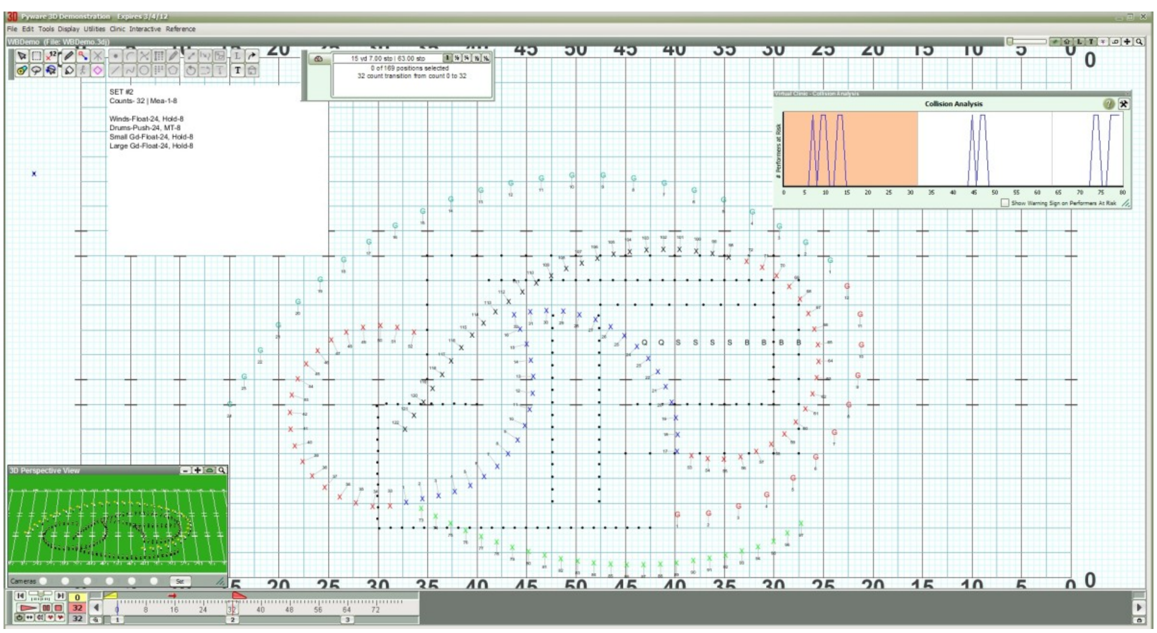


User Interface

Pyware 3D uses one main window, which displays the field where the user is creating the drill. It has a menu at the top and contains panels that can be moved around inside of the window. At the bottom of the window are tabs to select the current set, as well as controls for playing back the drill sets. A free-floating window also exists: the 3D Perspective View. The first time any option is clicked, a window appears and explains what the choice just made was and directions on how to use it.

Menus

Most of the menu choices create new windows related to the task selected. Some of the menus are redundant, such as the Tools menu, an exact content copy of the floating toolbar. The same factor applies with the Display menu, repeating options. The utilities bring up new windows which act as their own programs to run on the drill, generally affecting the whole drill and its properties. The Clinic menus bring up panels that display information about the individual sets and the transitions between them.



Toolbar

The toolbar is very straightforward, with each tool having a unique, distinct icon. Tool names appear when the mouse hovers over the options. They are put into four groups, keeping like functions together. The first section of toolbar is for selection tools, while the second group manipulates shapes already placed on the field. The third is for new shape creation, and the last section is for special functions, such as placing text. There is no ambiguity of what each tool does, and each has its own discrete function.

3D Perspective View

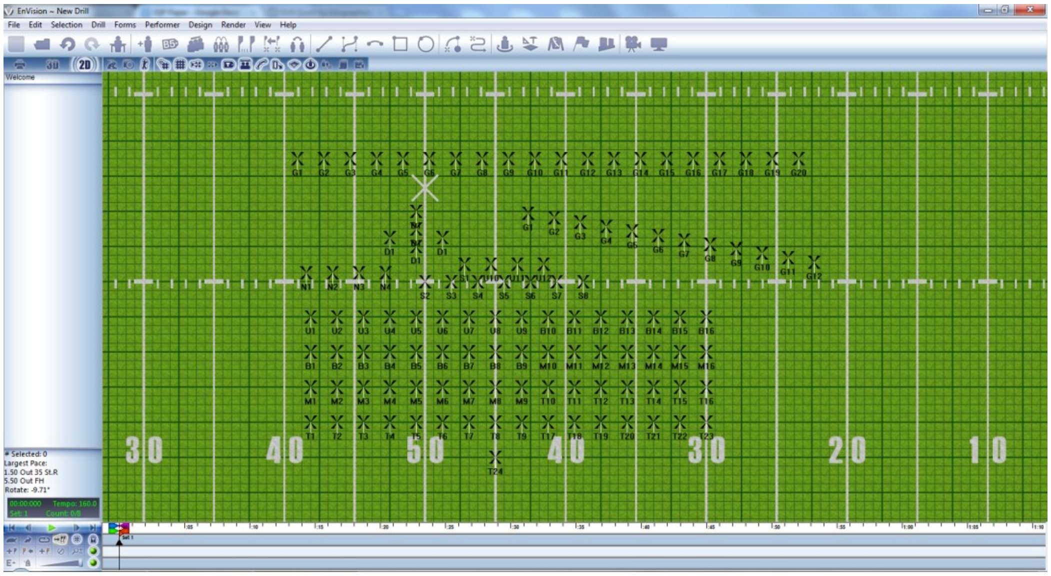
This window allows the user to see the drill as performed from different angles with 3D models. The cameras can be set for quick jumping to various angles.

Statistics, Image, and Rhetoric

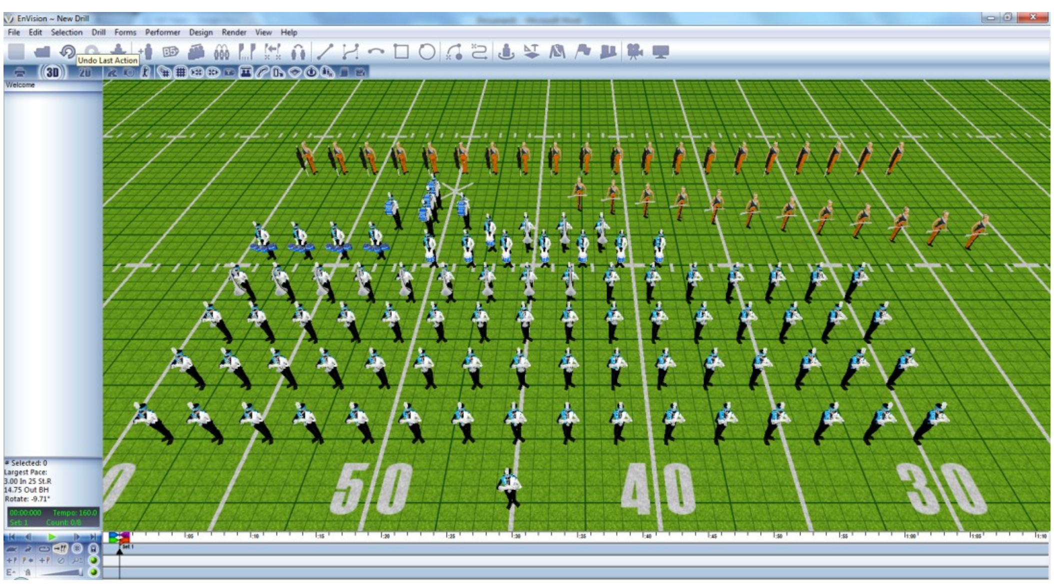
Pyware's current audience is drill writers, from high school band directors to professional drill creators. The company presents its product as the professional yet affordable solution to all drill writing needs, with numerous text quotes expressing satisfaction and a few example drill screenshots. The layout of the program is fairly simple; however, the tools are not as beginner-friendly as one might hope. This aspect discourages amateur drill writers without proper training. Three brackets of software packages exist: Basic, Interactive, and Professional. While Basic is extremely affordable, many useful features are saved specifically for the Interactive and Professional levels.

7b) _*Envision*_

-- Box5, Starting at $449.99



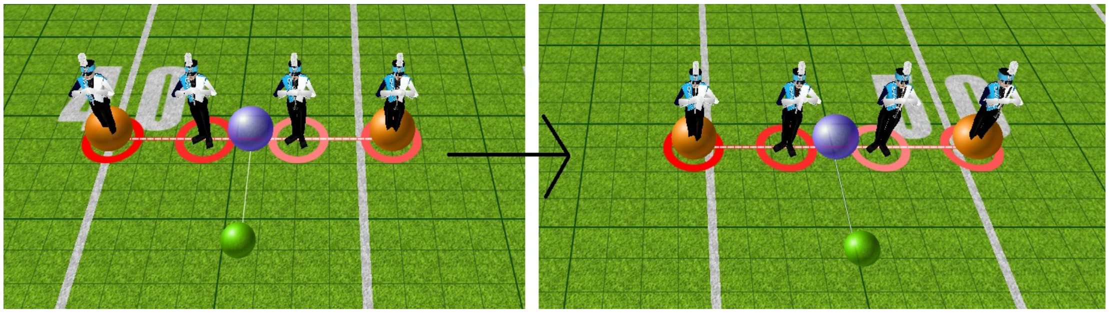
Box5 Software was founded in 2010. The company aims to create a splash in the drill design software industry with a software program called *Envision*. As mentioned on the company website, *Envision* is a "user-friendly, 3D visual performance design software for educators and designers in the performance arts industry" (Box5 Software). *Envision* attempts to be more intuitive for beginning drill writers to use.



As mentioned in the research section, *Pyware* can be very difficult to learn. *Envision* has many features that help the beginning drill designer to learn the ropes of drill writing. For example:

- Trial version of software for beginners to learn / critique.
- New Show creator with intuitive menus for inputting show title, band composition, music files, wardrobe, and field settings.
- Real-time animation in both 2D and 3D.
- Camera controls, including ability to follow a performer through an animation.
- Tools for automatically creating blocks, lines, curves, circles, and arcs.
- Capability for follow-the-leader / curved paths.
- Capability for upper-body rotation.
- Easy printing, along with support for adding notes to drill sets.
- Checks for largest move in a set.
- Capability to render a drill animation to a video file.

Several of these characteristics were requested in the surveys, such as easy printing and real time animation. The interface is very easy-to-use; but some tools are not intuitive. In specific, the Multi Curve Tool uses different controls from the similar Arc Tool and Circle Tool, becoming confusing to beginners. The program has a “snap-to-grid” option, a very useful tool that ensures each member is on a precise division of the field. In certain cases, however, this tool is ineffective. If a number of performers are selected, it is possible to move them as a group around the average location.



I want to move these performers to the right five yards, so I select them as a group. Each performer starts on the grid. When I move them over, the ***average*** *location is on the grid; each individual performer is off the grid. This effect is counter-intuitive.*

If an even number is selected, the average may not be on the grid, and when moving the group, the average location will snap to the grid. This quirk produces the counter-intuitive effect of putting each performer off the grid so the average falls on a grid line, which is not useful for any of the marchers involved. Overall, despite these bugs, the software is intuitive to use effectively.

Although *Envision* has many strong points, it falters in a few key areas. The software has the capability to determine step size dynamically, but it does not have relevant data to report to the drill designer. *Envision* states the largest move per set, but it does not say if the longest move is possible or not, such as if the entire band is moving at a snail's pace, so the longest move is trivial. In testing the software, it was possible to create a marcher who could cross the entire football field in seconds. *Envision* did not throw an error (or even a warning!) to say it is unlikely a high school marcher could break the 100 meter sprint world record. In addition, *Envision* will happily animate collisions and will not warn the drill designer when two marchers are too close together. Seasoned drill designers will intuitively avoid collisions, but it may be difficult for a beginner to visualize them, especially with large bands. *Envision's* animation controls allow drill writers to see where performers are between sets and view collisions, but it would be easier for the program to notify the user of them as well, as these constraints are easily computable.

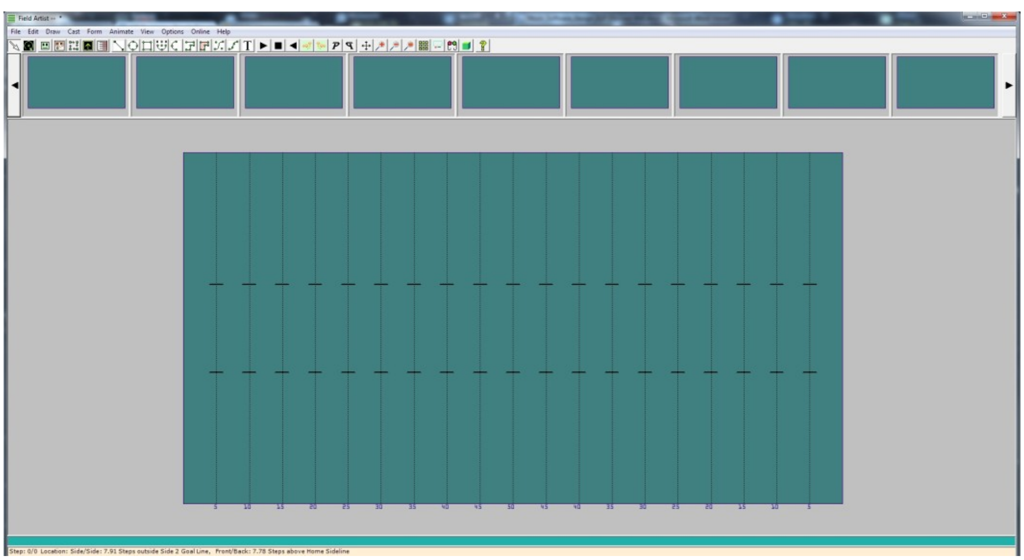
Envision has an abundance of useful features that experienced drill writers can utilize. Its wide array of "visualizing tools," such as real-time animation, 3D perspective, body/equipment positioning, and others help to give an impression of what the final presentation will look like. Despite these features, *Envision* could use a more substantial statistical analysis feature, as well as improvements to user interface and controls. Overall, *Envision's* shortcomings are minor against the many things it does right. With a few improvements, the program could certainly become a contender in the marching band software race.

7c) *Drill Quest*

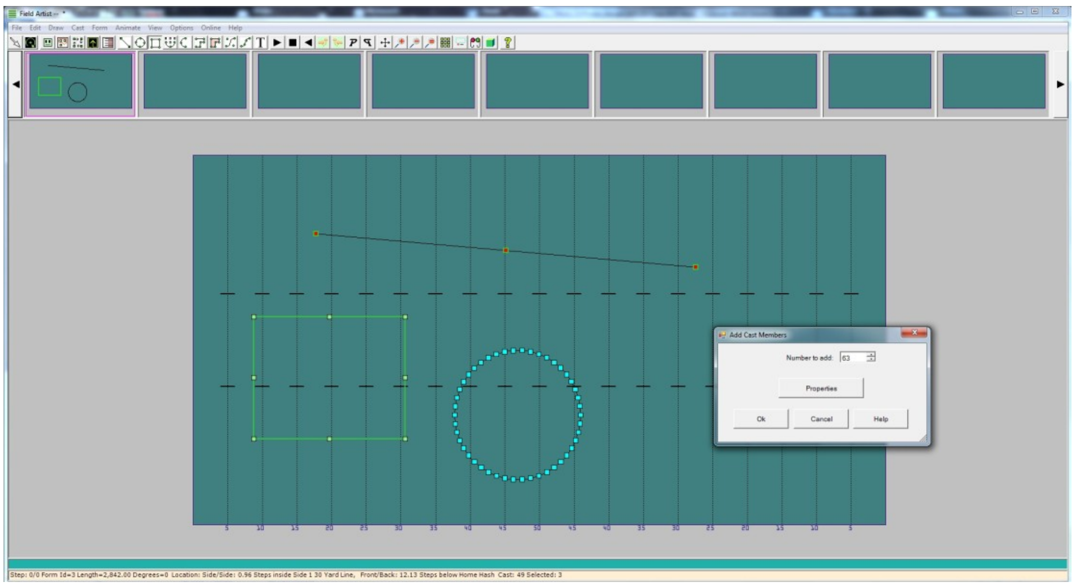
Drill Quest was a drill writing program developed by Edugraphics in the 1990s. It has since been discontinued and as a result, the group was unable to perform any in-depth analysis on the software.

7d) *Field Artist*

--Raven Labs, Starting at $135



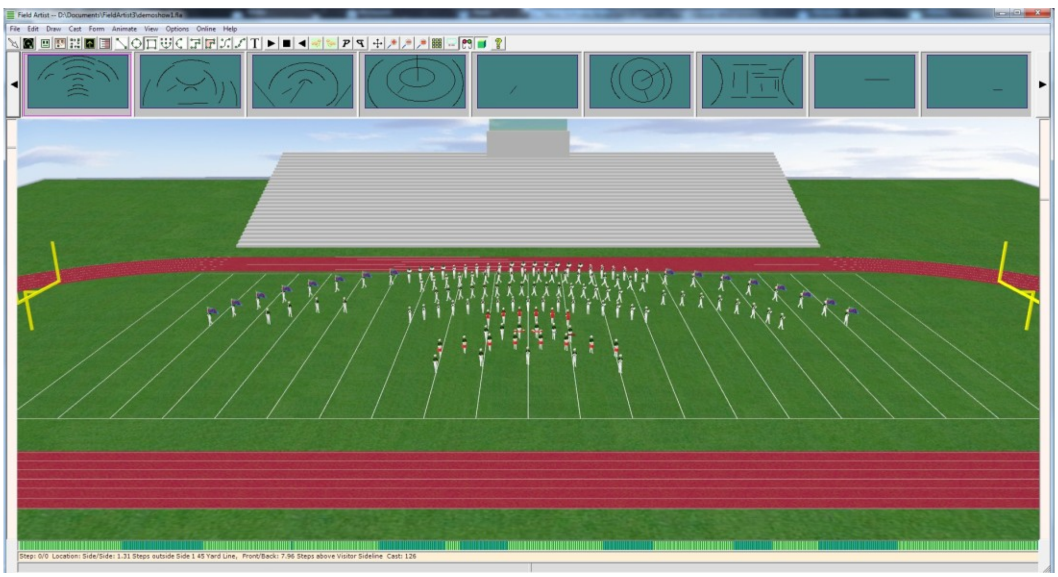
Field Artist is a current drill design software on the market. In 1999, Raven Labs released the first iteration of the product for an introductory price, keeping in mind they were making a product for teachers and students. They believed that being able to see the show they made in 3D would be beneficial to the users of the software, and in 2008 they added lifelike 3D animations. *Field Artist* is now in its 3rd version. (Field Artist Central)



The entire interface is designed around the 3D animation feature. At the top of the window, the Menu Bar sits above both the Toolbar, and more importantly, the TimeLine. Other programs may have similar timelines, but the program wants you to move shapes of performers around to make a field position, instead of making a new position from scratch. The main view can be turned from top down, to perspective, and to full 3D, switching out the current view. Each step of the drill is fully editable in each of the views. It is also very easy to watch the animations, as well as move one step of the drill at a time. During the animations, the cast leaves dashed lines to show how their position changed from the previous set.

Making new forms is about taking the current shape and splitting off different members to create new shapes instead of adding new cast members in new shapes. This feature ensures that the correct number of cast members is on the field and that each player marker has a place it moved from and will move to in the future. The split can be done manually or can be separated by information such as instrument section, such as High Winds.

The 3D view is the highlight of the product. Each band member representation can be given their own uniforms and colors, allowing a truer representation of what the field show should look like when performed. The animations allow for the designer to more easily see where problem would arise with members marching into other members. The software allows for visual effects like follow the leader, as opposed to always facing front, which has a dramatic effect on how the audience sees that move. There is also an easy way to show Color Guard choreography in the 3D view.



This program also allows the syncing of music to the drill, and it uses Windows Media Player to do so instead of their own built-in solution. There does not seem to be a way to make a dot book easily within the program, but they can always be made by hand. The 3D animation can be exported to a video for easy viewing and transfer.

This program made the user interface different from the user interface in other drill design software. The emphasis on where the performers are in between the marked sets make it easier to follow the show as a whole. It does not have some of the analytical tools found in the other software, but because the writer can follow the show more easily, these tools are not really needed in this software.

Section VIII - Research Methods

In order to create a useful drill-writing program, research was conducted to become literate in drill and software applications as well as to be able to facilitate accurate and appropriate software design. Four main source categories allowed sufficient information to create the design document: surveys of bands, relevant books and periodicals on the subject, and personal marching and drill writing experience. A set of surveys was implemented to gather statistics from potential users, focusing on what drill writing and use aspects marching members, drill writers, and directors reported were most important. Our books pertained the following subjects: drill and organized marching history, how to learn drill, drill writing, arranging techniques, social aspects of marching, and marching competitions. Being a team of marching members, our own marching experience as well as exploration into the different marching programs gave a practical facet to the research, allowing a rounded grasp of the subject.

Literature on drill and marching bands consisted of three distinct categories: marching history, drill instruction, as well as drill writing and arranging. To have a good background in our subject matter, books such as *Keeping Together In Time* by William McNeill and *Drill and Ceremonies* by the United States Army were invaluable to learn how and why drill came to be. We also included John Philip Sousa's *Marching Along*, which gave insight into a famous marching music composer and expanded our core knowledge. This information helped our fundamental understanding of what a drill writer must include in the creation for a successful show. Other books, including the same *Drill and Ceremonies* and *The Student's Guide To Marching*, taught basics of marching form: how the interpretation of visual dots on paper turn into the kinetic movements of a great marching show. These books provided literacy of the medium so drill nomenclature could be reproduced. Finally, drill-writing literature was needed.

The Dynamic Marching Band by Wayne Marksworth was one example, revealing what creative tools would be called upon by a professional, thus establishing required equivalents in our hypothetical software.

Beyond these historical and technical accounts of drill, knowledge of more current marching aspects was lacking. To remedy the gap, a well-known marching band periodical, *Halftime Magazine*, was referenced. *Halftime* contains a wealth of information pertaining to recent competitions as well as professional band successes and social trends. Absorbing a more contemporary drill angle allowed for the adaptation of newer needs and technology.

The surveys, distributed through SurveyMonkey.com, consisted of two separate questionnaires: one for all three parts of our target audience and the other only for drill writers. In our inclusive survey, 98 people answered various questions about their demographics and what they would like to see in a hypothetical new drill-writing program. 72 of the responders described themselves as marching members or performers, and one response was from a declared drill writer.

The marching-member-heavy results showed most marchers looked for synchronizing music to drill, being able to export videos and show animations of the drill, and checking for collisions on the field. Twenty responses were received in the second survey which gave direct insight to what drill writers need most to facilitate intuitive and creative drill creation. Overall, the drill writers survey reported that exporting to a usable video format would be very useful as well as more customizable tools in creating drill, such as editing the marching field and easily changing symbols. Between the two surveys, many marching members suggested collision detection with differently sized instruments to make learning drill smoother, but one particular drill writer noted that collision checks and size adjustments were aspects “which competent drill designers should know/do inherently". From these two surveys, we determined we needed to include intuitive controls and useful tools for both beginning and veteran drill writers, improving drill production when handed off to the directors and marching members.

Having a marching basis and researching commercially-available software provided a large amount of research and direction. Coming from a background of marching bands, our knowledge provided a solid foundation for expansion, so we did not have to relearn jargon and knew what qualities made a great marching show. Three of four of the group members were in previous high school and college marching bands and have performed both marching shows and parades, directly interacting with various levels of drill writing. In addition, each team member researched the existing software, downloading trial versions and attempting to learn them in the time allotted. Certain programs, such as *Pyware 3D*, were fairly difficult to master in the trial, and others, such as *Envision*, were geared towards short trial periods.

Section IV - Design Document

9a) Title:

9b) Tagline: “Power and ease in one smooth machine.”

9c) Summary: There are several problems currently facing drill writers and marching bands - the lack of complete and up-to-date information being presented to the drill writer, a high learning curve for new drill writers, and an inadequate number of features designed to assist students learn drill. *PowerDrill* is designed to provide a solution to these problems. Different algorithms and tactics are used to provide information to the drill writer about dangers such as collisions and difficult moves, while the interface is as user-friendly as possible to ease technology-wary users into the art of drill designing. In addition, the software has several tools to make students learn drill more efficiently.

9d) Assessing the Market:

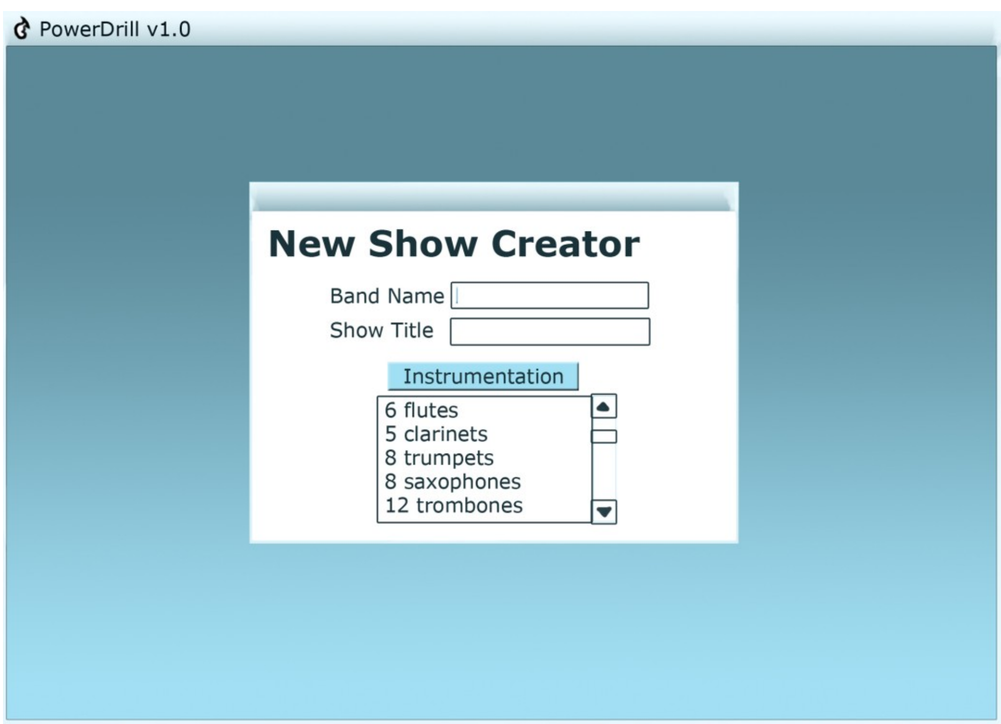
There are multiple pieces of drill writing software on the market today, and each piece has its own strengths and weaknesses. It is herein that lies the problem: there is not one definitive piece of drill writing software out there that includes *everything* the competition does. That's where *PowerDrill* comes in. With an already strong and established foundation thanks to the competition & feedback from a very vocal, passionate, and dedicated community, we were able to compile a list of features an ideal drill writing program would have. *PowerDrill* is designed with not only with the all-important features from each of the competing products, but also the ones that have been missing from those releases and most important to the users as well. To see the full list of features *PowerDrill* will include, look no further than the Use Description section below.

9e) Walkthrough:

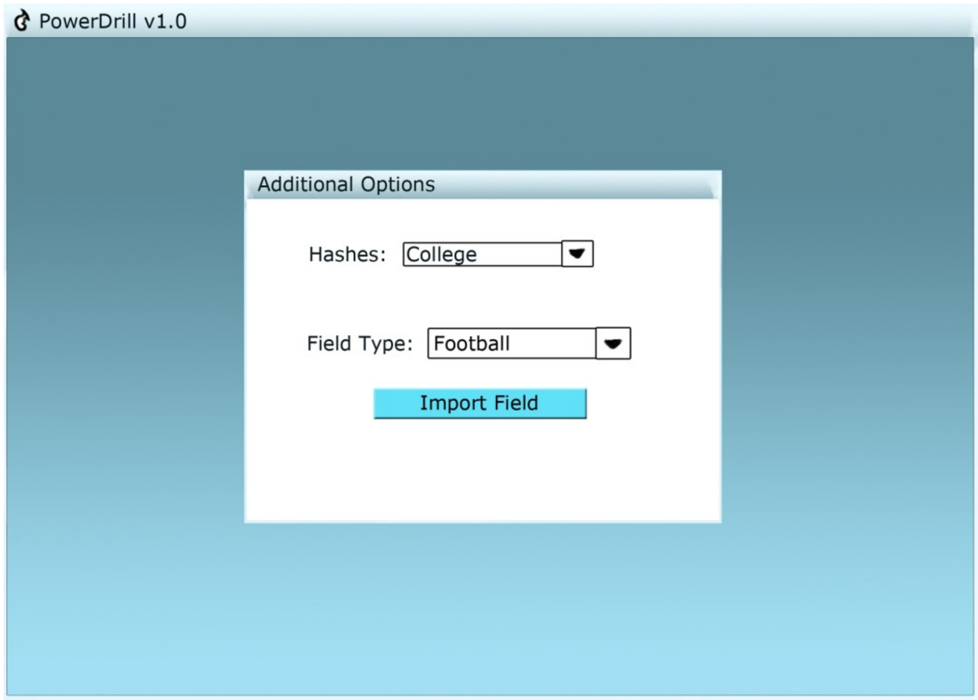
The walkthrough section is a visual user story for how a *PowerDrill* customer might use the software for the first time. It highlights some of the key features of the software as well as demonstrates how the user interacts with the interface.

1. I just got this new program for writing drill: *PowerDrill*! I wonder what it can do.

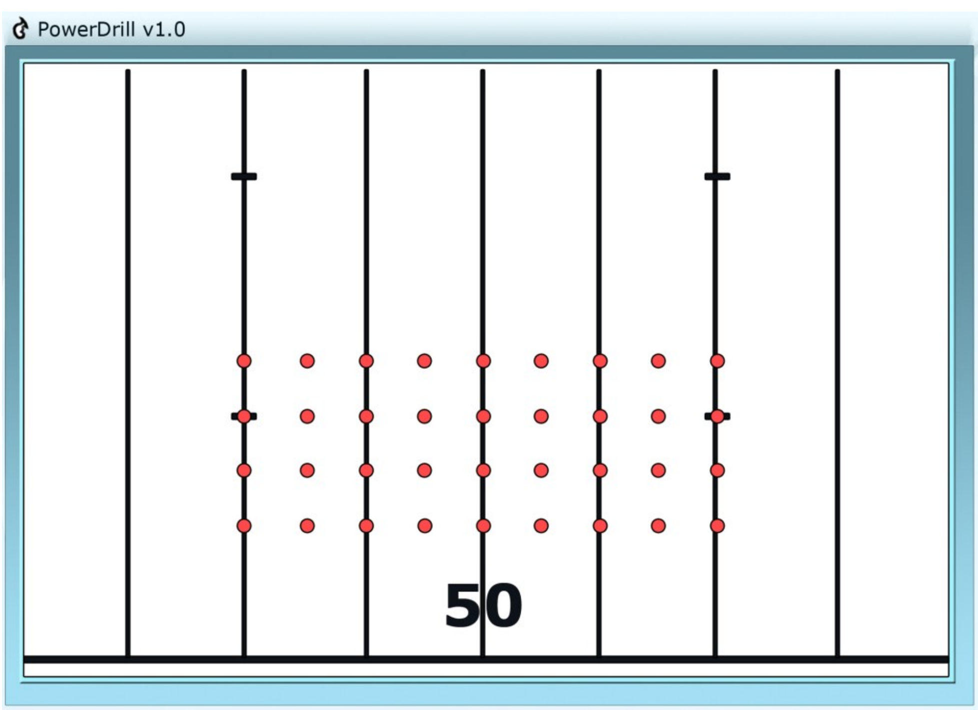
2. *PowerDrill* brings up a New Show Creator when I start the program. I'll enter my band name, the show title, and the instrumentation we have.



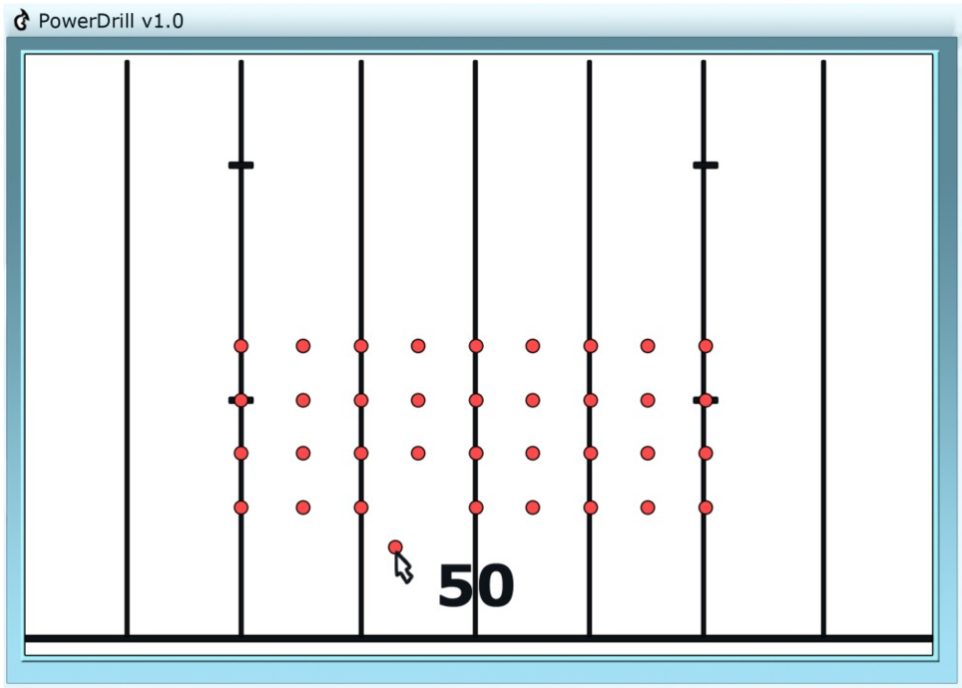
3. I have an option for switching type of field and the markings on the field. This is useful, since fields differ between high school and college marching bands.



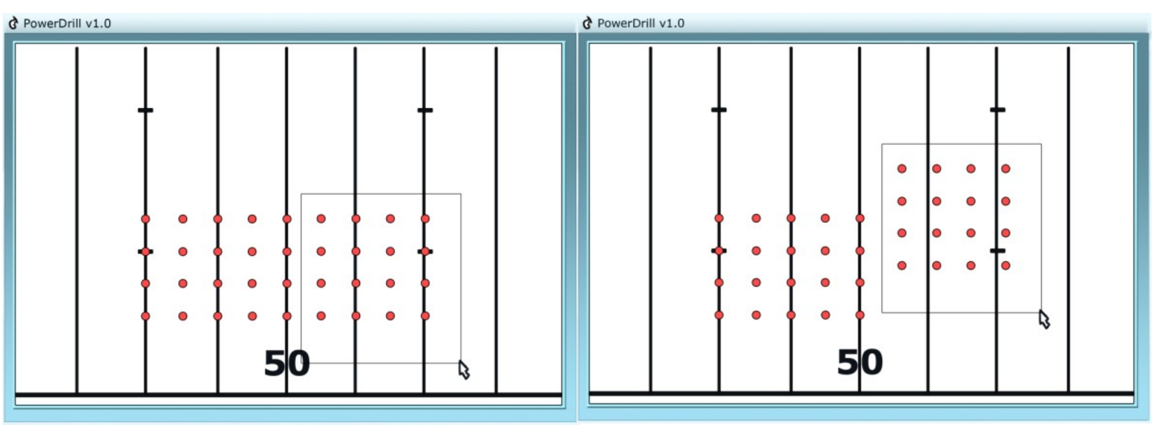
4. Ah, the first set. All the marching members are positioned in a block centered on the field. Now I can get writing!



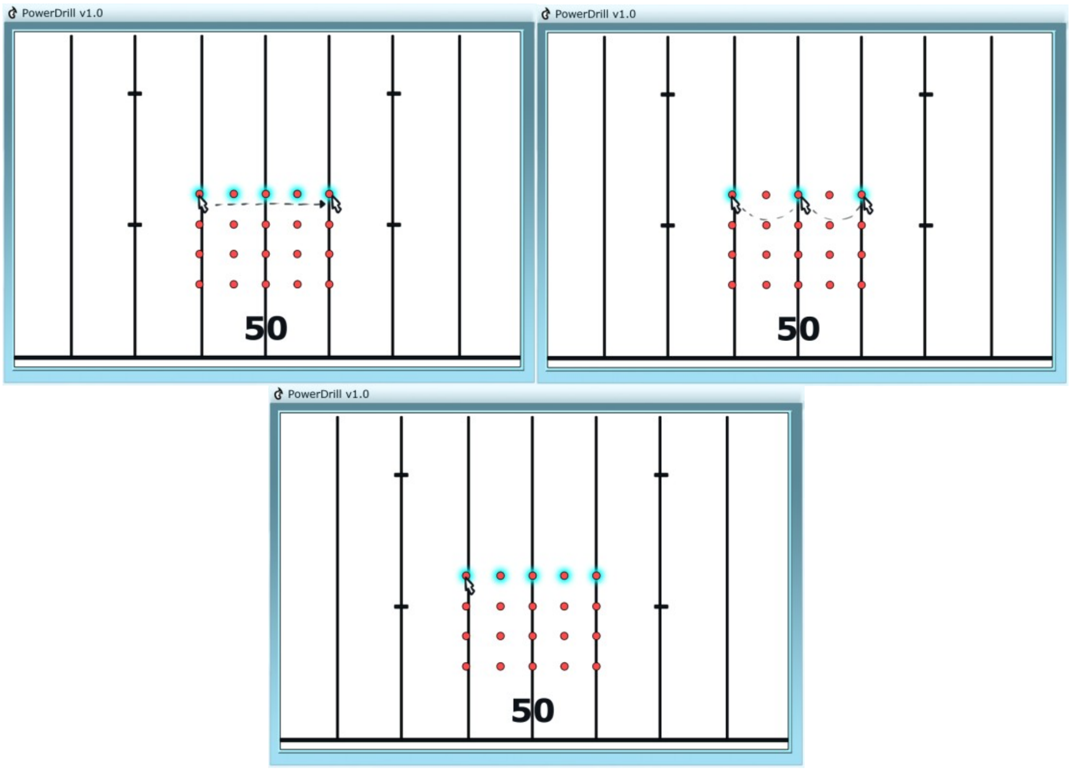
5. To move a performer, I can just click and drag on a marching member. This is pretty intuitive!



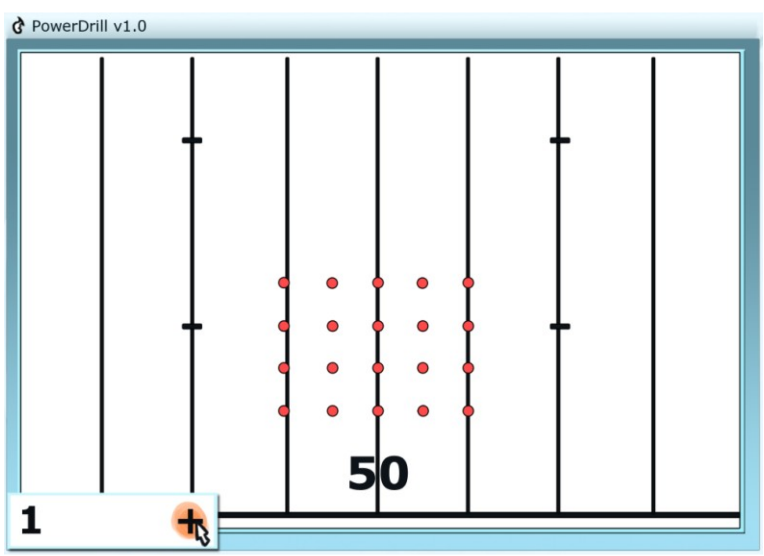
6. If I click and drag, I select a group of performers. I can then click the group to move them all at once.



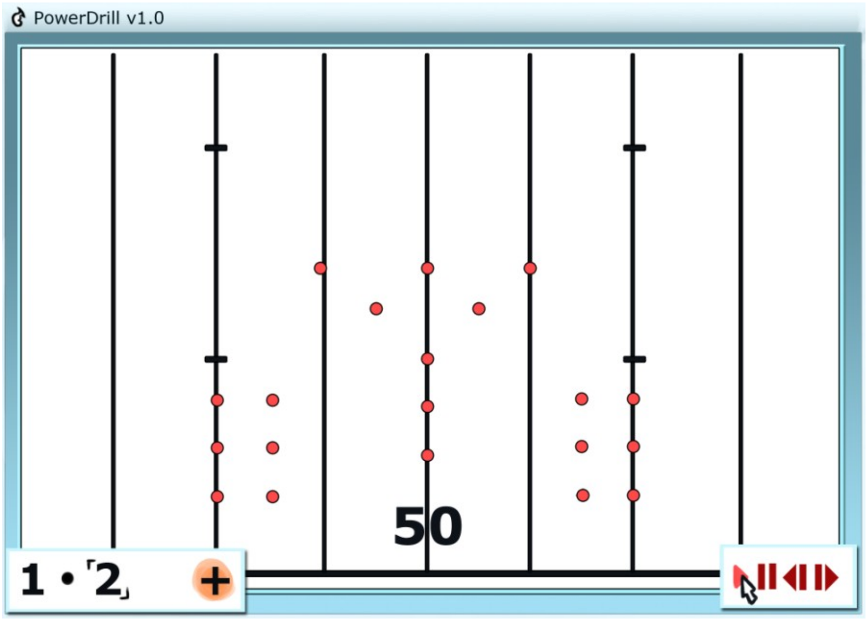
7. Shift clicking two performers selects all members between them. Control clicking selects or deselects individual members. Double clicking selects all performers in the shape.



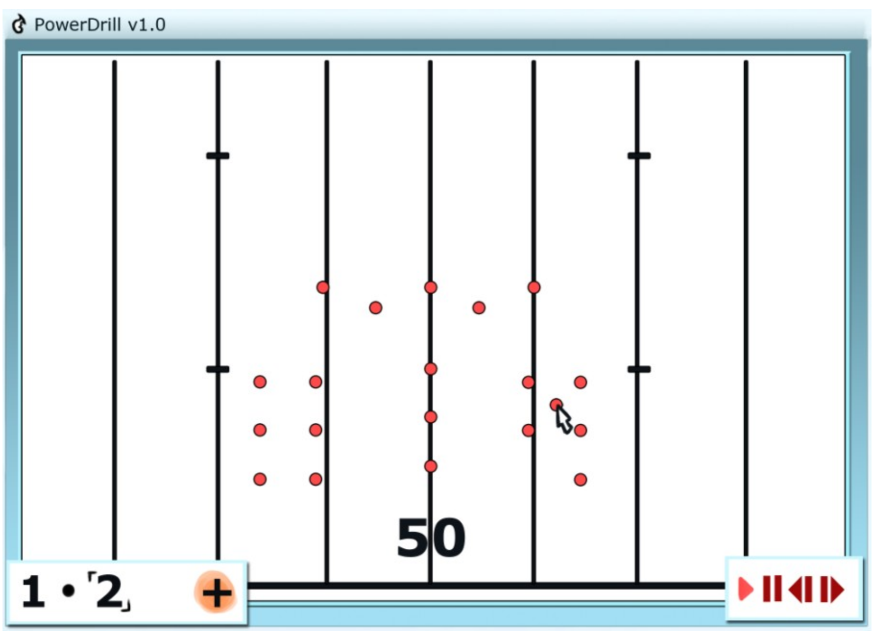
8. I've finally finished the first formation. Now, if I click the 'plus' button, I'll be able to add a new set, showing what the field will look like after the performers take a specified number of steps.



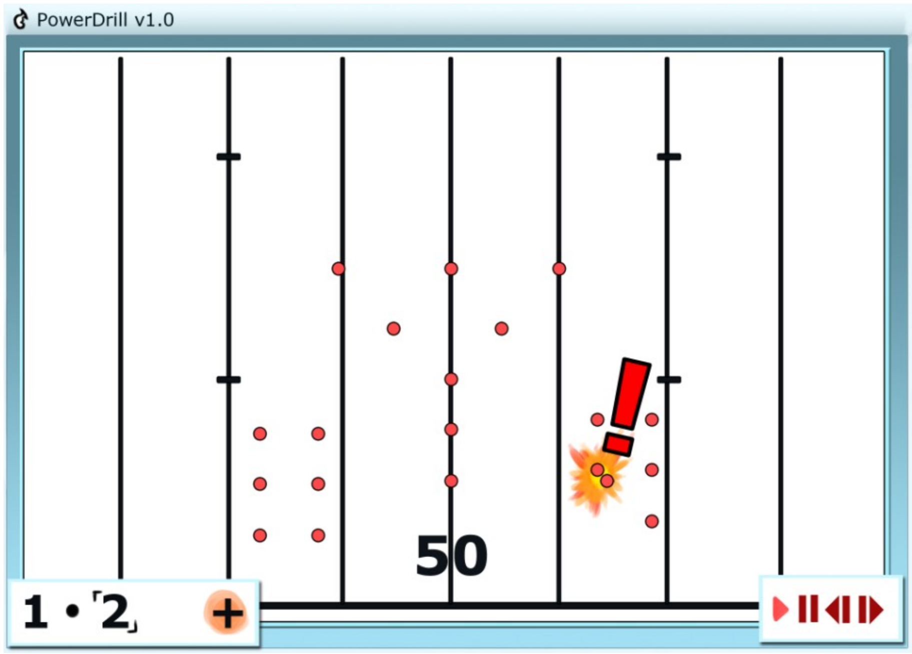
9. Now I can adjust the members in the second set. Keeping the mouse in the bottom right corner of the screen causes media controls to pop up. I wonder what happens if I click this play button…



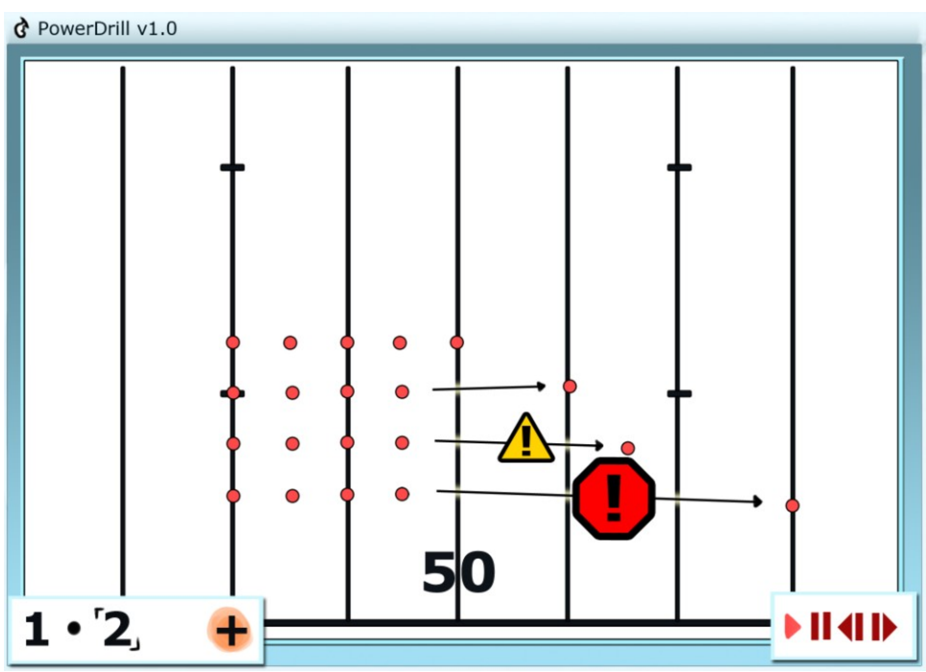
10. Whoa! It animates from one set to the next! Being able to see the movement makes me want to change some things. Let me just move this performer over here… and let's watch it again!



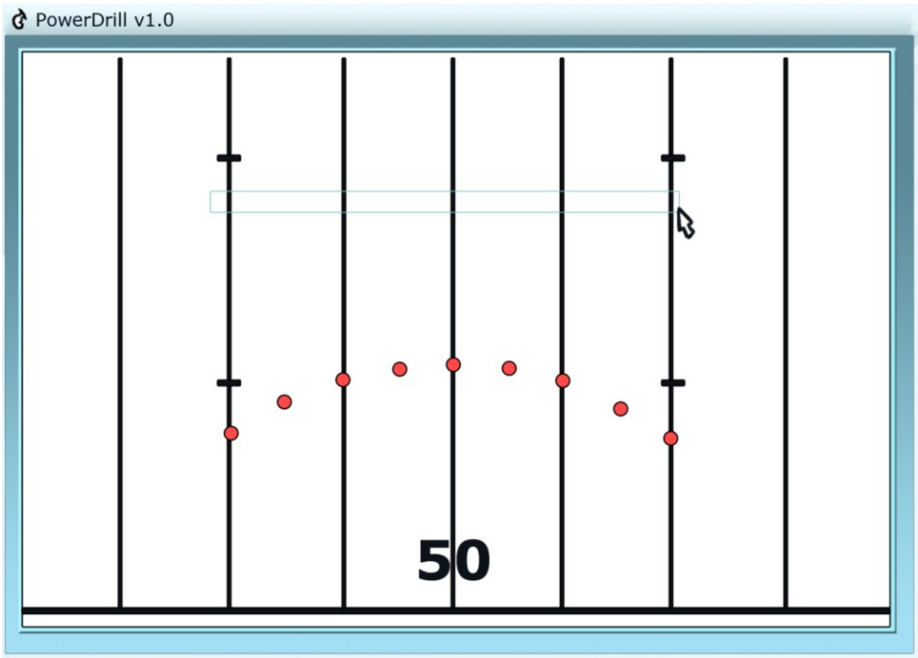
11. Oh no! I accidentally made two marchers collide with each other! I wouldn't want that to happen. I almost missed it!



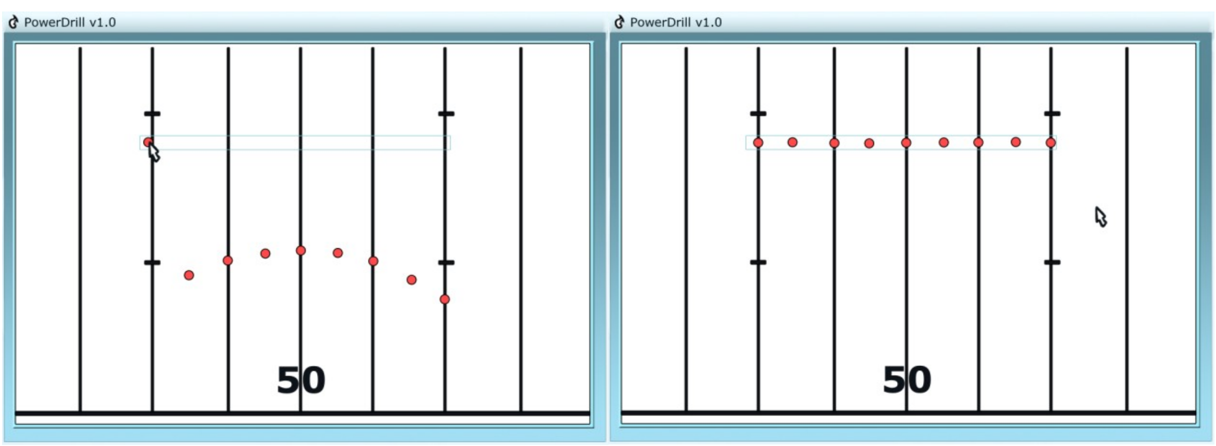
12. The program is also letting me know when I make moves that are a bit too difficult. Yellow warning icons appear on large, difficult moves, and red stop signs appear when a move is just impossible.



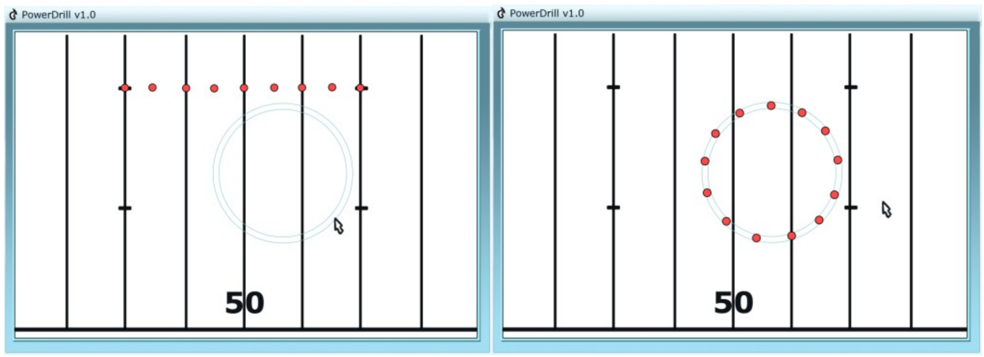
13. There are also some tools for adding shapes to the drill. If I click twice and create a line on the grid…



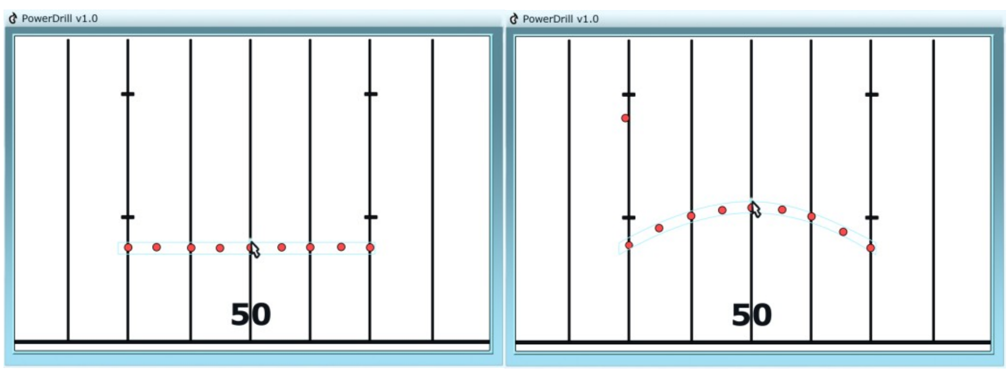
14. I can move performers into the line. They snap to the line making the drill very accurate, and the spacing adjusts depending on how many members are in the line.



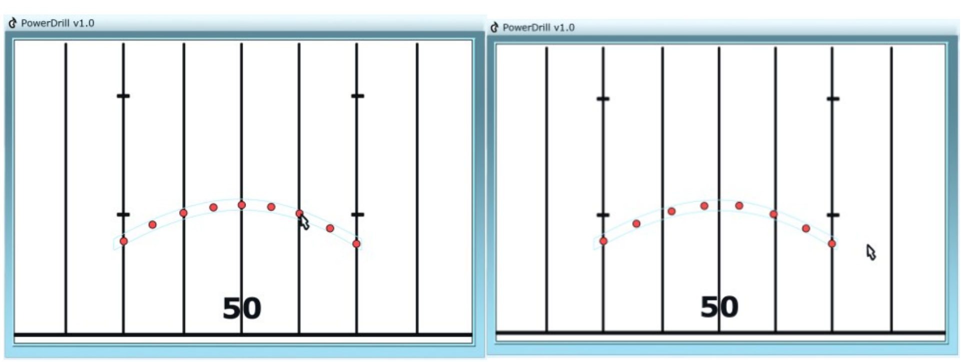
15. The circle tool is also an accurate way for me to add members into a circle. Circles are quite difficult to design by hand.



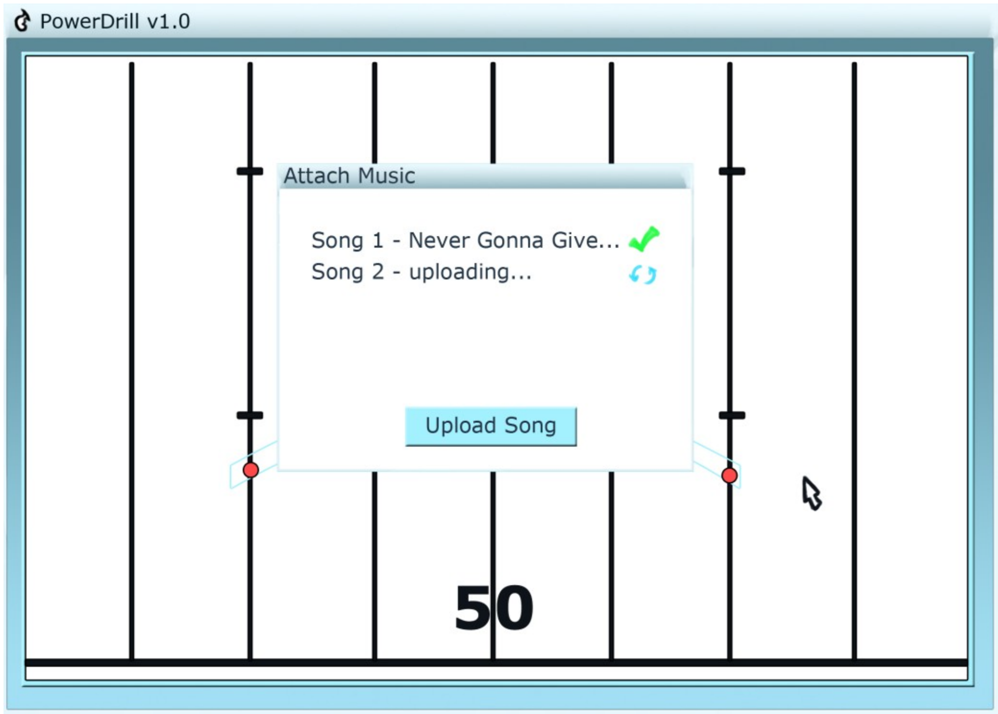
16. Curves are easy to create with the Curve Tool.



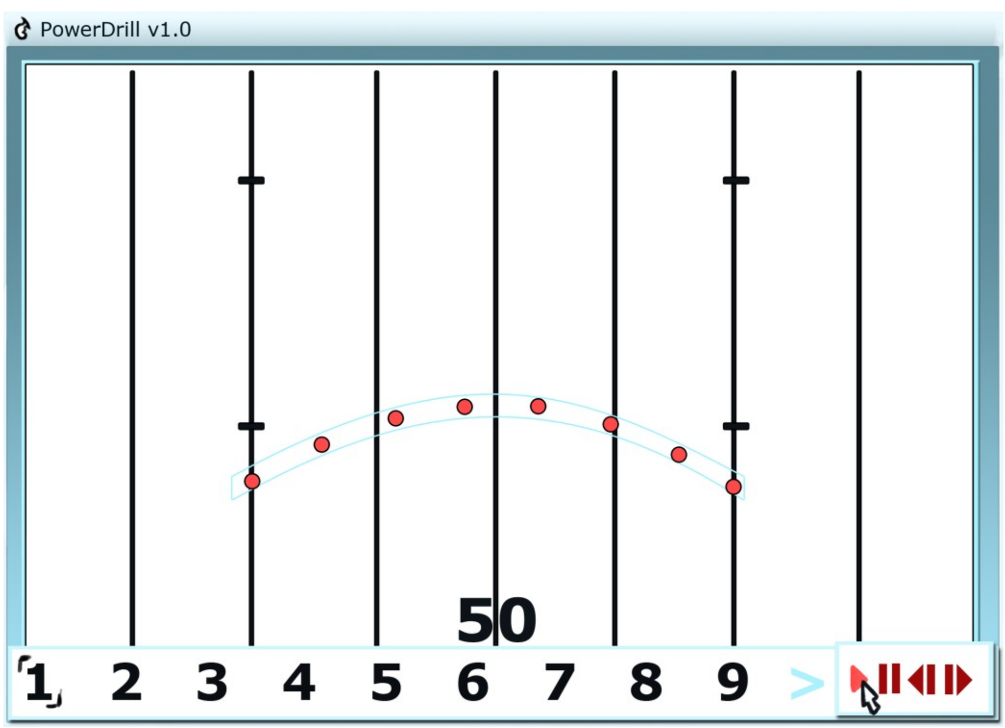
17. Oh no! A flute player in the band broke his leg and won't be able to march! Now I will have to fix every set in the show. This will take forever. Hey, wait! When I deleted the flute player from the drill, each set automatically resized to account for the missing space. That made everything much simpler!



18. Now the drill is done! I can attach these music files to the animation and be able to play back a full version of the show.



19. Ready! Let's click the play button…



20.

9f) Use Description

Tool Palette

a. Pointer

The Pointer tool is used to select performers on the field. Clicking on a performer once selects him, and clicking on the performer twice will select all the performers in the same shape.

b. Box Selection

The Box tool selects multiple shapes. Holding Shift selects a group of individual performers.

c. Line

The Line tool creates a straight line of performers. There are two ways to use this tool:

1. Click on a starting point on the field. It snaps to the grid by default, and a ghost line follows the cursor until the mouse is clicked again.

2. Double click on a point. A ghost line appears from that point to the cursor and a second ghost line is mirrored on the opposite side of the point.

Holding Shift snaps the line to angles of fifteen degrees. While holding Control, the point does not snap to the grid. After clicking the second point, a dialogue box pops up, and prompts for the number of performers in the line. The endpoints can be clicked and dragged to where they need to be before clicking “Accept” in the dialog to confirm the line. The entire shape can be moved by clicking and dragging the middle.

d. Curve Line Tool

The Curve tool acts almost identically to the Line tool, but the line can be dragged at the midpoint to form a curve with the mouse pointer. The midpoint snaps to different points; holding Control forces the point to not snap to the grid.

e. Box Tool

Click on a starting point on the field, which snaps to the grid. A ghost box will follow the cursor until the mouse is clicked again. Holding Control forces the point not to snap to the grid, while holding Shift forces a square. After clicking the second point, a dialogue box pops up and prompts for the number of performers in the box. The endpoints are clicked and dragged to where they need to be before clicking “Accept” in the dialog to confirm the line. The entire shape can be moved by clicking and dragging the middle.

f. Circle Tool

There are two ways to use this tool:

1. Click on a starting point on the field, which snaps to the grid. A ghost circle will follow the cursor until the mouse is clicked again.

2. Double click on a point. A ghost circle appears from that point to the cursor. The first

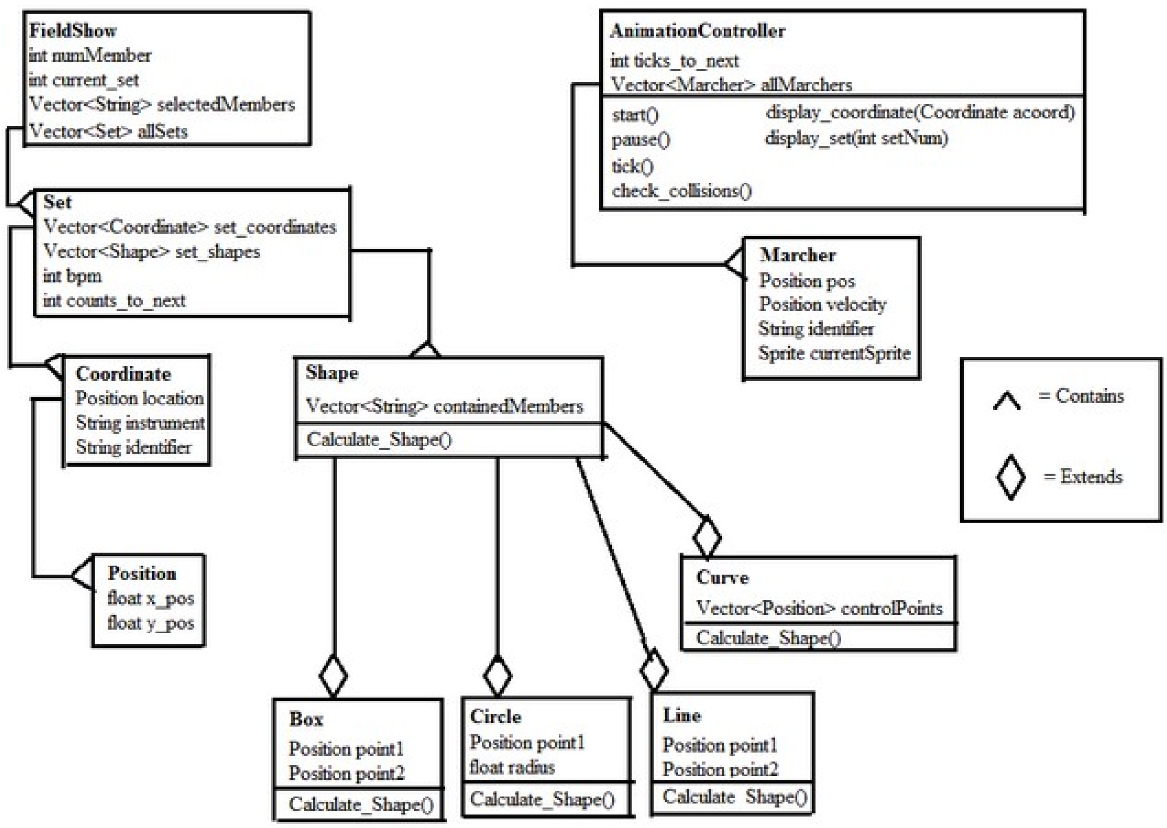
clicked point becomes the middle. Holding Control forces the point not to snap to the grid. After clicking the second point, a dialogue box pops up and prompts for the number of performers in the line. The endpoints can be clicked and dragged to where they need to be before clicking “Accept” in the dialog to confirm the shape. The entire shape can be moved by clicking and dragging the middle.

9g) Marketing

Selling *PowerDrill* requires positioning for a successful marketing campaign. With consideration to our target audience, we determined how to market the product to feature it in its best light. As software designers, we created *PowerDrill* to allow both pros and amateur drill writers to create works of marching band art by providing simple yet powerful tools, scaled for desired complexity. The software will be pitched to drill writers as a composition staple, to teachers as an educational tool, and hobbyists as the go-to recommended software. To do so, we will emphasize that the program has been made for marchers by marchers, understanding what crucial aspects are needed on either end. To be completely successful, the software would be available online and in music stores and advertised during college bowl games, releasing in early spring to cater to drill writers preparing for the start of football and marching band seasons in the fall.

9h) Implementation

The following section describes a possible technical implementation of *PowerDrill*. It is provided to give readers with a background in computer science a feel for the internals of the program.



The program is implemented in an Object Oriented style of programming. Each field show designed is represented by a separate instance of the FieldShow class. The data represented by a FieldShow object encompasses the show characteristics (such as music files, number of songs), number of sets, number of marching members, and the drill for the show. The data for the class is saved to an external file, which can then be loaded to work on the show at a later date. The FieldShow class contains an ordered vector of Set objects, which represent the individual sets in the field show. The Set class contains a vector of Coordinates (one Coordinate for each member in the FieldShow) and a vector of Shapes. In addition, each set contains an integer representing the BPM (beats per minute) of the set, as well as the number of counts to the next set. These two fields are used heavily in animation to determine the relative speed and amount of time each marcher will move.

The Coordinate class contains three pieces of information: the physical location of the member (x, y location), the instrument the member is playing, and a string that identifies the member. It is important to include instrument information in every set, as opposed to a lookup table, as this allows for equipment changes during a show. For example, if a member plays trumpet in one set and tuba in the next, the animation will reflect the change. Each member has a unique string identifier that differentiates it from the other members in the show, allowing for animation - the program needs to know which member to move from set A to set B. The member identifier is also useful in testing. If member T1 is present in set A and missing in set B, the show is not internally consistent and needs to be fixed. The string identifier is also displayed on the screen to differentiate between members playing the same instrument. By convention, members will be named by a letter and a number. For example, 16 trumpet players will be named T1 (Trumpet 1) through T16. When the members have to be moved or edited, the coordinate vector is searched through for the given identifier, and when it is found, the given coordinate can be edited.

PowerDrill also utilizes a Marcher class. The class does not store any data to an external file, but instead contains a position and velocity that are used in the display. The velocity is used to animate the class from one set to the next: each frame, the object is moved a certain distance based on the speed of the show and the velocity of the marcher. The velocity is implemented as a position, and it is described below in the "Display a Set" algorithm. The class also has a reference to the sprite or polygon it needs to display in the given position. The last piece of data the marcher class contains is a unique identifier. When the marcher object is initialized in the display_coordinate() command, the coordinate's identifier is copied over to the new marcher object, ensuring the correct identifier is displayed and that marcher objects can be re-used for multiple coordinates as long as the identifier is the same.

A prominent feature of *PowerDrill* is the ability to add and modify shapes to sets, which allows the program to dynamically adjust spacing and create sets with different numbers of marchers. These shapes are stored within the set class. There are four different types of shapes; each type of shape inherits from the shape superclass. Each shape contains two pieces of information: a description of the shape's location, and a vector that describes which members are in the shape. The location of a shape will be different for all shape implementations. For example, a circle can be described by stating its center and radius. However, a curve may have arbitrarily many control points, so the type of data necessary to store the location will vary. The most important shape function is Calculate_Shape(), which is implemented for all shapes. This function determines the location of the shape and the location of each included member and adjusts the members so that spacing is even. This function is discussed more fully in the algorithms section.

Error checking is in place to ensure each show's data is accurate. When a show is either loaded or saved, the file is checked for consistency in the following way. First, the initial set is iterated through, and each contained identifier is added into a separate table. Then, each set is examined to determine each identifier in the set is present in the table. For example, if set one contains members T1, T2, and T3, each following sets must contain those three members, or the error check will fail. In addition, each set is examined to ensure no identifier is present twice. These two checks together guarantee each member is present exactly once in each set, which is required for successful animation. In the rare case that either of these checks fail, *PowerDrill* can attempt to fix it, although data will probably be lost. As each set is iterated through, each superfluous or duplicate coordinate is deleted, and each missing coordinate is added with a default location. The user is warned that data was corrupted and that the show should be manually checked to ensure each member is present.

Design Consideration

In the design of this program, two clear choices existed for the class hierarchy. The program could either store a list of marchers, which hold their personal coordinates for each set, or a list of sets, which hold the coordinates of each marcher in the set. There are positives and negatives to each option, but after consideration it was decided the better option is to store a list of sets. This method offers many advantages to the program. Each set can store a list of associated Shapes, which would not be possible with a list of members. The display function for a set is also simplified; display() only has to examine one set, not iterate through every member in the show. This simplified display function improves the run-time efficiency of the field show animation. As well, the addition and deletion of individual sets is more efficient. A downside of this paradigm is inefficient addition and deletion of members; each set must be iterated through and edited to add a single member. Overall, the efficiency of adding members is not as critical as efficient animation, where a drop in performance could seriously lower the frame rate of the program. Therefore, because of these benefits, *PowerDrill* shows contain a list of sets in their class hierarchy.

PowerDrill uses vectors to store important application data. A vector was chosen for these tasks over other container types such as lists, arrays, and deques. A vector is an abstract container in C++ that supports random access to components and dynamic resizing. An array is a container that has efficient random access, but cannot be resized, and for this reason, a vector is preferable to an array. A deque (also known as a double-ended queue) is a modified vector that allows the user to efficiently insert objects at both the front and the back of the vector, while a regular vector only has efficient back insertion. The choice between a vector and a deque is fairly simple. The majority of sets will be appended at the end of a show, as it is assumed most drill writers will write set one, add a new set, and continue in order. The only benefit to a deque will be if drill writers often add sets to the beginning of a show, which would presumably be a rare occurrence. Therefore, a vector is preferred to a deque. A list is non-contiguous memory that supports efficient insertion and iteration, but inefficient random access. Other downsides of a list include, "two additional pointers associated with each value and the indirect access of the value through a pointer" (Lippman, Lajoie 256). For both containers, random access will be necessary. The program will often have to display the contents of any set in the show, which requires efficient random access. Also, if the user wants to, for example, print out a member's coordinates in every set, the list of coordinates can be indexed, which will be more efficient than iterating through each set to find the correct coordinate. Since efficient random access is needed for both containers, a vector is a better alternative to a list.

The vector does have several downsides that need to be addressed. Insertion or deletion of elements other than at the end of the vector requires each element to be copied and shifted to account for the empty space, which is very inefficient. The overhead is particularly expensive for large, complex class objects, such as the set class, containing a list of potentially hundreds of coordinates. To reduce the complexity of the class, the list of coordinates is instead stored indirectly through a pointer. Instead of storing all the coordinates itself, the class stores the location it can look to find the vector of coordinates. In this case, the pointer is much smaller than the vector it refers to; therefore, when a set has to be copied and shifted over, the operation is much faster. In addition, the total number of sets, or coordinates, will be less than five hundred in almost every case. Since the class objects are being kept as small as possible, each copy/shift operation is relatively efficient. A large amount of these operations will theoretically still be accomplished quickly, especially on modern processors. Through these strategies, the downsides of a vector can almost be completely negated, which is why it is the best container for the job.

Algorithms

- Initialize a Show: Create a new fieldshow class object. The show contains the information gathered during the quickstart wizard, such as type and number of marchers. The fieldshow is initialized to contain exactly one set, containing coordinates equal to the number of marchers in the show. Each marcher is placed in a default, unique location. Each member's identifier comprises a letter for their instrument and a number; ex: T1 (trumpet 1), F5 (flute 5). The first set will, by default, have a BPM of 120 and 16 counts to the next set. In addition, display the field described in the fieldshow class (i.e. what size hashes, what color, etc).
- Add a Set: Push a new set object to the end of the fieldshow's set vector. The coordinate vector from the previous set is copied and used to initialize the new set. Therefore, the new set contains the same positions as the previous set. When the set is created, its BPM and counts_to_next are also copied from the previous set. These can then be edited easily through other facilities.
- Insert a Set: If the set to be inserted is not at the end of the fieldshow, insert a new set

object as above into the fieldshow's set vector. For a vector, this requires every object after the inserted object must be copied and moved. The operation is inefficient, but the benefits of using a vector outweigh this disadvantage.

- Delete a Set: Delete the given set from the fieldshow's set vector. The operation is final. If the set is not the last set of the show, each object in the vector must be copied and moved so the vector's memory is still contiguous.
- Adjust Counts for a Set: Update the set's counts_to_next field to the new value.
- Adjust BPM for a Set: Update the set's BPM field to the new value.
- Initialize the Field: Iterate through the first set's vector of coordinates. For each coordinate, call the display function (Display a Coordinate; described below). This command populates the AnimationController with all the marcher objects it needs to animate the show.
- Display a Set: Iterate through the given set's vector of coordinates. Then, for each marcher object, find the location it will move to in the next set. This is accomplished by iterating through the next set's list of coordinates, searching for the marcher's identifier. Once the next coordinate is found, calculate the velocity based on the distance between the two coordinates. For example, if a marcher in the current set is at (0, 2), and in the next set the marcher is at (1, 5), the velocity will be initialized to (1 – 0, 5 – 2) = (1, 3). In other words, from this set to the next, the marcher needs to move 1 unit in the X direction, and 3 units in the Y direction. If the instrument for the marcher has changed, update the sprite reference in the object. Once the velocity for each marcher object is updated, examine the BPM and counts_to_next fields in the current set to determine how far to move each marcher per tick. For example, assume the BPM is 120, and counts_to_next is 16. *PowerDrill* animations run at 30 frames per second. 120 beats per minute divided by 16 beats is 7.5 seconds. At 30fps, this turns

out to 7.5 times 30: 225 ticks. Therefore, the velocity of each marcher has to be divided by 225 to create a smooth, even animation where every marcher reaches their next position at the correct time. The amount of ticks to the next set (225, in this case), is then passed to the animation controller. This function does not begin the animation, rather it prepares the animation to run when AnimationController:start() is called.

- Display a Coordinate: Create a marcher object on the field at the given x and y location. The program will determine which sprite or polygon the marcher object will reference, based on the instrument in the coordinate data. In addition, display the identifier for the coordinate next to the given sprite or polygon.
- Begin Animation: Call AnimationController::start(). This starts or resumes calling of tick() every frame. When the animation begins, it stops the rest of the program from activating functions. The only action allowed after the animation begins are camera controls in 3D mode and AnimationController::pause().
- Pause Animation: Call AnimationController::pause(). This causes tick() to not be called, which effectively pauses the animation. All menu items that were locked out when animation began are re-enabled.
- Tick: Move each member in the animation controller based on their velocity. The new position will be at (pos.x + velocity.x, pos.y + velocity.y). When each member moves, check collisions with the bounding box of each other member. If there is a collision, report that information to the user. In addition, decrement the ticks_to_next field by one. When ticks_to_next reaches zero (and another set exists after the current set), the following occurs: the fieldshow's current_set field is incremented, and AnimationController:display_set() is called on the new set. This loads the next set's information and velocities into the AnimationController's

list of marchers.

- Add a Member: Increment the fieldshow's numMember field by one. Then, iterate through the fieldshow's vector of sets. For each set, push a new coordinate onto the vector of coordinates with the following data: a default location (0, 0), the given instrument, and a unique identifier. The identifier is the instrument's letter, followed by the first available number. For example, if the show has trumpets 1-16, a new trumpet player's identifier will be T17. If the show has trumpets 1-4 and trumpets 6-9, a new trumpet identifier will be T5.
- Delete a Member: Decrement the fieldshow's numMember field by one. Then, iterate through the fieldshow's vector of sets. For each set, iterate through the vector of coordinates, and delete the member with the given identifier. The operation is final.
- Select a Member: Determine which member was clicked. Iterate through the fieldshow's selectedMembers vector. If the member is not present, push the member's id onto the vector.
- Select a Group: For each member selected, push the member's id onto the fieldshow's selectedMembers vector, if it is not already present. This could happen in one of two ways: the user could click and drag the mouse over a group, or the user could select every performer in a shape by double clicking.
- Deselect a Member: Determine which member was clicked. Iterate through the fieldshow's selectedMembers vector. If the member is present, remove the member's id from the vector.
- Deselect Members: Clear the fieldshow's selectedMembers vector so that it is empty.
- Move a Member: Adjust the position of the selected marcher to the new position (where the marcher was dragged). Iterate through the current set's list of coordinates, and find the given coordinate. Edit the found coordinate as well to reflect the change. Both the coordinate and the

marcher objects need to be edited at the same time so the animation does not de-sync. For example, if the marcher is edited but the coordinate is not moved, the display will reflect the change, but the new position will not be saved internally.

- Remove a Shape: All shapes can be removed in the same way: remove the given shape data from the set's list of shapes. When the shape is removed, each member stays in their current location.
- Add Member to Shape: Iterate through the shape's containedMembers vector. If the given identifier is not already present, push the member's identifier onto the vector.
- Remove Member from Shape: Remove the given member's identifier from the shape. Then, call Calculate_Shape() to adjust the spacing for the members remaining in the shape.
- Collision Detection: (Bounding boxes): During animation, each time a marcher moves, their position is compared to the positions of each other member.

Section X: Conclusion

While problems exist in current software, the hypothetical program *PowerDrill* is one step to solving the disconnect between drill writer and marching member. After reviewing popular software titles and gaining insight from the community, we determined the best changes to make for a new product. The lack of communication between member and writer was addressed by adding media export features and facilitating social media inclusion. To ease the learning curve of standard programs, the hypothetical software was crafted to be as intuitive and simple, yet powerful and useful. Overall, we have determined *PowerDrill* should be used as an example for future drill-writing software as well as a possible finished product, achieving our goal of creating a problem-solving marching band program design.

Section XI: References

"History of Drill." *The Drill Pad*. Web. 05 Nov. 2011.

"Football Field Dimensions." *Tru Mark Athletic Field Markers*. Tru Mark. Web. 28 Feb 2012.

Ambridge Area High School Junior Reserve Officer Training Center. "Category 5, Skill 2: History of Drill." 2003. *Microsoft Word* file.

Bailey, Wayne. *The Complete Marching Band Resource Manual.* Pennsylvania: University of Pennsylvania Press, 2003. Print.

Cummings, Steve. "Reviews: Pyware Music Writer 1.1." *Macworld* Dec 1990: 245. Print.

Drill Quest. Midi Classics. 23 June 1998. Web. 26 Oct. 2011.

Field Artist Central. Raven Labs, 2011. Web. 29 Feb. 2012

EnVision. Box 5 Software, 2011. Web. 28 Feb. 2012..

Lippman, Stanley B., Lajoie, Josée. *C++ Primer, Third Edition*. Reading, MA: Addison-Wesley, 1988. Print.

Marksworth, Wayne. *Dynamic Marching Band*. Michigan: Isaac Publishing, 2008. Print.

McNeill, William Hardy. *Keeping Together in Time: Dance and Drill in Human History*. Cambridge, MA: Harvard UP, 1995. Print.

National Collegiate Athletic Association. "Field Diagram." 2011. PDF file.

Pearl Drums. Pearl Corporation, 2005. Web. 28 Feb. 2012.

Pyware. Pygraphics, Inc, 2011. Web. 5 Dec. 2011.

Seney, Brian. "Drillwriting Software Research Survey." Survey. 30 September 2011.

Seney, Brian. "Marching Band Software Research Survey." Survey. 12 September 2011.

Sousa, John Philip. *Marching Along: Recollections of Men, Women and Music*. Boston, MA: Hale, Cushman & Flint,1928. Print.

Section XII: Production Details

Designing *PowerDrill* required collaboration and effort by four individuals. Responsibilities of each member over the course of development are listed below.

Beth Hankel

- Requested books through an inter-library loan system that aided in our understanding of marching bands and drill writing.
  - Wayne Markworth - *The Dynamic Marching Band*
  - William Spencer-Pierce - *Marching Band Arranging*
  - Kristen Laine - *American Band*
  - Chris Previc - *The Student's Guide To Marching*
  - Kerstin H. Becker - *Through These Doors (Massachusetts Minutemen History)*
  - William H. McNeill - *Keeping Together In Time (History of drill and dancing)*
  - *Drill and Ceremonies (Official US Army booklet)*
  - John Philip Sousa - *Marching Along: Recollections of Men, Women and Music*
- Found E-Books
  - John Philip Sousa - *Experiences of a Bandmaster*
- Downloaded *Pyware* for early analysis of feature set
- Collaborated on creation of surveys that provided the team with information on what features the community would want added to their ideal piece drill writing software
- Researched and ordered *Halftime* Magazine issues to help with research
- Collaborated on research analysis
- Created art assets and concepts

- Responsible for Research, How-To, Marketing, art including final storyboard and photos, and Conclusion
- Proofread and formatted the document into final format

Michael Metzler

- Responsible for collecting survey results/suggestions for features for analysis
- Collaborated on creation of surveys for features and suggestions
- Contacted various individuals in the industry for input on features and research
- Downloaded *Pyware 3D* for analysis of feature set and wrote analysis
- Downloaded *FieldArtist 3* for analysis of feature set and wrote analysis
- Collaborated on research analysis
- Responsible for defining the Project Goals
- Responsible for defining the tools for *PowerDrill*

Brian Seney

- Requested books through and inter-library loan system that aided in our understanding of marching bands and drill writing.
    - Wayne Bailey - *The Complete Marching Band Resource Manual*
    - Kim R. Holston (compiled by) - *The Marching Band Handbook*
    - Dan Ryder - *Techniques of Marching Band Show Designing*
- Created, monitored, and managed the surveys used to gather suggestions and desired features from the community
- Collaborated on creation of said surveys
- Contacted about 50 industry professionals about survey
- Collaborated on research analysis

Anthony Sessa

- Collaborated on creation of survey
- Researched marching bands and drill writing using some of the acquired texts and magazines
- Contacted about 100 industry professionals about survey
- Created meeting agendas and scheduled weekly meetings to keep the design team going
- Responsible for research of *Drill Quest* software
- Wrote *Drill Quest* analysis
- Wrote History of Drill Writing section
- Responsible for table of contents
- Added photos to document
- Helped format document into final format

Section XIII: Authorship, Acknowledgements

Beth Hankel is a senior Interactive Media & Game Development major at Worcester Polytechnic Institute. Outside of creating art, she enjoys playing the trombone for the WPI Pep Band and sewing. She resides in Worcester, MA. She would like to thank the WPI Pep Band and all its patience- it has been a tremendous help and inspiration. She would also like to thank *Halftime* magazine for supplying a wealth of knowledge for a highly specific subject. Finally, the WPI Pep Band President Adam Thibault deserves more than just thanks in an IQP paper; he deserves a medal for all he has done. Thank you, Adam, for your unbounded help and advice.

Michael Metzler is a junior studying both Computer Science and Interactive Media & Game Development at Worcester Polytechnic Institute. He is an Eagle Scout, a brother of the national service fraternity Alpha Phi Omega, and a member of the WPI Pep Band. He spends his free time playing video games, and has a growing interest in developing software for Android devices. He would like to thank the WPI Pep Band and their help and interest in our project was beneficial and welcomed. He would like to thank the Sales team of *Pyware* for helping with information of demographics for their software. Thanks also go to Heidi I. Sarver, Associate Professor of Music at the University of Delaware for giving suggestions for research material. Michael would like to thank his rest of his IQP group for putting up with his shenanigans. Finally he would also like to thank Professor Moriarty for his insight on this project as our advisor.

Anthony Sessa is a junior studying Interactive Media & Game Development at Worcester Polytechnic Institute in Worcester, Massachusetts. When he's not toiling away on assignments,

you can usually find him playing video games, at a Yankee game, or locked away in his home studio making music. He hails from the faraway land of New York. He would like to thank the other members of the Music Software Design IQP for being such awesome people to work with. Without amazing group members, this document would not possess the high amount of quality that it does. He would also like to thank Professor Moriarty for being a great advisor and guide during this whole process. We couldn't have done it without you, Professor! Thanks also go out to the WPI Pep Band for taking the time out of their schedules to participate in the surveys and picture taking.

Brian Seney is a junior Computer Science / Interactive Media & Game Development major at WPI. He participates in several music organizations on campus such as Concert Band and the men's Glee Club. He has been a member in several marching organizations, such as the Rockville High School Marching Ram Band, and the WPI Pep Band. He would like to thank the rest of his group for being invested in the quality of the final document, as well as Professor Moriarty for his successful advising. He would also like to thank the numerous people who took the time to fill out the surveys, which certainly raised the quality of our final paper.