2020

DrillPad: A Marching Band Drill Writing Web Application

Justin Wickenheiser
Grand Valley State University

DrillPad: A Marching Band Drill Writing Web Application

Justin T. Wickenheiser

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

April 2021

The signatures of the individuals below indicate that they have read and approved the project of Justin T. Wickenheiser in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

<name of project advisor>, Project Advisor	Date
<name of GPD>, Graduate Program Director	Date
<name of unit head>, Unit head	Date

Abstract

The idea for this project came as a friend going through the Music Education program lamented how difficult it is to write drill for marching band. There is one primary application that is used for writing marching band drill, and it comes with two major pitfalls: It is complex, and it is expensive. DrillPad is designed with the goal of addressing those pitfalls. Creating a web-based drill writing application makes it available for free, or a low cost. The different features that DrillPad provides revolve around adding and moving performers easily and quickly.

This project makes heavy use of the web graphics library, PaperJS, from the creation of the marching field's grid, to rendering the performers, even down to the animation. DrillPad utilizes PostgreSQL on the back-end and is deployed on Heroku.

Introduction

This project is a web application for writing marching band drill. I was inspired to create this application to address to pitfalls of the industry standard drill writing tool, Pyware 3D. Pyware is complex and expensive. It also does not handle rotations around reference points well. My goal was to create an application that is easy to use, inexpensive, and can handle rotational movements. The decision to create DrillPad as a web application addressed making drill writing software inexpensive. While selecting and creating features, simplicity was the focus.

Project Management

I tracked the scope requirements as a list in the GitHub wiki. As I thought of additional features, they were tracked on a separate wiki page. After the project was tested and feedback received, I started created issues inside the GitHub project.

Organization

DrillPad is built on the SailsJs model/view/controller framework, connected to a PostgreSQL backend, and is deployed on Heroku. I make heavy use of the PaperJs web graphics and animation library. There are four base JavaScript classes: DPFeature, DPChart, DPPerformer, and DPEditor.

The DPFeature class is a template that is extended by all the custom features. The DPChart class acts as a transparent layer. It allows for a title and a chart number, but the most important piece is that it tracks the number of counts it takes to get to the next chart.

The DPPerformer class extends the PaperJs PointText class. A performer has a drillNumber structure, positionSet structure, and an animationPositionSet array. The drillNumber consists of the actual drill number and the angle that it is rendered at on the canvas. The positionSet is where the performer tracks all their specific positions. The key is a chartId, and the value is an array of PaperJs Point objects. The number of elements in the array is equal to the number of counts for that specific chart plus one for the starting position of the chart. The animationPositionSet array is all the necessary PaperJs Point objects for the animation feature.

The DPEditor is the backbone of the project. It keeps an array of DPChart objects, and an array of DPPerformer objects. It keeps an array of all the custom features. It tracks the global show settings, such as which hash line to use, and the pixels per step value used in all the calculations. The editor also tracks the active chart index, active feature, and selected performers. It stores a reference to the PaperJs View object and the animation settings. The constructor builds the user interface including the HTML canvas element, header, toolbar with features, modal to be used by the features, sets the animation onFrame function, and creates PaperJs Layer objects.

Reflection

This project set out to create a drill writing application that was easy to use, inexpensive, and can handle rotational movements. I succeeded in creating a drill writing application that is inexpensive, as it is deployed on Heroku and can be used for free. DrillPad also implements a feature to apply rotational movements to performers as well as movements in relation to a reference point. Two of the three objectives were successfully completed, however the objective regarding ease of use is less clear if it was achieved. I find DrillPad easy to use, but I am biased. I wrote the application and all the features. I know what to do and where to click in order to accomplish a task. I asked two friends who went through the Music Education program and are both music educators to test the application and provide any feedback. Only one person provided feedback, but none of it was in regards to the complexity of using the application.

Regardless of if I successfully made a drill writing software that is easy to use, I did get plenty of experience in working with a NodeJs application, deploying to production, and debugging between development and production environments. I encountered many problems when deploying to production where they were nonexistent in development. This primarily boiled down to simply not knowing the SailsJs framework from the beginning. I had to learn the quirks of the framework as I progressed through the project.

Conclusions

I was able to complete all the requirements outlined in the scope without any major errors or complications. Ideas for future features have been noted in a wiki page or as an issue in the GitHub project. Future ideas include written instructions, performer orientation, snaping performers to reference points, and an adjusted step move feature.

Appendices

Deployed Application: https://jtw-drill-pad.herokuapp.com/

Reference Manual: https://jtw-drill-pad.herokuapp.com/reference/manual

GitHub Repository: https://github.com/justinwickenheiser/cis693-drill-pad