**Software Proposal**

**General Description**Create a program that allows for FTC teams to create Autonomous programs for their robot by dragging points around a picture of the Velocity Vortex field.

**Specific Details/Features**  
 - Drag and drop points that the robot will navigate to.

- Calculate the angle between all points on the field so the robot knows how far to turn

- Provide configuration for robot hardware (DcMotors, base drift, sensor names)

- Generate Java code for teams to paste into android studio for both red and blue alliance

- Adjust speed for driving and turning for each point

- Navigate between points, drag and drop, arrow keys

- Save and load .FTC files

- Convert program “pixel” space to “actual” space

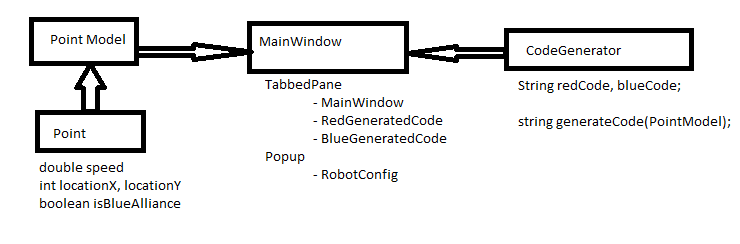
- Save to android studio project for fast compile

**Target Audience**Rookie FTC teams/ teams without any software members that need to quickly create an autonomous program during competitions.

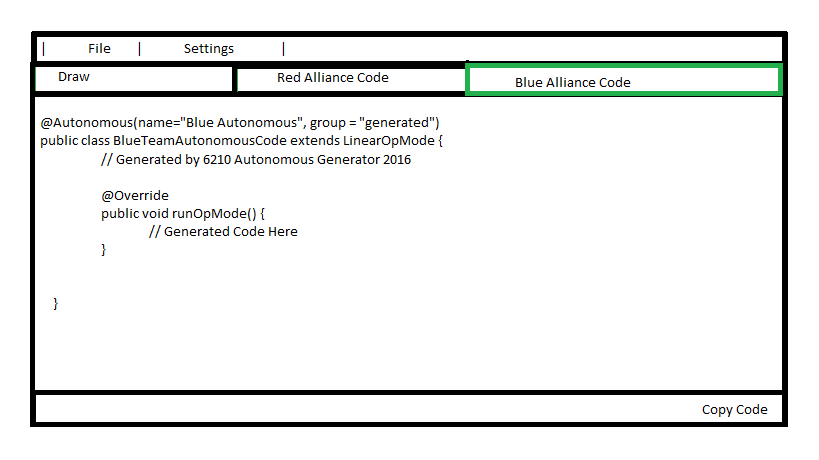
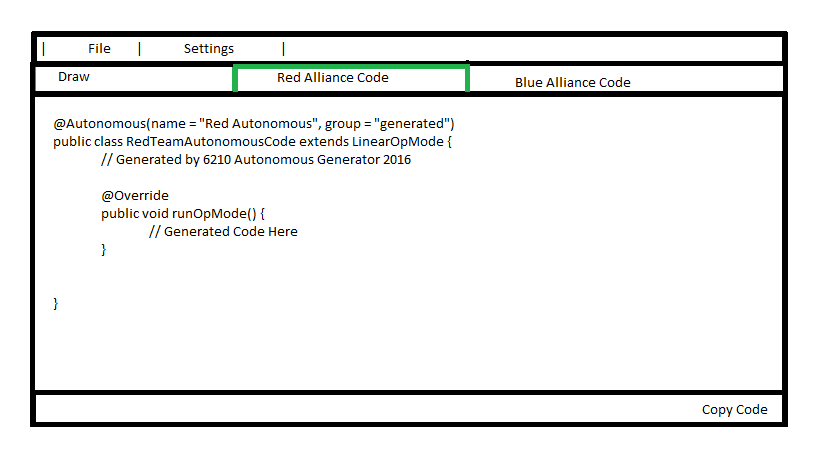
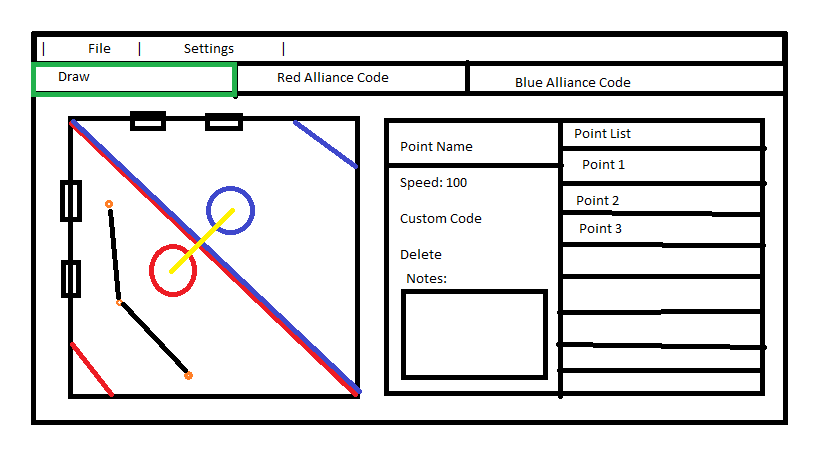
**Purpose: Why would anyone use it?**Many teams at FTC competitions don’t run an autonomous program because it takes too long to create, their software team isn’t experienced enough, or they simply have no software members. This program will allow for anyone to create an autonomous program; all you need to do is draw out where you want the robot to go!

**Design Approach**Chaos, because this program is straightforward and has 1 major task it will be easy enough to develop the most important systems first. Because of the close deadline, the Chaos model will make sure the fundamentals of the program are there, and if we must cut off some parts of the program they won’t be very important.

1. Design and create GUI
2. Place points on image and drag them around
3. Calculate angle and distance of points & convert measurements to inches
4. Generate code for single alliance
5. Load points into DB list & workout navigation
6. Reflect points for other alliance code
7. Create custom code window to add motion to servos ect.

**Organization (Class Structure, Data Classes)**

**GUI**

Tabbed pane with 3 parts: Drawing, Red alliance, and blue alliance  


**Things you must figure out**Before you start, you must identify stuff that you must figure out. For example, write to a spreadsheet, make clickable graphics, select items in a drop-down menu, make images auto update.

-Workout double buffering for image with points displayed on top of field image

-Work out how to drag points around on the field while keeping a constant FPS.

-Auto-update text fields for red and blue alliance code output.

-Figure out how to save to android studio project

**Programming Tasks**

**-** Calculate angle between any 3 given points

- Calculate error for actual driving from robot (percent error)

- Convert the distance between points on the image to inches on the field

- Reflect points for the opposite alliance over y =-x

**Research**None