



# Introduction to Vuforia

**FIRST**  
LEGO  
LEAGUE JR.

**FIRST**  
LEGO  
LEAGUE

**FIRST**  
TECH  
CHALLENGE

**FIRST**  
ROBOTICS  
COMPETITION



# Introduction to Vuforia

- What is Vuforia?
- Programming Vuforia
- Using Vuforia in Rover Ruckus
- Vuforia Resources

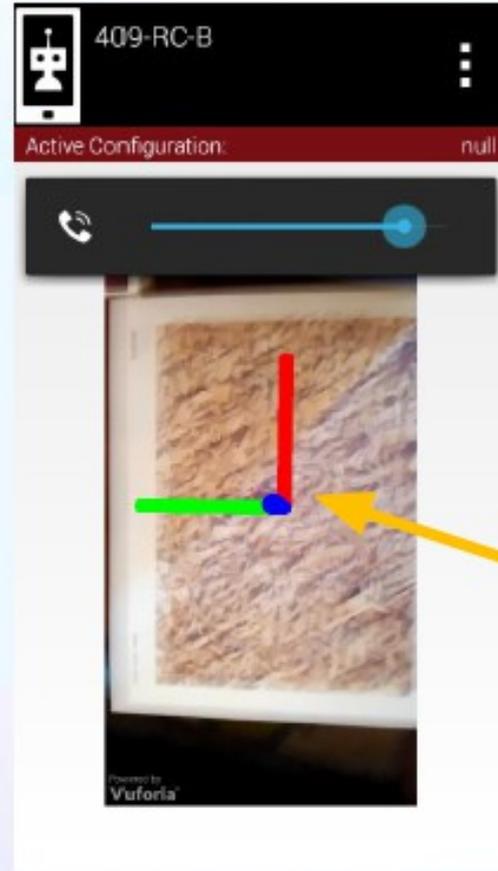
# What is Vuforia?

Vuforia is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track Image Targets in real time.

This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world images when they are viewed through the camera of a mobile device.

# Vuforia screen samples

Driver Station



Robot Controller Camera

Vuforia Overlay

- Camera view with overlay

# Blocks Programming Vuforia

The screenshot displays the FIRST robot controller console interface. At the top, the browser address bar shows the URL: `192.168.49.1:8080/?page=FtcBlocks.html?project=BlocksVuforia&pop=true`. The interface includes a navigation bar with "Blocks", "OnBotJava", and "Manage" options. Below this, there are buttons for "Save Op Mode", "Export to Java", "Download Op Mode", and "Download Image of Blocks". The main workspace shows the "Op Mode Name: BlocksVuforia" and "Autonomous" mode selected. The programming area contains the following logic:

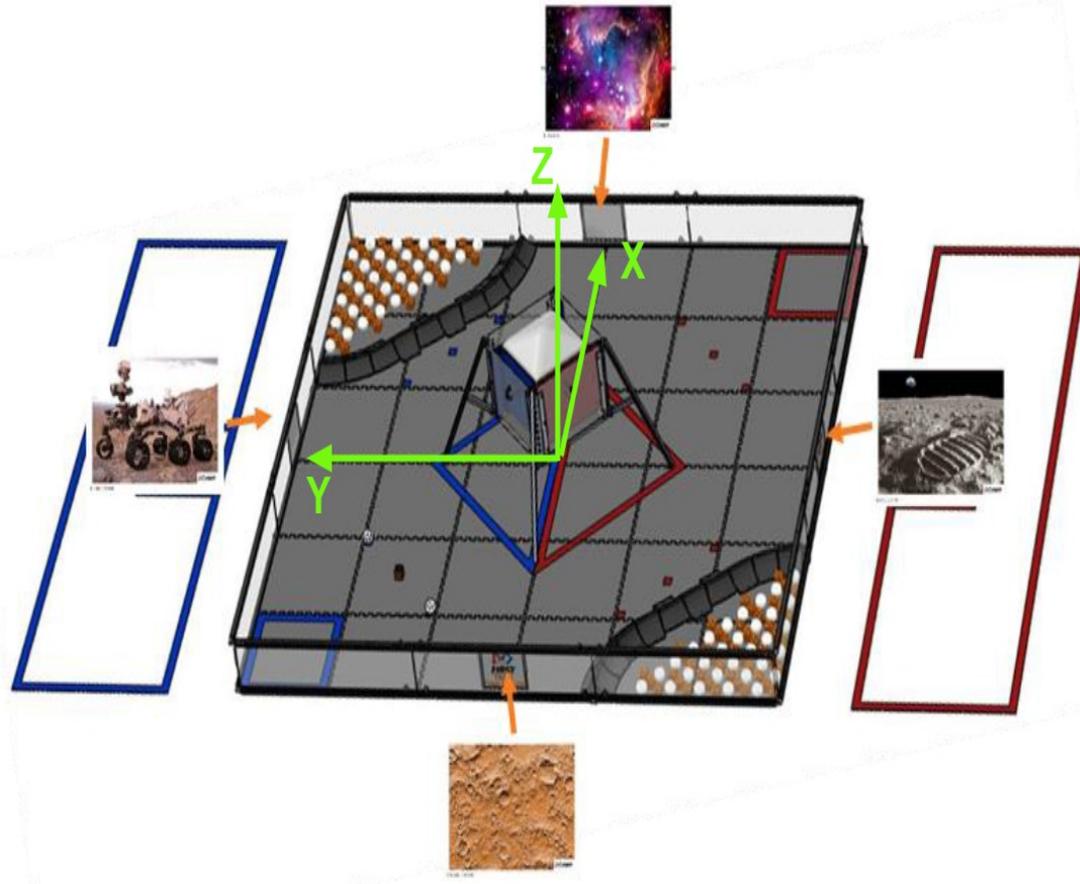
- Call `Vuforia . activate`
- Repeat `while` loop: call `BlocksVuforia . opModelsActive`
- Do loop:
  - Get the tracking results.
  - Set `vuMarkResult` to call `Vuforia . track` with `trackableName` set to `TrackableName . BackPerimeter`.
  - Is a VuMark visible?
    - If `VuforiaTrackingResults . isVisible` is true:
      - Set `vuforiaTrackingResults` to `vuMarkResult`.
      - Do loop: "Yes, we see one."
        - Call `Telemetry . addData` with key `"VuMark"` and text `"A VuMark is visible."`
        - What type of Relic VuMark is it?
          - If `VuforiaTrackingResults . Name` equals `TrackableName . BackPerimeter`:
            - Do loop: call `Telemetry . addData` with key `"Back Wall"` and text `"Stars"`
          - Else if: call `Telemetry . addData` with key `"Front Wall"` and text `"Front Perimeter"`

# OnBotJava Programming Vuforia

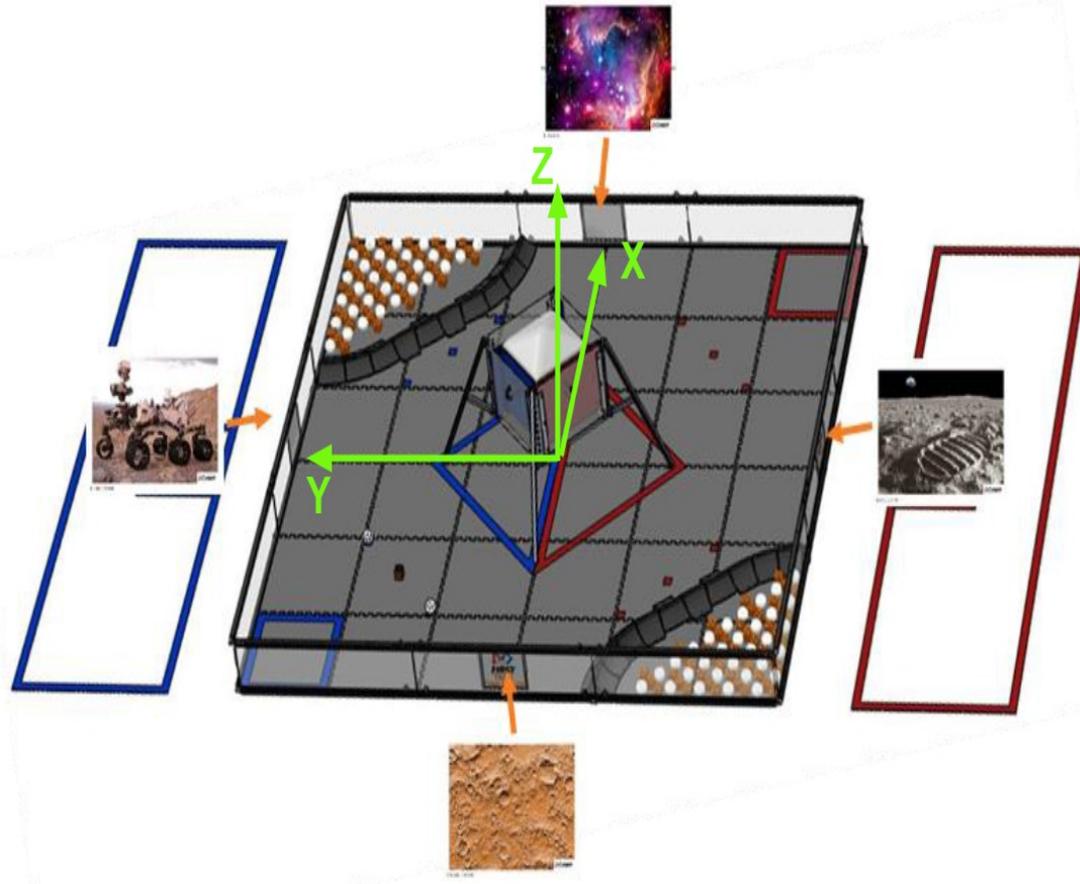
92 \* is explained below.  
93 \*/  
94 @Autonomous(name="OnBotJava Vuforia NRR", group="Demo")  
95  
96 public class OnBotJavaVuforiaNRR extends LinearOpMode {  
97  
98 /\*  
99 \* IMPORTANT: You need to obtain your own license key  
100 \* 'parameters.vuforiaLicenseKey' is initialized is  
101 \* A Vuforia 'Development' license key, can be  
102 \* web site at <https://developer.vuforia.com/lic>  
103

Build started at Fri Oct 12 2018 17:42:43 GMT-0700 (Pacific Daylight Ti)  
Build finished in 5.9 seconds

# Rover Ruckus Field Coordinates



# Rover Ruckus – Vuforia Notes



# References

## Programming Resources

<https://www.firstinspires.org/resource-library/ftc/technology-information-and-resources>

## Using Vuforia Vumarks:

[https://www.firstinspires.org/sites/default/files/uploads/resource\\_library/ftc/using-vumarks.pdf](https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/using-vumarks.pdf)

## FTC Field Coordinate System:

[https://github.com/ftctechnh/ftc\\_app/files/989938/FTC\\_FieldCoordinateSystemDefinition.pdf](https://github.com/ftctechnh/ftc_app/files/989938/FTC_FieldCoordinateSystemDefinition.pdf)

# Q&A

- Questions?

# Demonstrations





# Introduction to Vuforia



**FIRST**  
LEGO  
LEAGUE JR.

**FIRST**  
LEGO  
LEAGUE

**FIRST**  
TECH  
CHALLENGE

**FIRST**  
ROBOTICS  
COMPETITION



## Introduction to Vuforia

- What is Vuforia?
- Programming Vuforia
- Using Vuforia in Rover Ruckus
- Vuforia Resources

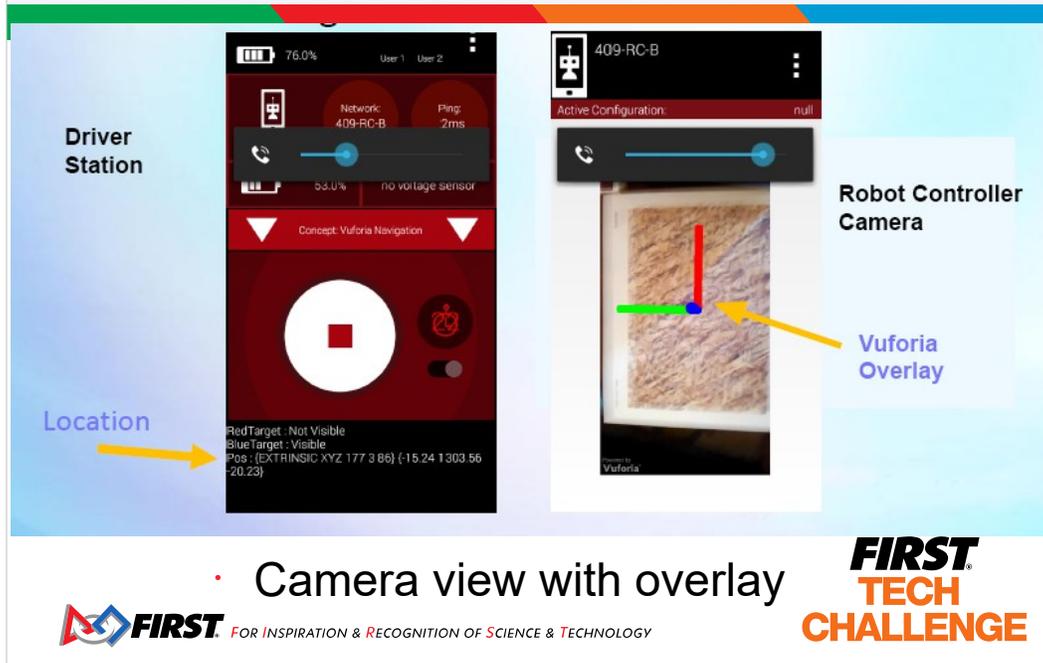
## What is Vuforia?

Vuforia is an augmented reality software development kit (SDK) for mobile devices that enables the creation of augmented reality applications. It uses computer vision technology to recognize and track Image Targets in real time.

This image registration capability enables developers to position and orient virtual objects, such as 3D models and other media, in relation to real world images when they are viewed through the camera of a mobile device.



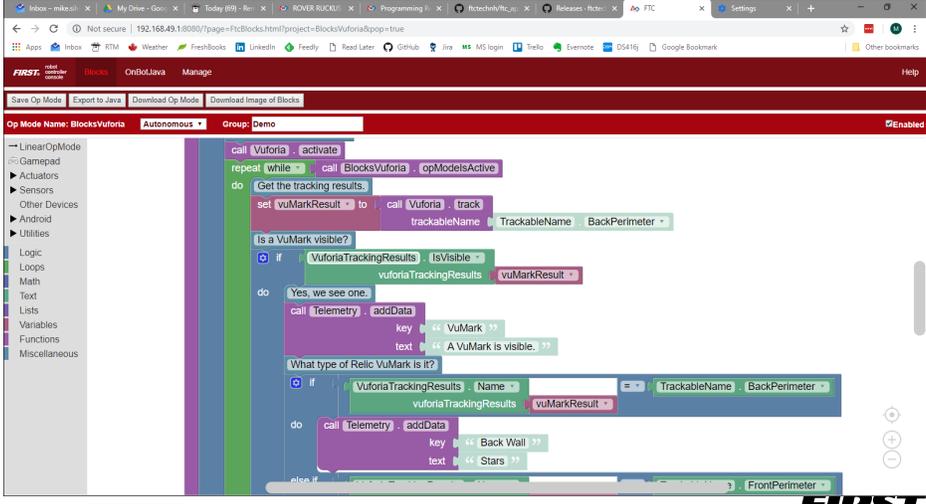
# Vuforia screen samples



The right hand screen image shows Vuforia adding x,y,z axis to the camera view on the robot controller.

Using Telemetry the program can display the robot location at the bottom of the driver station display.

# Blocks Programming Vuforia



The screenshot displays the FIRST Tech Challenge Blocks programming interface. The top navigation bar includes 'FIRST Tech Challenge', 'Blocks', 'OnBotJava', and 'Manage'. Below this, there are options to 'Save Op Mode', 'Export to Java', 'Download Op Mode', and 'Download Image of Blocks'. The main workspace shows a program titled 'Op Mode Name: BlocksVuforia' with a group of 'Demo' and a status of 'Enabled'. The program logic is as follows:

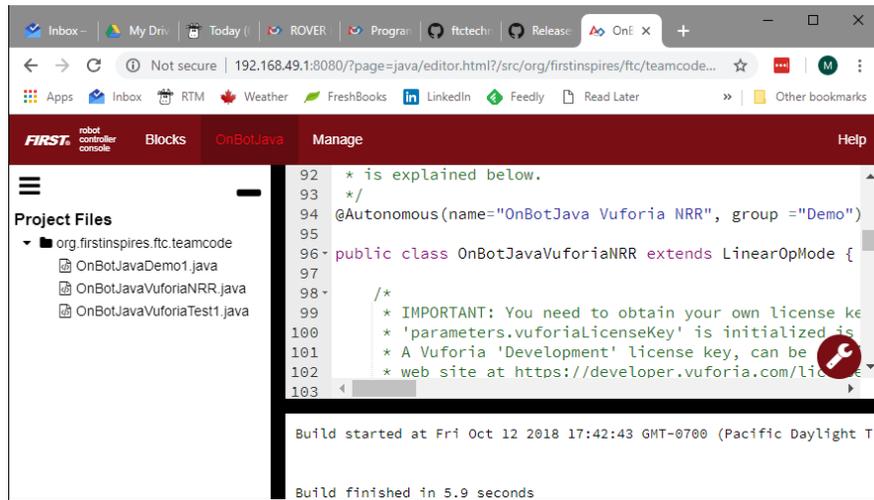
```
call Vuforia activate
repeat while call BlocksVuforia opModelsActive
do
  Get the tracking results.
  set vuMarkResult to call Vuforia track
  trackableName BackPerimeter
  Is a VuMark visible?
  if VuforiaTrackingResults isVisible vuMarkResult
  do
    Yes, we see one.
    call Telemetry.addData
    key VuMark
    text A VuMark is visible.
  What type of Relic VuMark is it?
  if VuforiaTrackingResults Name = TrackableName vuMarkResult
  do
    call Telemetry.addData
    key Back Wall
    text Stars
  else if FrontPerimeter
```

The FIRST Tech Challenge logo is visible in the bottom right corner of the screenshot.

If you use Blocks, if you want to use Vuforia, you should view the Java sample program called Concept Vuforia Nav Rover Ruckus. It has many comments that explain how to use Vuforia. Somethings like a Vuforia developers key and positioning of the image targets have been done for you by Blocks.

Show Concept VuMark Detection sample, as modified to work with Rover Ruckus.

# OnBotJava Programming Vuforia



The screenshot shows a web browser window displaying the OnBotJava interface. The browser's address bar shows the URL: `192.168.49.1:8080/?page=java/editor.html?src/org/firstinspires/ftc/teamcode...`. The interface has a dark red header with the text "FIRST robot controller console" and navigation tabs for "Blocks", "OnBotJava", and "Manage". On the left, a sidebar titled "Project Files" lists three files: `OnBotJavaDemo1.java`, `OnBotJavaVuforiaNRR.java`, and `OnBotJavaVuforiaTest1.java`. The main area is a code editor showing Java code for `@Autonomous` mode. The code includes a class `OnBotJavaVuforiaNRR` that extends `LinearOpMode`. Comments in the code mention the importance of obtaining a Vuforia license key and provide a URL: `https://developer.vuforia.com/licenses`. At the bottom of the editor, a status bar indicates "Build started at Fri Oct 12 2018 17:42:43 GMT-0700 (Pacific Daylight T1)" and "Build finished in 5.9 seconds".

 **FIRST** FOR INSPIRATION & RECOGNITION OF SCIENCE & TECHNOLOGY

**FIRST**  
**TECH**  
**CHALLENGE**

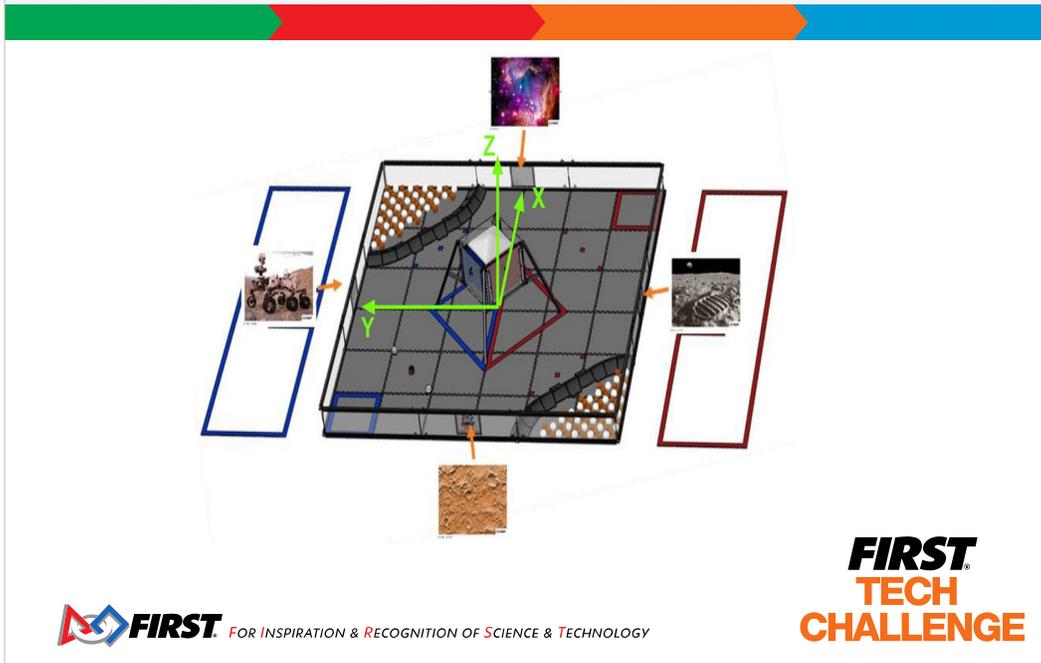
For most things Blocks and Java have the same functionality. However, I think there are more sample programs in Java.

Both Blocks and OnBotJava have tutorials in the FTC programming resources page at the FIRST website.

Show my `OnBotJavaVuforiaNRR` program which is based on the Vuforia Rover Ruckus sample.

Show my `OnBotJavaVuforiaTest1` program which drives the robot towards the back wall target.

# Rover Ruckus Field Coordinates



The FTC SDK makes use of something they call the Field Coordinate System, (see PDF in GitHub ftc\_app releases).

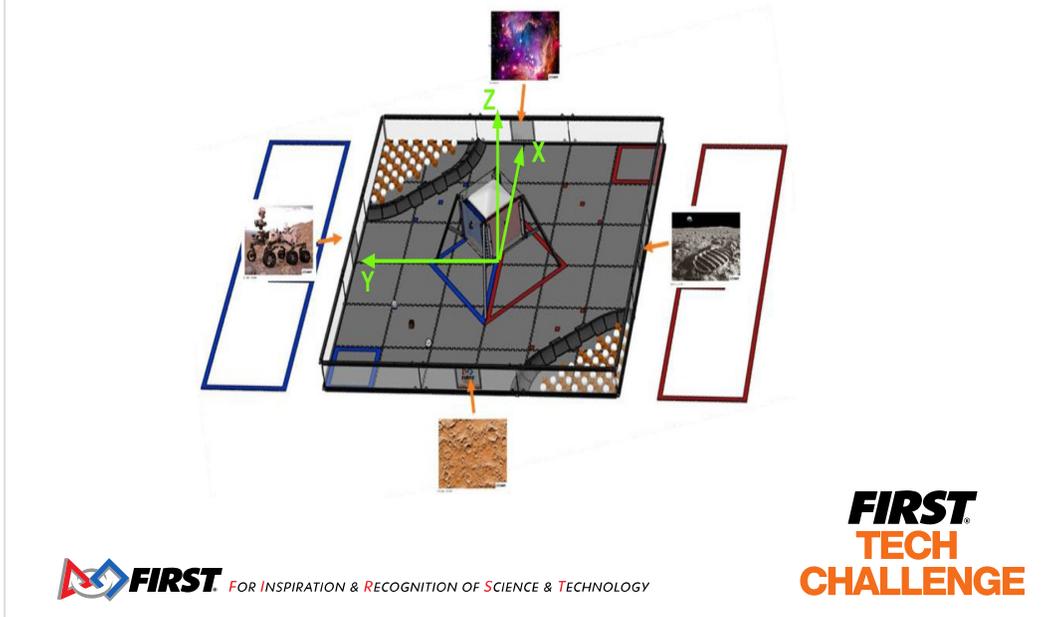
**Origin:** The 0,0,0 origin of the FTC coordinate system is the point in the center of the field, equidistant from all 4 perimeter walls (where the four center tiles meet). The origin point rests on the top surface of the floor mat.

**X Axis:** Looking at the origin from the RED WALL, the X axis extends through the origin point and runs to the right and left, parallel with the RED WALL. The X axis values increase to the right.

**Y Axis:** Looking at the origin from the RED WALL, the Y axis extends through the origin point and runs out and in, perpendicular to the RED WALL. Increasing Y values run out (away) from the RED WALL.

**Z Axis:** Looking at the origin from the RED WALL, the Z axis extends through the origin point and runs up and down in a vertical line. Increasing Z values extend upwards.

# Rover Ruckus – Vuforia Notes



So Vuforia can be used to locate your position on the field.

Pros

- you get a x,y position on the field in mm or In. Plus your orientation (rotation).
- you can drive based on the position or rotation depending on your needs.
- Ideally Vuforia is combined with other navigation techniques and used to aid or correct your driving.

Cons

- depending on your phone camera or web cam, the image targets typically have a range limit beyond which Vuforia can't find the target. The perimeter also reduces range.
- If the camera view doesn't include the target you don't get any information, you might need to add a web cam or two, or perhaps one that is on a servo so you can rotate it to face the target as you drive by.
- Using position to navigate might involve complex calculations.

# References

## Programming Resources

<https://www.firstinspires.org/resource-library/ftc/technology-information-and-resources>

## Using Vuforia Vumarks:

[https://www.firstinspires.org/sites/default/files/uploads/resource\\_library/ftc/using-vumarks.pdf](https://www.firstinspires.org/sites/default/files/uploads/resource_library/ftc/using-vumarks.pdf)

## FTC Field Coordinate System:

[https://github.com/ftctechnh/ftc\\_app/files/989938/FTC\\_FieldCoordinateSystemDefinition.pdf](https://github.com/ftctechnh/ftc_app/files/989938/FTC_FieldCoordinateSystemDefinition.pdf)

## Q&A

- Questions?

# Demonstrations



Demonstrate the various programs

- start with OnBotJavaVuforiaNRR to demonstrate the x,y positions
- Show range limit, try range with panel
- use OnBotJavaVuforiaTest1 to demonstrate autonomous driving