AAMC POST FLIGHT DIRECT REQUIREMENTS:

images collected in flight..map them with the gps location where the aircraft was and skew the image in horizontal plane. PPR3 is a box where the flights moves.They have an XML header.

Interface with Dr. Qi's project – Milestone!

Plug in SSD

Find the data

Click on the correct data set

Puts the images on the Esri basemap

Requires camera info

must satisfy lan's needs.

Things needed to fly the plane: altitude, drone..Current s/w is dying cuz not being updated. Get data from servers to https for better security. – How to get this? Ask Jo!!

Technologies ised: Java, SDK

Tasks to do: Rescope

1. Pre mission
2. Post mission

Based on what camera u use, the image u capture.

Code snippets

Setting the image:

```
    protected static Image getImage(String name)
    {
        for(Image a : images)
        {
            if(a.getFileName().equals(name))
            {
                return a;
            }
        }
        return null;
    }

    public static void setImages(DataGetter getter)
    {
        getter.getBaseMap();
        images = getter.getImages();
    }
    public static void setImageVisibility(Image i, boolean visibility)
    {
        images.get(images.indexOf(i)).setVisible(visibility);
    }

    public static ArrayList<Image> getImages()
    {
        return images;
    }

    public static void warpImages(Reference ref)
    {
        for(Image i : images)
        {
            ref.imageWarp(i, null);
        }
    }
}
```

Capturing the image of the plane in horizontal view:

```
    private ArrayList<Image> parseCapture(Element captureElement)
    {
        ArrayList<Image> captureList = new ArrayList<Image>();
        NodeList nList = captureElement.getElementsByTagName("GPhotoImage");

        for (int index = 0; index < nList.getLength(); index++)
        {
            captureList.add(parseImage((Element) nList.item(index)));
        }

        return captureList;
    }

    private Image parseImage(Element imageElement)
    {

        AerialImage theImage = new AerialImage();
        theImage.setIMUData(parseIMUData(imageElement));
        theImage.setCam(parseCamera(imageElement));
        theImage.setFileName(imageElement.getAttribute("ImageName"));
        // TODO: DIRECT-load images
        return theImage;
    }
```

Camera

```
public Camera(File file) throws FileNotFoundException, IOException
{
    if(FilenameUtils.getExtension(file.getName()).equals("cal"))
    {
        fromCalFile(file);
    }
}
```

Elevation:

```
private void getElevationFromRaster(Point latLong)
{
    IdentifyParameters identifyparam = new IdentifyParameters();
    identifyparam.setGeometry(latLong);
    identifyparam.setMapExtent(map.getExtent());
    identifyparam.setSpatialReference(map.getSpatialReference());
    identifyparam.setMapHeight(map.getHeight());
    identifyparam.setMapWidth(map.getWidth());
    identifyparam.setLayerMode(IdentifyParameters.ALL_LAYERS);
    identifyparam.setDPI(ArcGISRuntime.getDPI());

    IdentifyTask task = new IdentifyTask(dynamicLayer.getUrl());
    try
    {
        IdentifyResult[] results = task.execute(identifyparam);
        System.out.print(results[0].getGeometry());
    } catch (Exception e)
    {
        e.printStackTrace();
    }
}
```

Pre Flight:

Generate Waypoins, Flight lines, and Survey Areas.

Draw them on ESRI map.

Generate boundaries and draw them.

Import/Export to the following formats of XML(import optional)

PPRZ

KML(and convert to KMZ)

FPL

The ability to save.

The ability to edit waypoints (and thus flightlines and survey areas) graphically and textually.

Origin point(Zoom in)

Area of study.

Zoom to area on import

Ground Elevation

Change flight altitude.

Setup the pre flight code on laptop.

Milestones: Did not setup ESRI Account.

```java
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.stage.Stage;

import com.esri.arcgisruntime.mapping.ArcGISMap;
import com.esri.arcgisruntime.mapping.Basemap;
import com.esri.arcgisruntime.mapping.view.MapView;

public class AggieAirClass extends Application {
    private MapView mapView;
  @Override
  public void start(Stage stage) throws Exception {
    StackPane stackPane = new StackPane();
    Scene scene = new Scene(stackPane);
    stage.setTitle("Display Map Sample");
    stage.setWidth(800);
    stage.setHeight(700);
    stage.setScene(scene);
    stage.show();
    ArcGISMap map = new ArcGISMap(Basemap.createImagery());
    mapView = new MapView();
    mapView.setMap(map);
    stackPane.getChildren().addAll(mapView);
  }
  @Override
  public void stop() throws Exception {

    if (mapView != null) {
      mapView.dispose();
    }
  }
  public static void main(String[] args) {

    Application.launch(args);
  }
```

Load flight image:

```java
    public void loadImage(String imagePath)
    {
        RGBmat = Imgcodecs.imread(imagePath, Imgcodecs.IMREAD_COLOR);
        grayMat = Imgcodecs.imread(imagePath, Imgcodecs.IMREAD_GRAYSCALE);
        originalSize = new Point(grayMat.rows(), grayMat.cols());
        setVisible(true);
    }
```