### **School of Electronics and Computer Science**

### **University of Southampton**

### 

### **Runway Redeclaration Tool - Increment 3**

### **COMP2211 Software Engineering Group Project**

Team 10: Andrew John Evans (aje3g16)

Lyubomir Tsvetanov Hristov (lth1u17)

Suyash Datt Dubey (sdd1n17)

Kemal Yesildagli (ky1g17)

Denis Kovachev (dk1g17)

**Responses to feedback**

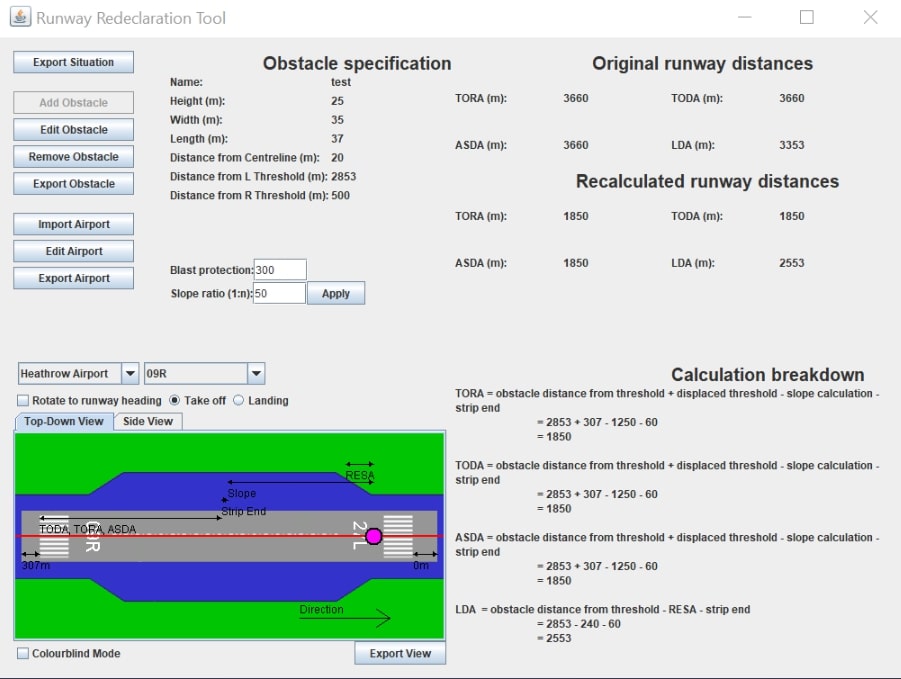
* Margins around the edge of the GUI
  + Added small margins to the edge of the various dialogs and screens to improve visibility of the GUI.
* Scenario testing
  + Scenario 1: A user opens the application, and chooses a runway. He then proceeds to add an obstacle using the “Add Obstacle” button. He creates a new obstacle with a specific set of parameters and adds it to the current runway. The user then clicks on the side view tab in the visualisation corner and the exports the Image as a JPEG format.
  + Scenario 2: A user opens the application, and chooses a runway. He then proceeds to add an obstacle using the “Add Obstacle” button and chooses one of the preset obstacles. He then adds the obstacle to the current runway and gets a notification that he has been successful at doing so. The user then navigates to the top-view tab and rotates to the runway heading. He can then uses the mouse and mouse wheel to zoom and pan the image to get a better idea of the current situation.
  + Scenario 3: A user opens the application, and chooses a runway. He then proceeds to add an obstacle using the “Add Obstacle” button and chooses one of the preset obstacles, after which he adds the obstacle to the current runway. User is colourblind (tritanopia) and so he selects the colourblind mode, which changes the colour scheme to from green and blue to yellow and blue. That helps the user comprehend the information better.
* Specify testing for each requirement
  + Feedback is taken into account. In the testing section of the report, we’ve specified the testing performed for each requirement for this increment (and testing performed on requirements from previous increments).
* Validation of the XML file imports against an XSD Schema
  + Implemented the functionality of checking the validity of the structure of XML files before importing the airport into the application.

**Discussion of key design choices / decisions**

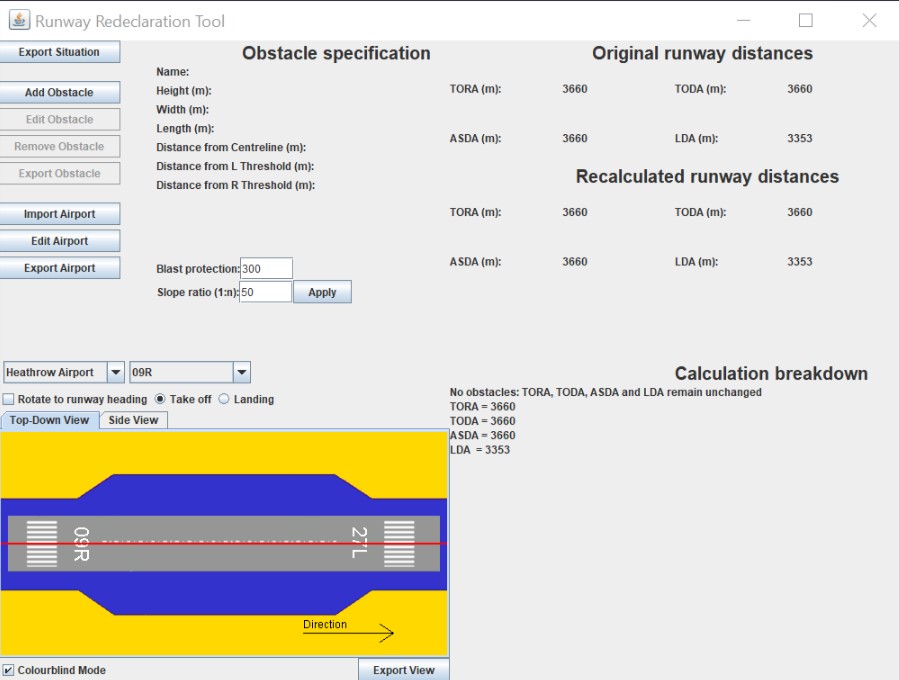
* Top-down Visualisation
  + We have implemented the top-down view, which shows the current runway, the actual obstacle on it and the different distances. The design of the view has been chosen so that it is nice and intuitive for the user. The colours have also been carefully chosen and moreover we have added a mode specifically for colour blind users that changes some of the colours. Additional functionalities like zooming, panning and rotating can be used on the top view as well.
  + We chose to have limited colours, but made sure that all the information was clear to all users.
* Graphic Threshold orientation
  + We made sure that in our views that the lower threshold is always displayed on the left of the screen (unless rotated to runway heading).
    - We misunderstood the requirement but it became clear when implementing the views the actual meaning of the requirement and made sure we complied with it.
* Graphic Runway Strip Rotation
  + We added the “Rotate to runway heading” checkbox, which when selected rotates the orientation of the runway in the top view, to its compass heading.
  + Performed acceptance testing on this to ensure that it rotated correctly.
* GUI Notifications
  + We have added notifications for every action the user can make in order to give him/her information about whether it has finished successfully.
  + Acceptance testing to ensure that all situations had been accounted for.
* Ability to zoom and pan the views
  + We implemented pan with mouse drag and zoom with scroll onto both views.
  + We decided to remove the button controls on the view, to reduce clutter on the view and make the information more clear.
* Accessibility for people with colourblindness
  + We began by identifying the types of colourblindness from online sources (e.g <https://nei.nih.gov/health/color_blindness/facts_about>).
  + Our application had limited use of colour, mainly limited to the top view (in which we planned to have 2 colours). As such, we chose a pair of colours that can be distinguished easily in all situations - in our case, yellow and blue).
  + Based on this research, we ascertained that this colour scheme would be adequate for accessibility.
  + We also checked our obstacle colouring (magenta) and ascertained that it’d be visible easily by all types of colourblindness against the grey runway strip.
  + Added a toggle between the colourblind colour scheme and the regular colour scheme. The view is usable by all in both modes, but the user may have a preference to which one they used.

**Key storyboards and screenshots**

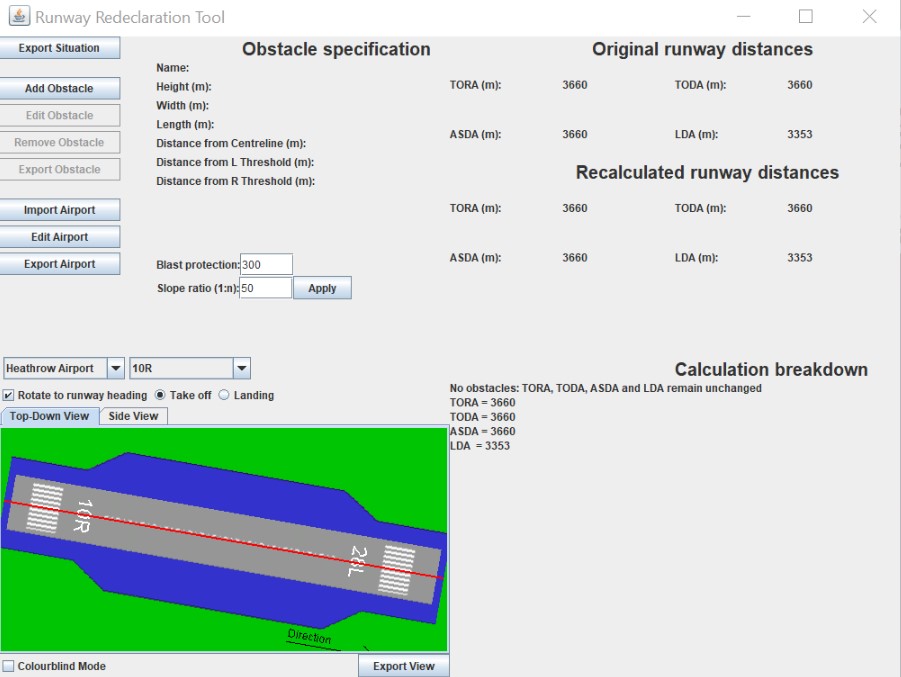
**1. Top-View**

****

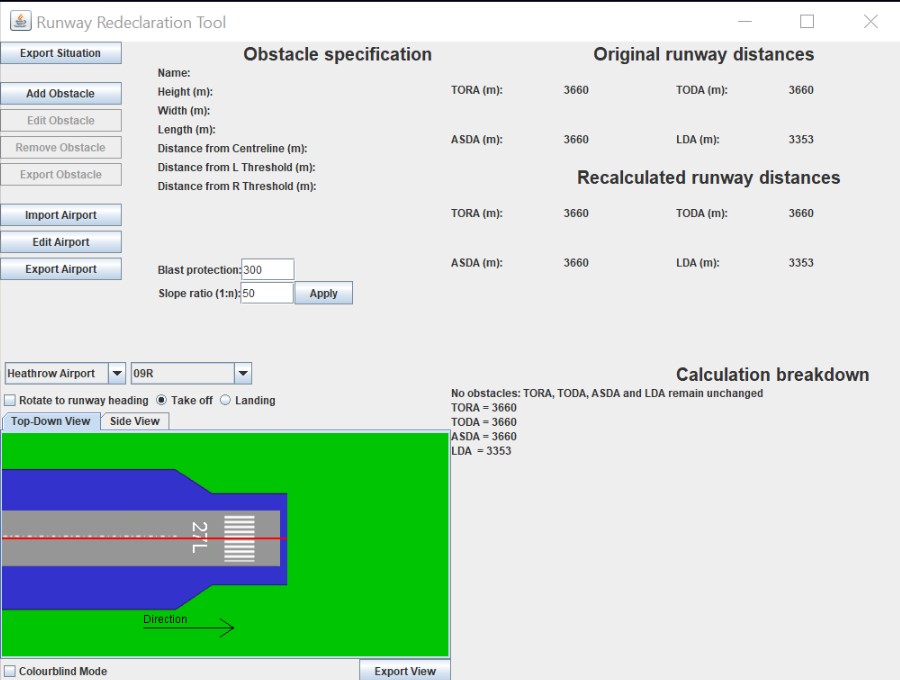
**2. Colourblind Mode**

****

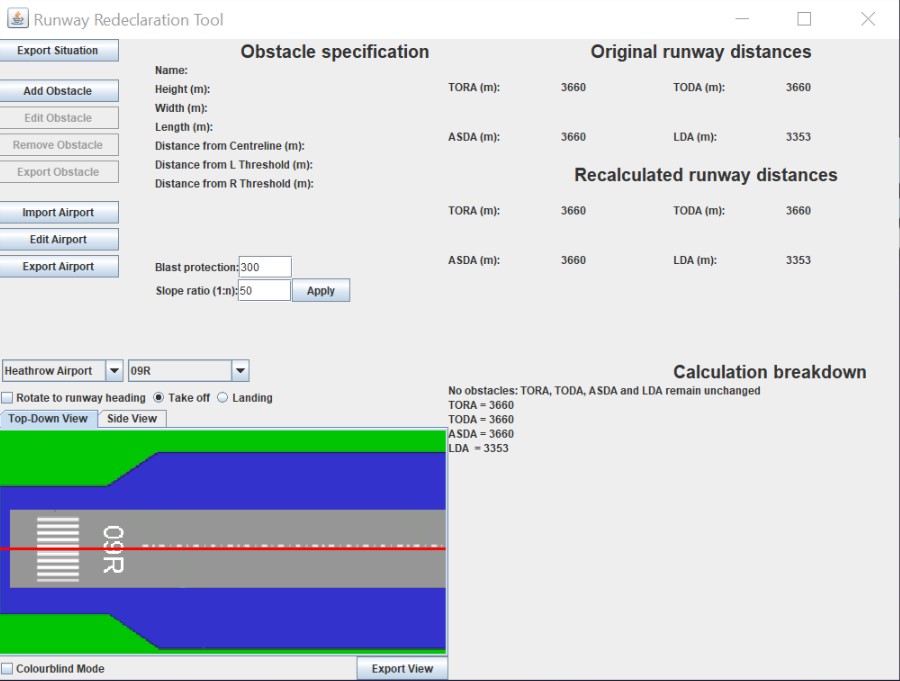
**3. Rotation of Top-View**

****

**4. Panning of Top-View**

****

**5. Zooming of Top-View**

****

**Testing during this increment (including retesting of components from previous increments)**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Testing type** | **Description of testing** |
| Side view | Regression testing/Acceptance testing | Retested the views and noticed some errors when choosing different scenarios. Fixed, re-tested and confirmed that side view now satisfies the requirement. |
| Top view | Acceptance testing | Tested against the same scenarios the side view was tested and checked whether all distances are correct and everything is displayed.Accuracy of the distances were not vital but we made sure that the distances were representative of the actual distances. |
| Accessibility for users with colour blindness | Acceptance testing | Once implemented, we revisited our research and checked that the colour schemes chosen would make for easy use by all individuals, regardless of colour blindness. |
| View rotation | Acceptance testing | Manually checked if different runways were being rotated appropriately with respect to their compass headings when the checkbox was selected. |
| Graphic threshold orientation | Acceptance testing | Checked that in all scenarios (8 scenarios per view) that the lower of the thresholds is on the left of the view (when not rotated). |
| Pan/zoom views | Acceptance testing | Once implemented, we tested zooming in and out for both views worked, scaling appropriately. Similarly, for panning we panned the views, to make sure the views were translated appropriately. |

We also re-ran tests from previous increments. For example, we performed regression testing by re-running the unit tests for the calculations written in Increment 1. Similarly, we performed boundary testing again on the input components of the GUI to check that the sanitisation of inputs was still present and correct. Acceptance testing on the views and GUI elements were re-performed in this increment, to ensure we are meeting all the criteria given to us in the specification.

We performed integration testing of the complete product and tested the jar in its final state (noticing some severe problems with our build procedure, which we rectified). We noticed some problems with the functionality that we could not rectify (in time), and made notes of these in the user manual.

**Sprint 3 burndown chart (Key)**

**8:** Graphic Threshold orientation

**9:** Graphic Runway Strip Rotation

**12:** GUI Notifications

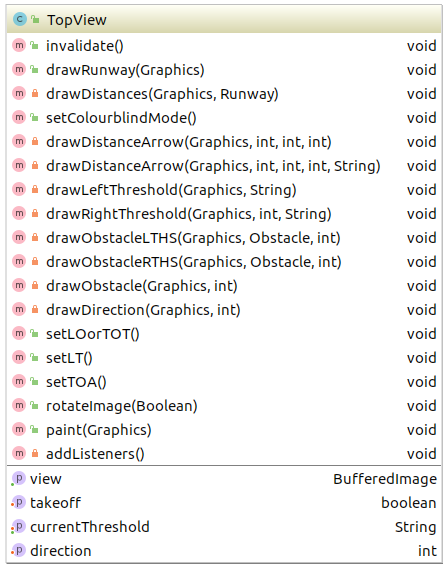
**13:** Top-down visualisation

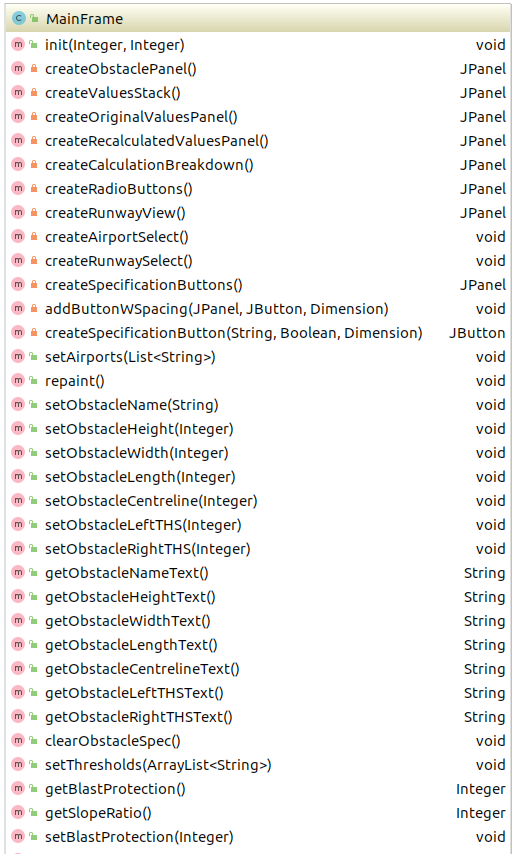
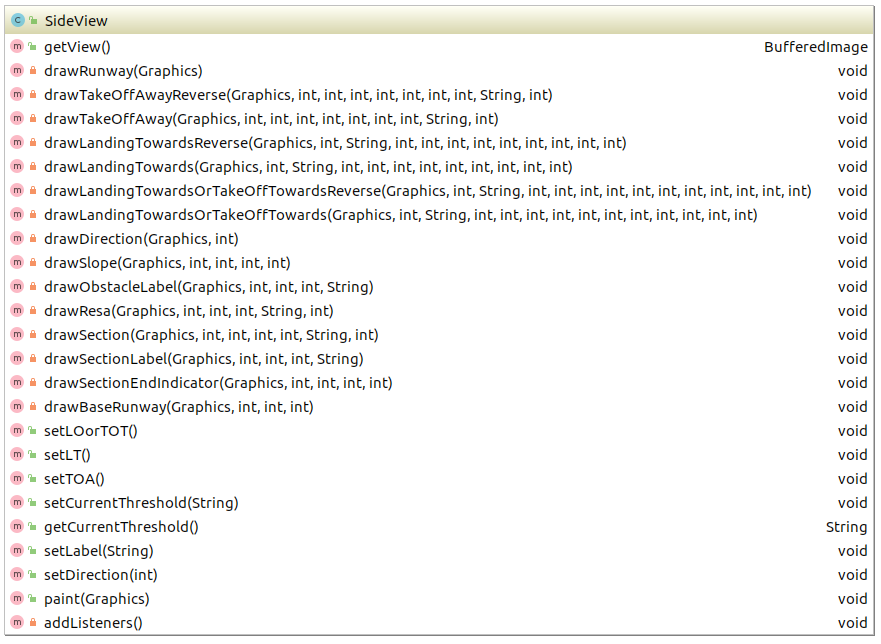
**15:** Ability to zoom and pan the views

**18:** Accessibility for colour blind people

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tasks | 15 | 15 | 15 |  | | | | | | | | | | | |
| 15 | 15 |  | | | | | | | | | |
| 18 | 18 | 18 |
| 8 | 8 | 8 | 8 | 8 | 8 |  | | | | | | | | |
|
|
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |  | | | | |
|
|
| 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |  | | |
| 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |  |
|
|
|
| Day | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

**UML Diagrams**

****

****