COMP2211 Deliverable 3 - Project Envisioning Report
Group 23

Andrew Sansum (ajs2g20@soton.ac.uk)

Patrik-Tibor Csanyi (ptc1g20@soton.ac.uk)

Guillaume Comet-Vernet (gcv1u20@soton.ac.uk)

Madhav Muralikrishnan (mkm1g20@soton.ac.uk)

Adam Clarke (ac4g20@soton.ac.uk)

David Stefanov (ds1u20@soton.ac.uk)

Introduction

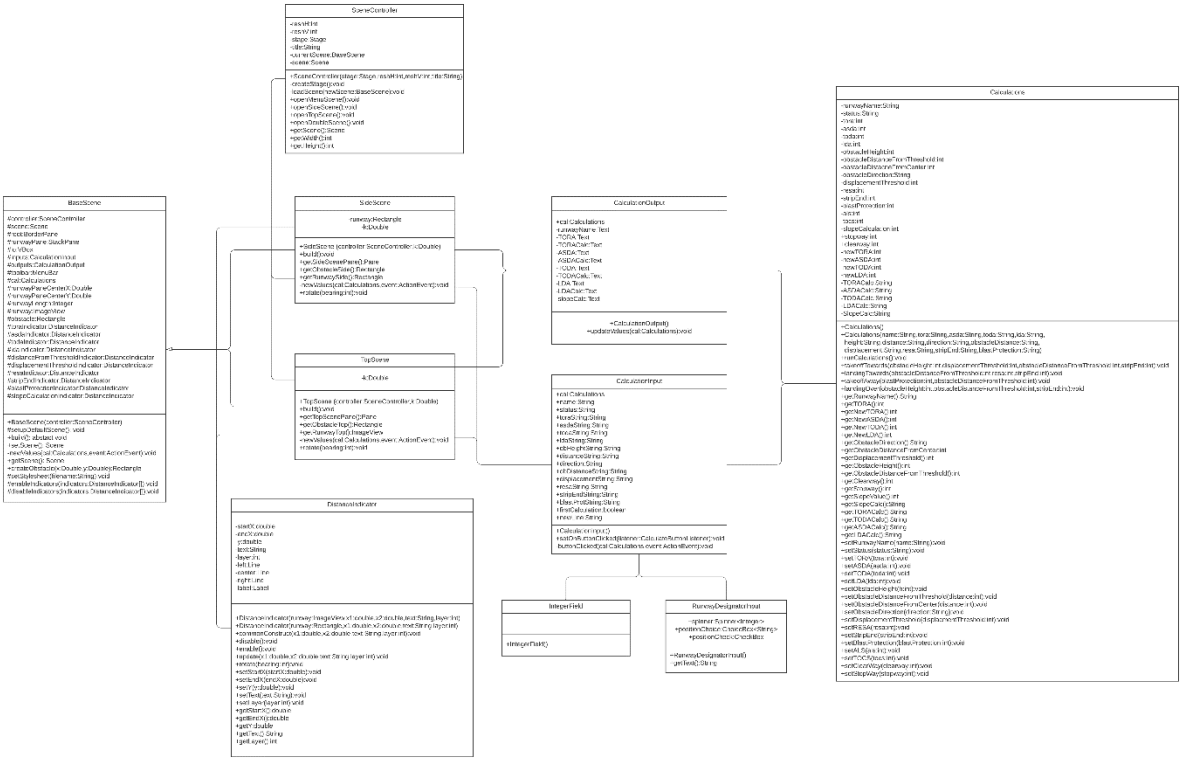
In the second sprint, our main aim was to finish the tasks we didn't finish in the previous sprint which mainly consisted of adding or fixing visual components. After, we focused on getting the majority of our calculations done, consequently giving us a basic working prototype of our application. Furthermore, we added indicators onto the runway.

We have omitted the Top & Side View due to bugs that we caught late into the sprint and thus constrained by time. We have also found that the calculated results can be concealed by the window size. Moreover, the distance indicators goes beyond the runway when it should be limited to the runway.

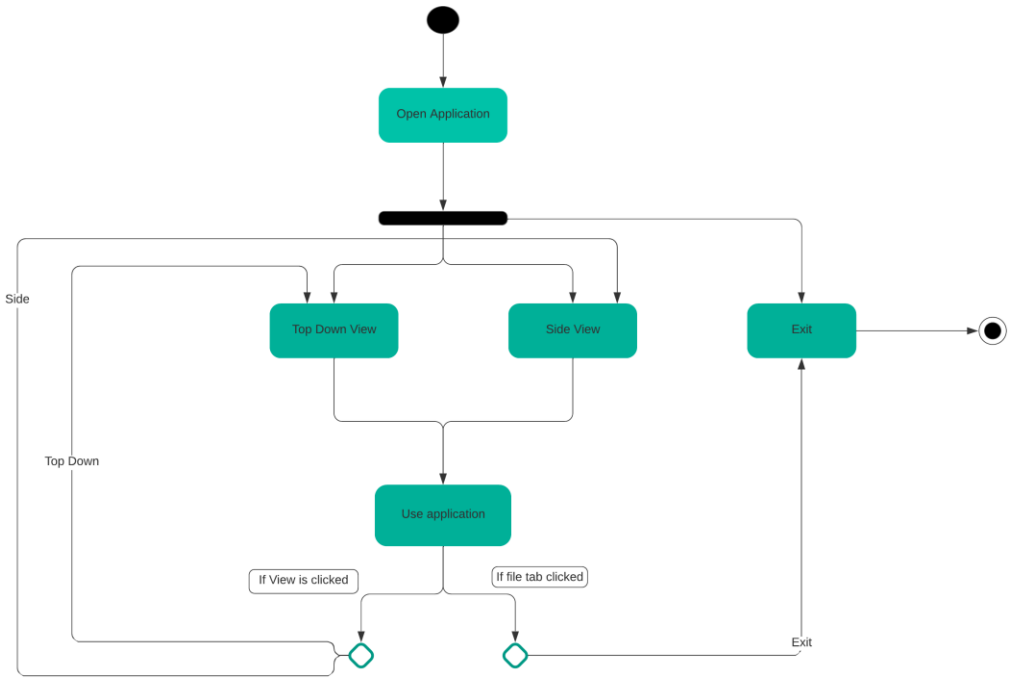
In the next sprint, our focus will be to fix these aforementioned issues as fast as possible in addition to creating features such as printing results in different formats, zooms, maps, etc. The tasks we will complete can be seen in the 3rd Sprint backlog.

UML Diagrams

1.Class diagrams



2.Activity Diagram



Logbook

Feedback after the envisioning:

During the review session we received feedback on numerous aspects of our code. Firstly, we were told to have significantly more tests prepared for the next submission, including tests of edge cases.

In terms of diagrams, we were advised to indicate the Model-Control-View scheme that we used with boxes or other such methods. With regards to the activity diagram, we were instructed to make it more detailed, as in its current form it only displays how the user can change views. In the next submission, it will also include how the code reacts to different types of input.

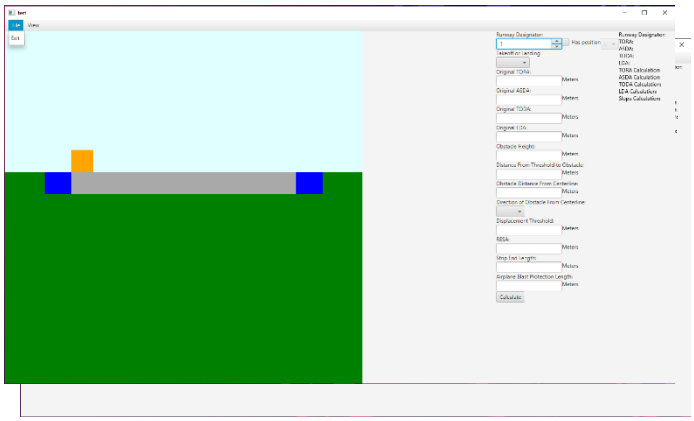
In terms of the actual code, we were advised to remove the main menu/menu scene, as it served little purpose. We were also told to send some sort of warning to the user in case they try to input negative values, and to not allow this behaviour.

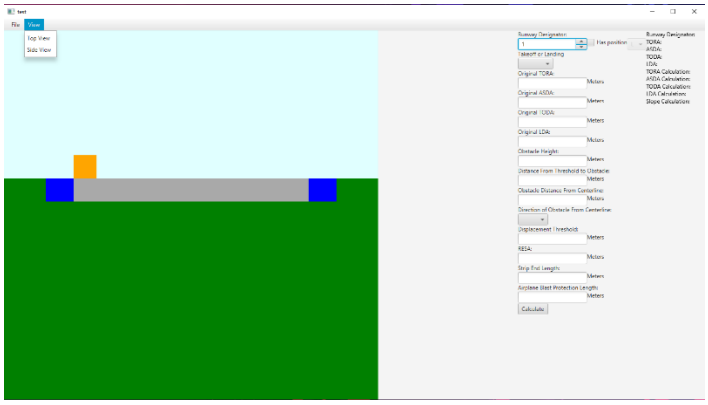
With regards to the burndown chart, while the previous changes received positive feedback, they lacked an indicator (a line) which would show the ideal progress of the sprint.

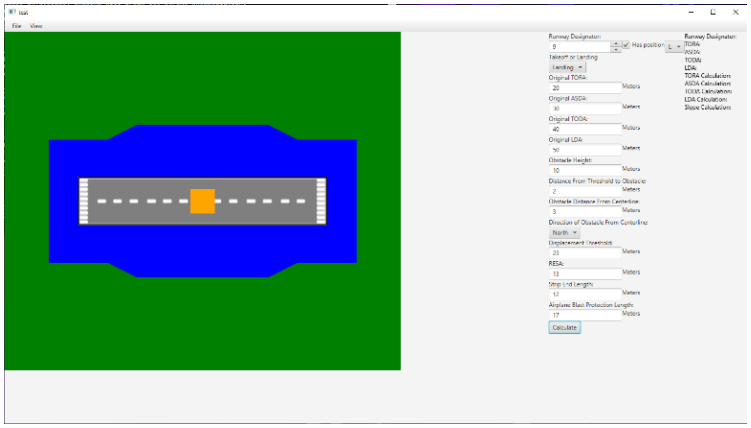
Scenarios

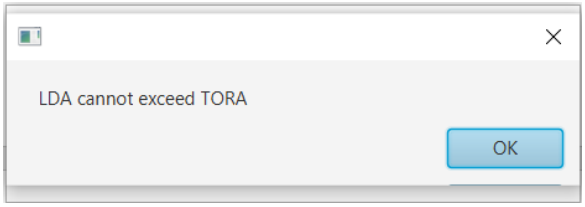
1. Jaden - aerodrome controller

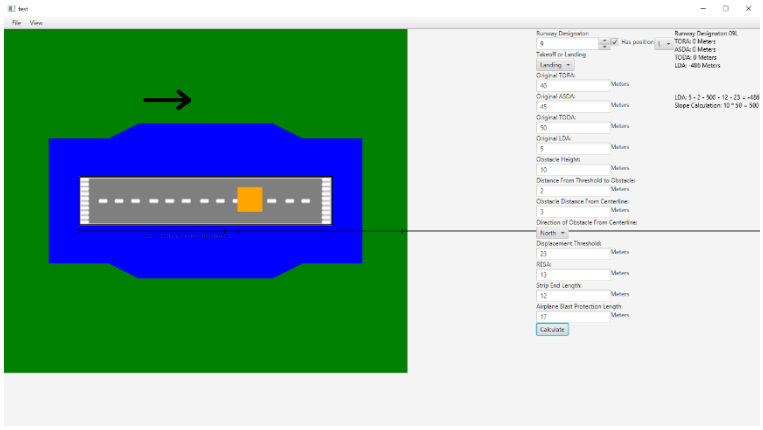
- Jaden opens the runway redeclaration app
- The home page is displayed
- Jaden can select between seeing the top-down, the side view or exiting
- Jaden chooses the top-down view
- A top-down view page is displayed
- Jaden enters the values of a runway with an obstacle
    - He accidently enters the value of LDA to be larger than the TORA
    - After pressing the Calculate button alert pops saying “LDA cannot exceed TORA”
    - He reenters the correct value
    - He presses the Calculate button
- Jaden sees exact top-down visualization of the runway
- Jaden clicks “View” at the toolbar and then clicks “SIde View”
- A side view page is displayed
- Jaden once again enters the same correct values
- A side view of the runway is visualized
- The indicators show the exact values of the runway





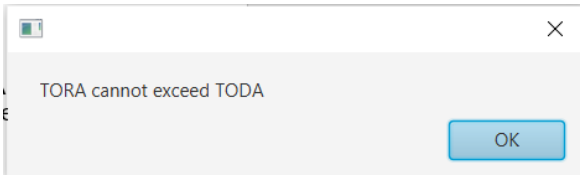



LDA cannot exceed TORA
OK



2. Jane - approach controller

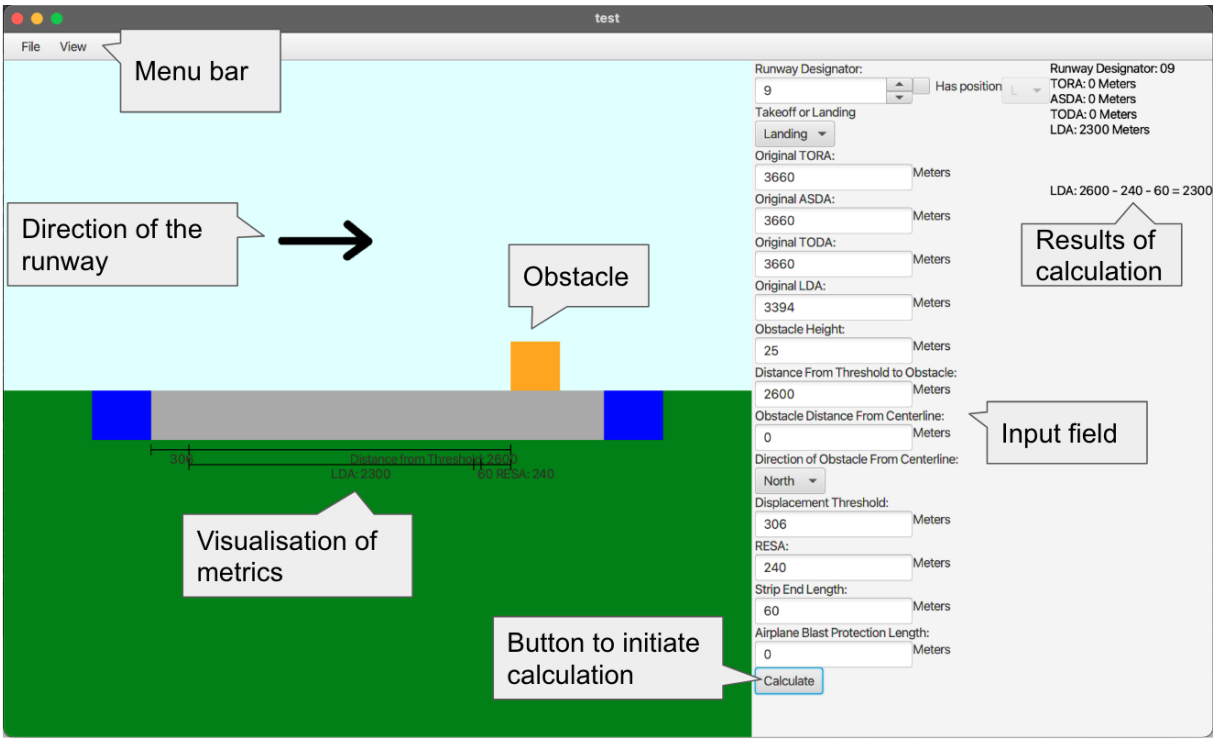
- Jane opens the runway redeclaration app
- The home page is displayed
- Jane can select between seeing the top-down, the side view or exiting
- Jane chooses the side view
- A side view page is displayed
- Jane enters the values of a runway with an obstacle
    - She accidently enters the value of TODA to be larger than the TORA
    - After pressing the Calculate button alert pops saying “TODA cannot exceed TORA”
    - She reenters the correct value
    - She presses the Calculate button
    - Jane sees how much of the runway is usable and in what direction is the plane coming/going

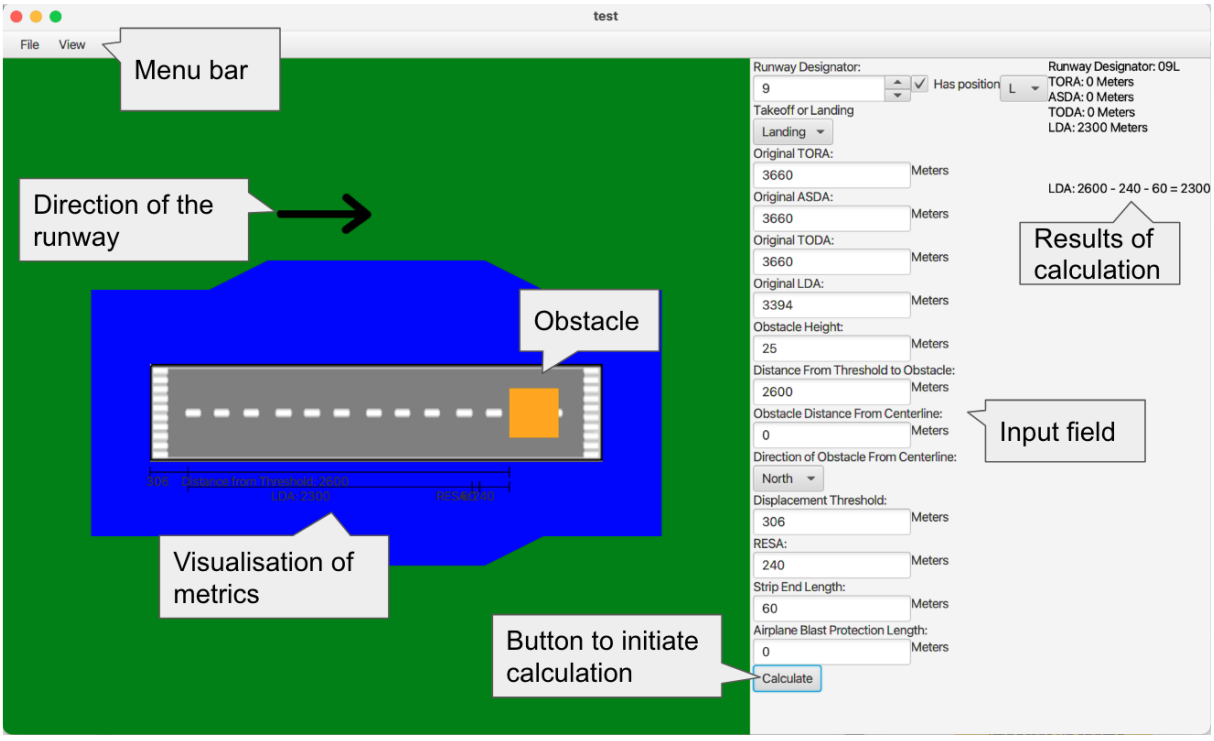


3. Gemma - independant calculator

- Gemma opens the runway redeclaration app
- The home page is displayed
- Gemma can select between seeing the top-down, the side view or exiting
- Gemma chooses the side view
- A side view page is displayed
- Gemma enters the values of a runway with an obstacle
    - She presses the Calculate button
- Gemma checks all the newly displayed values
- Gemma clicks “File” at the toolbar and then exit
- The application closes

Storyboards





Backlog

Second Sprint Backlog:

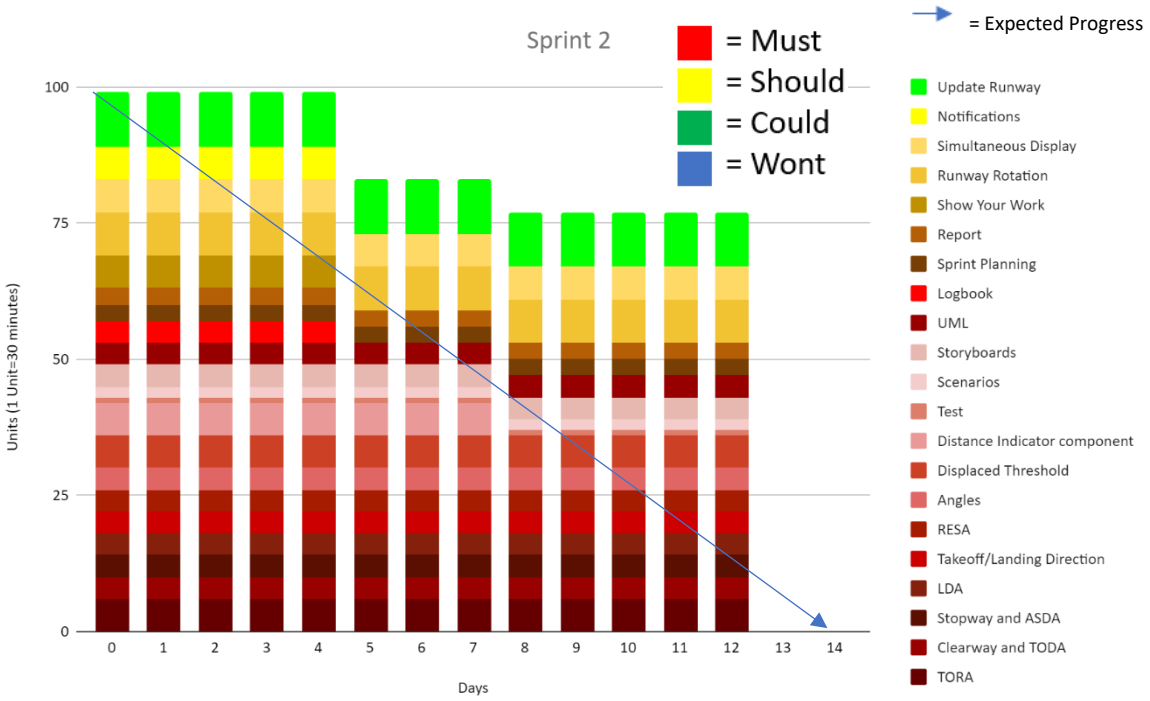
Task ID	Name	Description	Workforce	Estimated Difficulty	Actual Difficulty	Date Of Finish
0	Descriptions	Short Description of the Task	People undertaking the task	In units (1 unit ≈ 30 mins)	How long it actually took	Date on which task was finished
2	Simultaneous display	Side-on and Top-Down visualisations can be displayed simultaneously	Guillaume	6		
4	TORA	Runway strip displayed in both views and TORA indicated	Adam	6	2	23.03.2022
5	Clearway and TODA	Clearway and new TODA displayed on both views	Guillaume, Adam	4	2	23.03.2022
6	Stopway and ASDA	Stopway and new ASDA displayed on both views	Guillaume, Adam	4	2	23.03.2022
8	LDA	LDA calculated and displayed on both views	Guillaume, Adam	4	2	23.03.2022
9	Takeoff/Landing Direction	Takeoff and landing directions are displayed on both views	David, Adam	4	2	23.03.2022
10	RESA	RESA displayed on both views	David, Adam	4	2	23.03.2022
11	Angles	Offsets caused to ALS and TOCS calculated and displayed on side-view	Adam	4	2	23.03.2022
12	Displaced threshold	Displaced threshold calculated and displayed on both views	David, Adam	6	2	23.03.2022
18	Show Your Work	Calculation steps displayed on request	Adam	6	3	15.03.2022
20	Runway Rotation	Automatically rotate runway so it matches compass direction	Adam	8	2	23.03.2022
21	Update Runway	Views update in real time if new runway, obstacle, data selected or changed		10		23.03.2022
22	Notifications	Notifications must be given to user whenever data changes	Patrik	6	6	15.03.2022
92	Distance Indicator Component	UI Component for distance idicators	Adam	6	6	18.03.2022
93	Tests	Write tests for current code	Andrew	6		23.03.2022
94	Scenarios	Update scenario for each persona	David	2		23.03.2022
95	Storyboards	Update storyboard	Madhav	4		23.03.2022
96	UML	Update UML diagrams	Madhav	6	8	23.03.2022
97	Logbook	Write the logbook for the sprint	Patrik	4	2	15.03.2022
98	Sprint Planning	Create/Update sprint backlogs and burndown charts	Patrik, Madhav	3	4	23.03.2022
99	Report	Write the report for the current increment	Madhav	3	3	23.03.2022
100	Overall difficulty	Overall difficulty of current sprint		106	50	
MoSCoW Prioritisation for the Sprint:		Must	Should	Could	Won't	

Third Sprint Backlog:

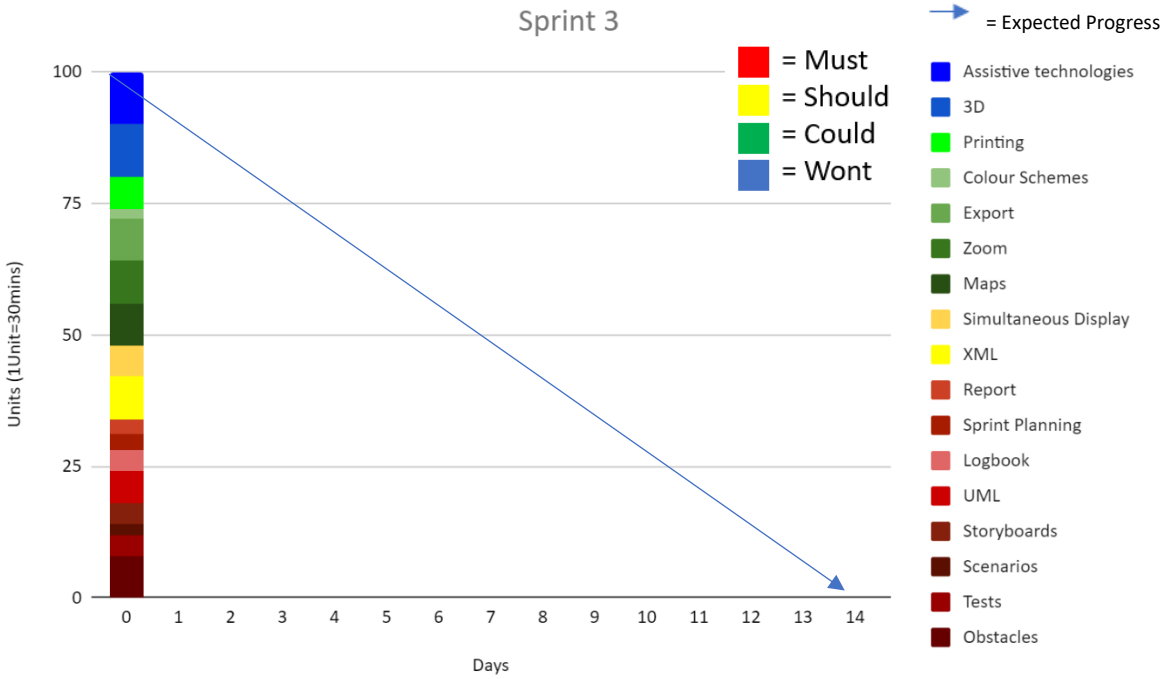
Task ID	Name	Description	Workforce	Estimated Difficulty	Actual Difficulty	Date Of Finish
0	Descriptions	Short Description of the Task	People undertaking the task	In units (1 unit ≈ 30 mins)	How long it actually took	Date on which task was finished
2	Simultaneous display	Side-on and Top-Down visualisations can be displayed simultaneously		6		
16	Obstacles	Pre-defined obstacles implemented and usable		8		
19	XML	Import and export details of obstacles and airports as well as results from files (XML)		8		
23	Maps	Overlay real world images on display		8		
24	Zoom	The ability to zoom and pan views		8		
25	3D	3D Visualisation		10		
26	Export	Export display in JPEG, PNG, etc.		8		
27	Assistive technologies	API support for assistive technologies		10		
28	Colour Schemes	Alternative colour schemes present		2		
29	Printing	Ability to print out results		6		
93	Tests	Write tests for current code		4		
94	Scenarios	Update scenario for each persona		2		
95	Storyboards	Update storyboard		4		
96	UML	Update UML diagrams		6		
97	Logbook	Write the logbook for the sprint		4		
98	Sprint Planning	Create/Update sprint backlogs and burndown charts		3		
99	Report	Write the report for the current increment		3		
100	Overall difficulty	Overall difficulty of current sprint		94		
MoSCoW Prioritisation for the Sprint:		Must	Should	Could	Won't	

Burndown Charts

Second Sprint burndown chart:



Third Sprint burndown chart:



Tests

Unit Tests for Calculations Class

Pre-calculation Input Value Ranges:

Ranges for the numeric parameters before calculations are as follows:

Parameter	*Lower Bound*	*Upper Bound*
TORA	0	--
ASDA	0	--
TODA	TORA	--
LDA	0	TORA *(inclusive bound)*
Obstacle Height	0	--
Obstacle Distance from Threshold	--	--
Threshold Displacement	0	TORA
RESA	240	TORA
Strip End	60	TORA
Blast Protection	0	TORA

NOTE: Lower bounds are inclusive and upper bounds are exclusive unless otherwise specified.

The only non-numeric parameter is Status which must be either of the strings “Landing” or “Takeoff”.

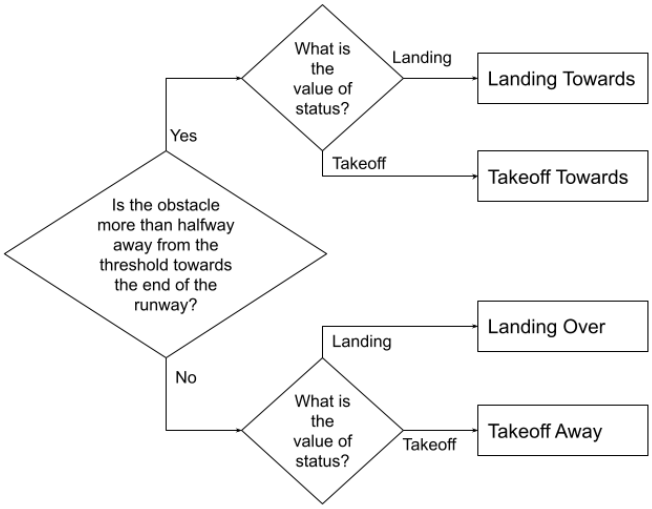
Tests will assert that the class throws an exception on values out of these ranges and that no exception is thrown on valid ones. Partitions for these tests will be erroneous values from outside the ranges, border values on the ranges, and regular values from within the ranges.

Calculation Cases:

There are four major cases for calculations:

- Landing over the obstacle
- Landing towards the obstacle
- Takeoff towards the obstacle
- Takeoff away from the obstacle

The diagram below illustrates the logic behind choosing a case:



Tests will ensure that the calculations chooses the correct case based on its parameters. The cases can be differentiated through the results of the calculations. Values changed for each case will be the 'object distance from threshold' and 'status'.

Object Distance	**Status**	**Expected Case Chosen**
-50	Landing	Landing Over
100	Landing	Landing Over
1800	Landing	Landing Over
3500	Landing	Landing Towards
3700	Landing	Landing Towards
-50	Takeoff	Takeoff Away
100	Takeoff	Takeoff Away
1800	Takeoff	Takeoff Away

3500	Takeoff	Takeoff Towards
3700	Takeoff	Takeoff Towards

NOTE: The assumed TORA here is 3600 i.e. the halfway point is 1800

Results Bounds:

For all four declared distances (TORA, TODA, ASDA and LDA), the respective redeclared distances may exceed the original. In the case that they do, the original should be returned instead.

Redeclared distances exceeding the original are a result of the absolute value of the obstacle distance being large. It is intended behaviour that the obstacle does not have to be on the runway. How this affects calculations is different for each of the four major cases.

Landing Over:

In this case we calculate the LDA with:

LDA1= LDA0- RESA - Strip end - Obstacle distance

With this, if the obstacle distance is negative and |obstacle distance| > RESA + Strip end then the new LDA will be larger. Cases to test here are:

- |obstacle distance| > RESA + Strip end
- |obstacle distance|= RESA + Strip end
- |obstacle distance| < RESA + Strip end

Landing Towards:

In this case, we calculate LDA with:

LDA1= Obstacle distance - RESA - Strip end

As we are landing towards, we know the obstacle distance is positive and larger than half the TORA. If Obstacle distance > LDA0 +RESA + Strip end then the new LDA will be larger. Cases to test here are:

- Obstacle distance > LDA0 +RESA + Strip end
- Obstacle distance= LDA0 +RESA + Strip end
- Obstacle distance < LDA0 +RESA + Strip end

Takeoff Towards:

In this case, we calculate TORA with:

TORA1=TORA0-RESA - Strip end - Obstacle distance

New ASDA and TODA simply return the new TORA. Similar to landing over, if the obstacle distance is negative and |obstacle distance| > RESA + Strip end then the new TORA will be larger. Cases to test here are:

- |obstacle distance| > RESA + Strip end
- |obstacle distance|= RESA + Strip end
- |obstacle distance| < RESA + Strip end

Takeoff Away:

Calculations for this case are:

- TORA1=TORA0-Blast protection - Obstacle distance
- ASDA1=ASDA0-Blast protection - Obstacle distance
- TODA1=TODA0-Blast protection - Obstacle distance

With this, if the obstacle distance is negative and |obstacle distance| > Blast protection then the new values will all be larger. Cases to test here are:

- |obstacle distance| > Blast protection
- |obstacle distance| = Blast protection
- |obstacle distance| < Blast protection