

LOW FIDELITY PROTOTYPES

Documentation



VAST



Fraunhofer
AUSTRIA

FREQUENTIS

/fh///
st.pölten

Project title: VAST – Virtual Airspace and Tower

Funding: Austrian Research Promotion Agency – FTI-Initiative Take Off

Duration: 01.09.2016 - 28.02.2019

Project ID: FFG 855215

Consortium: Fraunhofer Austria Research GmbH, Graz (Project lead)
Frequentis AG, Wien
St. Pölten University of Applied Sciences

Table of contents

Table of contents	3
1 Introduction	5
2 Interaction Cheat Sheet	6
2.1 Start Screen	6
2.1.1 Keyboard	6
2.2 vast2D, vast2&3D, vastAbstract	6
2.2.1 Mouse	6
2.2.2 Keyboard	6
2.3 Only vast2&3D	7
2.3.1 3D mouse	7
2.3.2 Keyboard	7
3 Hardware and Software Requirements	8
3.1 Hardware	8
3.1.1 Basic hardware	8
3.1.2 Audio hardware	8
3.2 Software requirements	8
4 Checklist: Setup and running	9
4.1 Setup hardware	9
4.2 Setup software	9
4.3 Common Problems	11
5 Start Screen	12
6 Two-dimensional representation (vast2D)	13
6.1 Visual implementation	13
6.1.1 Airplane label (see Figure 6 left)	14
6.1.2 Visual changes	14
6.2 Implicit and explicit interaction	15
7 Combined 2D and 3D representation (vast2&3D)	16
7.1 Visual implementation	16
7.2 Implicit and explicit interaction	16
8 Abstract representation (vastAbstract)	18

8.1	Visual implementation	18
8.1.1	Visual changes	18
8.1.2	Additional flight information	19
8.2	Implicit and explicit interaction	20
9	Configuration	21
9.1	Changing the simulation mode	21
9.2	Working with Simulation Points	21
9.3	Changing the threshold for level bust & loss of separation	21
9.4	Declare automatic pauses (stop points)	21
10	Flight Data	23
10.1	Description of Data sets	24
10.1.1	Events Data set 1: vast2D	25
10.1.2	Events Data set 2: vast2and3D	25
10.1.3	Events Data set 3: vastAbstract	26

1 Introduction

This document describes the three low fidelity prototypes which were developed within the research project VAST.

The document is organized as follows:

- The basic interactions, which can be performed within the prototypes are described under “Interaction Cheat Sheet” in Section 2.
- Section 3 describes the Hardware and Software requirements for the prototypes.
- Section 4 is a checklist to get the prototypes fully running.
- The three prototypes are described in detail in Section 6 (vast2D), Section 7 (vast2&3D) and Section 8 (vastAbstract) in terms of visual implementation and implicit and explicit interaction.
- Some global parameters for the prototypes can be changed in a configuration file, which is described in detail in Section 9.
- The airplanes are stored in CSV-Files, which are described in Section 10.

2 Interaction Cheat Sheet

2.1 Start Screen

2.1.1 Keyboard

Alt + F4	Close application and go back to operating system.
----------	--

2.2 vast2D, vast2&3D, vastAbstract

2.2.1 Mouse

Scroll wheel	Zoom
Left click + hold + drag	Navigating left, right, up, down.
Left click on airplane label	Select airplane

2.2.2 Keyboard

1	Play
Space/2	Pause
3	Slow playback
4	Fast forward
R	Reset zooms
ESC	Exit prototype

2.3 Only vast2&3D

2.3.1 3D mouse

Push left	Move to the left side
Push right	Move the right side
Push forward	Zoom in
Push backward	Zoom out
Push up	Get higher
Push down	Get lower

2.3.2 Keyboard

Z	When airplane already selected: auto-zoom
Arrow UP	Change camera angle
Arrow DOWN	Change camera angle
W	Zoom in
A	Move to the left side
S	Zoom out
D	Move to the right side
Q	Get higher
Y	Get lower

3 Hardware and Software Requirements

3.1 Hardware

3.1.1 Basic hardware

The prototypes were tested with the following hardware setup.

- Monitor: Size: 28 inch; Resolution: 3840 × 2160 Pixels
- PC: CPU: Intel Core i7-7700K, 4 Cores, 8 Threads, 8MB Cache, 4.20 GHz; RAM: 16GB DDR4 2133MHz Kingston Hyper X Fury; Graphics: Zotac NVIDIA GeForce GTX 1060 Mini 6GB GDDR5 PCIe; Harddrive: 250 GB SSD
- 3D mouse (not required, but handy for better navigation for vast2&3D): 3Dconnexion SpaceNavigator
<https://www.3dconnexion.de/products/spacemouse/spacenavigator.html>
- Keyboard
- Mouse

3.1.2 Audio hardware

- 4-channel (quadrophonic) audio system, alternatively: binaural output via headphones
- recommended: SPL meter for level adjustment

3.2 Software requirements

- Windows 10
- Driver for 3D connexion 3DxWare 10 for Windows (64-bit)
- Max7 (demo version is sufficient): <https://cycling74.com/downloads>
- If possible: TeamViewer (for remote access)

4 Checklist: Setup and running

This checklist is for building and setting everything up.

4.1 Setup hardware

- ☐ Build up and wire the monitor, PC, 3D mouse, keyboard and mouse on a table
- ☐ Build up the speakers: Two speakers should be placed beside the monitor, the other two speakers should be placed behind the person on the left and right side (in parallel to the speakers placed in front). The test subject should sit in the center of the installation (see Figure 1). To ensure the best possible sound quality for the test subject, the test lead should not sit directly in front of one speaker.
- ☐ Wire up speakers and the audio interface
- ☐ Power on everything
- ☐ Plugin the audio interface to the PC

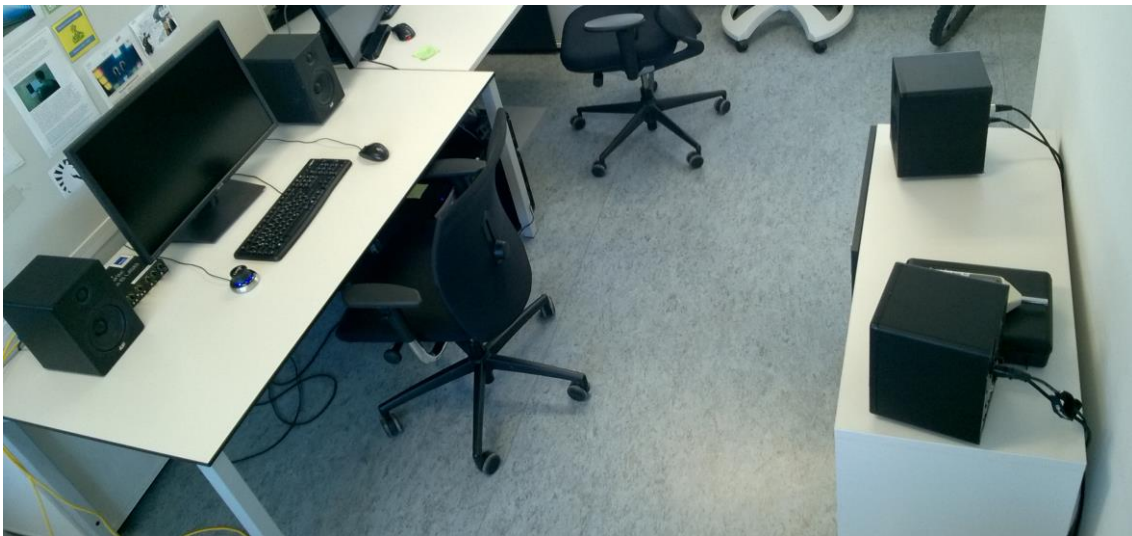


Figure 1: Possible test setup

4.2 Setup software

- ☐ Test audio interface on the PC if it's being recognized

- ☐ Make sure the scaling in Windows 10 is set to 150% (Settings > System > Display)
- ☐ Download the .zip-File and decompress it.

Folder structure:

```
VAST_LoFi_Prototypes
  Data
    config.json
    vast2D
    vast2and3D
    vastAbstract
  Sonification
  VASTLo_Data
  README.md
  VASTLo.exe
  Documentation_LoFi-Prototypes.pdf
```

- ☐ Start **VASTLo.exe**
- ☐ Select the highest screen resolution (4K), disable "Windowed", Select Graphics Quality to fantastic.

Use Sonification

- ☐ Download, compile (if applicable) and install the following libraries (available for Windows 32- /64-bit, OSX) either in the Max7 standard directories or in the ./Sonification/externals subfolder of the downloaded project:
<https://github.com/CICM/HoaLibrary/releases>
<https://github.com/CICM/CicmTools/releases/tag/v2.1>
<https://www.zhdk.ch/downloads-ambisonics-externals-for-maxmsp-5381>
- ☐ Start Sonification.maxproj by double clicking.

- ☐ At Step 1: Select the right audio interface. Note: It may be possible, that the audio interface is not being saved when opening the patch the next time.
- ☐ Check the following settings in *Options > Audio Status*: I/O vector size could be 512, 1024 or 2048 (Maybe the audio driver needs to be configured); Signal vector size: 64.

Step 2: Activate the test sound button and check, if all four speakers (or your headphones) output sound. Calibrate speakers by the delivered noise signal. Every speaker should be at 75dBA slow (use a SPL meter for measuring).

- ☐ Step 3: Nothing needs to be changed, Data will be sent out via OSC on 127.0.0.1:2000 (localhost)
- ☐ Step 4: Automatic setup, nothing to be changed.
- ☐ Optionally: To adjust the volume of the sound, the level can be changed in “main volume” (not recommended) and in more detail when double clicking on “multichannel”.

4.3 Common Problems

- ✓ Check, if only one **VASTLo.exe** and one Max7 instance is running at the same time.
- ✓ Only one monitor should be connected to avoid display problems

5 Start Screen

When starting **VASTLo.exe**, a start screen shows up (see Figure 2). Then you can start the prototypes vast2D (see Section 6), vast2&3D (see Section 7) and vastAbstract (see Section 8). By clicking “Play Audio Samples”, all three audio samples (sound for level bust to high, sound for level bust to low, sound for loss of separation) will be played at once.

With ALT + F4 on the keyboard you can close the start screen.

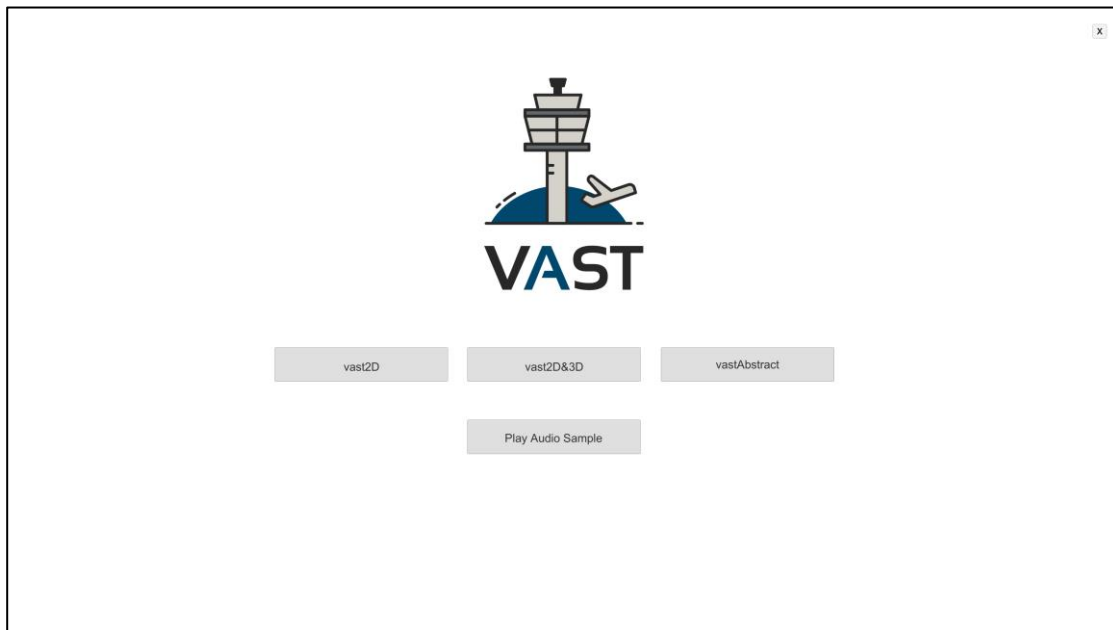


Figure 2: Start screen

6 Two-dimensional representation (vast2D)

6.1 Visual implementation

The screen is split into two parts (see Figure 4). The air traffic is shown in a two-dimensional way on the right side of the screen. The left side shows only a grey background. One point stands for a single aircraft and has a different filling which represent the wake turbulence category (see Figure 3). Routes are represented as a solid white line (see Figure 4). They will appear, when the airplane is visible on the screen. The distance of airplane that has already flown will disappear.

The darker grey represents the sector an ATCO is responsible for. The runway, visualized with a black rectangle, is in the middle of the scene. A mini map shows always the whole scene, even when the ATCOs is zooming in. When zoomed in, the mini map also shows the current part visible on the 2D-representation. The scene also shows the current time (system time).

The geographical north is always on top of the scene and rotating the scene is not possible. For clarification, an icon on the bottom right side is placed.

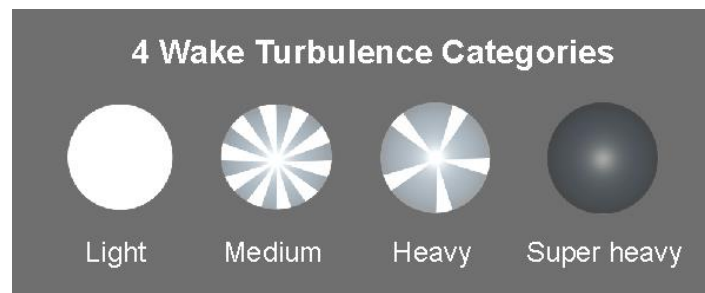


Figure 3: Representation of Wake Turbulence Categories in 3D. From left to right: Light (L), Medium (M), Heavy (H) and Super heavy (J).

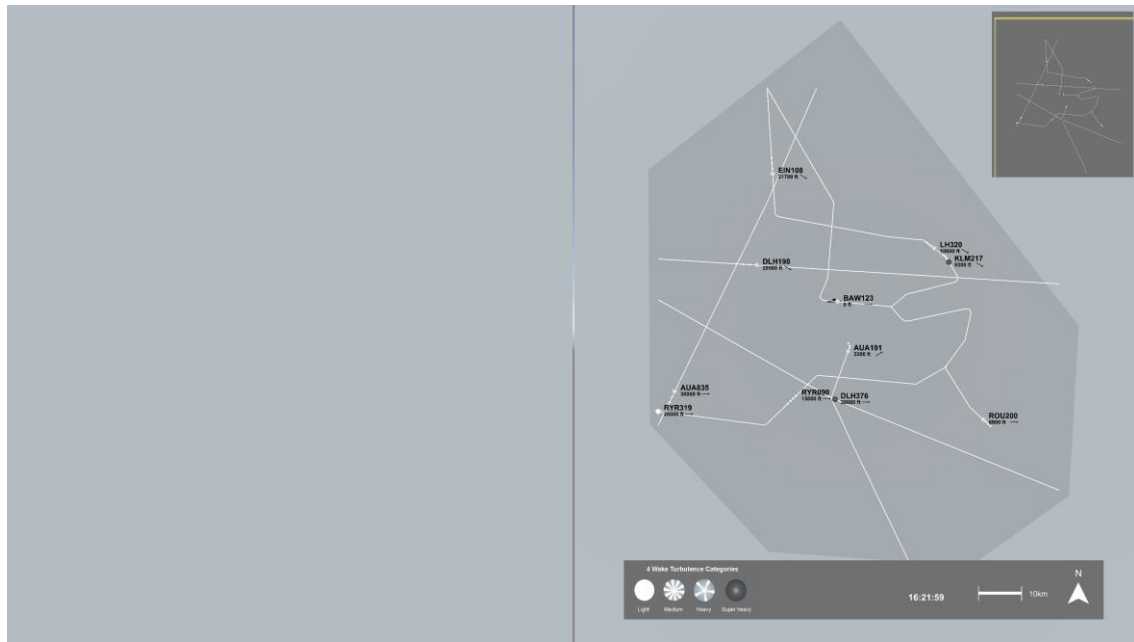


Figure 4: Overview vast2D

6.1.1 Airplane label (see Figure 5 left)

One airplane is represented by a dot. Next to it the airplane label is placed, which shows the following information:

- Call sign/Airplane ID (for example: AUA123)
- The current height (Entity: ft). The current height will be rounded.
- Indication, if aircraft is ascending (arrow up), on the same flight height (horizontal arrow) or descending (arrow down).

6.1.2 Visual changes

Visual changes occur for different events:

- **Loss of separation:** If a loss of separation (two aircraft will fall below the minimum distance) between two airplanes occurs, both airplane labels change their colour to red and will have a white background (see Figure 5 middle). The threshold can be defined in the configuration file (see 9.1).
- **Level bust:** If a level bust occurs (aircrafts falls below or above the cleared height), the airplane label changes its colour to red and will have a white background (see Figure 5 right). The threshold can be defined in the configuration file (see 9.1).

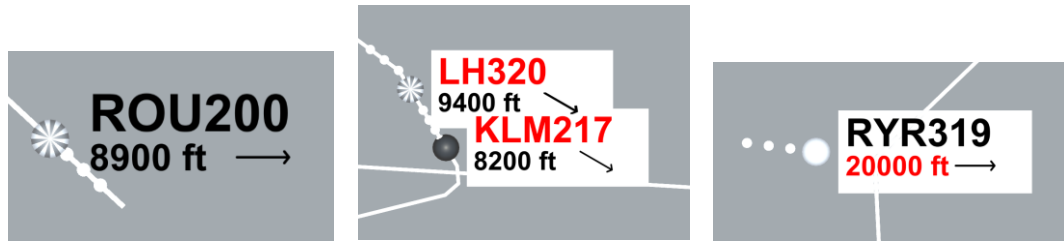


Figure 5: left: Airplane label; middle: Loss of Separation; right: Level bust;

6.2 Implicit and explicit interaction

The following mouse interaction is possible in the vast2D concept:

- **Zooming:** By using the mouse wheel, zooming in and out is possible.
- **Moving:** By clicking and holding the left mouse key, the scene can be moved to the top, bottom, left and right side.
- **Highlighting a flight path:** By clicking on an airplane or on the airplane label, the flight path is coloured blue and the label shows a blue border. When selecting a new airplane or when clicking into an empty space, the old one will automatically be deselected.

The following keyboard interaction is possible in the vast2D concept:

- 1: PLAY
- Space / 2: PAUSE
- 3: Slow playback
- 4: Fast forward
- R: Reset zoom
- ESC: Exit application

The following implicit interaction occurs in vast2D:

- Pauses can also be defined in the configuration file to pause the scene automatically, via a simulation point.
- The flight path as well as the aircraft disappears, when the aircrafts finished landing.

7 Combined 2D and 3D representation (vast2&3D)

7.1 Visual implementation

The 3D view is presented on the left half of the monitor, the 2D view on the right side (Split screen, see Figure 6). The geographical north is always on top and the scene can't be rotated. In the 3D view, the height is multiplied by a factor of five to gain better situational awareness. The wake turbulence categories are also visualized in the 3D scene.

The 3D view has some additional visual elements:

- Airplanes are connected to the ground with a vertical line.
- The current viewers position is visualized with a green position icon in the mini map.
- The 2D view includes a coloured rectangle.

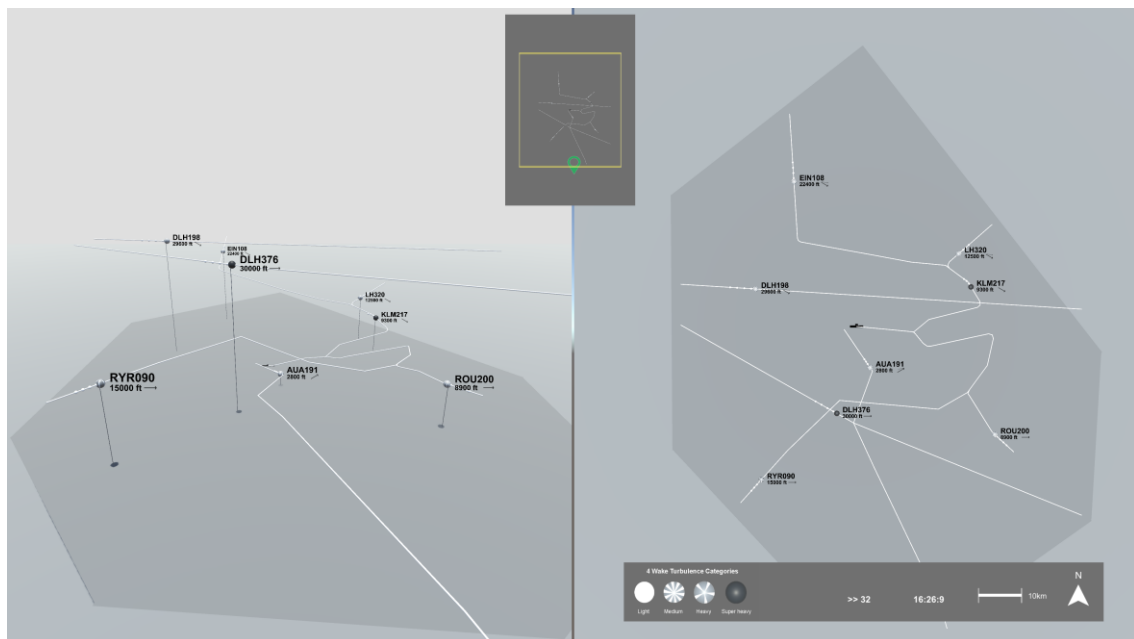


Figure 6: vast2&3D representation

7.2 Implicit and explicit interaction

Important: Any interaction performed in one view has no effect on the other one. In addition to the interaction described in Section 6.2, more interaction is implemented:

- **Auto-zoom:** By clicking on an airplane it will be selected. Using “Z”, it performs an auto zoom on the airplane in the 3D view.
- **Changing the camera angle:** By pressing the arrow key “up” on the keyboard, the camera angle can be changed in the 3D representation. By pressing the arrow key “down”, the previous camera position will be active.
- **By using the 3D mouse**
(<https://www.3dconnexion.de/products/spacemouse/spacenavigator.html>), the interaction in the 3D space is more comfortable than with the regular mouse.

The following interactions are possible:

- Push left: move to the left side (Keyboard alternative: A)
- Push right: move the right side (Keyboard alternative: D)
- Push forward: Zooming in (Keyboard alternative: W)
- Push backward: Zooming out (Keyboard alternative: S)
- Push up: get higher (Keyboard alternative: Q)
- Push down: get lower (Keyboard alternative: Y)

The following implicit interaction occurs when starting the app:

- **Animation:** When starting the app, the 3D representation automatically tilts 90 degrees to clarify, that this is actually a 3D scene.

8 Abstract representation (vastAbstract)

8.1 Visual implementation

The concept vastAbstract combines the above-mentioned Arrival Manager and the conflict detection tool using parallel coordinates (see Figure 7). On right side the normal RADAR is shown, the left side represents the height of the current aircraft (split into departing and arriving aircraft) in the given sector and the estimated landing time of airplanes arriving in the following format: “mm:ss min” (example: “03:24 min”). For the departing aircraft, the time component is not relevant, so a time axis will not be shown.

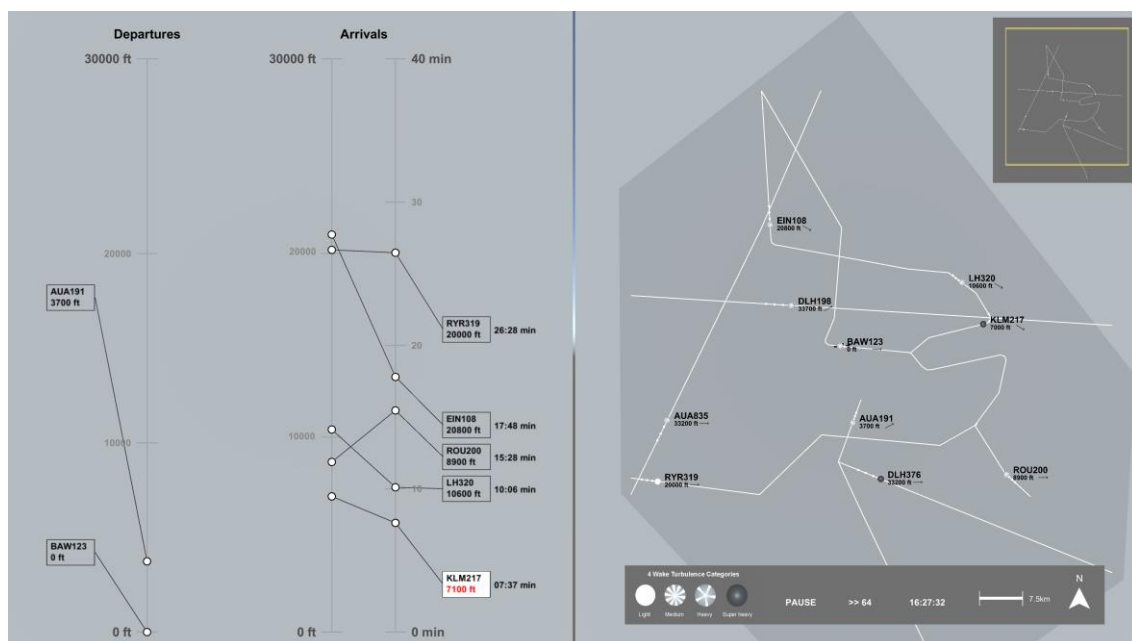


Figure 7: Abstract representation.

8.1.1 Visual changes

Visual changes occur for different events:

- **Loss of separation:** If a loss of separation (two aircraft will soon fall below the minimum distance) between two aircraft occurs, both airplane labels change their colour to red. On the axis on the left screen the airplane label border, the call sign and the dots will change their colour to red and the involved airplanes will be connected through a red line (only for problems within arriving aircraft).
- **Level bust:** If a level bust occurs (aircrafts falls below or above the cleared height), the airplane label change its colour to red. The threshold can be defined in the configuration file.
- **Aircraft selected:** If an aircraft is selected on the RADAR screen, the selection also appears on the axis (in blue).

8.1.2 Additional flight information

Planned & actual position and conflicts (not implemented in the low fidelity prototypes)

A blue filled circle shows the planned corridor, a yellow circle represents the actual position of the aircraft. This information can be placed next to the airplane label (see below). Other states can be seen in right.

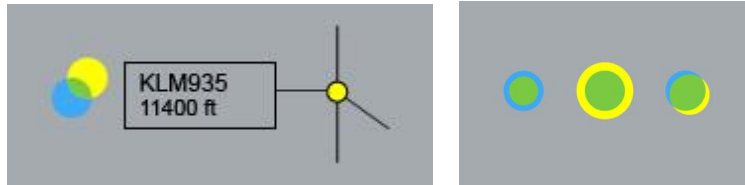


Figure 8 left: Actual and planned position of aircraft; right: from left to right: Airplane is too slow, airplane is too fast, airplane is almost perfect on the planned flight path.

Another visual representation can be seen below.

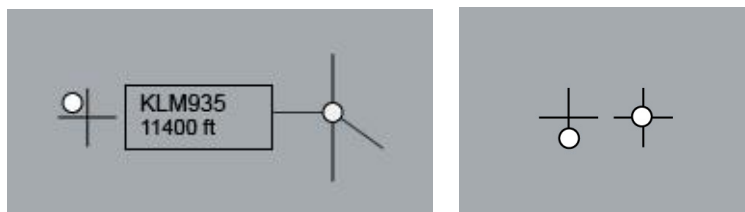


Figure 9 left: Actual and planned position of aircraft; right: from left to right: Airplane is below the planned flight path, airplane is almost perfect on the planned flight path.

An idea for a conflict representation tool can be seen in the next figure. In a matrix, all airplanes will be checked against each other. If a loss of separation occurs, the box will appear in red.

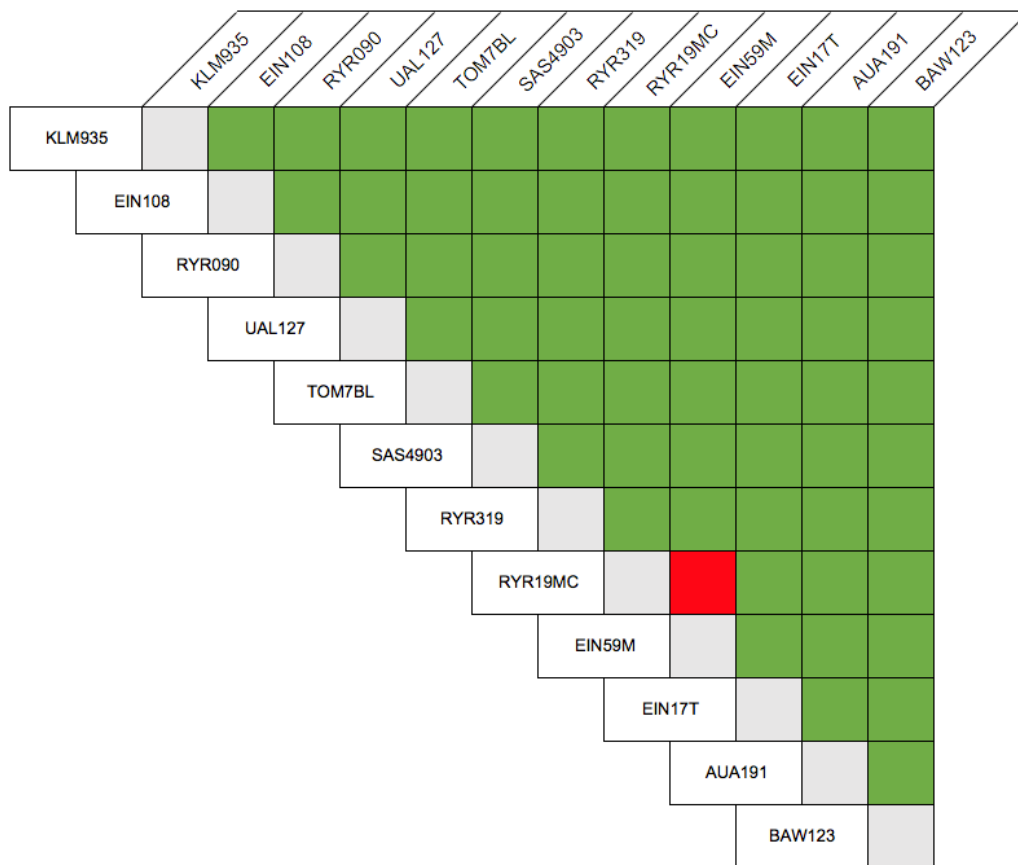


Figure 10: Matrix of conflicts

8.2 Implicit and explicit interaction

The interaction is the same as with the RADAR. Additionally, an airplane can both be selected by clicking the airplane label on the left side of the screen or the airplane on the RADAR.

9 Configuration

The configuration file can be found in the apps folder under **Data > config.json**.

9.1 Changing the simulation mode

gameMode can have three different options:

- **test:** Debug information is not visible in the app.
- **demo:** Breakpoints will be ignored, and all debug information is not visible.
- **debug:** All debug information is visible in the app.

9.2 Working with Simulation Points

In **debug mode**, the current simulation point is visible. A simulation point can be seen as a counter. When starting the app, the current simulation point is 0. Every two seconds it counts higher. When doing a fast forward, the simulation point counts also faster.

9.3 Changing the threshold for level bust & loss of separation

Set **levelBustTolerance** & **lossOfSeperation** to the preferred value (in meters).

- The airplane is flying a given flight path. If it's too high or low, there is a tolerance which prevents the application from warning the ATCO (**levelBustTolerance**). If the airplane falls below or exceeds the tolerance, a visual and auditive warning will occur.
- The airplane has to keep a minimum distance to the other aircraft. This minimum distance is defined in **lossOfSeperation**. This value equals on all directions.

9.4 Declare automatic pauses (stop points)

10 automatic pauses are currently possible and can be defined for the three prototypes separately.

- **vast2D:** In **breakPoints2D** fill the **breakpoint_x** with the preferred value, were pauses should happen.
- **vast2&3D:** In **breakPoints3D** fill the **breakpoint_x** with the preferred value, were pauses should happen.

- vastAbstract: In breakPointsABS fill the **breakpoint_x** with the preferred value, where pauses should happen.

You can see the current values in the app in the debug windows under “Simulation Point”. If you want to set a pause at this exact value, enter this as described below. If a breakpoint isn’t needed, the value must be set to 0.

10 Flight Data

Each aircraft is represented in a CSV file which contains the route and a few more airplane information (see table below). Each prototype has a different data set which are stored in the following folders:

Data

- **vast2D:** Data set for vast2D prototype
- **vast2and3D:** Data set for combined 2D & 3D prototype
- **vastAbstract:** Data set for vastAbstract prototype

The flight paths are based on real flight data (http://iaip.iaa.ie/iaip/aip_eidw_charts.htm, EIDW AD 2.24-17.1). The format of the file looks as follows:

Key	Value	Description
Arrival	Boolean	True = arriving aircraft; false = departing aircraft.
Start	sek	Starting point of aircraft.
cs		Call sign of aircraft.
Entfernung Y	km	Zero point is the airport. Distance airplane to airport on the Y-axis.
Entfernung X	km	Zero point is the airport. Distance airplane to airport on the X-axis.
Höhe	km	The actual height of airplane. 0 = sea level.
speed	km/h	The speed of the airplane at the current point. If the next point has a different speed, it will be interpolated in Unity so that the aircraft accelerates or decelerates.
bottom sep	km	Not used yet.
top sep	km	Not used yet.
left sep	km	Not used yet.
right sep	km	Not used yet.
forward sep	km	Not used yet.
back sep	km	Not used yet.

up	km	The current deviation of the airplane from the planned point (z-axis). Only this value is relevant for the lo-fi test ("Level Bust"). If the airplane is out of tolerance, there is a „Level Bust“.
forward	km	Not used yet.
left	km	Not used yet.
category	L,M,H,J	Defines the wake turbulence category of aircraft: L (Light), M (Medium), H (Heavy), J (Superheavy)

10.1 Description of Data sets

All three data sets contain the same aircrafts, which are the following.

Arrivals

Callsign	Category	File name
DAL110	L	arr_DAL110.CSV
EIN108	M	arr_EIN108.CSV
GW1420	M	arr_GW1420.CSV
JOS248	M	arr_JOS248.CSV
KLM217	J	arr_KLM217.CSV
LH320	M	arr_LH320.CSV
MAS180	H	arr_MAS180.CSV
ROU200	M	arr_ROU200.CSV
RYP090	H	arr_RYP090.CSV
RYP319	L	arr_RYP319.CSV

Overflights/En route

Three airplanes appearing at different heights on the screen. These are only intended to simulate more air traffic and they don't arrive at the airport. **Attention: To avoid errors, aircraft should not fly higher than 30 000ft.**

Callsign	Category	File name
AUA835	M	enrout_AUA835.CSV
DLH198	M	enrout_DLH198.CSV
DLH376	J	enrout_DLH376.CSV

Departures

Callsign	Category	File name
AUA191	M	dep_AUA191.CSV
BAW123	M	dep_BAW123.CSV

10.1.1 Events Data set 1: vast2D

Note: The Events occur with the following settings: LevelBustTolerance: 500m, LossOfSeparation: 2000m.

SimPoint*	Event
185	KLM217-LH320 loss of separation
252	RYP319 level bust at waypoint 2, up = -0,6km
485	EIN108-ROU200 loss of separation

*Simulation Point (see 9.2)

10.1.2 Events Data set 2: vast2and3D

SimPoint	Event
161	ROU200 level bust at waypoint 4, up = -0,6km
372	GW1420-DAL110 loss of separation
547	JOS248-EIN108 loss of separation

10.1.3 Events Data set 3: vastAbstract

SimPoint	Event
155	KLM217 level bust at waypoint 7, up = -0,6km
290	MAS180-JOS248 loss of separation
339	RYP319-RYP090 loss of separation