TWRsim+ - Aerodrome control tower simulator

by Petr Dvorak

Manual

Version 1.487 (build 13.4.2020) Corona edition

www.papadelta.cz

Content

Description

TWRsim+ is a simple yet robust ATC simulator suitable for all types of training.

Features

- universal simulation engine suitable for all types of air traffic control (ACC, APP, TWR)
- editable aircraft performances
- implementation of Magnetic Declination (variation)
- wind influence and wind shear areas
- easily configurable and editable scenarios (areas, fixes, routes, ...)
- real data from AIP, easy updates
- simulation time and weather information window
- broad scale of setting and modification possibilities
- standalone or networked use
- interface for third party data feeds

- unified GUI for all roles (PP and ATCO)
- pseudopilot control via command line and mouse inputs
- intuitive and simple commands
- command parameters allow precise control
- sequence of commands with events
- editable command macros
- simple route planning with trajectory prediction
- enhanced Aircraft info window and EFIS window for aircraft details and control

- main radar window and independent second radar window with individual settings
- configurable radar revolution (refresh rate) for each radar screen
- measuring tools, range rings, RWY center line

- integrated Voice Communication System with call/bell feature

- 3D graphical visualisation
- realistic Sun, Moon and stars movement considering simulator time and date
- realistic day and night sky lighting
- various types of clouds
- fog and visibility setting
- possibility to extend visualisation to three screens/windows
- models of many aircraft and vehicles
- landscape / real elevation model
- binoculars
- target tracking (ground and air)
- possibility to show target callsign and trajectory in visualisation

- Precision Approach Radar window
- possibility to set PAR location at the airport
- configurable help lines and parameters
- switch between units (NM,ft and km,m)
- synthetic target information (switchable)
- blip size according to aircraft type and distance
- clutter areas
- possibility of degradation of Elevation and/or Azimuth

- simple use and installation
- no special HW required
- ideal for multi screen workstations
- continuous development and upgrades

User interface

The user interface consists of several parts:

- Main window which includes:
  - Radar screen
  - Aircraft information window (GUI control)
  - EFIS (electronic flight information system)
  - Weather information window
  - Target list
  - Point/Area/Route list
  - Command line and command history
  - VCS window
  - Debug window
  - Menu and Toolbar
- 3D Tower view
- Map overview (second radar window)
- Precision Approach Radar (PAR) view and PAR configuration window
- Help and configuration window

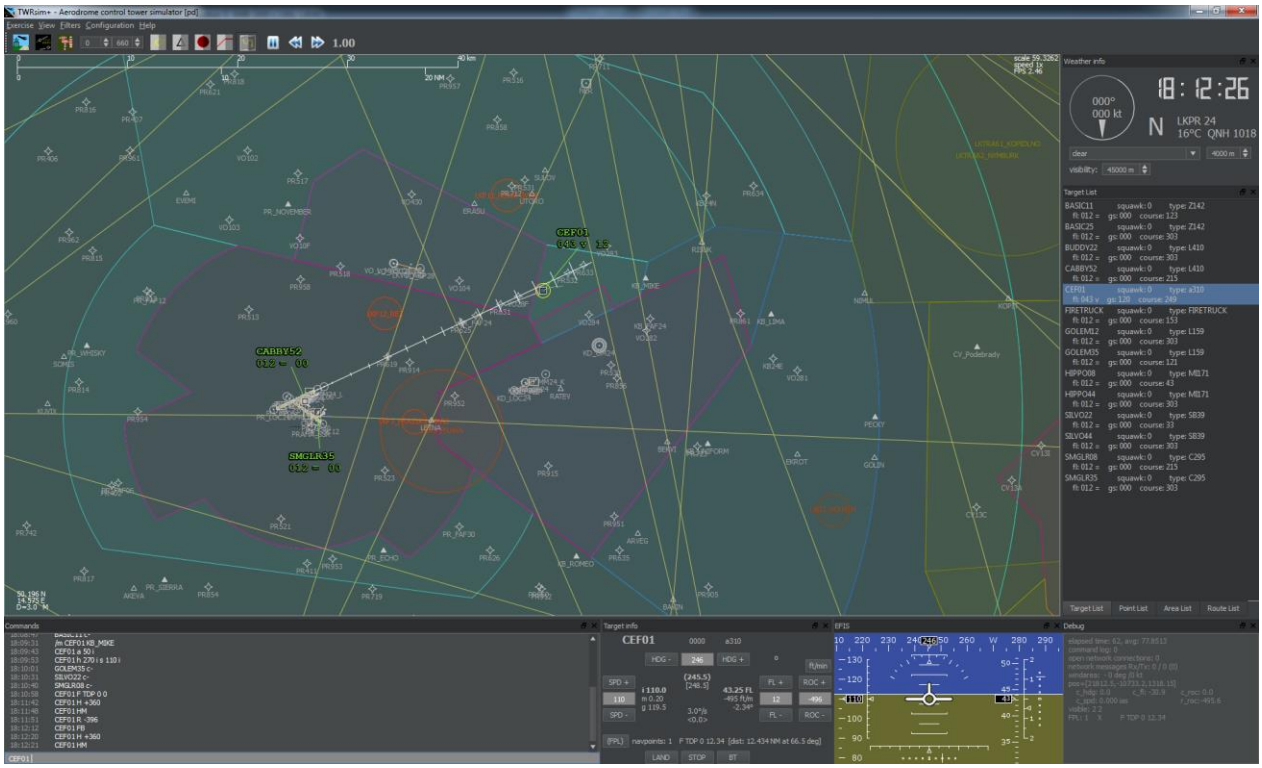

Figure 1: Main window

The radar screen shows the following:

- targets (aircraft, ground vehicles, …)
- points (waypoints, navaids, …)
- areas (sectors, CTR, TMA, CTA, FIR, …)
- routes (runways, taxiways, RNAV routes, air routes, SIDs, STARs, …)
- measure tools (measure vector, runway center line, range rings, …)

The radar target contains target actual position, speedvector, history points and label. The actual selected target is indicated by a ring around it.

The target label consists of the callsign, actual flight level, indication of vertical movement and the ground speed (x10 kts).

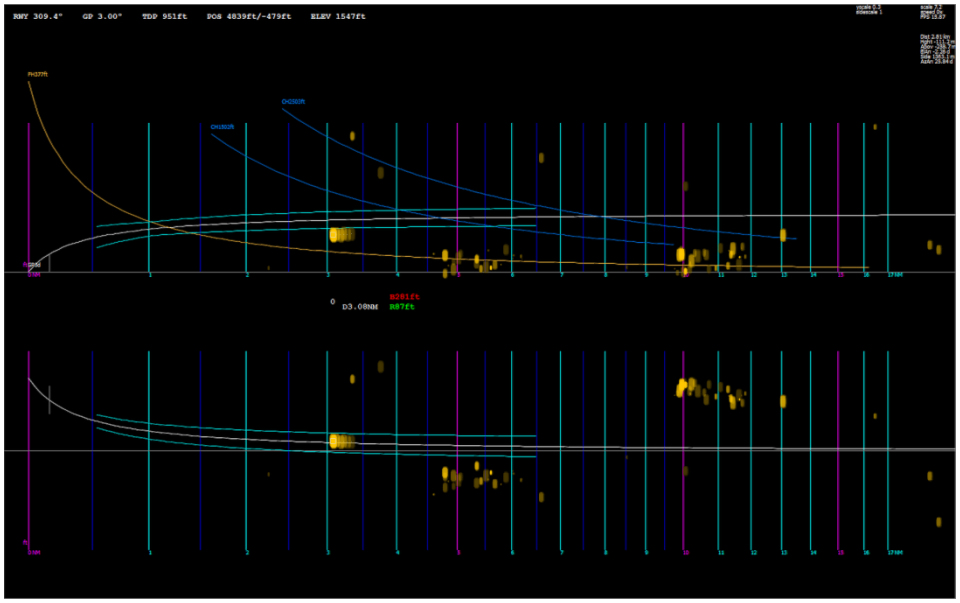

Figure 2: PAR window

Figure 3: PAR config.

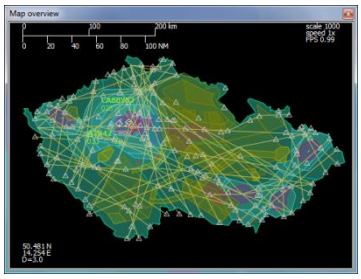

Figure 4: Map overview

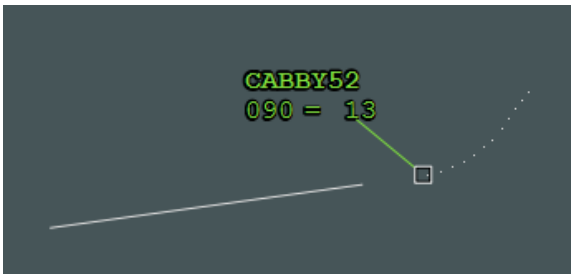

Figure 5: Radar target

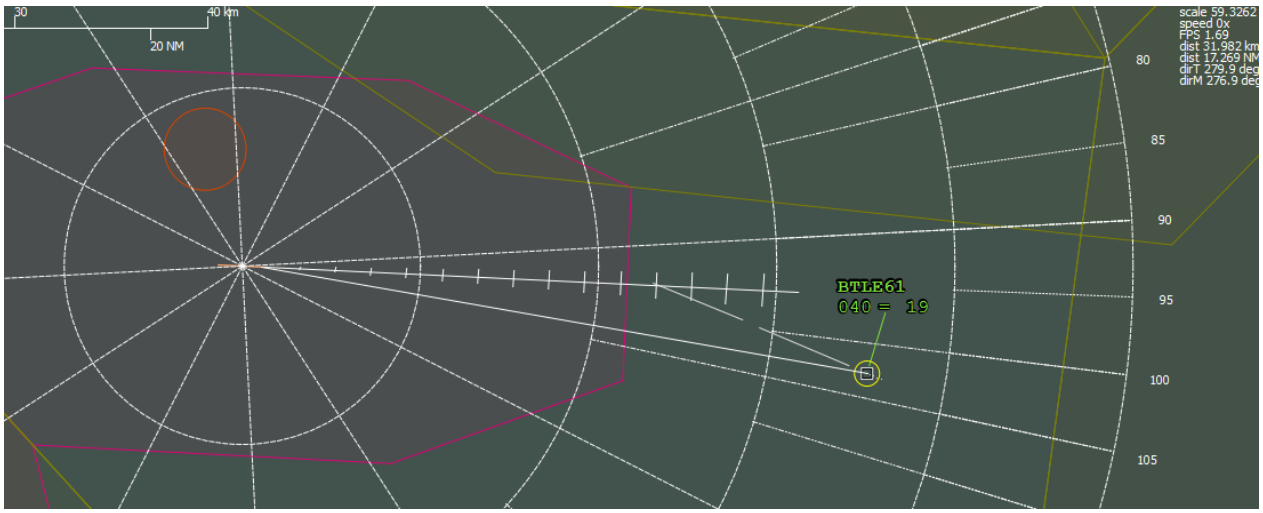

Figure 6: Measure tools (runway axis, range rings, speed vectors and measure vector)

Figure 7: 3D Tower view with realistic topography model and sky

Figure 8: 3D Tower view (night time)

Figure 9: Aircraft tracking (target view)

Figure 10: Aircraft tracking (binocular view)

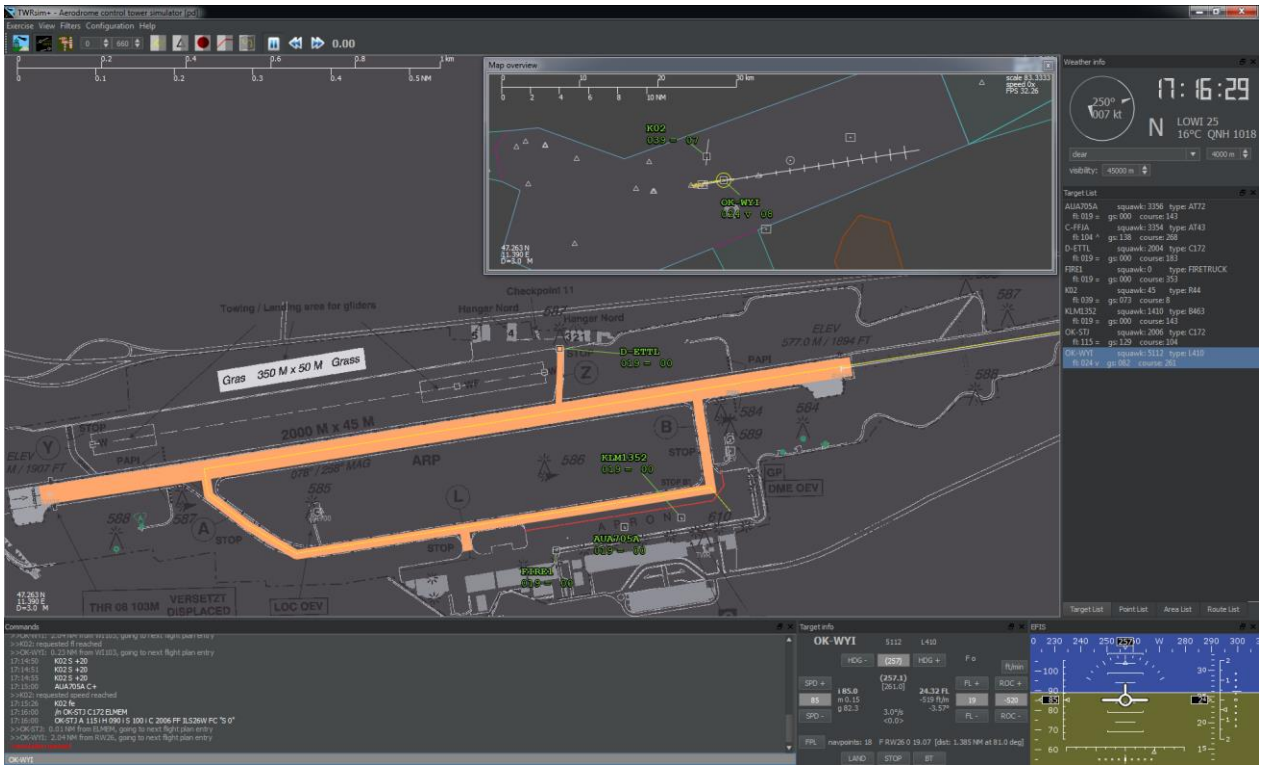

Figure 11: Single screen working position example

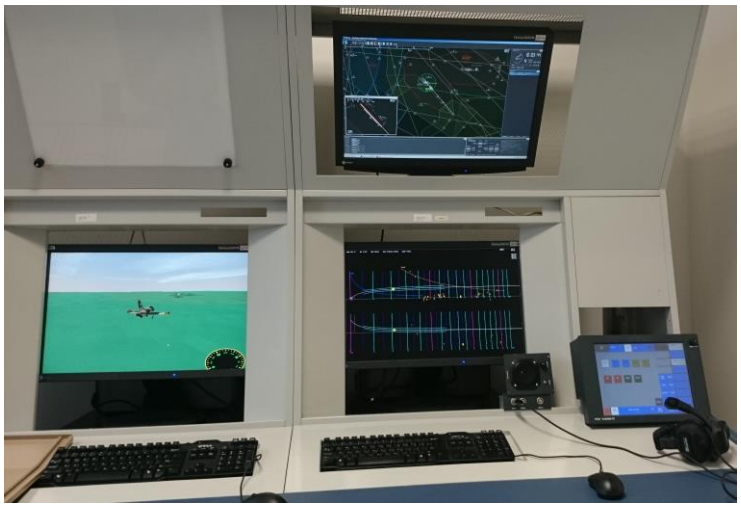

Figure 12: Working position example

Figure 13: Working position example

Minimal hardware configuration

x86 processor 1.8 GHz
1 GB of RAM
graphics card with 256 MB of RAM and supporting OpenGL
operating system Windows XP
screen resolution 1280x1024
extended desktop (recommended)
200 MB of HDD space

Please note that application performance depends on the data used. Large topography or aircraft models, textures, number of visible areas and other elements can cause some performance issues. However, the system can be scaled and adapted to the individual requirements.

Help

Menu

Exercise

Open exercise – opens saved command log (exercise), all loaded commands are queued and performed at specified time.
Save exercise – saves command log or full log into a file (command log can be used to replay the exercise next time); also allows saving of configuration (visible points/areas/routes, position and setting of different windows).
Quit – exits application.

View

Switches selected windows on/off.
Main radar window is always visible.
There are three independent windows:

- 3D TWR view,
- Map overview,
- PAR window.

And ten dock windows:

- Commands (bottom left),
- Target List, Point List, Area List, Route List (tabbed on right),
- Weather info (upper right),
- Target info (bottom right),
- EFIS (bottom right),
- VCS – Voice Communication System (bottom right)
- Debug information (bottom right).

Also allows switching the main window to full screen.
Position of the windows can be saved using Save exercise – Configuration.

Filters

Disables target labels, points, areas, routes or trajectory.

Configuration

Load configuration – loads configuration from a selected file; it is recommended to use this for previously saved configuration only, but can be used globally.
Edit configuration – displays current configuration and allows editing; after closing the window and confirmation, changes are performed (warning: it will not delete current lists).
PAR configuration – displays PAR configuration window with real-time editing capability.
Reset – clears target list and resets some configuration to default (does not disconnect from network).
Network information – shows information about network setting.
Connect to host – connects via network to the host (IP address defined in config).

Help

Shows this help and About window

Tool bar

3D TWR view

Displays 3D TWR view window with visualisation.

Map overview

Displays additional 2D radar window (filters out ground targets).

PAR window

Displays Precision Approach Radar (PAR) window.

Spin boxes

Sets lower and upper vertical limits (in flight level) for filtering targets, areas and routes.

Filters

Disables target labels, points, areas, routes or trajectory, enables aircraft silhouette.
Aircraft silhouette is taken from the _shadow.png file and show actual direction of the aircraft.

Simulation speed buttons

Pauses/resumes simulation or sets double or half simulation speed.

Simulation speed

Indication of current simulation speed.

Figure 14: Tool bar

Network connection status

Right corner of the Tool bar indicates network connection status:
Master / Slave, number of connected slaves (in case of Master) and connection icon.
Doubleclick on connection icon connects to host or disconnects from host and removes all slave connections.

Figure 15: Indication of disconnected Master, Connected Slave and Master with 2 connected slaves

Username is shown besides the network connection status as well.

Control

3D Tower View

The 3D Tower View allows visualisation of the current situation including operational, meteorological and time of day conditions in 3D.
It allows also showing some additional synthetic information, e.g. target name, to support future technology developments (remote tower).
It uses unique and precise sky-dome illumination including Sun, Moon and stars positions and different cloud types.

F1	switch labels and trajectory off/on
Alt+F1	switch constellation lines and labels off/on
Ctrl+F1	shows target history as transparent wall
F2	switch rwy/twy lights off/on
F3	switch spotlight off/on
F4	disable/enable fog (fog density based on visibility value)
F5	switch wireframe model on/off
F6	switch textures off/on
F7	switch antialiasing off/on
F8	disable/enable Moon and stars position calculation and visualisation
F9	switch shadow mode (sun/under/both/off)
F10	hides/shows the disk representing airport ground
+	zoom in
-	zoom out
*	disable zoom
F	switch to fullscreen
LMB or Left/Right arrow	rotate view (change view direction)
RMB or Up/Down arrow	horizontal move
Alt+LMB	change tilt
Alt+RMB or Alt+Up/Down arrow	vertical move
Shift+Arrow keys	faster move
wheel	zoom
Home	set position to TWR
End	set position above TWR (for better view)
X	enable/disable extended view on 3 windows
LMB doubleclick	lock view to the nearest target (enable tracking mode), rotate view or move to disable tracking
T	change target tracking mode (fixed position or relative to the target)

Radar

Radar screens (both "radar" and "map") show so-called Plan Position Indicator – the classic target positions in the map.
It uses Lambert conformal conic projection.
Screen update rate (sweep) is adjustable as well as label (font) size.
The "map" window by default filters out targets which are on the ground.

+	zoom in
-	zoom out
wheel	zoom
RMB	move view (drag and drop), copy earth coordinates of mouse position to clipboard (suitable for navigation)
LMB	select target or move target label (via callsign), doubleclick on callsign selects the target; selected target's callsign will be entered into the command line
Shift+LMB	add the selected fix to the flight plan of selected target (fills in the command line)
Ctrl+LMB	measure distance and angle from mouse position
F	switch to fullscreen
F1	switch range rings on/off distance between range rings is 5 NM
Alt+F1	switch range rings orientation to magnetic or true north
Ctrl+F1	switch range rings between basic (only rings) and full (rings+angles)
F2	switch runway axis indication on/off uses runway parameters (threshold position and direction) from PAR distance between distance marks is 1 NM
F3	switch between label with transparent background or background outline
F4	switch area borders off/on
F7	switch antialiasing off/on

Clicking into the radar/map gives the focus to the command line in order to allow immediate input.
Due to this fact, it is necessary to hold mouse button in radar/map while performing the above mentioned keyboard commands.

Precision Approach Radar (PAR)

Shows Precision Approach Radar screen with Elevation (top) and Azimuth (bottom) position view. Targets are visible only within the coverage area (-1 to 9° in El, ± 15° in Az).
Synthetic information about a target is shown in between (could be switched off by setting squawk to 8 or by command CP-).
Distance scale is logarithmic, vertical scales changes with distance (the closer to the runway threshold, the more details are shown). Blip size may change with distance.
Center line in Azimuth view is curved because of the location of the PAR respective to the runway threshold/touch down point.
Configuration of different help lines and PAR setting can be done in separate PAR configuration window.
Elevation and/or Azimuth element can be degraded for particular target.
Image can be influenced by clutter areas.

+,-	change range
wheel	change range
Shift+wheel	change scale in Elevation
Ctrl+wheel	change scale in Azimuth
LMB	select target (on synthetic info) or move target label
Shift+LMB	measure height, distance and angle in Elevation (draws a direct line from one point to another)
Ctrl+LMB	measure side deviation, distance and angle in Azimuth (draws a direct line from one point to another)
F	switch to fullscreen
F5	switch units of measurement (NM,ft / km,m)
F6	mirror the view (runway threshold on left/right)
F7	switch antialiasing off/on

Aircraft information window

Shows all relevant information about a selected target.
Also allows easy control of the target and changing of requested values (heading, FL, ROC, speed).
Entered values or button clicks are transformed into corresponding commands.
ROC can be entered either in ft/min or in degrees of glide path angle.

LMB on a button	increses (+) or decreses (-) the relevant requested value
Shift+LMB on a button	increses (+) or decreses (-) the relevant requested value more faster
Ctrl+LMB on a button	increses (+) or decreses (-) the relevant requested value more slower

If the flight plan (FPL) is in brackets or marked with X, the flight plan is not followed.

EFIS window

Shows all relevant information about a selected target in well organised and broadly recognized way.

EFIS window is put together from different modules (files efis0.png to efis7.png in img directory). It can be adjusted according to user's preferences (pixel steps for scales needs to be the same).

EFIS window is directly correlated with Aircraft information window, but does not allow any inputs. Use Aircraft information window and EFIS window together.

For more details about the use of EFIS window see Commands section.

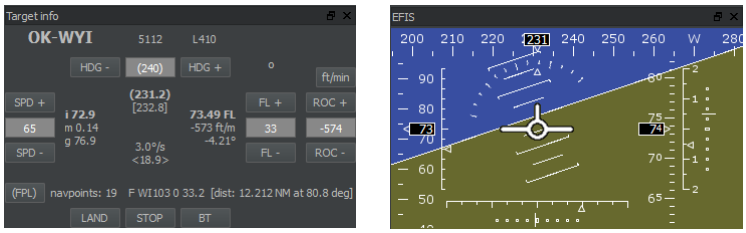

Figure 16: Aircraft information window and EFIS window

VCS window

Shows integrated Voice Communication System window.

Integrated VCS allows voice communication between networked positions (VoIP).

Pressing the PTT button transmits audio signal form selected input to the network.

Outgoing and incoming audio is indicated on the PTT button.
Input audio level (outgoing audio) is indicated in the bar during transmittion (bottom to top).
Output audio level (incoming audio) is indicated in the bar during reception (top to bottom).
The source of incoming audio is indicated in the line under the PTT button.

UDP checkbox allows switching between TCP audio connection and UDP audio connection.

Input/Output field allows selection of devices used for audio input and output.

Bell button sends /BELL message and plays bell.wav sound. This can be used as a ring tone.

Speaker / Mute button mutes all incoming audio.

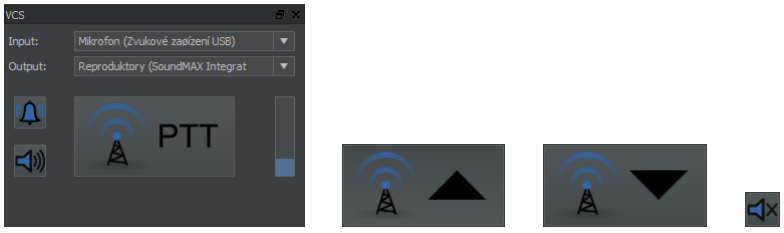

Figure 17: VCS window, indication of outgoing (Tx) and incoming (Rx) transmission and mute button

F12	transmits audio from input device to network – hold to transmit (can be used without VCS dock being active)

Debug window

Shows debug and status information (elapsed recount time, number of commands in the stack, network messages, audio bytes, ...).

Weather information window

Shows simulation virtual time and basic weather information (wind speed and direction, ATIS information, selected airport, etc.).
ToolTip information shows some more details about wind (variable wind, area with different wind).
Also allows configuration of clouds (type, base) and visibility (fog).

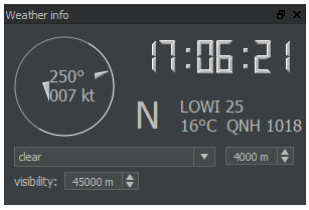

Figure 18: Weather information window

Target list

Shows the list of all targets in the dock window.

LMB	show full target info, select aircraft drag and drop to change order

Target list allows manual sorting of the aircraft using Drag&Drop method.

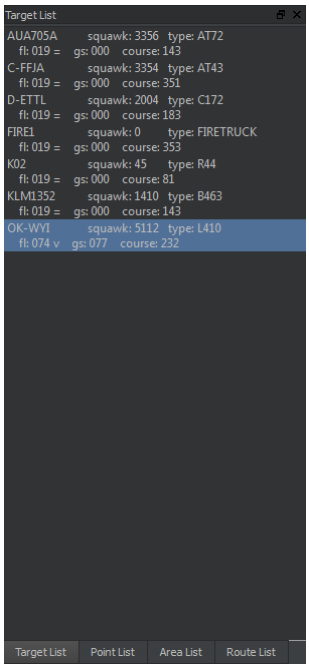

Figure 19: Target list

Point/Area/Route list

Shows the list of all points/areas/routes in separate dock windows.
ToolTip information shows some more details about the point/area/route.
Multiple selection is possible.

RMB	view or disable selected points/areas/routes in radar view or map overview (context menu) selecting visible items switches labels on/off

The order of the items in these lists is according to the order in which they were loaded.
These lists allow hot searching between the items.

Three calculated points are always added (but not shown in the list): TWR, TDP and PAR.

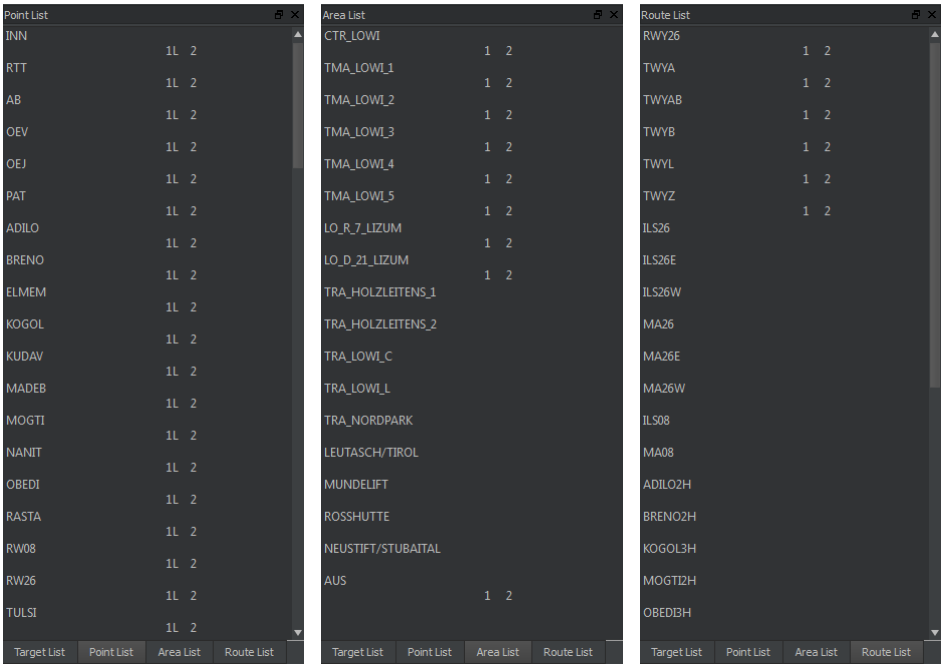

Figure 20: Point list, Area list and Route list

Command line

All commands for aircrafts/vehicles must be preceded by its name *I* and separated by space.
It is possible to enter multiple commands in a row, name *I* is then entered only once.
Other commands (general) not related to a target use prefix /.
Every entered command is copied to clipboard, so it can be easily used again.

Enter	confirm command and copy it to clipboard selected target's callsign will stay in the command line
Esc	delete the command line
Ctrl+V	paste command or map coordinates from clipboard
I H *hdg* [L|R|I]	turn [left|right] heading *hdg* degrees [immediate]
I H +|-*deg* [I]	turn right|left by *deg* degrees [immediate]
I HR [L|R|I]	turn [left|right] to the runway direction [immediate]
I HRO [L|R|I]	turn [left|right] to the opposite runway direction [immediate]
I O *num* [L|R]	orbit [left|right] *num* times
I HM [I]	maintain current heading [immediate]
I TR *num*	set turn rate to *num* degrees per second
I TRN	set normal turn rate
I S *kt* [I]	maintain speed *kt* knots IAS [immediate]
I S +|-*kt* [I]	change speed by *kt* knots IAS [immediate]
I S *.m* [I]	maintain Mach number *m* [immediate]
I S +|-*.m* [I]	change speed by Mach number *m* [immediate]
I SN [I]	maintain normal speed (IAS) [immediate]
I SN. [I]	maintain normal Mach number [immediate]
I SNM [I]	maintain minimal speed (IAS) [immediate]
I SNT [I]	maintain taxi speed (IAS) [immediate]
I SM	maintain current speed (IAS)
I SM.	Maintain current Mach number
I A *fl* [I]	climb/descent to flight level *fl* [immediate]
I A +|-*ft* [I]	climb/descent by *ft* feets [immediate]
I AM [I]	maintain current altitude (stop climbing/descending) [immediate]
I R *roc* [I]	set rate of climb/descent to *roc* ft/min [immediate]
I RNM [I]	set rate of climb/descent to maximum value [immediate]
	if rate of climb/descent is not explicitly set, normal value is used
I F *fix* [dist] [fl]	flight direct to *fix* [to distance *dist* from *fix*] and reach flight level *fl* at this position
I FD *fix* [*dist*] [*fl*]	flight to distance *dist* from *fix* and reach flight level *fl* at this position
I FR *fix angle*	flight to radial *angle* degrees from *fix*
I FT *sec*	wait *sec* seconds, then go to next flight plan entry
I FH	wait until requested heading is reached, then go to next flight plan entry
I FS	wait until requested speed is reached, then go to next flight plan entry
I FA	wait until requested altitude is reached, then go to next flight plan entry

I FC "*cmd*"	perform command cmd, then go to next flight plan entry
I FF *route* [I]	add [inverse] *route* into flight plan
I FE [*n*]	erase [*n* first entries of] flight plan (if *n*<0, then erase from back)
I FP	interrupt navigating via flight plan
I FB	resume navigating via flight plan
I F0	land at TDP, then go to next flight plan entry (F TDP 0 0)
I F1	set speed to 0 (stop), then go to next flight plan entry (FC "S 0")
I FI	report current flight plan
I EEE	erase target (alternative to /e)
I G+	gear down
I G-	gear up
I Q *sec*	squawk ident for *sec* seconds
I C *code*	set squawk to *code*
I C+	show hidden track (activate SSR transponder)
I C-	hide track (target and its label) – switch off SSR transponder
I CP+	show synthetic label in PAR window
I CP-	hide synthetic label in PAR window
I CE+	show track in Elevation part of PAR window
I CE-	hide track in Elevation part of PAR window
I CA+	show track in Azimuth part of PAR window
I CA-	hide track in Azimuth part of PAR window
I LV	lock view to current target
I SCALE num	set scale multiplier for target model
/W *dir spd*	set wind direction to *dir* degrees and wind speed to *spd* knots
/VW *dir spd*	set maximums of variable wind: direction change up to *dir* degrees and speed change up to *spd* knots
/WA *area dir spd*	set different wind in a defined *area* to *dir* degrees and *spd* knots
/SV *num*	set speed vectors to *num* minutes
/HP *num*	set history points to *num*
/RR *fix*	set center point for range rings to *fix* (fix or lat/lon coordinates can be used)
/P *num*	set simulation speed to *num*
/T *hh*:*mm*:*ss*	set simulator time to *hh* hours, *mm* minutes and *ss* seconds
/D *dd-mm-yyyy*	set simulator date to *dd* day, *mm* month and *yyyy* year
/E *i*	erase target *i*
/N *I type fix*	insert new target with name *I*, type *type* to position *fix* (existing fix or lat/lon coordinates can be used)
/M *I fix*	move target *I* to position *fix* (existing fix or lat/lon coordinates can be used)
/AREA *name id num*	set visibility of area *name* in widget *id* (0=radar, 1=map) to *num* (0=none, 1=without label, 2=with label)
/SAY "*text*"	void command, only displays *text*
/HOSTIP *address*	set host IP address to connect to

/USER “*name*”	set username (will be visible in tool bar)
/BELL	plays bell.wav ring tone

Command macros:

It is possible to define user command macros in the MACRO_FILE file.
Macro commands use prefix $ and can contain a sequence of any of standard commands.

For more details about definition of command macros see Macros definition section.

First five macros are mapped to the buttons in the bottom part of Aircraft information window.

GUI for commands in Aircraft information window:

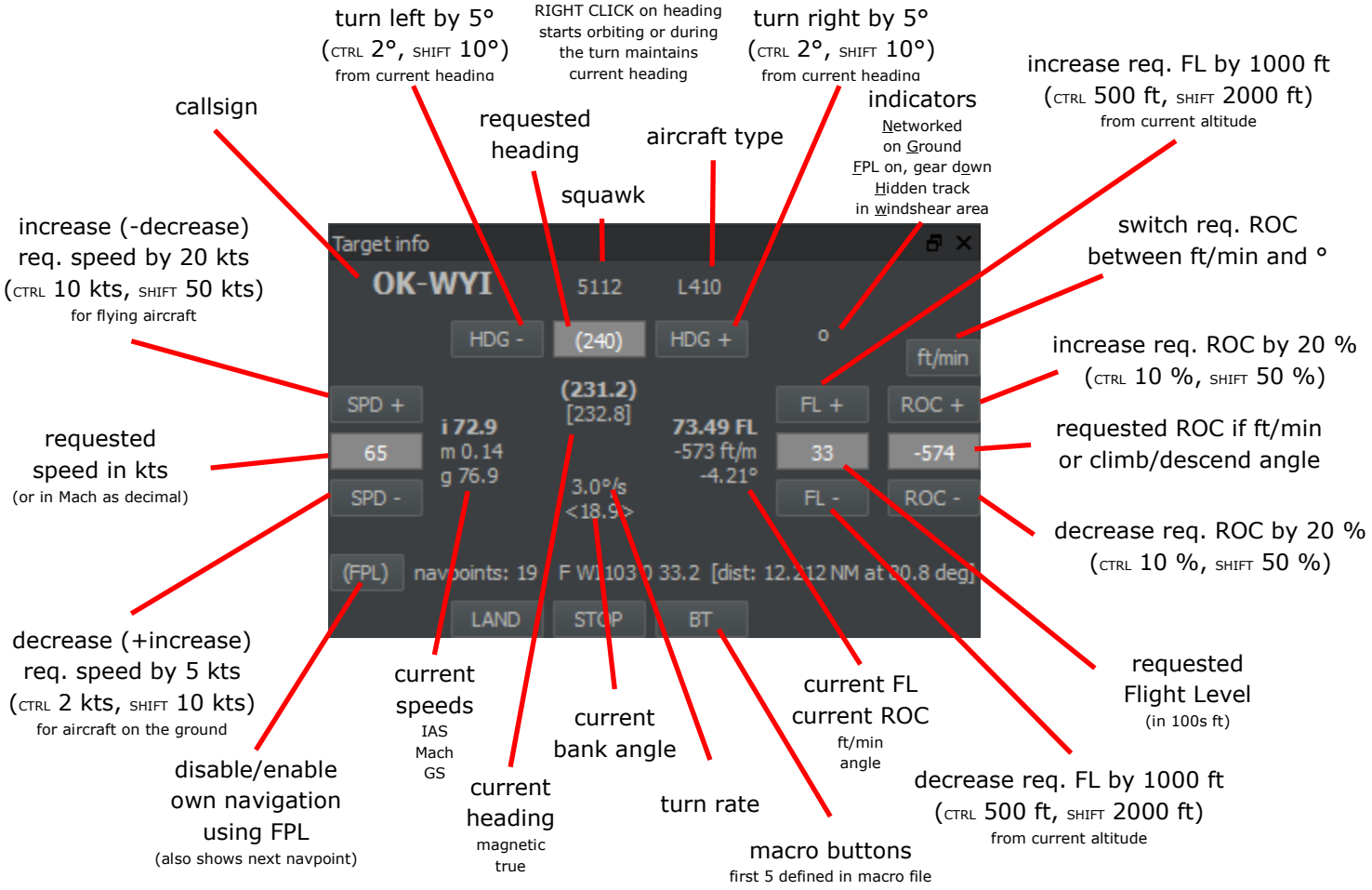

Figure 21: Aircraft information window description

EFIS window description:

EFIS (Electronic Flight Information System) shows a simple and well readable summary of current aircraft status.

It is possible to easily correlate values between Target info and EFIS.

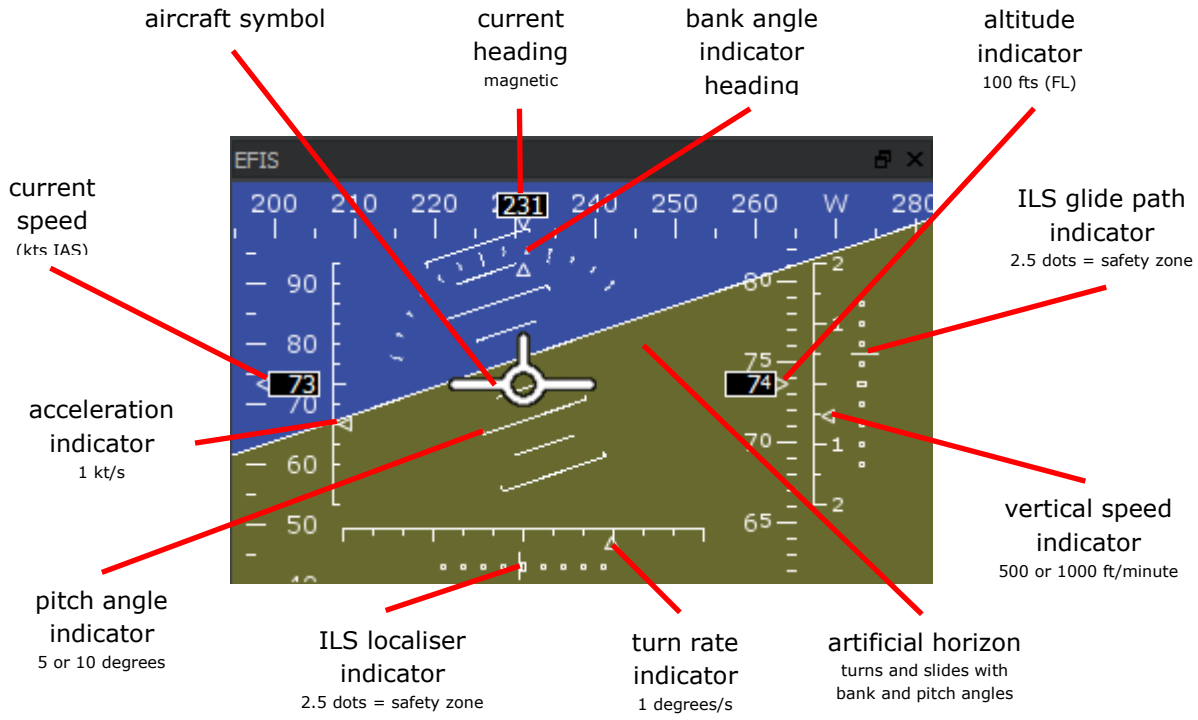

Figure 22: EFIS window description

Tips and Tricks

Networked use

TWRsim+ can be used in networked configuration, where one or more slaves are connected to one master (server) via TCP. In this case, the server holds the simulation and sends regular updates of target positions (internal OBELIX messages) and control commands entered by user (internal IDEFIX messages). Between these updates, each slave takes care about recomputing the new position/status, but this is overwritten when the new update message arrives.
TWRsim+ also automatically listens to UDP messages in ROSE format (ATCSim Fischer, ROSE 2.10.429) and adds these targets into the simulation.
Networked targets are automatically deleted if there is no update in the last 5 seconds.

VCS uses a separate TCP connection. It is possible to switch between TCP and UDP audio stream. Audio from clients is sent to host and retransmitted to all other clients.

Aircraft movement

The targets move according to the current performance values. Each aircraft type has a set of standard and limiting values.
The aircraft can only climb if its speed exceeds the minimal take off speed (IAS_min), otherwise stays in the current altitude.
Aircraft heading is affected by wind, however it is possible to use flight plan navigation commands to navigate direct to a given fix. This approach will compensate the wind influence and use the track instead. It is also possible to order a command to reach a given altitude at given fix. In this case, the actual rate of climb/descend is calculated automatically to reach that. If the aircraft is following the flight plan and uses its own navigation and a superior command is used (e.g. vectoring, rate of climb/descend change when reaching a fix at a given altitude), the flight plan is suspended and can be returned manually.
The flight plan trajectory (based on navigation waypoints) of the selected aircraft is displayed on the radar screen.
Right click on turn left/right button starts orbiting. If the aircraft is in turn, next right click stops the turn and maintains current heading. Ongoing turn is indicated with the brackets in the requested heading field.
There can be a different wind (e.g. windshear) in a defined area. Once the aircraft is in this area, it will be influenced by these defined values, not by the default wind. The actual different wind is displayed in the ToolTip of Weather information window.
Use command macros for quick use of predefined more complex commands.

Other

Some information messages are shown as modal dialog windows. These might be configured with timeout for automatic close. Press any key or click into the window to stop the timer.

Configuration

TWRsim+ allows broad customization and modification by changing the data set and setting the configuration variables.
Required configuration files are the following:
- ./config.txt
- model/models.txt
- weather/clouds.txt
- topo/topo.txt

Configuration variables

Most configuration variables can be set in file config.txt.
The structure and meaning the variables is:

REF_POINT *lat lon*	reference point for map projection, latitude [degrees minutes seconds N|S] and longitude [degrees minutes seconds E|W]
MIN_LAT *lat*	latitude of the first standard parallel for map projection [degrees minutes seconds N|S]
MAX_LAT *lat*	latitude of the second standard parallel for map projection [degrees minutes seconds N|S]
EARTH_RADIUS *num*	earth radius on Equator (for map projection) [m]
SIMSPEED *num*	simulation speed at start time
TICK_INTERVAL *num*	minimal interval for recomputing aircraft position [ms]
REPAINT_INT_0 *num*	interval for the update of main window docks [ms]
REPAINT_INT_1 *num*	interval for radar update [ms]
REPAINT_INT_2 *num*	interval for map update [ms]
REPAINT_INT_3 *num*	interval for PAR update [ms]
BLICK_INTERVAL *num*	interval for aircraft identification (flashing) [ms]
NET_INTERVAL *num*	interval for sending aircraft position updates via network [ms]
WIND_CHANGE_INT *num*	interval for performing wind change [ms]
GND_LEVEL *num*	elevation of the reference point [FL, hundreds of ft]
TWR_ROT *num*	rotation of the TWR building, initial view rotation [degrees from the north]
TWR_HEIGHT *num*	height of the TWR building [m]
TWR_POS *lat lon*	position of TWR, latitude [degrees minutes seconds N|S] and longitude [degrees minutes seconds E|W]
AIRPORT_NAME *name*	name of the airport to be shown in the weather window (max. 8 characters)
QNH *num*	local pressure adjusted to the mean sea level (no influence)
ATIS letter	ATIS information designator (no influence)
TEMPERATURE *num*	local temperature (no influence)
DECLINATION *num*	value (angle) of magnetic declination (variation) at reference point [deg]
AIRPORT_IMG *x y rot*	position (from reference point) and rotation of the airport texture (airport.png) in 3D view [m], [deg]
AIRPORT_SCALE *num*	scale of the airport texture (how many meters are between the

	edges)
AIRPORT_RADAR_IMG *x y rot*	position (from reference point) and rotation of the airport texture (airport_radar.png) in radar [m], [deg]
AIRPORT_RADAR_SCALE *num*	scale of the airport texture (how many meters are between the edges)
FILTER_MIN *num*	minimal value of radar altitude filter [FL]
FILTER_MAX *num*	maximal value of radar altitude filter [FL]
HISTSTEP *num*	time interval between history points [ms]
HISTCOUNT *num*	maximal number of history points to store
LIGHT_STEP *num*	distance between runway/taxiway lights [m]
LABEL_ALPHA *num*	transparency (alpha channel) of label background (0-255)
LINE_WIDTH *num*	width of target lines, area borders and routes
USERNAME *name*	username (text)
USERROLE *role*	user role (text)
UDP_PORT_NUM *num*	UDP port for listening of ROSE aircraft update messages
TCP_HOST_IP *ip*	IP address of the host to connect to
TCP_PORT_NUM *num*	IP port of the host to use
AUDIO_PORT_NUM *num*	IP port of the audio host to use for TCP audio
AUDIO_PORT_IN *num*	IP port for listening to UDP audio
AUDIO_PORT_OUT *num*	IP port to which UDP audio will be sent
AUDIO_CONFIG *rate chan samp buff*	configures audio format: *frequency rate* (def. 8000, 11025, 22050, 44100, 48000, 96000), *channel count* (def. 1, 2), *sample bits* (8, def. 16), *buffer size* (1024, 2048, 4096, def. 8192)
AUTOCONNECT num	allows automatic connection to host after startup (*num*=1)
GND_TURN_MULT *num*	multiplier of TURN_STEP once the aircraft is on the ground
SMALL_DIST *num*	distance tolerance when reaching points [NM] (is then adjusted by current speed)
SMALL_SPEED *num*	speed tolerance when reaching IAS_min [kt]
SMALL_ANGLE *num*	angle tolerance when reaching radial [deg]
LABEL_Y *num*	offset of default label position from the target/point position [px]
LABEL_FONT *num*	font size of radar label (default 16) [pt]
LABEL_FAMILY *name*	*name* of font used for the label (e.g. Arial, Courier, Times)
POINT_FILE *file num1 num2*	file name to load the points from *num1* and *num2* defines the visibility of the points in radar and map (0=hidden, 1=only point, 2=point and label)
AREA_FILE *file num1 num2 alpha1 alpha2 style pattern*	file name to load the areas from *num1* and *num2* defines the visibility of the areas in radar and map (0=hidden, 1=only area, 2=area and label) *alpha1* defines the transparency level of area (0 to 255) *alpha2* defines the transparency level of area border (0 to 255) *style* defines line style (0 to 5): 0=NoLine, 1=Solid, 2=Dash, 3=Dot, 4=DashDot, 5=DashDotDot *pattern* defines fill pattern of area (0 to 14): 0=No, 1=Solid, 2..8=Dense, 9=Horiz, 10=Vert, 11=Cross, 12=ForwardDiag, 13=BackDiag, 14=DiagCross

ROUTE_FILE *file num1 num2 alpha style*	file name to load the routes from *num1* and *num2* defines the visibility of the routes in radar and map (0=hidden, 1=only route, 2=route and label) *alpha* defines the transparency level of line (0 to 255) *style* defines line style (0 to 5): 0=NoLine, 1=Solid, 2=Dash, 3=Dot, 4=DashDot, 5=DashDotDot
TYPE_FILE *file*	file name to load the aircraft types from
TARGET_FILE *file*	file name to load the targets from
SCALE_1 *num*	scale (zoom) of radar window (how many meters represent one pixel)
SCALE_2 *num*	scale (zoom) of map window (how many meters represent one pixel)
SCALE_3 *num*	scale (zoom) of PAR window – distance (how many meters represent one pixel)
SCALE_3Y *num*	scale (zoom) of PAR window – elevation (how many meters represent one pixel)
SCALE_3S *num*	scale (zoom) of PAR window – azimuth (how many meters represent one pixel)
AREA *name num1 num2*	sets the visibility of area in radar window (*num1*) and map overview (*num2*); area *name* must exist
ROUTE *name num1 num2*	sets the visibility of route in radar window (*num1*) and map overview (*num2*); route *name* must exist
POINT *name num1 num2*	sets the visibility of point in radar window (*num1*) and map overview (*num2*); point *name* must exist
CENTER_1 *x y*	sets the center of radar window to *x, y* [meters]
CENTER_2 *x y*	sets the center of Map overview to *x, y* [meters]
ZOOM_MULT *num*	zoom step multiplier for radar window and map overview
PAR_ZOOM_MULT *num*	zoom step multiplier for PAR window
GL_ZOOM_MULT *num*	zoom step multiplier for 3D TWR view
EFIS_MULT *num*	multiplier for EFIS size
BASIC_RINGS *num*	show range rings as basic (*num*=1) or full with angles (*num*=0)
WINDOWS *file*	loads the previously saved ini file with the settings of windows positions and statuses
PAR_LOGMULT *num*	coefficient for PAR logarithmic (distance) scale
PAR_YMULT *num*	coefficient for PAR elevation scale
PAR_SMULT *num*	coefficient for PAR azimuth scale
PAR_BLIPMULT *num1 num2*	coefficient for shrinking blip size in PAR and minimal blip size
RANDPX *num*	random shiver of PAR targets (including clutter) of *num* pixels
PAR_RWY_TH *lat lon*	PAR: position of runway threshold - latitude [degrees minutes seconds N|S] and longitude [degrees minutes seconds E|W]
PAR_RWY_DIR *num*	PAR: true heading of runway [deg]
PAR_GP_ANGLE *num*	PAR: required glide path angle to runway [deg]
PAR_TDP_FROM_TH *num*	PAR: distance between touch down point and runway threshold [m]
PAR_FROM_TDP *num*	PAR: distance between touch down point and PAR location (on runway) [m]
PAR_FROM_CL *num*	PAR: side distance between runway center line and PAR location

	[m]
PAR_SAFEELDEV *num*	PAR: safety margins for elevation (vertical angle from glide path to both sides) [deg]
PAR_SAFEELMIN *num*	PAR: minimal safety margins for elevation (vertical distance from glide path to both sides) [m]
PAR_SAFEAZDEV *num*	PAR: safety margins for azimuth (horizontal angle from center line to both sides) [deg]
PAR_SAFEAZMIN *num*	PAR: minimal safety margins for azimuth (horizontal distance from center line to both sides) [m]
PAR_SAFEDIST *start end visible*	PAR: distance [km] in which the safety margins shall be applied, show safety margins? (0 no, 1 yes)
PAR_FHEIGHT *value start end visible*	PAR: value [m] and distance [km] in which the floating line for height (FH) shall be applied, show? (0 no, 1 yes)
PAR_FELEVPAR *value start end visible*	PAR: value [m] and distance [km] in which the parallel floating line for glide path (FE) shall be applied, show? (0 no, 1 yes)
PAR_FAZPAR *value start end visible*	PAR: value [m] and distance [km] in which the parallel floating line for azimuth (FM) shall be applied, show? (0 no, 1 yes)
PAR_FAZANGLE *angle start end visible*	PAR: angle [deg] and distance [km] in which the converging floating line for azimuth (FA) shall be applied, show? (0 no, 1 yes)
PAR_FDIST *value visible*	PAR: distance [km] in which the floating line for distance shall be applied, show? (0 no, 1 yes)
PAR_HEIGHT1 *value start end visible*	PAR: value [m] and distance [km] in which the constant line for height (CH) shall be applied, show? (0 no, 1 yes)
PAR_HEIGHT2 *value start end visible*	PAR: value [m] and distance [km] in which the constant line for height (CH) shall be applied, show? (0 no, 1 yes)
PAR_HEIGHT3 *value start end visible*	PAR: value [m] and distance [km] in which the constant line for height (CH) shall be applied, show? (0 no, 1 yes)
PAR_HEIGHT4 *value start end visible*	PAR: value [m] and distance [km] in which the constant line for height (CH) shall be applied, show? (0 no, 1 yes)
PAR_ELEVPAR *value start end visible*	PAR: value [m] and distance [km] in which the parallel constant line for glide path (DE) shall be applied, show? (0 no, 1 yes)
PAR_AZPAR1 *value start end visible*	PAR: value [m] and distance [km] in which the parallel constant line for azimuth (DM) shall be applied, show? (0 no, 1 yes)
PAR_AZPAR2 *value start end visible*	PAR: value [m] and distance [km] in which the parallel constant line for azimuth (DM) shall be applied, show? (0 no, 1 yes)
PAR_AZANGLE *angle start end visible*	PAR: angle [deg] and distance [km] in which the converging constant line for azimuth (DA) shall be applied, show? (0 no, 1 yes)
PAR_DISTMARKS *num1 num2 num3*	PAR: maximal distance units in which the distance marks of 0.5, 1 and 5 shall be applied
PAR_CLUTTER_COUNT *num*	PAR: number of false targets (clutter) to be randomly generated in the coverage area
PAR_CLUTTER_MAXSIZE *num*	PAR: maximal size of false targets (blip size multiplier)
PAR_CLUTTER_AREA *dist above right dr ar rr*	PAR: define an area for random clutter generation (center in *dist* [km], *above* [m], *right* [m], size/radius in each dimension *dr*, *ar*, *rr* [m]). In case all parameters are 0, uses whole coverage area. Can be used repeatedly to define more clutter areas.
STARS_FILE *file*	file name to load the starry sky from

STAR_COUNT *num*	number of stars to load from the file
STAR_MULT *num*	maximum size of a star (px)
CONSTELLATIONS_FILE *file*	file name to load star constellation names from
CONSTLINES_FILE *file*	file name to load star constellation lines from
MACRO_FILE *file*	file name to load command macros from
LIMIT_GEO *num*	defines the distance from reference point (± degrees) in both Lat and Lon direction to generate the .limited topography files
GEO_LEVELS *n0 n1 n2 n3 n4 n5*	defines elevation for the different color levels of topography model [m amsl]
GEO_TRIANGLES *num*	defines how the topo model shall be displayed – smooth triangle strips (*num*=0) or simple flat triangles (*num*=1)
GEO_DISK *num*	defines the airport ground disk should be initially displayed - no (*num*=0) or yes (*num*=1)
CLIPPING_PLANE *near far*	distance of *near* and *far* clipping plane for normal objects (meters)
CLIPPING_PLANE_GEO *near far*	distance of *near* and *far* clipping plane for topo mesh (meters)
COLOR *type r g b*	defines different colors to be used by different types of elements each color part (*r*, *g*, *b*) can be from 0 to 255
	possible color types: B (background) f (FIR) s (sector) C (CTA) t (TMA) T (MTMA) z (CTR) Z (MCTR) p (prohibited area) r (restricted area) d (danger area) R (TRA) S (TSA) O (point) o (point label) X (target) x (target label) Y (target identification) G (ground route) g (groud route label) I (air route) i (air route label) E (scale and additional information) w (weather area) pH (PAR - horizon line) pG (PAR - glide path/center line) pS (PAR - safety margins) pF (PAR - floating lines) pC (PAR - constant lines - elevation) pc (PAR - constant lines - azimuth) pD (PAR - deviation (parallel) lines - elevation) pd (PAR - deviation (parallel) lines - azimuth) pA (PAR - angle deviation (converging) lines - elevation) pa (PAR - angle deviation (converging) lines - azimuth) p1 (PAR - distance mark 1)

	p5 (PAR - distance mark 5) p05 (PAR - distance mark 05) pm (PAR - runway threshold mark) sO (star spectral class O - Violet-Blue) sB (star spectral class B - Blue) sA (star spectral class A - White) sF (star spectral class F - Yellow-White) sG (star spectral class G - Yellow) sK (star spectral class K - Orange) sM (star spectral class M - Red) sW (star spectral class W - Gray)

Models definition

3D models (aircraft, ground vehicles, ...) are defined in the model folder.
Each models to be loaded is defined on one row in models.txt file with the following structure:
name scale
Meaning of the variables is:

name	name of the model (max. 20 characters, without space), refers to corresponding model files
scale	scale of the DXF model to transform its units to meters (minus rotates y axis = head x tail) e.g. 1 ft = 0.3048 m or 2 mm = 0.002 m

Each model consists of four files:

name.dxf	3D model in simple Autodesk DXF format - can contain only entities of 3DFACE type, parameters 0 (new entity), 62 (color), 10,20,30, 11,21,31, 12,22,32, 13,23,33 (verteces) - front side shall be oriented in -y axis (down)
	Different parts of the model shall be identified by different colors: 1 red (fuselage - side texture is applied) 2 yellow (rotating propeller) 3 green (front gear) 4 cyan (rear left gear) 5 blue (rear right gear) 6 magenta (wings - top texture is applied) 7 white/black (engines, propellers) 8 grey (parts with front texture – front texture is applied)
name_shadow.png	texture for aircraft shadow (top projection) - model shall be oriented down - shadow should have 50% transparency and gray color - background shall be transparent
name_side.png	texture for aircraft fuselage (side projection) - model shall be oriented to the left
name_top.png	texture for aircraft wings (top projection) - model shall be oriented down
name_front.png	texture for parts with front texture (front projection) - model shall be oriented towards us
	All textures should have following parameters: - texture sizes should be 512x512 pixels - the scale shall be set in the way the aircraft exactly fits into the rectangle (bigger model size equals texture size) - aircraft shall be centered in the direction of smaller model size In case a new model is created and textures are missing, TWRsim+ will automatically generate the basic textures from wired model.

Please note that using too complex or detailed models with high number of polygons can cause some performance issues.

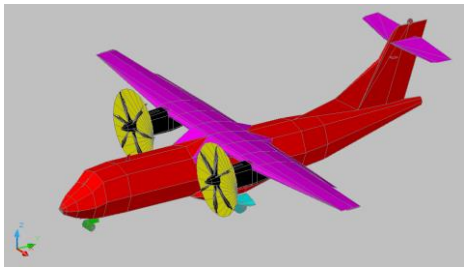

Figure 23: Aircraft model example

Aircraft types definition

Aircraft types and their performances are defined in a file set by TYPE_FILE variable.
One type of aircraft or vehicle is defined on each row with the following structure:
name class model FL_max IAS_max Mach_max IAS_norm Mach_norm IAS_taxi IAS_min ROC_norm ROC_max MTOW blip_size turn_step max_roll speed_step roc_step
Meaning of the variables is:

name	name of the aircraft type (max. 20 characters, without space)
class	aircraft class, possible classes: G (ground vehicle) H (helicopter) J1-J4 (aircraft with one to four jet engines) T1-T4 (aircraft with one to four turboprop engines) P1-P4 (aircraft with one to four propeller engines) B (building – not selectable and not moving)
model	3D model name to be used (in case of nonexisting model, some existing is used instead)
FL_max	maximal reachable Flight Level [hundreds of ft]
IAS_max	maximal Indicated Air Speed [kt]
Mach_max	maximal Mach speed [Mach]
IAS_norm	normal cruising speed [kt IAS]
Mach_norm	normal cruising speed [Mach]
IAS_taxi	normal taxi speed on the ground [kt IAS]
IAS_min	minimal take off speed [kt IAS] - if the aircraft does not reach this speed, it is not able to climb
ROC_norm	normal rate of climb/descend [ft/min]
ROC_max	maximal rate of climb/descend [ft/min]
MTOW	maximal take-off weight [kg]
blip_size	size of a primary radar (PAR) blip (coefficient, standard is 1.0)
turn_step	rate of performing a turn [deg/s] (how fast is the heading changed)
max_roll	maximal roll [deg] (roll also changes using *turn_step* rate) performing a turn also changes roll
speed_step	rate of speed change [kt IAS/s] (how fast is the speed changed)
roc_step	rate of ROC change [ft/min/s] (how fast is the ROC changed)

One of the aircraft types should be named "default" (defines the default performance).

Points definition

Points (twr position, waypoints, navigation aids, ..) are defined in a file set by POINT_FILE variable.
Repeated setting of this variable adds new points.
One point is defined on each row with the following structure:
name type lat lon
Meaning of the variables is:

name	name of the point (max. 20 characters, without space)
type	point type and subtype, possible types: p (airport point) e (en-route navigation aid in the FIR) E (en-route navigation aid outside FIR) a (navigation aid on civil aerodromes) A (navigation aid on military aerodromes) v (VFR entry and exit significant points to/from CTR) w (waypoints around an aerodrome) s (significant points - civil) S (significant points - military) t (parking stands)
	subtypes (second character) could be: v (VOR/NDB) V (VOR only) d (DME) n (NDB) o (other)
lat	latitude [degrees minutes seconds N|S]
lon	longitude [degrees minutes seconds E|W]

Point color is defined by variable *COLOR* with parameter *type*='O'.
The color of point label is defined by variable *COLOR* with parameter *type*='o'.
Point label position is defined by variable *LABEL_Y* (pixels under point position).
One of the points should be named "TWR" (defines tower position for 3D visualisation) - if it is not set in config file.

Areas definition

Areas (sectors, CTR, TMA, CTA, FIR, ...) are defined in a file set by AREA_FILE variable.
Repeated setting of this variable adds new areas.
One area is defined on several rows with the following structure:
$name type ul ll
area_border
...
Meaning of the variables is:

name	name of the area (max. 20 characters, without space)
type	area type (one character), possible types: f (FIR - flight information region) s (sector) C (control area) t (TMA - terminal control area) T (MTMA - military terminal control area) z (CTR - control zone) Z (MCTR - military control zone) p (prohibited area) r (restricted area) d (danger area) R (temporary reserved area) S (temporary segregated area) w (weather area)
ll	lower vertical limit of the area [FL (hundreds of ft)] - if *ll*=0 then it is replaced by the value of variable GND_LEVEL - if *ll*<0 then it is recalculated above the value of variable GND_LEVEL
ul	upper vertical limit of the area [FL (hundreds of ft)]

The border of the area is defined on the next rows in one of the following possibilities:

lat lon	sequence of points defined by their latitude and longitude
A radius fix	circular arc with the *radius* [NM] around previously defined *fix* (point) - the arc limits are set by previous and next point defined in the area - the arc with the smaller center angle will be used from the two possibilities
C radius lat lon	circle with the *radius* [NM] centered on the point with latitude *lat*, longitude *lon* - in case the area contains also some other points, the circle is just added to that area

Area color and the color of its label is defined by variable *COLOR* with the corresponding parameter *type*.
The area is filled with a transparency set by variable *AREA_ALPHA*. Only the border is shown in full color.
Are label position is set to the geometric center of the area.
The areas are displayed in the order of their loading (the order in which they appear in Area list window).
Please note that showing too many areas in the radar/map can cause some performance issues during redrawing.

Route definition

Routes (RNAV routes, air routes, SIDs, STARs, runways, taxiways, ...) are defined in a file set by ROUTE_FILE variable.
Repeated setting of this variable adds new routes.
One route is defined on several rows with the following structure:
$name type ul ll
route_commands
...
Meaning of the variables is:

name	name of the route (max. 20 characters, without space)
type	route type and subtypes, possible types: G (runway) - also shown in the 3D view g (taxiway) - also shown in the 3D view a (air route)
	subtypes define the color of RWY/TWY lights (second character) and the number of light rows (third character). The light colors could be: w (white) r (red) g (green) b (blue) y (yellow) The light rows could be 1, 2 or 3.
width	width of RWY/TWY [m], for air routes set to 1
ll	lower vertical limit of the route [FL (hundreds of ft)] - if *ll*=0 then it is replaced by the value of variable GND_LEVEL - if *ll*<0 then it is recalculated above the value of variable GND_LEVEL
ul	upper vertical limit of the route [FL (hundreds of ft)]

The route trajectory is defined on the next rows as a sequence of commands. All standard commands for navigating an aircraft (e.g. F *fix1* F *fix2*) can be used, just without target name *i*.

Route color and the color of its label depends on the route *type* and is defined by variable *COLOR* with the corresponding parameter *type*.
For ground routes (*type*='G' or 'g'), the color for the route is set by parameter *type*='G' in *COLOR* variable, their label color is set by parameter *type*='g' in *COLOR* variable.
For air routes (*type*='a'), the color for the route is set by parameter *type*='I' in *COLOR* variable, their label color is set by parameter *type*='i' in *COLOR* variable.
The route is displayed as a sequence of points defined by command F.
Route label position is set to the geometric center of the middle segment of the route (or closer to the route beginning in case of even number of segments).

Target definition

Targets (aircraft, ground vehicles, ...) which should be loaded at a start time are defined in a file set by TARGET_FILE variable.
One target is defined on several rows with the following structure:
$name type lat lon
commands
...
Meaning of the variables is:

name	name of the target (max. 20 characters, without space)
type	name of the existing aircraft type previously defined by *TYPE_FILE* variable in parameter *name*
lat	latitude [degrees minutes seconds N|S] - position on which the target should appear
lon	longitude [degrees minutes seconds E|W] - position on which the target should appear

The next rows define commands, which are to be performed once the target is created. All standard commands for controlling a target can be used, just without target name *i*..
Typically, the immediate setting of target heading, altitude and speed (possibly followed by flight plan definition) should be done.

Target color is defined by variable *COLOR* with parameter *type*='X'.
The color of target label is defined by variable *COLOR* with parameter *type*='x'.
Target color during identification (flashing) is defined by variable *COLOR* with parameter *type*='Y'.
Default target label position is defined by variable *LABEL_Y* (pixels under target position).
Targets can be put into the exercise also manually by using /N or automatically by loading the command log.

Stars definition

Stars on the skydome are defined in a file set by STARS_FILE variable.
One star is defined on each row with the following structure:
RA Dec mag class subclass
Meaning of the variables is:

RA	right ascension of the star (angle eastward along the celestial equator from the vernal equinox)
Dec	declination of the star (angle north of the celestial equator, along the hour circle passing through the star)
mag	apparent magnitude of the star
class	spectral class of the star first character shall be one of the following: O, B, A , F, G, K, M or W
subclass	subclass of the star (N/A) should be one of the following: I, II, III, IV, V

Recalculation of the stars positions is done once per a defined number of frames (currently 120).
Please note that showing too many stars can cause some performance issues.

Star constellations definition

Star constellations on the skydome are defined in a file set by CONSTLINES_FILE variable.
Constellations are defined as a sequence of line, where each line is defined on one row with the following structure:
RA1 Dec1 RA2 Dec2
Meaning of the variables is:

RA1	right ascension of the first point
Dec1	declination of the first point
RA2	right ascension of the second point
Dec2	declination of the second point

Star constellation names definition

Star constellation names on the skydome are defined in a file set by CONSTELLATIONS_FILE variable.
One constellation name and its position is defined on each row with the following structure:
name RA Dec
Meaning of the variables is:

name	name of the constellation (max. 20 characters, without space)
RA	right ascension of the constellation
Dec	declination of the constellation

Constellation names are just labels which could appear on the sky dome.

Topography definition

Topography (surface elevation model) is defined in the topo folder.
Each topography file to be loaded is defined on one row in topo.txt file with the following structure:
name add skip
Meaning of the variables is:

name	name of the topography model (max. 20 characters, without space), refers to corresponding topography file in .asc format
add	how many rows and columns are to be duplicated at the tile right and bottom end (some data sources have one extra row and column of the neighboring tile in order to connect the tiles smoothly)
skip	how many vertices are to be skipped in both directions in order to simplify the model

In case there is also a corresponding file name_tex.jpg with satellite image of the topography model, this texture is applied. If the file is missing, color of topography elevation model is calculated automatically. The name_tex.jpg file needs to be e.g. 2048x2048.

Each topography file is an .asc (ascii file) and has the following structure:
ncols *cols*
nrows *rows*
xllcorner *lon*
yllcorner *lat*
cellsize *size*
NODATA_value *value*
elev_data
...
Meaning of the variables is:

cols	number of columns of the elevation data table
rows	number of rows of elevation data table
lon	longitude (degrees with decimals) of the lower left corner of the elevation data table
lat	latitude (degrees with decimals) of the lower left corner of the elevation data table
size	size of one cell of elevation data table (degrees of latitude/longitude) e.g. 0.00083333333333333 for 3 minutes of arc per cell
value	value of a cell with no elevation data (usually -9999)
elev_data	table [rows x cols] of elevation data representing elevation at the given latitide and longitude [m]

Please note that using large topography files can cause some performance issues.
When using more topography files with large number of rows/columns, the rounding and incremental calculations may cause some voids at boundaries.
While processing the topography data, TWRsim+ also generates .limited file, which limits the input elevation data to the area around the reference point (± LIMIT_GEO value).

Clouds definition

Clouds (and possibly other meteo phenomena) are defined in the weather folder.
Each type of clouds to be loaded is defined on one row in clouds.txt file with the following structure:
name
Meaning of the variables is:

name	name of the clouds (max. 20 characters, without space), refers to corresponding cloud file

Each type of clouds consists of one file:

name.png	texture for this type of clouds (top projection) - clouds should be semi-transparent - background shall be transparent - texture size should be 512x512 pixels - selected type of clouds is shown as a horizontal layer centered at the sky dome

Macros definition

Command macros can be defined in a file set by MACRO_FILE variable.
Repeated setting of this variable adds new macros.
One macro is defined on several rows with the following structure:
$name "description"
macro_commands
...
Meaning of the variables is:

name	name of the macro (max. 20 characters, without space) macro must be used with the prefix $, e.g. $land
description	text description of the macro (for understanding, it is not used)
macro_commands	sequence of any standard commands

Before processing a command, list of defined macros is checked and all those used in the command are replaced by the defined macro_commands.

Other

During the run, some debug messages are stored in the debug.txt file.
See product documentation or contact the author for more details.

Structure of TWRsim folder:

/data	- default directory for loading the data set
license.dat	- license key file (simulation will not run without proper license)
...	- data set files referenced in config.txt
/exercises	- default directory for saved exercises (command logs)
...	- saved exercises (command logs and configuration) windows position configuration is saved in .ini file of the same name
/img	- directory with default images and textures used by the application
...	- .png and .wav files with images, textures and sounds
/model	- directory with models of aircraft and ground vehicles for 3D visualisation
models.txt	- definition (list) of models to be used
_default.dxf	- 3D model of the aircraft in simple DXF format
_default_shadow.png	- shadow texture for the given model
_default_side.png	- fuselage texture (side projection) for the given model
_default_top.png	- wings texture (top projection) for the given model
_default_front.png	- front texture (front projection) for the given model
...	- definitions of other models
/topo	- directory with topography definition for 3D visualisation
topo.txt	- definition (list) of topography files to be used
prague.asc	- example topography model file
prague.asc.limited	- example processed topography output file
prague_tex.jpg	- topography texture (satellite image) for the given topography model
...	- definitions of other topography models
/weather	- directory with weather definition (clouds layers) for 3D visualisation
clouds.txt	- definition (list) of clouds textures to be used
cirrus.png	- example clouds texture (Cirrus, 5/8 coverage)
...	- definitions of other clouds textures
config.txt	- application and data set configuration
dark.css	- style sheet for dark colors of the Qt widgets if file is missing, uses standard styles
debug.txt	- output file with debug information
QtCore4.dll QtGui4.dll QtNetwork4.dll QtOpenGL4.dll QtMultimedia4.dll	- dynamic libraries used by the application
readme.htm	- help file with application description and all necessary information
release.txt	- release notes and version history
TWRsim.exe	- application main executable file
twrsim.png	- application logo

License

Application needs to be properly licensed to allow running the simulation.
License is bound to the computer with its hardware fingerprint.

License can be provided by the author after receiving the code shown on the license screen.
License file (license.dat) is to be located in the /data folder.

Once the program is licensed, the license screen is not shown anymore and the simulation can be started without any restrictions.

To obtain the license, please contact the author:

Petr Dvorak
petr@papadelta.cz
www.papadelta.cz

Application requires Qt libraries and Visual C++ Redistributable package for VS 2008 (vcredist_x86.exe). Qt libraries are distributed under GNU LGPL v2.1.
No installation or third party software is required.