EmptySkies Game Manual

Description

This document describes the usage of EmptySkies (hence abbreviated as ES), a non-commercial play aid for the board game Black Cross / Blue Sky (hence abbreviated as BCBS) by Blue Sky Enterprises (http://www.blueskygameworks.com). The program was written by Tobias Boehler (BoardGameGeek.com user: bolero) and is not affiliated with the publisher of BCBS or any official product. Permission was given by the game designer.

Purpose

ES allows a single player to play scenarios of the original game without available human opponents. It can be considered as some kind of game artificial intelligence (AI), although this term does not hold in a scientific sense. In fact, the underlying logic guiding the ES game decisions follows a heuristic approach (see the section Information). Still, these heuristics provide a more challenging opponent than the solitaire rules for the single-player scenario included in the original game. AI-controlled aircraft in ES may

- move and fire to achieve their prescribed goals
- use opportunities to destroy other aircraft
- perform complex maneuvers such as side-slipping and braking
- change attitudes to get into advantageous positions.

Only a subset of BCBS rules have been implemented, roughly corresponding to the basic rules set for dogfights. In this version (1.0), the following rules are **not** implemented:

- emergency high-g turns, looping, rolling
- critical hits and resulting aircraft behavior
- formation rules for fighter aircraft
- explosion hexes

However, it is completely possible to play basic scenarios, e.g., of type *Approach*.

Requirements

The original game BCBS is required to play. The ES application does not contain any rules or required game charts and tables. Familiarity with the BCBS rules is necessary to play the game. In this respect, the ES program can be seen as providing an additional/alternative game board.

The ES application is programmed in the JAVA language, which is a platform-independent programming platform maintained by Oracle (http://www.java.com). In order to start the ES program, a runtime environment must be installed and accessible on your computer. It can be downloaded for free for all current computer operating systems from the JAVA website.

Installation

The ES software can be downloaded as a an archive file in the compressed ZIP format (http://en.wikipedia.org/wiki/ZIP_(file_format)). Simply decompress the file into an accessible directory of your choice with a suitable decompression tool, e.g., WinZip. There is no need to perform an additional installation of the software apart from extracting the original archive. The decompressed (unpacked) ZIP archive contains:

- emptySkies.jar, a JAR (JAVA archive) file containing the program code and resources
- HexLibSettings.xml, a settings file allowing to change some program options of the internally used hex-board library written by the author
- manual.pdf, this documentation and manual file in the PDF format
- LICENSE.txt, a plain text file containing information about legal software usage
- README.txt, a plain text file containing brief information

Users of ES are free to distribute or modify all files at their own risk under the license conditions (see the license text in this document or in the contained LICENSE.txt file). The JAR file is a special kind of ZIP file that contains all required compiled program files (.class files) as well as the resources (data, images, etc.) required to run the application.

Usage

This section describes a typical execution and usage of the ES application.

Running EmptySkies

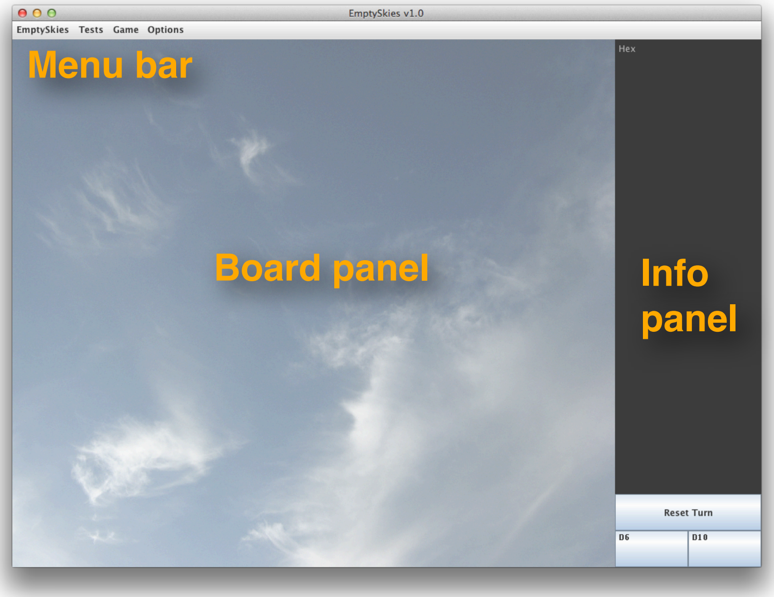
The application requires an invocation of the JAVA runtime environment as follows:

```
java -jar emptySkies.jar
```

This will start the application class in the JAR file that contains the main() method. Depending on the operating system, ES is runnable by double-clicking the JAR file or typing the commands above in a shell window. A graphical window will appear on screen.

Basic usage

After starting ES, the main window (see screenshot below) appears. The main window is divided into the *board panel*, the *info panel* to the right, and the *menu bar* at the top of the application.The board panel displays the game board, including the board hex fields (or hexes) and the game pieces. The info panel provides additional information about the game pieces each time the mouse is hovering over one of them. It also contains a *dice panel* at the bottom-right corner of the window, which allows to quickly roll dice (D6 or D10) to resolve combat and movement situations, and a *Reset Turn* button.

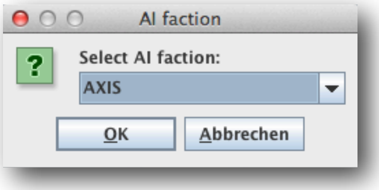


From the menu bar, four sub menu items are selectable (at the top of the application):

EmptySkies		
	About:	displays an about dialog, showing information about ES.
	Quit:	close this application and release all resources.
Tests (do not use during active games)		
	Off:	turns off all tests activated.
	Drawing test:	draws a default texture and colors the neighboring hexes.
	Pathfinding test:	dragging from any unit shows the current path to the hex over which the mouse is currently hovering.
	Arcs test:	dragging the hex of any unit shows the 120° arc in the current direction of this unit.
Game (playable scenarios)		
		Solitaire - The Lone Wolf
		Standard 01 - The Opening Shots
Options (general game options)		
	Execute moves:	immediately position any unit to its new position after movement. This does not change other unit attributes such as altitude, attitude, momentum, etc. The user still needs to make these changes.

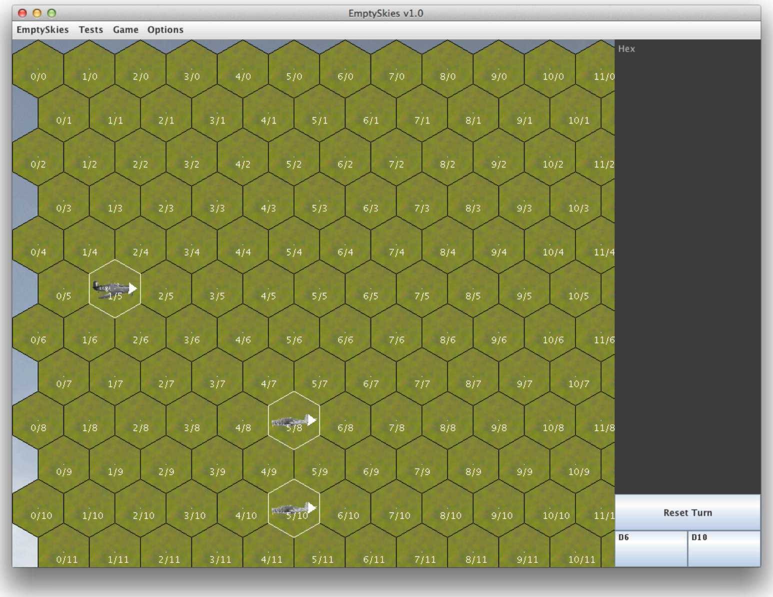
Playing

Choose one of the playable scenarios from the menu. Supported scenarios are choosable from the Game menu. Any choice will automatically load the associated board game map and place all pieces. For the details and goals of each unit refer to the BCBS manual.

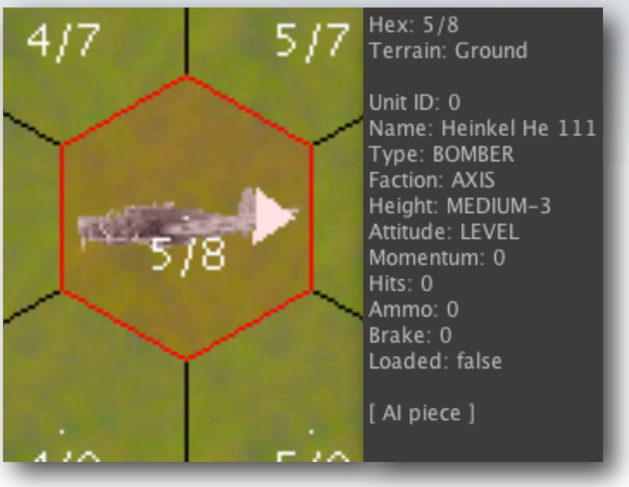


Select AI faction: AXIS

Initially, a small choice dialog will open and prompt for the side to be controlled by the AI logic (see screenshot to the right). One can choose between the Axis or Allies side from the corresponding drop-down menu. Note that you cannot change this at a later point. After choosing one of the two opposing sides, the game map will automatically be loaded and all relevant units will be placed upon it (see screenshot below).



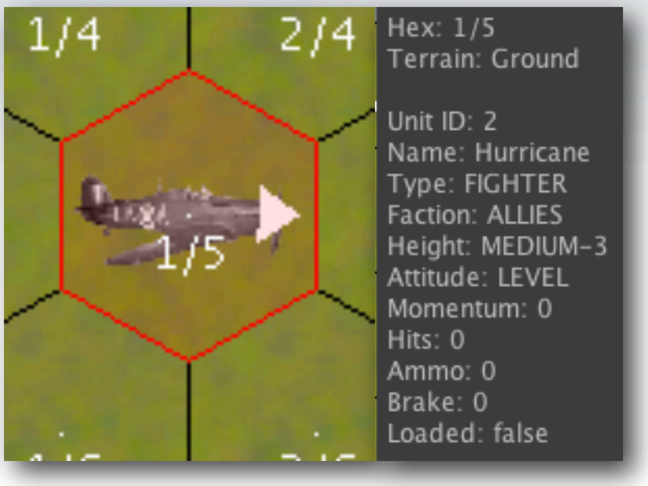
With the loaded game map, you may then execute all game turns as you would normally do with the physical game boards of BCBS. It is advisable to use both, as the original game board provides a better overview and has special graphical tokens. Hovering the mouse over any visible unit shows the corresponding aircraft information in the right-sided info panel. Here are two examples:



Hex: 5/8
Terrain: Ground

Unit ID: 0
Name: Heinkel He 111
Type: BOMBER
Faction: AXIS
Height: MEDIUM-3
Attitude: LEVEL
Momentum: 0
Hits: 0
Ammo: 0
Brake: 0
Loaded: false

[AI piece]

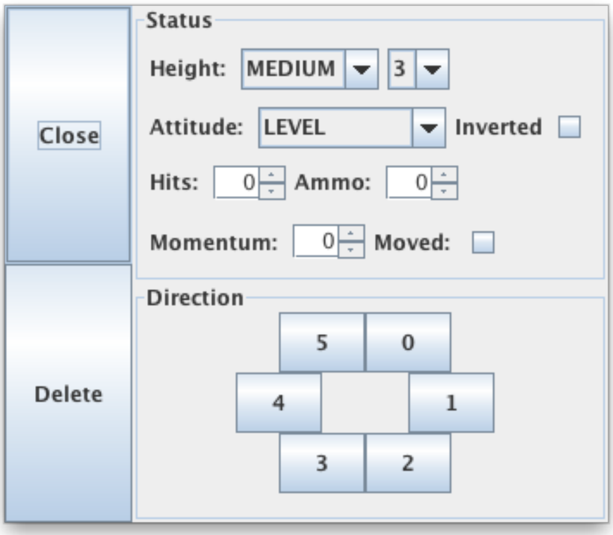


Hex: 1/5
Terrain: Ground

Unit ID: 2
Name: Hurricane
Type: FIGHTER
Faction: ALLIES
Height: MEDIUM-3
Attitude: LEVEL
Momentum: 0
Hits: 0
Ammo: 0
Brake: 0
Loaded: false

Note that the info panel also shows if the unit is AI-controlled or not, and if it has moved.

Any aircraft can be displaced anytime by dragging it (using the left mouse button) from its current location to the desired target location. After dropping the aircraft onto its target hex, a status dialog will appear right next to the mouse position:



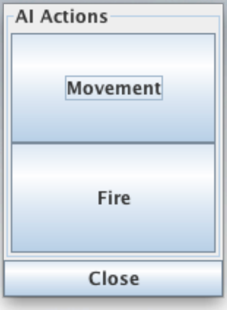
Status
Height: MEDIUM 3
Attitude: LEVEL Inverted ☐
Hits: 0 Ammo: 0
Momentum: 0 Moved: ☐

Here, all required adjustment must be made that describe the current unit‘s status after making the move. For example, after moving the aircraft to its new location, the direction and momentum are typically adjusted. This is also required for AI-controlled units. The dialog can be closed using the *Close* button, which is conveniently positioned next to the mouse cursor to accelerate the closing operation. Units can be deleted using the *Delete* button. This is required if the unit has been destroyed or left the game board.

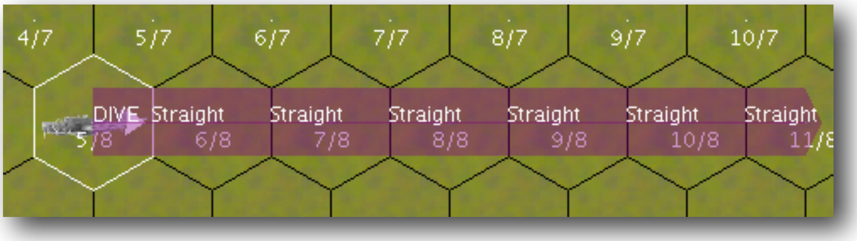
Resolving any game turn must be completely performed by the player. Here, ES only serves as the play aid it is intended to be. No turn scheme has been implemented.

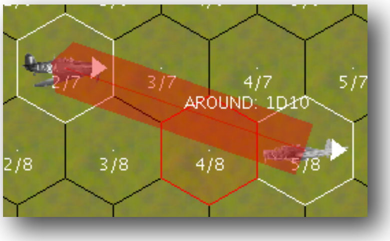
In order to resolve AI-movement and firing actions, all units designated as being AI-controlled can be activated. For any such unit, right-clicking an AI-controlled unit will bring up a corresponding *AI Actions* menu (see screenshot to the right). Closing will cancel any kind of AI action. Otherwise, two options are available:

Movement: computes the new location of this game piece
Fire: computes the targets to attack with this game piece



The results of either options will be displayed graphically as paths towards the target locations. In case of movement, each hex-by-hex movement will be explicitly stated. Currently, the movement maneuvers supported are: *Straight*, *Turn*, *Brake* and *Slip*. Direction-changing maneuvers will also indicate the direction chosen. At the beginning of the movement path, the new attitude will be announced, which can be either *Level*, *Dive*, *Steep Dive*, *Climb* or *Steep Climb*. Firing actions will point to each target unit, with the fire factors used being stated accordingly. Here are exemplary movement and firing actions:

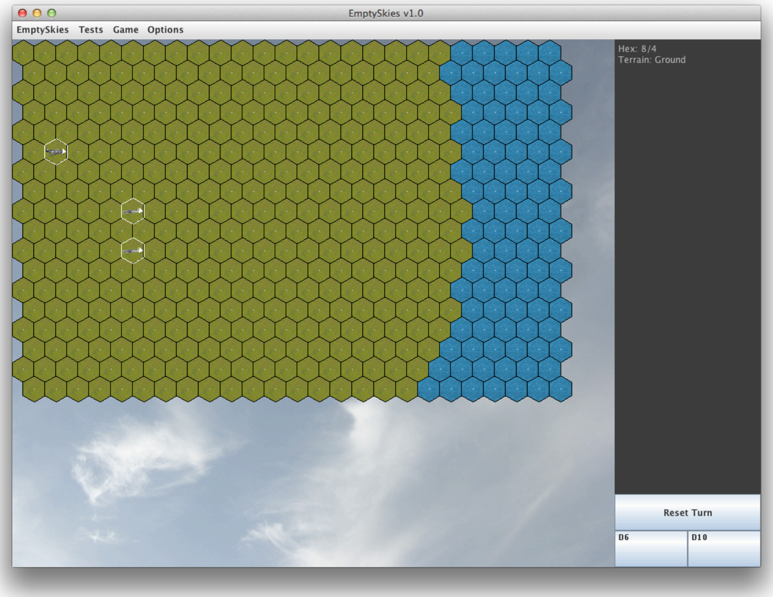




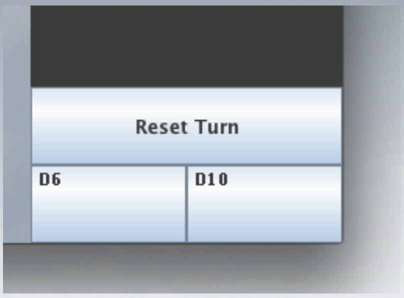
Both types of paths will remain visible until any mouse button is pressed again. In order to simulate opponent movement and firing, simply activate the two corresponding actions whenever required during the game turn. Setting the *Execute Moves* option (see above) automatically moves any AI-controlled to its destination position and adjusts its status. If no fire targets can be reached, a corresponding message box appears (see right) to indicate that the selected unit is unable to fire at its enemies.

No valid fire targets!

To get an overview of the current game situation, simply move the mouse wheel. In the normal view, moving the mouse wheel upwards will display an overview map with reduced information (see screenshot to the right). Still, any kind of detailed info is shown in the right-hand info panel. The overview map allows you to quickly analyze the current game situation. Moving the mouse wheel downwards while hovering over any hex automatically centers the view at that hex location. This way, it is possible to quickly traverse the map and access different aircraft formations.



Hex: 8/4
Terrain: Ground



To quickly resolve any required dice rolls, the dice panel has been integrated into the bottom-right part of the main window (see screenshot to the right). Two different types of dice can be rolled by simply clicking on each panel. For example, clicking once on the *D6* panel will display a random number from 1 to 6, while a click on the *D10* panel will show a random number from 1 to 10. Each mouse click generates a new random number.

Directly above the dice panel, the *Reset Turn* button is located. Pressing this button resets all *Moved* flags for all AI-controlled units. This can also be done manually per selected unit via the actions dialog. However, the *Reset Turn* button greatly simplifies this handling. Resetting the flags per turn is necessary to indicate which units have been already moved. This in particularly influences the way formations are maintained, as units try to stay together after movement as much as possible.

Additional options

General settings for the application are stored in the *HexLibSettings.xml* settings file. All settings are provided as XML (Extended Markup Language) tags with appropriate values. These settings can be used to customize the appearance of the ES game board:

WindowSettings
- *Width*: the desired height of the main window.
- *Height*: the desired width of the main window.
- *Fullscreen*: toggle fullscreen modes:
  - 0 - fullscreen off (set to width/height above)
  - 1 - fullscreen windowed
  - 2 - fullscreen exclusive (nothing else visible except ES)
- *HexDiameter*: diameter of all hexes in pixels.

You can freely adjust any settings values, but be careful in changing them, particularly the exclusive fullscreen mode. Use these settings at your own risk.

Information

This section contains further game-relevant information and resources used in ES.

Supported game data

Currently, the ES play aid supports only a small sub set of the original scenarios and aircraft found in BCBS.

Scenarios
- The Lone Wolf
- The Opening Shots

Aircraft
- Hawker Hurricane
- Junkers Ju 87B Stuka
- Heinkel He 111

Further scenarios might be integrated in future releases.

AI heuristics

The game currently uses three logic layers of AI heuristics:

- a *piece-level AI* logic guiding decisions for each of the displayed game units. Particularly, during movement computations this will
  1. compute an exhaustive list of movement locations for all types of attitudes and maneuvers possible. These locations will be stored in form of a movement tree. Given the few collision obstacles and short movement distance, this is preferable over influence map computations.
  2. evaluate all locations and compute scores for
     - achieving the goal of this unit
     - attacking other aircraft
     - being attacked by other aircraft
     - avoiding enemy aircraft
     - hunting enemy aircraft
     - avoiding collisions
- a *group-level AI* logic guiding groups of game pieces, particularly aircraft formations. Airplanes will try to maintain formation or return into formation if possible. This is particularly true for bombers. Corresponding scores are computed.
- a *strategic-level AI* logic rewarding progress towards the current scenario goals (e.g., escaping any hunters) for individual game pieces, and penalizing any actions preventing the achievement of these goals. Corresponding scores are computed.

Given all scores, the AI logic chooses the most efficient (best-scoring) location and the movement path to it. There might be different valid paths to the same hex location.

Firing actions are computed in a simpler fashion, as only the targets in range have to be considered and no blocking of firing by other units must be taken into account. The computation of firing actions also considers opportunity targets.

TODOs

- implement further types of maneuvers occurring in the game
- allow simpler usage of arc and distance computation tools
- show explosion hexes graphically
- ...

Useful hints

- the current AI implementation favors attacking different targets instead of focusing the fire on a single one.

Textures

All textures and resources can be found in the jar-file, which is a ZIP-format archive.

Heinkel He 111: Bundesarchiv, Bild 101I-647-5211-33 / Wilzek / CC-BY-SA
http://upload.wikimedia.org/wikipedia/commons/d/dd/Bundesarchiv_Bild_101I-647-5211-33%2C_Flugzeug_Heinkel_He_111.jpg

Junkers Ju 87B: Bundesarchiv, Bild 183-J16050 / CC-BY-SA
http://upload.wikimedia.org/wikipedia/commons/d/d3/Junkers_Ju_87Ds_in_flight_Oct_1943.jpg

Hawker Hurricane: public domain image, http://upload.wikimedia.org/wikipedia/commons/8/80/Hurricane_IIC_87_Sqn_RAF_in_flight_1942.jpg

Default: Comco Ikarus C42-B ultralight aircraft, image photographed by the author
Ground: image photographed by the author
Water: image photographed by the author
Canvas background, about dialog: image photographed by the author

License

The software is distributed as a jar-file containing pre-compiled byte code and licensed under a version of the MIT/X11 license (http://en.wikipedia.org/wiki/MIT_License):

EmptySkies v1.0

Copyright (c) 2013 Tobias Boehler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish and/or distribute copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.