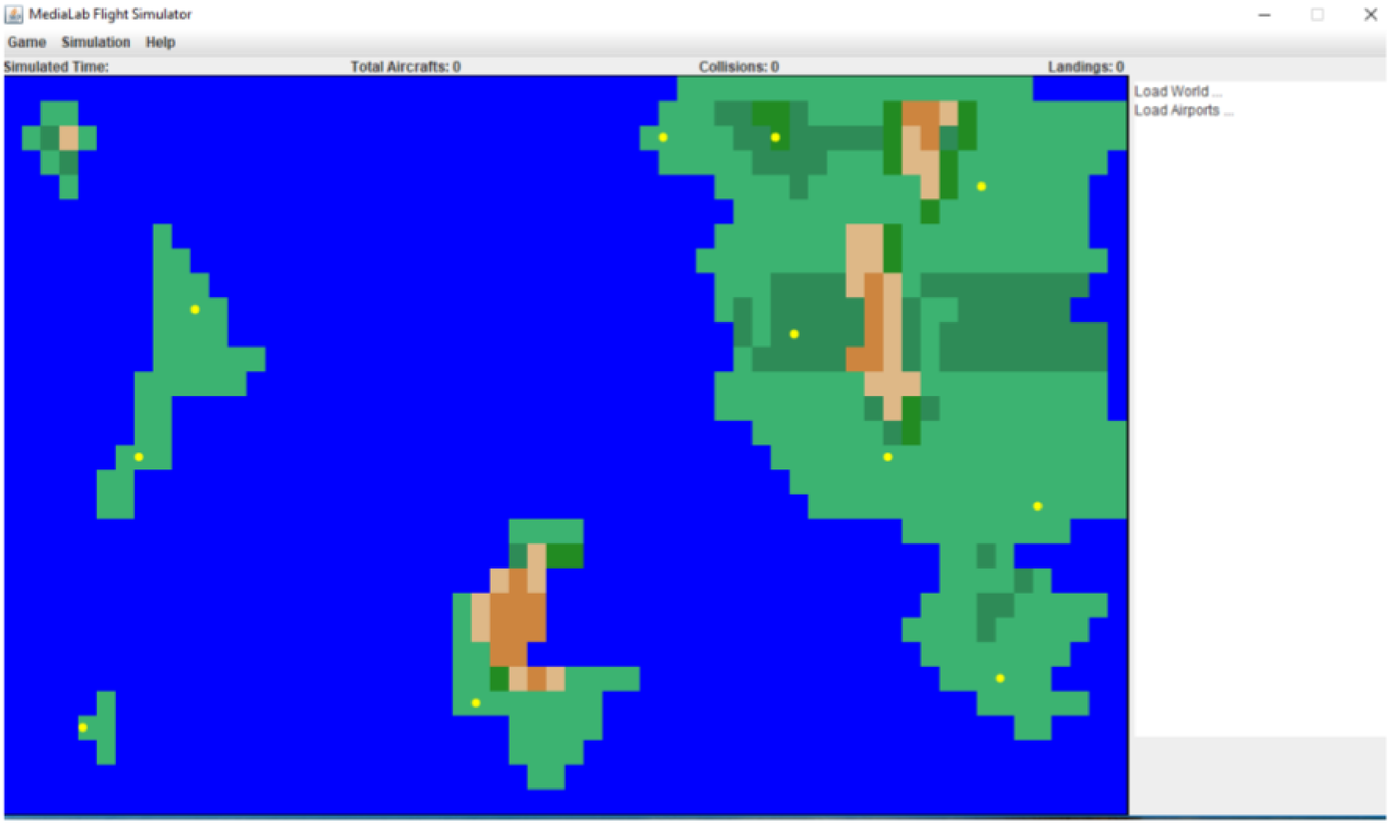
Aiplanes Simulation

It is a simulation of planes moving in an overhead area. In the application there will be different types of airplanes moving within a predetermined space, which will be the map of the simulation world.



Simulated Time: Total Aircrafts: 0 Collisions: 0 Landings: 0

Package: Airplanes

CLASS : Small

→ Describes all the features of single-engine airplanes.

CLASS : Middle

→ Describes all the features of Turboprop airplanes.

CLASS : Big

→ Describes all the features of Jet airplanes.

CLASS : Flight

→ It contains all the necessary information we need to know about each flight. The comments describe the operation of each parameter.

Package : Airports

CLASS : airports

→Describes the airports.

Package : Graphics

CLASS : createGraphics

→It contains the Jpanel, which is created by MakeJpanel(). Jpanel extends with:

• Map through the MakeMap() function. Here are used the readMapFile and readAirportsFile classes.

• Information bar via the infoList() function.

• The timer bar and the current number of Airplanes, Conflicts and Landings with the MakeNorth() function.

• Menubar is calling his own class.

→ It contains the functions:

- AirplanesMove(), which refreshes the airplane position on the map taking into account its orientation
- returnPrevious(), restores the previous position on the airplane, checking whether there was an airport in it.

CLASS : readMapFile, readAirportsFile

→ They read file of the map and the airports respectively.

CLASS : Menubar, popWindowAircrafts, popWindowAirports, popWindowFlights,popWindowLoad

→ The MenuBar what button is pressed by the user.

- If a button from the “Simulations” is pressed, calls the popWindow classes that print the requested information
- Start: the program starts from the beginning in a new window and the previous one stops. Flights are read from the readFlights class.
- Load: popWindowLoad appears and prompts the user to enter the file name. When enter is pressed, starts automatically.

CLASS : readFlights

→ Reads the flight record. It is important to note that, depending on the airplane type of each flight, all valid flights are divided into 3 lists: smallFlightsList, middleFlightsList, bigFlightsList

CLASS : controlClass (most important class)

→ Every time the timer increases, the following are checked / updated::

1. The counter is increased and the information is updated on the top bar.

→For each of the valid flight lists checks:

2. If it is time to start, checkStartTime()
3. If it is active (active =1)
4. Calculate the distance that has traveled, calcDistance().
5. The remaining fuel checkFuels().
6. The height at which it is located checkHight().
   ---if it's still active and not crashed---
7. Whether the airplane needs to be moved to the next box shouldImove().
8. If an airplane has already landed so it must be removed from the map and marked that it is over (active=2) checkLandings.
9. → If it must move to the next box:
   1. calls move* which moves it optimally
   2. check whether it has reached landing position
10. Whether there is a crash between planes

Package : MyMain

CLASS : MyMain

→It contains the main function of my program. It calls the functions to create the graphics and then the controlClass class from which all the moves are controlled.

NOTES

✗ The implementation of the function which checks if two airplanes should be crashed is only applied if both planes are in the same box.