{EPITECH}

MY RADAR

THE AIR TRAFFIC CONTROL PANEL

MY RADAR

binary name: my_radar
language: C
compilation: via Makefile, including re, clean and fclean rules

- ✓ The totality of your source files, except all useless files (binary, temp files, obj files,…), must be included in your delivery.
- ✓ All the bonus files (including a potential specific Makefile) should be in a directory named bonus.
- ✓ Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

In this project, you are asked to render a 2D visualization panel simulating air traffic (AT).

The air traffic is a complex system with very strict rules to manage a large number of aircrafts. For this first version of the simulation, you have to consider 2 types of entity:

- ✓ aircrafts,
- ✓ control towers.

The basic rules for the my_radar are as follows:

- ✓ aircrafts fly from given places to other ones.
- ✓ aircrafts appear on the simulation panel when they take off.
- ✓ aircrafts disappear from the simulation panel when they land on.
- ✓ aircrafts move in a straight line at given constant speeds.
- ✓ aircrafts colliding with another one is destroyed and disappear from the simulation panel.
- ✓ control areas allow aircrafts to collide with each other without being destroyed and they can continue on their way.
- ✓ control towers don't move and have control areas on the map.
- ✓ control towers appear on the simulation panel at launch.

This project will seem harder than the previous one to you. Take your time and work on it iteratively and it can be done !

Requirements

MUST

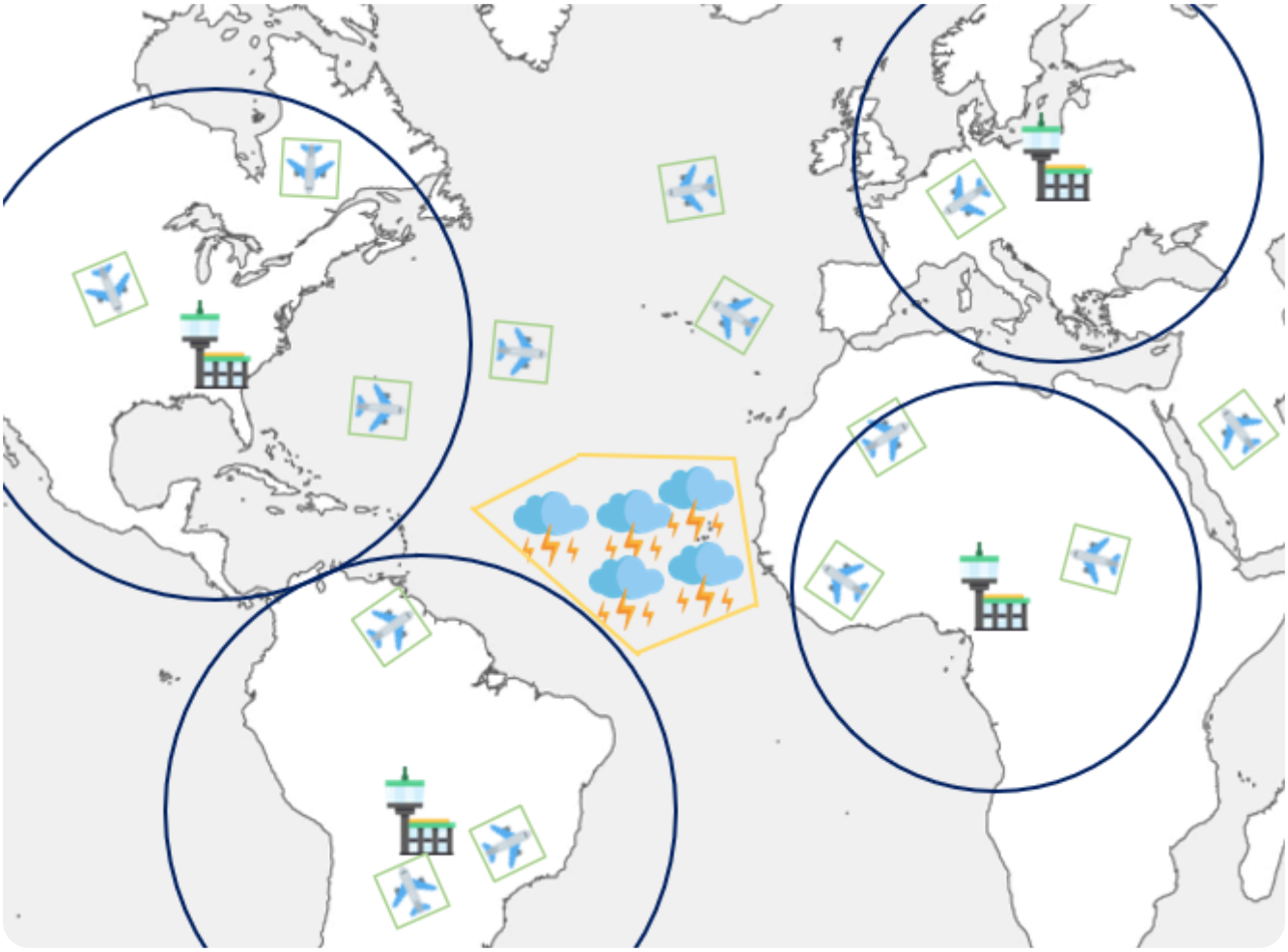
- ✓ The program **must** take exactly one parameter.
- ✓ The program **must** accept any filepath as argument. It corresponds to the file containing the script for the simulation containing characters and integers.
- ✓ The program **must** displayed an error message if the script is not found or incorrect.
- ✓ Your window **must** be 1920x1080 pixels.
- ✓ The window **must** be closed using events.
- ✓ The program **must** display aircrafts' hitbox and control towers' areas.
- ✓ The program **must** manage as many displayed entities as possible.
- ✓ Aircrafts **must** have 20x20 square shaped hitboxes.
- ✓ Aircrafts' hitbox and sprite **must** be axis-aligned.
- ✓ Control towers **must** have circled shaped control areas.

SHOULD

- ✓ The program **should** display and update a timer in seconds at the top-right corner of the window.
- ✓ The program **should** stop once all aircrafts have landed on or have been destroyed.
- ✓ Aircrafts **should** be able to take off after a given delay.
- ✓ Aircrafts and control towers **should** be displayed using sprites.
- ✓ The program **should** accept the "-h" option, then display the usage.
- ✓ The program **should** be able to switch visibility of hitboxes and areas by pressing the key 'L'.
- ✓ The program **should** be able to switch visibility of entities' sprites by pressing the key 'S'.
- ✓ Possible user interactions **should** be explicitly displayed in the usage.

COULD

- ✓ Aircrafts' hitbox and size **could** be in a 3D space.
- ✓ Aircrafts **could** cross any side of a screen and appear on the opposite one.
- ✓ Aircrafts' hitbox and sprite **could** rotate depending on the aircrafts' direction.
- ✓ Aircrafts' hitbox **could** follow the sprite boundary.
- ✓ Aircrafts' hitbox **could** have different sizes.
- ✓ Control towers' areas **could** cross any side of the screen and affect the opposite one.
- ✓ The program **could** manage floating numbers coming from the script.



Script

The script file contains all the information about the simulation and the entities.
Below is a example of a script file. You are free to use any kind of script formatting as long it's described in a .legend file.

Aircrafts are described by:

- ✓ The letter 'A',
- ✓ Two integers corresponding to the departure x- and y-coordinates,
- ✓ Two integers corresponding to the arrival x- and y-coordinates,
- ✓ One integer corresponding to the aircraft's speed (in pixels per second),
- ✓ One integer corresponding to the delay (in seconds) before the aircraft takes off.

Control towers are described by:

- ✓ The letter 'T',
- ✓ Two integers corresponding to the control tower x- and y-coordinates,
- ✓ One integer corresponding to the radius of the tower's control area.

Entities are separated by a '#cr'.
The pieces of information are separated by *tabs* or *spaces*.

Terminal

```
~/B-MUL-100> cat scripts/example.rdr
A  815 321  1484 166  5  0
A  1589 836  811 936  2  0
A  202 894  103 34  3  0
T  93 47  19
T  49 56  25
```

Usage

```
Terminal
~/B-MUL-100> ./my_radar; echo $?
./my_radar: bad arguments: 0 given but 84 is required
retry with -h
84
```

```
Terminal
~/B-MUL-100> ./my_radar -h
Air traffic simulation panel
USAGE
  ./my_radar [OPTIONS] path_to_script
    path_to_script    The path to the script file.
OPTIONS
  -h                print the usage and quit.
USER INTERACTIONS
  'L' key           enable/disable hitboxes and areas.
  'S' key           enable/disable sprites.
```

Authorized functions

Here is the full list of authorized functions.

from the C library

- ✓ malloc
- ✓ free
- ✓ stat
- ✓ memset
- ✓ rand
- ✓ srand
- ✓ time (only with srand)
- ✓ (f)open
- ✓ (f)read
- ✓ (f)write
- ✓ (f)close
- ✓ getline

from the CSFML library

- ✓ sfRenderWindow_create
- ✓ sfRenderWindow_destroy
- ✓ sfRenderWindow_isOpen
- ✓ sfRenderWindow_close
- ✓ sfRenderWindow_pollEvent
- ✓ sfRenderWindow_setFramerateLimit
- ✓ sfRenderWindow_clear
- ✓ sfRenderWindow_drawSprite
- ✓ sfRenderWindow_draw*Shape
- ✓ sfRenderWindow_drawText
- ✓ sfRenderWindow_display
- ✓ sfMusic_*
- ✓ sfSprite_create
- ✓ sfSprite_destroy
- ✓ sfSprite_setTexture
- ✓ sfSprite_setPosition
- ✓ sfSprite_setRotation
- ✓ sfSprite_move
- ✓ sfSprite_rotate
- ✓ sfTexture_create
- ✓ sfTexture_createFromFile
- ✓ sfTexture_destroy
- ✓ sfTexture_updateFromPixels
- ✓ sfText_*
- ✓ sfView_*
- ✓ sfShape_* (except _getLocalBounds and _getGlobalBounds)
- ✓ sfConvexShape_* (except _getLocalBounds and _getGlobalBounds)
- ✓ sfRectangleShape_* (except _getLocalBounds and _getGlobalBounds)
- ✓ sfCircleShape_* (except _getLocalBounds and _getGlobalBounds)
- ✓ sfTransform_*
- ✓ sfClock_*

from the math library
All functions

Any unspecified functions are de facto banned.

v 2.6