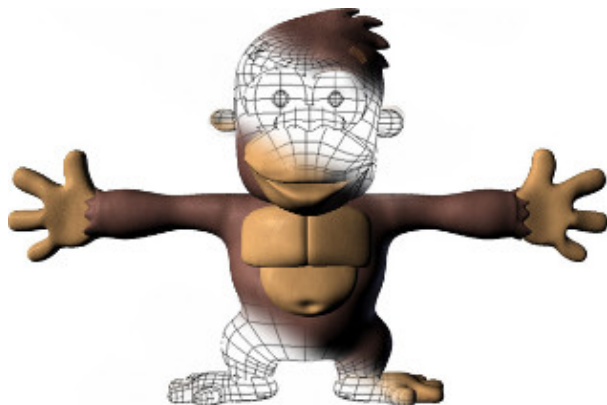


# B1 - C Graphical Programming

B-MUL-100

## My Screensaver

Graphical Effects Renderer





# My Screensaver

binary name: my\_screensaver  
group size: 1  
repository name: my\_screensaver\_\$ACADEMIC\_YEAR  
repository rights: ramassage-tek  
language: C  
compilation: via Makefile, including re, clean and fclean rules



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- 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 graphical effects and animations.

Rendering techniques and functions will be needed to develop complex and efficient animations.

They might then be reused in other graphical programs like video games.

Here is the requirement list:

## MUST

- The program **must** take exactly one parameter.
- The program **must** accept any integer as parameter. It corresponds to the ID of the animation to display.
- The program **must** implement as many **distinct** animations as possible.
- The program **must** accept the “-h” option as parameter. The program **must** then display the usage of the program.
- The animation ID **must** be between 1 and MAX\_ID, equal to the number of implemented animations.
- The MAX\_ID value **must** be specified in the usage.



A circle moving from left to right and another going from right to left are not two distinct animations.

## SHOULD

- The program **should** accept the “-d” option. Then, it displays the ID followed by a description of all the animations.
- Your window **should** stick between 800x600 pixels and 1920x1080 pixels.
- Your window **should** have a fixed frame rate such that it can be computed without lagging.
- Possible user interactions **should** be explicitly explained in the usage.

## COULD

- The program **could** implement user interactions to modify the ongoing animations.

## WOULD

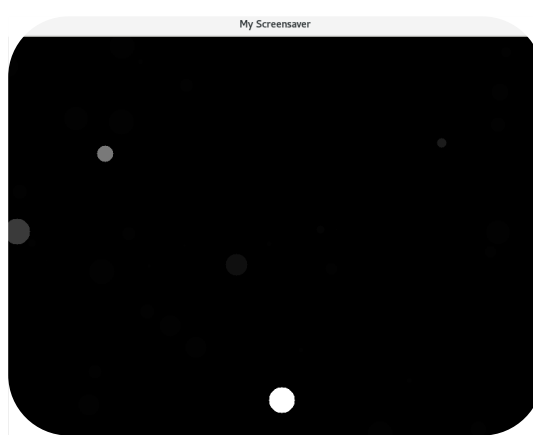
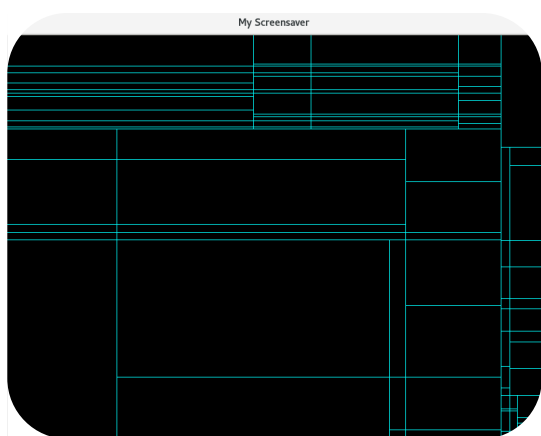
- The program **would** be really used as a Xscreensaver.

## EFFECTS

---

You are free to choose the animations your program will implement.

A binary is provided as an example to show you some animations that can be implemented. You can add more, mix them up and modify them at will.





## USAGE

```
Terminal
~/B-MUL-100> ./my_screensaver && echo $?
./my_screensaver: bad arguments: 0 given but 1 is required
retry with -h
84
```

```
Terminal
~/B-MUL-100> ./my_screensaver -h
animation rendering in a CSFML window.

USAGE
./my_screensaver [OPTIONS] animation_id
  animation_id    ID of the animation to process (between 1 and 20).

OPTIONS
-d                print the description of all the animations and quit.
-h                print the usage and quit.

USER INTERACTIONS
LEFT_ARROW       switch to the previous animation.
RIGHT_ARROW      switch to the next animation.
```

```
Terminal
~/B-MUL-100> ./my_screensaver -d
1: horizontal lines crossing the window at random height.
2: circles moving in the window and bouncing when touches a border.
3: circles appearing and fading out one after the other.
...
20: starry sky animation (stars going in and out and whose speed depends on their size)
```



## AUTHORIZED FUNCTIONS

---

Here is the full list of authorized functions.

### from the C library

- write
- malloc
- free
- memset
- rand
- srand

### from the SFML library

- sfRenderWindow\_create
- sfRenderWindow\_destroy
- sfRenderWindow\_isOpen
- sfRenderWindow\_close
- sfRenderWindow\_pollEvent
- sfRenderWindow\_setFramerateLimit
- sfRenderWindow\_clear
- sfRenderWindow\_drawSprite
- sfRenderWindow\_display
- sfSprite\_create
- sfSprite\_destroy
- sfSprite\_setTexture
- sfTexture\_create
- sfTexture\_destroy
- sfTexture\_updateFromPixels
- sfColor\_fromInteger
- sfColor\_toInteger
- sfClock\_create
- sfClock\_copy
- sfClock\_destroy
- sfClock\_getElapsedTime
- sfClock\_restart

### from the math library

All functions



Any unspecified functions are de facto banned, except for bonus features.