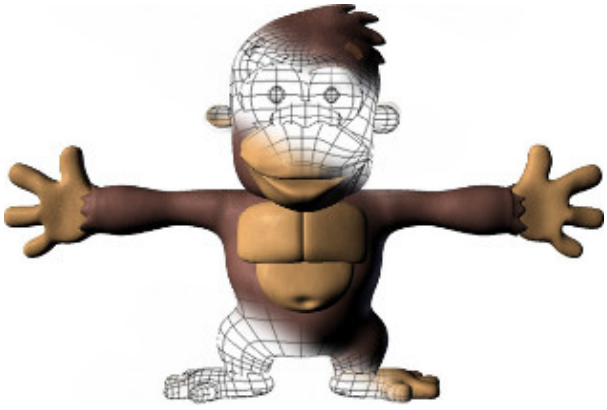
B1 - C Graphical Programming

B-MUL-100

My Screensaver

Graphical Effects Renderer



1.7

{EPITECH.}

My Screensaver

binary name:	my_screensaver
repository name:	MUL_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,...).
- 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 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.

The effects can be:

- Moving or scaling entities
- Bouncing, fading, and speed modification
- Or even more complex like noise or blur

The entities can be:

- Lines or any other 2D shapes perfect (e.g. circles) or not.
- Plain or bordered entities
- Different size and colors

Mixed up entities and effects can produce better animations.

Requirements

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*.

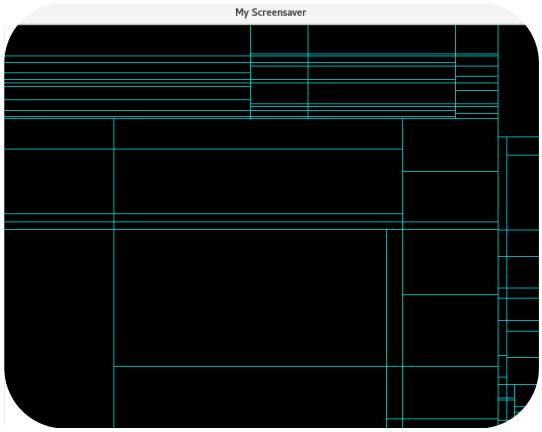
COULD

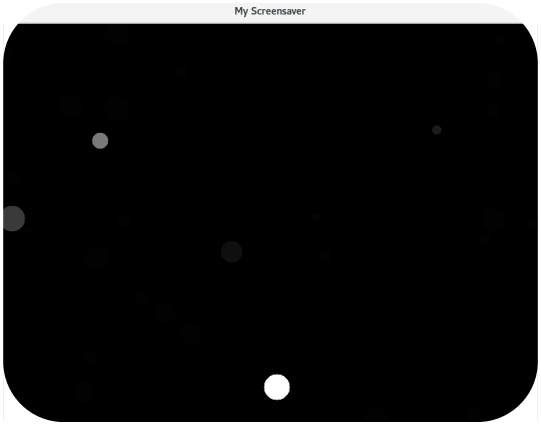
- The program **could** implement user interactions to modify the ongoing animations.
- The program **could** add texture mapping to fill shapes.
- The program **could** apply transformations upon textures.

Conception

These effects will be produced using a framebuffer. It means that no premade image will be used. You must modify the pixels to get the expected effects.

In C Graphical Programming, coding style must be thought from the very beginning. Otherwise, you might end up with dirty spagetthi code.


My Screensaver


My Screensaver

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

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

from the CSFML 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.