

## Setting Up miniLibX

updated: 04/03/2021

During the Spark Session, we'll be writing a little program that can open windows, draw pixels, and maybe even move pixels!

Before we can do that, we need to set up the miniLibX library and **link** the compiled library with our source code.

1. Download the minilibx library into the **root directory**.
  - For macOS: from intra, whichever version (OpenGL/mms\_beta) that works with your system
  - For Linux: from the [42Paris repo](#)
  - **The next steps assume you've called the folders `mlx` or `mlx_linux`.**
2. Create a `main.c` in the root directory with the following content (remember to include the `mlx.h` header!):

```
int    main(void)
{
    void    *mlx_ptr;

    mlx_ptr = mlx_init();
    return (0);
}
```

This will help you check at the end if you're linking your `mlx` correctly to your source files.

3. Now let's create our own Makefile in the root directory, which will compile our `main.c` into a program (name it whatever you want).  
Having a Makefile makes our lives easier.  
Add the required rules - `$(NAME)`, `clean`, `fclean`, `re`, `all`.

Here's a [helpful guide on Makefiles](#) written by another student, Noah Loomans.

4. Compile the `mlx` library, so that you get a `libmlx.dylib` (if you're using the `mms_beta` version of `mlx`) or `libmlx.a` file (for Linux & OpenGL versions).  
For macOS `mms_beta` library: you'll need to move `libmlx.dylib` into the same directory as your build target (as it's a dynamic library).

**Tip: you could have your Makefile do all this too.** Hint: `make -C dir`

5. Using miniLibX requires that we link the necessary **internal API's**. Here's what you should add to your project Makefile:
  - Once again, **the following commands assume you've named your `mlx` folder `mlx` (for Mac) or `mlx_linux`**. Also, `OBJ` here refers to the **object files of your project**, e.g. `main.o`, not the `mlx` files.
  - For macOS: *(make sure the compilation command is on one line)*

```
$(NAME): $(OBJ)
    $(CC) $(OBJ) -lmlx -framework OpenGL -framework AppKit -o
$(NAME)
```

- For Linux: first run `sudo apt-get install gcc make xorg libxext-dev libbsd-dev` to install the required `xorg`, `libxext-dev`, and `libbsd-dev` dependencies

```
$(NAME): $(OBJ)
$(CC) $(OBJ) -lmlx_linux -lmlx -lXext -lX11 -lm -lz -o $(NAME)
```

6. Additional steps if you're doing this through **Windows Subsystem for Linux**: you need to install [Xming](#) first.

- Once Xming is installed, exit Xming and launch XLaunch. Choose the following options:  
Multiple windows -> Start no client -> Enable "No access control" -> Finish
- Then execute this command: `export DISPLAY=localhost:0.0`
- You can check if everything's working by running `sudo apt-get install x11-apps` and then executing `xeyes`.
- **Note:** XLaunch has to be active and the `export DISPLAY` command above must have been run before you can launch graphic programs using `miniLibX`.

7. Now try running `make` in your project directory. Does your program compile without errors? Your program itself won't do anything for now. If everything works, great! Now you're ready for the SparkSession and your project!