

42 Docs

[Libs](#) / [MiniLibX](#) / Sync

Sync

TABLE OF CONTENTS

- 1 [What is sync?](#)
- 2 [Using sync](#)
- 3 [Test your skills!](#)

What is sync?

As mentioned before, you could manage your own frame buffering with MLX, however this is fairly tedious and time consuming. Besides, we use more memory and our frames need to constantly be FULLY updated. This is not very efficient hence we need to avoid it at all costs.

From the 2020 MLX version, you will be able to synchronize your own frames, which should make the hacky multiple images for screen buffering no longer required.

Using sync

We have three defines that we need to understand first:

```
#define MLX_SYNC_IMAGE_WRITABLE    1
#define MLX_SYNC_WIN_FLUSH_CMD    2
#define MLX_SYNC_WIN_CMD_COMPLETED 3
```

```
int mlx_sync(int cmd, void *ptr);
```

`mlx_sync` ought to be called with the defined command codes. The first one, `MLX_SYNC_IMAGE_WRITABLE` will buffer all subsequential calls to an image (`ptr` is a pointer to the MLX image object). If you want to propagate changes, you will have to flush the window in which the image is present, using `MLX_SYNC_WIN_FLUSH_CMD` and the window you want to flush as a `ptr`.

Test your skills!

Now that you have a faint idea of what `mlx_sync` can do, grab your previous tiny circle-game that you made in [loops](#) and add `mlx_sync` to your rendering!