

---

## Super Template Tetris

Implementation of Tetris as a C++ template metaprogram. All game logic is implemented at compile time. The runtime program is only needed to print out the game world and serialize it for next play.

Post discussing implementation.

### Playing

Plays one step of the game every time the game is recompiled.

Start a new game:

```
1 $ ./reset.sh
```

Use compiler flags to control input:

```
1 $ clang++ main.cpp -std=c++1z -Wno-gnu-string-literal-operator-template  
-D COMMAND -o tetris ; ./tetris
```

Valid commands are:

- **LEFT** - Move piece left by one.
- **RIGHT** - Move piece right by one.
- **LROT** - Rotate piece left (counter clockwise).
- **RROT** - Rotate piece right (clockwise).
- **HARD** - Hard drop the current piece. This drops the piece all the way down and places it.
- **SOFT** - Drop the piece all the way down but don't place it.
- **DOWN** - Move the piece down by 4.
- **NONE** / anything else - Do nothing, advance game by one.

Pieces will continue to fall for all commands. Movement takes place before the piece is moved down. Pieces have a one step delay before being auto placed (that is, when a piece collides with a block, it but will only be placed on the next step if no further movement is performed).

### Compiling

- `-std=c++1z` since the program uses a few C++17 features.
- `-Wgnu-string-literal-operator-template` is used for generating compile time strings.

---

## **Example**

A few screenshots:







