# Figuring Stuff Out

EECS 211

Winter 2019

# Strategies

- Read the compiler output
- Ask for help *effectively*
- Read the documentation
- The scientific method
- Try the debugger

# Read the compiler output

Errors are there to help you.

```
/Users/tov/school/eecs211/web/lec/14figuring/
src/ui.cpp:76:9: error: no viable conversion
from 'ge211::geometry::Position' (aka
'Basic_position<int>') to 'int'
   int col_no = screen_to_board_(screen_posn);
```

# Don’t ignore warnings!

They’re there to help you to. This indicates a bug:

```
/Users/tov/school/eecs211/web/lec/14figuring/
src/model.cpp:21:1: warning: control may reach
end of non-void function [-Wreturn-type]
}
^
```

# Don’t ignore warnings!

They’re there to help you to. This indicates a bug:

```
/Users/tov/school/eecs211/web/lec/14figuring/
src/model.cpp:21:1: warning: control may reach
end of non-void function [-Wreturn-type]
}
^
```

So does this:

```
/Users/tov/school/eecs211/web/lec/14figuring/
src/ui.cpp:48:34: warning: expression result
unused [-Wunused-value]
    screen_to_board_(screen_pos).x;
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ^
```

# Ineffective help requests

- “It doesn’t work.”
- “I get an error.”

# Ineffective help requests

- “It doesn’t work.” — What did you expect and what happened?
- “I get an error.” — What did the error say? Do you understand it?

# Effective help requests…

…are similar to effective bug reports:

- What did you do?
- What did you expect to happen?
- What happened instead?
- What else have you tried?
- Can you construct a minimal example? Is it reproducible?
- What's the context? (OS, software versions, etc.)

# Read the docs

Two main places:

- GE211 documentation: `https://tov.github.io/ge211/`
- C++ library reference: `https://en.cppreference.com/w/`

# Let’s look up some stuff

- `std::string`
- `std::is_permutation`
- `ge211::Rectangle`
- `ge211::Text_sprite::reconfigure`

# The debugger

The debugger is the bug button is CLion. It lets you step through your program, print variables, set breakpoints, etc.

Let’s try it.

– Next: the Model–View–Controller pattern —