

Currents: Coding with Cinder

Week 1: Course Introduction / Introduction to Cinder

Instructors

Luobin Wang (luobin@newschool.edu)

Weili Shi (weili@newschool.edu)

Welcome to Currents: Coding with Cinder!

Here we take creative coding seriously.

Who are we?



Luobin Wang
Developer, Potion
MFA DT '16
<http://peterobbin.me>
luobin@newschool.edu



Weili Shi
Developer, Bluecadet
MFA DT '16
<http://shi-weili.com>
weili@newschool.edu

Who are you?

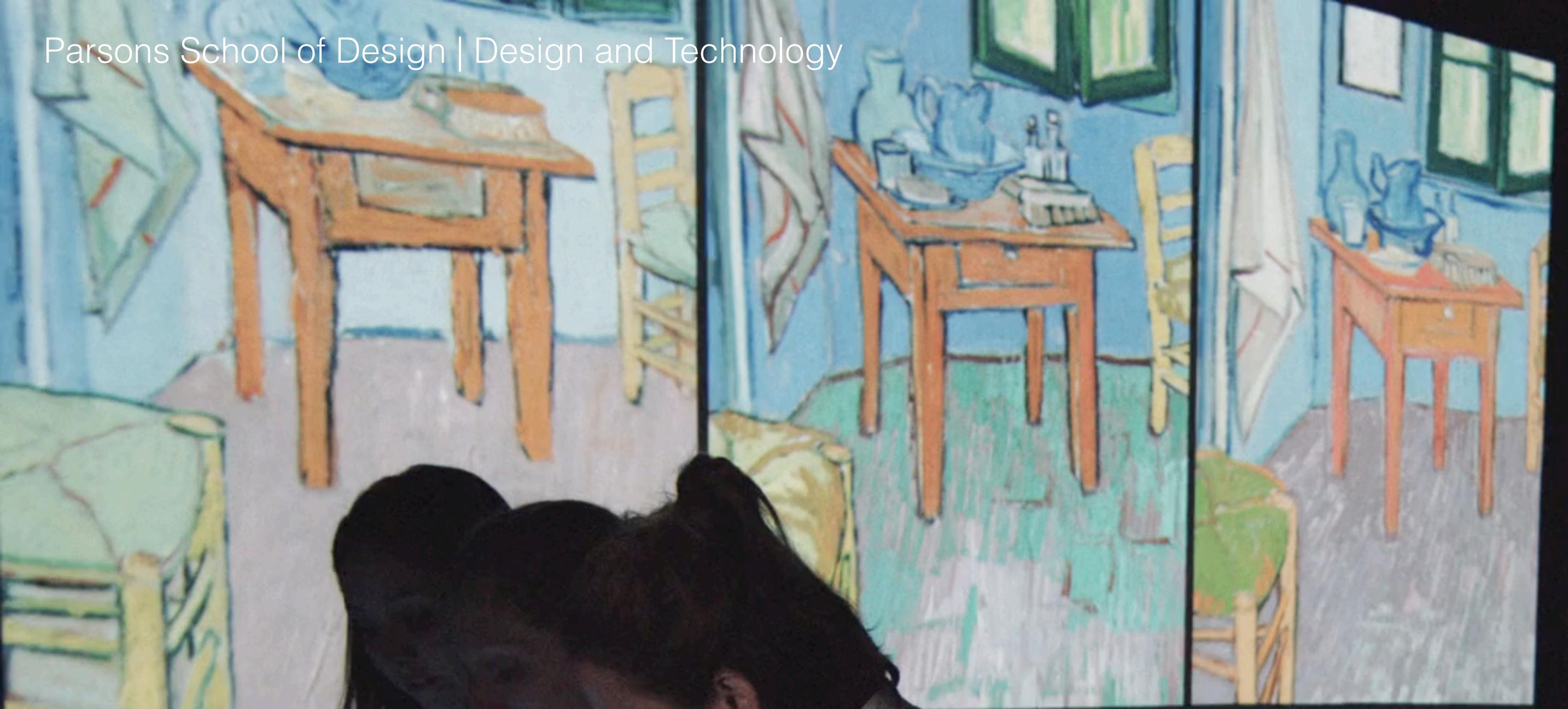
Tell us about your background, programming experience, and interested areas!

What is Cinder

Cinder is an open source creative coding library written in C++. It provides a toolset similar to openFrameworks, but exposes more infrastructures of the C++ programming language and the computer, which makes it more powerful and stable.

What can Cinder do?

Simply speaking, Cinder provides a framework for building interactive multimedia applications. It is a wrapper of OpenGL. It also provides a united cross-platform programming interface to functionalities like graphics, audio, video, computational geometry, and input/output.



Bluecadet Van Gogh's Bedrooms



Potion
Forest Friends



Robert Hodgin et al.
Taxi, Taxi!

Red Paper Heart The Good Deed Machine



The screenshot shows a web browser window displaying the libcinder.org/gallery page. The page features a header with the Cinder logo and navigation links for About, Gallery (which is highlighted in red), Learn, Docs, Forum, and Download. Below the header, a large section titled "Selected Cinder Projects" with the sub-instruction "Check out what Cinder users have made." includes a "SUBMIT A PROJECT" button. Three project cards are displayed: "Taxi, Taxi!" by Robert Hodgin & Jonathan Kim, Ryan Bartley, The Guild; "Cosmograph" by Eric Renaud-Houde; and "Dancing Robots" by Quarso, Thomas Sanchez Lengeling.

Taxi, Taxi!

by Robert Hodgin & Jonathan Kim, Ryan Bartley,
The Guild

Taxi, Taxi! is a digital installation featured on the 3-story display housed within Samsung 837, the new Samsung Experience Center located in

Cosmograph

by Eric Renaud-Houde

The Cosmograph is an audio-reactive cinder application that generates a visual language for spiritual jazz inspired by Sacred Geometry.

Dancing Robots

by Quarso, Thomas Sanchez Lengeling

The piece is a 5m x 3.2m sculpture formed by 68 steel triangles and weighted around 600kg. The sculpture is inspired by the assembling robotic arms from the Detroit's Chrysler plant.

More examples on <https://libcinder.org/gallery>

Why teach Cinder?

While Cinder is the creative coding framework of choice in many digital agencies, it is rarely taught in schools. We want to fill the gap, and prepare you for professional creative coding work.

Why learn Cinder?

Cinder encourages good programming/
software engineering practices, which will
empower you to accomplish comprehensive
programming goals.

This is an intermediate-level course.

We expect you to be comfortable with writing interactive applications using at least one creative coding platform (openFrameworks, Processing, etc.)

Learning to code *is* challenging,
and rewarding.

It is true to all of us.

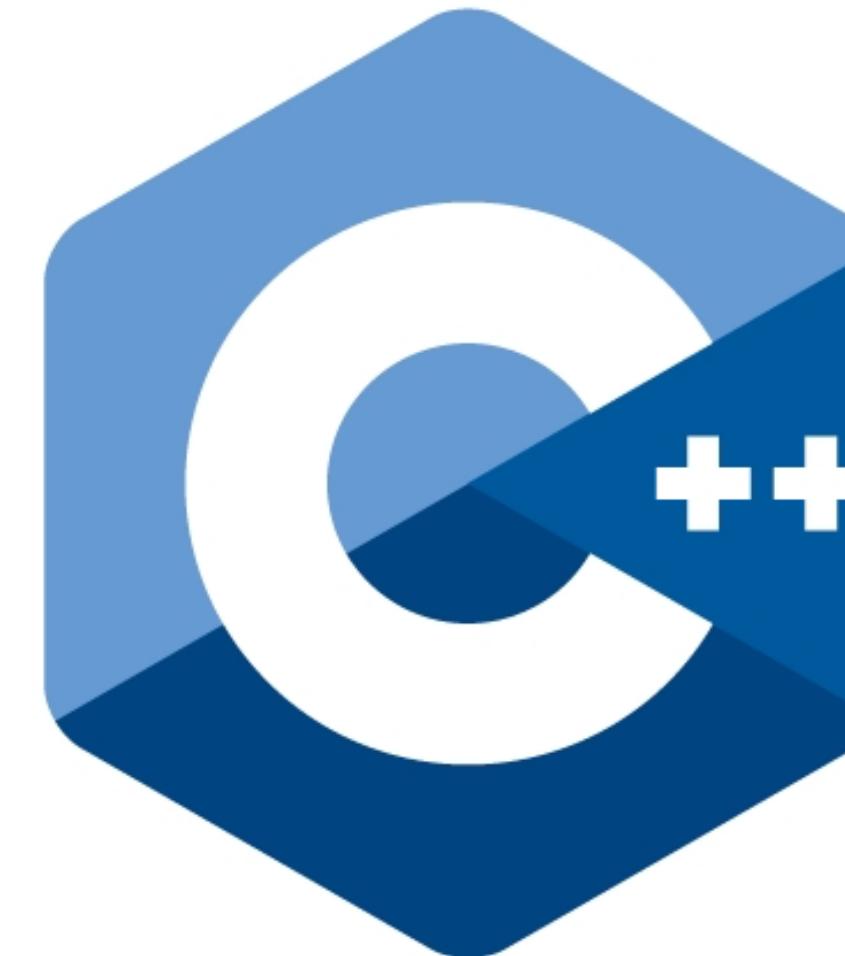
If you feel you may not be fully prepared, but are still eager to follow
the course, talk to us after class!

We put even efforts into two things:

1. **Luobin** Introduces Cinder functionalities.
2. In support of Cinder programming, **Weili** provides related knowledge of C++, computer science, and software engineering.

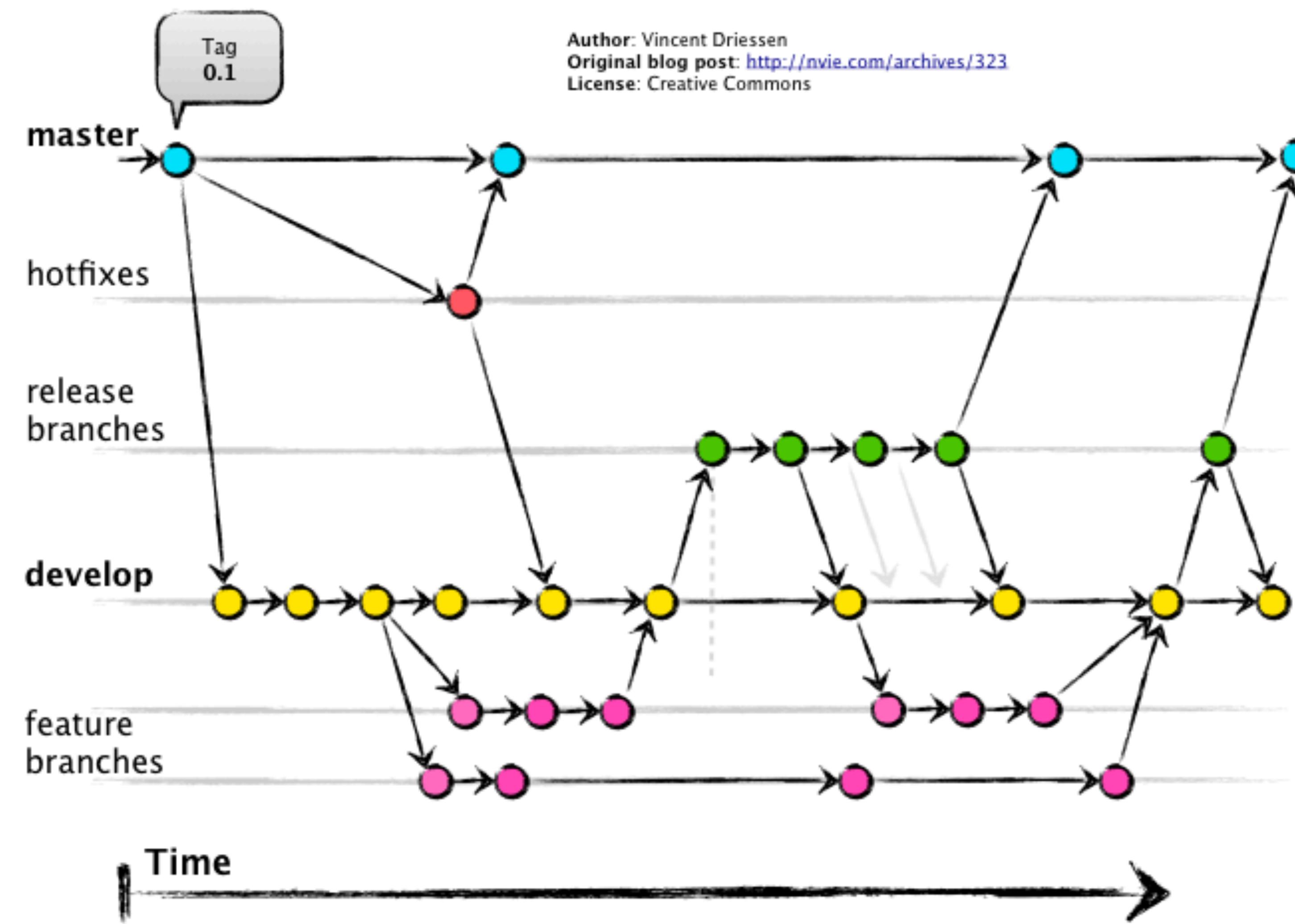
The screenshot shows the official Cinder documentation website at libcinder.org. The page has a dark background with red and white text. At the top, there's a navigation bar with links to "Main Page", "Reference", and "Guides". A search bar with a placeholder "Search documentation..." and a "SEARCH" button is also at the top. On the left, there's a sidebar with a "Latest Release (v0.9.0)" button and a "Master (v0.9.1.dev)" link. Below that is a section titled "CINDER GUIDES" with links to "Setup Cinder on OS X", "Setup Cinder on Windows", "Cinder + Git", "TinderBox", "CinderBlocks", and "OpenGL in Cinder", along with a "All Guides" link. The main content area features a large heading "Cinder Documentation". Below it, a paragraph of text provides guidance for new users, mentioning the "OpenGL in Cinder" guide, the "Categories list", and the samples. To the right of the text are ten circular icons representing different categories: GEOMETRY (triangle), MATH (function f(x)), SYSTEM (gear), PLATFORM (laptop), OPENGL (OpenGL logo), 2D GRAPHICS (square with triangle), AUDIO (waveform), VIDEO (camera), IMAGES (image icon), and APP (mountain). Each category has its name written below its respective icon.

We are going to cover a good part of these topics.
Note that we'll not focus on 3D in this course.



C++ Programming Language

- C++ fundamentals recap
- Object oriented programming
- Memory management
- Building / debugging / libraries



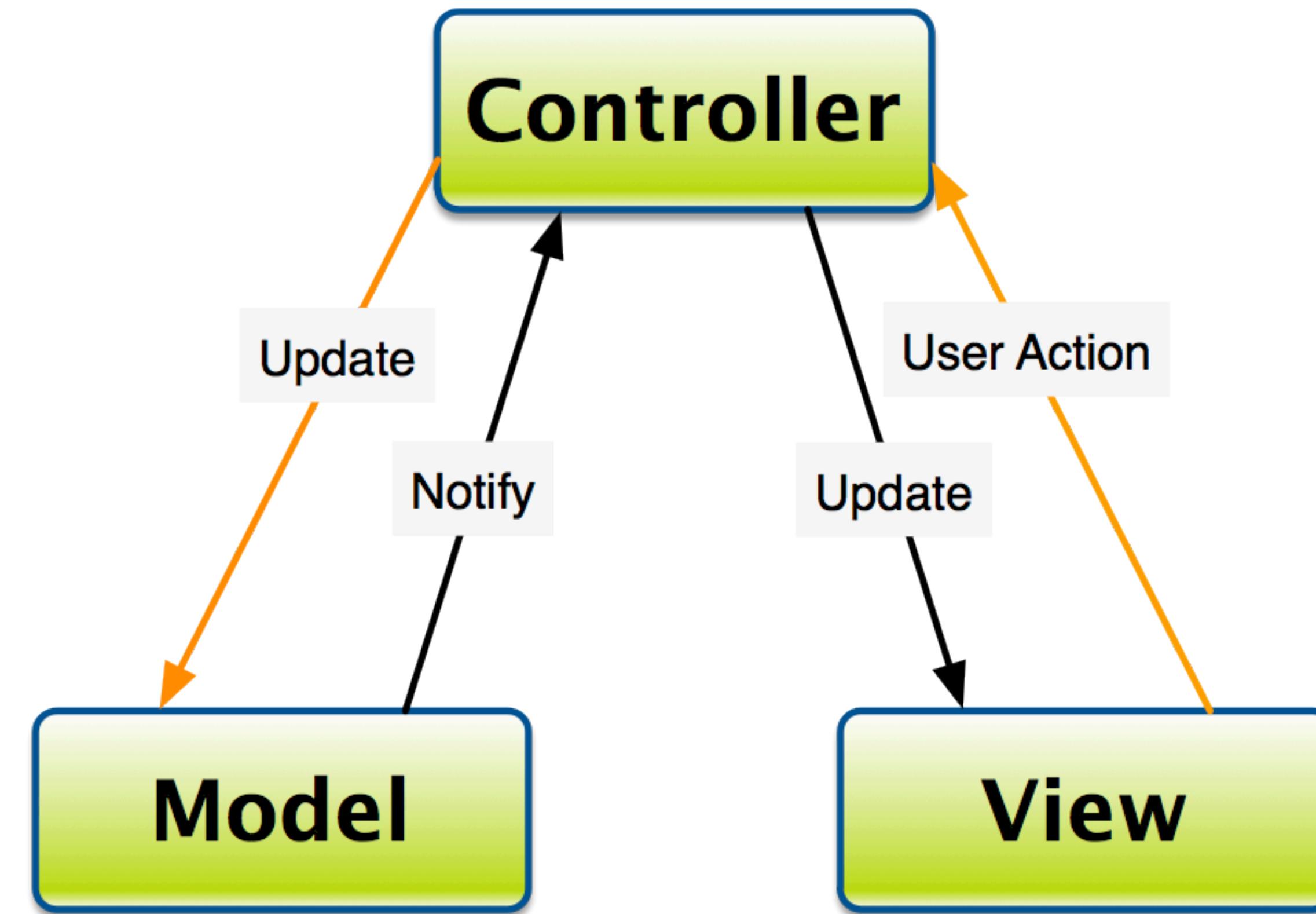
Version control and collaboration using git, GitHub, and git-flow

The screenshot shows a ZenHub project board for the repository 'routific'. The board is organized into six columns representing different stages of the development process:

- New Issues (3)**: Contains 12 issues, including routific #11 (Bug in Maps - Fix styling), routific #16 (Feature request: Instant driver feedback), and routific #17 (Better handling of route dispatch).
- Icebox (5)**: Contains 32 issues, including routific #8 (Mobile Screen View) and routific #7 (Beta Testing Feedback).
- Backlog (3)**: Contains 6 issues, including routific #27 (Real-time schedule updates) and routific #1 (Loading Testing API).
- In Progress (4)**: Contains 10.5 issues, including routific #29 (Improve API documentation on new website) and routific #10 (Mobile Interface).
- Testing (4)**: Contains 16 issues, including routific #12 (First Demo) and routific #19 (Miles / Km Popup).
- QA (2)**: Contains 5 issues, including routific #13 (Promo Video) and routific #32 (Create vehicle-tracking).

Each issue card includes a thumbnail, the issue number, a title, and labels such as Bug, Engineering, Help Wanted, Duplicate, etc.

Project management using ZenHub or Trello



MVC design pattern and poScene

Course schedule

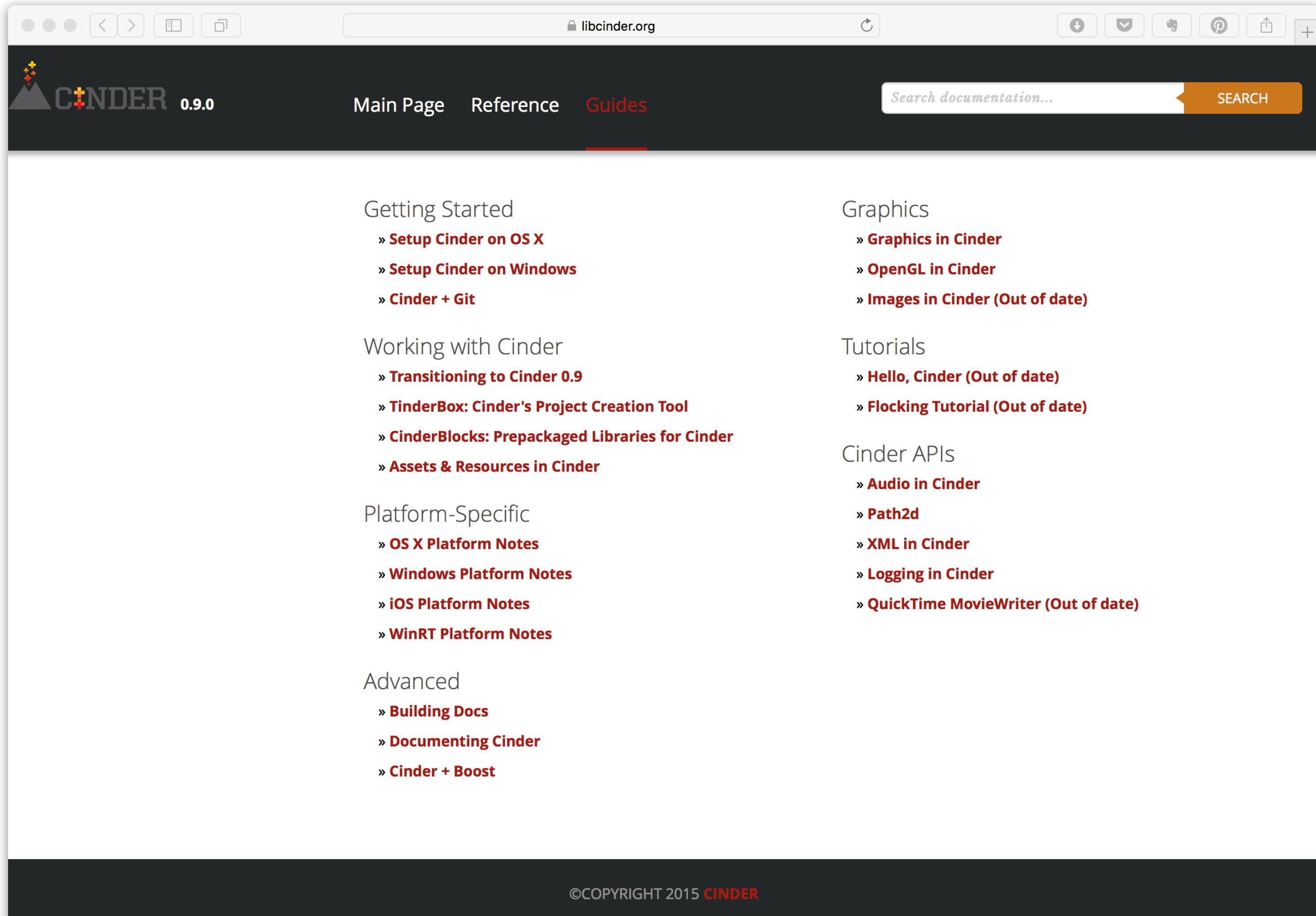
WEEK 1	01/25	Course introduction / introduction to Cinder
WEEK 2	02/01	C++ fundamentals recap / texture and shader
WEEK 3	02/08	Object oriented programming / particle system
WEEK 4	02/15	Memory management / particle system revisited
WEEK 5	02/22	Animation with timeline / version control with GitHub
WEEK 6	03/01	Video & audio playback / input & output
WEEK 7	03/08	Building / debugging / libraries / Cinder blocks
WEEK 8	03/15	Guest lecture 1 (Ritesh Lala from Potion)
WEEK 9	03/22	<Spring break / no class>
WEEK 10	03/29	poScene scene graph / MVC design pattern / final project kick-off
WEEK 11	04/05	poScene components / touch event handling
WEEK 12	04/12	poScene components continued / software architecture & project management
WEEK 13	04/19	Final project workshop 1
WEEK 14	04/26	Guest lecture 2 (Ben Bojko from Bluecadet)
WEEK 15	05/03	Final project workshop 2
WEEK 16	05/10	Final project presentation

Assignments & Grading

1. Attendance/participation 20%
2. Weekly assignments (individual) 40%
3. Final project (group) 40%

Recommended Readings

To name a few. For the curious minds.

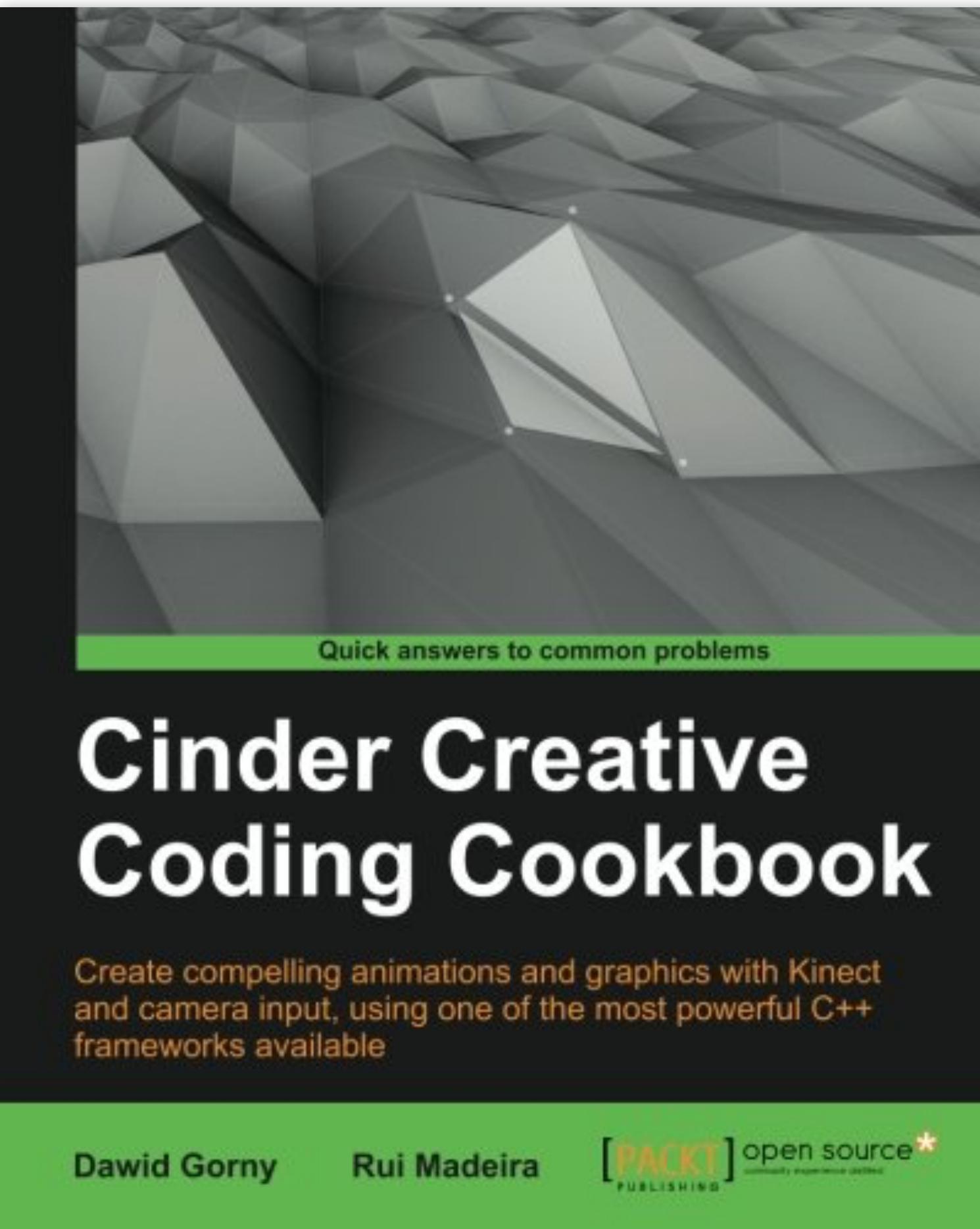


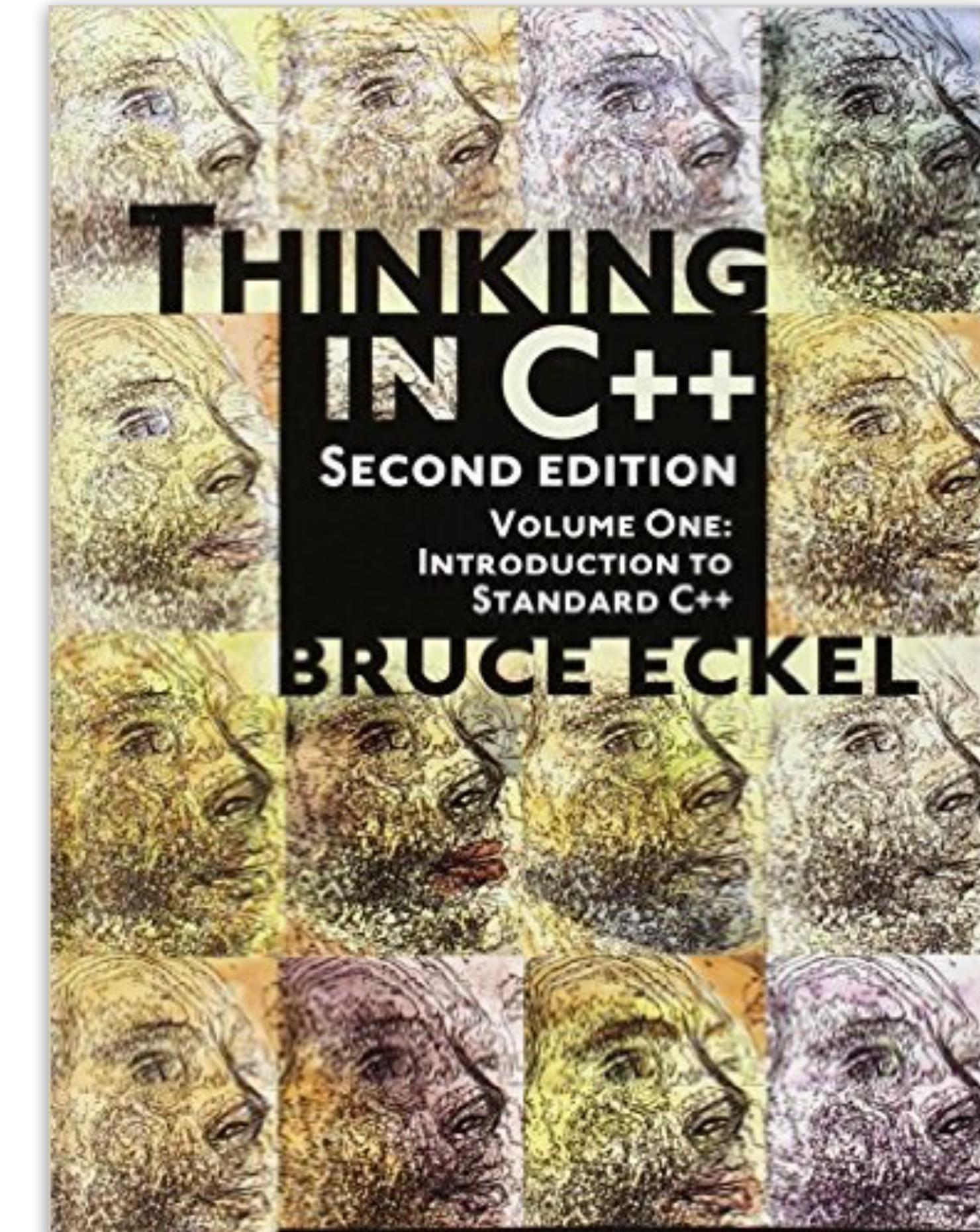
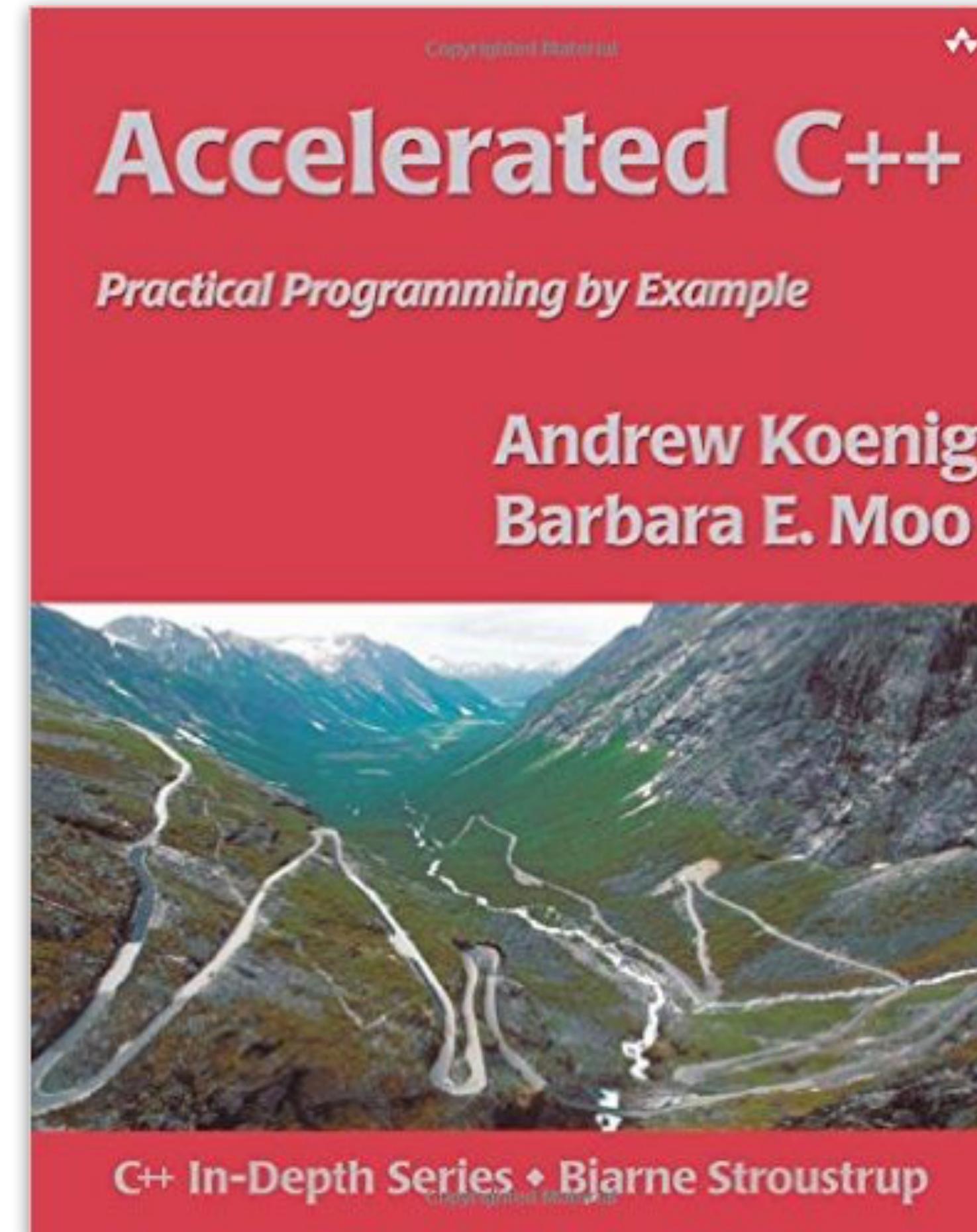
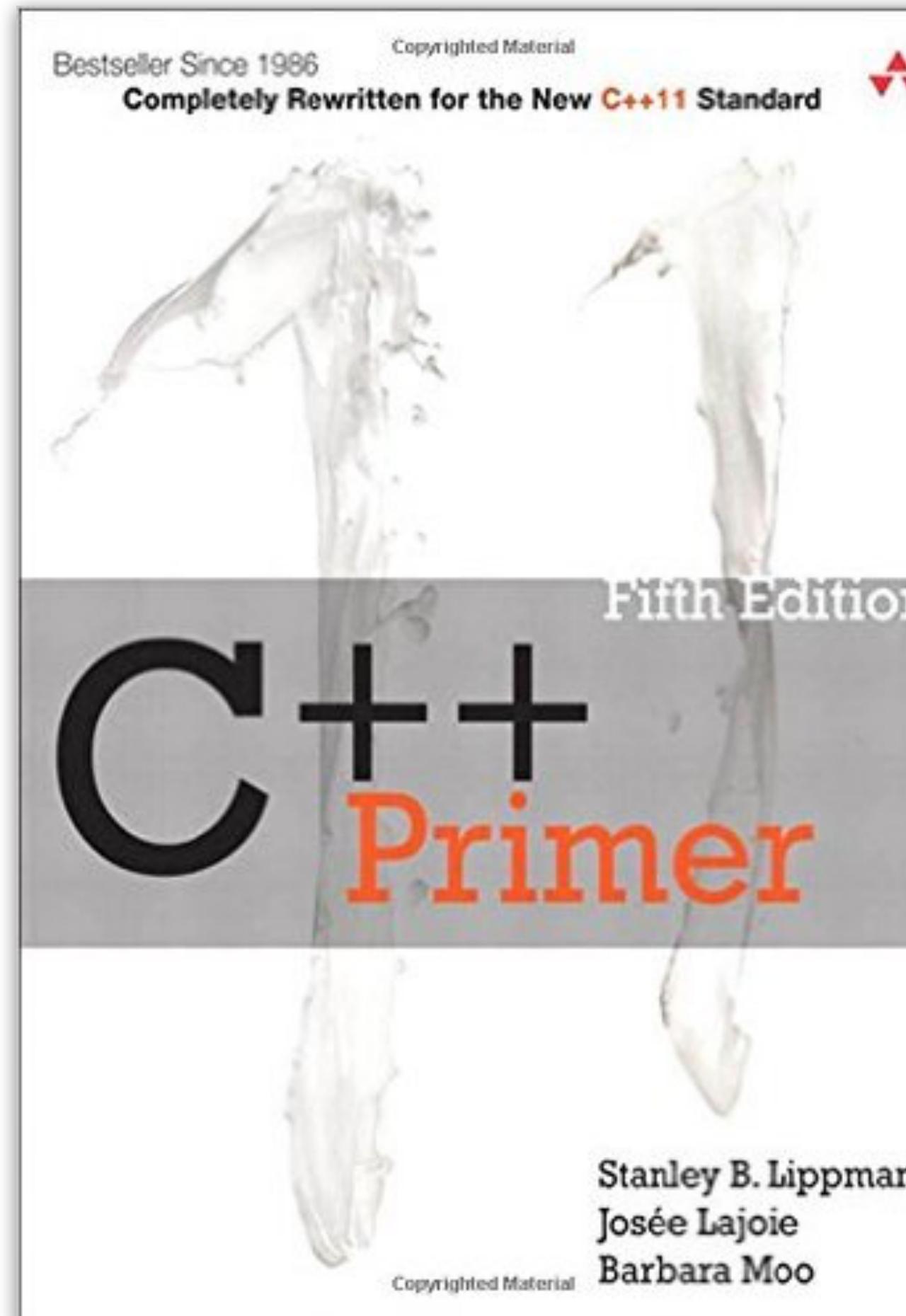
The screenshot shows a web browser window displaying the official Cinder documentation at libcinder.org. The page is titled "CINDER 0.9.0". The navigation menu includes "Main Page", "Reference", and "Guides", with "Guides" being the active tab. A search bar is located at the top right. The main content area is divided into several sections:

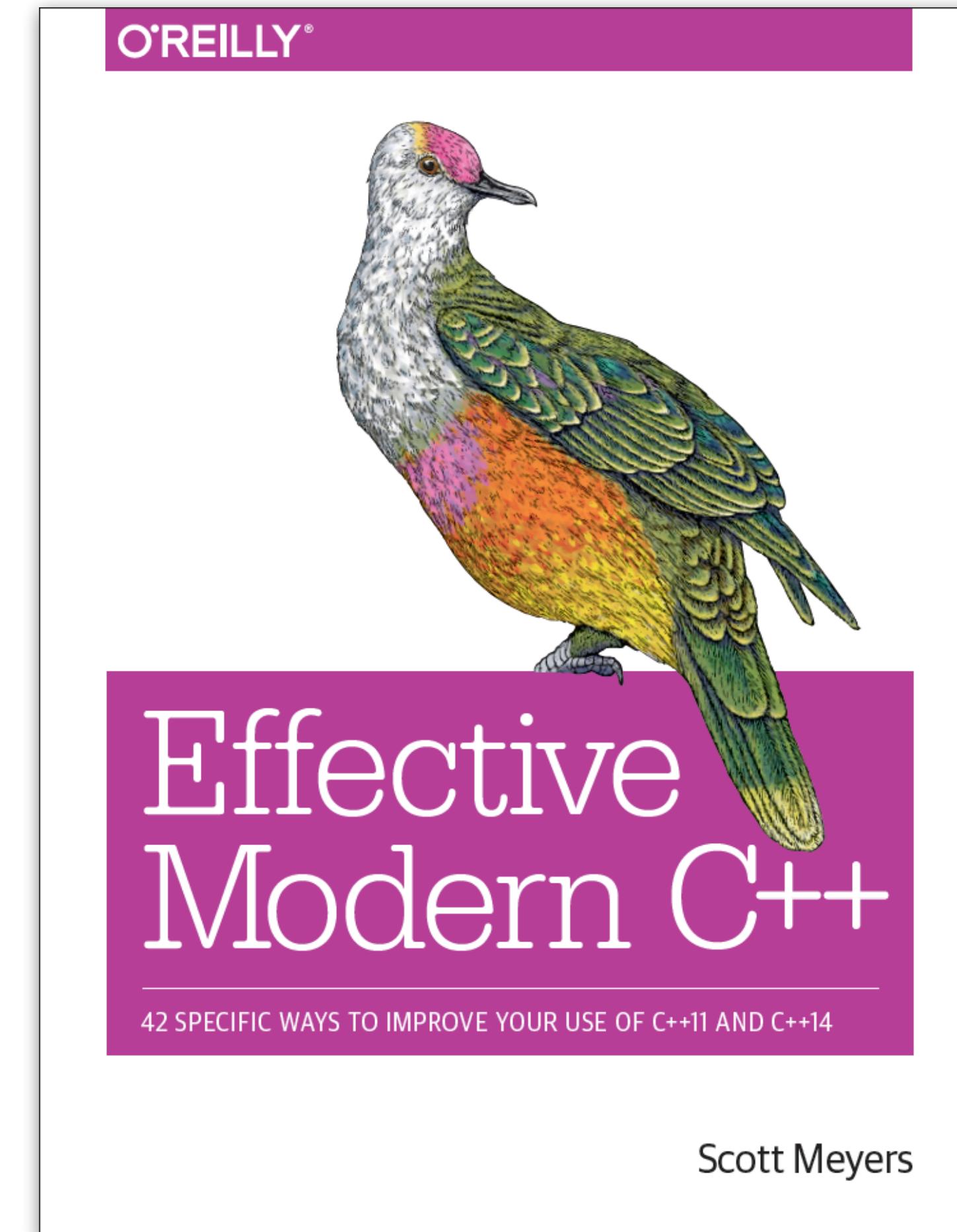
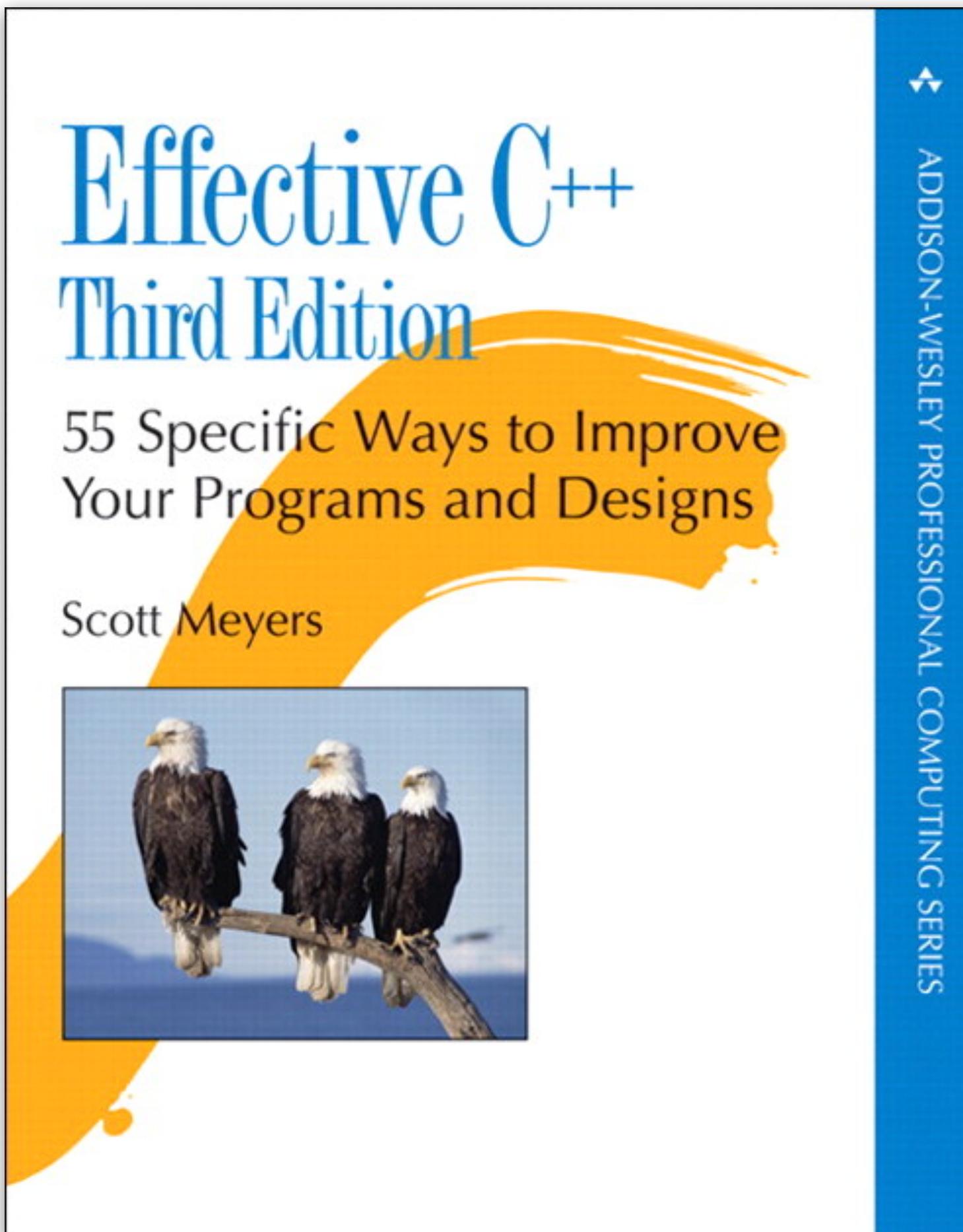
- Getting Started**
 - » [Setup Cinder on OS X](#)
 - » [Setup Cinder on Windows](#)
 - » [Cinder + Git](#)
- Working with Cinder**
 - » [Transitioning to Cinder 0.9](#)
 - » [TinderBox: Cinder's Project Creation Tool](#)
 - » [CinderBlocks: Prepackaged Libraries for Cinder](#)
 - » [Assets & Resources in Cinder](#)
- Platform-Specific**
 - » [OS X Platform Notes](#)
 - » [Windows Platform Notes](#)
 - » [iOS Platform Notes](#)
 - » [WinRT Platform Notes](#)
- Advanced**
 - » [Building Docs](#)
 - » [Documenting Cinder](#)
 - » [Cinder + Boost](#)
- Graphics**
 - » [Graphics in Cinder](#)
 - » [OpenGL in Cinder](#)
 - » [Images in Cinder \(Out of date\)](#)
- Tutorials**
 - » [Hello, Cinder \(Out of date\)](#)
 - » [Flocking Tutorial \(Out of date\)](#)
- Cinder APIs**
 - » [Audio in Cinder](#)
 - » [Path2d](#)
 - » [XML in Cinder](#)
 - » [Logging in Cinder](#)
 - » [QuickTime MovieWriter \(Out of date\)](#)

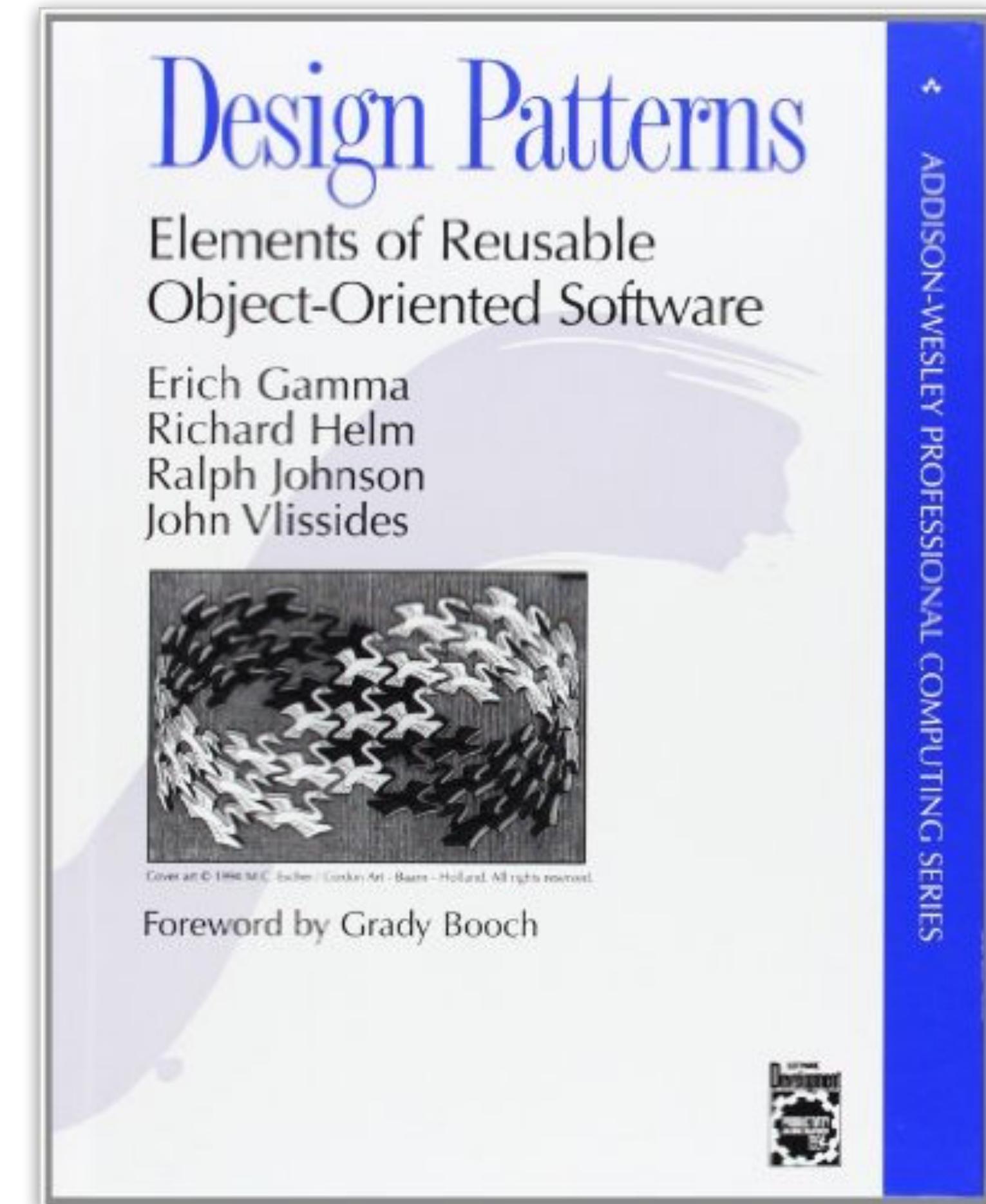
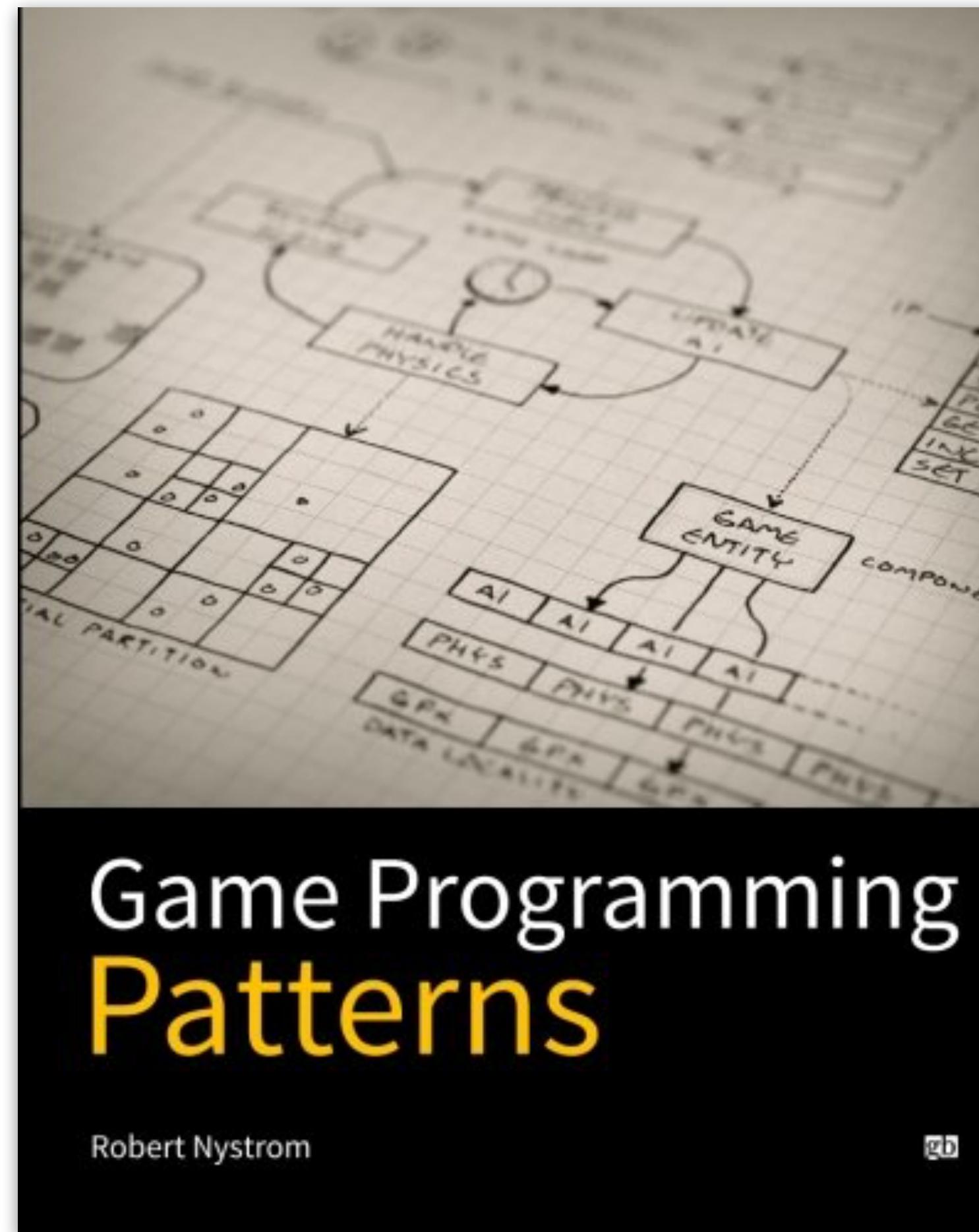
At the bottom of the page, a dark footer bar contains the text "©COPYRIGHT 2015 CINDER".

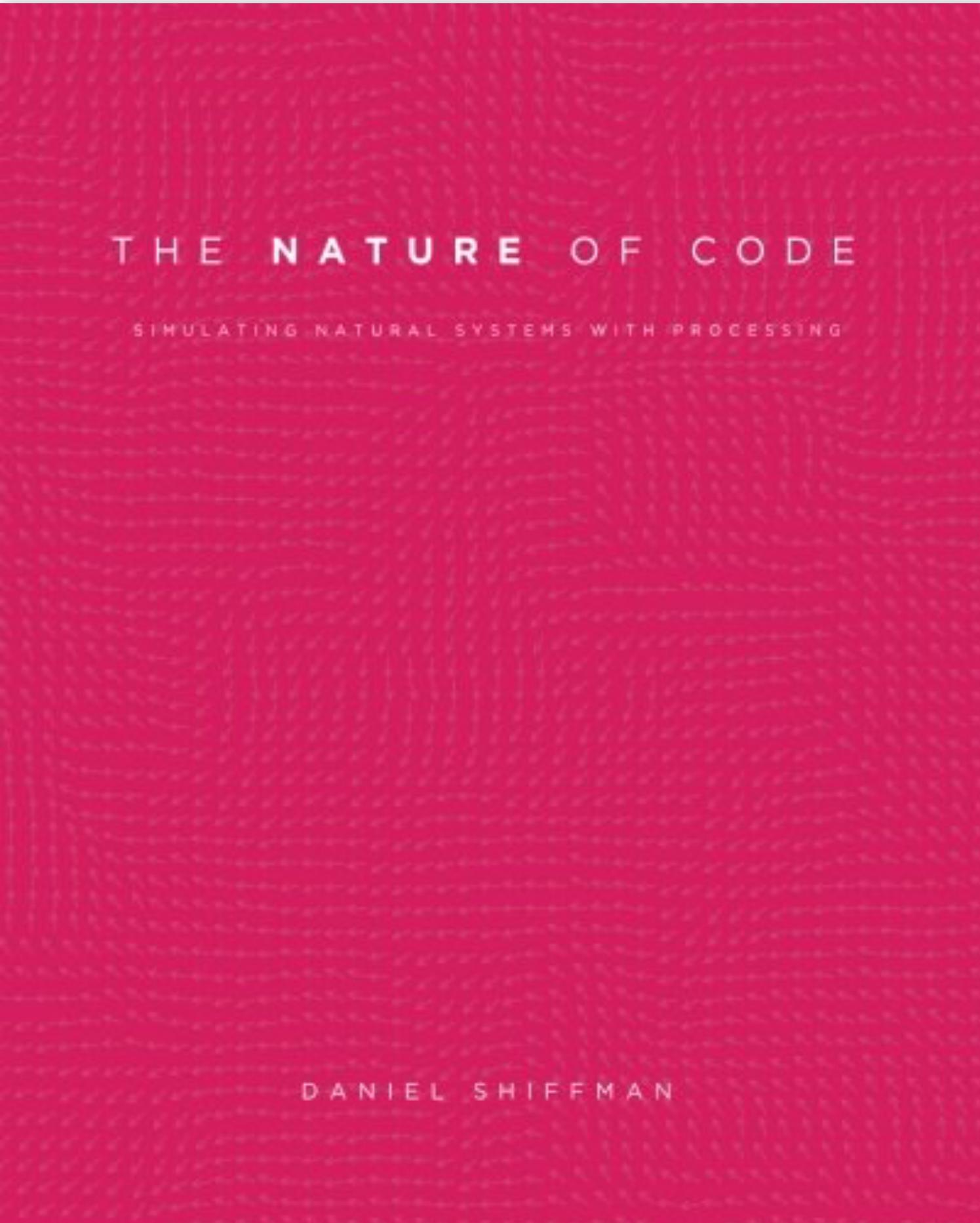
Official Cinder guides



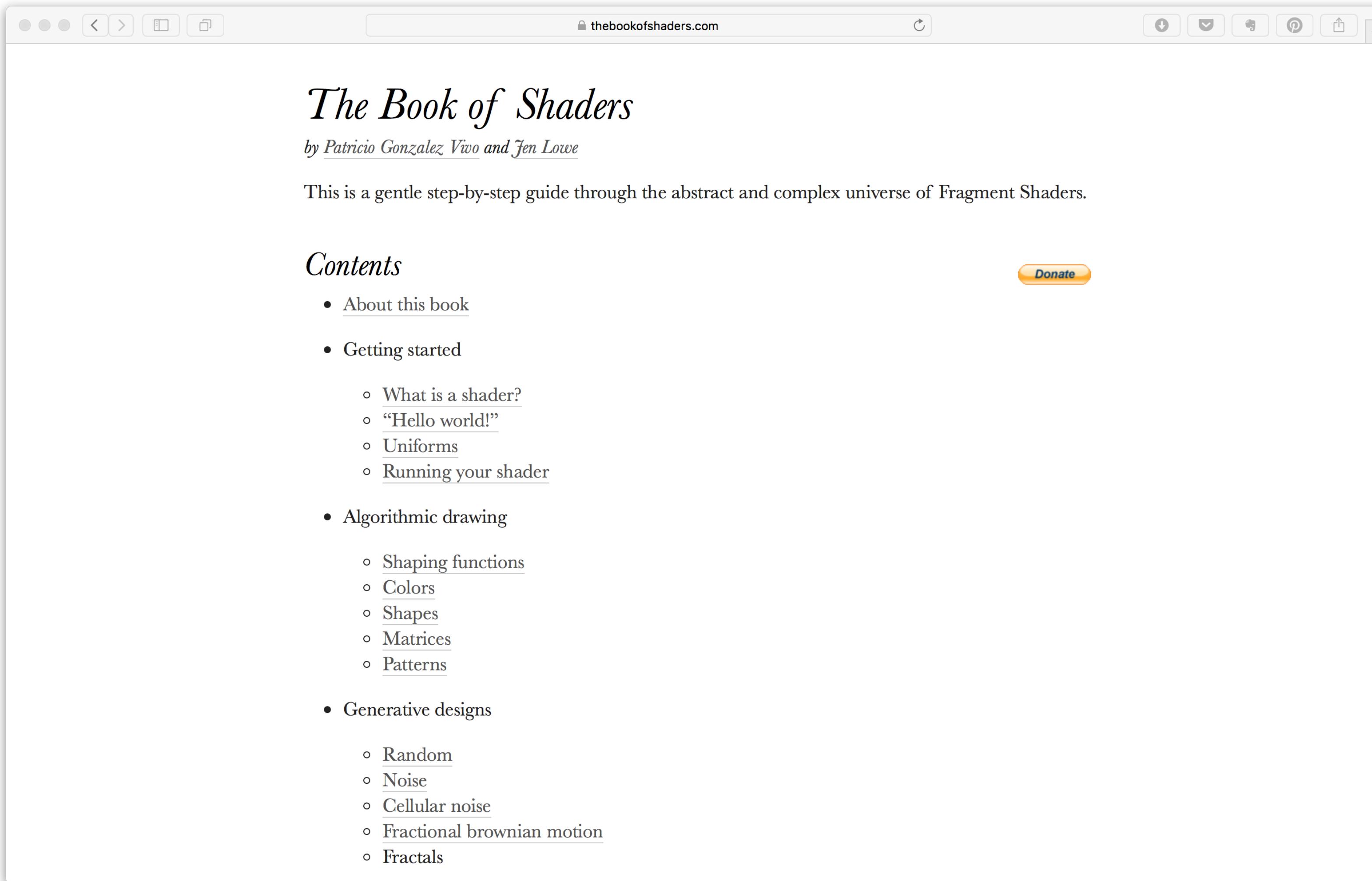








The Nature of Code

A screenshot of a web browser displaying the homepage of "The Book of Shaders". The title "The Book of Shaders" is at the top, followed by "by Patricio Gonzalez Vivo and Jen Lowe". A short description states: "This is a gentle step-by-step guide through the abstract and complex universe of Fragment Shaders." Below this is a "Contents" section with a hierarchical menu:

- [About this book](#)
- Getting started
 - [What is a shader?](#)
 - [“Hello world!”](#)
 - [Uniforms](#)
 - [Running your shader](#)
- Algorithmic drawing
 - [Shaping functions](#)
 - [Colors](#)
 - [Shapes](#)
 - [Matrices](#)
 - [Patterns](#)
- Generative designs
 - [Random](#)
 - [Noise](#)
 - [Cellular noise](#)
 - [Fractional brownian motion](#)
 - [Fractals](#)

A "Donate" button is located in the top right corner of the page content area.

The Book of Shaders



Join our Slack channel for course communications!

<https://currents-cinder.slack.com>

Survey

Are you a Mac user or a Windows user?

Getting started

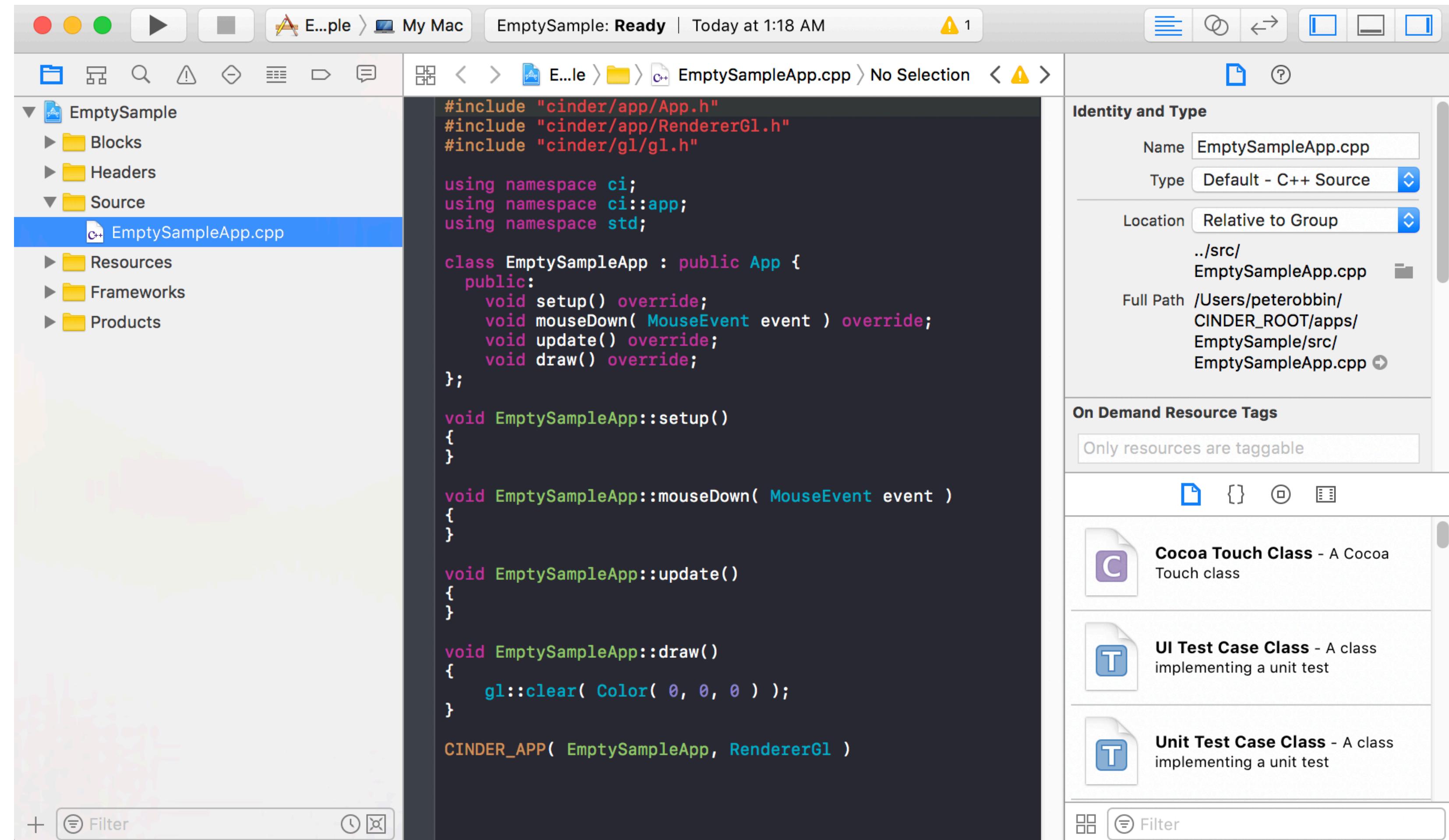
- 1. Get your favorite IDE installed (Xcode or Visual Studio)**
- 2. Download Latest Cinder from <https://libcinder.org>**
- 3. Build Cinder**

Using TinderBox



- 1. Specify your project name (don't just type in random names)**
- 2. Specify the location of where you want to store your project**
- 3. Select your target IDE to build project files (you can do both)**
- 4. Only create a git repository if you know what you're doing**
- 5. We are not using any Cinder Blocks for now**

Your first Cinder project

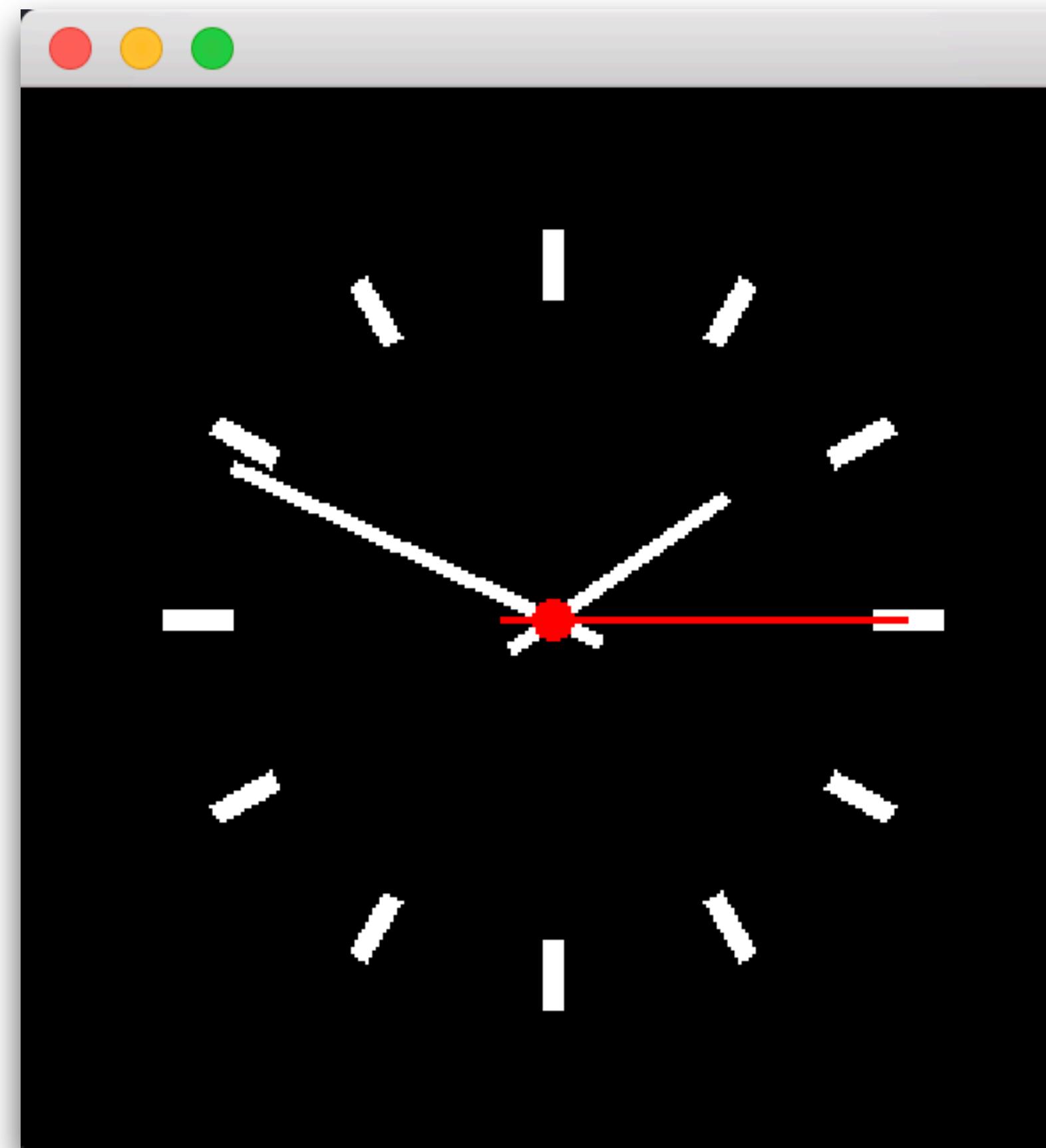


The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "EmptySample". The "EmptySampleApp.cpp" file is selected.
- Editor:** Displays the code for "EmptySampleApp.cpp". The code defines a class "EmptySampleApp" that inherits from "App". It overrides methods for setup, mouseDown, update, and draw. The draw method uses OpenGL to clear the screen. A macro "CINDER_APP" is used to register the application.
- Identity and Type:** Shows the file's properties:
 - Name: EmptySampleApp.cpp
 - Type: Default - C++ Source
 - Location: Relative to Group
 - Full Path: /Users/peterobbin/CINDER_ROOT/apps/EmptySample/src/EmptySampleApp.cpp
- On Demand Resource Tags:** A section indicating that only resources are taggable.
- Quick Help:** Shows three options: Cocoa Touch Class, UI Test Case Class, and Unit Test Case Class.

Paul Houx's analog clock example

<https://github.com/paulhoux/Cinder-Samples/tree/master/AnalogClock>



Namespaces

Namespaces provide a method for preventing name conflicts in large projects. Symbols declared inside a namespace block are placed in a named **scope** that prevents them from being mistaken for identically-named symbols in other scopes.

Using Namespace

Using-directive (effective until the end of the scope): **using namespace std;**

Or

Specify the namespace when using it: **std::string;**



ci::gl::

**A group of functions under cinder's openGL namespace,
which uses the power of your graphic card.**

ci:: or cinder::

Cinder namespace. Check Cinder documentation for namespaces and functions: <https://libcinder.org/docs/>

std::

C++ standard namespace. All the entities (variables, types, constants, and functions) of the standard C++ library are declared within the std namespace.

`ci::vec2(x, y)` is also `glm::vec2(x, y)`

Store values from a vector, or just two numbers you want to put together. GLM stands for OpenGL Mathematics.

`gl::clear()`

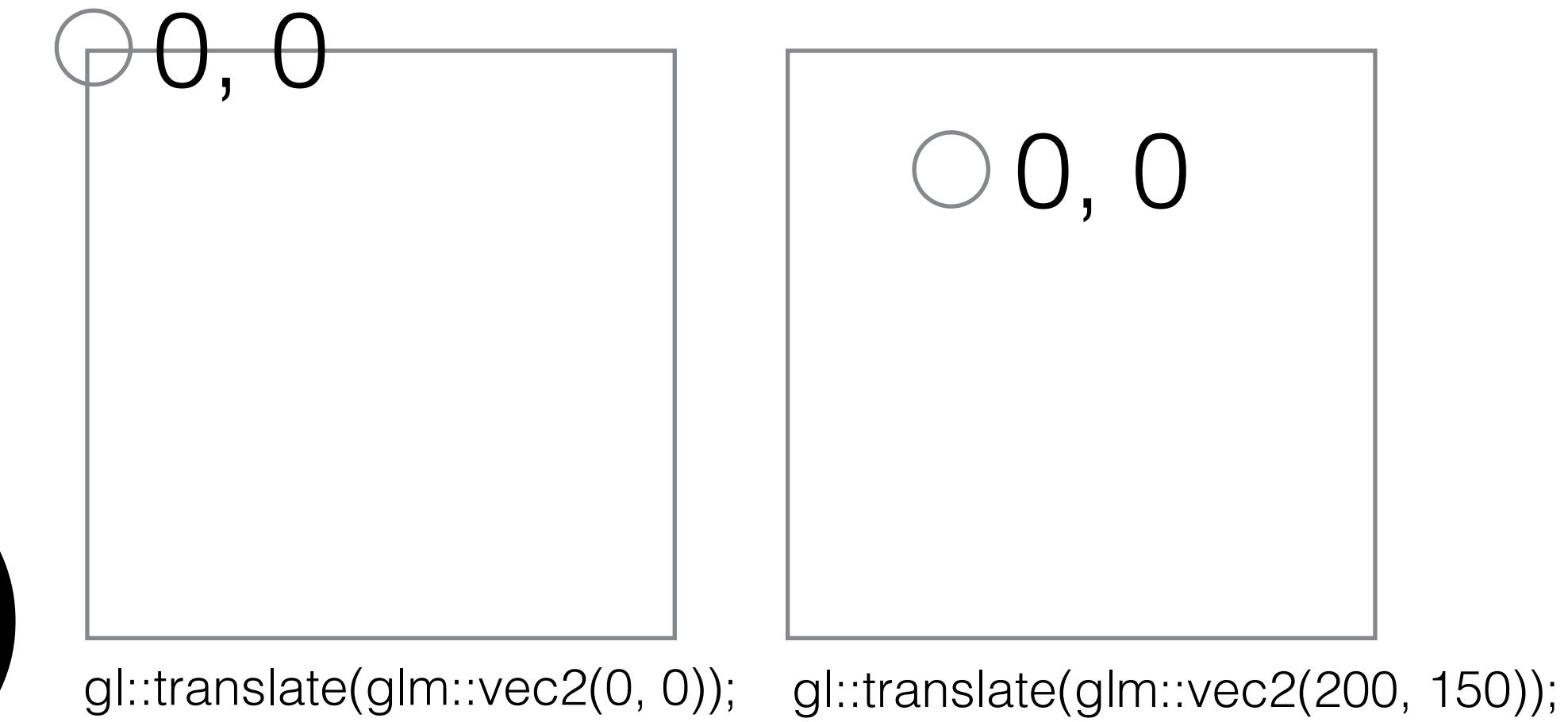
Clean up everything that is on your canvas.

`gl::pushModelView()`

`gl::popModelView()`

Wrap translation inside push and pop to ensure that changes will only be applied inside.

gl::translate(glm::vec2)



Translate the origin point of a canvas to another location.

`ci::Color(r, g, b)`

A cinder object which stores RGB value.

gl::color(r, g, b)
gl::color(c1::Color(r, g, b))

Change your current drawing color.

gl::drawSolidRect()
gl::drawSolidCircle()

Draw basic shape using gl function.

std::time(nullptr)

Returns the current calendar time encoded as a **std::time_t** object, and also stores it in the object pointed to by arg, unless arg is a null pointer.

std::localtime(const time_t *)

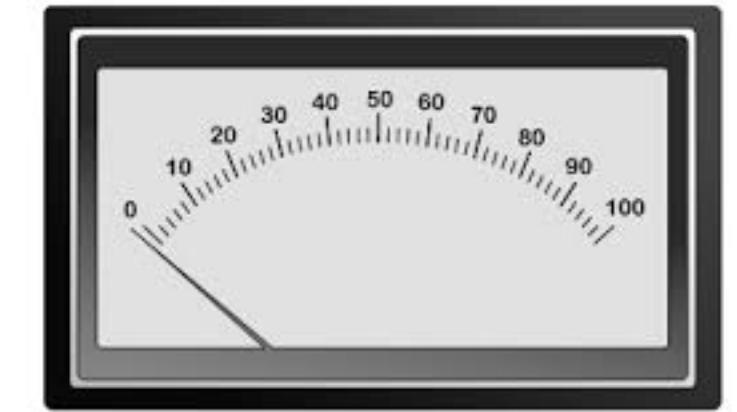
```
auto t = std::time( nullptr ); // getting an time object  
auto d = std::localtime(&t); // get calendar time from time object
```

Homework:

1. Create an analog digit (non interactive)



2. (Bonus project) Create an analog meter that responds to mouse or keyboard input (if you hold key longer the meter should go higher)



Due: Jan 31 Tue

Upload a video demo of each project to our slack channel.

Keep the coder for later integration into your GitHub repo.