Catch

Test everything that could possibly break.

What is unit test?

- Focused on small part of the system.
- Expected to be significantly faster than other kinds of tests.

What is the unit?

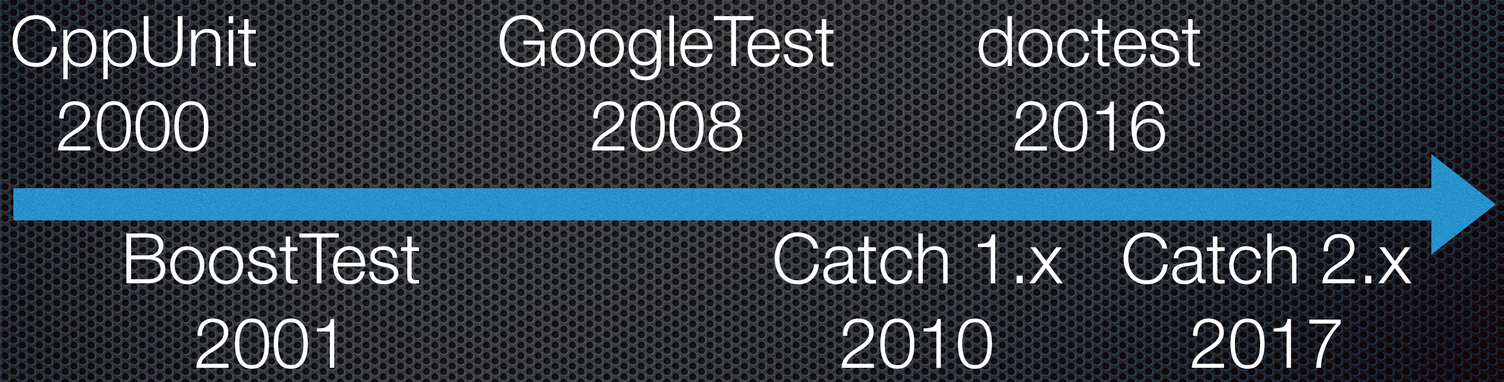
- In OOD is a class.
- In procedural or functional approach is a function.

Solitary or sociable tests?

- Solitary - use stubs, mocks, fakes to substitute dependencies.
- Sociable - use real, production objects.

Jay Fields
Working effectively with unit tests

C++ unit test frameworks



Catch

- C++ Automated Test Cases in Headers.
- Phil Nash - framework author.
- Discussion forum on groups.google.com.
- Project repository on github.com.

Catch - Framework Features

- Easy to start - include catch.hpp.
- Single header - no external dependencies.
- Test cases as self registering functions.
- Both sections and fixtures support.
- Support Behavioral Driven Development style.
- Reduced number of assertion macros.

Catch - Test Case

- TEST_CASE(test name, tag name list)
  - Test name - string unique test identifier.
  - Tag name list - optional, additionally test case string identifiers allowing for grouping test cases.
  - Test names and tag names are not case sensitive.

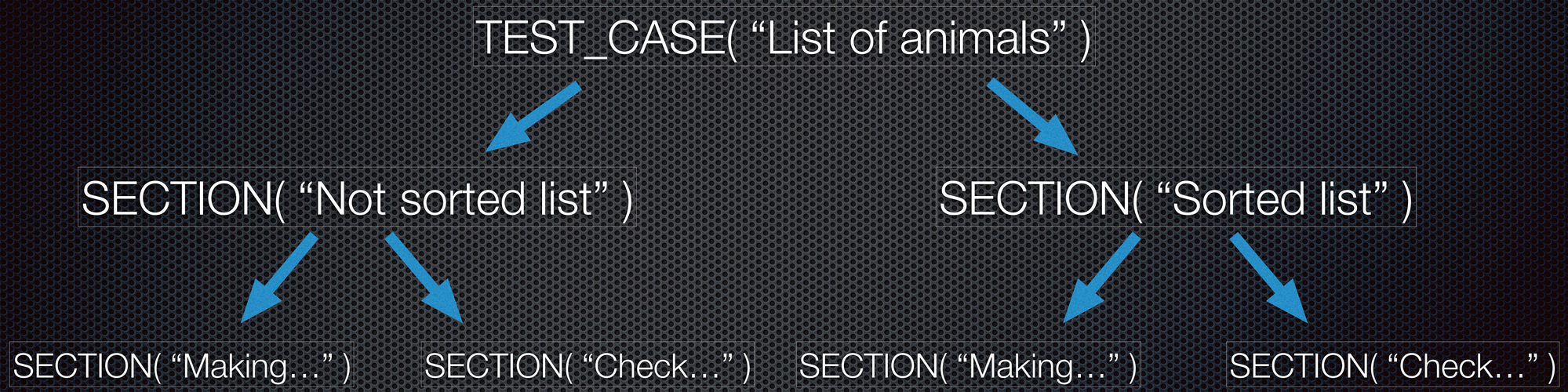
Catch - Assertion Macros

- REQUIRE(expression) - if fails, aborts failure test case.
  - REQUIRE_FALSE(expression)
- CHECK(expression) - if fails, continue test case flow.
  - CHECK_FALSE(expression)

Catch - Section

- SECTION(section name)
  - Section name - not-unique string section identifier.
- Test case can be composed of multiple sections.
- Each leaf section is executed only once.
- Sections can be nested.

Catch - Section



Catch - Assertion Exception Macros

- REQUIRE_NOTHROW(expression)
- REQUIRE_THROWS(expression)
- REQUIRE_THROWS_AS(expression, exception type)
- REQUIRE_THROWS_WITH(expression, string or string matcher)
- All variants are applicable for CHECK macro.

Catch - Matchers

- Works with:
  - REQUIRE_THAT(obj, matcher)
  - REQUIRE_THROWS_WITH(expression, string or matcher)
- Easy way to provide custom matchers.
- All variants are applicable for CHECK macro.

Catch - Comparing Floats

- Tolerant comparisons using Approx class providing:
  - epsilon - the percentage by which a result can be erroneous
  - margin - absolute value by which a result can be erroneous
  - scale - adjust the base for comparison used by epsilon.

Catch - Test Fixtures

- TEST_CASE_METHOD(fixture obj, test name, tag name list)
- SetUp(), TearDown() realized through RAII mechanism.

Catch - Test Output

- Reporters transform test output into specified format:
  - console (default)
  - compact - similar to console, but optimized for minimal output
  - junit - corresponds to Ant’s junitreport, Jenkins ready.
  - xml.

Catch - Event Listeners

- Registering on begin and end of each test.
- Access to test case details.
- Access to test case results.
- Multiple listeners support.

Catch - Compilation

- C++11 - check if supported, can be suppressed by toggle macro definitions.
- 20% compilation speed up by definition CATCH_CONFIG_FAST_COMPILE.

Catch - CLI

- Wide range of command line functions, i.e.:
  - Specifying test cases, tags, sections to run.
  - Listing all test cases, tags, sections.
  - Exporting output to file.
  - Transforming output to specified format.

Catch vs Google Test

Framework feature	Google Test	Catch
Assertion set	Expanded	Basic
Fixtures (xUnit)	Supported	Supported
Floating points comparisons	Supported	Supported
Multiprocess testing	Forking	None
Thread safety	None	None
Parametrized tests	Supported	None

Catch vs Google Test

Framework feature	Google Test	Catch
Typed tests	Supported	None
Exceptions	Supported	Supported
Matchers	Google Mock	Supported
Event listeners	Supported	Supported
Output reports	Supported	Supported
Command Line Interface	Supported	Supported

Catch vs Google Test

Framework feature	Google Test	Catch
Installation	Library	Header
Difficulty level	Medium	Low
Behavioral Driven Development style	None	Supported
Readability	Medium	High
Documentation	CookBooks	Enough
C++11	None	Supported
Can be used with mocking framework	Google Mock	FakeIt

Catch vs Google Test

Community	Google Test	Catch
First release date	2008	2010
Contributors	98	96
Releases	11	60
Commits	1288	1878
Code base	26,5K	11,6K

Catch vs doctest

Pros doctest	Agains doctest
Including header is 20 x lighter on compile time.	No output reporters.
Assertion macros are compiled 30% - 70% faster.	No event listeners.
Test execution 2.5 - 26 times faster.	No matchers.
Test cases can be implemented in headers.	Not many users.

Framework features nice to have

- Minimal amount of work needed to add new tests.
- Easy to modify and port.
- Supports setup/teardown steps.
- Handles exceptions and crashes well.
- Good assert functionality.
- Supports different outputs.
- Supports suits.

gamesfromwithin.com

Exploring the C++ Unit Testing Framework Jungle

The end