July 1st, 2020

BDD and TDD

Agenda

- BDD and TDD Concepts
- Real talk from Industry
- Hands-on with Catch
- Hands-on Emscripten testing

BDD and TDD Concepts

- BDD: Behavior Driven Design
  - Get User Stories before any code
- TDD: Test Driven Design
  - Write tests before any (non-test) code
- In common: don’t just start coding!

BDD and TDD Concepts

- User Story:
  - Varying levels of formality
  - Format:
    - Feature name, “As a [stakeholder], I want to [task] so that [goal]”
    - “As a researcher, I want to be able to change settings on the command line so that I can easily run experiments with different settings”
    - “As a teacher, I want to be able to control settings in a visual interface so that I can use this with young students.”

BDD and TDD Concepts

- User stories should be broken down until they are SMART:
  - Specific – vague goals are impossible to achieve
  - Measurable – you need to know when you are done or making progress to done
  - Achievable – it’s usually always harder than you think
  - Relevant – keep focused on most important goals
  - Timeboxed - specify how long you think it will take

BDD and TDD Concepts

- Test Driven Design:
  - After you know what the user wants, you should write tests
  - Red-Green-Refactor cycle
  - You should ALWAYS verify a test fails first!

BDD and TDD Concepts

- TDD can feel really weird at first
  - How do you write a test when you don't know what the code will be yet????
  - Write a test for the code you wish you had
  - It gets better with practice

BDD and TDD Concepts

- Good tests are:
  - Fast
  - Independent
  - Repeatable
  - Self-checking
  - Timely – ie written before the subject code, not hurriedly at the end of the month for all the code you wrote earlier in the month

BDD and TDD Concepts

- Code coverage: measurement of how much code is tested and how thoroughly
- There are lots of levels of code coverage, just two:
  - S0/Method coverage: every method is called in a test
  - C2/Path coverage: every path that can be taken through the code is taken in a test
- Thorough test suites have more lines of test code than subject code!

BDD and TDD Concepts

- Types of testing:
  - Unit: test one component/method
  - Functional: Test several methods/classes working together
  - Integration/System: Test the whole system at once
  - Characterization: when you have untested code that you don’t know what it does, make a test to capture current behavior before refactoring so you don’t break anything unknowingly