Late-term Assignment

SC-T-303-HUGB, fall semester 2018.

Purpose

Learn to set up technical infrastructure to support agile development. The main challenge is to glue together the bits and pieces through the latter half of this class, into a coherent infrastructure.

Description

Each team should implement the game TicTacToe. The main focus is on the infrastructure and best coding practices described below but not on implemented features.

The project and infrastructure should tackle the following:

1. The codebase is stored in source control system on GitHub from early start of the project (the teacher should be able to track the history of the project from the creation of the very first file).
2. The system is setup with automatic build tool.
3. The build tool runs all unit tests.
4. The output is a deployable or executable artifact (*note this is not applicable if the project running a Web App*).
5. The build script has deploy/install target that deploys the executable outside of the working directory (console app) or on external server (web app).
6. The business logic should be coded using Test Driven Development.
7. The code should be loosely coupled and follow good object oriented design practices.
8. Documentation uses Markdown syntax.
9. Use Automated Continuous Integration Server.
10. Use feature branches and pull request for all features. It must be clear that this workflow was used the whole time and that the team was using reviews before accepting pull-requests.
11. More build targets (0,5 point per item, never more than 1 for this part) here are examples of build targets.
    - Code coverage and other code inspections. Report or other kind of result must be available outside of the build machine after each run.
    - Integrate database (for storing games, players, or something else)
    - Documentation is converted to another format (e.g PDF or Html) and is easily accessible outside of the build machine after each run (the build needs to publish the documents to an external server).
    - Something else ... Surprise us!!!
12. Set up web interface for the game. (1 point)
13. Run End-to-end tests (e.g. Puppeteer). It is necessary to run this on external staging server. (1 point)
14. Continuous Delivery. For each code change the CI server runs these steps (1 point):
    - Run build script (all targets)
    - Run unit and integration tests (if you have integration tests) Deploy on staging and run End-to-end tests.

    - Deploy to production server.
    - Only continue next step if previous step is success
    - Generate feedback to developer

Grade

Solving parts 1-10 perfectly will together give a grade of 6, it's OK to have all targets in build script manual. Next parts need to be solved incrementally, e.g. finish part 11 before starting with part 12. Continuous Delivery is mandatory to get 10. We expect participation from each team member in the git history. If you are pairing please switch drivers regularly (and user on git).

Return form

1. URL of root of the project in the source control system. If the repo is private you need to share it with the user **hap2000**.
2. Documentation. The following report should all be stored in a folder called `docs` in the root of the repository.
    - Development manual: what is needed and how to get the project to build on a fresh machine, Source control client and access to source control Build environment, Other necessary dependencies for development.
    - Administration manual: How to set it up and get it to run, also on a fresh machine. How to install and run the program on clients machine (in case of simple desktop program). How to deploy, run and maintain in case of client/server or web application.
    - Design report: Document describing the initial design.

Demonstration

The team will have a 15 minutes demonstration of the technical infrastructure and the simple functionality developed. The demonstrations will be held in the last week, the detailed schedule will be published later. The distributed groups will demonstrate using shared desktop. The demonstration is part of the grade for this assignment. If you are not able to show some part of the build process (if it is broken) in the demonstration it will not count in the grade.