Build Engineer tech test

Purpose

This exercise's main purpose is to have a starting point for the tech interview later on.

Exercise

The exercise consists of 2 parts. The first one is about creating a build system for a provided C++ project. The second part is about making a Jenkins job that triggers such build system and runs the project's tests.

First part

Given the provided C++ project (named CoolGame/), create a `CMake` solution that is capable of building it and its tests.

Note that one of the unit tests is failing on purpose. You can change it if you prefer it not to fail.

Feel free to move any files around if this can simplify the problem for you. However keep in mind that we value project scalability, and it would have to be easy to extend it with new modules and compilation units.

Second part

Create a Jenkins job that takes this project and builds it and its tests (using the CMake solution from the first part). If the generation or compilation fails, the job should fail.

Make the job run the tests in another stage. If any of the tests fails, the job should be marked as unstable.

The resulting binaries have to be archived in the job itself for a potential future diagnosis.

Hint

You can choose how you want to set up your development environment in order to get Jenkins running in your end. However, we would value very positively that you used a dockerized solution for both the master and the nodes.

Also, feel free to choose how you want this Jenkins instance to fetch the game code.

Submission

Send us a zip file that contains the C++ project with the CMake files and the Jenkins pipeline scripts.

Having a solution for Jenkins that we can run in our machine by executing a script would be ideal.

Add a document file explaining how everything is set up and executed. Add screenshots for reference if it helps to the readability. Feel free to add personal notes or thoughts in there about how things could be improved if you had more time.

Evaluation

Those are the key points that will be evaluated:

- The C++ project must compile and run by using your CMake. Both the main target and the tests.
- We want to see a Jenkins pipeline script that compiles it and runs its tests, and archives the resulting binaries.
- We want the Jenkins pipeline to live in a separate file so that potentially it can be versioned. The job should fetch the C++ project somehow.
- We do value configuration as code. Bonus points if the Jenkins instance can be deployed in our end by running a script that you provide to us with minimum setup.
- We will value creativity. Try to make things easy to be tested in our end.

Estimated time

4 hours