

Testing and Maintenance Phase

In this phase, there are two parts:

1. Your team will be given the URL for the project repository on GitLab of a game created by another team. You will then:
 - a. Use a variety of system-level testing techniques to test the game.
 - b. Provide feedback to the implementing team by filing bug reports/feature requests in the implementing team's issue tracking system.
2. Your team will improve your program, such as:
 - a. Tasks or features you wanted to complete, but didn't before the deadline.
 - b. Responding to any reported bugs or feature requests from the testing team.
 - i. You are expected to fix any bugs that are reasonable to fix, or provide a detailed rationale for why the team is choosing not to address this bug report.

DELIVERABLES

The deliverables for this phase are:

1. A team report describing the actions taken to test the other team's game and improve your game. The report will contain the following sections:
 - a. **Introduction** – An overview of the document. (This need not be very long.)
 - b. **Testing Approach** – A description of how the team tested the software created by the other team, including what techniques were used and a summary of the testing results.¹
 - c. **Improvements** – A detailed accounting of how the your software was improved, such as (but not limited to):
 - i. Number of bug reports/feature requests closed
 - ii. Design changes made
 - iii. Quality improvements to the source code
2. A report on how the team worked during this phase and how the team members contributed to the project.

SUBMISSION

Submit the URL of your team's repository. The reports will be graded from there. Place the reports in the following locations:

<u>REPORT</u>	<u>LOCATION</u>
TESTING/MAINTENANCE REPORT	docs/testing
TEAM REPORT	docs/team/maintenance

¹ Do not cut and paste all of the issue reports here. That is not a summary.

GRADING

The grading of this project phase is flexible to account for the different quality of software you are asked to test and the quality of your own software. Some example situations are:

- A team that is asked to test a buggy project. Such a team may spend more effort on testing than improving their own code in response to the other team fixing the bugs and re-testing their project.
- A team is asked to test a well written project (i.e. few bugs). Such a team may spend little effort on testing and more effort on improving their own code.
- A team created a buggy project that barely works. Such a team would spend a lot of effort improving their own code.
- A team created a well written project (i.e. few bugs). However there were features that were planned but not implemented, and/or design patterns could be added to make the software more flexible. Such a team would spend their effort improving their code.

In a four-person team, it is common that two members are focus on the task of testing the assigned project and two members focus on responding to change requests and improving the team's code.

A way to think of the grading for this phase is that there is bucket that is filled by two hoses. The level of water in the bucket represents the grade as assessed by such things as the phase report, issue tracking systems (both their own and the project they are testing), and commits in your source control system. These items will indicate which of the two hoses (testing or maintenance) provided the water (activities) to the bucket and how much of the bucket was filled. **The phase report where you will put forth your case as to what is the water level in the bucket.**

