2018

Tanks

TEAM ZERO
KOVARIK, DENNIS J.

Revision Hisotry

Document History

Version	Date	Author	Description of Change
1	8/26/2018	Dennis Kovarik	Started First Draft
1	8/31/2018	Dennis Kovarik	Finish 1st draft of Test Plan
2	10/1/18	Dennis Kovarik	Updated Test Plan

Contents

1. Introduction

1.1. Overview

Tanks is a turn-based computer game in which multiple players (computer or human) play against each other by controlling individual tanks. As per the specifications outlined by the minimum viable product, the game will manage at least 2 players, 2 projectiles, along with a turn counter. In addition, the hits and misses for each player will be recorded and displayed on the screen.

1.2. Plan Purpose

This test plan contains a description of the testing approach for developing the game "Tanks". This document includes the following

- Strategy: This includes a list and description of the test rules, project assumptions, and a comprehensive description of the test setup.
- Implementation: Includes detailed descriptions of the defect management process and how testing is performed.
- Management: Description of the logistics of testing. This includes the chain of command, risk identification and management, and coordinating the activity of team members.

1.3. Objectives

The objective is to verify that the Tanks application is bug-free and meets the criteria set forth by the stakeholders. Testing will include unit tests, integration tests, feature tests, performance tests, and system tests. Defects and issues will be forwarded directly to the Development Team, the DevOps Team, and the project manager.

Criteria for cessation of testing:

- The software meets the criteria outlined by the minimum viable product.
- The software passes all tests designed by the team
- Unable to design any more tests to break the software.
- Software has no identifiable bugs or defects.

1.4. Anticipated Audience

- Team members mentioned in this document
- Development Team
- Project managers
- Stakeholders
    - Everyone on Team Zero
    - Everyone on all the other teams
    - Dr. Hinker

2. Testing Scope and Principles

2.1. Assumptions

- The test environment will exactly duplicate the production environment.
- Issue reporting will include both the issue itself and details on how to reproduce the issue.

2.2. Test Principles

- Quality and meeting the project objectives are the top priority.
- Test Driven Development
    - You are not allowed to write any production code unless it is to make a failing unit test pass
    - You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
    - You are not allowed to write any more production code than is sufficient to pass the one failing unit test.
- Testing is divided into 2 Test Cycles (Certification Cycles).
- When an issue has been reported and fixed, regressions tests must be performed.
- Tests will be automated whenever possible
- Tests are designed to test the quality of the software

2.3. Testing Scope

2.3.1. Unit Tests

Tests a single component of the program. These tests are performed throughout the development process.

2.3.2. Integration Tests

Tests a combination of functions and/or modules. These tests may be manual or automated.

2.3.3. Functional Tests

Typically tests a requirement or feature

2.3.4. Exploratory Tests

Exploring and playing with the software to find out what it can do. Integrates learning about the program being developed with plans for future testing

2.3.5. Regression Tests

Tests to ensure that changes due to bug reports don’t reintroduce bugs

2.3.6. Performance Tests

Testing to determine the speed, responsiveness, and stability of software under a workload

2.3.7. System Tests

Testing of complete and fully integrated software product.

3. Effort Estimate

The first phase of the project should take around 8 weeks to complete.

4. Strategy

4.1. Test Cycles (Certification Cycles)

- There will be 2 Test Cycles.
- Test Cycle 1
    - Begins at the end of Sprint 1 (9/15/18)
    - The main objective of Test Cycle 1 is to preform exploratory testing, so the QA team can become familiar with the software. Additionally, the QA team will identify critical defects and barriers to advancement.
    - Development team will compile automated unit tests to adequately test individual modules
- Test Cycle 2
    - Begins at the end of Sprint 2 (9/29/18)
    - Develop and run integration tests to test integrating modules. These tests may be manual or automated.
    - Develop and perform performance and stress tests for software.
    - Perform end to end testing of the product to verify that it is ready for release. Additionally, issues reported fixed from Cert 1 will have regression test performed on them.

4.2. Issue Tracking and Management

- Issues will be tracked using the GitLab “Issues” feature. Issues will require the following:
    - The File in which the issue can be found in
    - A description of the issue
        - If the issue is a bug, a description of how the bug can be replicated will be included.
    - A date for when the issue should be addressed/corrected
    - May include an assignee to correct the issue
- Code Reviews will be conducted during the testing cycle to detect bugs and improve the efficiency of the software.
- QA team will execute all agreed upon automated tests and scripts. The QA team can write and perform additional tests if deficiencies in the test protocol are observed.

Severity	Risks
1 (Critical)	• Defect causes application to crash, hang • Corrupts application or system data • Consumes system resources to the point that other system processes are adversely affected
2 (High)	• Missing major application functionality without a workaround
3 (Medium)	• Missing minor application functionality without a workaround • Missing major application functionality with a workaround • Defect causes other features to be unavailable for review or testing
4 (Low)	• Minor feature not working as per requirements but functionality is testable using a workaround
5 (Trivial)	• Tooltip, help screen, error message incorrect or unclear • User interface icons, color palette, font size/type inconsistent or at variance with UX specifications • Typos in help files or help screens • Test misaligned or inconsistently aligned between screens

Figure 1: Reference table for assigning severities different types of defects

4.3. Metrics

Tracking the progress of the project.

- Weekly status report: All teams meet to provide a status report. The meeting time is typically Tuesday from 2:00 p.m. to 4:00 p.m.
- Regular team meetings: Induvial teams (development, devOps, ect…) will meet in order to develop the software.

4.4. Defect Reporting

- All defects will be tracked on GitLab using the “Issues feature”. All defects will be reported to the Development and DevOps team as soon as possible.
- All reported defects will require the following:
    - The File in which the issue can be found in
    - A description of the issue
        - If the issue is a bug, a description of how the bug can be replicated will be included.
    - A date for when the issue should be addressed/corrected
    - May include an assignee to correct the issue

5. Test Management

5.1. Test Management Tool

- Code Coverage

5.2. Test Design

- All unit test will require the following
    - Test one thing
    - Automated

        - Repeatable meaning that anyone can run the tests and get consistent results
        - Fasts and runs quickly
        - Independent meaning on unit test depends on a different test
        - Self-validating
            - Failure indicates where the problem is
            - Passing tests indicates working code
- QA team reviews requirements under test and prepares a test which verifies that the requirement is met.
- Test cases are mapped to User Stories and Requirements as part of the requirement tracking.
- The Development, DevOps, and QA teams will use prototype, user stories, and functional specifications to write step by step test cases.

5.3. Executing on the Test Plan

- There will be 2 Test Cycles.
- Test Cycle 1
    - Begins at the end of Sprint 1 (9/15/18)
    - The main objective of Test Cycle 1 is to preform exploratory testing, so the QA team can become familiar with the software. Additionally, the QA team will identify critical defects and barriers to advancement.
    - Development team will compile automated unit tests to adequately test individual modules
- Test Cycle 2
    - Begins at the end of Sprint 2 (9/29/18)
    - Develop and run integration tests to test integrating modules. These tests may be manual or automated.
    - Develop and perform performance and stress tests for software.
    - Perform end to end testing of the product to verify that it is ready for release. Additionally, issues reported fixed from Cert 1 will have regression test performed on them.
- The Development and DevOps Team will be responsible to creating and maintaining all unit and integration tests.
- Test cases are reviewed by the Development, DevOps, and QA teams to ensure the test faithfully validates existing requirement(s).
- The DevOps Team will be responsible for creating regression tests for each bug reported.
- QA Team will be responsible for validating integration tests developed by the DevOps and Development teams.
- QA team is responsible for creating and running Exploratory and Functional Tests.
- Defects are logged using an issue tracking system. QA team is responsible for the initial assignment of severity, but the final determination is made by the Development or the DevOps team.

- Any defects marked as fixed in a previous test cycle are verified using test scripts and regression tests.

5.4. Risks and Risk Response

Risk	Likelihood	Effect	Response
Losing a team member.	Possible but not very likely.	Interruption of development process.	The project manager reevaluates and reassigns teams based on need.
Unforeseen delays	There's no way to anticipate something like this so let's mark it low and hope for the best.	Impossible to tell the impact due to not knowing the issue.	If necessary, we can use some time from holidays or weekends to make up for this sort of issue.

6. Team Leads and Contact Info

Name	Position	Contact Info
Aidan Anderson	Development Team	aidan.anderson@mines.sdsmt.edu Cell: 360-525-4927
Wesley Adams (Wes)	Project Manager Communications Team	Wesley.adams@mines.sdsmt.edu Cell: 818-792-6869
Joao Pessoa (JD)	Development Team	joao.pessoa@mines.sdsmt.edu Cell: 719-232-7459
William Doering (Will)	UI	william.doering@mines.sdsmt.edu Cell: 605-680-2324
Samuel Backes (Sam)	Development Team	samuel.backes@mines.sdsmt.edu Cell: 605-431-9829
Dennis Kovarik	QA	dennis.kovarik@mines.sdsmt.edu Cell: (605) 391-1646
Jonathan McKee	DevOps	jonathan.mckee@mines.sdsmt.edu Cell: (951) 764 -7432
Levi Butts	UI	levi.butts@mines.sdsmt.edu Cell: (605) 520-2935
Gwyneth Kardelis (Gwyn)	UI	gwyneth.kardelis@mines.sdsmt.edu Cell: (435) 299-5190
AdeshKumar Naik	DevOps	adeshkumar.naik@mines.sdsmt.edu

6.1. Roles and Responsibilities

6.1.1. Project Management

- Review, verify, and confirm
  - Test Plan
  - Test Strategy

6.1.2. DevOps Team

- Design Unit Tests
- Design Integration Tests
- Provide clarification on functionality and aid the QA Team in becoming familiar with the product.
- Fix defects discovered by Team as prioritized.
- Aid QA Team concerning product features (when requested)

6.1.3. Development Team

- Provide clarification on functionality and aid the QA Team in becoming familiar with the product.
- Fix defects discovered by QA Team as prioritized.
- Aid QA Team concerning product features (when requested)
- Fixes issues uncovered by tests and bug reporting.

6.1.4. QA Team

- Ensure environment availability
- Create and maintain the Test Plan
- Set up process for identifying, recording, and communicating defects.
- Communicate test progress and completion for each test cycle.
- Give go-ahead for next test cycle at the completion of each cycle.
- Perform exploratory testing and report, develop, communicate observed inconsistencies, gaps, or ambiguous requirements.
- Execute tests (test scripts, automated tests, and regression tests as needed)
- Identify, record, and report defects. Provide initial severity rank.

7. Test Environment and Product Requirements

As per the specifications outlined by the minimum viable product, the product will support a Graphical User Interface (GUI). There will be at least 1 playfield, and the game will manage at least 2 players, 2 projectiles, and a turn counter. In addition, the hits and misses for each player will be recorded and displayed on the screen.

The test environment will include a computer running a Linux operating system and/or a windows device running a Linux subsystem.