

Christy Bergevin, Riley Campbell, Nate Fagrey, Logan Larson

<https://gitlab.mcs.sdsmt.edu/7416949/tank-project-ii>

March 10, 2019

Version: 2.0  
Last Updated: 04/25/2019  
Status: Finalized

Test Plan & Design   
Easy-Eight

Design Document

**Revision History**

**Document History – Track changes made to the document**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of Change |
| 1 | 03/10/2019 | Christy Bergevin | Draft |
| 2 | 04/25/2019 | Christy Bergevin | Final |
|  |  |  |  |

Table of Contents

[1. Introduction 3](#_Toc3207842)

[1.1. Plan Purpose 3](#_Toc3207843)

[1.2. Overview 3](#_Toc3207844)

[1.3. Anticipated Audience 3](#_Toc3207845)

[2. Strategy 4](#_Toc3207846)

[2.1. Objectives 4](#_Toc3207847)

[2.2. Assumptions 4](#_Toc3207848)

[2.3. Test Principles 4](#_Toc3207849)

[2.4. Testing Scope 5](#_Toc3207850)

[2.4.1. Functional 5](#_Toc3207851)

[3. Effort Estimate 6](#_Toc3207852)

[4. Strategy 6](#_Toc3207853)

[4.1. Test Cycles (Certification Cycles) 6](#_Toc3207854)

[4.2. Issue Tracking and Management 6](#_Toc3207855)

[4.3. Metrics 7](#_Toc3207856)

[4.4. Defect Reporting 7](#_Toc3207857)

[4.5. Start and End Criteria 8](#_Toc3207858)

[5. Test Management 8](#_Toc3207859)

[5.1. Test Management Tool 8](#_Toc3207860)

[5.2. Test Design 8](#_Toc3207861)

[5.3. Executing on the Test Plan 9](#_Toc3207862)

[6. Team Leads and Contact Info 1](#_Toc3207864)0

[6.1. Roles and Responsibilities 1](#_Toc3207865)0

[6.1.1. Team Lead 1](#_Toc3207866)0

[6.1.2. QA Team 1](#_Toc3207867)0

[6.1.3. Development Team 1](#_Toc3207868)0

[7. Test Environment and Product Requirements 1](#_Toc3207869)0

# Introduction

## Plan Purpose

This test plan contains a description of the testing approach used to create a comprehensive plan for the testing of the Easy Eight Version 1.0 tank AI development for CSC 340 Spring ‘19. This document includes:

* Strategy: Test rules and project assumptions including number of test cycles, test objectives, milestones, and dates); end-to-end description of test set up (start / end conditions, test case creation, task lists, data sources).
* Implementation: how is testing performed, defect management process (reporting, escalation, fix planning and execution)
* Management: description of the logistics of testing (e.g. chain of command, risk identification and management, team members)

## Overview

Easy-Eight is a tank AI developed for the Spring 2019 Software Engineering course at SDSM&T. The AI will be a part of a battle royale game in which the opponents can include a Target (immobile, non-attacking), a Mouse (fleeing, non-attacking), a Turret (immobile, attacking), and a Sentry (mobile, attacking). The tests will include one or many opponents. Tests will also include a map with no obstacles, a map with obstacles, and an optional test of a blocked path.

## Anticipated Audience

* Team members mentioned in this document who will perform assigned duties and provide addition recommendations.
* Dr. Paul Hinker who is in charge of the final grade.
* Classmates from other teams who are also developing tanks

# Strategy

## Objectives

Test objectives are meant to verify that the Easy-Eight tank AI meets design specifications.

Testing will include the execution of automated tests, test scripts, and performance tests. Issues will be forwarded to the Product Manager and prioritized as high, medium, or low severity. High and Medium severity defects must be retested as per the acceptance criteria. Low severity defects will be scheduled by the Product manager or deferred to a subsequent release.

Criteria for cessation of testing:

* Easy-Eight tank wins 100% of games played for given scenario
* Automated tests and test scripts suitable for reuse as function and user-acceptance testing.

## Assumptions

**General**

* Test design and environment is owned by QA.
* Environment downtime will adversely impact test schedules.
* Test environment will exactly duplicate production environment.
* Performance testing is owned by QA but reviewed and signed-off on by Team Lead (Nathan Fagrey).
* Issue reporting includes **complete** reproduction details (as per Issue Reporting Template).
* Issues are tracked using GitLab.
* Issues reported fixed after Certification N will include regression tests which will be added to the test plan for Certification N+1.

**Key Assumptions**

* Release cannot go in to production with any severity 1, 2 (high) or 3 (medium) defects.
* Functional testing requires production-like data.

## Test Principles

* Testing will, at a minimum, cover all scenarios for the target, mouse, turret, and sentry opponents.
* Testing cycles will be controlled by agreed on start and end criteria.
* Testing is divided into test cycles (Certifications).
* Certification environment must exactly duplicate the production environment to support meaningful test results.

## Testing Scope

### Functional

Who: QA Team.

When: At the beginning of each

Why: Thorough testing of application functions.

Scope: Game Play, with and without GUI

How: Tests are performed using agreed on scripts using specific edge-case scenarios and at least 100 random scenarios

Acceptance Criteria:

1. Functional spec and use case documents available before Test Design phase.
2. Test environment available, configured, verified, and ready to use.
3. Unit tests results compiled and shared with QA to avoid duplication of effort.
4. Test cases reviewed, validated, and signed off on by Team Lead or developer

Deliverables:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | Owner | Reviewer |
| 1. | Test Plan | QA | Team Lead: Nathan Fagrey |
| 2. | Functional Tests | QA | Developer: TBD |
| 3. | Defect List | QA | Developer: TBD |
| 4. | Completion Report | QA | Team Lead: Nathan Fagrey |

Timeline:

Milestones are subject to slip under the following conditions:

1. Test environment unavailable or unresponsive during test cycle.
2. Change or expansion of requirements (scope creep).
3. Unforeseen circumstances which impact availability of personnel, environment, or equipment.

# Effort Estimate

With 4 team members, it is expected that with each week, the whole team will be working toward the same feature goal. Effort estimate is on track for passing the requirements for grading. Once those requirements are met, if there is time, we will add additional functionality for PVP and Battle Royale.

# Strategy

## Test Cycles (Certification Cycles)

* There are three planned certification cycles. Each cycle will consist of one week of testing followed by a weekend in which another developer will compile, construct, and submit defects through GitLab.
* The objective of Cert 1 is to create the testing scenarios with all standard grading opponents, including all edge cases and a randomizer.
* The objective of Cert 2 is to test the newest tank against our own previous versions and alterations of previous versions that are identified as possible design flaws.
* The objective of Cert 3 is to identify all remaining high and medium severity defects, obtain performance metrics, execute any regression testing from Certs 1 & 2, and ensure the system is ready for grading.
* (Optional) The objective of additional Certs is to test against other unclassified opponents and blocked paths

## Issue Tracking and Management

* All team members are responsible for opening issues using Google Drive. The Team Lead is responsible for reviewing and assigning severity to each issue in a timely manner. It is the responsibility of the Development Team to request additional information as needed, fix the defect, update the defect’s status, and integrate the fix so that it can be regression tested during the next test cycle.
* Issues tracked through Google Drive.
* Each team member is responsible for executing all agreed on test scripts and automated tests. It is also understood that other developers may perform additional testing when deficiencies in the test protocol are observed as long as the additional testing can be reproduced if defects are discovered during the additional testing.

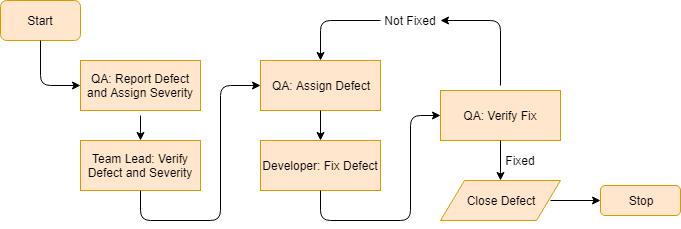
|  |  |
| --- | --- |
| **Severity** | **Risks** |
| 1 (Critical) | * Defect causes application to crash, hang * Corrupts application or system data * Consumes system resources to the point that other system processes are adversely affected |
| 2 (High) | * Missing major application functionality without 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 |

## Metrics

Tracking progress and success of the tests for each test cycle. Delivered to the whole team.

* Daily status report: Pass/Fail/Complete percentages. Detailed list of status for New, Existing, Closed defects. Identify and highlight any defects marked Showstopper or Critical.
* Weekly Status Report: Provided to Project Manager and Business Analyst as well as Dev. Team. Includes burn down charts, weekly pass/fail/complete percentages. Identify and troubleshoot any defects in the Showstopper or Critical category which have persisted for over a week. Report ongoing risks and mitigation plans.
* Cycle end report: Compile trajectory graphs for defect lists broken out by status, severity, and age.

## Defect Reporting



## Start and End Criteria

* Start criteria detailed in the Test Planning Strategy section.
* Start criteria refer to the desirable and necessary conditions which need to be in place before test execution can be started.
* Start and end criteria are flexible since, especially during Cert 1, it is understood that environments, accounts, data, and documentation may still be in an immature state. Start criteria will be evaluated by Product Manager for a go no-go determination at the start of a cycle.
* Start criteria expectation is that the activities listed in the Test Execution section of the schedule are all in place at the start of each cycle.
* End criteria
  + 100% Test Script execution
  + 95% pass rate on Test Scripts
  + 0 severity 1 or 2 level defects
  + 95% severity 3 level defects closed
  + 100% Test strategy metrics collected
  + 100% of defects logged in GitLab
  + Final Test report reviewed, verified, and signed off on by Team Lead

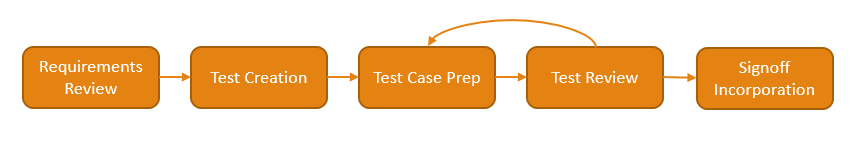
# Test Management

## Test Management Tool

Test Management is accomplished using a variety of tools. All testing artifacts, documents, issues, test cases, and results are stored, verified, and updated using GitLab.

* Developer technical communications including internal chat logs, forum conversations, and WebEx technical presentation recordings will be placed under workflow control.
* The whole team has Read/Write access to test scripts, expected results, and automated tests. Team lead will have sole access to merge tests to master.
* During test design, tests will be placed under revision control to ensure logging of change history.
* Each team member has access to individual test results, issue documentation, and daily issue tracking reports.
* Defects marked as fixed are re-assigned to another developer for validation and either re-opening or closing.

## Test Design



* Individual developers member reviews requirement under test and prepares a test which verifies requirement is met.
* Test cases are mapped to User Stories and Requirements as part of the requirement tracking.
* Test cases are reviewed by a Developer to ensure the test faithfully validates existing requirement(s).
* Individual developers will use prototype, user stories, use case, work flow, and functional specifications to write step by step test cases.
* Change requests must be reviewed and accepted by Dev Team.

## Executing on the Test Plan

* During the first test cycle, exploratory testing is performed to aid in all developers familiarization with the product and environment. Exploratory testing may result in addition to, clarification of, or removal of requirements via Change Requests.
* Each developer performs testing tasks as per test plan. Defects are logged using TFS or other issue tracking system. Individual developer responsible for initial assignment of severity but final determination made by Product Team or Business Team.
* Daily progress charts are created tracking test run, passed, failed, and remaining. Additionally, burndown and hotspot reports are generated for weekly aggregation by Product Manager.
* Non-product related issues are also logged and escalated to the Support / Dev teams as appropriate. E.G. Environment unavailable, production data unavailable or incomplete, etc.
* Product issues related to defects that prevent execution on test plan are reported, logged, and escalated as necessary to the Product Team and Dev Team. E.G. defects causing product features to be unavailable for testing.
* Any defects marked as fixed in a previous test cycle are verified as fixed using test scripts and regression tests.

# Team Leads and Contact Info

All team members are also developers

|  |  |  |
| --- | --- | --- |
| Position | Name | Contact Info |
| Team Lead | Nathan Fagrey | Nathan.Fagrey@mines.sdsmt.edu |
|  | Riley Campbell | Riley.Campbell@mines.sdsmt.edu |
|  | Logan Larson | Logan.Larson@mines.sdsmt.edu |
|  | Christy Bergevin | Christina.Bergevin@mines.sdsmt.edu |

## Roles and Responsibilities

### Team Lead

* Review, verify, and confirm
  + Test Plan
  + Test Strategy
  + Test Estimates

### QA Team – Each developer became a member of our QA team

* Initial draft of test plan
* Process for identifying, recording, and communicating defect reporting
* Initial draft of issue reporting document (for review by Team Lead)
* Acknowledge and communicate test progress and completion for each test cycle.
* Give go-ahead for next test cycle at the completion of each cycle.
* Execute tests (test scripts, automated tests, and regression tests as needed)
* Identify, record, and report defects. Provide initial severity rank.
* Produce testing metrics as outlined above.

### Development Team

* Review test plan, QA deliverables (burndown charts, test scripts, functional tests, etc.). Sign-off or facility modifications as appropriate.
* Provide clarification on functionality and aid in initial QA Team product familiarization.
* Deliver agreed on product components as per scheduled dates.
* Communicate barriers to the schedule or product features in a timely manner.
* Implement fixes to defects discovered by QA Team as prioritized by Product Team.
* Provide assistance to QA Team concerning product features (when requested)

# Test Environment and Product Requirements

Easy-Eight tank AI is designed for implementation into the Spring 2019 Phase II Software Engineering Tank Game platform.

A Windows, MacOS, or Linux device with GLUT is needed to run the UI version of the game.