Test Plan

Introduction

- **Project Name**: Lance of Fate
- **Version**: 0.8
- **Dates**: 13/05 - 19/05
- **Prepared By**: Group 8 (ALP, MAKSAT, TAN, YILMAZ); ANIL, TOYGAR (TBD)
- **Objective**: Testing the project using JUnit before moving on to the implementation of network features.

Scope

- The following will be tested:
  - Testing the GameObject class and extended classes to verify they are working correctly.
  - Testing the Manager classes to verify game mechanics and additional functionality are working correctly.
  - Testing the GUI classes mainly to determine the related threading issues; Understanding how they are updated compared to the manager classes.
- The following will not be tested:
  - Database: There is still room for improvement when it comes to integrating it with the rest of the code.
  - Spells: Testing them before making sure the FireBall and other game objects are working as intended will not be productive.
  - Build Mode: Not a priority at the moment.

Strategy

- Testing Levels
  - Units: To verify classes are fine on their own.
  - Interactions: Testing the manager interfaces to test the interaction of different game components.
  - System (Threads): Validating the threads are updated correctly so that the game mechanics are working as intended.
- Test Types
  - Boundary / Hit detection: Ensuring that objects are intersecting correctly (right now there are issues with higher game speeds).

    - Assets: For some machines the game fails to find the assets on specified paths even though we use the relative path.
    - Physics: Ensuring the ball, barriers, and the staff are moving as intended and the ball bounces in the specified way.

Environment

- Hardware
    - MacOS & PC platforms to ensure that Swing GUI doesn’t have problems since behavior can change across platforms.
    - An arbitrary computer on one of the labs in school to ensure the game runs on those.
- Software
    - VirtualBox (Linux environment test)
    - Different IDE’s (VSCode, IntelliJ IDEA, Eclipse, etc.)

Test Cases

`<<Not finalized>>`

- TC 1: Bounce test
    - Preconditions:
        - A level is initialized.
    - Steps:
        - Launch fireball.
        - Wait for the fireball to come into collision with a barrier. Confirm that it bounces off of the barrier.
        - Wait for the ball to come down and catch it with the staff.
        - Confirm that the fireball bounces from the staff.
        - Repeat multiple times to ensure the behavior doesn’t change as the game goes on.
    - E. Result
        - FireBall bounces off of barriers and the staff in the intended way.
- TC 2: Hit test
    - Preconditions:
        - A level is initialized.
    - Steps:
        - Launch fireball.
        - Wait for the fireball to come into contact with a barrier.
        - If the barrier is not a firm barrier, it should be destroyed; Otherwise its hp should be

        decremented by 1.
    - E. Result
        - If the barrier is destroyed it should be removed from the level, and should no longer be visible on the GameGUI.

Test Data

- Unit Tests: Mock objects (using mockito library and JUnit).
- Integration / System Tests: Play Testing, Debugging.

Resources

- Tools: IDE’s and relevant java libraries.
- Members
    - Confirmed
        - Alp
        - Maksat
        - Tan
        - Yilmaz
    - Unknown
        - Anil
        - Toygar

Schedule

- A decent amount of tests should be written by the meeting time (12:00) on Friday, ALL tests should be written by Saturday at noon.
- All testing should be completed until saturday midnight.
- Deliverables should be prepared on Sunday or earlier if possible. SUBMISSION WILL BE MADE ON SUNDAY, 23:30; Complete and share your work then. We will not adjust the report to include your part if you deliver it on the last minute.

Deliverables

- Test Case documentation — Java Docs will be added to a PDF
- Test Outputs — Screenshots that will be added to a PDF (When appropriate)
- A list of found bugs.
- Summary of tests.

Risks & Mitigations

- Risk: Deadline
  - Mitigation: As mentioned in the schedule, we will not include the tests that are made on the last minute; We expect all testing to be completed on Saturday.
- Risk: Code Overhaul due to Threading
  - Mitigation: When solving the Threading issues we shouldn't change unrelated stuff, even if we have better ideas for it; Otherwise our tests will be obsolete.

Approval

- Awaiting TA review.