**COIT13229**

**Applied Distributed System**

**ZMQ Test Documentation**

|  |  |
| --- | --- |
| **Name:** | Ujjwal Dhakal |
| **Student Number:** | 12222900 |

|  |  |
| --- | --- |
| **Name:** | Sultan Zahid |
| **Student Number:** |  |

Table of Contents

[Tests 3](#_Toc199116643)

[Functional Test Cases 3](#_Toc199116644)

[TC-1: Send base64 chat message 3](#_Toc199116645)

[Fault Tolerance Test Cases 3](#_Toc199116646)

[FT-1: Inject Delay Before Socket Receive 3](#_Toc199116647)

# Tests

## Functional Test Cases

### TC-1: Send base64 chat message

**Test Name:** test\_message\_hello\_world()

**Test Type**: Automatic and manual

**Description:** Sends “Hello world” as a chat message from one peer and checked via /update

**Expected**: Boolean value to verify that chat message is received (True)

**Actual:** True

**Status:** Pass

**Screenshot:**

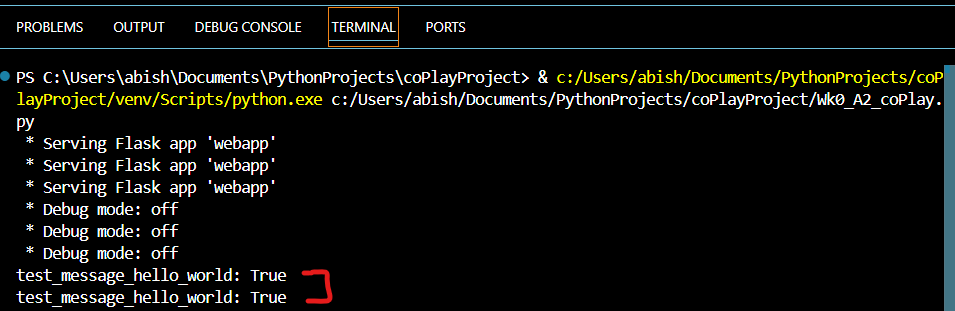


Fig: hello world message test output

**A screenshot of a computer

AI-generated content may be incorrect.**

Fig: Manual hello world message test

### TC-2: Click tower and trigger GET

**Test Name:** Click Tower Test

**Test Type**: Manual

**Description:** Triggers disk movement via GET and reflects across all peers

**Expected**: Same disk moves across all peers

**Actual:** Exact same disk movement among host and peers.

**Status:** Pass

**Screenshot:**

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Initial Game State

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Same disk moves across all peers

### TC-3: Reset button clicked

**Test Name:** Click Reset button

**Test Type**: Manual

**Description:** Game resets after reset button is clicked.

**Expected**: Game resets across all peers when reset button is clicked

**Actual:** Game resets but only on the peer whose reset button is clicked

**Status:** Partial; [Pass (Game resets); Fail (Game resets only for that specific peer)]

**Screenshot:**

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Game completed in fewest moves

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Games resets for only one peer.

TC-4 Browsers Polling /update

**Test Name:** Polling /update across peers to test consistency **Scope:** In-scope   
**Test Type:** Manual   
**Description:** Checking the consistency across 2 peers considering chat, disk movement and reset

**Expected:** Chats, Disk movements and Reset are in sync and consistent at all times.

**Actual:** Initial chat and disk moves are consistent but inconsistent after reset.

**Status:** **Partial**

* Chat and initial tower moves are consistent (Pass)
* Reset does not re-broadcast entire state, causing divergence post-reset (Fail**)**

**Screenshots:**

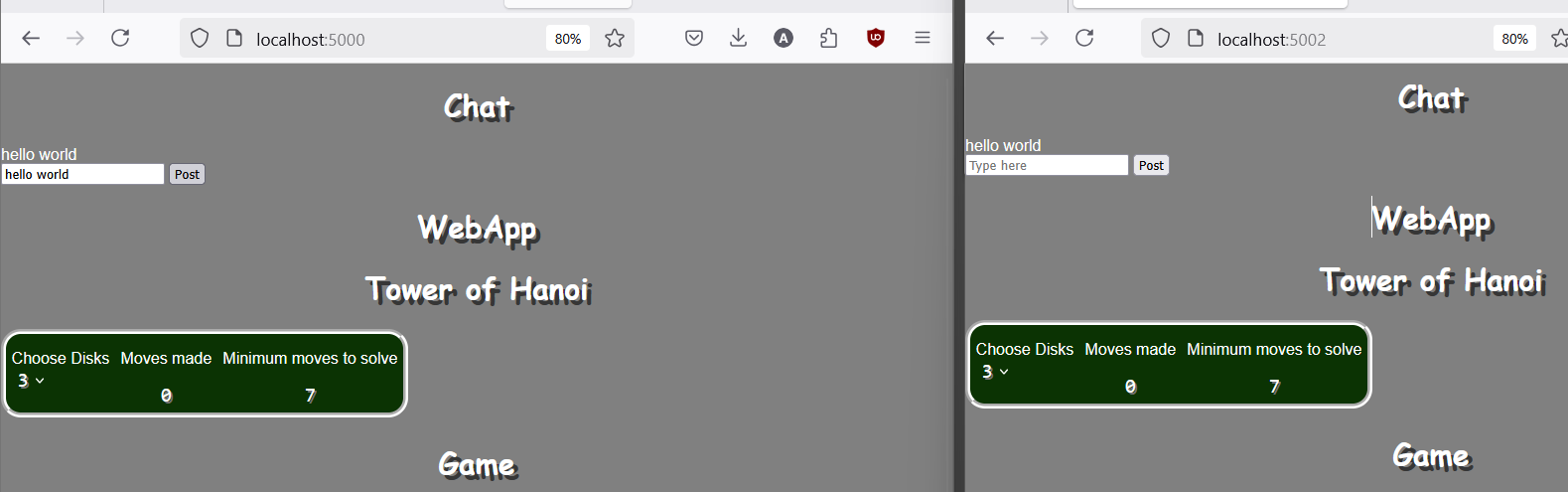
****

Fig: Consistent Chat State

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Consistent Disk Movements

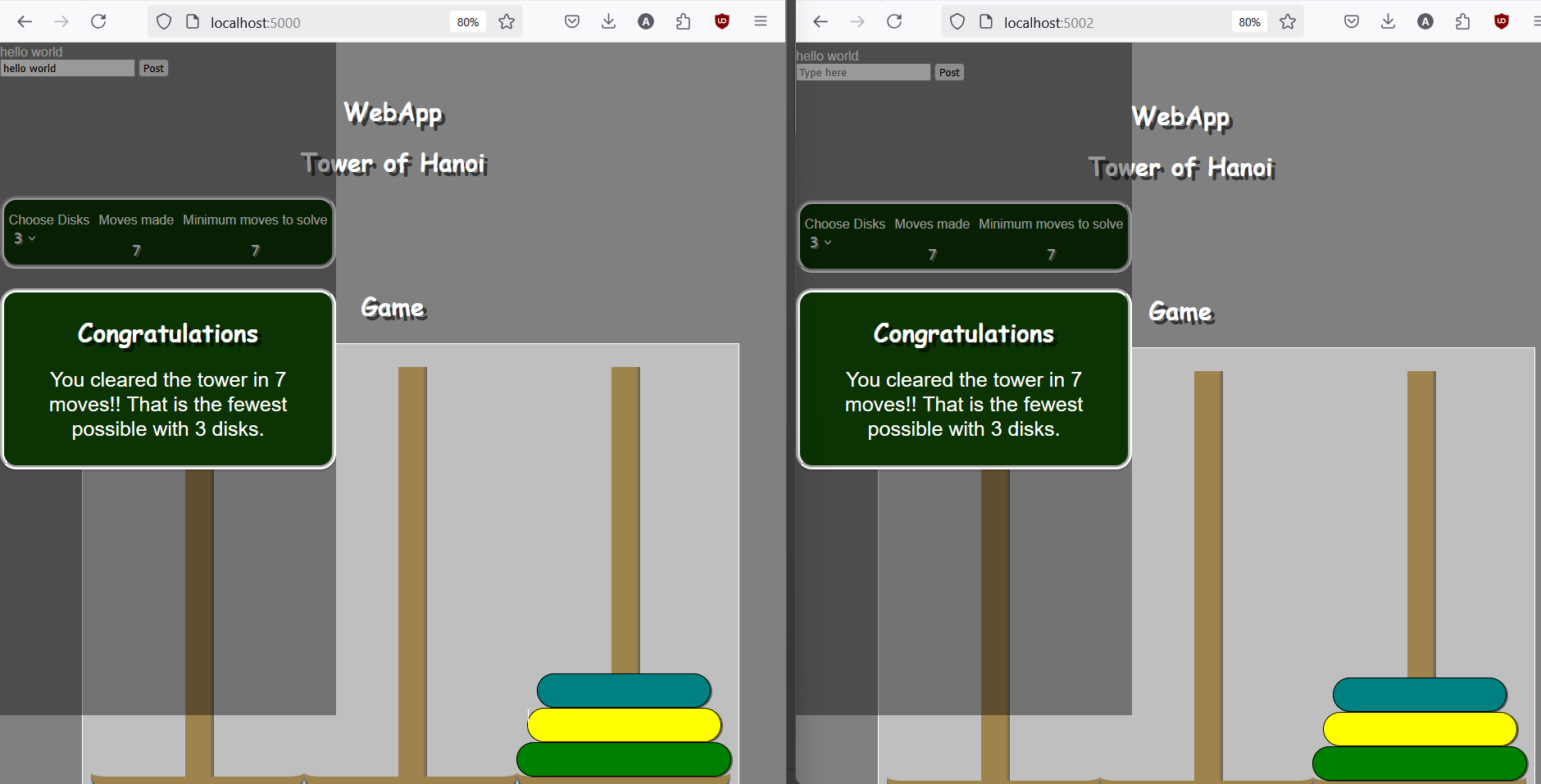
****

Fig: Consistent game clearence

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Inconsistent Game Reset

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Inconsistent disk movement

## Fault Tolerance Test Cases

### FT-1: Inject Delay Before Socket Receive

**Test Name:** test\_simulated\_lag()

**Test Type:** Automatic

**Description:** Simulates 1.5s delay before receiving message via /update.

**Test Scope:** In-Scope

**Expected**: Message still received after delay.

**Actual:** Message received successfully with 1.5s delay.

**Status:** Pass

**Screenshot:**

A screen shot of a computer

AI-generated content may be incorrect.

Fig: Lag Test Output

### FT-2: Reorder Tower Messages in ZMQ

**Test Name:** test\_message\_reorder()

**Test Type**: Automatic

**Description:** Sends multiple messages quickly to test out-of-order delivery.

**Test Scope:** In-Scope

**Expected:** All messages received, regardless of order.

**Actual:** Messages received in different order. Logic preserved.

**Status:** Pass

**Screenshot:**

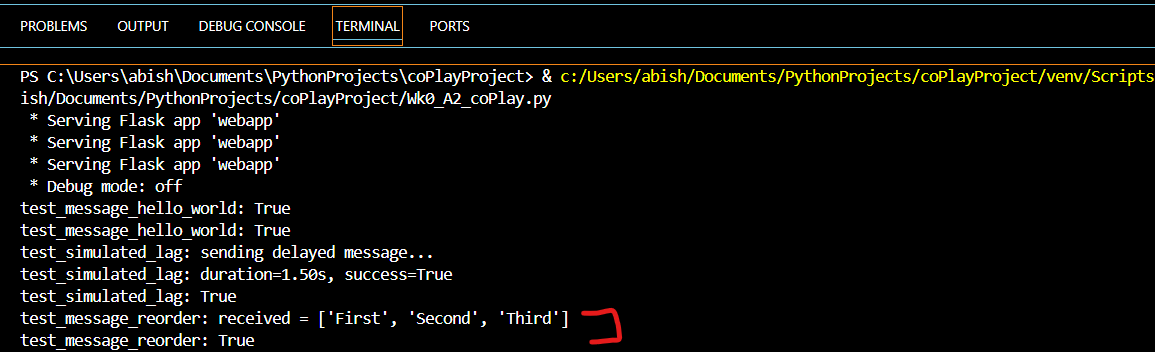


Fig: Message Reorder Test Output

### FT-3: Kill Peer During Active Game

**Test Name:** test\_peer\_crash\_mid\_broadcast()

**Test Type:** Automatic

**Description:** Simulates shutdown of peer during broadcast.

**Scope:** Out-of-scope

**Expected:** Other peers continue functioning.

**Actual:** Runtime error occurred on live peer during update.

**Status:** Fail (expected; crash not handled in prototype)

**Screenshot:**

A screenshot of a computer program

AI-generated content may be incorrect.

Fig: Peer Crash Output

### FT-4: Send Same Message Twice

**Test Name:** Duplicate Message Test

**Test Type:** Manual

**Scope:** Out-of-scope

**Description:** Send identical chat message twice.

**Expected:** Only one copy displayed.

**Actual:** Duplicate shown in chat.

**Status:** Fail (expected; deduplication not implemented in prototype)

**Screenshot:**

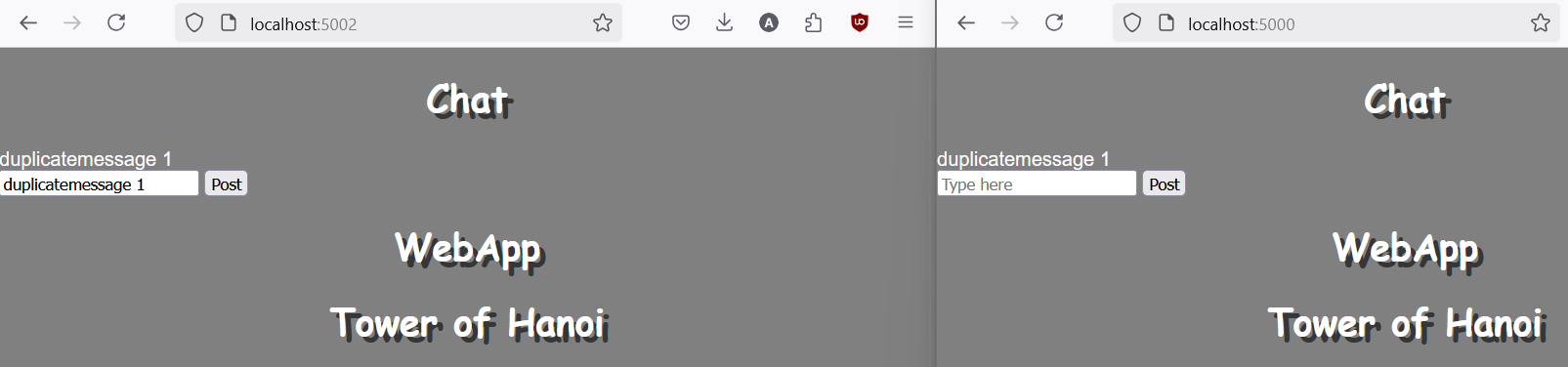


Fig: Duplicate message Test

A screenshot of a computer

AI-generated content may be incorrect.

Fig: Duplicate message Retest

A screen shot of a computer

AI-generated content may be incorrect.

Fig: Duplicate message logs