ixie

Success Story

Performance Testing: Resolving Deadlock in Aggressive Game user actions

Application:
Mobile Game

Services Offered:
Game Play and Performance Testing

Tools:
Game Play and Performance Testing

Key Highlights

Key Success:
Gaming

Duration:
1.5 Months

QA Team:
1 Test Lead
1 Performance Engineer

Technology:
Game is deployed in IBM Cloud
Virtual load is generated in AWS

About the Client

The customer is one of the most innovative Mobile Game Developers in the industry. Their games are focused on Strategy and City Building genres, providing rich simulation experiences that make players stick for a long time. Their games have a large userbase with access across Play store, Appstore, Amazon and Nook.

Application Overview

The application under test is one of the trending genres of Strategy Games. Players can build cities with unique buildings using 120+ items and become real city tycoons as they complete tasks and levels with progressive challenges. The game uses typical gamification and engagement features like reward coins, time targets to finish tasks, resources for purchase etc.

1 Technical and Business Challenges

The client's game development vision was to make the 'user experience' unique and enjoyable. While the beautiful graphics and game themes were a sure-shot attraction to players, any compromise on non-functional metrics would fail the game engagement goals. The gameplay involved users taking instinctive routes to complete levels and explore quests using the featured tools. The combinations of buildings, items, quests and sectors would indicatively seek a custom and robust testing strategy to avoid any performance lags.

The client sought to verify performance of the application with extensive metrics associated with uninterrupted user experience.

» Ensure responsive game launch timing

» Smooth loading throughout the game play

» Game experience for lags, hang ups and crashes due to simultaneous/aggressive usage

» Optimize memory usage during game run

2 Our Approach - Solutions Offered

How iXie put the challenges to rest with a custom test strategy

iXie set out to create a Performance Testing strategy in the backdrop of user experience for the mobile game. With a niche experience in Game QA & Performance Testing, an expert testing team was employed to define most common performance weak-links in a typical mobile game. The problem was handled at two high-level affiliated scenarios:

1. Performance tests on game play
2. Stress Tests with concurrent user access

The game play was tested for 'stability' simulating real scenarios in the following dimensions

» **Game Start Up –** The game response time on the first launch was established to achieve < 2 seconds (Competitive Industry Standards)

» **Mobile resource usage –** Report processor utilization dynamics and memory consumption metrics to guide the development team on resource optimization measures.

» App precedence relations on device – Check for interferences with other applications on the mobile device by switching apps and verifying for game status retrieval. The game includes monetary transactions and spending coins for extra play efficiency; distractions from other apps would result in possible data loss or session injury. Possible interruptions from apps and device conditions were simulated for testing.

» Server and API interaction response times.

» Network performance.

Stress Tests in progressive batches of concurrent users

The key approach defined to perform stress tests was to generate an aggressive user load using a standard performance tool. Indium proposed to use JMeter (opensource tool) to generate load for about 10000 users and simulate real scenarios.

iXie Testing Center modelled a phased approach to achieve the 10000-user load test.

Test Scenarios –

» Game launch by a new user and sustained response/loading time performance for upto 6 levels on the Game Play.

» Free play with extensive Game theme coverage (Use of Tools, Building types …).

» Generate mass virtual load.

» Understand and credit game architecture design for stress test plan. For instance: The game elements buildings, products, etc., hold unique IDs (Dev Conventions) and all the combinations had to be tried in the test scripts.

» MD5 Security Hash Code scenarios.

» Simulated load for 100 concurrent users in AWS Cloud, assessed the performance parameters and reported critical bugs. The game dev team (Client) optimized the game for 100-user-load resistance based on the test recommendations.

» The concurrent user load was ramped up in balanced sprints (100, 500… upto 10000 Users) fixing the performance issues progressively with incremental loads.
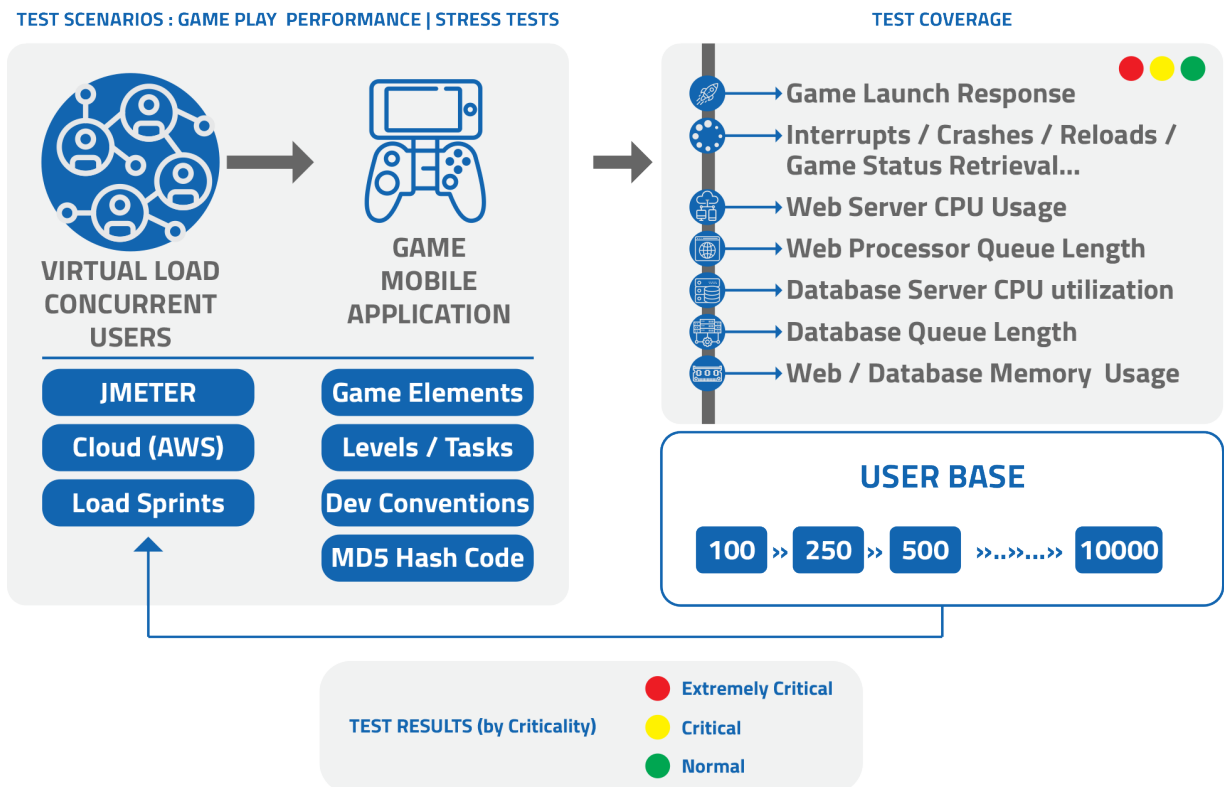
3 Test results

The performance results after a detailed test coverage included defects in different dimensions.

» Response Time Stats (average transaction response times) were recorded with time stamps of Gameplay levels.

» Game App launch cracked 504 errors.

» Crashes at first couple of levels insisted frequent game reloads.

» Reported recurring 'Deadlock' issues after initial levels (4-5 levels) of Game Play.

» Memory and CPU usages hit undesirable high values.

4 Illustrative Test Case

TEST SCENARIOS : GAME PLAY PERFORMANCE | STRESS TESTS

TEST COVERAGE

VIRTUAL LOAD CONCURRENT USERS

→ GAME MOBILE APPLICATION

→ Game Launch Response
→ Interrupts / Crashes / Reloads / Game Status Retrieval...
→ Web Server CPU Usage
→ Web Processor Queue Length
→ Database Server CPU utilization
→ Database Queue Length
→ Web / Database Memory Usage

JMETER | Game Elements
Cloud (AWS) | Levels / Tasks
Load Sprints | Dev Conventions
| MD5 Hash Code

USER BASE

100 » 250 » 500 »..»...» 10000

TEST RESULTS (by Criticality)

● Extremely Critical
● Critical
● Normal

5 The iXie Impact

» Testing as a Performance Compliance function as well as a profit center – iXie's custom approach is a detailed route of app compliance to standard performance metrics.

» In addition, the entire test process was modelled in alignment to game profit objectives: user experience, player engagement, gamification aspects.

» Custom strategy to execute performance tests in app-tolerable ramps.

» Detailed test artifacts with root-cause matrix and recommendations to performance issues.

ixie

---

www.ixiegaming.com │ info@ixiegaming.com