



## EXPLORE: TESTING THE GAME

by KiranKumar Marri & Sundaresasubramanian G

It has been widely reported on internet that the “Angry Birds” game has hit the 250 million download mark and the “Pacman” doodle created by Google to celebrate its 30th anniversary has cost a few million hours in productivity in offices worldwide. There are few games on one end of the gaming spectrum which we have grown up by playing, and there are few on the other side, which are simply irresistible and enjoyable. All the other games in between compete with each other for their share in the gaming market. They target a specific user group, render good graphics/sound and apply excellent marketing strategies along with an interesting theme of play.

With the advent of internet, network gaming has become the new world order. While everyone thought of it as the pinnacle of gaming, user centric gaming came into the scenario (such as X-Box, Wii), where the player has a physically active role in the game instead of controlling a few keys. The days are not far off when the fantasies created by movies (player inside the game, player controlled by game, virtual gaming world) will become a reality.

The critical success factor of a game is its underlying development and testing. The expectations of a gamer have increased substantially so that a simple error can spell doom to the future of the game concerned. Also, the tester is really required to live the game from its user’s perspective to understand and test it well. We will look at some of the nuances of game testing through this article as we explore them with practical scenarios.

### **Functionality is the key!!**

Functionality like any other software testing is the very basic building block of game testing. It checks the basic working of the game, i.e., the controls (mouse/keyboard if it’s a PC game, joystick if it’s a controller based game, remote or sensor for user-centric games). Most of the functionality components (except for brand-new games) are taken off the shelf, and so there is a scope of automation for functionality/regression testing as well.

Functionality testing would mostly follow the development pattern. In some game development environments, the software would be developed in a simulator without the real hardware or controller available. After being thoroughly tested, the software is brought out of simulation and tested in a combination of software and real hardware.

This testing requires the most amount of technical knowledge because it involves monitoring controller network messages and identifying errors, setting up test beds, debugging them, testing the sequence of triggers, etc.

### **Go with the flow!**

Any software game would follow a flow or a pattern or sets of sequences. Testing the flow is of two types: understanding the documented flow and verifying it step by step, or go with the flow based on the tester’s intuition. The advantage with the first method of testing is that it helps in thorough verification of the documented flow (happy path). However, the same point is its disadvantage too; the documented path is limited to a set of actions performed by the user and might not take into account various branches of flow or pattern identified by a gamer in real-world.

The second method of testing, which is intuitive is basically not dependent on documentation and tests the game’s reaction according to the tester’s flow and exploratory concepts. The tester is assumed to be a mid/high level gaming expert in this scenario, and the experience of the tester is really helpful in determining hidden paths (or) cheat codes in a particular flow. However, this type of testing is not exactly repeatable, since the knowledge resides with an individual and not in a document.

The best method is hybrid flow verification – a combination of document based flow testing and intuition based testing. The tester should document the various possible flows occurring in the game and test them thoroughly. In addition to this, an experienced gamer/tester should test the flow or pattern, and all of their inputs should be documented to build future test scenarios.

Some of the basic checks that would be performed in a flow/pattern testing would be to check all levels of the game (including hidden/bonus levels), saving and retrieving a level, soft and hard power reset to test the game’s restorability.

### **Is the content appropriate?**

Content varies from game to game depending on the level of its users. Let us take a scenario where the tester is testing a game called “Blaster-Man” by “XYZ media”, which is a novice adventure game involving the use of knives, guns and bombs. The protagonist of the game has to clear various levels after crossing multiple

obstacles on the way. This game targets the age group of 5-11. As part of content verification, the tester should verify that the game is not too graphic or detail oriented (for instance, a character after getting shot just disappears instead of shedding blood, no obscene languages, no loud sounds, etc.).

Let us take the example of another war game (World War I or II) targeting mid/high level gamers (teens & above). The keyword for content verification here is realism – uniforms used, building layouts, weapon designs, spoken language accuracy (specific to the period in history), vehicle design, facial expressions of the characters involved, historical relevance of the challenges designed as part of the game, etc. The game provides leeway for a certain level of violence and language, but it should still be within 'acceptable' limits.

There are various standards for content verification defined by game manufacturers. Apart from that there are general standards designed for games to be used by general audience, parental guidance or restricted. The tester's responsibility is to verify that the content is appropriate to the target audience. The biggest and the most unexpected challenge that any game might face is IP violation or use of copyrighted/trademark material as part of its flow. The tester's experience and general awareness on IP/copyright laws and material is very essential in ensuring that the content is IP appropriate. For example, placing billboards of real soft drink companies or oil companies in a racing game will look realistic, but might lead to potential law-suits if prior approval or clearance is not obtained from the respective organization.

### ***Play!! Strategize!! Improve!!***

This is the lifeline of any game and the most critical factor in its success. A functionally efficient game with excellent rendering of graphics & sound might still fall flat if the strategy base of the game is weak. This is also the part which requires the most inputs and intuition from the tester. There are linear games where the toughness gradually increases with each level. However, there are strategy based non-linear games which train and equip the player with all basic skills and suddenly throw a series of challenges at them.

Linear games are either low or mid level games – such as racing, novice adventure or sports based. The emphasis for victory is more based on player's control. The game is predictable beyond a point with challenges thrown here and there. The main factor behind these games is to maintain the interest level of the user. Non-linear games are usually of mid-high or high level such as strategy games, war based scenarios, etc. The emphasis for victory is equally split between player's control and their presence of mind. The levels are unpredictable and usually hard to win. The main factor behind these games is kindling the ego of the players, thereby maintaining the interest level.

The total play time consumed to complete a level is a very important metric for testers. While testing the playtime for these games, the tester should consider the average play time for other similar industry standard games and benchmark against them. However, the deviation should be measured and validated against the strategy of the game to decide if it is acceptable or not.

Also, the tester should test that there are sufficient challenges/mini-targets available on the way while completing a level. This is to ensure that the same player when playing and clearing the

same level at different times should get different scores. This can be tested by playing/testing the same level several times and ensuring that the score line differs almost every time.

The most important factor in a game that ensures repeat audience is the 'unexpected' factor which motivates the players to improvise. For example, double jump in a novice adventure game opens a sub-level full of challenges, or crashing at a specific billboard sign in a racing game opens a new shortcut for the racers. Testers should ensure that there are interesting sub levels available in a game to keep the audience interested. At the same time, it shouldn't be an overdose which would dilute the main objective of the game.

There should be a clear differentiation between essential intervals in a game and fillers. Essential intervals are the ones which explain the strategy of the next level to the player once a level is cleared (example: war based games). Any interval that is not necessary for the continuation of the game can be classified as filler. Testers should ensure there is an option to 'skip' the fillers and that the essential intervals are made as interesting as possible.

The strategy of a game is put to its ultimate test during the user testing where a target audience provides pointed comments/ratings on the game based on their prior gaming experience. However, with the techniques described above a tester can add value during the normal testing phase itself thereby uncovering bugs/strategy flaws that typically get uncovered later.

### ***Sound and light show!!***

Graphics and music/sound effects are the two sensory media that help players in associating themselves with the game and hence go a long way in reinforcing their interest in the game.

The primary factor in graphics/video testing is the target audience. For low level games targeting pre-teens and children, the color pattern should be attractive. As the level of the game increases, the focus is more on realistic graphics, since the players at this level will be able to identify and appreciate the game surroundings better. Testers should be adopting a check-list based approach and score the game video based on the game (color pattern LOW/HIGH, usage of bright colors, realistic graphics Yes/No, top view graphics Vs lateral view graphics, intrusion of menus with actual gaming screen, etc.). Another good test will be to check the video resolution against various display systems (monitor, tablet, laptop, computer, televisions of various inches, home theatre, etc.).

Testing music/sound effects gets more complex as the level of the game increases. For low level games, the emphasis is on children or pre-teen friendly music/sound effects and not on realistic sound. However, as the level of the game increases, especially for graphic intensive and interactive games, the lip sync of the voice over with the character needs to be tested. This is true especially in case of movie-based games. Also, the music should not obfuscate the game dialog, and the sounds should be realistic and traceable, i.e., firing of a gun should trigger a gun sound.

In most cases, music or background score is created once the game is fully developed, and so the emphasis will be on the test user group to react to the sound/music and provide feedback.

### ***What is my performance index??***

Most of the games developed today are network friendly, and so

the game should support multi-player scenarios without any issues. The main focus of game testers is to test the performance of the game under various loads. This is not very different from software performance testing where the load (number of users) is increased over a period of time and the game's response is measured. At first, the game's response time is tested/benchmarked using simulators (multiple users/simulators present in the same network node). Then the game's response time is measured in real-time scenarios (various players accessing the game via network). Understanding of network's latency time is essential in determining the actual response time of the game in a real-time scenario.

Apart from testing the response time in a network, the game is also tested for any possible memory leaks, i.e., maintaining the integrity of various players in a game. The endurance of the game can be tested by leaving the game unattended for a long period of time and determining the response by providing a sudden input after a long time. The key in all of these tests is that the game should not hang, and even if it is faced with an unexpected input, it should make a graceful exit from the scenario (for example, re-setting to menu screen instead of hanging).

### Compatibility

It is self-explanatory that the game should be software/hardware/browser agnostic as much as possible, or various versions of game should be available to the user on an 'on-demand' basis. Testers should be testing the game for all the above parameters explained under multiple combinations of operating systems, hardware and browser. A matrix of all possible permutations should be maintained, and test results should be recorded against each cell of the matrix. If repeating all the tests given above is time consuming, at least the functionality, graphics/video and audio tests should be performed to determine the compatibility.

### Other tests

There are a host of other tests that a game tester can normally perform. These are highlighted in the following table:

Games/Type of Testing	Kids	Learning - Fun	Sports	Adventure/ Strategic	Serious
Reference/ Name of Popular Game	Interactive	Black-Berry: Word Mole	18 wheels of steel; Cricket	Prince of Persia: The two Thrones	Virtual Fitness
Feature Testing	High	High	High	High	High
Sequence or Flow Testing	High – Med	High	High	High	High
Content Testing	High – Med	High – Med	High – Med	High	High
Strategy Testing	Low	Low	Med	High-Med	Low
Visual Testing	High	High	High	High	High
Sound/Audio Testing	Med	Med	Med	Med	Med
Error Handling Verification	Low	Low	Med	High	High
Performance Testing	Low	Low	Med	Med	Med
Moves Testing	Low	Med	High	High	High
Usability Testing	High	High	High	High	High
Localization Testing	Low	Low	Low	Med	Med
Abuse Testing	Low	Low	Low	Low	Med
Interoperability Testing	Med	Med	Med	Med	Med
Controller/ joystick Testing	Low	Low	Low	Low	Low
Compliance Testing	Low	Low	Low	Low	Low
Compatibility Testing	Med	Med	Med	Med	Med

Type of testing	Comments
Localization	Employing localized testers especially in case of cross-continent or cross-cultural games
Abuse testing	Multiple incoherent inputs through controller and determine game's performance (graceful exit from the scenario expected at the minimum)
Error handling	Testing for all error scenarios in the game to check if error messages are triggered
Independent testing of controller/ joystick	Testing if the buttons generate system specified levels of signals (either measured as impedance or signal strength)
Regression testing	Focused/targeted testing around a fixed bug to ensure the unaffected functionality of the game is not broken by the fix
Compliance testing	Any compliance enforced by the game manufacturer, customer or important stakeholder

A game tester should have the quality to ExPLORE the game in order to test it properly:

- **Experience** the game as a player first
- **Progressive thinking**
- **Logical ability**
- **Observant**
- **Reasoning Ability**
- **Exhaustive verification** of features, functionality & game strategy

Practical Example – Authors' perspective of the extent of testing required for some of the games available in the market:

References: Infosys sources

## > biography



### Kiran Marri

BE MS, is currently working as a Delivery Manager at Infosys Limited, Bangalore (NASDAQ: Infy [www.infosys.com](http://www.infosys.com)). He has over 15 years of IT experience in project management, client relations and developer roles. He has published and presented several papers at conferences in the field of project management, software testing, clinical data management and biomedical engineering. He has

also conducted workshops, tutorials on creativity, thought leadership, risk management, test management and defect prediction methods. His current research interest and publications are primarily in specialized testing, test maturity practices and innovation strategies. He received his Bachelors in Electronics & Communication engineering from Madras University in 1993 and a Masters by Research in Biomedical Engineering from Indian Institute of Technology Madras, Chennai in 1996. Kiran is also PMP certified. Kiran Marri can be contacted at [ki-rankmr@infosys.com](mailto:ki-rankmr@infosys.com)



### Sundaresasubramanian

has about 10 years of experience in software testing, embedded systems testing and digital signal processing research & development. Currently, he is a Senior Project Manager with Infosys Technology Limited (NASDAQ: Infy) and part of the Independent Validation Solutions business unit. Before Infosys, he was involved in strategizing and delivery of testing projects in the automotive multi-

media domain, and prior to that he was involved in research and development of digital signal processor-based embedded applications. He believes that excellence can be achieved by the eagerness to learn new things and the attitude shown in implementing them at work. He earned his Bachelor of Engineering degree from Manonmanium Sundaranar University, India. He can be reached at [ssubramanian\\_gv@infosys.com](mailto:ssubramanian_gv@infosys.com)



[people.steria-mummert.de](http://people.steria-mummert.de)

Faszination: Consulting

Leidenschaft: Testing

Perspektive: Andersdenkermacher



➔ Weitere Infos

Wir zählen zu den zehn führenden Management- und IT-Beratungen im deutschen Markt. Was uns von allen anderen unterscheidet: wir beraten nicht nur, sondern setzen die Ideen auch in die Tat um und begleiten den Kunden während des gesamten Veränderungsprozesses.

Consultant / Senior Consultant / Principal Consultant

## Zertifizierter Software Test Manager (m/w)

### Anders denken

Bei Steria Mummert Consulting haben Sie deutlich mehr Freiheiten als anderswo. Wir setzen großes Vertrauen in Sie und lassen Sie von Beginn an selbstständig arbeiten. Sie dürfen Ihren eigenen Kopf mitbringen und – noch viel wichtiger – Sie dürfen ihn auch behalten. Mitsamt Ihrem Wissen, Ihren Erfahrungen und Ihren Ideen.

### Anders machen

Als Software Test Manager unterstützen Sie unsere Kunden in der operativen Konzeption und Steuerung von Testprojekten in heterogenen IT-Landschaften. Sie agieren selbstständig im Dialog mit dem Kunden und lenken Ihr Team erfolgreich durch die Testphasen. Dabei übernehmen Sie eine Teilverantwortung für die erfolgreiche Akquise aus den Projekten heraus.

### Anders sein

Teamfähigkeit, gesunder Pragmatismus, Engagement und viel Verantwortungsbewusstsein ermöglichen es Ihnen, Ziele schnell zu erreichen. Da Sie als Berater oft beim Kunden vor Ort sein werden, ist die Reisebereitschaft innerhalb Deutschlands eine Selbstverständlichkeit für Sie. Darüber hinaus verfügen Sie über ein erfolgreich abgeschlossenes Studium oder eine vergleichbare Ausbildung, mehrjährige Erfahrung im operativen Test Management, Kenntnisse von Standardvorgehen wie TMap, Zertifizierungen nach ISTQB Advanced Level oder vergleichbare Zertifikate und tiefgehende Erfahrungen im Projektmanagement.

Ihr Einsatzort: Bei unseren Kunden im gesamten Bundesgebiet.

Für Fragen zu dieser Anzeige steht Ihnen Claas Bendels telefonisch unter 040 / 22703-7608 oder -7680 gerne zur Verfügung.

**Jetzt online bewerben.** Bewerben Sie sich jetzt mit Angabe Ihrer Gehaltsvorstellung unter [people@steria-mummert.de](mailto:people@steria-mummert.de) oder mit dem Bewerbungsformular auf [www.people.steria-mummert.de](http://www.people.steria-mummert.de)



Die Andersdenkermacher.