Experience Requirements in Video Games

Definition and Testability

David Callele, Philip Dueck
Experience First Design Inc.
Saskatoon, Canada
callele@cs.usask.ca

Krzysztof Wnuk[2], Peitsa Hynninen[1]
[1]Aalto University Espoo, Finland
peitsa.hynninen@aalto.fi
[2]Blekinge Institue of Technology
Karlskrona, Sweden
Krzysztof.Wnuk@bth.se

***Abstract*—A properly formed requirement is testable, a necessity for ensuring that design goals are met. While challenging in productivity applications, entertainment applications such as games compound the problem due to their subjective nature. We report here on our efforts to create testable experience requirements, the associated scope challenges and challenges with test design and result interpretation. We further report on issues experienced when performing focus group testing and provide practitioner guidance.**

***Index Terms*—Experience requirements, testing, validation, design goal, user experience.**

I. Introduction

Defining gameplay is simultaneously straightforward and challenging [2]. A simple statement like "I want to build a game in the style of Flappy Bird" immediately communicates a wide range of requirements such as a very simple user interface (single tap is the only user interaction with the game), very simple rules (if the player avatar contacts any visible world element then the player must restart) and many others (such as hedonic qualities like "fun") that are inferred by just playing the game. Entertainment applications diverge from productivity applications in the design of the "fun" aspect. Productivity applications are designed to accomplish one or more tasks as necessary elements of a non-entertainment endeavor (*e.g.* business endeavors). Entertainment software is usually deliberately designed to foster a long-term learning curve to promote replayability); it is difficult to imagine productivity software targeting anything but achieving competency as soon as possible.

Unfortunately, a simple statement like the above example does not capture many other critical requirements that are necessarily part of the design and development process. The sentence fragment "in the style of" expresses at least some of the stakeholder's *wants* (I want a Flappy Bird clone) but not necessarily any of their *needs* (I need the Flappy Bird clone to be a commercial success). For example, the statement identifies similarities but the statement is silent about the differentiators: What makes the new game different from the original game?

Emotional requirements [3] (requirements that capture the intent and the means by which a designer expects to induce an emotional state in the player) and experience requirements [1] (descriptions of user, player, and customer experiences that must be met (functional experiences) or are satisfaction goals (nonfunctional experiences), for products or services) were proposed in an effort to address some of the challenges associated with expressing these requirements. Briefly, these are techniques and guidance for expressing the intended playing experience as emotional, gameplay (cognitive, mechanical) and sensory (visual, auditory, haptic) requirements. As such, they represent a subset of the requirements that could be generated across all aspects of user experience – defined in ISO 9421-210 [5] as including "all the users' emotions, beliefs, preferences, perceptions, physical and psychological responses, behaviors and accomplishments that occur before, during and after use."

In this work we focus on expressing gameplay design elements as *cognitive requirements* (the head), *mechanical requirements* (the hands), and visual *sensory requirements* (include visual, auditory and haptic senses) [2]. These requirements interact to define the gameplay experience and it is our goal to verify that we can capture these requirements and to validate that the implementation delivers the intended user experience.

This paper reports the results of experimental efforts directed towards creating testable user experience requirements and associated challenges in experimental design. In other words, we attempted to define requirements for player experiences that were to be induced by the videogame and then design tests to determine whether the implementation satisfies these requirements. . While requirements verification typically encompasses aspects such as inspection, demonstration, evaluation and test, we constrained our investigation to only those aspects necessary to facilitate experimental design for testing within this domain. We plan to address more of the aspects of verification in future work. The experiments were performed using Bachelor students in Computer Science who played an endless runner game in a controlled environment.

The remainder of this work is as follows. In Section III we present experience requirements within the study context and Section IV presents practitioner guidance for converting experience requirements into testable functional requirements. Section V addresses experience requirements testing, the specific requirement that was tested and how the test design and implementation evolved as we gained practical experience. The test results are presented in Section VI. Section VII discusses

978-1-4673-6905-3/15 © 2015 IEEE

RE 2015, Ottawa, ON, Canada
Industry Paper

the overall investigation and provides practitioner guidance. Section VIII concludes this work and provides directions for future work.

II. Related Work

The current work builds on prior work in this area. Callele *et al.* [3] introduced emotional requirements as a technique for capturing the game designer's intentions for the emotional state induced in the player. In other work, Callele *et al.* [2] introduced cognitive requirements, derived from observations of a work-for-hire proposal, to help capture the requirements for cognitive challenges in games. Finally, a more holistic view of this class of requirements, presenting a stimulus-perception-response model for gameplay and extending emotional and cognitive requirements to include mechanical and sensory requirements is presented in [1].

Daneva [4] investigated gameplay requirements in massive multiplayer online role-playing games (MMOGs) via practitioner interviews, noting that "'paper-prototyping' and playtesting are pivotal to gameplay validation" illustrating that practitioners have adopted a very loose validation and feedback loop without rigorous Deign Of Experiment (DOE) based testing. She further notes that "balancing the elements of the gameplay is an on-going task, perceived as the most difficult and labor-consuming" which is consistent with our experience in attempting to use experience requirements validation to the same purpose.

Work on user experience and requirements engineering shares some perspectives with this work. Lee *et al.* [5] analyzed the role of the *specialist* in Agile-UXD, a position that comprises some of the responsibilities of each of the game designer, producer and director of a game development team. They propose the use of a standardized notation to help bridge the gap between user experience and feature implementation – a role analogous to that played by experience requirements in game development. Loeffler [6] investigated the desire for intuitive interfaces (as expected in videogames) and addressed resource shortfalls through the use of image schemas and image-schematic metaphors; a technique that parallels our work in emotional requirement representations.

Testing implementations and validating requirements is always challenging. Davis [7] investigated aspects of user experience validation and showed that, when used appropriately, "preprototypes" such as paper-based prototypes can be used much earlier than software prototypes for user acceptance testing. Cost-effectiveness when a change occurs is significantly greater than within a software prototype. Within the game industry, storyboards and gameplay prototyping with minimal graphics are similar project management constructs. Assessing compliance for soft goals such as usability is challenging. Primrose [10] and Stroe [11] used Kano categories to assign usability grades to requirements, features and implementations. The current work faced similar challenges but chose to use the player scoring system to ensure that the interests of the testers and the players were aligned.

Experimental methodologies and design of experiment guidance is always welcome. Wohlin *et. al.* [12] provides guidance for many scenarios but does not provide significant guidance for multi-dimensional non-functional scenarios such as validating the intended user experience. Most of the guidance assumes homogeneous populations but the game domain dis plays significant heterogeneity of subject, platforms and performance. While [12] identifies learning as a threat to validity, it does not provide explicit guidance to either minimize or explicitly measure learning effects. Learning effects are a significant threat in the current work and there is significant opportunity for future work in this area [13].

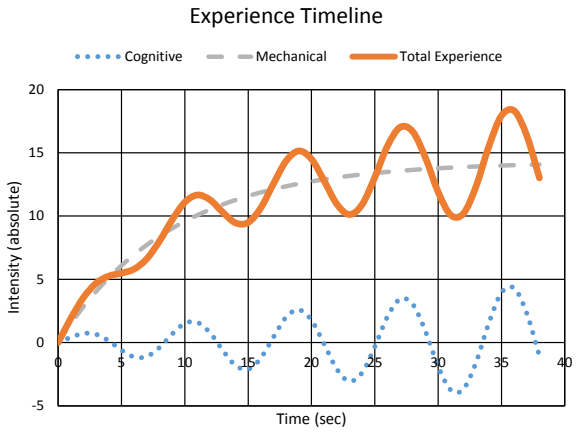

Fig. 1 Example experience timeline

Videogames are high dimension spaces and addressing confounding factors during design of experiment efforts is challenging. One of the aspects identified in our prior work [13], calibrating for *a priori* user capabilities, was a significant confounding factor in the current work. We addressed some of the confounding factors using a form of A/B testing [9].

III. Experience Requirements

Experience requirements are descriptions of user, player, and customer experiences that must be met (functional experiences) or are satisfaction goals (nonfunctional experiences), for products or services [13]. In this context, experience requirements are intended to capture the game designers "look and feel" description of the gameplay experience and to translate that description into requirements that can be used by the production team to deliver that intended experience.

A. A Simple Game Example

The *endless runner* genre is one of the simpler schemes to define. Forward motion within the game world is outside of the players control and the player must take deliberate action to either avoid threats or to intersect with rewards. Visualizations are often two-dimensional but even if they are three-dimensional player motion is usually constrained to 2D – for example, jumping up, staying neutral (running), crouching, crawling, *etc.* Games of this genre are mathematically modeled as a maze with temporal dependencies.

The gameplay experience in endless runner games is a sequence of challenges requiring the following efforts:

1. Recognition (of clues, threats and rewards)

2. Identifying possible solution strategies
3. Choosing a solution strategy
4. Executing the solution strategy

The intensity of the experience grows over time, usually requiring more complex puzzle solving and/or ever quicker reactions.

B. Defining the Experience

As mentioned in Section I, there are three distinct dimensions for which requirements must be generated for this example: the cognitive challenge, the mechanical (user action) challenge and the visual sensory challenge. We focus here on the cognitive and mechanical challenges for the visual aspects are more properly the domain of the art department. Figure 1 is an *experience timeline* wherein the intensity of the cognitive and mechanical components are abstracted to an intensity value that indicates the relative magnitude of the player experience. These two dimensions interact as shown, where the gameplay intensity is the sum of the intensities of the cognitive and mechanical challenges (other transformations are possible and reasonable, linear summation is assumed for illustrative purposes).

The dotted blue line in Figure 1 represents the intensity of the cognitive challenges over time. As shown, the challenges appear regularly but their intensity grows (for example, the required accuracy increases) as the player stays alive within the game. In a side-scrolling game where the player must dodge incoming obstacles, the peaks represent when the player avatar is coincident with the obstacle (the point of maximum danger), the decreasing slope represents successfully evading the obstacle and the trough represents a 'relaxation point' just before the next obstacle becomes visible. As the obstacle comes nearer, the intensity increases (increasing slope).

The dashed gray line in Figure 1 represents the more general dimension of mechanical difficulty. In the example, it might represent the apparent velocity of the player within the world: the apparent velocity starts off slowly but grows rapidly for the first 15 seconds before tapering off.

The orange line in Figure 1 represents the combination of these dimensions. The player perceives that their velocity within the world is rapidly growing but the challenges are not very difficult to begin with – the player is expected to learn how to control their avatar during this period and to become accustomed to "how fast" they are moving. As the rate of velocity change reduces, the player is presented with increasingly difficult cognitive challenges that they must overcome.

It is important to note the interaction between these dimensions: player focus is initially upon the environment and basic skill acquisition but is then transitioned to focus upon skill refinement within a relatively constant environment. This combination of interactions between the dimensions provides the player with the opportunity to learn how to perform within the environment and, by extension, to achieve ever-growing degrees of success as they learn how to survive longer within the game environment.

The guidance provided by Figure 1 must now be converted into a gameplay experience. This conversion includes the artistic aspects (the *look*) and the game mechanics (the *feel*). In this work, we are concentrating upon the game mechanics, the associated experience requirements and how they translate to testable functional requirements and our efforts to validate these requirements.

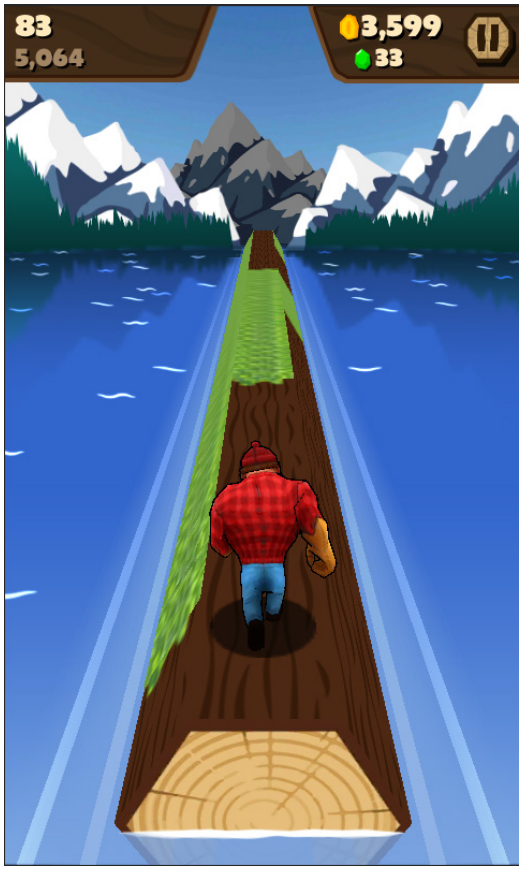

Fig. 2 Gameplay test platform

IV. Experience Concept to Experience Realization

The game development company decided to design a game in the style of *Temple Run* and *Subway Surfer*. These games are three-lane, endless runners where the player must choose to move to the left, right, up or down to dodge oncoming obstacles or to intersect oncoming rewards.

The resulting game, *Lumberjack Run*, differentiates itself via its *visibility mechanic:* how the player perceives the world. *Lumberjack Run* is a six-lane maze where three of the lanes are clearly visible at all times, two of the lanes are partially visible (to provide hints) and the remaining lane is always invisible at a given instant. The visibility mechanic was achieved by placing the character on a log with a hexagonal cross-section where three facets were visible at all times and two facets were submerged but were dimly visible. The final, invisible facet is the opposite side of the log from that upon which the player is positioned.

A sample image from the final game and test platform is shown in Figure 2.

The mechanical challenge (Figure 3) is a strictly linear process wherein the forward speed upon the sequence of logs is constantly increasing. The associated cognitive challenges are

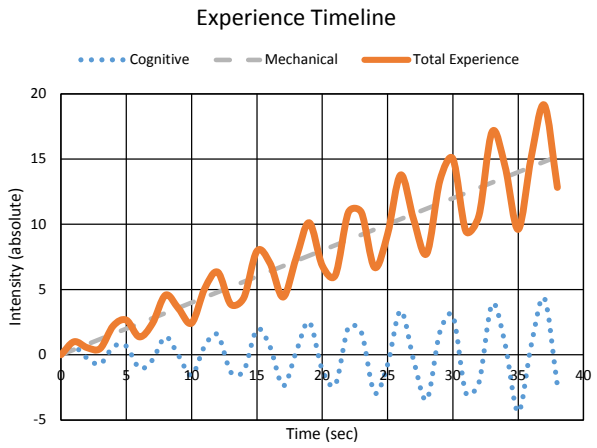

Fig.3 Gameplay test experience

presented more frequently than those in Figure 1 and are drawn from the available gameplay (maze) patterns via a random sampling of overlapping distributions of low, medium and high difficulty challenges.

The player is able to jump up from the log with a vertical swipe motion but can abort their jump with a downward swipe motion. The player can transition to the facet to the left by swiping left (which rolls the log to the right) or transition to the facet to the right by swiping right. The challenges are further complicated by segmenting the lanes into individual logs and the player is required to jump from log to log or fall off into the surrounding water.

The challenge patterns, as described above, are defined as three lane patterns or six lane patterns with associated low, medium and high difficulty levels. Each pattern can be repeated from 1 to N times and each instance of the pattern can be placed as defined or it can be mirrored along either or both of the X and Y axes. Three lane patterns can also be copied (with one or more of the described transformations then applied) to convert them into six lane patterns. These patterns are placed in the game world in a quasi-random manner where a family of heuristics is applied to the output of a random number generator before choosing the next challenge pattern and the manner in which it is instantiated.

V. Translating Experience Requirements to Testable Functional Requirements

Experience requirements must be translated into functional requirements before they can be tested. For example, the experience requirement: *The player must remain on a clear path by dodging all oncoming obstacles* can be translated into a series of explicit Functional Requirements (FRs, such as the following example requirements):

FR1. The game must support left, right, up and down actions.

FR2. The user interface for the game must provide a means for the player to indicate that they need a specific action to occur.

FR3. The response time for the user interface must meet the (game designer's) specifications.

There are also implicit functional requirements (which may also be presented as constraints). Examples include:

FR4. It must be possible for the player to successfully jump from log to log. *As constraints:* Obstacles must not be larger than the distance that the player can jump in the game. Further, the maximum length of an obstacle must include a margin *(that needs to be specified)* for error in player timing.

FR5. There exists at least one sequence of player actions that leads to player success for a given challenge.

Lumberjack Run implements the experience timeline shown in Figure 3. The dominant Experience Requirement (ER) is stated as follows:

> ***ER1****: The intended player experience shall be induced in the player and the difficulty of the cognitive challenges is directly correlated with the difficulty of the mechanical challenges.*

In other words, high difficulty cognitive challenges are only placed later in a given play session, once player speed has also substantially increased.

This experience requirement was expressed by the following functional requirements.

FR6. Player speed (perceived forward motion) must increase linearly with time in-game
- a. There is no upper-bound on player speed
- b. Player speed is zero at the start of every run

FR7. Cognitive challenge difficulty must increase with time in-game.

FR8. Mechanical challenge difficulty must directly correlate with cognitive challenge difficulty.

We note that requirements like FR7 and FR8 contain the ambiguous terms "must increase with time" and "must directly correlate." RE practice generally requires that these terms are disambiguated before proceeding. However, the game design context is quite different: the game designer is expressing a concept that is intended to induce an emotional or other experiential response. These induced responses are the targets of the noted requirements and, while these requirements could be disambiguated with qualifiers, these qualifiers would likely only apply to a specific market segment (*e.g.* perceived "cognitive challenge difficulty" could vary greatly depending upon age, game play skills and game play experience).

In practice, a game may de conceived and designed to target a given market segment but it is common for the final target market segment to differ from the original design target. Therefore, there is significant reluctance to try to be precise at the time that requirements could be generated – it is perceived as just too difficult to predict what will be perceived as "fun".

VI. Testing Experience Requirements

Given the functional requirements as expressed in the prior section, we began to develop our test plan. Testing options were greatly constrained by the requirement that the testing must be performed within the actual in-game environment – it was neither feasible nor practical to develop an independent testing platform. However, as we began to explore options for testing, we realized that substantial effort would be required to modify the existing game engine for this type of testing (essentially converting the game engine into a research platform). For example, the algorithms for challenge placement had to be suppressed and new testing algorithms had to be created. These new algorithms had their own set of requirements, implementations and testing and we significantly underestimated the required development effort. We also had to develop remote data acquisition, a centralized collection infrastructure and data analysis procedures.

The functional requirements presented in Section IV were insufficiently precise to allow us to define test regimes. The greatest challenge that we faced was with confounding factors: identifying the possible confounding factors is challenging and attempting to control for those confounding factors was even more challenging.

Player (gameplay) skill is a significant confounding factor. For example, how do we derive meaningful results (especially results that support comparison of perceived difficulty) when we do not have any way of measuring, on an absolute scale, one player's *a priori* skill compared to another? Guidance from the design of experiments literature suggests that the players self-assess and report upon their skill levels. However, we have no way to know whether this self-assessment has any validity – for example, what bias does vanity express in the assessment? Alternatively we could recruit a large number of test participants in an effort to create a statistically valid population to give us greater confidence in the results but the associated recruitment and retention costs are prohibitive. There are numerous sub-populations (*e.g.* children, causal gamers, experienced gamers, male, female, *etc.*) that define the target market segments and acquiring an independent population for each market segment would require recruiting hundreds of test participants.

Dexterity and hand-eye coordination are also significant confounding factors. Given that this is a game where reaction time is important, is it possible to gather meaningful interpretation of difficulty when we have not calibrated for or compensated for user dexterity?

Finally, we are attempting to assess difficulty. In this game, difficulty is an interaction between a cognitive challenge and a mechanical challenge. Therefore, we are attempting to map interactions in two dimensions into a single scalar value – of necessity, some information will be lost. As a result, it is possible that we have met the stated requirements in one dimension but not the other dimension and there may not be any way to identify which dimension is failing to pass the test.

Fig. 4 Possible solution paths

A. Metrics

Metrics and measurement are critical components of testing. We have already noted a weakness in our ability to measure the requirements as stated. Unfortunately, natural language definitions do not always render easy-to-understand definitions for metrics. When we express the requirement in terms of difficulty it "feels right" but what does this really mean?

Is difficulty related to some measurement of perceived success such as score? If it is then we suffer from the same loss of information as noted above. However, if we represent difficulty as a function of two or more dimensions than we are faced with communication and comprehension challenges with audiences that do not feel comfortable communicating in this manner.

From the player's perspective, what is difficulty? The game developer uses an internal (working) definition to guide their work: "difficulty is positively correlated with joyful stress and negatively correlated with score". In other words, as a player, we enjoy the stress that is placed upon us by the challenges imposed by the game and feel a sense of reward and a sense of self-validation when we overcome the presented difficulty. If our evaluation (the score) goes up slowly despite our best efforts then the game is difficult.

When we keep in mind that the game exists to satisfy the player then it becomes clear that difficulty must be measured in the same manner as the player receives feedback. This is, by tradition, a scalar value such as a score for that run but it can

also contain other attributes such as rewards gathered along the way, special achievements, *etc*. As a result, our experimental methodologies are severely constrained.

B. Research Questions

A brainstorming session generated candidate research questions for the test program, research questions whose answers could guide further refinement of this game and guide future development efforts. These questions are summarized here, informally grouped by topic rather than as discrete questions.

RQ1. What differences were there between player performance on the challenges when performed at a slower speed? When performed at a higher speed?

RQ2. Are there any observations that can be drawn from the histogram of death distances for each challenge? Are these observations different for the speed at which the test was run? Are there differences in the histograms for players who ran slow tests then fast tests compared to fast tests then slow tests?

RQ3. Is there evidence of a learning effect, did the players get better over time? Is the learning effect constrained to a single pattern? More than one pattern? All patterns?

There were many questions to which we wanted answers. We were particularly concerned over whether or not we could accumulate enough data, from enough runs, to provide acceptable confidence in the statistical validity of our results. Even though we recruited 30 participants, only 14 participants completed the entire test regime. Upon follow-up with those participants that did not complete the test, we were informed that the participants either became bored and quit or became tired and quit. Given that the participants were all volunteers we had no recourse but to accept this limitation to our study.

C. Test Protocol

The original definition of the experience intensity was presented in Figure 3; the definition called for a linear increase in the mechanical difficulty combined with a cyclic presentation of challenges of ever-increasing cognitive difficulty. We performed quantitative testing only and addressed the challenge of measuring two dimensions with a single value by partitioning the tests themselves in two dimensions.

Mechanical difficulty was assessed by testing a given maze pattern at two speeds: as if the player was at the start position and again at the game speed as if the player was already at the 800 meter mark within the game.

Cognitive difficulty was assessed by presenting each player with six different maze patterns, allowing us to compare their relative success across these patterns. All maze patterns were drawn from the high difficulty group (as designated by the game designer) – in the then-current game design these patterns were only placed in the world once the player had traveled a significant distance down the logs. Figure 4 is an example image taken from the in-game test. This figure shows that there are (at least) three visible solution paths to the challenge presented to the player. If the player is traveling at high speed then they are able to jump over the green slime in their current lane. If the player is traveling at low speed then the player must shift one lane to the right then jump over the green slime *or* the player can shift two lanes to the right and go around the green slime without jumping. The path without jumping only becomes completely visible once the player has shifted one lane to the right.

Finally, the test group was partitioned into two subgroups, A and B. Group A was presented with each maze pattern as if they were at the start position and then again as if they were starting at 800 m down the log sequence. Group B was presented the maze patterns as if they were starting at 800 m and then again as if they were starting at the beginning.

Each run can require from approximately 5 seconds to 2 min. to complete (it is extremely unlikely that any player will last beyond 2 minutes on a given run). If the average run time is approximately 30 seconds and a test session is targeted at one hour duration (or less) then each participant should be able to complete at least 60 runs. The target test session duration constraint (60 minutes) left a budget of five runs per challenge (maze) at each of the starting conditions.

Thirty students and personal contacts were recruited for the test. Every participant was provided with a copy of the game via our beta-test infrastructure. After installing the game, each participant was required to participate in the mandatory in-game tutorial. Once the tutorial was complete, each participant was required to perform 10 practice runs within the standard game to provide them with greater familiarity with the look and feel of the game. After 10 practice runs were completed the participant was assigned to one of the two groups (A: slow first then fast or B: fast first then slow). Each participant then performed five runs through each of the six mazes at their initial assigned speed followed by a further five runs through each of the six mazes at the alternate speed (60 runs in total).

VII. TEST RESULTS

The test results that we were able to gather are presented in Figures 5, 6 and 7.

The testers almost always performed better or significantly better at slow speeds rather than high speeds (Figure 5). The results for Maze 1 are mixed, however, and the differences are less pronounced for Maze 5 and Maze 6.

The histograms of death distances for each pattern, and for each test group A and B, are depicted in Figure 6. This figure similarly shows that the differences are less pronounced in mazes 5 and 6.

Maze 6 shows some learning effect at the slower speeds (Figure 7). A small learning effect may be occurring in Maze 4 at higher speeds, especially in the last run.

Given the challenges described elsewhere in this document, we were unable to gather sufficient data to properly guide our design efforts. Effort is ongoing to recruit more testers so that we can receive the desired difficulty assessment information and apply that knowledge to our design efforts.

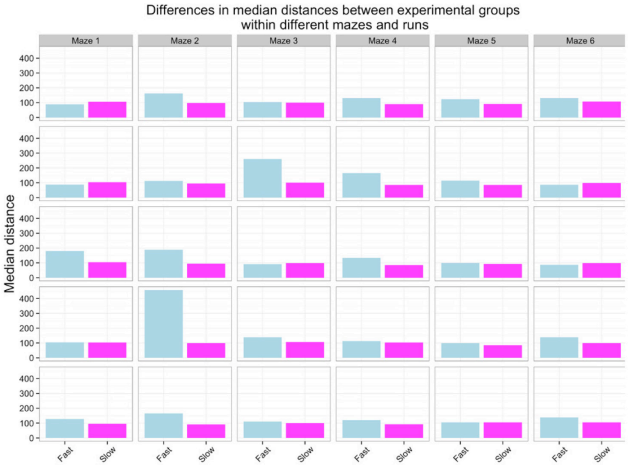

Fig. 5 Median Distance (RQ1)

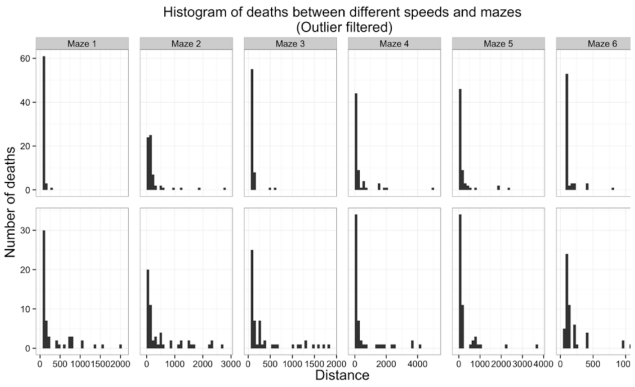

Fig. 6. Death Histogram (RQ2)

VIII. Discussion and Practitioner Guidance

Our experience illustrates that it is possible to evolve an experience requirement to a set of (seemingly) testable functional requirements. However, practically validating the requirements in a system as complex as a videogame is a daunting effort. In the remainder of this section we briefly summarize what we learned and attempt to identify specific challenges that practitioners may face in their own efforts.

Game design: Game design may appear to be an *ad hoc* effort compared to productivity software design and this perspective may be justified in many cases. However, it is possible to be very deliberate in many aspects of the design effort, particularly in the game mechanics as we have explored in this work. We have found it particularly useful to develop a mathematical abstraction of the user experience for supporting our design of experiment efforts. This abstraction also helps the game designer understand what their game really "is" and can help to maintain consistency throughout the gameplay experience. Without this understanding of the core concept(s) then the team is not performing deliberate design; they are essentially relying upon getting lucky that the emergent gameplay will satisfy their players.

Experience requirements: Experience requirements work well to convey substantial information in a very compact manner, information that can be challenging to verbalize. Converting experience requirements into testable functional requirements can be difficult (especially when referring to induced emotional states) and can lead to a much larger set of requirements than anticipated. Fortunately, these resulting functional requirements can be interpreted within the context of the experience requirement which increases the probability of successful communication. Unfortunately, as shown here, the validation process for these functional requirements may be very challenging to implement.

Dimensional analysis: Performing a detailed dimensional analysis of the gameplay environment can be very helpful before attempting to design the experiments. While some of the dimensions are immediately apparent upon a visual examination of the game, we further identified many more dimensions during a code review process. During the code review, we learned that attention must be paid to factors that are time variant, imply a player learning curve or (most importantly) an interaction between them. These elements may lead to significant confounding factor challenges that are difficult to address. In this work we needed to double the number of test cases in an attempt to account for a single interaction.

Experimental design: It is important to understand the dimensional complexity and apply appropriate design of experiment techniques to compensate. In the current work, many of the desired answers could not be obtained within our operational budget and the scope of the planned test program had to be severely reduced.

One issue that we noted that has the potential to complicate experimental design is short-term memory bias during gameplay. Players learn patterns and there is a strong tendency to continue executing a prior successful pattern – for example, if a player is successful with a sequence of operations such as Left-Left-Left-Left (LLLL) and the next pattern needed for success is Right-Left-Right-Left (RLRL) then the pattern (LLLL-RLRL) will have an inherent bias toward failure compared to (RRRR -RLRL).

***Combinatorial complexity*:** The user experience design goals for a game could be significantly more complex than the user experience design goals in productivity software. For example, in-game lighting, music, sound effects, artwork, modeling and animation all contribute to the gameplay user experience. In contrast, we posit that productivity software generally attempts to ensure that these elements are not present as they are considered a distraction to "getting the job done" – which may or may not be an appropriate decision but one which is outside of the scope of this work.

The described additional complexity can lead to a combinatorial explosion of affecting and confounding factors making test design and results interpretation difficult – particularly when causation must be identified rather than correlation [8]. Practitioners may expect that substantial changes to the game engine may be required to accommodate the necessary data acquisition.

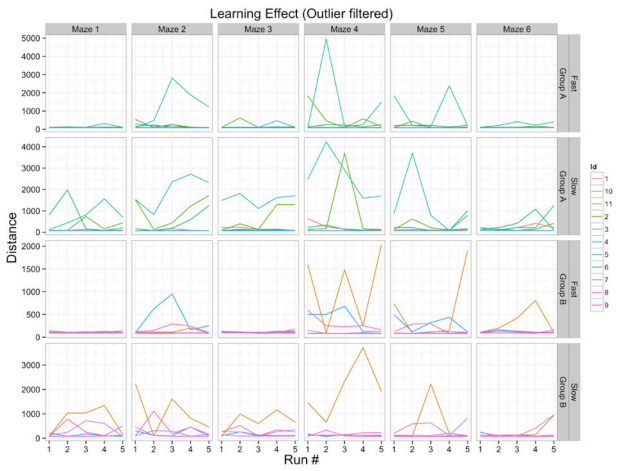

Fig. 7 Learning Effects (RQ3)

Confounding Factors: As noted in prior examples in this section, it is difficult to identify and assess the effects of the confounding factors. In the current work, confounding factors related to human-computer (human-game) interaction were challenging: prior experience, learning effects, exhaustion and boredom. Learning effects are significant: increasing success within a game implies constant learning. When attempting to assess difficulty, the results must always be interpreted in the context of the player's current skill level; a skill level that is expected to continually evolve. And, if there are multiple patterns, does the fact that the subject has experienced one pattern affect the results for the later pattern(s)? Finally, these games require focus and if the subject tires or loses focus then the results can be dramatically affected. These effects can occur within tests of a single pattern but they may also be cumulative over the course of a test sequence.

We note that these same confounding factors exist in productivity software but caution that there can be significant divergence in design intent: Productivity software is typically designed to promote mastery as quickly as possible while entertainment software is typically designed to promote mastery only after an extended period (to promote re-playability). This divergence can have significant effects upon how the confounding factors affect the results.

Statistical validity: Determining the number of data points needed to ensure statistical validity is challenging. Games are high-dimension spaces that could require very large data sets to ensure validity across all combinations of dimensions. Practitioners should do their utmost to identify the dimensions that are "important" if they need to identify causality (and not just correlation) within practical budgetary constraints.

Making a practical determination of whether a large number of players performing a short test *vs.* few players performing longer tests is better is problematic. Learning effects, especially for the timing inherent within the game controls and game mechanics, can be so significant that short tests may not yield useful results. Therefore, volunteer testers may be inappropriate and paid testers may be required.

Learning bias: Playing a game is a learning experience. Players must be able to learn quickly enough to feel validated yet not so quickly that they feel the game is too easily mastered (otherwise the replayability of the game may be impaired). A number of mechanical examples are presented elsewhere in this section but the role of misconception, where the player fails to comprehend the nature of the cognitive challenge has not. If players are consistently failing, even at the lowest challenge levels, this may point to visual requirements weaknesses, particularly the effectiveness of the clues provided to the player.

Tester profiling: Comparing results across testers is meaningless without appropriate contextual information such as their experience level (recognition time) and their physical abilities (reaction time). For example, during development there was a long-standing issue with swipe gestures reportedly failing for one tester. Eventually we invested in developing very low-level diagnostics that allowed us to understand that this user was moving their finger up to 10 times faster than the device could recognize. If the device had been able to keep up with this tester then that tester would have a significant advantage when playing this game compared to the average user. Players who are practiced in recognizing, memorizing and executing action sequences have similar advantages compared to novice players.

Tester motivation: Maintaining player interest during a test was much more challenging than anticipated. The perceived effort was greater than our volunteers expected and participants got "bored" or "tired". It is important to remember that these are symptoms of cognitive exhaustion and physical exhaustion respectively. This feedback implies that it may be necessary to employ more study participants with a reduced number of runs but this practice may create issues with respect to learning bias.

Given the issues associated with boredom and/or fatigue, practitioners must investigate whether these issues translate into actual gameplay problems. In other words, is the game only interesting enough to be played a few times? Such poor replayability will greatly affect commercial success.

Generalization and application: We are in the early stages with this work and have identified some mechanisms to apply these results to our work. We continue to try to "close the feedback loop" to identify whether the test results are accurate indicators of the in-game experience and whether that experience is "as intended" by the game designer. We feel comfortable in asserting that the techniques and their caveats are likely to support generalization but more work is necessary to be more definitive.

IX. Conclusions and Future Work

Experience requirements can convey substantial information in a compact manner, information that can be challenging to verbalize. However, converting experience requirements into testable functional requirements can be difficult (especially when referring to induced emotional states) and can lead to a large set of functional requirements. The resulting functional requirements can be interpreted within the context of the experience requirement which increases the probability of successful communication but the validation process for these functional requirements may be very challenging to implement.

Testing experience requirements is a non-trivial exercise with significant confounding factors. Controlling for these factors may be difficult, impractical or perhaps even impossible. Testing may require (potentially expensive) changes to the game engine and translating the results into practitioner guidance can be difficult. Validating the resulting change(s) can also be difficult and expensive.

Players learn while playing a game and this shifting performance baseline makes it difficult to interpret test results. Gathering sufficient data to be able to accommodate the learning effects means that players have to run the test many times. This can be boring and exhausting with the consequence that volunteer testers may not be appropriate for this domain. If this is generally true, then how can game developers gather accurate data for new players? Further work is needed to understand the effects of learning bias on test results in this domain.

Games require real-time observation, decision and action. Working in this domain dramatically identifies the need to characterize the *a priori* capabilities of participants if the test results are to be compared between participants. While games might be an extreme case, this factor may be much more important than previously thought in other studies, particularly in software productivity investigations. Further investigation of this area is strongly urged, particularly the sensitivity of traditional measurement techniques to distortion by individual capabilities.

The project described in this paper incurred over one person-month of development effort – and that was for a single experience requirement within a relatively simple game. While future validation efforts will be much less costly, the return on investment is questionable. Guidance for developing test and validation regimes for heterogeneous populations, soft factors like *experience* and *emotion,* and high-dimension environments is greatly needed if we wish to pursue cost-effective validation of this class of requirements.

However, if we cannot practically develop a test to validate our requirements than how can we expect to have a deliberate design process? We expect the player to learn as they play the game but are they learning, and experiencing, according to our intentions? If we empirically validate that the player *has* learned, can we learn *what* the player has learned and *how* the player has learned so that we can incorporate this feedback into our processes for future improvement? Can we then show that the feedback can be incorporated into the design *and* have the intended effect upon user experience? Does the oft-mentioned *art of game design* simply mean that there are too many dimensions to simultaneously control (for experimental purposes) and therefore we cannot have a deliberate design process at this time?

We did not address qualitative testing of experience requirements in this work. Our test population was too small to make the feedback meaningful as anything other than anecdotal evidence. There are significant opportunities for future work in this area.

Lenberg *et al.* [7] propose and define Behavioral Software Engineering as "the study of behavioral and social aspects of software engineering activities performed by individuals, groups or organizations." Their systematic literature review was unable to identify any RE papers that addressed behavioral aspects of RE despite the fact that RE is one of the most human-centric knowledge areas within software engineering. The contributions made by the present and related works could form the basis of further investigation into the behavioral aspects of RE. The CHASE workshop series (www.chaseresearch.org) could also provide contributions to this line of research.

Requirements validation typically focusses on functional requirements, model validation, *etc.* and non-functional requirements do not receive as much attention. Experience requirements could be investigated within the NFR context and techniques such as modeling and model validation could be explored. Further research into mechanisms and techniques to address the challenges noted in Section VII could lead to greater certainty in game design and development efforts.

ACKNOWLEDGMENT

The authors thank all of the people that participated in this study; your time and energy is greatly appreciated.

REFERENCES

[1] D. Callele, E. Neufeld, and K. Schneider, "An Introduction to Experience Requirements," In Proceedings of the 2010 18th IEEE International Requirements Engineering Conference, September 2010, Sydney, Australia, pp. 295-296.

[2] D. Callele, "A Proposal for Cognitive Gameplay Requirements," In Proceedings of the 5th International Workshop on Requirements Engineering Visualization (REV10), September 2010, Sydney, Australia, pp. 43-52.

[3] D. Callele, E. Neufeld, K. Schneider, "Emotional Requirements in Video Games," In Proceedings of Requirements Engineering 2006, September 2006, Minneapolis, MN, USA, pp. 299-302.

[4] M. Daneva, "How practitioners approach gameplay requirements? An exploration into the context of massive multiplayer online role-playing games", Proceedings of the 22nd International Requirements Engineering Conference (RE14), Karlskrona, Sweden, August 2014, pp. 3-12.

[5] ISO 9421-210 "Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems", International Standards Organization

[6] S. H. Lee, "A usability-pattern-based requirements-analysis method to bridge the gap between user tasks and application features", In Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference (COMPSAC), Seul, South Korea, 19-23 July 2010, pp. 317-326.

[7] P. Lenberg, R. Feldt and L.-G. Wallgren, "Behavioral Software Engineering: a Definition and Systematic Literature Review", in the Journal of Systems and Software, vol. 107, Sep 2015, pp 15-37.

[8] D. Loeffler, A. Hess, A. Maier, J. Hurtienne and H. Schmitt, "Developing intuitive user interfaces by integrating users' mental models into requirements engineering", Proceeding BCS-HCI '13 Proceedings of the 27th International BCS Human Computer Interaction Conference, Steve Love, Kate Hone, and Tom McEwan (Eds.). British Computer Society, Swinton, UK, UK, , Article 15 , 10 pages.

[9] F. D. Davis, “Toward preprototype user acceptance testing of new information systems: Implications for software project management” IEEE Transactions On Software Engineering, vol. 51 no. 1, Feb 2004, pp. 31-46.

[10] D. Callele, K. Wnuk and M. Borg, “Confounding Factors When Conducting Industrial Replications in Requirements Engineering,” In Proceedings of the First International Workshop on Conducting Empirical Studies in Industry, May 2013, San Francisco, pp. 55-58.

[11] E. B. Seufert, Freemium Economics: Leveraging Analytics and User segmentation to Drive Revenue. Morgan Kaufmann, 2014.

[10] M. C. Primrose, “User experience grading via Kano categories”, Proceedings of the 18th IEEE International Requirements Engineering Conference (RE10), Sydney, Australia, September 2010, pp. 331-336.

[11] A. Stroe, “User Experience Grading via Kano Categories” Master in Business Informatics Utrecht University, can be accessed at http://foswiki.cs.uu.nl/foswiki/pub/MethodEngineering/UserExperienceGradingViaKanoCategories/final_AnaStroe.pdf

[12] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, “Experimentation in Software Engineering”, Springer 2012.

[13] Hans-Bernd Kittlaus, P. N. Clough, “Software Product Management and Pricing: Key Success Factors for Software Organizations”, Springer 2009.