FTC

Proceedings of Future Technologies Conference

6-7 December 2016, San Francisco

ISBN (IEEE Xplore): 978-1-5090-4171-8
ISBN (USB) - 978-1-5090-4170-1

IMAGE PROCESSING

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

Understanding Game Actions: The Development of a Post-Processing Method for Audio-Visual Scene Analysis

Gareth Schott
University of Waikato
Hamilton, New Zealand
g.schott@waikato.ac.nz

Raphaël Marczak
LaBRI
University of Bordeaux
Bordeaux, France

***Abstract*—This paper introduces a video processing method designed specifically to give researchers access to players' behaviors within any commercially available off-the-shelf PC game. Insights into players' experiences with videogame violence are achieved from automatically processing significant elements of the audio-visual communication/feedback (i.e. moving image and sound) over long periods of uninhibited game play. The method exploits a wealth of significant and (behaviorally) influential information that is conveyed to the player as they progress through a game. Such information can relate to virtual progress (e.g. maps, mission logs), player reserves (e.g. acquisitions) or vitality (e.g. health or stamina levels), to name but a few. By analyzing what is transmitted to the player during play, feedback-based game metrics are produced that utilize the same content that the player perceives and processes during their gameplay, thus, tying the method directly to individual 'player experiences.' This method was developed for a project that addressed the persistent concerns over the impact of game violence and its translation into societal violence. The method provided the project with an alternative approach to understanding the use and function of violence within game systems, taking research in a new direction from the approach and research-designs employed by experimental psychological over the last few decades.**

Keywords—media violence; game metrics; audio-visual feedback; post-processing

I. Introduction

The arguments for the appropriation of conventional methods of gathering game metrics to further player experience research (as distinct from user experience research) has been compelling (Nacke et al., 2009) and well received by the wider research community associated with game studies, although not fully applied within research studies. The core appeal of incorporating game metrics into player research is not that dissimilar from its appeal to user (design oriented) research, in that it constitutes a quantitative method capable of mapping player behavior based on the conditions of play. Within user research, game metrics represents an advancement in accuracy and process on qualitative methods that were more typically employed to articulate user-experiences with game systems under development (Hilbert & Redmiles, 2000). Game metrics have come to represent a potentially powerful technique that has the capacity to transform what has been conceptualized or theorized to-date in relation to player experience. Here we refer specifically to the value of game metrics for explaining the exact nature and function of player-system relations during play (Choi, 2000; Drachen, 2008; Waern, 2012). Typically, game-play metrics are capable of providing researchers with information on player activity in relation to specific areas such as: "navigation, item- and ability use, jumping, trading, running and whatever else players actually do inside the virtual environment" (Drachen et al., 2013, p. 10). In addition, they are capable of covering the wider conditions of managing play that absorbs "all interaction the player performs with the game interface and menu" (p. 11) termed interface metrics.

With the exception of recording player input, such as pressing keys on a keyboard (or keystrokes) that can be recorded using any type of key logger software system, the gathering of game play metrics typically prescribes a certain level of access to gaming software that is authorized or sanctioned by game developers. This can take the form of 'access' to the game source code, through agreements, modding provision (Sasse, 2008) or if a release takes the form of an open source game (Pedersen et al., 2010). The approach outlined in this paper seeks to circumvent these conditions by utilizing and translating the video and sound feedback experienced by the player during play into metrics. To this effect, games employ a large amount of audio- visual feedback in order to communicate to the player during play. For example, the 'heads up display' (HUD) draws attention to the way videogames compensate for the player's physical disconnection from the game world thus requiring the game system to articulate physical condition and status of the player character. This paper describes how the graphical and sound streams from games were exploited, captured and measured automatically with the aid of algorithms that were initially developed to detect the presence of specific objects within moving-image (Bay et al., 2008), automatically segment moving-image into sub-scenes (Huang & Liao, 2001; Saraceno & Leonardi, 1997) or extract symbolic information such as annotations (Chen et al., 2001) and numbers (Ye et al., 2005).

Signal processing of audio streams constitutes an established and active research field within the computer sciences. Algorithms have been created to detect speech and music in radiophonic streams (Richard et al., 2007), music recognition (Orio, 2006) or to indicate moments of 'intensity' in movies via audio tempo analysis (Yeh et al., 2009). In processing sessions of digital game play we have found it

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

possible to adapt such algorithms as they process large amounts of information in abstract and schematic ways (e.g., via life-bar representation, icons of obtained objects, on-screen blur, screen desaturation, heartbeat sfx). Our method is based on the idea that different audio and visual design elements can be extracted accurately from output streams and analyzed to create a picture of the nature and developmental changes in players' interaction and experience with a particular game.

II. Segmentation of Game Play Footage

Prior to executing automatic processing of game play, it is first necessary to segment the game play experience. That is, inititate a process in which "a game is broken down into smaller elements of gameplay" (Zagal et al., 2008, p. 176) for the purpose of analysis. To this effect, it is possible to segment based on the structure of the game itself (based on how it is divided into levels, missions, etc.), then identify the possibilities and affordances for (inter)acting within those sections. Games are sub-divided in many different ways, the most basic being a discrete level or mission (e.g. Battlefield 3) that is often divided by a pause whilst the next section of the game is loading, or bridged by a cut-scene. In doing so, breaks in game play can then be detected and time-stamped and used to illustrate the pace at which a player is progressing through the game.

The concept of indexing is then applied to further distinguish the identification and location of information that denotes a) where in the structure of the system the player is active, e.g. in-game verses menus, b) the nature of the player's involvement, e.g. Calleja's (2011) distinction between different types of involvement (e.g. narrative or ludic), or c) the degree of interactivity, ranging from fully, semi to non-interactive. To summarise, segmentation is therefore the determination of the boundaries of a coherent section of play that is also comprised of a set of indexical properties. For example, the presence of a cut-scene will often represent the end of a large section of play and beginning of a new one, with the content of a cut-scene often containing a significant plot point that drives the change (to a new segment). This event (the cut-scene) also denotes a distinction in the degree of interactivity employed by the player (index). Algorithms were employed to detect indexical differences between interactivity and non-interactivity of different segments, in order to avoid incorrectly detecting actions on-screen like fighting as 'play' when they occur in a sequence such as a non-interactive cut-scene.

In order to be able to fully categorize and localize aspects of a play experience captured by the metric data, the player experience is contextualized across multiple layers (outlined below):

- Game System
- Game World
- Spatial-Temporal
- Degree of Freedom
- Interaction

In processing footage captured from game play sessions we treat the game system as a whole. That is, the initiation of game-play, the experience of playing in a 3D fictional world, only occurs once players move from splash screens (e.g. copyright, production credits) to eventually reach a higher order 'main menu' where players are able to then activate play and enter the game world. Only when play is initiated does the player move from the game system layer to the game world layer, the 3D space in which the game is situated and play is realized. From that point onward, play is either broken or paused by the player, typically exiting play through higher order menus.

The game world layer contains what we term 'instances' of game play (that permit segmentation). During post-processing audio-visual analysis of such 'instances' the player is present only as the entity behind, and responsible for triggering the game footage under examination. A key task in this process is to distinguish between in/out game and active/inactive and what this entails in terms of coherence between two consecutive frames. After this we begin to distinguish the spatial-temporal information contained within the game world layer as we identify deliberate pausing or detachments from the game world by the player, or information that is indicative of player progress relating to terrain traversed or activities completed that might trigger cut-scenes or new missions via a loading screen. These elements constitute identifiable nodes that map the progress and journey of the player and also the timing of when players experience core events in the game (useful for cross-player comparisons). Related to player progression through a game are the degrees of freedom and interactivity layers that constitute the manner in which the logic and rule system of the game is conveyed to the player and the degree to which the player is required to engage with the information provided by the game, or is permitted to ignore cues provided by the system.

III. Automatic Processing

The paper now provides examples of the kinds of audio and video processing algorithms that are being applied to gather data on player experience. The aim of automatic processing is obviously to reduce the complexity and time-demanding task of manually segmenting and coding gameplay performances of players. It is worth noting that when algorithms are applied, they are done so with knowledge of the value of symbolic information in terms of what it signifies in relation to player activity and involvement. For example, when a logo-detection algorithm is applied to footage of game play, in order to confirm the presence of a logo the intent is not to learn about the morphological properties of the logo (for instance colour, size, shape, potential text content) but the meaning of its presence or absence in terms of play. For instance, the presence of a HUD on-screen signifies game-play is in session, whereas its absence can signify other events are in progress (e.g. a cut-scene). Similarly with sound detection, the intent is not to analyse the sound itself (in terms of frequency, amplitude or tonality) but the function of the sound (e.g. it might signify successful object acquisition). The researcher is therefore required to engage with the game under consideration in advance of processing in order to elicit the key gameplay concepts and match them to their various audio-visual representations.

Identify applicable sponsor/s here. If no sponsors, delete this text box (*sponsors*).

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

The audio-visual feedback method of producing game metrics typically utilized:

1) Detection algorithms to identify the presence or absence of static and moving graphical information which produce discrete values. That is, whether a logo is detected or not and how many logos are detected and where. These algorithms permitted the isolation of moments of play from other elements of the game experience that possess reduced levels of interactivity (in which logos disappear). Detection of moving logos were typically required as they provide information such as the number of enemies on screen (number of logos) and the player's reaction (i.e. cross-hairs in the center of the image signal if the player is preparing to fire at enemies).

2) Other forms of information constantly present on-screen (during play) provide continuous updates on player's standing in the game, for example, health, power or stamina bars. Bar progression assessment addresses information sources that present continuous data where the value can increase or decrease between minimum and maximum values.

3) Sound correlation techniques are used to identify key and significant sounds enclosed within the multi-layered audio-stream of the gameplay performance footage. The sonic atmosphere of most videogames carries a considerable amount of information. Some sounds extend beyond the graphic information directly presented to the player as objects in the diegetic world of the game can generate sounds even when out of the player sight. Furthermore, most of the graphical elements experienced within games are also paired with audio feedback that confirms choices and actions have been taken by the player. A loss of health is accompanied by avatar screams in *Bioshock 2*, the use of slow motion in *Max Payne 3* possesses its own sound motif and even the most basic functions such as menu activation is also accompanied by specific sounds.

A. Example 1: Detection of Static Information

Beginning with the objective of differentiating moments of play from other experiences that games also offer, it is useful to determine experiences categorized by reduced levels of interactivity. The algorithm that proved extremely useful for discerning player activity is derived from research that pursues the automatic detection of logos enclosed in TV-streams (Mikhail & Vatolin, 2011; dos Santos & Kim, 2006). Typically, this strand of research seeks to automatically evaluate the presence, or absence of specific TV-channel logos that are indicative of normal stream (standard logo), live re-transmission (modification of logo) or commercial breaks (disappearance of logo). The application of logo detection to game play footage extends its application significantly due to the high number of symbolic elements broadcast via the video stream to the player (Fagerholt, 2009; Ruch, 2010). Indeed, detecting logos in the context of game analysis can signify:

- The detection of the presence of the Head-Up Display (HUD) that is indicative of the player being fully interactive (walking, running, crouching, crawling, collecting, fighting etc.), while its absence typically indicates cut- scenes, menu activation, etc.
- The detection of the appearance or disappearance of a pop-up panel, informing or warning the player about a change in the game world, for example, when a new goal is created for the player to fulfill, when an NPC is trying to communicate with the player (e.g. radio icon) or when the nature of the challenge faced by the player intensifies (enemy wave icons, Zagal et al., 2008) or increases in levels of difficulty (e.g. timed tasks).
- The detection of cues that relate to player choices (e.g. skip a cut-scene) or affordances available to the player, for more effective or strategic play (e.g. loot a corpse or container, spend money). When a player opts to follow a cue, this too is often visually represented (illustrating compliance or adherence to the cues presented), for example, the presence of an 'audio-diary' icon in Bioshock 2 is triggered when a player opts to listen to an audio-tape.
- The detection of specific graphical or design features illustrative of a distinct space, for example, in *Bioshock 2* where a neon logo is indicative of the vending machines which are accessible to the player. Any kind of static information on screen can indeed be abstracted as a logo including any static text or static portion of screen (typically non-diegetic information that is superimposed over the game world). This processing method can be executed using either edge or colour recognition, depending on the design of the game under evaluation. In the case of Max Payne 3 the game employs a semi-transparent logo that updates the player on the status of avatar health. The transparency of the logo combined with player movement permits colours from the bottom layer (game world) to alter the colour of that logo. In this instance edge detection, focusing on logo shape, is preferable to colour detection as a means of recognizing the logo.

When an element of interest is not static or fails to appear in the same screen position, it is necessary to execute an algorithm capable of dealing with the localization and detection of objects that change position on screen. Here image registration can be employed. This research executes algorithms that are efficient in identifying commonalities between two images thus highlighting how they relate to each other. Thus, a smaller object can be located with a larger image when its position is not guaranteed. One method employed within this strand of research is cross-correlation (Dos Santos et al., 2006). This works by taking a discretized image, and scanning the reference image over each section of the larger image to seek a match. For each scanned position a correlation value is calculated (using a cross-correlation formula) with a value of 1 representing a perfect match.

1) To execute detection of a non-static object, the algorithm employed requires the following inputs:

2) An object reference, representing the object that will be cross-correlated.

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

3) A mask image can be used to restrict the search area for an object in cases where the object only occupies a smaller area on screen. If no mask image is specified then the whole screen is scanned.

4) Once the scanned image locates likely matches, then an error tolerance value, will be employed to determine whether the object identified is a correct match.

5) A time tolerance value, specifying the minimum period of time that the object should appear on-screen.

6) A frame step that reduces the processing process by not requiring each individual frame to be processed, but instead every 2nd or 10th frame.

The algorithm functions similarly to the static object algorithm with the addition of several processing functions that account for the challenges of having to locate an object (or several objects) that may be repositioning. Once the object image has been converted into a binary edge representation it is shifted sequentially through every possible position over the video frame (beginning in the upper left corner of the frame and ending in the bottom right corner).

The correlation process (as shown in Figure 1) illustrates the process of locating a "cross" logo that appears in the game *Battlefield 3* and functions to indicate the positions of enemies at a distance (enemies that are also hidden behind objects). The cross logo has been sequentially shifted across the image, in doing so, correlation scores have been generated and stored in the image matrix (the whiter the pixel, the better the correlation score). Figure 1 shows one instance that displays two perfect correlations in the upper right hand corner. These logos inform the research about the nature of the space (the player is in an environment where enemies are present), it informs the researcher as to the number of enemies (number of logos) and also provides information on the player's reaction (i.e. if the logo is in the center of the image, then the player is typically attempting to aim at those enemies).

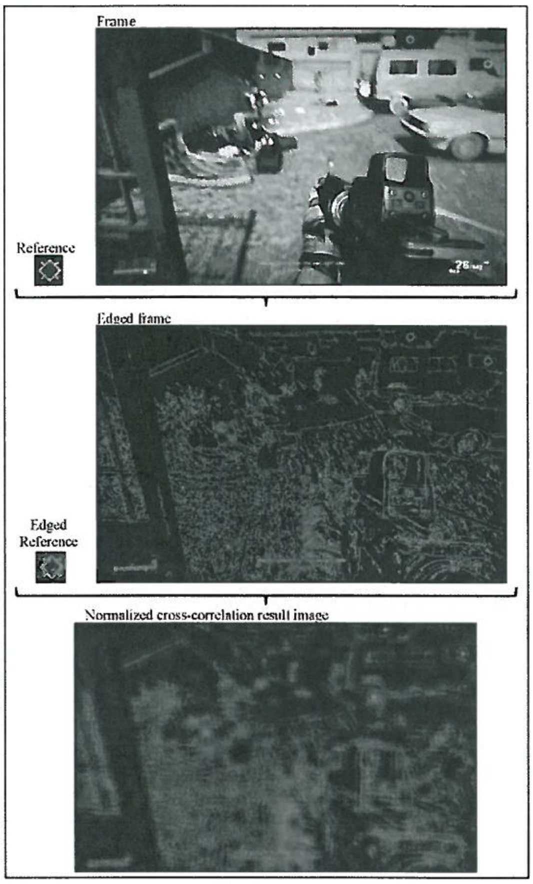

Fig. 1. Cross-Correlation employed to locate a 'x' logo in Battlefield 3

Figure 2 illustrates an example from when a logo representing enemy position (*Battlefield* 3) can appear anywhere on-screen. Figure 2 demonstrates the value of the

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

logo position for the player. While the top graph demonstrates the appearance of logos on-screen, the lower graph indicates how the player has then responded to this information by adjusting their position, thus centering the logo position on-screen in order to take aim and fire at the enemy (an action that can be further confirmed by keystroke measurement).

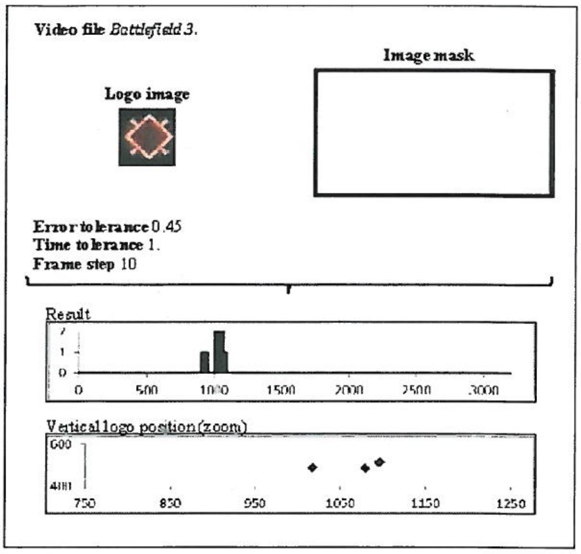

Fig. 2. Analysing logo changing logo position: Enemy location in *Battelfield 3*

IV. CODES OF ANALYSIS

This section provides some illustration as to how the algorithms function, drawing upon captured game sessions conducted with participants who provided the study with hours of gameplay footage to analyze. Different modes of analysis were employed in utilizing the metric data: *contextual* (where different metrics are combined to explain player actions), *semantic network* (gameplay elements that are known to influence one another), *loops* (the detection of sequences that are (nearly) identical as they are replays post screen death), and *comparitive* (comparing segments from one performance, with other segments occurring in different performances). Each mode was employed during the analysis of four different games (*Bioshock 2* (2K Games, 2010), *Dead Island* (Deep Silver, 2011), *Battlefield 3* (Electronic Arts, 2011) and *Max Payne 3* (Rockstar Games, 2012) played by participants (n = 10 per game, except *Dead Island* n = 1, as *Dead Island* was used as a pilot study to test the organisation of the gameplay sessions). Examples are provided below of a contextual analysis and the semantic network analysis of game play.

It is important to note, prior to extracting metrics of interest a pre-analysis stage of each game is necessary in order to identify key logos of interest, or sounds representative of specific gameplay actions. Once the inputs have been determined, different algorithms are then executed, generating curves as outputs (converted into a "timestamp – value"), each curve representing one specific feedback-based gameplay metric. Finally, the different metrics are assembled into a single graph, in order to study their co-evolution and characterized the nature of players' gameplay experience. The figures included in the following section not only display the combined metrics for the same performance, but they also indicate the metrics layer of affiliation and the algorithm used to process them.

A. Contextual Interpretation of Metrics

Contextual Interpretation describes sequences in which a specific gameplay metrics provides context necessary for understanding several other metrics. *Context* can range from 1) *temporal*, being an interval delimited by a beginning and an end, inside which a change occurs that contributes to an understanding of other gameplay elements; 2) *perennial*, being a specific point after which the particular change is then persistent for the rest of the performance; to 3) *hierarchical*, referring to an interval during which a particular subset of metrics are known to be operational.

Figure 3, for example, displays an example of what is described as *temporal*. This illustrates the notion of a change that is able to explain specific player behaviors, until the change is reversed. Figure 3 displays intervals in which the player encounters 'enemy waves.' During these periods, the game system introduces a boss-like challenge (Zagal et al., 2008), meaning that the player is presented with more and tougher enemies. These intervals are interesting to evaluate for player's reactions to sudden health drops and challenging enemy confrontations. In between 150 seconds and 1200 seconds, the waves are very short, ending with the player's screen death. However, before entering the next wave, the player collects as many items dropped by the previously killed enemies as they can. This can be interpreted as a *desperate* response as the player tries to locate the few items that can provide some help, but without any real success. However, between 3000 seconds and 3120 seconds, a longer enemy wave can be detected, yet the player is still alive at the end of the interval, and a rise of health shows that the player is finally able to cure themselves to overcome the challenge. The player, who is more prepared and ready for the fight, appears to engage in much more controlled behavior. By identifying the context of play, it is possible to assess how a player engages not only with specific instances but also similar challenges over the course of the performance.

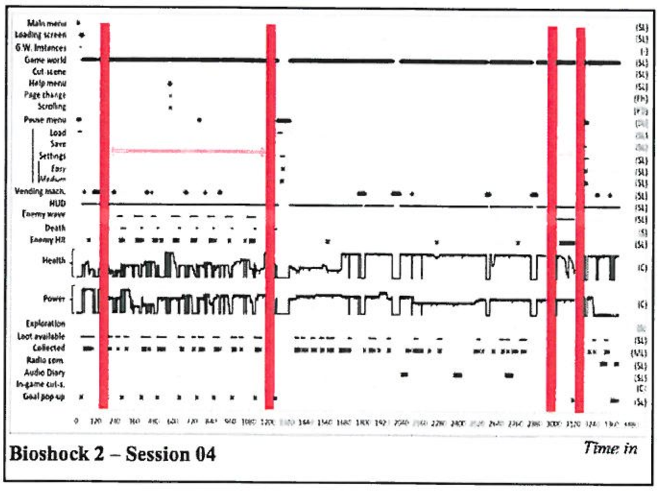

Fig. 3. illustrates a temporary change of context (implying the detection of an interval, where the change of context starts, and when it ends)

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

The enemy wave interval highlighted in Figure 3 (between 3000 seconds and 3120 seconds) occurs during the "easy" interval, probably also explaining part of the player's success to overcome it. During the "easy" interval, prior to the final "wave", the player takes time, profiting from the easier condition or execution of the game rule system, allowing the player to replenish their inventory (item collection), explaining the rise in health during the "wave". The player is equipped with enough items to cure themselves when hit or damaged. However, once the "wave" has been passed successfully, the player decides to return the difficulty level to "medium".

Play can also be interpreted based on prior metrics (*preceding*), co-occurring metrics (*concurrent*), or metrics appearing later (*subsequent*). An example from *Bioshock 2*, showed a player is inside the help menu space between around 1680 seconds and 1920 seconds of a game play session. Within the help menu space, no real action occurs. The player is only presented with text they can read, that will inform them about *Bioshock 2*'s gameplay mechanisms and the fictional game world of Rapture. In this instance, it is sensible to wonder why the player choses this moment to spend a significant amount of time (approximatively four minutes) inside the "help menu" space. The rationale for staying in the "help menu" is again understood by considering the precedent metrics. Figure 4 shows that, at around 1600 seconds, the player encounters their first demanding fighting sequence, with two important and consecutive losses of life (despite the use of first aid kit to replenish health level). With this information, the player's actions points to the more likely explanation of the player requiring further knowledge of the game mechanisms and the game world (to understand the enemies they were fighting against). It is possible to suppose that the player probably felt insecure after being confronted by their first intense fight, and needed to reinsure themselves, by updating their knowledge of the game.

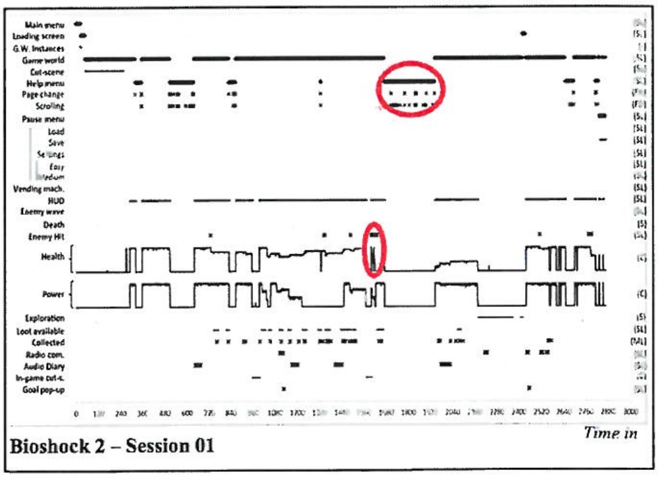

Fig. 4. Preceding metrics: difficult section of play leading to use of help menu in *Bioshock*

Semantic Network

A semantic network analysis of the data deals with the identification of gameplay elements that are known to influence one another. That is, the evolution of one is likely to directly impact the evolution of another. The role of a semantic network analysis is to focus on intertwined elements in order to see how the player experiences their conjoint evolution, and how the player modifies their behavior in response. For instance, the relationship between fighting and loss of health can easily be seen as part of the same semantic network as they constantly influence each other.

Figure 5 illustrates a combat related semantic network for *Bioshock 2* that draws on data in the form of the identification of the HUD (degree of freedom that permits fighting), identification of enemy waves (indicative of a tougher fighting sequence), death (failure as an outcome), enemy strike (an interaction representing close combat where the player successfully hits an enemy), health and power (interaction representing the remaining energies of the player, in terms of life and weapon power).

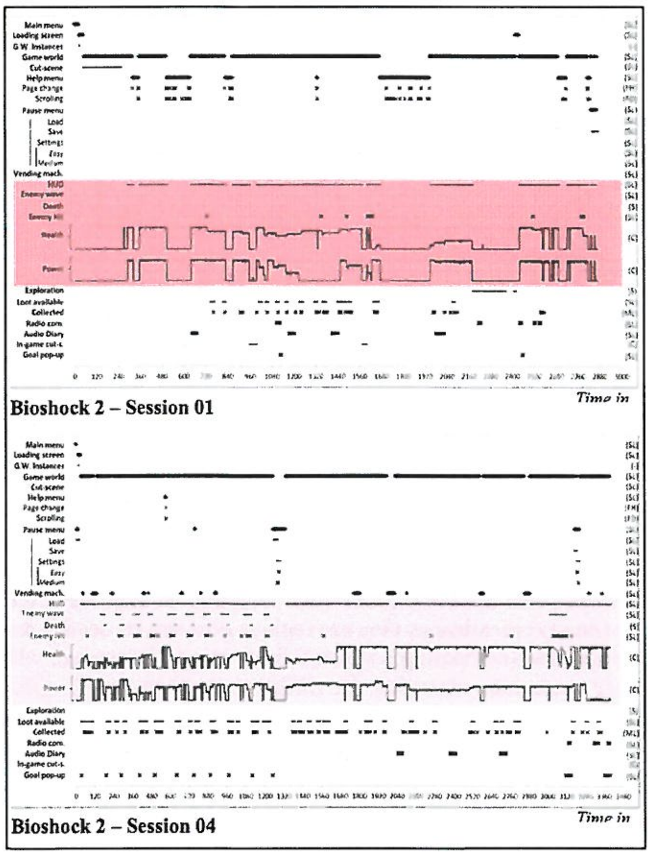

Fig. 5. Semantic Network Analysis of Play

It is interesting to note that the player uses all their power for the first time at 1000 seconds (the former zero points represent the moment when the player is in the 'help menu', thus removing the power bar from the screen). The player tries to kill an opponent using power, fails, and then tries using a weapon. But more interesting is the graph summarizing the fourth session (bottom summary), demonstrating a potential match between the player's behavior and the game designer's intentions. Indeed, the player only dies during enemy waves. During these waves, the player is confronted with a lot of close combat (as highlighted by enemy strikes), consequently the

978-1-5090-4171-8/16/$31.00 ©2016 IEEE

health/power levels are somewhat tumultuous, illustrating a sequence in which the player is fighting for survival. During a successful battle sequence (3000 seconds), the health drops more incrementally, and no power is used, showing a more controlled weapon oriented fight (the player is able to heal themselves during the fight). This would suggest that the player, from session 1 to 4, learned how to adapt their strategy, which is probably the expectation of the game designers when designing the enemy wave sequences.

V. SUMMARY

This paper has introduced a new approach for the generation of game metrics that exploits the audio and visual feedback experienced by the player during gameplay. Whilst it may appear a cumbersome method at this stage of its development, the application of algorithms are combined and tailored to the specific visual and sound design of the game under examination providing flexibility in its application. The method also has the advantage of allowing game researchers to accurately process footage from any PC-based game, reducing the demands and error associated with a manual analysis and permitting an examination of game- play across a sample of players. Underpinning this method is also a framework for the segmentation and analysis of a gameplay performance to which the specific tools of the method can be contextualized and the value of specific informational sources can be assessed in relation to player experience. For an example of the application of this method to the issue of videogame violence readers are pointed to AUTHORS (DATE) use of the method.

This paper aimed to outline just a small number of examples of the type of algorithms that can be appropriated and adapted in order to automatically detect key graphical and sound streams in games. All the examples provided function with a high degree of accuracy, and contribute to the construction of an automatically generated summary of a play performance. Such summaries can then be employed in an interpretation of a player's experience of play from a behavioral perspective. The algorithms represent a straightforward method in terms of implementation, providing meaningful results that can contribute to the wider endeavors of player experience research.

REFERENCES

[1] Bay, H., Ess, A., Tuytelaars, T., & van Gool, L. (2008). SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), 110(3), 346 – 359.

[2] Calleja, G. (2011). In-Game: From Immersion to Incorporation. Cambridge, MA: The MIT Press.

[3] Chen, D., & Bourlard, H. (2001). Video OCR for Sport Video Annotation and Retrieval. In Proceedings of the 8th International Conference on Mechatronics and Machine Vision in Practice (pp. 57 – 62).

[4] Choi, F. Y. Y. (2000). Advances in Domain Independent Linear Text Segmentation. In NAACL 2000 Proceedings of the 1sr North American chapter of the Association for Computational Linguistics Conference (pp. 26–33).

[5] Drachen, A., & Canossa, A. (2008). Defining Personas in Games Using Metrics (pp. 73–80). Presented at the Future Play, New York, USA. doi:10.1145/1496984.1496997

[6] Drachen, A., Seif El-Nasr, M., & Canossa, A. (2013). Game Analytics - The Basics. In Game Analytics. Retrieved from http://andersdrachen.files.wordpress.com/2012/08/thebasics.pdf

[7] Fagerholt, E., & Lorentzon, M. (2009). Beyond the HUD - User Interfaces for Increased Player Immersion in FPS Games (Master's Thesis). Göteborg : Chalmers tekniska högskola. Retrieved from http://publications.lib.chalmers.se/records/fulltext/111921.pdf

[8] Hilbert, D. M., & Redmiles, D. F. (2000). Extracting usability information from user interface events. ACM Computing Surveys (CSUR), 32(4), 384–421. doi:10.1145/371578.371593

[9] Huang, C.-L., & Liao, B.-Y. (2001). A robust scene-change detection method for video segmentation. IEEE Transactions on Circuits and Systems for Video Technology, 11(12), 1281 – 1288.

[10] Mikhail, E., & Vatolin, D. (2011). Automatic Logo Removal for Semitransparent and Animated Logos. In Proceedings of GraphiCon 2011.

[11] Nacke, L. E., Drachen, A., Kuikkaniemi, K., Niesenhaus, J., Korhonen, H. J., van den Hoogen, W. M., & de Kort, Y. A. W. (2009). Playability and Player Experience Research. London, UK: DiGRA. Retrieved from http://www.bth.se/fou/forskinfo.nsf/17e96a0dab8ab6a1c1257457004d59ab/e 0a8cdd8cfc0c7e6c125762c005557c0!OpenDocument

[12] Orio, N. (2006). Music Retrieval: A Tutorial and Review. Foundations and Trends in Information Retrieval, 1(1), 1 – 90.

[13] Pedersen, C., Togelius, J., & Yannakakis, G. N. (2010). Modeling Player Experience for Content Creation. Computational Intelligence and AI in Games, IEEE Transactions on, 2(1), 54 – 67. doi:10.1109/TCIAIG.2010.2043950

[14] Richard, G., Ramona, M., & Essid, S. (2007). Combined supervised and unsupervised approaches for automatic segmentation of radiophonic audio streams. In Proceedings - ICASSP 2007 (Vol. 2, pp. 461 – 464).

[15] Ruch, A. W. (2010). Videogame Interface: Artefacts and Tropes. In Videogame Cultures and the Future of Interactive Entertainment Global Conference. Oxford, UK.

[16] Santos, A. R., & Kim, H. Y. (2006). Real-Time Opaque and Semi-Transparent TV Logos Detection. In Proceedings of the 5th International Information and Telecommunication Technologies Symposium (I2TS).

[17] Saraceno, C., & Leonardi, R. (1997). Audio as a support to scene change detection and characterization of video sequences. In Proceedings of 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 4, pp. 2597 – 2600).

[18] Sasse, D. (2008). A Framework for Psychophysiological Data Acquisition in Digital Games (Master's Thesis). Otto-von-Guericke-University Magdeburg. Retrieved from http://www.gamecareerguide.com/thesis/080520_sasse.pdf

[19] Reference removed for review

[20] Ye, Q., Huang, Q., Jiang, S., Liu, Y., & Gao, W. (2006). Jersey number detection in sports video for athlete identification. In Proc. SPIE 5960, Visual Communications and Image Processing.

[21] Yeh, C.-H., Kuo, C.-H., & Liou, R.-W. (2009). Movie story intensity representation through audiovisual tempo analysis. Multimedia Tools and Applications, 44(2), 205 – 228. doi:10.1007/s11042-009-0278-8

[22] Zagal, J. P., Fernandez-Vara, C., & Mateas, M. (2008). Rounds, Levels, and Waves: The Early Evolution of Gameplay Segmentation. Games and Culture, 3(2), 175–198.

[23] Zagal, J. P., Mateas, M., Fernández-Vara, C., Hochhalter, B., & Lichti, N. (2005). Towards an Ontological Language for Game Analysis. Retrieved from http://www.digra.org/dl/db/06276.09313.pdf

978-1-5090-4171-8/16/$31.00 ©2016 IEEE