

Biofeedback: a Player's Anxiety as Input into a Video Game Environment

Christopher Bischke, Narayan Debnath

Department of Computer Science, Winona State University, Winona, MN 55947, USA

Abstract — This paper introduces a hardware-software solution for biological feedback game development. Currently, game developers do not utilize any dynamic inputs that are based off of player's biological reactions. The goal of this paper is to research how effectively and efficiently biofeedback can be implemented into a videogame. This process is focused on the hardware and software implementation. On the software side, the game dynamically generates events based on a player's heart rate. We use a standard heart rate monitor that can be attached to the player's thumb. Basing the software response the biofeedback yields an intimate player-game experience.

Keywords- *biofeedback; video game; game interaction; hardware; software implementation.*

I. INTRODUCTION

Video game developers aspire to create more personal and intimate games for players. However, static standard input devices such as a keyboard and mouse vary the experiences for each player. A game that wants to invoke emotion of fear into players, may only invoke the correct emotion to a percentage of its player base. If the game was able to be tailored to each individual player's preference, games may have a higher rate of invoking the correct emotions. A third dynamic input device to monitor stress levels of players might improve a game's ability to create a more personalized experience. The third input would monitor a player's heart rate. The program would assume a player's stress level and dynamically craft a unique experience in the game, thus crafting an intimate experience for a player. The aim of this research is to validate the hypothesis stating that player heart rate monitoring will aid in creation of intentional dynamic game events to sustain elevated player heart rate during play through.

II. "FLOW"

One of a game developer's main goals is to create a game that fully emerges, engages, and challenges a player enough to make the game enjoyable [1]. According to the Flow Theory researched by Mihaly Csikszentmihalyi in the early 1980's, video games need to maintain a constant balance between levels of difficulty [2].

"Flow" is described as when the balance of challenge and skill of a player is achieved [2]. In this project's case, the challenge/anxiety is derived from fear, and the skill/boredom is a product of how frightened a player is. When a player is in "Flow" the game is easy enough to be enjoyable, but difficult enough to be challenging [3]. Ideally, a player wants to remain in flow throughout the

entire game, but with static input devices that is not always possible. With aid of a biofeedback input such as a heart rate monitor, the game has access to a wealth of player based information.

With static inputs, a developer has to assume what is going to challenge the player. A developer may run a statistical experiment between two game events: A and B. Event A has a more positive feedback, so event A gets implemented into the game. But what about the players that enjoyed event B more? Even if it is a small subset of players, they are not getting as good of an experience as Event A players. It is possible with enough research into the psychology of videogames along with biofeedback, developers can design games that appeal to a much wider audience.

This research will focus on a horror genre game. When evaluating heart rate of a player, it is easier to assume that a higher heart rate is increased stress, while a lower heart rate is boredom. In order to evaluate other emotions such as, ecstasy, excitement, sadness, and anxiety against each other, the biological input device needs to be more robust than a heart rate monitor. For instance, a heart rate basically has two states: elevated and resting. A developer is going to have to assume and interpret what an elevated state means depending on context. However, a biological device that could monitor electricity signals in the brain can sense an array of feelings, not just two assumed states. Developers want to keep the player within flow. For the ease and purpose of this experiment, this project is going to maximize the anxiety of the player.

III. HARDWARE AND SOFTWARE

National Instruments documents the original specifications for the heart rate monitor [4]. The monitor is relatively easy to build and can be constructed by common and inexpensive items from an electronics store [4]. The LED is powered by a voltage source, then the LED is faced in such a way that the light pierces the subject's fingertip where the light is received into a photo resistor. The photo resistor outputs a value to an automation I/O device [5] which interprets the data for the game program.

The monitor records a player's heart rate through the thumb digit; however, this device is bulky, so the thumb digit is not able to be used on a keyboard or game controller. A more slimming device, like a glove, could be mounted on the thumb so all digits can be used. However the software does not rely on the hardware implementation, as long as the data is discernable between a calm and stressed state. In the current hardware solution,

the photo resistor reads a different voltage level based on the amount of blood flowing through the finger [6]. This provides for a non-intrusive and easy way to monitor the heart rate of a player [6].

VALVE, a game company based out of Bellevue, Washington, United States, offers their powerful game engine for public use. The game engine offers robust and professional tools such as a map editor, model poser, and the actual game code itself. VALVE's Source Engine was utilized to create the game for this project. The code had to be modified to accept a heart rate monitor as input. The game interprets the heart rate input, analyses the heart rate, and then finally outputs an event to maximize a player emotion.

IV. THE GAME

The game keep tracks of certain game events while measuring a player's heart rate. If the heart rate increases past a relative threshold after a noted game event, the game remembers that event for future use. When playing a horror game, if a player is frightened of a certain game event, the game remembers that event and then reuse the same elements later in the game to frighten a player again. The purpose of this is to narrow down what type of game events scares the player. Each player will have different fear tolerance for different events. It is up to the game to personalize each game event to maximize fear for a player.

The game map that the participants will play through is linear style map. Linear means the map has a fix path for the player to follow; however, the game events that generate within the map are completely dynamic. The game generates events based on how the player previously reacted to past events.

The game generates events that are aimed to scare and heighten a player's heart rate. This game follows a similar model to a critically acclaimed horror game called, "Amnesia: The Dark Descent" developed by Frictional Games [7]. Amnesia's gameplay is based around a player being defenseless against perusing monsters. The only way to 'defeat' monsters is to quickly make a decision of where to run and hide to evade opponents. If caught, the monsters kill the player and the game is over.

Most generated events and encounters in Amnesia are based around two premises. First, a planned chase scene: a player is able to see the monster, but the monster is not yet aware of a player. This scenario allows for a player to plan his route and possible actions before he initiates the chase scene. This type of event also allows for anticipation and anxiety to build before the chase scene is initiated. The second type of events are 'jump' events. This type of event is where a player doesn't expect a chase scene and is required to make a quick decision to survive the event. A 'jump' event results in a jump in anxiety or heart rate. The game events are based on either event.

All the game events that are dynamically generated are being generated for a reason. That reason is based off of the player's biological feedback reaction to certain game events. Either the game is attempting to heighten the level

of stress of a bored player or maintain a stress level for a player to keep him engaged. Since the game has a reason for a generated event, the game is able to output that reason for a moderator to view and interpret. This helps with data analysis and understating of the game handler that generates game events.

Many map makers and coders release their maps for public use. I used a map created by the username, Riman21 [8]. The map resembles a dirty apartment complex theme. Low lighting and ambient sounds are to increase the overall eeriness theme of the map. When the game begins, the participant is forced into an apartment where static game events are generated. These events are the same to each participant no matter what. The events are jump scene events. Based on how the player reacts, the next event will either be an anticipated stress scene or a jump scene to increase maximum stress.

Once the player has completed the events within the apartment. The player is free to wander a restricted and guided path around the halls of the apartment building. In the halls of the apartment building, all the events generated are dynamic and based off of the reaction of the previous game event.

V. IMPLEMENTATION

Participants were tested within a confined controlled room. The lights were turned off for the purpose of not interfering with the heart rate monitor and add to the immersion of playing a horror game. A researcher was the only other person in the room with a participant. The researcher's job was to help begin the test, evaluate the stress of a participant, and help if any problems arise. When beginning the assessment, the researcher had the participant sign a consent form that goes over the risks, data, rewards, and purpose of the study. The participant was a complete volunteer and was able to leave at any moment during the evaluation.

Once the consent form was signed, the researcher placed the heart rate monitor onto the participant, and discussed the basic controls about the game. The researcher loaded up an introductory map that has no game events just for the purpose of the player learning game controls. Once the player was comfortable with the game controls, the researcher turned the lights off and started the actual map with the dynamic game events.

While the participant was playing the game, the researcher was evaluating the player's physical stress and the events that were generated during the game. For example, if a participant screams during the game event, the researcher would note the level of stress he assume the player is experiencing. This evaluation did not have to be extremely descriptive since it acted as a small aid later in the evaluation of the biological feedback results.

During the participant play through. The game generated events based on the biological feedback of the participant. Even though game developers want perfect balance of challenge and ease (flow), the game events generated are for maximum scare or stress. The reason for maximum stress is that this study is really focusing on the

implementation of the biological feedback through software and hardware. Once an implementation has been proven, further studies can easily work in the psychological flow model into the implementation.

Once the game was completed, the researcher viewed a graph generated by the participant. The graph will display the heart rate data over time, and which events were generated. The game has reasons for which events were generated, and those reasons are based off of the participant's reaction of the previously generated event. By interpreting that data, the researcher surveys what events scared the participant. It is possible that the participant may intentionally or unintentionally gave false information on whether they were scared of an event. So by combining the heart rate results from the video game, and the researcher's evaluation of physical stress during the play through, a confident conclusion may be made on if the correct event was generated for maximum stress. Evaluating the player's stress through survey, physical stress, and heart rate also ensures that the hardware did not malfunction during the test. For example, if the participant clearly demonstrated physical stress during a play through, and the survey confirms the player was stressed – A confident conclusion may be that the hardware failed during the experiment.

VI. RESULTS AND ANALYSIS

A total of 20 participants were tested. The number of events that each participant played through were 4 events each. The first event was not dynamically generated, so each participant played through 3 dynamically generated events. So there was a total of 60 dynamically generated events based off of a player's heart rate. Unfortunately, participants number 002 and 003, were not generating numbers that made a lot of sense. Upon further investigation it was extremely probable that the heart rate monitor was broken. For the rests of the tests a new heart rate monitor was issued and the results were more accurate. So participant 002 and 003 are going to be removed from the data analysis because of the monitor fault.

With the two test cases removed, six out of the fifty four (11%) events were generated incorrectly. Generating an event incorrectly means that the player was visibly scared but the game did not catch the elevation in heart rate. There are numerous possibilities why the program didn't catch a person being scared. One reason could be that not everyone reacts the same to fear or stress. When participants were scared, the graph would represent a large spike in heart rate. However, a small portion of heart rates had a gradual increase in heart rate even though they had physical traits of being scared. The game looks specifically for large spikes in a short amount of time, so if a person's heart rate rises gradually over time, the game will not sense this.

Forty eight out of fifty four (89%) events were generated successfully. To be generated successfully, the game needs to be attempting to scare the player, or continuing to scare the player with the correct event. In

the case where a player is bored, the game will alternate and vary the type of events until the player becomes interested again. If the player becomes scared of an event, the next event generated is a similar type of event. Once the player gets bored of a certain type of event, the game identifies the player is bored and generate a new type of event to attempt to scare the player.

Twelve out of eighteen (67%) participants were considered engaged and frighten. To be considered frighten, the game figured out an event that the player was scared of, then kept generating the same event, if the player got bored of the event, the game generates a new event to scare. If the new event continued to scare the player until the end of the game, then the player was fully frightened throughout the game. Six out of eighteen (33%) participants were bored and disengaged throughout the game. This means that the game kept generating different types of events to scare the player, but none of the events raised the player's heart rate. This does not mean that the implementation failed, but may be a reflection of the game design and scare events. By possibly having more events, bored players could eventually be engaged and frightened.

Figure 1 is a graph of a representation of a participant's heart rate who was scared of the first jump event, but was bored of the second jump event. The game knew the player was becoming bored so generated anticipated stress events for the rest of the game. Figure 2 represents a participant who was scared of all jump events. The game was able to determine the player's fear, so it kept generating jump events.

VII. CONCLUSIONS

Between frightened and bored players, the biofeedback implementation was mostly successful. In general, this implementation seems to be a fairly reliable approach. Though, there are instances where this particular biological feedback implementation did not work as expected. While evaluating the hardware, it should be noted that the heart rate is extremely context dependent.

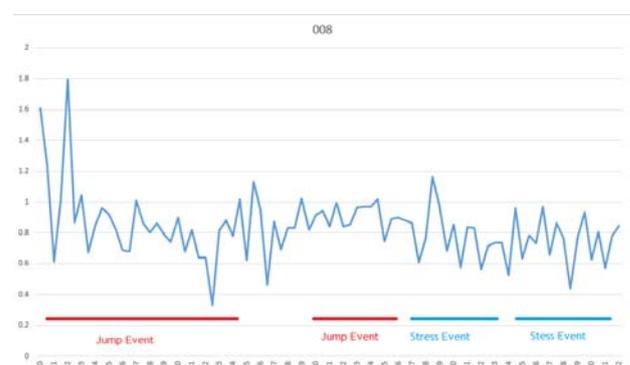


Figure 1. Representation of the heart rate of player 008.

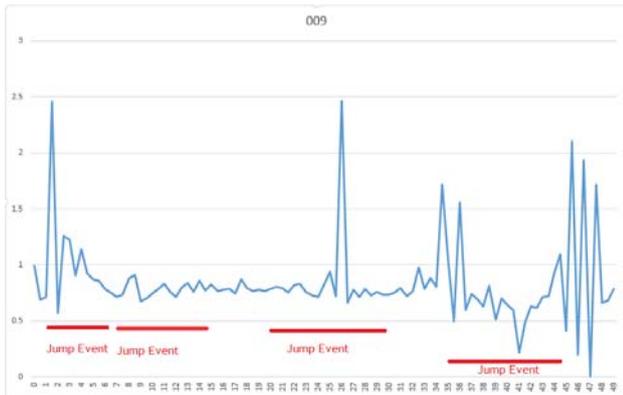


Figure 2. Representation of the heart rate of player 009

The game has to know what type of event is being generated and what emotion the player should be experiencing. The player could react with different emotions than anticipated, but still giving the correct heart rate. For example, a developer wants the player to experience fear at a certain part in the game. The game activates this event and the player's heart rate elevates and the game assumes the player is scared. But the player's heart rate may be elevated for a number of reasons: fear, excitement, exercise, etc. There is no way to confidently say that the player is experiencing the emotions that are intended. A better way to approach the hardware could be by monitoring the player's electrical signals in the brain. Certain signals fire in the brain for certain emotions, so it is a lot easier to assume what emotion the player is experiencing.

This implementation also measures the relative jump of heart rate from one sample to the next sample. This method is advantageous to finding quick jumps in heart rate that are abnormal when the player is calm. This method worked for most players, but a smaller set of player's heart rates reacted slower than the rest. This means the participant was notably scared during the play through, but his increasing heart rate climbed slower to

events. In order to accommodate slower reacting heart rates, the algorithm for determining if a player is scared needs to be re worked. An algorithm that can evaluate a player's heart rate over the course of a specific timespan may be a better implementation. Though there are limitations of this implementation, the cost efficiency, ease of implementation, and general effectiveness of a heart rate monitor is suitable for biological feedback in games to invoke or sustain certain emotions of the player.

VIII. ACKNOWLEDGEMENTS

The authors of this paper would like to extend a warm thank you to April Dawn Valet from the University of Minnesota for the aid in the creation of the survey implementation. The authors want to thank Dr. S. Iyengar and Dr. M. Zhang from Winona State University for helpful advice and guidance during the length of the project, and also to Dr. G. Cichanowski from Winona State University for the critical review of this paper.

REFERENCES

- [1] Nacke, L., "Measuring players' experience of games and real-time simulations", iTEC08 - Your Technology Hotspot, Hessen-IT, Darmstadt, Germany, 2008.
- [2] Cowley, B., Charles, D., Black, M. & Hickey, R., "Computers in Entertainment (CIE) - Theoretical and Practical Computer Applications in Entertainment", Volume 6, Issue 2, Article No. 20, April/June 2008.
- [3] Ambinder, M., "Biofeedback in Gameplay: How Valve Measures Physiology to Enhance Gaming Experience", Game Developers Conference, March 2011.
- [4] National Instruments, "Build Your Own Heart Rate Monitor", <http://www.ni.com/white-paper/14248/en/>, August 2012. Accessed December 2013.
- [5] LabJack, LabJack's U3 Platform, <http://labjack.com/u3>.
- [6] Embedded Lab, "Easy Pulse Sensor Overview", <http://embedded-lab.com/blog/?p=7336>. Accessed December 2013.
- [7] Frictional Games, Amnesia: The Dark Descent, <http://www.amnesiagame.com>. Accessed December 2013.
- [8] Riman21. Dirty Apartment. GameBanana. <http://hl2.gamebanana.com/maps/177693>. Accessed December 2013.