The Mood Game - How to Use the Player's Affective State in a Shoot'em up Avoiding Frustration and Boredom

David Halbhuber
Media Informatics Group
University of Regensburg
Regensburg, Germany
david.halbhuber
@stud.uni-regensburg.de

Jakob Fehle
Media Informatics Group
University of Regensburg
Regensburg, Germany
jakob.fehle
@stud.uni-regensburg.de

Alexander Kalus
Media Informatics Group
University of Regensburg
Regensburg, Germany
alexander.kalus
@stud.uni-regensburg.de

Konstantin Seitz
Media Informatics Group
University of Regensburg
Regensburg, Germany
konstantin.seitz
@stud.uni-regensburg.de

Martin Kocur
Media Informatics Group
University of Regensburg
Regensburg, Germany
martin.kocur@ur.de

Thomas Schmidt
Media Informatics Group
University of Regensburg
Regensburg, Germany
thomas.schmidt@ur.de

Christian Wolff
Media Informatics Group
University of Regensburg
Regensburg, Germany
christian.wolff@ur.de

ABSTRACT

In this demo paper, we present a shoot'em up game similar to *Space Invaders* called the "Mood Game" that incorporates players' affective state into the game mechanics in order to enhance the gaming experience and avoid undesired emotions like frustration and boredom. By tracking emotions through facial expressions combined with self-evaluation, keystrokes and performance measures, we have developed a game logic that adapts the playing difficulty based on the player's emotional state. The implemented algorithm automatically adjusts the enemy spawn rate and enemy behavior, the amount of obstacles, the number and type of power ups and the game speed to provide a smooth game play for different player skills. The effects of our dynamic game balancing mechanism will be tested in future work.

MuC '19, September 8–11, 2019, Hamburg, Germany

ACM ISBN 978-1-4503-7198-8/19/09.
https://doi.org/10.1145/3340764.3345369

CCS CONCEPTS

• **Human-centered computing** → User interface programming.

KEYWORDS

affective computing, gaming, dynamic game balancing, game engineering, space invaders, emotion detection

ACM Reference Format:
David Halbhuber, Jakob Fehle, Alexander Kalus, Konstantin Seitz, Martin Kocur, Thomas Schmidt, and Christian Wolff. 2019. The Mood Game - How to Use the Player's Affective State in a Shoot'em up Avoiding Frustration and Boredom. In *Mensch und Computer 2019 (MuC '19), September 8–11, 2019, Hamburg, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340764.3345369

1 INTRODUCTION

First described by Mihaly Csikszentmihalyi [3], the psychological effect of *being in the zone*, called *Flow*, is a mental state of being highly focused and thus feeling the strongest immersion with a high level of enjoyment and fulfillment. Besides its relevance in various areas of life like reading, factory working, medical working and sports [4], Flow is also a fundamental aspect in player centered game design. It is a central goal for every game designer to put the player into a "state in which people are so involved in an activity that nothing else seems to matter; the experience itself is so

enjoyable that people will do it even at great cost, for the sheer sake of doing it" [2]. The question is how can we create an appropriate game design to trigger a sense of flow?

One important concept is the *Flow Zone* [1]. Described as the balance of a game's challenge and the player's skill, the game will keep the player in the *Flow Zone* if the presented challenge is neither too hard nor too easy related to the player's ability. If the challenge is beyond the players' skills, they tend to feel overwhelmed. If the action is not challenging, the player is not engaged. Both causes the player to feel frustrated or bored and leads to quitting the game and poor retention. The most common approach in game balance design to avoid undesirable emotions like frustration and boredom is to let the user manually select the difficulty level on a scale from "easy" over "medium" to "hard". This could happen before the game starts or at any point during the game. As technology and thus games grow in complexity, video games evolved from relatively small interactive applications like *Pong* to complex software products that reach out a wide range of players worldwide, each of them with different skills. Thus, static difficulty adjustment is not enough [8] and a new research field evolved: *Dynamic game balancing* (DGB).

The idea behind DGB is to enhance user satisfaction and the gaming experience by automatically adjusting the difficulty level in real-time based on the player's ability and emotional state [14]. There are several methods applied in DGB to track both parameters. They cover genre-specific performance measures derived from in-game behavior like damage-taken, opponent damage, power up usage, levels completed [5] and inventory tracking [7], complex probabilistic models for player and difficulty progression based on player behavior [13] as well as methods from affective computing, like determining the emotional state by actions, physiological signals and facial expressions [9]. Based on the latter approaches, we present a 2D shoot'em up game with an unobtrusive DGB system that tracks the player's emotional state by combining different metrics to adapt the game play. If the user is getting frustrated, the game counteracts by reducing the enemy spawn rate or providing new power ups. If the user is bored, the game increases the challenge by handicaps. That's why we call it the "Mood Game"!

2 METHODS

Game Design

We developed a 2D shoot'em up game from top down perspective alike to the classic *Space Invaders*. A collection of the avatar, agents and game items is illustrated in figure 1. The objective of the game is to move across the screen, shoot descending hostile shuttles, preventing them from reaching the bottom of the screen and not getting hit by any obstacles

Figure 1: Game Objects: enemies, obstacles, powerUps and the player

(stones, hostile shuttles and their laser beam). The players control a space shuttle with two degrees of freedom (moving up and down, moving left and right) and uses a mounted laser cannon to defend themselves against the enemies (see figure 5). As the game progresses, the player will face a linear difficulty enhancement with new enemies, obstacles and end bosses. The common approach would be a pre-processed linear difficulty adjustment from level to level. However, we created an algorithm based on different metrics that makes the game react to the user's emotional state (see section 2). If the user shows frustration or boredom and hence crosses the *Flow Zone* boundaries, the game counteracts in different ways in addition to a linear difficulty enhancement based on gaming progress. The enemy spawn rate, the enemy speed, the type of power ups and the player speed are manipulated. We distinguish between immediate and level-based balancing. The first describes the immediate adjustment of the game difficulty within a level. For instance, if the player is highly frustrated the "Clearwave" power up is spawned

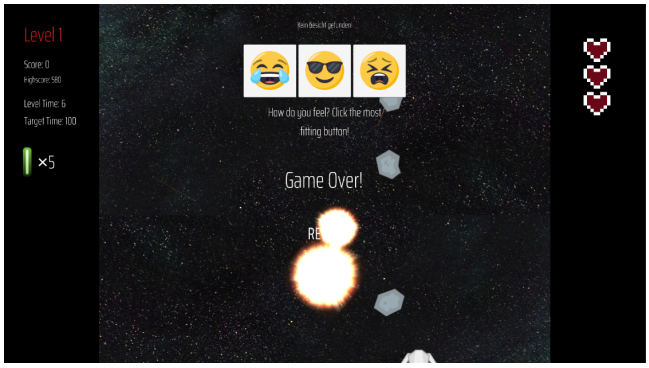

Figure 2: Game over screen with self-reporting icons

at once to destroy all enemies within the viewport. To prevent cheating by intentionally express frustration to receive the "Clearwave", we included a 20 second timeout for this item after spawning. Furthermore, the enemy speed is adjusted immediately as well, but only before spawning and the speed remains constant throughout lifetime to prevent abrupt motion changes. In contrast, level-based balancing reacts level to level. Before starting a level, the users report their current emotional state from negative over neutral to positive by selecting the appropriate icon and provides direct insights about their current emotional state (see figure 2). Self-reporting requests are exclusively made before or after a level not to interrupt the gaming experience. Consequently, we do not depend solely on objective data, but also include subjective self-reports to enhance the probability of detecting the correct user's emotional state. Furthermore, we visualize the progress of the emotional state over time with a graph on a second screen (see figure 3). We can use this data to evaluate the emotional state of the user as a function of the current game scenario to find out how the gaming events influence the player's emotions. We do not provide the user with live data to distract them from playing the game.

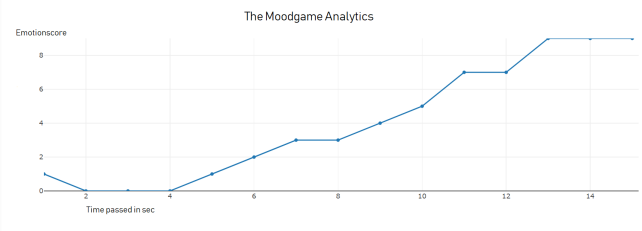

Figure 3: Graphical visualization of emotional state

Implementation

We have implemented the game using the *Unity 3D* game engine[1] and deployed it for windows computers. We have created a DGB mechanism that calculates an affective score from facial expressions (fe), keystrokes (ks) and player's self - evaluation (se). Based on the circumplex emotion model [12], we focused on valence and arousal, however, we can also detect the six basic human emotions anger, disgust, fear, happiness, sadness and surprise as defined in Ekman's emotion model [6]. To do so, we use the *Affectiva Emotion SDK*[2] which is able to recognize up to seven emotions based on facial expressions (fe). The *Affectiva SDK* processes regular webcam recordings by means of computer vision methods and returns probability scores for the emotional state. Additionally, we track keystroke dynamics (ks) as potential metrics for stress detection [11]. Unnecessary keystrokes, i.e. firing the laser beam without ammunition or during recharge time, are interpreted as arousal and integrated into the DGB algorithm. The last parameter for calculating an affective state is the user's self-evaluation (se) before and after levels. Thus, the emotional state will be calculated as follows:

$$EmotionScore = 1/3[0.6(se) + 0.3(fe) + 0.1(ks)] \quad (1)$$

As the most significant metric, self-reporting has the strongest impact for the result, i.e. each parameter is weighted by relative percentages due to their significance. Hence, equation 1 gives an emotion score between 0 and 15. We defined thresholds to infer the emotional state. If the score is between 0 and 5 as an indicator for negative emotion, the game decreases the challenge. A score of 5 up to 10 means a neutral affective state with a regular linear difficulty adjustment. Finally, a score larger than 10 leads to increasing the difficulty. Figure 4 shows the DGB system workflow.

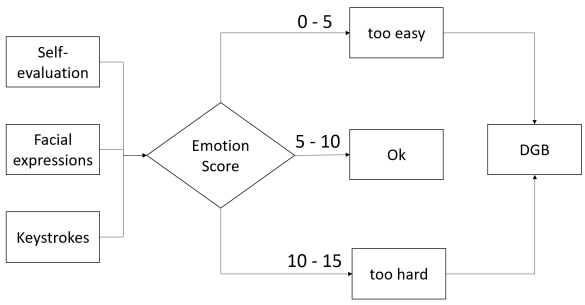

Figure 4: DGB workflow

Materials

This game is built to play on computers with a regular keyboard as input device and a webcam for detecting facial expressions. The software itself does not have any special hardware requirements. Indeed, we do not need external sensors and additional hardware components to detect user's emotions.

[1] https://unity.com/

[2] https://www.affectiva.com/product/emotion-sdk/

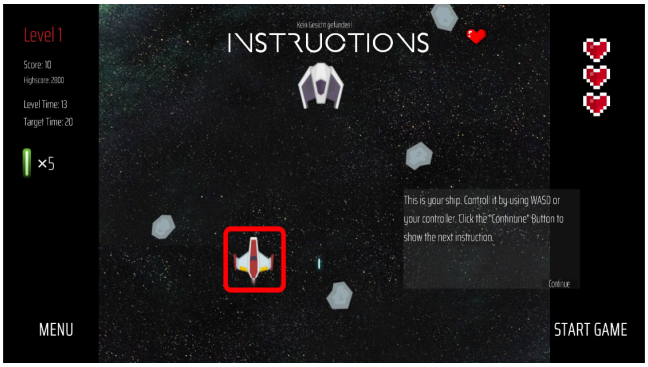

Figure 5: Tutorial screen with given instructions for the user

Conclusion

The presented game is at an early stage of development and provides first rudimentary approaches to dynamically adjusting the gameplay depending on the user's emotional state. We have yet to evaluate our application concerning game experience, emotion detection accuracy and overall impression. So far, we have conducted small play testing sessions with interviews afterwards. The participants did not notice the DGB mechanism, hence we have implemented an unobtrusive method to track players' emotional state in the first place. Note that the accuracy of the emotion detection algorithm has not been tested yet. Based on the results of the game prototype to date, we see possible difficulties with the current approach, as facial expressions and keystrokes may not be accurate enough to entirely capture the emotions experienced during a game. To underpin the automatically measured emotions, we attach increased significance to self-evaluation when calculating the emotion score. However, we plan to integrate additional metrics going beyond the smiley icons. As studies show, in-game performance measures could be promising to identify the player's skills and adapt the game difficulty to the skill level [15]. Furthermore, incorporating approved methods from affective computing, like physiological feedback should be considered as well: *Nevermind* [3] for example, is a horror game which uses biofeedback to enhance the gaming experience. The gameplay is dynamically adjusted based on mental stress derived from heart rate and emotional states detected by facial expressions through eyetracking [10]. However, we would like to continue the approach to detect user's emotions unobtrusively without the use of external sensors, like heart rate monitors or electrodermal activity (EDA) sensors thus not disturbing game experience.

[3] https://nevermindgame.com/

REFERENCES

[1] Jenova Chen. 2007. Flow in Games (and Everything else). *Commun. ACM* 50, 4 (April 2007), 31–34. https://doi.org/10.1145/1232743.1232769

[2] Mihaly Csikszentmihalyi. 1990. *Flow: The Psychology of Optimal Experience.*

[3] Mihaly Csikszentmihalyi. 1997. *Finding flow: The psychology of engagement with everyday life.* Basic Books.

[4] Mihaly Csikszentmihalyi. 1998. *Finding Flow: The Psychology of Engagement With Everyday Life.* –144.

[5] Anders Drachen, Magy Seif El-Nasr, and Alessandro Canossa. 2013. Game analytics–the basics. In *Game analytics*. Springer, 13–40.

[6] Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion* (1999), 45–60.

[7] Robin Hunicke and Vernell Chapman. 2004. AI for Dynamic Difficulty Adjustment in Games.

[8] Raph Koster. 2013. *Theory of Fun for Game Design* (2nd ed.). O'Reilly Media, Inc.

[9] Irene Kotsia, Stefanos Zafeiriou, and Spiros Fotopoulos. 2013. Affective Gaming: A Comprehensive Survey. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 663–670. https://doi.org/10.1109/CVPRW.2013.100

[10] Erin Reynolds. 2013. Nevermind: Creating an Entertaining Biofeedback-enhanced Game Experience to Train Users in Stress Management. In *ACM SIGGRAPH 2013 Posters (SIGGRAPH '13)*. ACM, New York, NY, USA, Article 77, 1 pages. https://doi.org/10.1145/2503385.2503469

[11] Manuel Rodrigues, Sérgio Gonçalves, Davide Carneiro, Paulo Novais, and Florentino Fdez-Riverola. 2013. Keystrokes and Clicks: Measuring Stress on E-learning Students. In *Management Intelligent Systems*, Jorge Casillas, Francisco J. Martínez-López, Rosa Vicari, and Fernando De la Prieta (Eds.). Springer International Publishing, Heidelberg, 119–126.

[12] James Russell. 1980. A Circumplex Model of Affect. *Journal of Personality and Social Psychology* 39 (12 1980), 1161–1178. https://doi.org/10.1037/h0077714

[13] Su Xue, Meng Wu, John Kolen, Navid Aghdaie, and Kazi A. Zaman. 2017. Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games. In *Proceedings of the 26th International Conference on World Wide Web Companion (WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 465–471. https://doi.org/10.1145/3041021.3054170

[14] Mohammad Zohaib. 2018. Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. *Advances in Human-Computer Interaction* 2018 (11 2018), 1–12. https://doi.org/10.1155/2018/5681652

[15] Alexander Zook, Stephen Lee-Urban, Michael R. Drinkwater, and Mark Riedl. 2012. Skill-based Mission Generation: A Data-driven Temporal Player Modeling Approach. https://doi.org/10.1145/2538528.2538534