Pressure as indirect feedback in an affective game

Ea Ehrnberg Ustrup,
Joakim Have Poulsen,
Mads Mathiesen

Group mta191033
Medialogy, spring 2019

Master's Thesis

STUDENT REPORT
AALBORG UNIVERSITY

Department of Architecture, Design and Media Technoloy
Aalborg University
Rendsburggade 14
DK-9000 Aalborg

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Pressure as indirect feedback in an affective game

Theme:
Master's Thesis

Project Period:
1st February 2019 - 28th May 2019

Project Group:
mta191033

Participant(s):
Ea Ehrnberg Ustrup
Joakim Have Poulsen
Mads Mathiesen

Supervisor(s):
Markus Löchtefeld

Copies: 1

Page Numbers: 62

Date of Completion:
May 24, 2019

Abstract:

Affective games conventionally detect emotional states through physiological signals. However, this poses the issue of obtrusive data collection and the requirement for specialized sensors. Researchers have thus looked into other types of feedback, with a promising metric being the pressure exerted on an input device, which relates to emotional arousal. Despite this, no related work has attempted the implementation of such feedback yet.
To explore the gap in the research, this project investigates the possibility of integrating pressure as feedback in an affective game. The game performs Dynamic Difficulty Adjustment (DDA) - i.e. continuously adjusts the difficulty to fit the individual player. Two DDA systems are developed, one being based on in-game performance and the other being based on exerted pressure.
From an evaluation of the two DDA systems and a game version with no DDA, it can be concluded that pressure-based DDA resulted in a more optimal game difficulty compared to no DDA. However, pressure-based DDA did not perform better than the other game versions in relation to the subjective player experience.

Contents

1
Introduction

In the latest years, the global game market has seen a tremendous growth in revenue and public interest [50]. Technological advancements have allowed for games to become increasingly more complex and refined, both in a gameplay and graphical sense [9]. However, as graphical improvements are reaching a plateau [8], new manners of creating novel gameplay are being explored.
An increasingly active topic in the research community [30] is the topic of "affective gaming" - a concept in which a game can detect and adapt to the emotional state of the user to deliver a better gaming experience. While different implementations of and arguments for affective games have been made, there are still some problems that have yet to be tackled. One such problem is the obtrusiveness of the data collection, especially in cases where physiological data is required, as obtrusive data collection can interfere with the immersion and emotional state of the player [28].

With this in mind, the project has been set up to explore how to overcome this issue. One possible solution could be to use indirect feedback - i.e. to infer the player's emotions from their behavior rather than measuring physiological signals with obtrusive sensors. While different types of indirect feedback have been explored, one seemingly promising metric is the exerted pressure on an input device, which has been shown to correlate with arousal in several cases. An increase in game difficulty, which has likewise been shown to correlate with arousal, resulted in increased pressure levels [20, 45, 47]. Furthermore, according to the theory of flow [10], a proper balance between skill and challenge invokes a state of flow in the player, while too big a challenge causes frustration, and too small a challenge causes boredom.
From this, it can be argued that adapting the game difficulty according to the pressure - which has been linked with both emotional arousal and perceived game difficulty - could increase the amount of time the player spends in a flow state, and as an extension of this, create a more enjoyable game experience. Despite this, there is a lack of research on the issue of integrating indirect feedback in an affective game as opposed to direct feedback [30]. Thus, the focus of the project is to integrate and test indirect feedback for an affective game.

This has lead to the following initial research problem:

How does adapting the game difficulty based on input pressure impact the player experience?

For the remainder of this thesis, research questions are first set up based on the initial research problem and answered through an analysis of existing background literature on the subject. After this, a final research problem is formulated to form the basis of an implementation and evaluation of a game which uses pressure as an indirect feedback to adapt the gameplay. The implementation consists of the development of a custom pressure reader to collect pressure data and a brick-breaker game. Three versions of the game are created, one performing pressure-based Dynamic Difficulty Adjustment (DDA), another performing in-game performance-based DDA, and the third performing no DDA. These three versions are then evaluated against one another, and the results of the evaluation are discussed in terms of indications and limitations, and finally, the project can be concluded upon.

2
Background

The following sections will present and analyze previous work related to the topic of this paper. To establish a thorough background research, a set of research questions have been formulated to be answered throughout this chapter:

1. How can you measure game experiences?
2. What is an affective game, and what does it involve?
3. How do you measure emotions?
4. How have affective games been implemented in related work?
5. How does exerted pressure on an input device relate to the game experience?

2.1 Game experience

One of the core arguments for affective games is to deliver a better game experience by improving the gameplay and interaction between player and computer [8, 30]. Throughout the years, different methods for measuring the game experience have been presented. Some studies discuss the game experience in relation to the theory of flow by Csikszentmihalyi [10]. The theory describes how a proper balance of challenge and skill should bring people into a state of flow, in which they are absorbed in their activities, as illustrated in Figure 2.1. A too high level of challenge compared to the level of skill causes anxiety or frustration, and the opposite case causes boredom.

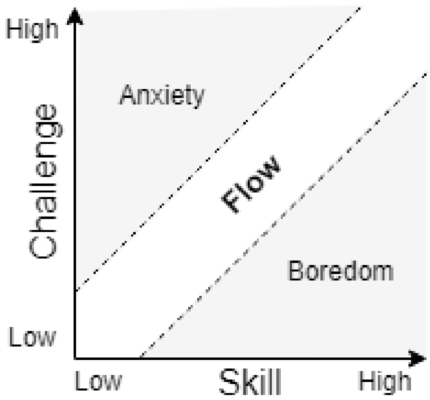

Figure 2.1: The theory of flow illustrated.

This connection between challenge and skill is why affective games tend to focus on arousal and game difficulty (explained more in detail in *2.3.1 Affective games*). In theory, too high levels of arousal (anxiety or frustration) calls for a decrease in difficulty, and too low levels of arousal (boredom) calls for an increase - thus remaining within the flow channel for an optimal experience. The balancing of the amount of challenge has to be continuous, as the player's skill levels should naturally increase as they continue playing.

Aside from the theory of flow, studies also use self reports as a method of measuring the game experience. While different types of self reports exist, a popular tool is self report questionnaires, as questionnaire data is often faster to gather and analyze.
One such questionnaire is the Self-Assessment Manikin (SAM) developed in 1994 [6]. The original version of SAM measures valence, arousal and dominance, each on a 5-point scale with an accompanying illustration. The participant marks the illustration they find most fitting to their current state in each of the three scales. Although it was originally not developed to measure game experience, it has successfully been used to measure the emotional state of players after a gaming session in several cases [7, 19, 20, 47].
Another relatively popular questionnaire in the literature is the Game Experience Questionnaire (GEQ) developed in 2013 [22]. It consists of several modules for measuring game experience aspects during and after playing a game. In each module, the participant is presented with different statements, where they can mark their level of agreement to each statement on a 7-point Likert scale. The core module of the questionnaire splits player experience into seven different aspects: Competence, Sensory and Imaginative Immersion, Flow, Tension/Annoyance, Challenge, Negative Affect and Positive Affect. Although this questionnaire is relatively new, it has already been used in several studies [20, 37, 46].

Each of the questionnaires have their strengths and weaknesses, and it should be noted that they do not set out to measure the same thing. While SAM is used to measure the general affective state of the player in terms of affective dimensions, GEQ measures their game experience in different categories or aspects. This may also be a reason for using both questionnaires rather than only one [20]. Common to both questionnaires is that due to the reflective and often retrospective nature of self reports, they have not been used as feedback in affective games, but rather as a tool to analyze and validate data post gaming session.

2.2 Affective computing and affective games

The notion of affective computing and affective gaming has been around for several years. Affective computing was originally defined as "computing that relates to, arises from, or deliberately influences emotions" [40], and it was proposed that giving computers emotional intelligence would improve and redefine the interactions between humans and computers. Extensive research has since been done on the subject of affective computing in various contexts, and it continues to grow as a multidisciplinary research area [11].
One of the branches of affective computing concerns itself with how video games can be designed and implemented to react to the emotional state of the user. This is called affective gaming, and has previously been described as gameplay "where the player's current emotional state is used to manipulate gameplay" [25], or as "games in which the players' behavior directly affects the game objectives and gameplay" [29]. Previous work on the subject has accomplished this in various game genres using various input modalities to measure the emotions of the player.

Affective games implement the affective loop, which is a concept in which the interaction between player and computer becomes two-way; the affective state of the player causes some changes in the game, which in turn then affects the player [30]. Figure 2.2 illustrates this loop.

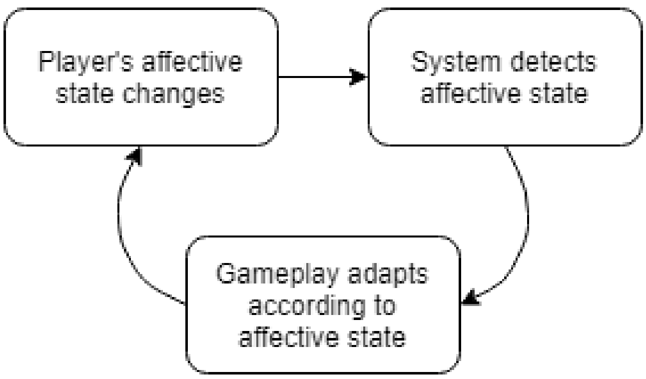

Figure 2.2: The affective loop.

As for how the system detects the player's affective state, Lara-Cabrera & Camacho argues that the kind of feedback used to adapt affective games can either be direct or indirect [30]. They describe direct feedback as the gathering and analysis of physiological reactions, as emotions have previously been shown to cause various physiological effects (more on this in *2.2.1 Measuring emotions*). The downside of direct feedback is the necessary monitoring hardware sensors, which can be uncomfortable or distracting to the player and thus potentially influence their affective state.

Indirect feedback attempts to overcome this shortcoming by inferring the affective state of the player without physiological data. This could for example include estimating emotions through the pressure applied to buttons on the input device, or building an emotional model of the player through the analysis of their in-game behavior. However, the downside of indirect feedback is that correctly defining and analyzing the gathered data can be time-consuming and not necessarily as precise as physiological signals.

2.2.1 Measuring emotions

For a computer to be able to respond to emotions, it requires the ability measure the emotional state of the user. However, emotions do not have a clear definition, and as a result, different models for categorizing or labeling emotions have been proposed throughout the years. Such models typically map emotions according to categories or in relation to dimensions [43]. As related work on affective gaming typically discusses the player's emotions in relation to dimensions of arousal or valence, a relevant emotion model is the two-dimensional circumplex model of affect created by Russell [42]. This model can be seen in Figure 2.3.

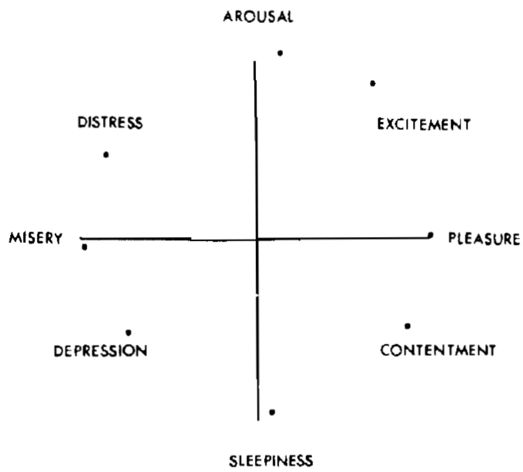

Figure 2.3: The circumplex model of affect created by Russell [42]. The vertical axis is the level of arousal (high/low), and the horizontal axis is the valence (negative/positive).

Previous work has looked into various modalities and how they relate to aspects of a person's emotional state during a gaming session. These different modalities can generally be split into physiological, behavioral and "self report" approaches. The physiological approach uses specialized sensors to measure various physiological signals. These signal can be used to assess the level of arousal, as arousal levels are controlled by the autonomic nervous system [24]. Some of the most commonly used signals in relation to affective games are the galvanic skin response (GSR) and heart activity (such as blood volume pulse or heart rate), which have both been shown to correlate with aspects that are related to arousal such as stress, anxiety or perceived game difficulty [12, 26, 27, 31, 36]. As for valence, it has previously been shown that emotions can be inferred through the analysis of facial microexpressions [15]. Facial electromyography measures the electrical activity of muscles in the face, and this approach has successfully been linked to the valence of the emotions that players have felt during gaming sessions [17, 31, 37]. Some studies have also discussed the applicability of respiration rate and skin temperature as indicators of the player's game experience and emotions [7, 41], although these modalities are less featured in the literature than the aforementioned.

The behavioral approach is indirect, as it involves inferring emotions from certain behaviors. Examples of such behaviors include body posture [8, 20], vocal characteristics [8, 34], facial expressions in image sequences or videos [53] or the pressure exerted on the buttons of an input device. The latter will be explained more in detail in *2.3.2 Measuring pressure in a gaming context*.

Finally, the self report approach entails measuring emotion or experiences through some form of subjective self assessment [34]. This could for example include thinking aloud while playing, reviewing footage of the gaming session retrospectively, performing an interview or filling out a questionnaire. It should be noted that self reports are retrospective rather than real-time (as opposed to physiological and some behavioral modalities), and think-aloud could potentially be disturbing the player's experience [17, 32]. While this would make self reports mainly unfit as direct input in an affective game, self reports can still be useful for determining the player's emotions and experiences with the game post gaming session, and this can be used to triangulate and verify the data or results obtained in experiments [39, 46, 48].

2.3 Related work

In the following subsections some of the previous work that has been done related to affective gaming will be presented and analyzed. The related work covers the implementation and evaluation of various affective games, as well as the evaluation and discussion of button pressure levels as an affective metric in a gaming context.

2.3.1 Affective games

While research has been done on how various modalities can be used to infer player experience, it would seem that there's a relative lack of research that discusses how to actually implement a game that uses these modalities in an affective loop. A literature review on adaption in affective video games found that setting a few criteria for paper selection, most notably requiring the presentation of a video game adapted by affect-based techniques, narrowed down their collected articles from an initial 300 to 14 suitable studies [5]. For this subsection, we have selected 8 studies that present their approach to implementing an affect loop in a video game. Table 2.1 gives an overview of the modalities, types of games and adaptive components that the studies have used.

Table 2.1: Overview of implemented affective games and their modalities.

Modality	**Game**	**Adaptive components**	**Source**
HR	Custom shooter	Difficulty, story, music	[33]
GSR	Custom racing game	Avatar speed	[4]
HR, GSR	Pong	Difficulty (paddle size, ball speed)	[13]
HR, GSR	Half-Life 2 (FPS)	Audio-visual effects, various gameplay components	[12]
Facial expression	Tetris	Difficulty (speed of tetrominoes)	[52]
RESP	Pacman, Bubble Shooter, custom "tilt to move" game	Various gameplay components	[55]
RESP, ECG, TEMP, GSR, EMG, Gaze	Custom 2D side- scrolling shooter	Various gameplay components	[38]
ICG, ECG, BVP, EMG, GSR, TEMP, Heart sound	Pong, custom anagram-solving game	Difficulty	[31]

Abbreviations used: BVP = blood volume pulse, ECG = electrocardiography, EMG = electromyography, GSR = galvanic skin response, HR = heart rate, ICG = impedance cardiography, RESP = respiration, TEMP = temperature

The first thing to note is that the modalities that are used to adapt the games mainly fall in the physiology category, with the main signals used being heart rate (HR) and GSR. This relates well to the previously stated fact that HR and GSR have been found to relate to arousal, which again is linked to the perceived difficulty of the game (see *2.2.1 Measuring emotions*). The arguments for using physiological modalities mainly consist of the research that has previously been done on the link

between in-game events and physiological responses [4, 13, 31].
However, one paper notes that their test participants found the physiological sensors to be distracting [12], and another paper chose to analyze facial expressions for the exact reason that no sensors needed to be in contact with the player [52]. It was also found that GSR may not be suitable for short, fast-paced gaming sessions [13], possibly because of a lack of significant change in the tonic response (the long-term component of a GSR signal).

The second thing to note is that although the papers use a wide array of games, they tend to stick to certain genres. All but one papers use either a shooting game or a remake of a classic arcade game. The arguments for choosing these genres tend to include having a variety of parameters to tweak [12, 33, 38] and the fast pace or stress-inducing nature of the gameplay [4, 31, 33, 55].
One paper tested an affective version of a game they classify as a horror FPS. They made the observation that those test participants who were unfamiliar with the FPS genre were also less engaged with the environment [12]. Furthermore, the test participants who noted that they did not enjoy the horror genre preferred the non-affective version over the affective version. This underlines the importance of using a game genre that is universal and easy to get into, so as to avoid alienating the players.

The third thing to note is that half of the papers concern themselves with adjusting the difficulty of the game dynamically to suit the player. This concept, called Dynamic Difficulty Adjustment (DDA), is meant to improve the gaming experience by ensuring that the game is at an appropriate difficulty level according to the individual player [14]. Maintaining this balance between challenge and skill evokes a state of flow where the player would neither be bored nor frustrated [10]. One paper also argues that an adaptive game difficulty can be blended in within the context of the game more easily than changing e.g. the virtual surroundings directly, so as to not disrupt the immersion of the player [33].

Finally, while the papers generally agree that affective gaming has potential, some also make some remarks about the actual applicability of affective games. One argues that while new forms of interaction in games may be more immersive, traditional interaction may be superior in terms of user preference or performance [38]. Another paper also noted that while their affective game had improved the user experience, it did not seem to improve the user performance [13].
Two of the papers that concern themselves with DDA furthermore comment that combining both in-game performance and affective metrics may be more optimal for making just decisions in changing game parameters than opting for only one of the two [31, 52].

2.3.2 Measuring pressure in a gaming context

As evident from the previous subsection, physiological modalities are prevalent in affective games; affective games with indirect feedback as opposed to direct feedback are a minority within the field. Furthermore, to our knowledge, no affective game has been implemented that relies on covert pressure on an input device to infer emotions and adapt the gameplay. This is despite the fact that studies that have explored pressure as an evaluation metric for game experience consider it to be a promising metric [23, 35]. In our own previous work on the subject, it was even concluded that in a test setup with two rounds of 5-minute gaming sessions, pressure was better at indicating arousal as opposed to HR and GSR [49].

The correlation between pressure exerted on an input device and the level of arousal has been discussed in multiple instances. Van den Hoogen et al. did a study in which they found a strong correlation between maximum pressure applied to a mouse in an FPS game and game experience aspects that relate to arousal (competence, frustration and challenge) [20]. Tijs et al. had test participants play Pacman at three different game speeds to induce different levels of arousal, and found that keyboard pressure could help distinguish between all three game speeds (slow, normal and fast) [47]. Sykes & Brown found somewhat similar results when test participants played Space Invaders with a gamepad at different difficulty levels. They found that the buttons were pressed significantly harder in the "hard" difficulty compared to the "medium" or "easy" difficulty [45].
Some studies have also looked into whether button pressure can indicate other aspects than arousal levels, and have found that it can relate both to the level of sensory immersion [19] and the type of in-game behavior that the player is currently engaging in [18]. One paper notes, however, that while they found that pressure related to the level of arousal that players experienced, they were unable to tell whether the arousal was positive or negative (i.e. no valence levels could be inferred) [45].

The type of games that are used in studies on pressure generally belong to the same game genres that other studies on affective games have previously used (regardless of whether they measured pressure or not). The best represented genre is classic arcade games, including both Breakout, Pacman and Space Invaders. These games, like the ones mentioned in *2.3.1 Affective games*, have various straightforward parameters that can be tweaked to manipulate the difficulty and have a fast paced gameplay.

As for how pressure data is gathered and treated, there have been various procedures. The input devices that have been used to evaluate pressure during gaming include:

- A keyboard rigged with Force Sensing Resistors (FSR) [18, 47].
- A mouse rigged with FSR [20].
- A Sony PlayStation 2 controller, either with FSR or using the native pressure sensitivity in the buttons [19, 45].
- A Microsoft Surface Pro 3 tablet, where the pressure level is inferred from the size of the contact area of the fingertip [35].

In general, there are two ways that the data is subsequently treated. One approach is to consider the maximum pressure value per key press and average that across all key presses [20, 35], while another is to include the full key press in the calculation of a mean pressure value (i.e. also the readings immediately before and after a peak) [18, 20, 35, 45, 47]. A single key press can be distinguished through e.g. peak detection algorithms or thresholds [47]. To avoid individual differences between participants to affect the data, papers also often normalize the pressure data per participant [18, 19, 47].

As a final note, although pressure has been shown to be a valid game-related metric in multiple cases, none of these cases have used pressure as real-time feedback in an affective game. Van den Hoogen et al. argues that in order to use pressure as an input in an affective game, the sensitivity of the measure would need to be established at an individual level, and at much shorter time-spans than what is the case for most of the referenced papers (seconds rather than minutes) [20].

2.3.3 Dynamic Difficulty Adjustment

Researchers have previously underlined the importance of presenting players with a proper level of challenge; it is a fundamental factor in game design [2]. An improper balance of challenge and skill can cause frustration or boredom in the player, potentially risking that they lose interest [54]. DDA attempts to overcome this issue by automatically adapting the game difficulty to fit the player's skill level, which may be one of the reasons why DDA has become such an active topic within the affective gaming research community [56]. Several of the papers that have already been presented also concern themselves with game difficulty, either dynamically adjusting the difficulty or presenting pre-determined levels of difficulty to participants in a test.

When designing a DDA system, there are certain considerations to keep in mind. The system should be designed in such a way that it can automatically track the

player's skill level and adapt to it, even as it rises or falls throughout a session [56]. It has also previously been argued that DDA is best done without the player's awareness, so as to not disrupt their immersion within the game [3] - players may even feel "cheated" if the game adjusts during the gaming session [21].
Furthermore, the system requires some measure of difficulty in order to map the levels of skill and challenge, which could for example be the players' current number of resources, losses or victories [56]. This measure can differ from game to game, as success conditions and relevant game aspects vary.

The literature offers various examples of implemented and tested DDA systems. Hunicke [21] developed a performance-based DDA in the FPS-game Half-Life 2. The system estimates a probability of player death based on the present enemy types, number of enemies and the damage done to the player. By predicting the likelihood of player death, the system knows when to intervene. Intervention can happen in the form of extra items (e.g. health packs and ammunition) or modifying player or enemy values (e.g. hit points and the strength of their attacks). Testing the system revealed that the system could indeed increase the enjoyment of players familiar with the FPS genre (while results were inconclusive in the case of novice players).
Sutoyo et al. [44] gives another example of a performance-based DDA. Their system was developed to adapt the difficulty of a tower defense game. Their system adapts the difficulty based on the player's and enemies' hit points, as well as which passive skills the player decides to invest their in-game skill points on. The difficulty is adjusted by changing the strength of enemies, amount of enemies per wave and the amount of in-game currency that the player receives upon clearing a level.

Another approach to DDA consists of doing adjustments according to a player's affective state rather than in-game performance. As an example, Dirrik & Lampropoulos [13] proposed and tested a DDA system that uses physiological signals (GSR and HR) to adjust the difficulty in an adapted version of the classic arcade game Pong. They found that while the user performance (measured in game score) did not improve with DDA, the user experience did.
Liu et al. [31] also proposed an affect-based DDA system, albeit with a wider variety of affective data. Their system recorded a wide variety of physiological signals (related to cardiovascular activity, muscle activity, skin conductance and skin temperature) and used the data real-time to adapt the difficulty of Pong and an anagram solving game. They concluded that most participants found the affect-based DDA game versions to be more satisfying and challenging compared to performance-based DDA game versions, and that the performance of the majority of the participants improved with the affect-based DDA. This is in contrast to the results from Dirrik & Lampropoulos, who found no difference in performance.

It should be noted that there are certain constraints to keep in mind when developing a DDA system. For example, the impact of DDA on the game experience may be dependent on the individual player's familiarity with a game genre. Alexander et al. [1] argues that while experienced players may enjoy the challenge, novice players may prefer the easier levels. This was also the case in a study by Hunicke [21], in which it was found that 'familiar players' had an increased enjoyment of a game with DDA, while no similar results were found for novice players.
Players' motivation for playing also poses an issue for DDA. Yun et al. [54] implemented a DDA system in a third-person 3D shooting game, and noticed that although one of their participants had been marked as an expert player, he played the game on the easy difficulty for most of the time. Since their data showed that he did indeed enjoy the game (despite it being too easy for his skill level), they note that his motive for playing could be classified as seeking easy fun (focusing on enjoyment of game activities) rather than hard fun (focusing on winning conditions). Alexander et al. [1] likewise argues that DDA is acting against players who want to play at their own pace - for example in the case of an expert player preferring the easy difficulty.

2.4 Research problem

Affective games seek to improve the game experience by adapting the gameplay according to the player's emotional state. The subjective game experience tends to be measured with specialized questionnaires post gaming session, while several different methods for measuring the emotional state during the gaming session have been proposed throughout the years. They can generally be categorized as either physiological, behavioral or self report approaches. These methods typically consider emotions in relation to the dimensions of arousal and valence.

An analysis of related work on affective games revealed that physiological approaches tend to be dominant, and that the existing work tends to use game difficulty as the adaptive component. Because difficulty is a frequently adapted component in affective games, related work on the concept of DDA was presented, which showed how DDA can effectively improve the game experience for the player.
No papers were found to use the pressure on an input device (a behavioral metric) in an implementation of an affective game, despite this metric consistently being shown to correlate with arousal and certain game experience aspects.

Based on the presented research, the research problem for this thesis has been formulated as follows:

How can pressure be used as indirect feedback in a DDA system?

In order to conclude upon this, it has been decided to design and conduct an experiment with the purpose of measuring and comparing the player experience of an affective game with pressure-based DDA against a regular performance-based DDA and no DDA.
Such an experiment should allow for a discussion on the success and effectiveness of this new type of DDA that infers the emotional state of the player from the pressure exerted on an input device.

3 Implementation

The implementation follows an iterative design process in which the different components of the prototype are designed, implemented and tested, with each test aimed at improving the next prototype iteration. The sections in this chapter are thus dedicated to describing the details of a new prototype iteration, with the subsections describing each component of the prototype as well as the test results for the specific prototype.

Before the implementation of any prototype, decisions had to be made on which game to use and how to implement the affective aspect.
Based on the background research, it was decided that the game should fall in the classic arcade game genre. Typical of games within this genre is that they are universal, easy to pick up and has simple controls. More specifically, it was decided to base the game off of *Arkanoid*, a classic arcade game by Taito from 1986 [51]. The concept of Arkanoid is similar to its predecessor *Breakout* by Atari Inc., which we have previously had success with using in relation to affective gaming [49].
The emotional state of the user will be inferred through the pressure they apply to the input device. Related work on pressure as a metric in affective games have tested on both computer mice, keyboards and gamepads with success. As an ordinary keyboard is likely a familiar tool for many, it has been decided to use a keyboard as the input device. Furthermore, the importance of abstracting from individual differences when analyzing the pressure data in the affective game has also been noted.
Finally, it was decided that the affective aspect of the game would be a DDA system that can change the difficulty based on the inferred emotional state of the user. Two requirements for the DDA system were formulated:

1. The difficulty should adapt continuously throughout a level at set time intervals rather than only in between levels.

2. The adaption should be hidden from the user.

The first requirement was set as each level in the game can potentially last several minutes. Therefore, it would be preferable to intervene several times during a level in order to maximize the time that the user spends in the flow channel, rather than being restricted to making changes only between levels. The second requirement was set based on what is known to be the best practice of DDA (as described in *2.3.3 Dynamic Difficulty Adjustment*). By hiding the adaption from the user, the DDA does not risk disrupting the player's immersion in the game.

Two DDA systems were developed, one being based on pressure and the other being based on performance. To obtain the data for the pressure-based DDA system, a custom pressure reader was built. Furthermore, an adapted version of Arkanoid was developed in the Unity game engine.

3.1 The pressure reader

As the game will be controlled with a keyboard, the pressure reader should be able to read the pressure applied to a keyboard and send this information to Unity to be used by the game to perform pressure-based DDA. The data should also be saved for later data analysis.
The pressure reader was built using an Arduino Uno single-board micro-controller and a 0.5 inch round Force Sensing Resistor (FSR). The sensor is connected to the Arduino with a 10kΩ pull-down resistor and 5V, as shown in the circuit schematic in Figure 3.1. The signal is sampled with a frequency of 20 Hz.

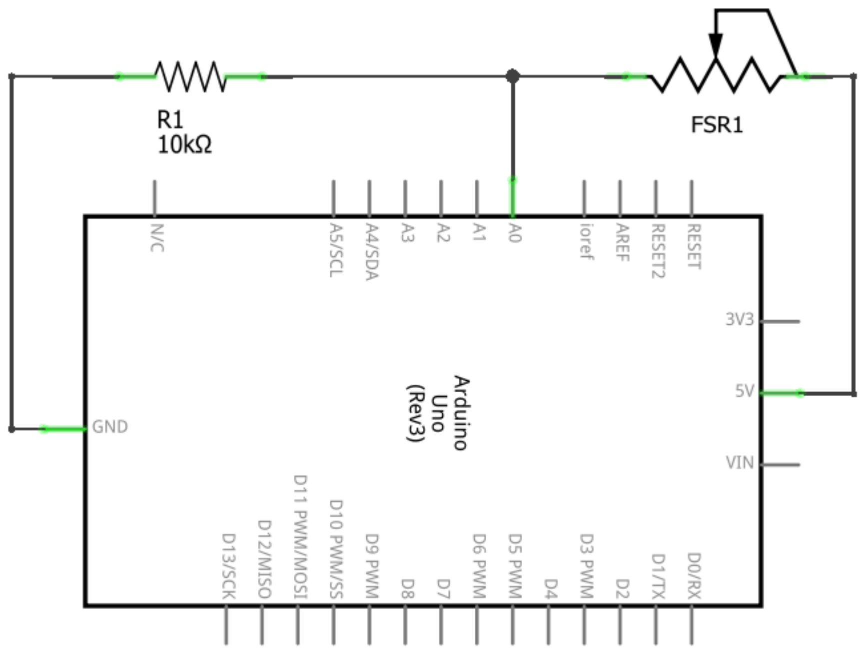

Figure 3.1: The circuit schematic of the pressure reader.

It was decided that the FSR should be placed under the foot of the keyboard closest to the arrow keys, as the arrow keys would be used for controlling the game. By placing the FSR underneath the keyboard, the sensor would be hidden from the player, thus reducing the possible change in behavior that could occur due to an awareness of the sensor. The keyboard is a "Magic Keyboard with Numeric Pad" by Apple, as this keyboard is made out of aluminum. This allows for better readings with the FSR, since the force of key presses is not dampened by the bending of a plastic frame.
The FSR works by decreasing its resistance as the force applied to the active surface is increased. However, this technology is susceptible to differences in the placement of force on the active surface. This means that consistency in the placement of the sensor and keyboard is very important in achieving comparable results. Therefore, a "placement board" was designed and cut out from foam board to fit precisely around the bottom of the keyboard, with a drawn-on silhouette of where the FSR should be placed. With this setup, also illustrated in Figure 3.2, the placement of the sensor in comparison to the foot of the keyboard remained constant. As the placement board raised the front of the keyboard by 0.5 cm, the weight of the keyboard was also more evenly distributed across the feet, resulting in more precise readings from the FSR.
After finalizing the design for the placement board, the board was cut out from plywood for additional sturdiness.

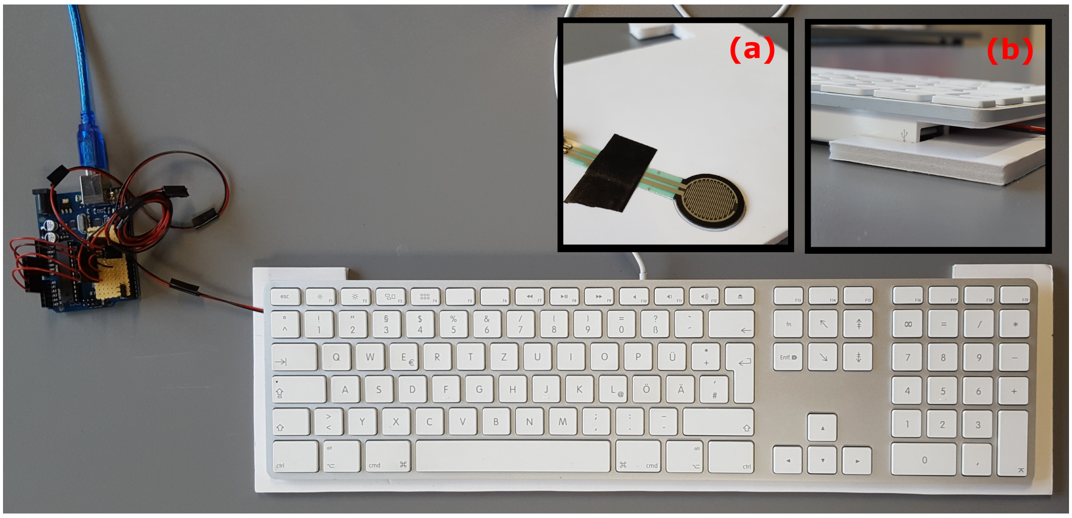

Figure 3.2: The setup of the pressure reader, keyboard and placement board. Picture (a) shows how the pressure reader is placed underneath the keyboard, and picture (b) shows how the keyboard fits into the placement board.

To use the pressure data during a gaming session, the Arduino Uno was connected via a USB port to a PC. Unity could then during runtime establish a connection

with the Arduino through the serial port and read the data in the serial port buffer that Arduino had sent. The data was kept compact by mapping the signal values from the original range of 0-1023 down to 0-255 before sending the data to the serial port. This mapping meant that each reading would not exceed 1 byte.
After Unity has obtained the latest analog reading form the Arduino, it is converted to conductance measured in $\mu\Omega$. The voltage that the Arduino has read is first mapped between 1 and 5001, then converted to resistance and finally converted to conductance:

$$resistance = \frac{(5001 - voltage) \cdot 10000}{voltage}$$

$$conductance = \frac{1000000}{resistance}$$

The conversion to conductance allows for the data to be interpreted on a linear scale due to the approximately linear relation between conductance and force when the force is between 0-1000 grams. This is not the case for the relation between the resistance and force, which instead closely resembles a logarithmic relation. These relations are also illustrated in Figure 3.3.

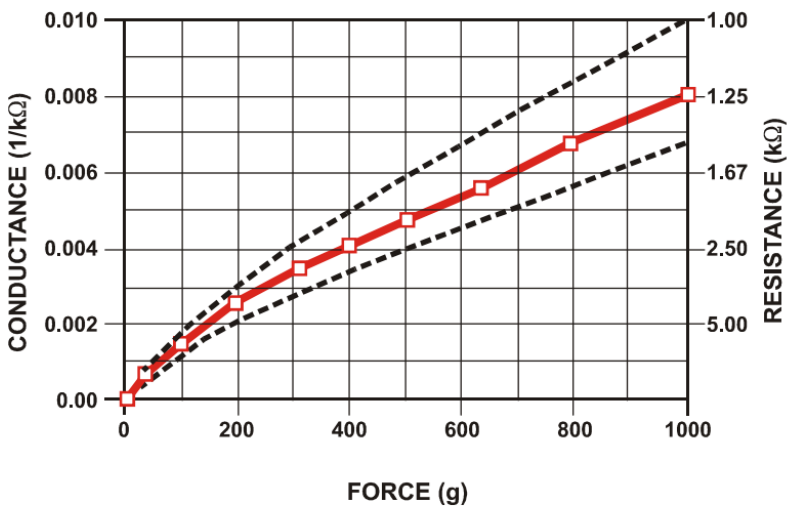

Figure 3.3: The relations between conductance, resistance and force, as presented in the data-sheets of the FSR [16]. The dashed lines represent the typical part-to-part error band.

After the conductance has been obtained, Unity saves it as a new data entry (consisting of the conductance value and whether either the left or right arrow is currently pressed) to a .txt-file for later data analysis. This data entry is also temporarily stored in the game to be used by the pressure-based DDA when necessary.

Finally, Unity can establish a UDP connection to a program written in Processing with the purpose of easy visualization of the data. This allows the test facilitators to visually inspect the data during tests to ensure that the setup is working as intended. By connecting a monitor to the PC running the game, the data visualization can be shown on a separate screen only to be visible to the test facilitators. Figure 3.4 shows an example of the data visualization.

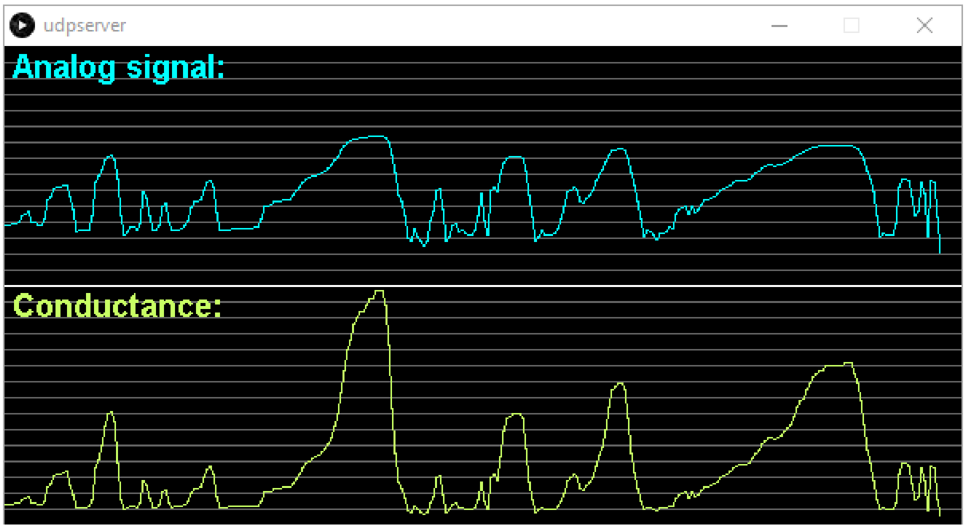

Figure 3.4: An example of the data visualization window. The data that was sent by Unity has been mapped to appropriate ranges in Processing to display the data in the confined space of the window.

3.2 First prototype iteration

The goal of the first prototype iteration was to implement a functional game version and implement communication between the game and the pressure reader. These components would allow for the collection of data on various aspects that would be needed in order to implement the pressure-based and performance-based DDA systems in later iterations. Furthermore, any issues related to design choices or implementation of the game could be corrected early in the process.

3.2.1 Game

The game, based off of the classic arcade game Arkanoid, was implemented in Unity, and largely follows the rules of the original game. The graphics and sounds are borrowed from a freeware ball-and-paddle game called *DX Ball*. Figure 3.5 shows a screenshot from the game implementation.

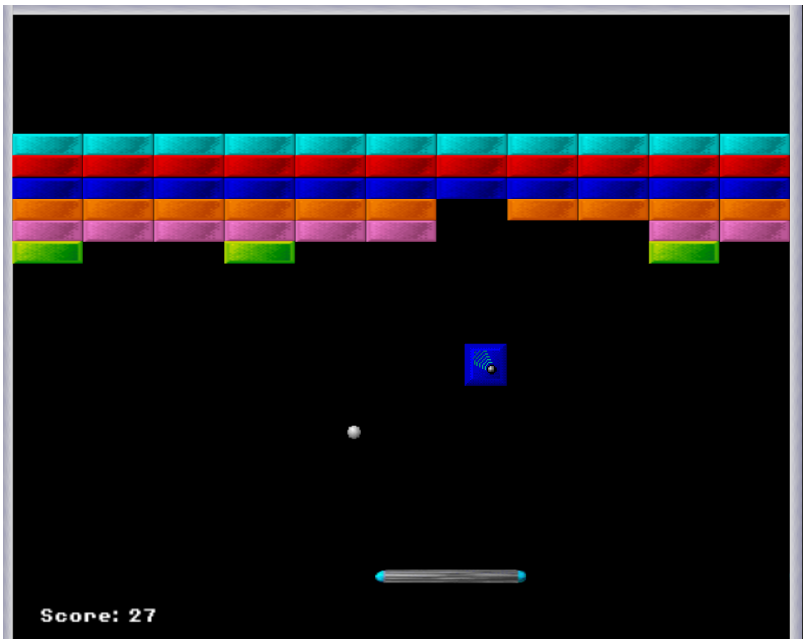

Figure 3.5: A screenshot from the developed brick breaker game. The blue square is a power-up falling towards the bottom of the screen.

The player can move the paddle at the bottom of the screen to the left or right with the left/right arrow buttons on the keyboard. If no ball is currently in play, they can launch a ball from the paddle with spacebar. Their task is then to bounce the ball between the paddle and the colored bricks, and each time a brick is hit, it is destroyed and the player's total score increases. To clear a level, all bricks must be destroyed. 10 of the original levels have added, but all special bricks (a feature of the original game) have been converted to regular bricks. It was decided not to feature special bricks, as the different level layouts are meant to add some variety to the gameplay rather than affect the perceived difficulty.

Figure 3.6 illustrates examples of how the ball's direction changes depending on the paddle bounce. The new direction of the ball is calculated as a percentage distance from the middle of the paddle, such that the middle of the paddle is a value of 0, the far left a value of -1, and the far right a value of 1. The "cap" zones mean that the calculated percentage distance from the middle can never be lower than -0.75, higher than 0.75, or anywhere in between -0.15 to 0.15 (in which case it will instead be rounded towards the closest of the two values). This ensures that approximately straight horizontal or vertical movement will not occur. To sum up, the further from the middle of the paddle the ball hits, the more horizontal the resulting direction will be. Additionally, the horizontal direction of the ball depends on which side of the middle of the paddle the ball hits.

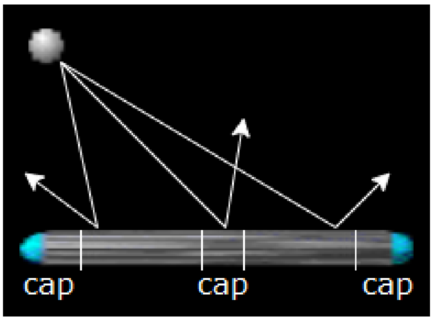

Figure 3.6: Examples of how the ball's direction changes depending on where on the paddle it bounces off of.

The game also includes three pairs of drops that consists of a positive and negative effect each. If a drop spawns, it will fall towards the bottom of the screen, and the player can then catch it with their paddle to activate its effects. Power-ups are signalled as positive with a blue background, and power-downs are signalled as negative with a red background. Table 3.1 presents an overview of the drops and their effects. The three pairs of drops can either impact the paddle size, ball size or ball speed. These three parameters have five settings and will always start at the neutral setting at the beginning of a new game. Picking up a drop increases (power-up) or decreases (power-down) the setting level by one, with a maximum of two levels from neutral.

Table 3.1: The effects of drops on different game parameters.

	Paddle size	**Ball size**	**Ball speed**
Setting -2	30%	30%	170%
Setting -1	65%	65%	135%
Setting 0	100%	100%	100%
Setting +1	135%	135%	65%
Setting +2	170%	170%	35%

This drop system is also used to aid the player if they fail to catch the ball. When the ball is dropped, the player must re-launch the ball from the paddle, with only minus points to their game score as a negative consequence. The player is also given a single power-up to decrease the difficulty slightly, as dropping the ball is seen as an indication that the game might currently be too difficult. Which power-up to give is based on which game parameter is currently most severe.

When deciding on which game parameters should be manipulated by the DDA system, two options were considered: Changing the ball speed or changing the chances of power-ups versus power-downs. It was decided to manipulate the latter, as this was argued to be the easiest manipulation to hide from the player.

Five difficulty settings were implemented, as seen in Table 3.2. Note that the chances in the table are not the chances of a drop spawning, but rather the chances of a drop being positive or negative when it has already been decided to spawn a drop. To ensure that the player will be able to pick up at least some of the drops, the bricks start with a 0% chance of spawning a drop upon breaking. This chance is then increased by 7.5 percent points for each brick that is destroyed until the player picks up one of the drops, which resets the chance of a drop spawn back to 0%.

Table 3.2: Different difficulty settings and the chances of power-ups versus power-downs.

Difficulty setting	Chance of power-up	Chance of power-down
Very easy	90%	10%
Easy	70%	30%
Medium	50%	50%
Hard	30%	70%
Very hard	10%	90%

3.2.2 Pressure-based DDA

The pressure-based DDA system should be able to analyze the amount of pressure that the player applies to the keyboard and use this information to increase or decrease the difficulty. The system should be able to do so without individual differences (e.g. muscle strength or general tendencies to press lighter or harder) influencing the decision.

It was decided to analyze the pressure data by calculating the linear regression based on the recorded button presses. The slope of the regression line thus serves as an indication of increased or decreased arousal. In this manner, the individual differences in general pressure levels should not have a major impact on the data analysis, as only the pressure change over time is considered.
The data would be split into windows of 20 seconds each. At the end of a window, the pressure-based DDA system should calculate the regression line for the recent 20 seconds. If the slope is positive and above a threshold, the difficulty should decrease (as a pressure increase indicates that the game is getting harder), while if the slope is negative and below a threshold , the difficulty should increase (as a pressure decrease indicates that the game is getting easier). If the slope is within the upper and lower thresholds, the difficulty will remain the same.

The groundwork for this system design was implemented in the prototype, but in order for the system to be usable, data from the technical evaluation is required in order to decide on realistic upper and lower thresholds for the difficulty change.

3.2.3 Performance-based DDA

The performance-based DDA system should be able to quantify the performance of the player throughout the game and use this information to increase or decrease the difficulty. In *2.3.3 Dynamic Difficulty Adjustment*, related work on performance-based DDA systems that offer some examples of different designs for performance-based DDA was presented. It was decided to create a "performance score" to evaluate, inspired by the performance-based DDA for tower defense by Sutoyo et al. [44].

A good performance in the implemented brick breaker game is defined as "breaking all the bricks in a level as fast as possible". Based on this definition, five key events were identified and given a scoring amount. These included hitting a brick (+1), hitting a wall or roof (-0.5), dropping the ball (-20) and collecting a power-up (+1) or power-down (-1). Similar to the pressure-based DDA design, the data would be split into windows of 20 seconds each. At the end of a window, the performance score for the recent 20 seconds would be evaluated against some thresholds. If the score is below the lower threshold, a difficulty decrease is triggered, while if the score is above the upper threshold, a difficulty increase is triggered. If the score is within the upper and lower limits, the difficulty remains the same. After evaluating the performance score, the score is reset to 0, so each 20-second window is an isolated case.

Similar to the pressure-based DDA system, the framework for this performance-based DDA was implemented, but the system now required data from the technical evaluation in order to determine proper scoring thresholds and scoring values.

3.2.4 Technical evaluation

The first prototype allowed for early data gathering and an analysis of the data based on how the DDA systems were currently designed. By analyzing the data in the same manner that the DDA systems would (although post gaming session), we can estimate how well the systems might act and adjust. Furthermore, the difficulty settings in the game could be tested in relation to how they affect the perceived difficulty.

Gathering data for the pressure-based DDA system

The purpose of this test was to gather data on the pressure that participants applied throughout the game in order to run an initial data analysis and assess how well the current design of the pressure-based DDA might perform. Furthermore, the data could be used to set an initial upper and lower bound for a difficulty change in the pressure-based DDA system.

In the test, the game was altered to either increase or decrease all three game parameters (ball speed, ball size and paddle size) every 20 seconds in order to simulate a difficulty change. All participants (n=7) played the first level from start to finish with the game parameters fluctuating throughout.
After the tests, the data was imported into Matlab for data analysis. The initial idea was to calculate a regression line based on the pressure values 10 seconds prior to and 10 seconds following a difficulty change. The expected outcome was a positive slope value in case the difficulty increased, and a negative slope value in case the difficulty decreased. Unfortunately, there turned out to be no distinct pattern in such slope values. Another attempt was made in which the regression lines were calculated for the data within each 20 second window to see whether there were any patterns in slope values based on which difficulty the participants played at. This resulted in a similar issue where slope values were positive and negative with no noticeable pattern. While other methods of splitting up the data was attempted, no proper results came from calculating the regression lines.
However, it was found that the mean value of the pressure in each 20 second window was dependent on the parameter settings that were active during the given window. Lower parameter settings resulted in a lower mean value, and vice versa. This was generally the case regardless of whether the value of a single button press was determined from the maximum reading or the average reading of the whole button press.

Performance score behavior
A part of the initial goal of the first prototype iteration included gathering some data for the performance-based DDA system. However, before starting such tests, it was found that there were some fundamental design issues in the performance score design, as the score generally behaved very different depending on how many bricks were left to hit.
Due to this, a different design for the performance-based DDA was suggested, in which the performance is based only on how many bricks can be expected to be broken with the given ball speed and how many bricks are currently left. This idea is further elaborated upon during the second prototype iteration.

Testing for significant differences in perceived difficulty
The purpose of this test was to determine whether the current difficulty settings could be said to be significantly different. In the test, each participant played level 1 in the game from start to finish three times. Each playthrough had a different difficulty setting (either "very easy", "medium" or "very hard", as described in Table 3.2). After finishing a level, the participant was asked to rate the difficulty on a 7-point Likert scale from -3 to 3 (-3 = very easy, 0 = optimal, 3 = very hard). The order of the difficulties was counterbalanced to avoid confounding variables such as the initial learning curve when playing.

A set of Wilcoxon Signed-Rank tests (n=6) were used to test for significant differences in perceived difficulties between the difficulty settings. The results of these tests are presented in Table 3.3.

Table 3.3: The results of Wilcoxon Signed-Rank tests on the perceived difficulty.

Compared difficulty settings	p-value
Very easy vs. medium	0.156
Medium vs. very hard	0.031
Very easy vs. very hard	0.031

With a significance level of 0.05, no significant difference was found between the very easy and medium difficulty settings. From observations made during the test, this seemed to be due to the fact that power-downs were easy to dodge in both the very easy and medium difficulty settings, while this was not the case in the very hard difficulty setting (due to the sheer amount of power-downs spawning).

3.3 Second prototype iteration

The goal of the second prototype iteration was to improve upon the drop system from the first prototype, as it was found that the difference in perceived difficulty between the difficulty settings was only significant in two of the three cases that were tested. Furthermore, planned changes to the game design as well as a new design for the performance-based DDA system required new data to be gathered in gaming sessions. This data could then be used for a data analysis related to the further implementation of the DDA systems.

3.3.1 Game

When testing the first prototype, it was found that the power-downs were somewhat equally easy to dodge in both the very easy and medium difficulty settings. This meant that the perceived difficulty between those two difficulties could not be said to differ significantly. Due to this, it was decided to re-design the drop feature.

Instead of moving at a constant speed towards the bottom of the screen, the drops now follow the paddle at a continuously increasing speed until they hit the paddle, thus activating their effect. This means that the drops are now forced pick-ups. Furthermore, it was decided to remove the drops that affect the size of the ball and paddle. As the new performance-based DDA system only consider the ball speed and not the size of neither the ball nor paddle, the drops were irrelevant in that regard. Furthermore, removing the size-changing drops granted a better control of the difficulty of the game, as the impact of each type of drop on the perceived difficulty was otherwise unknown. With only speed-up and speed-down remaining, any changes in perceived difficulty should be explainable from the single parameter (ball speed) that is known to change.

With the new design of the drop feature, the consequences for dropping the ball were also changed, as automatically granting a speed-down was deemed too forgiving. Instead, when dropping the ball, the penalty is a wait-time of 1.5 seconds before the ball respawns as well as minus points to the player's total score. The speed of the ball remains the same as it was prior to dropping it.

3.3.2 Pressure-based DDA

During the technical evaluation of the first prototype, it was found that the regression lines of the pressure that was applied during the gaming session did not behave as initially expected. In contrast to this, the mean pressure values of each difficulty setting were showing a pattern in which harder difficulties resulted in a higher mean pressure.
As the game was reiterated upon for the second prototype, it was decided to do another round of data collection concerned with pressure values during play. An analysis of the new data should reveal whether the regression lines continue to be too unreliable to use for DDA. Thus, no changes to the pressure-based DDA system design were done for the second prototype.

3.3.3 Performance-based DDA

A good performance in the game was previously defined as "breaking all the bricks in a level as fast as possible". However, the performance score that was designed was difficult to balance and behaved differently depending on how many bricks were left in the level, as players tend to miss bricks more often when there are fewer. This meant that a player's performance score would typically decrease the further they got in the level, regardless of how well they were actually performing.
With this in mind, a new design for the performance-based DDA system was developed. Rather than an absolute score based on actions and events, this system uses a mathematical function to estimate the player's performance. The function is calculated from a curve that describes how many bricks can be expected to be broken within a certain amount of seconds based on how many bricks are left in the level. To obtain such a curve, data from gaming sessions was required. Therefore, the game was altered to collect the type of data necessary for constructing the plot, such that the data could be collected during the next technical evaluation of the prototype.

At the end of a 20-second window, the amount of bricks that could be expected to be destroyed can then be compared with the amount of bricks that were actually destroyed. If the difference is above some upper threshold, the player has performed better than what was expected, and the difficulty should be increased. If the difference is below some lower threshold, the player has performed worse than expected, and the difficulty should be decreased. If the difference is within the upper and lower thresholds, the difficulty should remain the same.

3.3.4 Technical evaluation

As some changes had been made to the game for the second prototype, it was decided to gather new data for evaluating the pressure data to be used for pressure-based DDA, as well as test whether the new drop system changed the perceived difficulty of the game. Furthermore, the new performance-based DDA system design required data to base the performance estimation curves on.

Gathering data for the pressure-based DDA system

Previously, it was found that the regression lines for the data did not show any persistent pattern, no matter whether the data was analyzed per difficulty setting (very easy/medium/very hard) or per difficulty change (increase/decrease). However, the mean pressure values revealed a pattern in which higher pressure values were observed in higher difficulties. As changes had been made to the game, it was decided to collect and analyze new data to determine whether to use regression lines or mean pressure values for future iterations.

For the test, the game was altered to change the difficulty (ball speed) every 30 seconds. The window size had been changed from 20 seconds to 30 seconds to determine whether more data per window might improve the later analysis result. The game always started at the "medium" difficulty setting, and then either increased to "very hard" or decreased to "very easy" before switching direction. All participants (n=4) played the first level from start to finish with the ball speed fluctuating throughout.

In a later analysis of the pressure data, the slopes of the regression lines were still found to be irregular, both when analyzing per difficulty setting and per difficulty change. However, as before, the mean pressure values were clearly higher at higher difficulties.

Gathering data for the performance-based DDA system

The new design for the performance-based DDA system required some data to be able to calculate a performance estimation curve. The purpose of this test was thus to collect the required data for such a curve.

The most optimal levels to include in the game would be ones in which a plot of bricks broken as a function of bricks left does not include spikes that are difficult to describe through curve fitting. To find the most optimal game levels, the research team played each level of the game, and the results were then averaged and plotted. Figure 3.7 shows the averaged performance plots for two of the game levels. The layout of the level clearly influences the shape of the corresponding performance plot - as an example, the large peak in level 1 around 30-40 bricks remaining was found to be a result of maneuvering the ball above the rows of bricks to bounce it between the bricks and the top of the level, which is a typical strategy observed to be used by previous test participants. In contrast, level 6 consists of 6 separated columns of bricks, which makes such a strategy difficult.

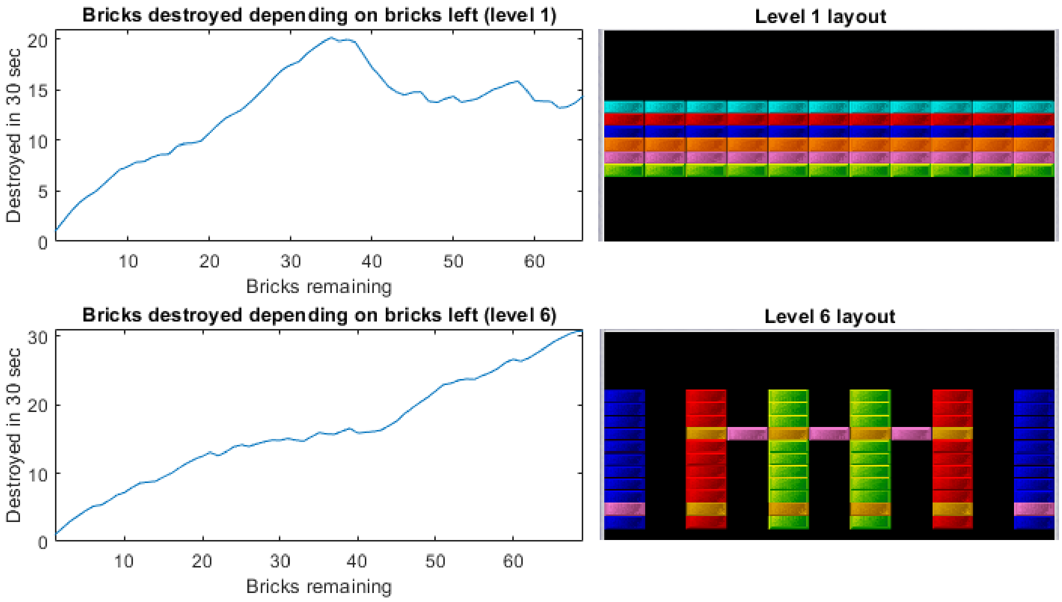

Figure 3.7: The averaged performance plots of level 1 and 6.

As the performance plots of several levels were approximately linear, a linear fit was calculated for each of the plots, and based on the r^2 values of the linear fits of each level, the most optimal levels were found to be level 6, 8 and 10 ($r^2 > 0.97$ in all three cases).

To collect data for the final performance estimation function, a test was done in which each participant (n=8) got to play all three levels in a counter-balanced order at the medium speed setting. After each level, the participants were asked to rate their own gaming experience level on a Likert scale from 0 to 5 (0 = never, 1 = less than once a year, 2 = a couple of times a year, 3 = monthly, 4 = weekly, 5 = daily). The performance data gathered during this test could then be used to calculate a performance estimation function for level 6, 8 and 10 based on the medium speed of the ball.

Testing for significant differences in perceived difficulty

The purpose of this test was to determine whether the difficulty settings in the new difficulty system could be said to be significantly different. Compared to previously, the difficulty now only changed the speed of the ball, rather than changing

the chances of power-ups vs. power-downs.
In the test, each participant played level 6, 8 and 10. In each of the levels the ball had a different difficulty setting (either "very easy", "medium" or "very hard"). The order of the levels and difficulties were counter-balanced such that each participant got to play each levels and each difficulty setting once, but in different combinations.

After finishing a level, the participants were asked to rate the difficulty on a 7-point Likert scale (as described in *3.2.4 Technical evaluation*). Performing a set of Wilcoxon Signed-Rank tests (n=9) resulted in the p-values seen in Table 3.4.

Table 3.4: The results of Wilcoxon Signed-Rank tests on the perceived difficulty.

Compared difficulty settings	p-value
Very easy vs. medium	0.031
Medium vs. very hard	0.007
Very easy vs. very hard	<0.001

With a significance level of 0.05, the new difficulty system can be said to have successfully achieved the goal of having three significantly different difficulties.
It should be noted that some participants expressed that the highest difficulty setting wasn't high enough for their experience level, and that some participants were observed to never drop the ball no matter the difficulty, suggesting that the difficulty setting might not have been high enough. Furthermore, several participants expressed annoyance at the very slow speed of the ball in the lowest difficulty setting. The experience level of the participants lie in the high end of the scale, with 55.6% playing games daily, 33.3% playing weekly, and 11.1% playing monthly.

3.4 Third prototype iteration

The goal of the third prototype iteration was to finalize the design and implementation of the pressure-based and performance-based DDA systems to bring the prototype to a final version in which both DDA systems have been iteratively designed, implemented and tested to a satisfying degree.
Based on the knowledge from the previous technical evaluations, the pressure-based DDA system was redesigned to use mean pressure values rather than the slope of the regression line calculated from button presses. Additionally, as the second prototype was used to gather data for calculating performance estimation functions, the performance-based DDA is at a point in the implementation where it is ready to be tested as a functional system.

3.4.1 Game

When testing the second prototype iteration, it was found that the perceived difference in difficulty between the lowest, medium and highest settings was significantly different. However, observations of or direct comments from test participants also suggested that the highest difficulty setting might not have been sufficient, and that the lowest difficulty setting was frustratingly slow. It was therefore decided to increase the overall speed of the ball slightly, as well as add two difficulty settings in the high end. This resulted in the difficulty settings presented in Table 3.5.

Table 3.5: The new difficulty settings and the corresponding ball speed. The previous names of the difficulties have been included for reference.

Previous name	Difficulty setting	Ball speed
Very easy	-2	10
Easy	-1	15
Medium	0	20
Hard	1	25
Very hard	2	30
-	3	35
-	4	40

As the difficulty in the game is now expressed in terms of ball speed rather than drop chances, the drop spawns were also changed from being based on chance percentages to being based on feedback from the implemented DDA systems. The DDA systems evaluate the player based on the most recent window of data, and from that determines a new difficulty. While this change in difficulty would optimally happen at the end of a window (i.e. every 30 seconds), this would make the DDA feature difficult to hide from the player, as is best practice, because a drop would spawn in response to seemingly nothing. Instead, after 30 seconds has passed, the game will spawn a speed-up or speed-down whenever the next brick is broken, which also marks the beginning of a new window.

The available levels in the game were restricted to be level 6, 8 and 10, as the layout of these levels had previously been found to result in a relatively linear performance plot. Aside from these ordinary game levels, two calibration levels were added to be used for calibrating the pressure-based DDA (described more in detail in the next subsection, *3.4.2 Pressure-based DDA*). The layouts and names of these final levels can be seen in Figure 3.8.

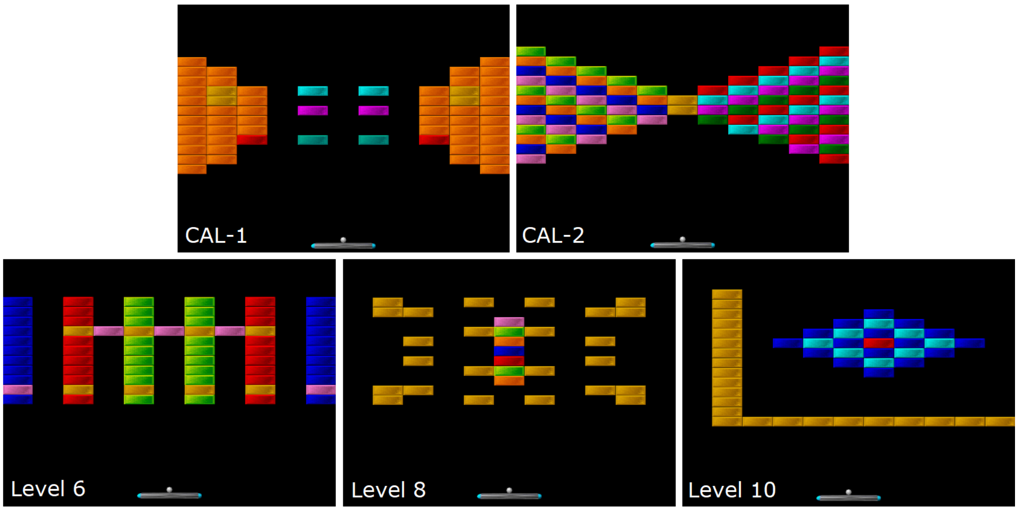

Figure 3.8: The final level layouts used in the game with their names annotated in the lower left corner. The calibration levels were added to be used by the pressure-based DDA, while the three ordinary game levels were picked from the original game levels in Arkanoid based on the linearity of their performance plots.

3.4.2 Pressure-based DDA

As the technical evaluation for both the first and second prototype had uncovered issues with using the regression lines for data analysis, it was decided that the pressure-based DDA system should instead use mean pressure values in the data analysis. However, this poses the issue of individual differences affecting the data, meaning that a global threshold for difficulty changes would be very impractical. Instead, it was decided to add two calibration levels to the game.

System design: Proactive and reactive

Two possible system designs were created to be tested. The first design, dubbed 'the proactive design', was designed to always spawn a speed-up at the end of a window, unless the player's pressure is above the upper threshold (signalling that the game had become too difficult), in which case a speed-down is spawned instead. The second design, dubbed 'the reactive design', would not change the difficulty of the game unless the player's pressure falls below a lower threshold (signalling that the game had become too easy), in which case a speed-up is spawned. After implementing both designs, it was decided to conduct a test during the technical evaluation to analyze how well the two system designs were working and which system design to base the final pressure-based DDA system on.

System calibration

The first calibration level (CAL-1) is used to calculate the baseline pressure for the individual player. Before starting the game, the player is prompted to choose either easy, medium or hard difficulty. The game then starts at the chosen difficulty

setting in CAL-1, and after 30 seconds, the player is again prompted to choose a difficulty. The second prompt was added due to the fact that the player would only have been able to base their first choice on the words "easy", "medium" or "hard" and their own perception of their skill level. For the second prompt, the player has had time to try their selected difficulty, and should be able to make a better informed choice if the first choice did not fit with their experience level. Throughout CAL-1, no drops are spawned.
The pressure for each button press is recorded (the average pressure value of all readings making up a single button press), and these pressure values can be averaged to get the mean pressure value for a single button press. This can be said to be the baseline pressure, as the difficulty was chosen by the player themselves, and it should therefore be somewhat representative of the pressure they exert when within their own comfort zone.

The second calibration level (CAL-2) is used to calculate a threshold for when the difficulty should change. In this level, the player starts out in their previously chosen difficulty. When a full window has passed, the next brick spawns a speed-up (which, during calibration, increases the difficulty setting by 2 rather than 1). After the window with the increased difficulty is finished and a brick is broken, a speed-down is spawned (decreasing the difficulty setting by 2 rather than 1). The whole level is played in this up-down manner. The reason for the enhanced effect of drops in this level is that the difficulty increase should purposefully force a sudden, steep difficulty change upon the player.
The system then averages the pressure that the player exerts at their chosen difficulty and at the increased difficulty. By calculating the difference between the two, the system obtains a value which represents the amount of change in pressure that can be expected when the game is too difficult. This is referred to as the upper threshold, which is used for the proactive system design. To get a lower threshold for the reactive system design, CAL-2 starts with a speed-down instead of a speed-up. At the end of the level, the difference between the baseline and the average pressure at a decreased difficulty can be calculated to obtain a lower threshold rather than an upper threshold.

3.4.3 Performance-based DDA

During the previous prototype iteration, a test was done in which some initial data for calculating the performance estimation function was collected. This allows for the actual implementation and testing of the performance-based DDA system to be carried out.
The performance of the test participants at difficulty setting 0 was averaged (n=8), and the resulting data for each level was plotted along with trend lines and their r^2 values. This resulted in the plots presented in Figure 3.9.

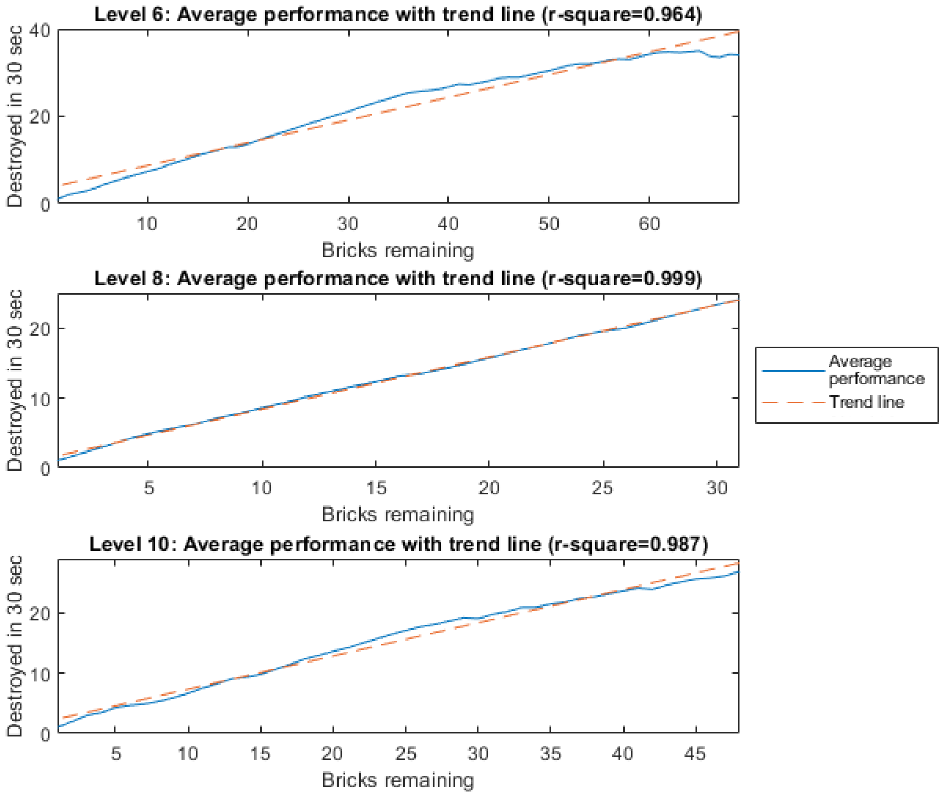

Figure 3.9: Plots of the averaged performance data (blue continuous line) of test participants at difficulty setting 0 in the three tested levels, along with the trend line (orange dashed line) for each level.

The performance estimation function for each level is thus the function describing the trend line. However, the current performance estimation functions are only calculated from difficulty setting 0, and therefore do not take the ball's speed into consideration. To include this parameter, it was hypothesized that increasing or decreasing the slope of the trend line with a percentage corresponding to the change in ball speed between difficulty settings would approximately fit with performance data for different speed settings. A ball speed change from 20 to 10 should for example correspond to a 50% lower slope value for the performance estimation function.
To test this, the data that was gathered to test for significant differences in perceived difficulty during the second prototype was used. This data includes 3 playthroughs for each each level at three different difficulties (difficulty settings -2, 0 and 2). By altering the slope of the performance estimation function according to the ball speed at difficulty settings -2 and 2, three different functions were obtained for each level, and these were plotted as trendlines as seen in Figure 3.10.

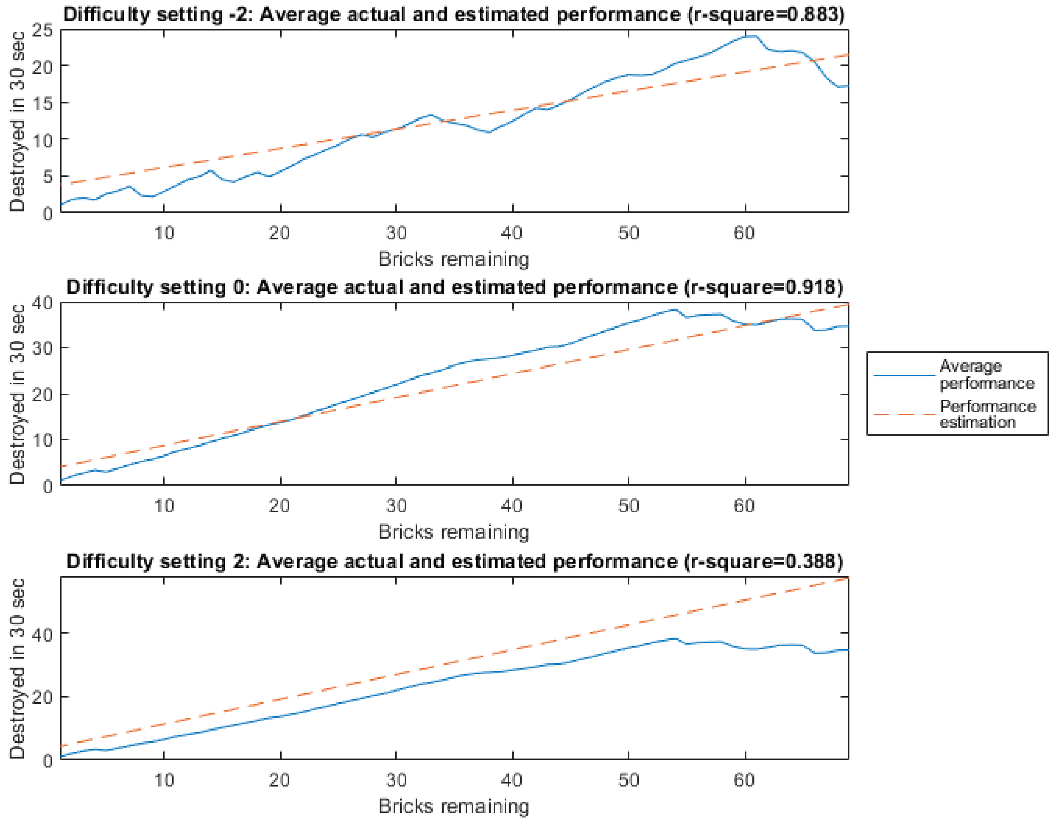

Figure 3.10: The averaged performance data (blue continuous line) for level 6 at three different difficulty settings with a performance estimation function (orange dashed line) plotted for each. The performance estimation function for difficulty setting 0 has been calculated from previously collected data, while for difficulty settings -2 and 2 it has been estimated based on the ball's speed in relation to difficulty setting 0.

While the performance estimation function had an acceptable r^2 in the case of difficulty setting 0, this was not the case for difficulty settings -2 and 2. This meant that in order to include the difficulty setting as a parameter in the performance estimation, a unique performance estimation function had to be calculated for the different difficulty settings. As data for difficulty settings -2 and 2 had been obtained in previous tests, it was possible to calculate initial performance estimation functions based on a smaller sample of data already. This resulted in the functions presented in Table 3.6. The functions for difficulty setting 0 are based on 8 playthroughs each, while the functions for difficulty settings -2 and 2 are based on 3 playthroughs each.

It was decided to use these functions to test the performance-based DDA system in the technical evaluation, and to subsequently gather more performance data to calculate improved performance estimation functions for the final prototype. For

Table 3.6: The initial performance estimation functions for the different levels and difficulty settings.

	Difficulty setting -2	Difficulty setting 0	Difficulty setting 2
Level 6	f(x) = 0.34x + 0.40	f(x) = 0.57x + 2.67	f(x) = 0.72x + 1.58
Level 8	f(x) = 0.30x + 2.69	f(x) = 0.75x + 0.95	f(x) = 0.91x + 0.76
Level 10	f(x) = 0.27x + 2.31	f(x) = 0.55x + 1.84	f(x) = 0.70x + 3.41

difficulty settings not included in Table 3.6, an approximation was calculated from the available data. For difficulty setting 4, the slope and offset were calculated as in the following formulas (with the difficulty settings annotated as subscripts):

$$Slope_{-4} = \frac{(slope_0 - slope_{-2}) + (slope_2 - slope_0)}{2}$$

$$Offset_{-4} = \frac{offset_{-2} + offset_0 + offset_2}{3}$$

For difficulty settings -1, 1 and 3, a performance estimation function was calculated by averaging the functions for neighboring difficulty settings.

To obtain upper and lower thresholds for difficulty changes (as described in *3.4.3 Performance-based DDA*), the average positive and negative deviation from the trend line for the different levels and difficulty settings was calculated. In the same manner as with the performance estimation functions, thresholds for difficulty settings -1, 1, 3 and 4 were approximated from the data available for difficulty settings -2, 0 and 2.
The proactive and reactive system designs described for the pressure-based DDA system (see *3.4.2 Pressure-based DDA*) would also be possible to implement for the performance-based DDA. It was decided to choose the design based on the technical evaluation of the pressure-based DDA that was already planned, such that the two DDA systems would use the same strategy for changing the difficulty based on different types of data.

3.4.4 Technical evaluation

The goal of the third prototype was to finalize the DDA system designs in order to arrive at a final prototype implementation which could be used for evaluation. As for the pressure-based DDA system, this entailed a test of the two designs that had been created, as well as a test of the pressure calibration step that had been added to the beginning of the game. As for the performance-based DDA system, this entailed a test of the general behavior of the system, as well as additional data gathering for more precise performance estimation.

Testing the proactive and reactive pressure DDA system designs
Two different designs for the pressure-based DDA system had been proposed and implemented, and a small-scale test was set up in order to compare the behavior

of the two systems to one another.
While checking the systems for bugs before starting the tests, it was noted that the reactive system design was very passive due to the logic behind it, to the point of not interfering with the game at any point in a 10-minute gaming session. In contrast, the proactive design was very active, but also had a tendency to push the player to lowest difficulty setting, depending on how well the game had calibrated its thresholds at the beginning.
In the test that was set up, the participant was asked to play five levels of the game (these being the two calibration levels and the three game levels). They were specifically not told about the influence of pressure on the game difficulty, nor were they told that the first two levels were used for calibration. The pressure DDA system design was switched between proactive and reactive throughout the tests.
After four testers (two for proactive and two for reactive), it was decided to stop the tests, as they had only confirmed what was already observed during the implementation phase; the reactive design barely interfered (in most cases spawning no speed-ups or just a single speed-up throughout the game), while the proactive design was active but not usable in its current state (in both tests pushing the player to the lowest difficulty setting despite this difficulty not suiting the testers).

It was decided to continue with the proactive system design, as this design had shown to be much more active in changing the difficulty. However, as the system in its current state was sensitive to bad threshold calibrations, more consideration had to be put into the system implementation.

Gathering data for the performance-based DDA system
The performance estimation functions presented in Table 3.6 have been calculated from different population sizes. For difficulty settings -2 and 2, the population size is 3, while for difficulty setting 0, the population size is 8. This difference in population size came about when it was found that a unique performance estimation function had to be calculated for each level at three different speeds for better precision, and only difficulty setting 0 had been tested more thoroughly.
10 tests were performed in which the participants were asked to play level 6, 8 and 10 in a counter-balanced order. The difficulty setting remained constant throughout the game. Half of the tests were done at difficulty setting -2, and the other half at difficulty setting 2. As previous participants could collectively be described as gamers (due to all participants reporting that they play games at least monthly), it was decided to screen the participants for this test to ensure that their experience level was in line with previous participants'. Potential participants were only asked to participate if they played games monthly, weekly or daily.

After finishing the tests, new performance estimation functions for difficulty settings -2 and 2 were calculated and implemented in the performance-based DDA system. This meant that the system now had nine mathematical functions that depended on the game level and ball speed, with each function being based on data from 8 playthroughs. These functions can be seen in Table 3.7.

Table 3.7: The final performance estimation functions for the different levels and difficulty settings.

	Difficulty setting -2	Difficulty setting 0	Difficulty setting 2
Level 6	f(x) = 0.31x + 1.07	f(x) = 0.57x + 2.67	f(x) = 0.65x + 1.46
Level 8	f(x) = 0.39x + 1.59	f(x) = 0.75x + 0.95	f(x) = 0.83x + 1.07
Level 10	f(x) = 0.26x + 2.31	f(x) = 0.55x + 1.84	f(x) = 0.56x + 3.87

Since it had been decided to use a proactive system design, new threshold values also had to be calculated. These thresholds were obtained by calculating the average performance of the testers for each level and difficulty setting, and then calculating how far the observed performance on average fell from the estimated performance. This resulted in the thresholds presented in Table 3.8.

Table 3.8: The final thresholds for performing a difficulty change in performance-based DDA.

	Difficulty setting -2	Difficulty setting 0	Difficulty setting 2
Level 6	2.65	1.56	3.22
Level 8	1.60	1.42	1.83
Level 10	1.34	1.83	2.96

Improving the decision-making of the pressure-based proactive system design
After testing the proactive and reactive DDA system designs, it was decided to use the proactive design moving forward. However, some changes to the calibration of the threshold had to be made, as it had not been satisfactory. This was an iterative process in which multiple solutions were designed, implemented and tested.

The first solution was to let the participants play all of CAL-1 at their chosen difficulty followed by all of CAL-2 at an increased difficulty, and then obtaining a threshold by calculating the difference in average pressure of the two levels. In this manner, more pressure data could be collected at the increased difficulty, possibly improving the precision of the threshold. However, no significant results came from this, and the solution was therefore discarded.

The second solution was to continuously update the threshold throughout the game. At the end of a window, a new "sub-threshold" value was calculated:

$$\Delta TH = P - (BL + TH)$$
$$TH_{new} = TH - \Delta TH,$$

where P is the window's mean pressure, BL is the baseline and TH is the threshold.

TH_{new} was then added to a list of values that held the five most recently calculated sub-thresholds. The total threshold was calculated as the average of each of the values in the list every time the list was updated. One version of this solution was implemented in which new sub-threshold values were only calculated in case the current window had started out with a difficulty increase. As this was not helpful in case the threshold had been calculated too low to begin with (making difficulty increases rare), another version was implemented in which new sub-thresholds were calculated at the end of every window. In case the window had started with a difficulty decrease, the calculated ΔTH was flipped by multiplying it with -1. When testing, it was found that the updated threshold easily spun out of control as older values were removed, typically increasing to extreme values. The details of the calculation of the threshold were changed slightly throughout several iterations, but no version showed improvement in the playability - most frequently the opposite.

The third solution was to keep the threshold as a static value, but to set it to a minimum of 1 to avoid the threshold being a negative value, as had been observed in a few cases throughout the technical evaluation. Furthermore, a margin was added around the threshold inside which no difficulty change would occur, so as to make the decision-making more forgiving in case the average pressure was close to the threshold. This concept is illustrated in Figure 3.11.

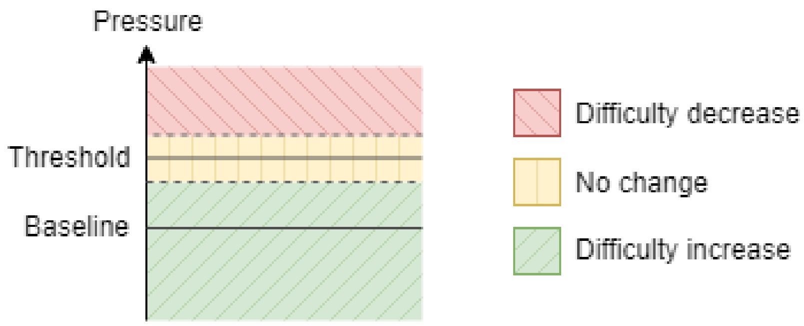

Figure 3.11: A visual representation of how the pressure-based DDA system manipulates the difficulty depending on the average pressure in relation to the baseline and threshold pressure.

This solution did not have the issue of spiraling out of control, and improved the playability by eliminating the situations in which an average pressure at a few decimals from the threshold resulted in a difficulty change. Therefore, this solution was kept to be tested more in-depth during the subsequent pilot test.

3.5 Final prototype

The game is a brick-breaker game based on the classic arcade game Arkanoid. The game has five levels in total (Figure 3.8), two of which are calibration levels that will only be shown once, while the remaining three are looped through indefinitely. Every 30 seconds during the game levels, a speed-up or speed-down will spawn whenever the next brick is broken. The two implemented DDA systems are responsible for determining which drop to spawn.
The ball's speed corresponds to the game difficulty, with a total of seven possible difficulty settings (Table 3.5). At the beginning of a new game, the player can select one of three difficulties to start at (easy, medium or hard).

A custom pressure reader has been built to enable pressure data collection during play sessions. It consists of an Arduino connected with an FSR, which is placed underneath an aluminum keyboard under the foot of the keyboard closest to the arrow keys. A placement board was cut out from plywood to keep the keyboard and FSR in the same position relative to one another.

The pressure-based DDA system gathers pressure data from the pressure reader and info about button presses from the game application, and uses this data to manipulate the game difficulty.
The first two levels in the game are calibration levels (CAL-1 and CAL-2), which have been added to fit the decision-making of the pressure-based DDA specifically to each individual player. In CAL-1, the player plays the full level at their chosen difficulty without any drops. 30 seconds into the level, the player is asked to re-select their difficulty (allowing them to choose a different difficulty if they felt they had made a wrong choice at the beginning of the game). The average pressure for the remainder of the level is calculated as the initial baseline. In CAL-2, the difficulty setting is changed by two steps approximately every 30 seconds in an up-down pattern, and the first 30 seconds of pressure data in each window is saved. At the end of the level, first the baseline is updated by calculating a new baseline from the data from CAL-1 as well as the data gathered during windows of the "original" difficulty in CAL-2. This is done to improve the precision of the baseline. After this, a threshold is obtained by subtracting the average pressure at the increased difficulty from the average pressure at the "original" difficulty. This should yield a positive value (as a higher difficulty should result in higher pressure), but to lessen the impact in case the calibration has calculated the threshold as a very small or negative value, the threshold is capped such that it cannot be less than 1.
During the remainder of the game, every time a new window is started, the next 30 seconds of pressure data are saved to be used for decision-making. When a drop should spawn, the DDA system evaluates the average pressure (calculated from the recently collected pressure data) against the baseline and threshold.

As for the decision-making (Figure 3.11), a threshold margin of 1 is added above and below the threshold, such that if the average pressure is within $\pm$ 1 from the threshold, no change in difficulty will happen for the upcoming window. This margin has been added to avoid a change in difficulty when the player is around their threshold, as this should be the optimal difficulty for them. If the average pressure is instead above the threshold with margins, the difficulty will be decreased, as a high average pressure implies that the game is too difficult. Similarly, if the average pressure is below the threshold with margins, the difficulty will be increased.

The performance-based DDA system quantifies the player's in-game performance as the amount of bricks that are broken within the first 30 seconds of each window, and uses this to manipulate the game difficulty.
Based on data from previous technical evaluations, a set of performance estimation functions (Table 3.7) and thresholds (Table 3.8) have been calculated. These take the current difficulty setting, ball speed and level layout into consideration when estimating how many bricks should have been broken within 30 seconds depending on how many bricks are currently left. Thus, if the observed amount of bricks broken is less than the estimated minus the threshold, the difficulty is decreased, as the player has been unable to meet the requirements of an average performance. Otherwise, the difficulty is increased.

A control version of the game has also been developed to be used in the evaluation of the DDA systems. It will be used to measure the player enjoyment and perceived difficulty when no DDA system is active, which can later be used for comparison.
The game will distribute 16 drops in total throughout 10 minutes of gameplay in levels 6, 8 and 10. Half of these are speed-ups, and the remaining half are speed-downs. This specific amount of drops was chosen to resemble the amount of drops that can be expected to drop in the DDA versions. These versions spawn a maximum of 20 drops (one every 30 seconds), but are likely to spawn less due to excluding the time between levels and the time spent before the ball is launched or re-launched. With 16 drops in total in the control game, it is thus expected that 2 of the total 10 minutes are spent waiting for the next level to load, waiting for the ball to respawn or positioning the paddle before launching the ball. Which type of drop to spawn is randomly decided, unless the difficulty setting is already at the minimum (or maximum), in which case a speed-up (or speed-down) is guaranteed. In this manner, every player will experience the same amount of positive and negative drops, but the order of them will be random. While the game difficulty is not static, the equal amount of positive and negative drops ensures that all players will eventually end up at the same difficulty setting they were in from the beginning.

4 Evaluation

The purpose of the evaluation is twofold. Firstly, it serves to determine whether the implemented DDA systems are successful in comparison to the game without DDA, and secondly, it serves to determine whether the pressure-based DDA is equally or more successful than the performance-based DDA.
The success of the systems is determined based on whether they deliver a better player experience and whether the perceived difficulty of the game is near optimal (thus following the channel of flow by balancing challenge and skill).

Three null hypotheses have been set up to be tested throughout the evaluation:

1. There is no significant difference in perceived difficulty between the control version and DDA versions of the game.
2. There is no significant difference in perceived difficulty between the performance-based and pressure-based DDA versions.
3. There is no significant difference in the Flow and Challenge aspect of player experience between the control, performance-based DDA and pressure-based DDA game versions.

Before starting the evaluation, a pilot experiment with 10 participants was conducted to ensure that the systems were functional and the test procedure was working as intended.

4.1 Conditions

The experiment contains three different conditions, one for each game version that has been developed. For ease of reading, the three conditions will be referred to as CON (control game version), PERF (performance-based DDA version) and PRESS (pressure-based DDA version).

While the two calibration levels in the game were only strictly required for the pressure-based DDA version, they were included in all tests. This ensured consistency in how long each participant got to play the game before the actual game levels began.
As for data collection, the game collects data on their pressure and performance, and data on their subjective player experience and perceived difficulty was subsequently collected with questionnaires. For participants testing CON, pressure data was not collected.

4.2 Participants

30 first-year students currently enrolled in a technical programme at Aalborg University participated in the experiment. Each condition was tested with 10 participants. All participants played games regularly, with 1 playing at least monthly, 10 playing at least weekly and 19 playing at least daily.
The participants included 27 males and 3 females aged 19-30, with the average age being 21.7. Each condition group included one female.

While the experiment originally had 34 participants, four tests were excluded due to technical difficulties with the collected data.

4.3 Materials and setup

The test material included two laptops such that two participants could play the game simultaneously. One of the laptops was connected to a monitor, a webcam and the pressure reader and keyboard, which allowed for the participant to see the game on the monitor while the facilitators could read the pressure graphs on the laptop screen. A camera mounted on a tripod was used to record the full test scene.

For each participant, a set of papers was prepared for them to fill out. This included a consent form to allow for video recordings and data collection, a perceived difficulty rating scale (-2 = too easy, -1 = easy, 0 = optimal, 1 = difficult, 2 = too difficult) and a gaming frequency rating scale (0 = never, 1 = less than once a year, 2 = a couple of times a year, 3 = monthly, 4 = weekly, 5 = daily). To measure the subjective player experience, a copy of the GEQ was also given to each participant. It consists of 33 items to measure the player experience in terms of Competence, Sensory and Imaginative Immersion, Flow, Tension, Challenge, Negative Affect and Positive Affect. For this experiment, the questionnaire was modified to exclude the six items concerned with Sensory and Imaginative Immersion, as these were deemed irrelevant in the case of an arcade-style brick-breaker game (items such as "I was interested in the game's story", "It was aesthetically pleasing" and "I felt that I could explore things").

The setup of the experiment is illustrated in Figure 4.1. Tests were either conducted with two participants simultaneously or with one participant by TEST-MONITOR only. TEST-PC was the laptop that was used to play the control version of the game. TEST-MONITOR was used to play either of the DDA versions of the game, and consisted of a monitor, webcam and pressure reader, which were all connected to DATA-PC. This allowed for the facilitators to see the pressure graphs on DATA-PC, while TEST-MONITOR only displayed the game to the participant. Not pictured is a camera on a tripod, which was placed at varying positions throughout the experiment. The camera was typically placed around 2 meters from the illustrated setup, and angled such that the full test scene could be recorded.

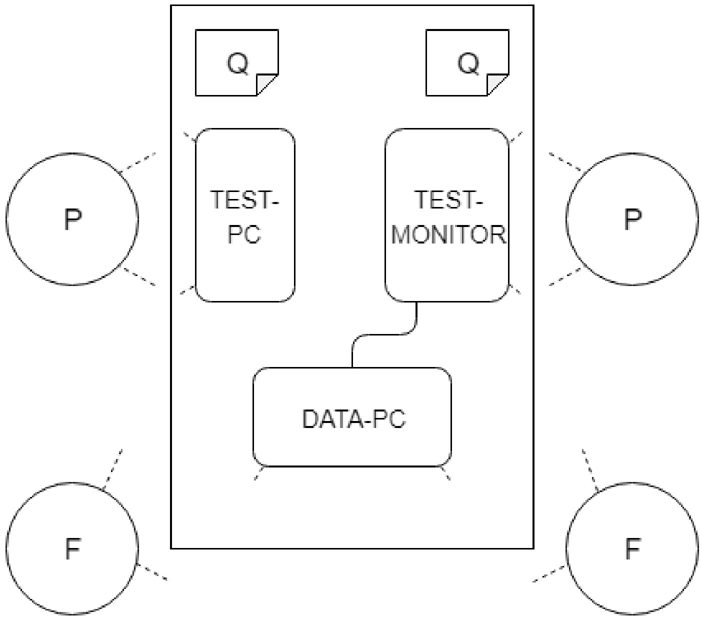

Figure 4.1: The experiment setup. P = participant, F = facilitator, Q = questionnaire and additional papers for the participants.

The setup corresponding to TEST-MONITOR can be seen in Figure 4.2. This setup was created specifically to hide the extra hardware necessary for measuring exerted pressure. The box below the screen thus contains the Arduino, and the wires for the keyboard and FSRs were taped together to hide the fact that additional wires were present.

Figure 4.2: The setup that was used to hide the hardware for the custom pressure reader.

4.4 Experiment design and procedure

The experiment uses a between-subjects design such that each of the three conditions has been tested by groups of 10 participants each.

At the beginning of each test, the main facilitator presented the project as a study on game difficulty and player enjoyment, and the participants were asked to sign a consent form and answer the gaming frequency scale. They were then introduced to the game by explaining the goal of the game, how to control the game and what the drops do. They were told that the very first level (CAL-1) was a test level for them to get comfortable with the game, which was the reason for no drops spawning in that level. They were also notified about the difficulty selection pop-up that is shown at the beginning of the game and 30 seconds into the game, and that they should just select the difficulty they felt fit their capabilities.
The participants played CAL-1 and CAL-2, followed by 10 minutes of playing through the game levels. After 10 minutes had passed for both participants, they were asked to fill out the papers next to them. These included the GEQ, the perceived difficulty rating scale.

At no point prior to or during the test was pressure, in-game performance or DDA mentioned.

4.5 Results

Table 4.1 gives an overview of the average GEQ scores for the different player experience aspects in each of the game versions. These values were obtained by averaging the aspect scores of all participants in each game version. The lowest value possible is 0, and the highest possible is 4. In case of Competence, Flow, Positive Affect and Challenge, a higher score is better. In case of Tension and Negative Affect, a lower score is better.

Table 4.1: The average GEQ scores of the different player experience aspects per game version.

	CON	PERF	PRESS
Competence	2.34	2.36	2.16
Flow	1.76	2.38	1.48
Tension	0.63	0.70	0.97
Challenge	1.02	1.72	1.50
Negative Affect	1.05	0.73	1.53
Positive Affect	2.44	2.68	2.48

Aside from the GEQ, the participants also rated the overall difficulty of the game on a Likert scale from -2 (very easy) through 0 (optimal) to 2 (very hard). For CON, the average Difficulty Rating is -0.60, and for PERF and PRESS it is 0.10.
From these ratings, the DDA systems can be said to have adjusted the difficulty to better fit the participants in comparison to CON, as the DDA systems' Difficulty Ratings are closer to a rating of 0 (optimal difficulty).

Table 4.2 gives an overview of the resulting p-values of performing two-sample t-tests on the GEQ answers and Difficulty Ratings. Each of the tested samples were tested for normal distribution with a Lilliefors test, and in a few cases, the two compared samples could not be assumed to be normally distributed. These cases use a Mann-Whitney Wilcoxon test instead of a two-sample t-test, and are marked with an asterisk in the table. P-values below the significance level of 0.05 have been emboldened in the table.

While Challenge was only found to be significantly different between CON and PERF, the table reveals certain tendencies, as Challenge between CON and PRESS is at 0.07, and Difficulty Rating between CON and either of the DDA versions is 0.06. Furthermore, Flow and Negative Affect were found to be significantly different between PERF and PRESS.
Hedge's g was calculated for each of the relations to determine the effect size of the significant differences that were found. All were found to have an effect size larger than 0.8. A rule of thumb on the interpretation of Hedge's g is that 0.2 can be considered small, 0.5 medium and 0.8 large.

Table 4.2: The p-values of two-sampled t-tests and Mann-Whitney Wilcoxon tests on GEQ scores and Difficulty Ratings.

	CON vs. PERF	**CON vs. PRESS**	**PERF vs. PRESS**
Competence	0.94	0.50	0.38
Flow	0.13	0.45	**0.02**
Tension	1.00*	0.22*	0.12*
Challenge	**0.04**	0.07*	0.76*
Negative Affect	0.32	0.17	**0.01**
Positive Affect	0.51	0.91	0.45
Difficulty Rating	0.06*	0.06*	1.00

**Mann-Whitney Wilcoxon test*

When performing a one-way ANOVA or Kruskal-Wallis (depending on whether the samples were normally distributed), no significant differences were found, as seen in Table 4.3. However, the p-values show certain tendencies approximately similar to the findings from the two-sample t-tests, as Flow, Challenge, Negative Affect and Difficulty all have p-values below 0.1.

Table 4.3: The p-values of one-way ANOVA and Kruskal-Wallis tests on GEQ scores and Difficulty Ratings.

	p-value
Competence	0.691
Flow	0.057
Tension	0.249*
Challenge	0.083*
Negative Affect	0.054
Positive Affect	0.741
Difficulty Rating	0.080*

**Kruskal-Wallis test*

For possible explanations of the results for the different game types, the average time spent in the different difficulties per game type was calculated from the data that was collected with Unity. Figure 4.3 thus shows the percentage of total game time spent in the different difficulties per game version. As it can be seen, CON is normally distributed around difficulty setting 1, PERF is normally distributed around difficulty setting 3, and PRESS looks to be increasing along with the difficulty settings.

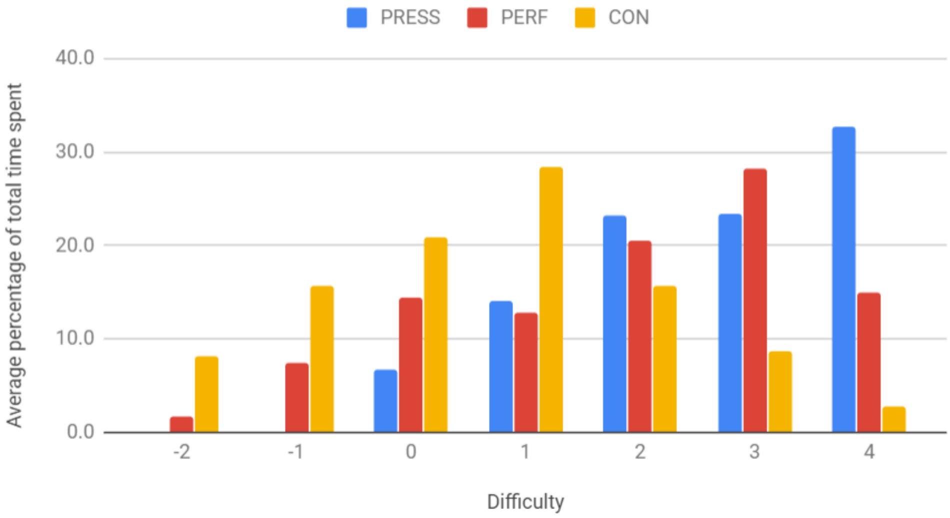

Figure 4.3: Percentages of total game time spent in different difficulties per game version.

It was also decided to calculate the death rate at different difficulties for the different game versions. To do this, the total amount of deaths per difficulty was calculated for each of the game versions. By dividing these values with the total game time per difficulty (the game time also being divided by 10), the resulting values are the amount of deaths per 10 seconds. These death rates can be seen in Figure 4.4.

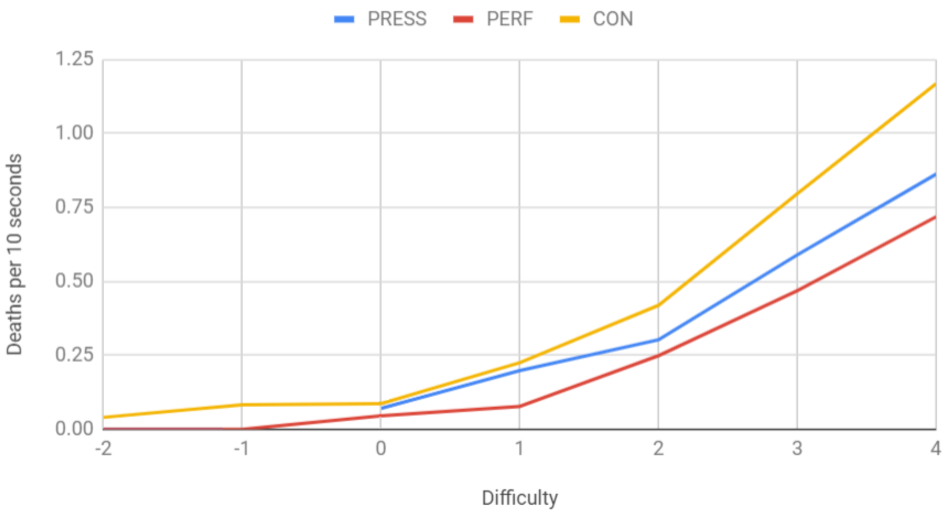

Figure 4.4: Death rates at the different difficulties per game version.

As it can be seen, PRESS in general has a higher death rate than PERF, indicating that the participants were less able to meet the skill requirements of PRESS.

It should be noted that while CON has a higher death rate than both DDA game versions, this may very well be due to the fact that the participants spent much less time in the higher difficulties, and thus were not as accustomed to the speed of the game at the higher difficulties.

As pressure data was collected for both the participants testing PERF and PRESS, it was furthermore decided to analyze how the pressure changed throughout the game. Here it was noted that in several cases, when the difficulty setting changed by one step, the average pressure changed by very little. As an example, Table 4.4 shows the average pressure of a participant in the PERF condition group, ordered by difficulty setting:

Table 4.4: The average pressure of a participant from PERF calculated per difficulty.

Difficulty setting	**-2**	**-1**	**0**	**1**	**2**	**3**	**4**
Average pressure	N/A	230.31	230.61	233.70	246.25	245.70	258.94

Difficulty setting -2 has no average pressure, as the participant never reached this setting. The remaining pressure values have been calculated from the average pressure of each window where the participant was in the given difficulty. When considering the differences in pressure at a 2-step interval, e.g. the difference between settings 0 and 2 (15.64) and between settings 2 and 4 (12.69), the change in pressure seems to nicely follow the assumption that a higher difficulty results in a higher pressure, and the amount of change seems reasonable. However, when looking at the difference between neighboring difficulties, the change in pressure is less obvious. Examples of this are the difference between settings -1 and 0 (0.30) and between settings 2 and 3 (-0.55). This seems to be the general case for most participants.

To get a better overview, the differences in pressure between setting 0 and 2 and difficulty 2 and 4 were calculated for each participant in PERF and PRESS, as seen in Figure 4.5. The differences between settings -2 and 0 were not calculated, as few participants had been in difficulty setting -2. As it can be seen, even when the difficulty setting changed by two steps, some participants had a negative pressure change, meaning that they pressed lighter in the higher difficulty setting of the two. Almost half of the participants included in the figure have also had a pressure change less than 5.

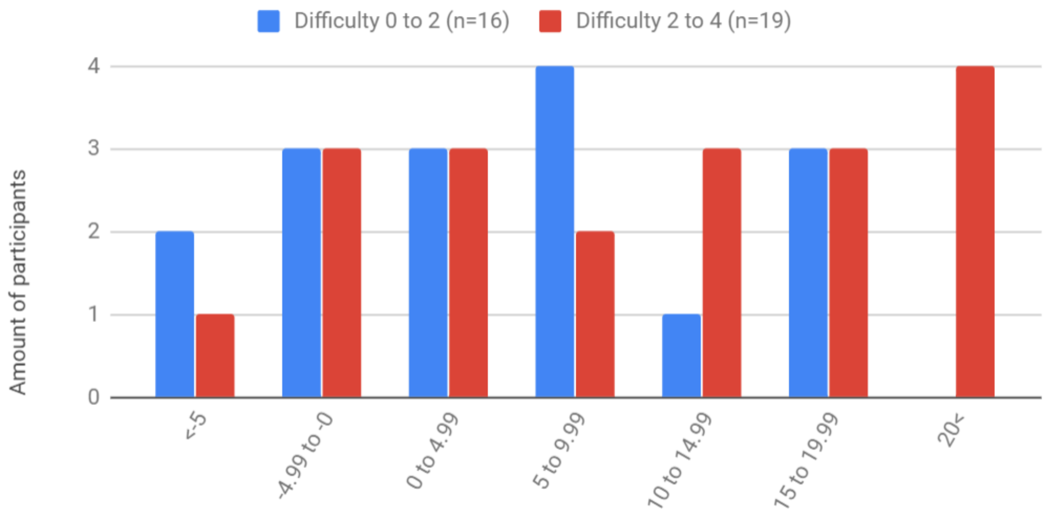

Figure 4.5: Changes in pressure between select difficulties, organized by how many participants' pressure levels fall within the different ranges of pressure change.

Finally, as for observations made during tests and from comments of participants, three important points can be noted about the game implementations and the test design.

Firstly, it was noted that some participants dropped the ball on purpose when only a few blocks remained in a level. As the game did not include any form of "player lives", and the participants had no incentive to care about the negative effect on their game score, this allowed them to re-position the paddle and aim the ball more directly rather than spending the effort to aim the ball while it was in movement.

Secondly, several participants noted that they would have liked to change difficulty at later points in the game. This indicates that they were not satisfied with the difficulty that was currently active.

Thirdly, based on the comments (or lack thereof) of participants, the DDA systems were deemed to be successfully hidden. The facilitator introduced the project as a study on game difficulty and player enjoyment, and the comments of participants reflects that this was supposedly the purpose of the study. Participants who commented on the difficulty of the game seemed to believe that the difficulty depended on the initial difficulty they were allowed to select during CAL-1, and in some cases also on how many levels they had cleared throughout the test. A few participants commented on the game design itself and how they imagined one might improve the player enjoyment. No comments or questions were made on the setup with the external keyboard and screen.

5 Discussion

With the results presented in the previous section, it can be concluded that none of the null hypotheses set up in *4 Evaluation* can be refuted:

1. **Not refuted** There is no significant difference in perceived difficulty between the control version and DDA versions of the game.
2. **Not refuted:** There is no significant difference in perceived difficulty between the performance-based and pressure-based DDA game versions.
3. **Not refuted** There is no significant difference in the Flow and Challenge aspect of player experience between the control, performance-based DDA and pressure-based DDA game versions.

Although none of the null hypotheses could be refuted, the results showed certain tendencies. The implications of these will be discussed further by considering the indications and limitations of the evaluation.

5.1 Indications of results

Multiple of the obtained results point towards the fact that the pressure-based DDA system was too aggressive in regards to increasing the difficulty. The percentage of time spent in the highest difficulty setting is approximately twice as high in the pressure-based DDA version as in the performance-based DDA version. From observations during tests, gameplay footage and the GEQ scores for the two versions, this was in most cases not the optimal difficulty. As an example, from the presented GEQ scores, it can be seen that the pressure-based DDA version has scored worst in both Competence, Tension and Negative Affect. This indicates that some participants may not have felt competent while playing, for example in cases where they frequently dropped the ball, and as a result of this became frustrated and lost interest in the game.
Furthermore, there was found to be a significant difference in the scores for Flow and Negative Affect between the two DDA game versions. Performance-based

DDA performed best of the two in both cases. These specific player experience aspects again relate to the fact that the pressure-based DDA made the game too difficult. From the perspective of the flow theory, the participants can be said to have been above the flow channel, resulting in high levels of frustration which eventually caused a disinterest in the game, as is expressed in the score for Negative Affect.

A possible explanation of the issues with the pressure-based DDA can be found in the pressure changes that were calculated for the different difficulties. It was found that the assumption that "a higher difficulty results in a higher pressure" did not hold true in all cases, and furthermore, that the pressure change between neighboring difficulties was often too small to be a reliable determinant of perceived difficulty.
While related work has repeatedly shown a connection between arousal, pressure and game difficulty, the findings were based on several minutes of pressure data. In this project, due to the requirement of a live analysis of the pressure data during the gaming session, the data was analyzed in windows of 30 seconds each. This may be a reason why the pressure did not adhere to the assumptions that were made based on related work that had several minutes of data per analysis.

In summation, the DDA versions partially succeeded in challenging the participants compared to the control game version, as the results showed clear tendencies in the differences in Challenge and Difficulty Rating, although they were not found to be significantly different. While performance-based DDA performed very well (scoring best in 5 of 6 player experience aspects in the GEQ), the pressure-based DDA caused a too high difficulty, with averagely 32.7% of the game time being spent in the highest difficulty setting. This is suspected to be an issue with using pressure as a metric in the manner it has been done for this project, but more research would be required to be able to say anything conclusive on the matter.

5.2 Limitations

The participants for the evaluation all fall in the category "young adult gamers". It cannot be assured that similar results will be obtained if a different target group was tested upon, especially in the case of DDA, as it has previously been shown to vary in effect depending on the experience level of the player [1, 21]. Furthermore, the fact that only 3 of the 30 participants were female poses a gender balance issue that should be taken into account when considering the findings, as this may have an effect on the results.
The game that has been used for the project was also specifically chosen due to the genre and gameplay fitting with several related studies on pressure in relation to games. Different games may yield different results, especially as the performance-based DDA system for this project cannot be directly translated to other games.

5.3 Impact for practitioners and future work

This project may serve as a stepping stone towards future affective games in which indirect feedback is integrated. As the developed pressure-based DDA system is not specific to the game, the different approaches to the design and development of a pressure-based DDA throughout this project can give some ideas for how to begin the development of a new system. Furthermore, although the pressure-based DDA was found to tend to increase the difficulty too high, there are still some interesting points for practitioners to take note of:

- Pressure-based DDA was able to both increase and decrease the difficulty of the game.
- Individual pressure calibration was successful in 11 out of 12 cases.
- Although the difficulty of the pressure-based DDA game version was generally too high, the difficulty rating was closer to optimal than the control game version.

As for future work within this subject, a suggestion would be to look into the behavior of pressure throughout a gaming session under various conditions to get a better understanding of how pressure as a measurement behaves. Examples of such conditions could be a constantly increasing difficulty, a sudden shift in difficulty, or perhaps different lengths of time between difficulty changes. This may also help clarify how to treat pressure in a future pressure-based DDA system.
Another suggestion would be to implement the pressure-based DDA system in another game in which the control scheme or gameplay causes the player to press keys in a different manner. In the developed brick breaker game, the input can be described as altering short and long key presses on the left and right arrows. A different input pattern could be single-button games, games that require short and quick taps or games that require long key presses. It is possible that different input patterns like these may cause the pressure levels to behave differently, perhaps even adhering better to the assumption of increased difficulty causing increased pressure.
Another possibility would be to investigate whether a different system design for pressure-based DDA might make the calibration steps irrelevant. While this was initially attempted by using regression on the data, it did not result in a successful system design. Cutting out the calibration steps would allow for the player to get started with the game right away, rather than having to calibrate the game at the beginning of every new gaming session.

6 Conclusion

The research problem for this project has been to explore how pressure can be used as indirect feedback in a DDA system. To determine this, a performance-based and a pressure-based DDA system was developed for a brick-breaker game and tested against each other and a control game version with no DDA system.
As a conclusion, the project has presented the design, implementation and testing of a pressure-based DDA system that worked to a moderate degree. This has uncovered some specific issues and promising aspects of pressure as indirect feedback.

Both DDA systems had a difficulty rating closer to optimal than the control game version. The performance-based DDA was also found to have significantly different Challenge rating in the GEQ compared to the control game version. This indicates that the DDA systems did indeed adjust the difficulty to better fit the skill level of the player.

However, there are still important issues to tackle with using pressure as feedback in an affective game. Although previous work on the subject of pressure and games has repeatedly shown a connection between difficulty and pressure, an affective game requires relatively small amounts of pressure data to be analyzed continuously to adapt the game on the go, and pressure in such a case cannot be expected to behave as clearly as it has in previous work.

Finally, aside from this issue, pressure-based DDA seems very promising in the regard of applicability across different games. While the performance-based DDA had to be designed and developed specifically for the brick breaker game, the pressure-based DDA system uses no game-specific parameters to determine whether the difficulty should change. This indicates that the pressure-based DDA system can easily be integrated with a different game, while for the performance-based DDA system, a completely new design must be developed, as game performance has to be estimated differently depending on the goals and parameters of the individual game.

Bibliography

[1] J. T. Alexander, J. Sear, and A. Oikonomou. "An investigation of the effects of game difficulty on player enjoyment". In: *Entertainment Computing* 4.1 (2013), pp. 53–62. DOI: 10.1016/j.entcom.2012.09.001.

[2] M. Aponte, G. Levieux, and S. Natkin. "Measuring the level of difficulty in single player video games". In: *Entertainment Computing* 2.4 (2011), pp. 205–213. DOI: 10.1016/j.entcom.2011.04.001.

[3] C. Bailey and M. Katchabaw. "An experimental testbed to enable auto-dynamic difficulty in modern video games". In: *Proceedings of the 2005 North American Game-On Conference*. Montreal, Canada, Aug. 2005, pp. 18–22. DOI: 10.1016/j.entcom.2011.04.001.

[4] D. Bersak et al. "Intelligent biofeedback using an immersive competitive environment". In: *Online Proceedings for the Designing Ubiquitous Computing Games Workshop, Ubicomp 2001*. Feb. 2001. URL: https://www.researchgate.net/publication/246492595_Intelligent_biofeedback_using_an_immersive_competitive_environment.

[5] B. Bontchev. "Adaptation in Affective Video Games: A Literature Review". In: *Cybernetics and Information Technologies* 16.3 (2016), pp. 3–34. DOI: 10.1515/cait-2016-0032.

[6] M. M. Bradley and P. J. Lang. "Measuring emotion: The self-assessment manikin and the semantic differential". In: *Journal of Behavior Therapy and Experimental Psychiatry* 25.1 (1994), pp. 49–59. DOI: 10.1016/0005-7916(94)90063-9.

[7] G. Chanel et al. "Boredom, Engagement and Anxiety As Indicators for Adaptation to Difficulty in Games". In: *Proceedings of the 12th International Conference on Entertainment and Media in the Ubiquitous Era*. MindTrek '08. Tampere, Finland: ACM, 2008, pp. 13–17. DOI: 10.1145/1457199.1457203.

[8] T. Christy and L. I. Kuncheva. "Technological Advancements in Affective Gaming: A Historical Survey". In: *GSTF Journal on Computing (JoC)* 3.4 (2014), pp. 32–41. DOI: 10.7603/s40601-013-0038-5.

[9] Computer History Museum. *Timeline of Computer History*. Retrieved 3rd May 2019. URL: https://www.computerhistory.org/timeline/graphics-games/.

[10] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. New York: Harper and Row, 1990. ISBN: 0061339202.

[11] S. B. Daily et al. "Affective Computing: Historical Foundations, Current Applications, and Future Trends". In: *Emotions and Affect in Human Factors and Human-Computer Interaction*. Elsevier, 2017, pp. 213–231. DOI: 10.1016/b978-0-12-801851-4.00009-4.

[12] A. Dekker and E. Champion. "Please Biofeed the Zombies: Enhancing the Gameplay and Display of a Horror Game Using Biofeedback". In: *DiGRA &#3907 - Proceedings of the 2007 DiGRA International Conference: Situated Play*. The University of Tokyo, 2007. ISBN: ISSN 2342-9666. URL: http://www.digra.org/digital-library/publications/please-biofeed-the-zombies-enhancing-the-gameplay-and-display-of-a-horror-game-using-biofeedback/.

[13] H. G. E. Dirrik and G. Lampropoulos. "BioPong: Adaptive Gaming Using Biofeedback". In: *Creating the Difference, Proceedings of the Chi Sparks 2014 Conference*. 2014, pp. 100–103. URL: http://mediatechnology.leiden.edu/research/publications/biopong-adaptive-gaming-using-biofeedback.

[14] D. Dziedzic. "Dynamic Difficulty Adjustment Systems for Various Game Genres". In: *Homo Ludens* 1 (Dec. 2016), pp. 35–51. URL: https://www.researchgate.net/publication/312028327_Dynamic_difficulty_adjustment_systems_for_various_game_genres.

[15] P. Ekman and W. V. Friesen. "Measuring facial movement". In: *Environmental Psychology and Nonverbal Behavior* 1.1 (1976), pp. 56–75. DOI: 10.1007/bf01115465.

[16] *Force Sensing Resistor Integration Guide and Evaluation Parts Catalog with Suggested Electrical Interfaces*. Version 1.0. Interlink Electronics. Camarillo, CA. URL: https://www.sparkfun.com/datasheets/Sensors/Pressure/fsrguide.pdf.

[17] R. L. Hazlett. "Measuring Emotional Valence During Interactive Experiences: Boys at Video Game Play". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '06. Montreal, Quebec, Canada: ACM, 2006, pp. 1023–1026. ISBN: 1-59593-372-7. DOI: 10.1145/1124772.1124925.

[18] W. van den Hoogen, E. Braad, and W. Ijsselsteijn. "Pressure at Play: Measuring Player Approach and Avoidance Behaviour through the Keyboard". In: *DiGRA &#3914 - Proceedings of the 2014 DiGRA International Conference*. Aug. 2014. URL: http://www.digra.org/wp-content/uploads/digital-library/digra2014_submission_111.pdf.

[19] W. van den Hoogen, W. IJsselsteijn, and Y. de Kort. "Effects of Sensory Immersion on Behavioural Indicators of Player Experience: Movement Synchrony and Controller Pressure". In: *DiGRA &#3909 - Proceedings of the 2009 DiGRA International Conference: Breaking New Ground: Innovation in Games, Play, Practice and Theory*. Brunel University, Sept. 2009. URL: http://www.digra.org/wp-content/uploads/digital-library/09287.18127.pdf.

[20] W. van den Hoogen, W. Ijsselsteijn, and Y. de Kort. "Exploring behavioral expressions of player experience in digital games". In: *Proceedings of the Workshop on Facial and Bodily Expression for Control and Adaptation of Games ECAG 2008* (Jan. 2008). URL: https://www.researchgate.net/publication/242468669_Exploring_behavioral_expressions_of_player_experience_in_digital_games.

[21] R. Hunicke. "The case for dynamic difficulty adjustment in games". In: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology - ACE '05*. ACM Press, 2005, pp. 429–433. DOI: 10.1145/1178477.1178573.

[22] W. IJsselsteijn, Y. de Kort, and K. Poels. *The Game Experience Questionnaire*. Jan. 2013. URL: https://pure.tue.nl/ws/files/21666907/Game_Experience_Questionnaire_English.pdf.

[23] W. IJsselsteijn et al. "Measuring the experience of digital game enjoyment". In: *Proceedings of Measuring Behavior* (Jan. 2008), pp. 88–89. URL: https://www.researchgate.net/publication/254415787_Measuring_the_experience_of_digital_game_enjoyment.

[24] W. Jänig. "Autonomic Nervous System". In: *Human Physiology*. Springer Berlin Heidelberg, 1989, pp. 333–370. DOI: 10.1007/978-3-642-73831-9_16.

[25] M. G. Kiel, A. Dix, and J. Allanson. "Affective Videogames and Modes of Affective Gaming: Assist Me, Challenge Me, Emote Me". In: *DiGRA &#3905 - Proceedings of the 2005 DiGRA International Conference: Changing Views: Worlds in Play*. 2005. ISBN: ISSN 2342-9666. URL: http://www.digra.org/wp-content/uploads/digital-library/06278.55257.pdf.

[26] J. M. Kivikangas et al. "A review of the use of psychophysiological methods in game research". In: *Journal of Gaming & Virtual Worlds* 3 (Sept. 2011), pp. 181–199. DOI: 10.1386/jgvw.3.3.181_1.

[27] M. Klarkowski et al. "Psychophysiology of Challenge in Play". In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*. ACM Press, 2016, pp. 1930–1936. DOI: 10.1145/2851581.2892485.

[28] I. Kotsia, I. Patras, and S. Fotopoulos. "Affective gaming: Beyond using sensors". In: *2012 5th International Symposium on Communications, Control and Signal Processing*. IEEE, 2012. DOI: 10.1109/isccsp.2012.6217768.

[29] I. Kotsia, S. Zafeiriou, and S. Fotopoulos. "Affective Gaming: A Comprehensive Survey". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2013, pp. 663–670. DOI: 10.1109/cvprw.2013.100.

[30] R. Lara-Cabrera and D. Camacho. "A taxonomy and state of the art revision on affective games". In: *Future Generation Computer Systems* 92 (2019), pp. 516–525. DOI: 10.1016/j.future.2017.12.056.

[31] C. Liu et al. "Dynamic Difficulty Adjustment in Computer Games Through Real-Time Anxiety-Based Affective Feedback". In: *International Journal of Human-Computer Interaction* 25.6 (2009), pp. 506–529. DOI: 10.1080/10447310902963944.

[32] R. L. Mandryk, M. S. Atkins, and K. M. Inkpen. "A continuous and objective evaluation of emotional experience with interactive play environments". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*. ACM Press, 2006, pp. 1027–1036. DOI: 10.1145/1124772.1124926.

[33] K. Mark Gilleade and J. Allanson. "A Toolkit for Exploring Affective Interface Adaptation in Videogames". In: *Proceedings of HCI International 2003*. Jan. 2003, pp. 370–374. URL: https://www.researchgate.net/publication/237636586_A_Toolkit_for_Exploring_Affective_Interface_Adaptation_in_Videogames.

[34] I. B. Mauss and M. D. Robinson. "Measures of emotion: A review". In: *Cognition & Emotion* 23.2 (2009), pp. 209–237. DOI: 10.1080/02699930802204677.

[35] M. K. Miller and R. L. Mandryk. "Differentiating in-Game Frustration from at-Game Frustration using Touch Pressure". In: *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces - ISS '16*. ACM Press, 2016, pp. 225–234. DOI: 10.1145/2992154.2992185.

[36] P. M. Mohan, V. Nagarajan, and S. R. Das. "Stress measurement from wearable photoplethysmographic sensor using heart rate variability data". In: *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2016, pp. 1141–1144. DOI: 10.1109/iccsp.2016.7754331.

[37] L. Nacke and C. A. Lindley. "Flow and immersion in first-person shooters". In: *Proceedings of the 2008 Conference on Future Play Research, Play, Share - Future Play '08*. ACM Press, 2008, pp. 5–6. DOI: 10.1145/1496984.1496998.

[38] L. E. Nacke et al. "Biofeedback Game Design: Using Direct and Indirect Physiological Control to Enhance Game Interaction". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 103–112. DOI: 10.1145/1978942.1978958.

[39] H. O'Brien and K. McLean. "Measuring the User Engagement Process". In: *Engagement by Design Pre-Conference Workshop, CHI* (Apr. 2009). URL: https://www.researchgate.net/publication/279692732_Measuring_the_User_Engagement_Process.

[40] R. W. Picard. *Affective Computing*. The MIT Press, 1997. ISBN: 9780262161701.

[41] P. Rani, N. Sarkar, and C. Liu. "Maintaining optimal challenge in computer games through real-time physiological feedback". In: *Proceedings of the 11th International Conference on Human Computer Interaction* 58 (July 2005), pp. 22–27. URL: https://www.researchgate.net/publication/228615462_Maintaining_optimal_challenge_in_computer_games_through_real-time_physiological_feedback.

[42] J. Russell. "A Circumplex Model of Affect". In: *Journal of Personality and Social Psychology* 39 (Dec. 1980), pp. 1161–1178. DOI: 10.1037/h0077714.

[43] P. S. Sreeja and G. S. Mahalakshmi. "Emotion Models: A Review". In: *International Journal of Control Theory and Applications* 10 (Jan. 2017), pp. 651–657. URL: https://www.researchgate.net/publication/319173333_Emotion_Models_A_Review.

[44] R. Sutoyo et al. "Dynamic Difficulty Adjustment in Tower Defence". In: *Procedia Computer Science* 59 (2015), pp. 435–444. DOI: 10.1016/j.procs.2015.07.563.

[45] J. Sykes and S. Brown. "Affective Gaming: Measuring Emotion Through the Gamepad". In: *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 732–733. DOI: 10.1145/765891.765957.

[46] C. T. Tan, S. Bakkes, and Y. Pisan. "Inferring Player Experiences Using Facial Expressions Analysis". In: *Proceedings of the 2014 Conference on Interactive Entertainment*. IE2014. Newcastle, NSW, Australia: ACM, 2014, 7:1–7:8. DOI: 10.1145/2677758.2677765.

[47] T. Tijs, D. Brokken, and W. IJsselsteijn. "Creating an Emotionally Adaptive Game". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 122–133. DOI: 10.1007/978-3-540-89222-9_14.

[48] T. J. W. Tijs, D. Brokken, and W. A. IJsselsteijn. "Dynamic Game Balancing by Recognizing Affect". In: *Fun and Games*. Springer Berlin Heidelberg, 2008, pp. 88–93. DOI: 10.1007/978-3-540-88322-7_9.

[49] E. E. Ustrup et al. "Comparing the applicability of pressure as a game metric to self-assessment and physiological metrics". In: *The 17th IFIP TC.13 International Conference on Human-Computer Interaction*. Springer. 2019.

[50] T. Wijman. *Mobile Revenues Account for More Than 50% of the Global Games Market as It Reaches $137.9 Billion in 2018*. Retrieved 3rd May 2019. Apr. 2018. URL: https://newzoo.com/insights/articles/global-games-market-reaches-137-9-billion-in-2018-mobile-games-take-half/.

[51] Wikipedia, The Free Encyclopedia. *Arkanoid*. Retrieved 4th March 2019. URL: https://en.wikipedia.org/wiki/Arkanoid.

[52] N. Xiang, L. Yang, and M. Zhang. "Dynamic Difficulty Adjustment by Facial Expression". In: *Lecture Notes in Electrical Engineering* 208 (Jan. 2013), pp. 761–768. DOI: 10.1007/978-1-4471-4796-1_97.

[53] Y. Yacoob and L. S. Davis. "Recognizing human facial expressions from long image sequences using optical flow". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.6 (1996), pp. 636–642. DOI: 10.1109/34.506414.

[54] C. Yun et al. "O' Game, Can You Feel My Frustration?: Improving User's Gaming Experience via Stresscam". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: ACM, 2009, pp. 2195–2204. DOI: 10.1145/1518701.1519036.

[55] M. A. Zafar et al. "Gaming away stress: Using biofeedback games to learn paced breathing". In: *IEEE Transactions on Affective Computing* (2018). DOI: 10.1109/TAFFC.2018.2816945.

[56] M. Zohaib. "Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review". In: *Advances in Human-Computer Interaction* 2018 (2018), pp. 1–12. DOI: 10.1155/2018/5681652.