A study on the use of eye tracking to adapt gameplay and procedural content generation in first-person shooter games

João Antunes and Pedro Santana

Abstract—This paper studies the use of eye tracking in a First-Person Shooter (FPS) game as a mechanism to (1) control the attention of the player’s avatar according to the attention deployed by the player and (2) guide the gameplay and game’s procedural content generation, accordingly. This results in a more natural use of eye tracking in comparison to a use in which the eye tracker directly substitutes control input devices, such as gamepads. The study was conducted on a custom endless runner FPS, Zombie Runner, using an affordable eye tracker. Evaluation sessions showed that the proposed use of eye tracking provides a more challenging and immersive experience to the player, when compared to its absence. However, a strong correlation between eye tracker calibration problems and player’s overall experience was found. This means that eye tracking technology still needs to evolve but also means that once technology gets mature enough players are expected to benefit greatly from the inclusion of eye tracking in their gaming experience.

I. INTRODUCTION

With eye tracking dating back from the XVIII century [2], many have been the applications for this technology, such as medicine [3], robotics [4], [5], advertising [6]) and, more recently, computer games [7]. In the past decade, research on computer games tried to compare traditional input (e.g., mouse, keyboard, gamepad) with eye tracking input, in terms of action accuracy and responsiveness [8], [9], [7], [10], [11]. These comparisons were often made after asking players to compete against each other or asking users to complete a given task, using the different input methods. Overall, these comparisons provided mixed results, with some studies claiming that the use of eye tracking contributed to better task completion [11], while others claiming that traditional input devices provided better overall results [3]. Hence, these previous studies have provided contradictory results regarding the effectiveness of the use of eye tracking in computer games as a direct control input, which may be an indication that eye tracking is not best suited to direct input control.

Bearing the limitations of eye tracking as a simple direct control input in mind, in this paper, we propose to use it in order to control the attention of the player’s avatar and the game’s procedural content generation. This use of eye tracking is focused in mapping the mental state of the player and her/his avatar, which we believe to be much more natural and useful than directly controlling a pointer with the eyes, which has no actual mapping to real life. That is, instead of replacing the traditional input for an eye tracker, we are more interested in studying meaningful ways about how eye tracking can improve gameplay, which includes co-existence with traditional inputs. This also means that, instead of analysing objective performance-based data, as in previous studies, we are more interested in a subjective analysis of how eye tracking improves enjoyability and how well the player adapts to the technology.

To demonstrate the value of the herein proposed alternative uses of eye tracking in computer games, we developed and tested our own endless runner First-Person Shooter (FPS), Zombie Runner (see Fig. 1). In this game, shot accuracy, automatic obstacle avoidance, and procedural obstacle spawn probability are controlled as a function of the avatar’s attention model, which, in turn, operates according to the player’s gaze estimated with an affordable eye tracker. The goal is to better represent the player’s actions in the game, thus contributing to a more immersive experience. Based on a set of testing sessions, we conclude that the use of eye tracking provides a more challenging and immersive experience to the player. Participants reported better levels of satisfaction while playing the game with the gaze tracking turned on. However, a strong correlation between eye tracker calibration problems and player’s overall experience was found. This means that eye tracking technology still needs to evolve but also means that once technology gets mature enough players are expected to benefit greatly from the inclusion of eye tracking in their gaming experience.

This paper is organised as follows. Section 2 presents an
Eye tracking has been explored in computer games as an alternative to traditional input methods, such as mouse or keyboard [7], [10]. By testing gaze input versus mouse input in three different computer games, Smith and Graham [7] concluded that the use of eye tracking can provide a more immersive experience to the player. Isokoski and Martin [9] conducted a preliminary study on the use of eye tracker in First Person Shooters. Each participant in the test was asked to play the same game using three different input method schemes: (1) mouse, keyboard, and eye tracker; (2) only mouse and keyboard; or (3) a console gamepad. The conclusions were not exactly encouraging, suggesting that the performance with the eye tracker was quite inferior to the other two. Still, Isokoski and Martin attributed these results to the players’ greater knowledge and contact with the traditional input methods, suggesting that this scenario could change with more training.

Other studies achieved similar conclusions. Leyba and Malcolm [8] created a simple test in which the player was asked to eliminate twenty-five balls that moved around the screen at different velocities. The player would move the pointer using the mouse or the eye tracker and would eliminate the balls by clicking on them with the mouse. Two conditions were tested: with and without time limit to complete the task. The results showed that, without time limit, precision and time to complete the task was worse while using the eye tracker than when using a mouse. The same results were obtained for the no time limit condition, in which performance was based on percentage of balls eliminated by the player. Michael Dorr et al. [11] achieved totally opposite results. After creating and adapting a clone of the classic game Breakout, twenty players were asked to participate in a tournament. Players were separated in pairs. The two players of each pair played against each other, one using the mouse and the other using an eye tracker. The control inputs were swapped between rounds. The results showed that the players who used the eye tracker achieved higher scores and won more rounds. The players also stated that using the eye tracker was highly enjoyable.

These discrepant results between studies suggest that the type of game and development method of the same game are key elements to achieve a satisfying final result.

Bearing the limitations of using eye tracking as a simple direct control input in mind, we propose to use it in order to control the attention of the player’s avatar and the game’s procedural content generation. This use of eye tracking is focused in mapping the mental state of the player and her/his avatar, which we believe to be much more natural and useful than controlling a pointer with the eyes, which has no mapping to real life. Another alternative and interesting use of eye tracking, not tackled in this paper, is to know when to actively redirect the player’s attention [19].

Procedural Content Generation (PCG) concerns all creation of game content (e.g., sounds, levels, objects, characters, textures) with algorithms, with limited or indirect stimuli from the user [20]. PCG is a way of coping with the daunting task of manually creating and populating massive open worlds. Besides this more traditional use of PCG, some researchers proposed that, through the analysis of the interaction between the player and the game, PCG could be used to create a playing experience that adapts itself to the player [21], [22], [23], improving the game’s replay value. This new approach is known as Experience-Driven Procedural Content Generation (EDPCG) [24].

Within the EDPCG framework, it is possible to generate levels that are adapted to the strengths and limitations of the player, in an attempt to maximize the fun factor. Many studies propose that the fun and challenge factors are directly linked [25], [26], [27], [28], [29], [30], [31], which means that to tune the fun factor one often has to tune the challenge factor. In this line, in this paper, we propose the integration of PGC with gaze tracking so as to adapt the challenge the player as to face according to his/her form of playing and, as a result, to improve the fun factor.

III. Zombie Runner

The game herein presented, Zombie Runner, was purposely designed to integrate gaze tracking at its core mechanic as a way of mapping both player’s and avatar’s visual attention, instead of as a way of directly controlling the avatar’s motor actions. The latter is done via a traditional input, a gamepad. This way gaze and hand orthogonally and, thus, more naturally, control their virtual counterparts. To better analyse the advantages and disadvantages of the use of eye tracking in games, focus should be given to game genres where the player’s visual attention has to be shared between different elements in the scene under tight temporal restrictions, where the eye tracking use is more central and challenging. For this reason, Zombie Runner, has been designed and implemented as the common genre (of high impact) FPS. The character’s running action is fully automated so as to reduce the number of actions the player had to memorize and control and, hence, reduce variability in the game evaluation phase.

A. System Configuration

The game has been with the following hardware configuration in mind (see Fig. 2): a Gazepoint GP3 eye tracker; a
Figure 2. The hardware configuration and a test subject during an evaluation session.

Microsoft Xbox gamepad; and a 32 inch computer screen. The player sits in front of the computer screen, with the gamepad in hand. The eye tracker sits below the computer screen. Behind it, a laptop running the game is available for the research team during the evaluation sessions. Eye tracking was done using the Gazepoint Control software \cite{32} and the game was developed on Unreal Engine.

B. Game Rules and Mechanics

Zombie Runner follows the following set of rules. The main objective of the game is to ensure that the avatar survives for as long as possible while running along a corridor with a non-controllable constant forward movement. The player can achieve this by killing enemies and by noticing elements in the scene (enemies and obstacles). The player can kill enemies by aiming and shooting the avatar’s gun via the gamepad. The player can notice the several elements in the scene by actively looking at them on the screen for a sufficient amount of time. The gaze of the player is estimated with the eye tracker. An element noticed by the player is also noticed by the avatar. An element noticed by the player is also noticed by the avatar.

If the avatar approaches a previously noticed obstacle, it will automatically avoid that obstacle (i.e., jump over a rock or dodge an hanging tree branch). If this obstacle was not noticed early enough, it will not be seen by the avatar, which will result in a collision and subsequent avatar’s health decrement. A shot enemy that was noticed dies instantly, whereas it will only be hurt if unnoticed. This intends to simulate the lack of shot accuracy resulting from an insufficient focus on the enemy. If a noticed enemy approaches the avatar, it will attack, causing the avatar to lose health. If the enemy was not noticed early enough, its attack will cause instant avatar’s death. The idea is that the avatar is able to dodge the attack provided the attacker had been noticed. The avatar dies when its health reaches zero, leading to game over. Fig. 3 depicts the state diagrams associated to these game rules.

C. Game Implementation

Zombie Runner was developed on Unreal Engine. The programming of Zombie Runner was split between C++ and Blueprints, the Unreal’s visual scripting language. C++ was used for the core algorithms while Blueprints was used to program actor behaviour, such as enemies and obstacles. The game is based on the Unreal’s FPS template, which already implements the expected behaviour for a generic game of the genre, including shooting, walking, and aiming mechanisms.

The corridor in which the game takes place is made of an unidimensional array of 3D tiles (see Fig. \ref{fig:tile_generation}), which are procedurally generated and removed from the scene once the avatar passes by the them. Tiles are comprised of one plane for the floor, two planes for the side walls, an array of marker points distributed randomly on the sides of the floor (not in the region in which the avatar will be crossing), and an array of tree models. Every time a tile is spawned, the array of marker points is traversed. For each marker, there is a 66\% probability of spawning a tree (non-obstacle) in its position, plus a small random offset. This tree is spawned with a random rotation. The tree’s height is also randomly set between reasonable values. The result are sufficiently different tiles that when placed in succession give the feeling of a dense and varied forest on each side of the player’s sight. Fig. \ref{fig:tile_generation} shows three different results of this algorithm applied to the tile generation.

A set of 15 tiles are initially generated. After the 8\textsuperscript{th} tile is spawned, obstacles or enemies can start being spawned with them. This value was achieved by trial and error and produces an initial tile generation that gives enough time for the player to settle in and prepare for the incoming obstacles and enemies. After this initial generation, all subsequent tile generation is recursively and procedurally handled by the tiles themselves. An obstacle (i.e., a rock or an hanging tree branch) or an enemy has a 33\% probability of being procedurally spawn in each new tile. If it does occur, there’s a 55\% probability for either spawning an obstacle or an enemy. A spawned enemy has a 20\% probability of being spawned as
a runner, which has double the speed of a walker, the default behaviour of an enemy. These percentages have been tuned according to a set of informal tests so as to ensure that the game was playable without training.

An obstacle or an enemy can be spawned in three different regions in the tile (see Fig. 5): central region, left region, or right region. If the previous spawned obstacle or enemy was spawned in either the left or the right regions, the new one will be spawned in the central region. This adds variety to the game and avoids objects of the same type to be spawned close to each other. If the previous spawned obstacle or enemy was spawned in the central region, the new one will be spawned in either left or right regions, depending on the player’s gaze. Concretely, the obstacle or enemy is spawned in the region of the tile least gazed by the player during the time spent running over the two previous tiles. This forces the attention of the player to be often alternating between regions, thus increasing the challenge. When the obstacle is spawned in the central, its asset is a rock, being a tree branch when spawned on one of the side regions. Tree branches are implemented as laying small trees (see Fig. 5).

Zombie Runners relies on ray casting to determine which objects are being attended by the player, given the gaze position in the screen. A given object is labelled as noticed if the gaze of the player intersects that same object over an accumulated period of 0.5 seconds. This period was obtained from a set of informal tests as the one that produced the least false positives and negatives when classifying obstacles as noticed/unnoticed. To speed up computation, ray-object intersections are carried out with accessory bounding boxes, rather than with the objects themselves (see Fig. 6). These bounding boxes are also used to determine object-object intersections, such as to determine when the avatar collides against an obstacle (see Fig. 7). The sizes of these bounding boxes had to be adjusted so as to accommodate the inaccuracy of the eye tracker. These adjustments were done during a set of informal tests.

All assets (e.g., obstacles) were freely obtained from the Unreal’s dedicated store. The rigged 3D model and basic animations for the enemies were obtained from Adobe Mixamo. An Unreal Animation Blueprint was built as a state machine which defining the different animation states and transitions between them (see Fig. 8). To provide the player with compelling visual feedback, the appearance of the objects changes when their are first gazed by the player and also later on when they become actually labelled as noticed. The first change consists in rendering the object in wireframe with shades of purple. The second change, more dramatic, consists in turning the object in light blue and in adding a particle effect representing a blue shock wave. Fig. 9 shows the evolution of the materials and effects used on a rock obstacle through the process of being noticed.

IV. EVALUATION AND DISCUSSION

As mentioned, several informal tests were run in order to guide the development and tune the game’s parameters. Then, the whole game was played by a set of ten people in formal game evaluation sessions in order to systematically assess what eye tracking technology brings to the game in terms of overall enjoyability.
A. Evaluation Method

Test sessions were carried out privately in a room without the presence of anyone but the participant and the research team. An email-based call for participants that considered themselves gamers was issued to an universe of 170 people. The first ten that responded to the email and complied with the requirements were selected for the test sessions. The participants had no previous knowledge about the game and its experience. The ages of the ten participants spanned from 25 to 49 years old (see Table I), with different occupations such as software developer, quality assurance tester, and student. All participants were male, with no female subjects volunteering for the experience.

In addition to bottom-line data stored during the test sessions, we also enquired the participants with Game Experience Questionnaires (GEQ) [33], which has been widely applied in previous studies related to control inputs in computer games [34], [35]. By using GEQ, we intend to evaluate if the use of eye tracker is enjoyable and comfortable for the player. We also intend to pinpoint possible advantages and disadvantages of the technology in the way it impacts the overall experience of the player and its relationship with the game, comparing the player experience with and without eye tracking.

1) Test Sessions: Each test session started with a brief questionnaire the participant had to fill in, regarding personal details, such as age and occupation, if the player had some degree of visual impairment, as well as classifying their experience as video game players, with the use of eye trackers, and with the use of gamepads in FPS games. As Tables II, III, and IV show, the average participant had a considerable amount of gaming experience, little to no previous exposition to eye tracking technology, and was moderately experienced using gamepads in FPS games.

To reduce any pleasing biases the participants might have, we conveyed the message that we are agnostic regarding the use of eye tracking in games by means of a brief paragraph on the top of the questionnaire with "the advent of eye tracking technology some developers are including this technology in games" and "this study aims at providing scientific validation of the advantages and disadvantages of such inclusion in..."
terms of gameplay and comfort to the user”. Then, a brief explanation of the game’s rules and objectives was given, followed by a briefing on the eye tracker calibration process. This process was ran as many times as deemed necessary until the calibration was acceptable. When this acceptable calibration was achieved, the participants were asked to state their feelings towards the process and if they would imagine themselves doing it at home before playing a game.

To get the participant acquainted with the game, two separate runs were done, with each one only having an input form enabled (gamepad versus eye tracker). These runs had no time limit and did not count for the evaluation. On the first run, only the eye tracker was enabled and no enemies were spawned. The result was a play session with only obstacles being spawned. This allowed the participants to freely use their eyes to notice the obstacles in front of them and get used to this mechanic, without having to worry about the gamepad. The participants were also asked to state any false positives or obstacles that they had noticed but were not tagged as such by the game. On the second run, no obstacles or enemies were spawned. This allowed the participants to get acquainted with aiming and shooting with the gamepad, adapt to its sensibility and button scheme. The main objective was for the participant to feel comfortable with the different inputs and the test session would only advance when the participants confirmed they felt acquainted with the controls.

The participant was then asked to play the game in its original form, as described in Section 3, for three sessions of 2 minutes each. For each session, the ratio between enemies killed and spawned, the ratio between overall (enemies and obstacles) noticed count and overall spawned count, as well as the number of deaths experienced by the player, were registered. After these three sessions, the player was asked to fill the core and post-game modules of a Game Experience Questionnaire (GEQ) [36]. This questionnaire is filled by answering a set of questions whose answers are provided on a Likert-type scale scored from 0 to 4. Afterwards, the participant was asked to play another set of three sessions of 2 minutes, but this time with eye tracking disabled, which means that all obstacles and enemies were automatically labelled as noticed, with the participant only being required to shoot the enemies. With these three sessions over, another GEQ was filled by the player. Then, the calibration process was performed again and the player was asked to play one 2 minute session of the game but this time with the visual effects, that occur when the obstacle or enemy is noticed, disabled.

Finally, an end questionnaire was then handed to the participant. This questionnaire was more subjective and consisted on three questions: (1) what do you think of the visual effects used on the first sessions, comparing with the last session you played? (2) How was the overall experience of playing the game? (3) Would you consider an eye tracker as part of your gaming setup and why?

B. Experimental Results

All test sessions were concluded with success, in the sense that all test subjects were able to perform the tasks required from them.

1) Eye Tracker Calibration Process: The calibration process revealed itself as the most challenging step of the testing session. A calibration was deemed good enough if the accuracy in the areas where the player mostly interacts with in the game, i.e., the whole screen except the top corners (see Fig. 10), was above a given threshold. Two participants had to remove their glasses so that their gaze could be properly detected. Table V shows the distribution of participants according to the number of calibration tries each participant required to obtain a good enough calibration. As the table shows, a single calibration process was enough for half the participants.

After achieving a calibration deemed successful, the participants were asked about their feelings towards it and if they would see themselves repeating this process at home before playing a game. From all the participants, 8 said that they would see themselves doing it, stating that a calibration process also had to be performed with other controllers. Three of these participants reported that the process was easy and fast, while others stated that it should be easier, but it is still acceptable. The other 2 participants did not see themselves calibrating an eye tracker each time they wanted to play, expressing that they wished the process was easier.

2) Play Sessions: The three 2 minute play sessions produced results that suggest an approximation to the ideal game flow, with the participant being able to progressively learn how to use play more competently. This can be observed in Tables VI, VII and VIII which show how the averages of all participants for the ratio of enemies killed and noticed obstacles and enemies, along with the number of deaths, evolved along the play sessions. The two ratios evolved positively, with the player improving in the tasks of killing enemies and noticing them, along with obstacles. The number of deaths

| Number of tries | Number of participants |
|-----------------|-----------------------|
| 1               | 5                     |
| 2               | 1                     |
| 3               | 2                     |
| 4               | 2                     |
The results were treated in the same way for the GEQ post-game module, whose results are summarised in Table IX. The differences in levels of positive and negative experience are too small ($\approx 0.05$) to allow any analysis from them. The more significant difference in Tiredness and Returning to reality components provide additional support to the idea that eye tracking renders the experience more immersive and engaging. The similar values obtained with and without the eye tracker suggest that the participant’s perception of the overall experience was more shaped by the game itself than by the use or lack of the eye tracker. Still, these results become more meaningful and easier to understand when compared with the answers the test subjects gave to the last set of informal questions (see below). The main goal of these questions was to extract from the participants more subjective perceptions they had from the game, which could help us understand the GEQ scores.

3) Informal Questions: The first question asked about the participant’s opinion about the used visual effects signalling which objects were labelled as noticed. Nine of the ten participants stated the importance of the effects as a means to give feedback to the player, with some pointing out that without effects the player may be forced to look more than needed as the player never knows if the obstacle or enemy component better graded with eye tracking on. participants reported higher levels on the component of Tension/annoyance with the eye tracker in use, which reveals the full game experience can lead to greater levels of frustration. This may be the result of the game being more challenging with eye tracking or of the issues related to the accuracy of the eye tracker (more on this below). Both positive and negative affects are slightly higher when eye tracking was on, suggesting that players were more emotionally invested in that condition. In both conditions, the positive affect is considerably higher than the negative one, suggesting that the game provides an overall positive experience.

Although GEQ scores are used in this study to compare two designs, we can also analyse the absolute meaning of the obtained scores. In this regard, although the GEQ scores are low when compared to the maximum of the scale, these are not substantially different from the ones obtained for full-fledged commercial FPS games, reported in a previous study [34]. The lower scoring can be explained by the awareness players have of the current state of the art in commercial games.

The GEQ was filled out by the participants after these three play sessions. As mentioned, after these sessions the participants were asked to play an additional set of three play sessions, this time without using eye tracker, i.e., with all obstacles and enemies automatically labelled as noticed. Table IX compares, for each of the components of GEQ core module (competence, sensory and imaginary Immersion, flow, tension/annoyance, negative affect, positive affect), the sessions with and without eye tracking. In the Competence component, the participants felt more competent and able to complete tasks without the eye tracker than with it, which can be explained by the higher degree of difficulty of the game when eye tracking is on. In the Sensory and imaginative immersion component, the participants reported higher scores with eye tracking, which means the feeling of immersion was felt at a greater level with the full game experience. The Flow component was also higher with eye tracking on, which reveals the participants felt a stronger feeling of game flow, a better balance within the game’s Challenge, another

| Play Session | Proportion of killed enemies [%] |
|--------------|--------------------------------|
| 1st          | 65.6 ± 4.4                     |
| 2nd          | 69.5 ± 7.2                     |
| 3rd          | 79.0 ± 5.0                     |

| Play Session | Proportion of noticed elements [%] |
|--------------|-----------------------------------|
| 1st          | 49.7 ± 6.3                        |
| 2nd          | 49.8 ± 6.1                        |
| 3rd          | 52.7 ± 5.4                        |

| Component | GEQ Average Score With ET | GEQ Average Score Without ET |
|-----------|---------------------------|-----------------------------|
| Competence| 2.22 ± 0.27               | 2.8 ± 0.25                  |
| Immersion | 1.33 ± 0.3                | 0.93 ± 0.3                  |
| Flow      | 2.00 ± 0.24               | 1.68 ± 0.26                 |
| Tension   | 1.06 ± 0.28               | 0.3 ± 0.14                  |
| Challenge | 1.50 ± 0.2                | 0.58 ± 0.19                 |
| Negative affect | 0.65 ± 0.2 | 0.43 ± 0.21 |
| Positive affect | 2.47 ± 0.26 | 2.26 ± 0.22 |

Table VI

Participants’ average proportion of killed enemies versus total of spawn enemies, for each play session. Results provided as STD ± STE, where STD stands for standard deviation and STE for standard error of the mean.

Table VII

Participants’ average proportion of noticed elements (obstacles and enemies) versus total of spawn elements (obstacles and enemies), for each play session. Results provided as STD ± STE, where STD stands for standard deviation and STE for standard error of the mean.

Table VIII

Participants’ average number of deaths, for each play session. Results provided as STD ± STE, where STD stands for standard deviation and STE for standard error of the mean.
they would not in the current state but that they could try it in the future, stating that depending on the game it could facilitate precision tasks such as aiming in FPS games or passing the ball to another player in a football game. The reasons these participants presented as to holding off in the adoption of the technology regarded its poor performance while wearing glasses, and the fact that it was not stable or precise, which allied with the calibration process ruined the experience. The other five participants said they would adopt the technology, stating that it was a new form of interaction, that opened new possibilities and created more immersive experiences.

These answers showed, along with the previous questionnaires, that the use of eye tracking in games has both pros and cons. On the positive side, the use of gaze-oriented gameplay provided a more immersive and richer experience, providing a better game flow. On the negative side, the technology’s limitations raised feelings on the participants that were not desired. The calibration process and its results, along with some disbelief that eye tracking could have an important role in a computer game, are the main reasons for the participants being adamant about adopting the technology.

V. Conclusions and Future Work

This paper presented Zombie Runner, an endless runner FPS game, whose core mechanics and procedurally generated content are modulated by the player’s gaze estimated with an affordable eye tracker. A set of testing sessions were carried out in order to assess the impact of eye-tracking in the player’s satisfaction when playing Zombie Runner. The results obtained from testing sessions show that the use of eye tracking provides a more challenging and immersive experience to the player. These results complement previous studies, which were mostly focused on performance-based metrics when using eye tracking as a direct control input. Conversely, Zombie Runner exploits eye tracking as a mechanism to match the avatar’s attention model with the one of the player. Moreover, Zombie Runner also exploits eye tracking to guide the procedural content generation of the game’s environment, contributing in an original way to the emerging field of Experience-Driven Procedural Content Generation (EDPCG) [24].

During the evaluation, a strong correlation between problems that surfaced with the eye tracker calibration and participants’ overall experience was observed. Participants for whom the hardware worked without major flaws reported better levels of satisfaction when contrasting with participants for whom the calibration process was not perfect or took a longer time. Among the participants’ complaints about the eye tracking technology, many were related the need to have the head mostly static during the play session, the calibration process requiring repetitions to meet the required accuracy, and overall eye tracker’s lack of precision. These complaints show that the affordable eye tracking technology still has to grow and develop until it is in a state that can be accepted by the overall gaming community. However, positive player’s experience when calibration was easily attained is a sign that the method will be a valuable asset for the game designer as soon as the eye tracking technology matures.
As future work, we intend to validate the use of eye tracking in other types of video games and allow for free avatar’s movement. We also intend to extend this testing framework, including the philosophy behind the way the player’s attention is integrated in the core gameplay, to games with other types of camera perspective, such as third-person games. We pretend to perform a more intensive set of tests, enlarging the participant population, in order to obtain more robust statistics.

REFERENCES

[1] J. Antunes and P. Santana. Gaze-oriented gameplay in first-person shooter games. In Proceedings of the 24th Portuguese Meeting on Computer Graphics and Interaction (EPCGI), pages 231–232, 2017.
[2] L. Javal. Essai sur la physiologie de la lecture. Annales d’Oculistique, 82:242–253, 1879.
[3] P. S. Holzman, L. R. Proctor, and D. W. Hughes. Eye-tracking patterns in schizophrenia. Science, 181(4095):179–181, 1973.
[4] David P. McMullen, Guy Hotson, Kapil D. Katyal, Brock A. Wester, Matthew S. Fifer, Timothy G. Mcgee, Andrew Harris, Matthew S. Johannes, R. Jacob Vogelstein, Alan D. Ravitz, and et al. Demonstration of a semi-autonomous hybrid brain-machine interface using human intracranial eeg, eye tracking, and computer vision to control a robotic upper limb prosthetic. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22(4):784–796, 2014.
[5] J. Gomes, F. Marques, A. Lourenço, R. Mendonça, P. Santana, and J. Barata. Gaze-directed telemetry in high latency wireless communications: The case of robot teleoperation. In Proceedings of the 42nd Annual Conference of the IEEE Industrial Electronics Society (IECON), pages 1–6. IEEE, 2016.
[6] Dean M Krugman, Richard J Fox, James E Fletcher, Paul M Fischer, and Tina H Rojas. Do adolescents attend to warnings in cigarette advertising?: An eye-tracking approach. Journal of advertising research, 34:39–39, 1994.
[7] J. David Smith and T. C. Nicholas Graham. Use of eye movements for video game control. In Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology - ACE ’06, 2006.
[8] J Leyba and J Malcolm. Eye tracking as an aiming device in a computer game. Course work (CPSC 412/612 Eye tracking Methodology and Applications by A.Duchowski), 2004.
[9] P Isokoski and B Martin. Eye tracker input in first person shooter games. In Proceedings of COGAIN 2006: Gazing into the Future, page 78–81, 2006.
[10] Poika Isokoski, Markus Joos, Oleg Spakov, and Benoit Martin. Gaze controlled games. Universal Access in the Information Society, 8(4):323–337, May 2009.
[11] Michael Dorr, Laura Pomarjanschi, and Erhardt Barth. Gaze beats mouse: a case study. Psychology Journal, 7(2):197–217, 2009.
[12] Kristian Lukander. Measuring gaze point on handheld mobile devices. Extended abstracts of the 2004 conference on Human factors and computing systems - CHI ’04, 2004.
[13] Kohei Arai and Ronny Mardiyanto. Eye-based hci with full specification of mouse and keyboard using pupil knowledge in the gaze estimation. 2011 Eighth International Conference on Information Technology: New Generations, 2011.
[14] William Charles Wells. An essay upon single vision with two eyes together with experiments and observations on several other subjects in optics. 1792.
[15] Raymond Dodge and Thomas Sparks Cline. The angle velocity of eye movements. Psychological Review, 8(2):145–157, 1901.
[16] H Singh and J Singh. Human eye tracking and related issues: A review. International Journal of Scientific and Research Publications (IJSRP), 2(9), Sep 2012.
[17] Jeffrey S. Shell, Roel Vertegaal, Daniel Cheng, Alexander W. Skaburskis, Changuk Sohn, A. James Stewart, Omar Audeh, and Connor Dickie. Ecsyglases and eyepieces. In Proceedings of the Eye tracking research and applications symposium on Eye tracking research and applications - ETRA’2004, 2004.
[18] John D. Smith, Roel Vertegaal, and Changuk Sohn. Viewpointer. In Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST ’05, 2005.
[19] M Perreira da Silva, V Courboulay, and A Prigent. Gameplay experience based on a gaze tracking system. In Proceedings of COGAIN 2007, 2007.