Gaze-Oriented Gameplay in First-Person Shooter Games

João Antunes, Pedro Santana
ISCTE-Instituto Universitário de Lisboa (ISCTE-IUL)
Instituto de Telecomunicações (IT)
Lisboa, Portugal
{jmmea,pedro.santana}@iscte-iul.pt

Abstract—The use of eye tracking systems in video games is still at an early stage. Commercial eye trackers and researches have been focusing in gaze-oriented gameplay as an alternative to traditional input devices. In this paper, instead of using eye tracking as a simple direct control input, we propose to use it in order to control the attention of the player’s avatar and the game’s procedural content generation. To demonstrate the value of this proposal, we developed and herein present the first-person shooter Zombie Runner. Preliminary informal tests showed that the game handles properly the eye tracking input. Moreover, subjects found the game more enjoyable when the gameplay was gaze-oriented.

Keywords—video games; eye tracking; gaze-oriented gameplay

I. INTRODUCTION

Several studies have addressed the use of eye-tracking integration in video games as a control input alternative to traditional controls [1-5]. These previous studies have provided contradictory results regarding the effectiveness of the use of eye tracking in video games as a direct control input, which may suggest that eye tracking is not best suited to direct input control.

Bearing the limitations of using 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 controlling a pointer with the eyes, which has no mapping to real life.

To demonstrate the value of the herein proposed alternative uses of eye tracking in video games, we developed and tested our own First-Person Shooter (FPS), Zombie Runner. In this game, shot accuracy, automatic obstacle avoidance, and obstacle spawn probability are controlled as a function of the player’s gaze. The game and its validation are described in the following sections.

II. ZOMBIE RUNNER

Zombie Runner is a procedurally generated endless-running FPS, developed in the Unreal Engine, that asks the player to survive by running along an endless corridor while avoiding obstacles (see Fig. 1). The moment the game starts, the player is running forward in a straight line along the corridor. Using an Xbox 360 controller, the player is able to orientate the gun so as to shoot incoming zombies (an overlaid crosshair represents to where the player is pointing the gun). The player’s gaze is estimated using an affordable 60Hz Gazepoint GP3 Eye Tracker.

Zombies die with one shot if previously noticed by the player and with two shots otherwise, simulating the relation between accuracy and attention. A zombie, as any other object in the game, is considered to be noticed if the accumulated time spent by the player gazing at the object is of at least 0.5 seconds. By considering accumulated time, instead of consecutive time, players are allowed to naturally alternate their attention over more than one object. Moreover, this way the system is able to report noticed objects even when facing jitter in the gaze estimates.

A non-killed zombie may result in a collision and, consequently, in the zeroing of the player’s avatar health. Conversely, colliding against a noticed zombie drains only one third of the player’s health. The intuition is that a last minute partial dodge of a noticed zombie is plausible and, so, it should be included in the game mechanics.

If noticed, rocks and tree branches (see Fig. 2) are automatically jumped over and ducked by the player’s avatar, respectively. If the player fails to notice these obstacles, a collision occurs and one third of the player’s health is drained. This intends to simulate the fact that people can only avoid obstacles that have been seen early enough. If a collision against an obstacle/zombie happens to occur, visual and audio feedback is provided to the player in the form of blood stains on the screen (see Fig. 2) and a shout from the player’s avatar. The game ends when the player’s health reaches zero. If defined in the game

Fig. 1. Zombie Runner’s hardware setup.
settings, visual effects are added to objects once they are noticed by the player (see Fig. 2). This aims at providing an immediate reward to players when they comply with the game rules, that is, when they gaze the objects of relevance.

In addition to modulate the avatar’s behavior and shooting accuracy, eye tracking is also exploited to control the procedural generation of the environment. Concretely, if the player’s gaze is concentrated in a part of the screen for too long, obstacles and zombies will start getting generated in other parts of the screen. This stimulates the player to better explore the environment.

III. EVALUATION

A set of informal tests were carried out so as to determine the best game’s parameterization (i.e., the avatar’s motion speed and the time spent gazing at an obstacle required to accept it as noticed) and to assess whether the system handles eye tracking input correctly. These tests were conducted with a group of 5 male subjects, with ages from 24 to 26, with 3 of them being avid gamers used to playing FPS games with controllers, one used to play FPS games with keyboard and mouse and one not used to play video games. The eye tracker did not work correctly with one of the subjects, who had previously done a retinal correction medical surgery.

Some of the subjects were handed the game in auto mode (i.e., without gaze-oriented gameplay, meaning with all obstacle being automatically avoided by the system even if unnoticed by the player) before playing it with its default options (i.e., with gaze-oriented gameplay). These subjects reported that in auto mode they felt that some of the automatically avoided obstacles were not noticed by them. Moreover, they stated that whenever they died or lost health with the default settings, they had not actually noticed the obstacle causing the collision and, so, they did not feel betrayed by the game. Therefore, overall, the subjects reported positive feedback in terms of usability and found the game more enjoyable when gameplay was gaze-oriented. The 0.5 seconds assigned to the time the player has to focus on an obstacle for the system to mark it as noticed was found to produce no false positives and no situations where the player felt he had noticed an obstacle that the system did not.

IV. CONCLUSION

Zombie Runner, a gaze-directed FPS video game was presented. In this game, instead of using eye tracking as a simple direct control input, it is used so as to control the attention of the player’s avatar and the game’s procedural content generation. Technical limitations impose that concepts such as peripheral vision had to be disregarded in Zombie Runner, but other technical limitations, such as screen size, helped to mitigate this problem. The set of preliminary informal tests showed that eye tracking is properly handled by the game and it actually renders the game more enjoyable to play. As future work, we intend to validate the use of eye tracking in other types of FPS games, more complex environments, and to allow for free avatar’s movement. Moreover, we also intend to perform a set of formal tests to better assess the impact of eye tracking in FPS games. The higher the accuracy of the eye tracker the more detailed gaze-oriented adjustments can be considered in the gameplay. Hence, we believe that the advance of eye tracking technology will further uncover the value of gaze-oriented gameplay.

REFERENCES