Running Eye – An Auto-Runner Game with Eye-Tracker Controls

Fabian Zürcher
University of Regensburg
Germany
fabian.zuercher@stud.uni-regensburg.de

Michael Hebeisen
University of Regensburg
Germany
michael.hebeisen@stud.uni-regensburg.de

Jonathan Seibold
University of Regensburg
Germany
jonathan.seibold@stud.uni-regensburg.de

ABSTRACT

Eye tracker enable multiple possibilities for humans with reduced arm, hand or finger mobility. We developed a game where the gameplay is controllable with only an eye tracker. The mouse and keyboard controls are only optional. The game is a kind of auto-runner whose aim is to reach the game end or to reach a new high score. It is inclusive as it is playable without reading ability, arm, hand or finger mobility and on cheaper low-level computers. It enables humans with no prior gaming experience to enjoy gaming through low entry hurdles while still encouraging more experienced players as archivers for longer periods of time to gather the highest score possible and succeed with adjusted difficulties. It strengthens enjoyment possibilities for a broad and inclusive range of people while avoiding much time consumption for a single game run and therefore easily fitting into everyone's daily routine. In two play test sessions, we evaluated the validity of dwell time selection and head rotation interaction for our game concept.

CHI'19 Extended Abstracts, May 4-9, 2019, Glasgow, Scotland, UK.

ACM ISBN 978-1-4503-5971-9/19/05.

KEYWORDS
Eye-Tracker; Auto-Runner; Game Development

INTRODUCTION

Video games are part of modern daily lives. What sets them apart from other media like television or films is their interactivity. As a part of that, input devices are a crucial part of games, because they enable player interaction. Common input devices are game controllers for consoles and mouse and keyboard for games on the computer. These common devices are all controlled by the player using his/her hands. Of course, this excludes or at least limits people with reduced hand or finger mobility when playing video games.
The emergence of eye trackers suited for home use is a chance to create games with different input methods, therefore including people limited by conventional controllers. We developed Running Eye, an auto-runner game which can be controlled using only an eye tracker.

RELATED WORK

Dechant et al. compared interaction mechanisms including mouse, controller and gaze interaction, as well as two hybrid methods. They conducted a Fitt's Law study evaluating the performance of aforementioned interaction mechanisms. Results substantial for our game design were the times of gaze control, which were significantly the slowest out of all evaluated interaction mechanisms. Gaze control was also the most susceptible to errors. [1]
Jacob explored ways to include eye tracking technology in a natural way in the process of human computer interaction. He discovered, that the limitations of eye tracking input does not lie in the technology, but in creating an interaction process eliminating human and technical factors which occur with eye tracking input and stand in the way of smooth human computer interaction. One of these factors is the "Midas Touch" problem. It occurs, because the eye tracking system cannot distinguish "exploring" gaze and input gaze. Therefore, every gaze at an interactable object triggers the object's functionality, which leads to frequent misinputs. [2]
Because of "Midas Touch" [2] and slow and error-prone controls with eye trackers [1], we assumed, that players will make many errors based on our chosen interaction technology eye tracking. Because we cannot prevent these errors, we included them in our game design. We created ways players can recover from errors, like the stop character mechanic, additional interactables, if the first interactable is moved to a wrong position, and a "safe point", which, when looked at, deactivates all interactables. It is also part of our concept, that players learn how to overcome obstacles by failing, restarting and trying again. Therefore, a restart button is essential to our user interface.
Krol et al. implemented alternative controls for Tetris, using an eye tracker and a passive brain-computer interface. Eye tracker inputs were based on dwell time, which means the player has to look at an element for a certain, minimal amount of time to select it. This mechanic was used in the Tetris context to change the column of the tetromino by looking at the column for 100 ms, to drop the tetromino by looking at a column for 1 second and to rotate the tetromino by looking at the top of the playing field for 100 ms. The elements game speed and errors were handled with the brain-computer

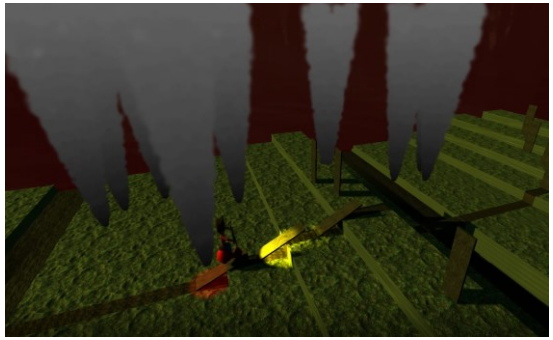

Figure 1: Character chased by a pollution tornado, game over if the tornado catches up to the character.

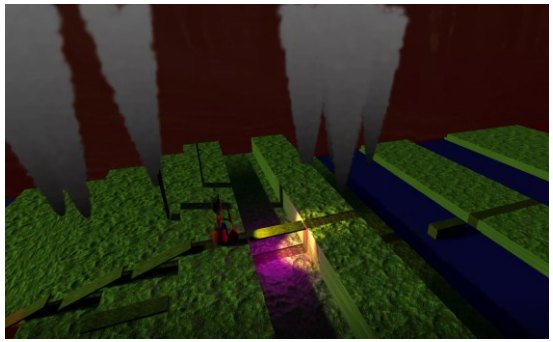

Figure 2: Character at the obstacle bridge, the bridge object is activated. When activated, the bridge collapses and the character can no longer pass. Therefore, the bridge should be ignored and not gazed at.

interface. [3] For Running Eye, we also use dwell time to activate objects and used this paper when deciding on the magnitude of dwell time.
Velloso and Carter discuss different game mechanics which can be controlled by eye tracker. These mechanics are navigation (moving the avatar through a virtual world), aiming and shooting, selection and commands (selecting objects and commanding them), implicit interaction (NPC interaction, triggering environmental effects, adapt the game AI) and visual effects. [4] We locate the interaction used in Running Eye in the section selection and commands. Subsections include picking up and dropping objects and action activation. [4] Activating physics for a focused object classifies as action activation, moving activated objects around with head movement is placed with pick and drop.

IMPLEMENTATION

For this project, we wanted to apply an alternative interaction technology to a common game genre. To include people with handicapped hands or arms, who are not able to play games with usual input methods, like controllers, we decided on an eye tracker as input device. To manage real time eye tracking, the Tobii Eye Tracker 4C is well suited. It comes with the added benefit of an extended Unity development SDK. Additionally, we implemented a standard input method with mouse and keyboard, because we wanted to be able to compare both input technologies for our game. For the game's genre, we chose an auto runner. However, we had to adapt the genre's usual properties to better match the new user inputs.

Gameplay

In our game, the protagonist Volker flees from a toxic tornado and has to overcome several obstacles that are in his way. Volker cannot stop running, so the player has to clear the path in front of him so that Volker does not get caught by the tornado seen in figure 1. The obstacles can be manipulated by the player by looking at them (thus activating an object of the obstacle) and by turning their head which adds a force on the head turn's direction. Other obstacles have to be managed by actively ignoring them to not activate a contra productive mechanism like the collapsing bridge in figure 2.
With mouse and keyboard controls, the player can activate an object by dragging the pointer on the required object. The player is under pressure because the time to clear an obstacle is limited. If Volker runs into a wall, the toxic cloud will catch up to him, which ultimately leads to a game over.

Interaction

Looking at an object in an obstacle activates physics for said object. A dwell time is applied to this gaze and if the player looks at the object for this set amount of time, the object is active. If the player does not look at an active object for a longer amount of time, the active effect wears off and the player has to activate the object again if she/he wants to manipulate it. With mouse and keyboard controls, an object is active if the pointer is on the object, we did not include dwell times with this interaction technique.

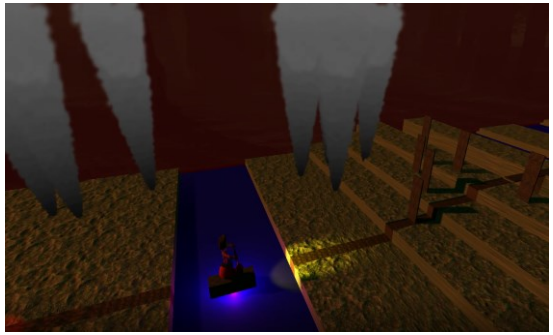

Figure 3: Character at the obstacle boat, boat object is activated. The boat can be moved left or right with head rotation.

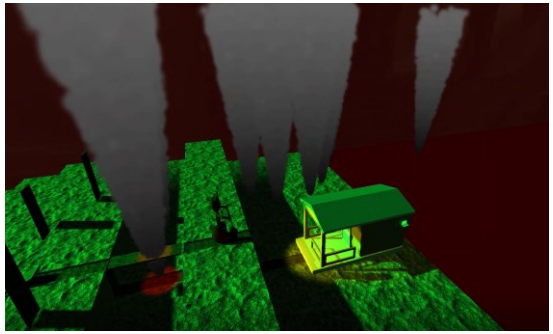

Figure 4: Character after the last obstacle in the run reaching the hut (at the right). Lighting has changed to a bright green.

In the case of an unstable initial position of the object, this can be enough to start a mechanism. If the object's initial position is stable, the player is able to add forces in four different directions to actively start the mechanism, like with the boat in figure 3. These forces are applied by turning the player's head in the corresponding direction:

- Head turned left (keyboard controls: arrow left): force to the left
- Head turned right (keyboard controls: arrow right): force to the right
- Head turned up (keyboard controls: arrow up): force to the back
- Head turned down (keyboard controls: arrow down): force to the front

If the player selects a new object, physics still stay active for the old object, but forces only apply to the new one. With eye-tracker controls, only one force can be applied at a time. This is always the force of which the direction aligns with the strongest head turn direction.

Adjustments

In most auto-runner games, the pace of the character is fairly high, which contributes to the difficulty in playing the game with conventional inputs. When controlling an auto-runner via eye-tracker, we found it necessary to slow down this pace. This is due to the fact that eye-tracker input still is not as precise compared to other input methods. On top of that, we found fail inputs to be more prevalent with eye-tracker input, for example because of the Midas Touch problem. Correcting fail inputs needs time which is given to the player by slowing down the pace.

However, we found problems in pacing even with this lower game speed. Players frequently needed more time to manage an obstacle or to recover from a bigger mistake. Since we did not want to further slow down the game, we added the possibility to temporarily stop the character by looking at it or dragging the mouse pointer onto it. After the difficult obstacle is cleared or the mistake is remedied, the character moves on by again looking at or dragging the pointer onto it. To not make this option game breaking, because the player can always stop if in jeopardy, the poisonous cloud still moves towards the character if he stops. Therefore, the player can use the stop option only for a limited amount of time.

Art Style

Running Eye has a 2.5 dimensions style, which means that objects are designed in 3D, but interaction takes place in 2D. We decided on a minimalistic, geometrical design of objects to avoid distracting the player. Distracting details possibly increase the Midas Touch problem.

Scene lighting is an important factor to scene atmosphere. In Running Eye, we use lighting to indicate progress by changing the color and brightness of the lighting based on traveled distance. The atmosphere changes from a threatening dark red, from which the character flees, to a bright green, which welcomes the character, as seen in figure 4. This change represents the transition from a polluted city environment to a liberating nature setting.

Figure 5: Play test setup with a gaming laptop and a Tobii Eye Tracker 4C.

Partcipant ID	Number of Game Overs	Time to reach the tatget (in minutes)
1	3	4:00
2	7	6:21
3	6	5:45
4	2	5:23
5	4	6:07
6	9	9:33

Table 1: Number of game overs and times to reach the target for all 6 participants in the second play test.

PLAY TESTS

We conducted two play tests while developing Running Eye. During the first play test, basic game mechanics were implemented. Controls worked with both eye-tracker and mouse and keyboard, but were in an early stage. Content wise, the game contained four fully functional obstacles and a target, which ended the game when reached. The level was auto generated at every start of the game, the order of obstacles was random. The length of the level could also be modified.

In the first play test, we set the length of the level to ten. For this first study, the focus was the degree of difficulty and the pace of the game. We had five participants, who had to reach the target in the game using eye-tracker controls, as shown in the setup in figure 5. If they got stuck, they had to newly start the game. We told the participants to think aloud while playing and also conducted a short interview after the play test, concerning difficulty and pace, as well as general observations the participants made. We also measured the play time of each participant.

The first play test showed, that players developed the strategy of trying to learn each of the obstacle and, in order to do this, restart the game a couple of times before attempting to reach the target in a perfect run. With this strategy, every participant was able to reach the target in a time between 2:44 minutes and 8:11 minutes. Players mentioned possible improvements in the player model (which was only a stand-in and not our final design) and in using more diverse colors. To reach a threatening atmosphere, we have a mainly dark, red lighting, which can appear monotonous. The input via turning the head proved to be not reliable enough to guarantee smooth controls. Therefore, we had to optimize eye-tracker controls. One obstacle, called canyon could not be reliably solved by players because there was not enough time. Because we already slowed down the game, this problem was why we decided to implement a mechanic to temporarily stop the character. In the interviews, participants stated that they liked the graphic style and that for them, realistic graphics are not important for games like this.

We conducted the second play test when the game was in a more advanced development state. Both input types were optimized and in conjunction with the mechanic to stop the character, it was possible to reliably beat the game after a short period of training and getting used to the obstacles. We tried to eliminate irritations from the first play test by adding our final character model and added a progress based lighting color change. Now, the character started in a threatening dark red environment, but as he runs towards his home, the target, lighting changes to a cheerful bright green. In addition to the changing color, progress could also be noticed by the UI, which now also showed the distance travelled. In this state of game development, our main work was designing more obstacles to create richer game content.

For the second play test, we again set a level length of ten obstacles. Our six participants had to use the eye-tracker controls to eventually reach the target. In this iteration, we were mostly interested in the difficulty of the individual obstacles. We recorded, at which obstacles players stopped the character and at which obstacles occurred a game over. Of course, we also wanted to notice other conspicuities, which is why we asked the participants to think aloud while playing.

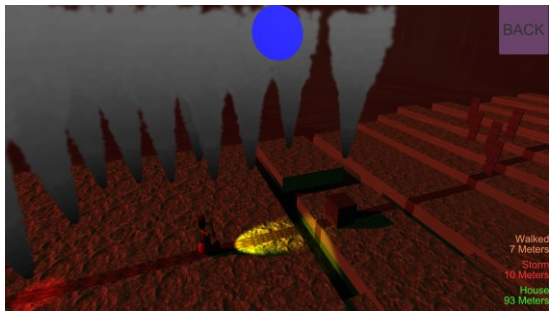

Figure 6: Game scene in eye tracker control mode, containing a blue safe point on top of the screen.

Our six participants started the game a total of 31 times, every participant was able to reach the target. So in this study, 25 game overs occurred, listed in table 1. The most difficult obstacles were the mountain (12 game overs) and the canyon (9 game overs). At the tree obstacle, players had far less problems, so this obstacle did not lead to a single game over. With regards to stopping the character, players made most use of this mechanic at the canyon obstacle. At this particular obstacle, players stopped the character a total of 33 times, which is by far the greatest use of the stop character function. This result goes along with the findings of the first play test, which showed that there was time shortage for the canyon. But because the canyon no longer caused the most game overs, we think our actions concerning this obstacle were successful.

CONCLUSION & FUTURE WORK

We can conclude that our concept for Running Eye works. It is possible to play with both input methods, eye tracker and mouse/keyboard. We can confirm the viability of dwell time as selection method for eye tracker interaction. Head rotation interaction with an eye tracker is also functional, but has limitations in our use case. Because it is not possible to always accurately get the gaze point when the head is turned, this has a negative effect on object selection. To compensate for this, we implemented a longer cooldown time for an object to deactivate and experimented with the concept of a safe point, which deactivates all selected objects. We did not fully evaluate this concept, which is shown in figure 6 in its current state, while working on Running Eye, so future studies concerning methods to deactivate objects are needed. It is also conceivable to replace the as game object manifested safe point with a certain gesture to get the same deactivation effect.

With regards to our game Running Eye, it is possible to design more obstacles and evaluate their suitability for eye tracker interaction. Concerning the concept, further research about combining head rotation interaction and dwell time based selection is needed. Alternatively, a valid alternative to head rotation interaction could be presented.

REFERENCES

[1] Martin Dechant, Ian Stavness, Aristides Mairena and Regan L. Mandryk. 2018. Empirical Evaluation of Hybrid Gaze-Controller Selection Techniques in a Gaming Context. In *Proceedings of the* CHI Play '18. Melbourne, VIC, Australia, pp. 73–85. DOI: https://doi.org/10.1145/3242671.3242699

[2] Robert J.K. Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the* CHI '90, pp. 11–18.

[3] Laurens R. Krol, Sarah-Christin Freytag, and Thorsten O. Zander. 2017. Meyendtris: A Hands-Free, Multimodal Tetris Clone using Eye Tracking and Passive BCI for Intuitive Neuroadaptive Gaming. In *Proceedings of the* 19th ACM International Conference on Multimodal Interaction (pp. 433437). New York, NY, USA: ACM. DOI: https://doi.org/10.1145/3136755.3136805

[4] Eduardo Velloso, Marcus Carter. 2016. The Emergence of EyePlay: A Survey of Eye Interaction in Games. In *Proceedings of the* CHI PLAY '16. Austin, TX, USA. DOI: http://dx.doi.org/10.1145/2967934.2968084