

Evaluating Hands-on and Hands-free Input Methods for a Simple Game

Mehedi Hassan¹, John Magee², and I. Scott MacKenzie¹

¹ York University, Toronto, ON, Canada

² Clark University, Worcester, MA, USA

Abstract. We conducted an experiment to test user input via facial tracking in a simple computerized snake game. We compared two hands-on methods with a hands-free facial-tracking method to evaluate the potential of a hands-free point-select tool: *CameraMouse*. In addition to the experiment, we conducted a case study with a participant with mild cerebral palsy. For the experiment we observed mean game scores of 5.89, 3.34, and 1.94 for the keyboard, touchpad, and *CameraMouse* input methods, respectively. The mean game score using *CameraMouse* was 0.55 during the case study. We also generated trace files for the path of the snake and the cursor for the touchpad and *CameraMouse* pointing methods. In a qualitative assessment, all participants provided valuable feedback on their choice of input method and level of fatigue. Unsurprisingly, the keyboard was the most popular input method. However, positive comments were received for *CameraMouse*. Participants also referred to a few other simple games, such as *Temple Rush* and *Subway Surfer*, as potential candidates for interaction with the *CameraMouse*.

Keywords: Hands-free input · Face tracking · CameraMouse · Gaming

1 Introduction

The concept of human-computer interaction (HCI) is no longer confined to only physical interaction. While actions such as hovering the cursor of a mouse or touch-pad with our hands, clicking a button, typing, and playing games with our fingers are common forms of interaction, performing such tasks without physical touch is an intriguing idea. This is interesting both for non-disabled users and also for physically-challenged users.

We conducted an experiment with a simple yet well-known game: *Snake*. Since MIT student Steve Russel made the first interactive computer game, *Spacewar* in 1961 [7], computer games have evolved considerably. But, we did not conduct our experiment with a game that requires expensive graphics equipment or intense user interaction. For our study, the game was played with two hands-on methods and a hands-free method with facial tracking. We have re-created a version of *Snake* for this experiment. Our version is named *Snake: Hands-Free Edition*. This is a *Windows*-platform game. It can be played with three different input methods: keyboard, touchpad, and *CameraMouse*. The first two methods

are hands-on and the third method is hands-free. By conducting this experiment, we explore the opportunities for accessible computing with hands-free input for gaming.

As part of our user study, we conducted a case study as well. The participant for this case study has mild cerebral palsy. Due to his physical condition, the case study was only conducted for the hands-free phase of our gaming experiment.

2 Related Work

Cuaresma and MacKenzie [3] compared two non-touch input methods for mobile gaming: tilt-input and facial tracking. They measured the performance of 12 users with a mobile game named *StarJelly*. This is an endless runner-style game. Players were tasked with avoiding obstacles and collecting stars in the game. A substantial difference was observed in the mean scores of the two input methods. Tilt-based input had a mean score of 665.8 and facial-tracking had a mean score of 95.1.

Roig-Maimó et al. [12] present *FaceMe*, a mobile head tracking interface for accessible computing. Participants were positioned in front of a propped-up *iPad Air*. Via the front-facing camera, a set of points in the region of the user’s nose were tracked. The points were averaged, generating an overall head position which was mapped to a display coordinate. *FaceMe* is a picture-revealing puzzle game. A picture is covered with a set of tiles, hiding the picture. Tiles are turned over revealing the picture as the user moves her head and the tracked head position passes over tiles. Their user study included 12 able-bodied participants and four participants with multiple sclerosis. All able-bodied participants were able to fully reveal all pictures with all tile sizes. Two disabled participants had difficulty with the smallest tile size (44 pixels). *FaceMe* received a positive subjective rating overall, even on the issue of neck fatigue.

Roig-Maimó et al. [12] described a second user study using the same participants, interaction method, and device setup. Participants were asked to select icons on the *iPad Air*’s home screen. Icons of different sizes appeared in a grid pattern covering the screen. Selection involved dwelling on an icon for 1000 ms. All able-bodied participants were able to select all icons. One disabled participant had trouble selecting the smallest icons (44 pixels); another disabled participant felt tired and was not able to finish the test with the 44 pixel and 76 pixel icon sizes.

UA-Chess is a universally accessible version of the Chess game developed by Grammenos et al. [5]. The game has four input methods: mouse, hierarchical scanning, keyboard, and speech recognition. The hierarchical scanning method is designed for users with hand-motor disabilities. This scanning technique has a special “marker” that indicates input focus. A user can shift focus using “switches” (e.g., keyboard, special hardware, voice control). After focusing on an object another “switch” is used for selection. It currently supports visual and auditory output. A key innovative feature of this game is that it allows multi-player functionality.

Computer gaming inputs range from simple inputs to very intense interactions. For the hands-free interaction described herein, we focused not on the complexity or visual sophistication of the game, but rather on how participants would fare in a recreational environment with *CameraMouse*. Hence, we chose a simple game, *Snake*, similar to the simple games evaluated by Cuaresma and MacKenzie [3] and Grammenos et al. [5].

Our research on accessible computing was also aided by a literature review on “point-select tasks for accessible computing”. Hassan et al. [6] compared four input methods using a 2D Fitts’ law task in ISO 9241-9. The methods combined two pointing methods (touchpad, *CameraMouse*) with two selection methods (tap, dwell). Using *CameraMouse* with dwell-time selection is a hands-free input method and yielded a throughput of 0.65 bps. The other methods yielded throughputs of 0.85 bps (*CameraMouse* + tap), 1.10 bps (touchpad + dwell), and 2.30 bps (touchpad + tap).

MacKenzie [8] discusses evaluating eye tracking systems for computer input, noting that eye-tracking can emulate the functionality of a mouse. The evaluation followed ISO 9241-9, which lays out the requirements for non-keyboard input devices. Four selection methods were tested: dwell time, key selection, blink, and dwell-time selection. The throughput for dwell-time selection was 1.76 bits/s, which was 51% higher than the throughput for blink selection.

Gips et al. [4] developed *CameraMouse* which uses a camera to visually track any selected feature of the body such as the nose or tip of a finger. The tracked location controls a mouse pointer on the computer screen. These were early days in the development of *CameraMouse*. At this stage, the system did not have any tracking history. Cloud et al. [2] describe an experiment with *CameraMouse* with 11 participants, one with severe physical disabilities. Two application programs were used, *EaglePaint* and *SpeechStaggered*. *EaglePaint* is a simple painting application with the mouse pointer. *SpeechStaggered* is a program allowing the user to spell out words and phrases from five boxes that contain the entire English alphabet. A group of subjects wearing glasses showed better performance in terms of elapsed time than a group of subjects not wearing glasses.

Betke et al. [1] describe further advancements with *CameraMouse*. They examined tracking various body features for robustness and user convenience. Twenty participants without physical disabilities were tested along with 12 participants with physical disabilities. Participants were tested for performance of *CameraMouse* on two applications: *Aliens Game* which is an alien catching game requiring movement of the mouse pointer and *SpellingBoard*, a phrase typing application where typing was done by selecting characters with the mouse pointer. The non-disabled participants showed better performance with a normal mouse than with *CameraMouse*. Nine out of the 12 disabled participants showed eagerness in continuing to use the *CameraMouse* system.

Magee et al. [11] discussed a multi-camera based mouse-replacement system. They addressed the issue of an interface losing track of a user’s facial feature due to occlusion or spastic movements. Their multi-camera recording system synchronized images from multiple cameras. For the user study, a three-camera

version of the system was used. Fifteen participants were tested on a hands-free human-computer interaction experiment that included the work of Betke et al. [1] on *CameraMouse*. They tracked a user's head movement with three simultaneous video streams and software called *ClickTester*. They report that users put more effort into moving the mouse cursor with their head when the pointer was in the outer regions of the screen.

Magee et al. [10] did a user study with *CameraMouse* where dwell-time click generation was compared with *ClickerAID*, which detects a single intentional muscle contraction with an attached sensor. An evaluation tool called *FittsTask-Two*³ [9, p. 291] was used for testing. Ten participants were tested and indicated a subjective preference for *ClickerAID* over dwell-time selection.

3 Software Systems

3.1 Camera Mouse

CameraMouse is a facial tracking application that emulates a physical mouse. Upon enabling *CameraMouse*, a point of the user's face is tracked. The preferable point of focus is usually the nose tip or the point between the eyebrows. If the user moves the tracked point of the face, the cursor moves on the screen accordingly. Figure 1 depicts how *CameraMouse* tracks a point on a user's face.

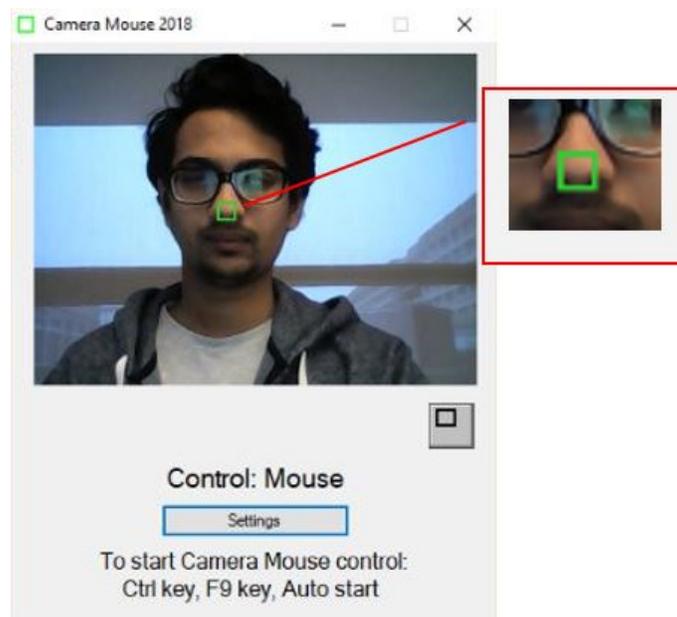


Fig. 1: Face-tracking by *CameraMouse*.

³ <https://www.yorku.ca/mack/HCIbook/>

3.2 Snake: Hands-free Edition

We have remodeled the famous video game, *Snake*. We named our version of the game *Snake: Hands-Free Edition*. The idea remains the same as in the original game. In our version, a snake moves within the bounds of a surface. Two kinds of objects appear on this surface randomly. White objects are *fruits*, the black objects are *poisonous* objects (see Figure 2). Colliding with a *fruit* will increase the snake's length and colliding with a *poisonous* object will kill the snake. The snake also dies if it collides with any wall or obstacle within the bounds of the surface.

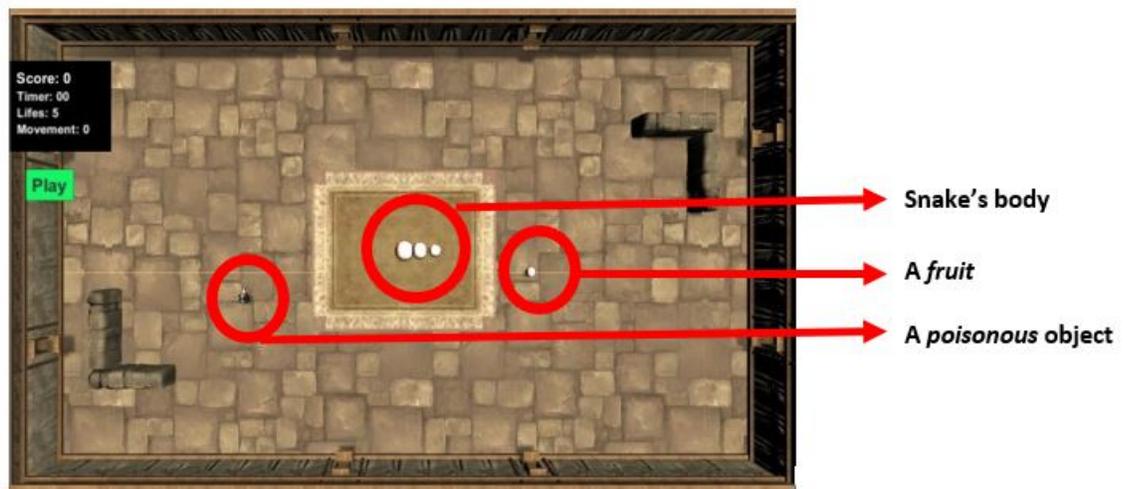


Fig. 2: The snake, the *fruit* and the *poisonous* object of the game.

The snake's speed gradually increases in each trial. The maximum time limit for a trial is one minute. There are three input methods for the movement of the snake: keyboard, touchpad, and *CameraMouse*. While altering directions of the snake with a keyboard, a user needs to press the four arrow keys of a standard keyboard: up, down, right, and left. The game window is divided into four regions (i.e., left, right, up, down) to aid movement with the touchpad and *CameraMouse* (see Figure 3). These regions represent the functionality of the four arrow keys. A user needs to hover the cursor of the touchpad or *CameraMouse* over these regions to change direction of the Snake.

Figure 3 shows how the game screen is divided into regions. If the cursor is in the area marked as left, the snake moves left. Corresponding mappings occurs with the other regions. One significant aspect of the snake's movement is that it is not allowed to move to an exact opposite direction from any direction it is moving towards. For example, if the snake is moving to the right, it cannot directly move to the left by pressing the keyboard's left arrow key or by hovering the cursor on the area marked as left with the touchpad or *CameraMouse*. It

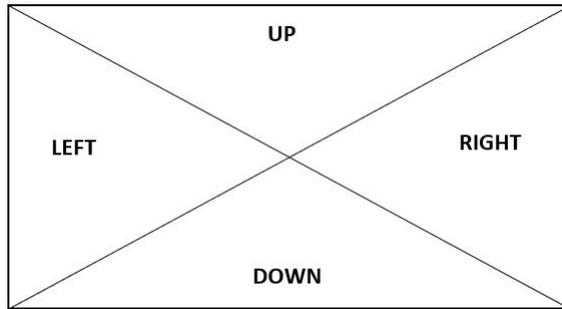


Fig. 3: Four screen regions for cursor control.

would either have to go up or down first and then left to achieve movement to the left. This idea is consistent with how the original *Snake* game is played.

4 Methodology

4.1 Participants

We recruited 12 participants for the experiment. Eight were male and four were female. The participants were university students at undergraduate and graduate levels. They belonged to different geographic regions such as Bangladesh, Canada, India, and Pakistan.

4.2 Apparatus

For hardware, we used an *ASUS X541U* series laptop which has a built-in webcam. The experiment software was based on our *Snake Hands-free Edition* game, as described earlier (see section 2.6). We used *CameraMouse* for facial tracking. Two software tools were used: *GoStats*⁴ for statistical analyses, and the *Processing*⁵ tool to generate trace files of the snake’s movement and corresponding cursor movement for the touchpad and *CameraMouse*.

4.3 Procedure

The experimenter explained the steps of the experiment to each participant and demonstrated with a few practice trials. The counterbalancing group for each participant was chosen at random. Each participant was allowed some practice trials which were not part of the data analysis. Participants were tested over three sessions, one for each input method (see Figure 4 parts a, b, and c). A maximum of two sessions was allowed within a day as long as there was a break of at least two hours between the sessions. Two consecutive sessions were not separated by more than two days. Each trial was one minute in duration.

⁴ <https://www.yorku.ca/mack/GoStats/>

⁵ <https://processing.org/>

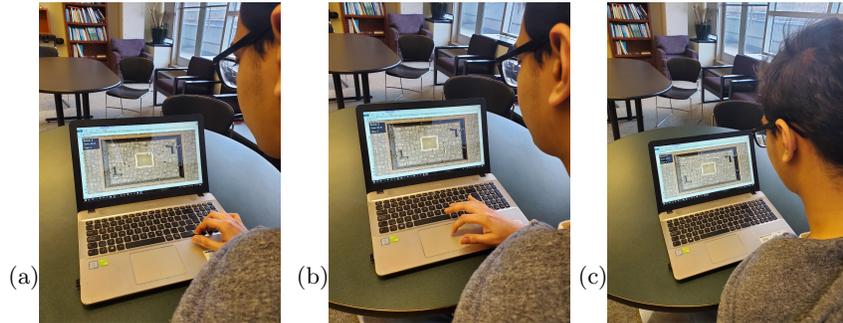


Fig. 4: A participant taking part in the Snake game with (a) keyboard, (b) touchpad, and (c) *CameraMouse*.

4.4 Design

The user study was a 3×8 within-subjects design. The independent variables and levels were as follows:

- Input method (keyboard, touchpad, *CameraMouse*)
- Block (1, 2, 3, 4, 5, 6, 7, 8)

There were three dependent variables: score, completion time (in seconds), and number of movements (count). The total number of trials was 1440 ($3 \times 8 \times 5 \times 12$). The three input methods were counterbalanced with four participants in each group to offset learning effects.

5 Results and Discussions

5.1 Score

Whenever the snake eats a fruit, the score increases by one. The mean scores were 1.94 (*CameraMouse*), 5.89 (keyboard), and 3.34 (touchpad). See Figure 5.

The effect of group on score was statistically significant ($F_{2,9} = 4.619, p < .05$), indicating an order effect despite counterbalancing. The effect of input method on score was statistically significant ($F_{2,18} = 83.569, p < .0001$). The effect of block on score was not statistically significant ($F_{7,63} = 1.225, p > .05$).

As seen in Figure 5, the mean score is much higher for the keyboard compared with the touchpad and *CameraMouse* methods. The keyboard method is the closest simulation of the original snake game on Nokia mobile phones, where users had to press physical buttons to play the original game. This similarity likely played a part in the keyboard being the best among input methods in terms of score. The low score with *CameraMouse* is likely due to the newness of the method while playing such a game.

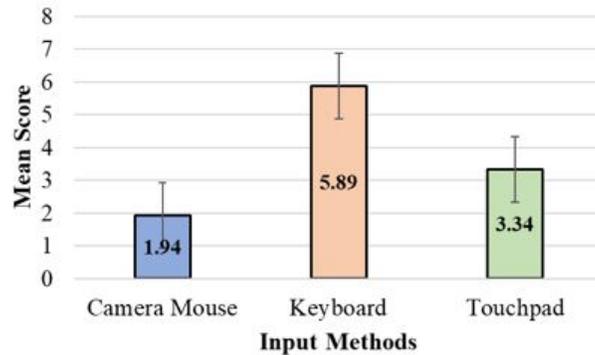


Fig. 5: Mean score by input methods. Error bars indicate $\pm 1 SE$.

5.2 Completion Time

Completion time signifies the time in seconds for which the snake was alive in each trial. In each trial, after 60 seconds, the snake would die automatically and the user would proceed to the next trial. The mean completion times were 35.38 seconds (*CameraMouse*), 40.79 (keyboard), and 39.32 (touchpad). See Figure 6.

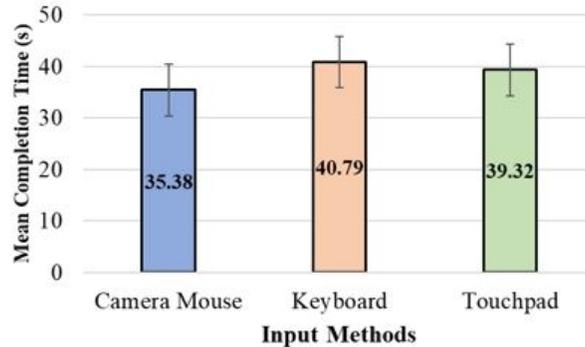


Fig. 6: Mean completion time (s) by input methods. Error bars indicate $\pm 1 SE$.

The effect of group on completion time was statistically significant ($F_{2,9} = 8.059, p < .01$), indicating again that counterbalancing did not adequately correct for order effects. However, the effect of input method on completion time was not statistically significant ($F_{2,18} = 2.185, p > .05$). The effect of block on completion time was also not statistically significant ($F_{7,63} = 0.865, ns$).

As stated earlier, the completion time for each trial signifies the in-game lifespan of the snake. The snake could die by hitting a poisonous object (black) or hitting a wall unless the allotted 60 seconds for a trial elapsed. The black objects appear on the game-screen randomly. Their appearance does not have any periodic or positional consistency and thus brings an element of surprise to

the player. This is true for all three input methods; hence, as seen in Figure 6, the mean completion times for each input method are not that different from each other.

5.3 Number of Movements

The number of movements were tallied as counts. Each time the snake changed direction, the number of movements increased by one. The mean number of movements was 21.57 (*CameraMouse*), 73.9 (keyboard), and 28.18 (touchpad). See Figure 7.

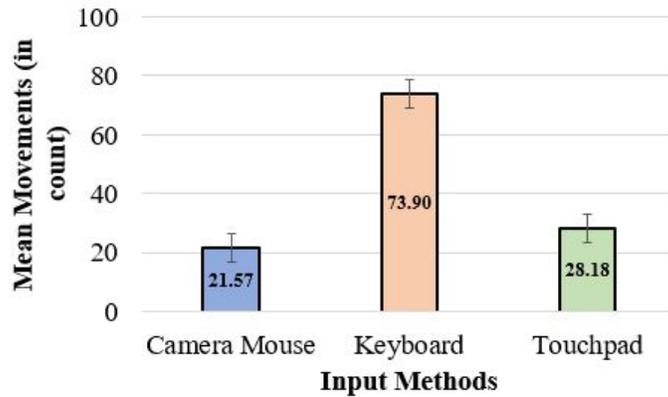


Fig. 7: Mean number of movements by input methods. Error bars indicate $\pm 1 SE$.

The effect of group on movements was not statistically significant ($F_{2,9} = 2.101, p > .05$), indicating that counterbalancing had the desired effect of offsetting order effects. The effect of input method on movements was statistically significant ($F_{2,18} = 65.157, p < .0001$). However, the effect of block on movements was not statistically significant ($F_{7,63} = 0.669, ns$).

The keyboard was the best performing input method for number of movements as well. The participants discovered that tapping diagonally positioned keys (i.e., left-up, up-right, left-down, down-right etc.) results in swift movement of the snake. They used this to move the snake faster in moving toward *fruits*. This resulted in a high number of movements with the keyboard. Participants struggled to bring this swift movement into control with the touchpad or *CameraMouse*, as covering the directional regions (see Figure 3) was obviously not as easy as changing directions with the keyboard.

5.4 Learning

While analyzing learning over the eight blocks of testing, we found an improvement in some cases, but not all cases.

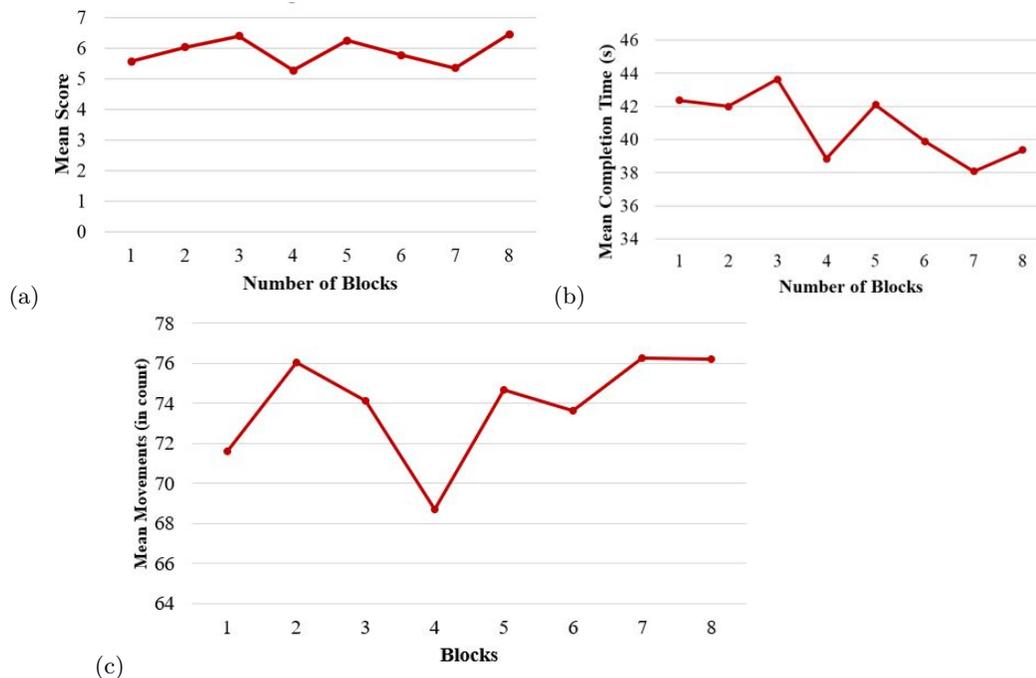


Fig. 8: Learning over eight blocks with the keyboard for (a) score, (b) completion time (s), and (c) movements (counts).

Beginning with the keyboard score (see Figure 8a), there was very little learning effect for score during the first three blocks and the final two blocks. Otherwise, there were dips in the score. The reason behind this is a combination of familiarity with the game and fatigue coming into effect after four blocks. It is significant to note that many participants took a break of five minutes after the first four blocks of each session. Hence, when they restarted they did better. But, when fatigue increased, their performance took a dip again. Figure 8b shows that completion time gradually decreased across the eight blocks with the keyboard. The fatigue effect is evident here. It also signifies, looking back at Figure 8a, that participants achieved a better score with less completion time during the final 2-3 blocks of the experiment while using the keyboard. For the number of movements with the keyboard, we see some learning taking place in Figure 8c during the final 2-3 blocks but there is a large dip during the blocks in the middle of the experiment.

For score using the touchpad, the learning effects line was mostly flat. But very little learning was observed for completion time (see Figure 9a) and number of movements (see Figure 9b) while using the touchpad.

With *CameraMouse*, there was no significant improvement with practice for score. But completion time gradually decreased over the first few blocks and it

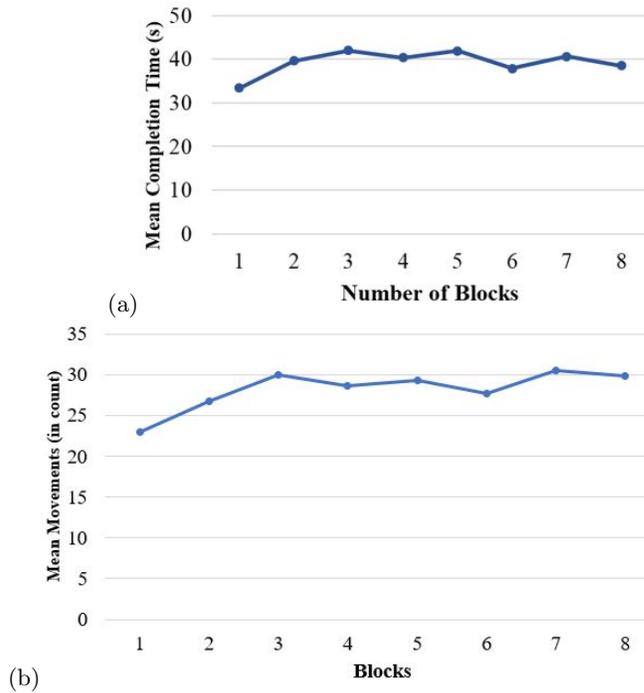


Fig. 9: Learning over eight blocks with touchpad for (a) completion time (s), and (b) movements (counts).

fluctuated across the rest of the blocks (see Figure 10a). Similar remarks can be made about movements while using the *CameraMouse* (see Figure 10b).

5.5 Traces of the Snake and Cursor

To further support our idea of dividing the game-screen into regions (see Figure 3 for touchpad and *CameraMouse* control), we generated trace files. For the simplicity of presentation, we show some trace file examples in Figure 11 and Figure 12. These trace files were generated from relatively short trials from one of the *CameraMouse* sessions. The red and blue dots in Figure 11 and Figure 12 decreased in size as time progressed along the trial. Note that the snake followed the cursor as the final point in the cursor's trace file was in the *UP* region and the final point of the snake's trace file shows that the snake was indeed going upwards. An overlapping trace file for both the cursor and the snake is depicted in Figure 12. Similar images for longer lasting trials are shown in Figure 13 and Figure 14.

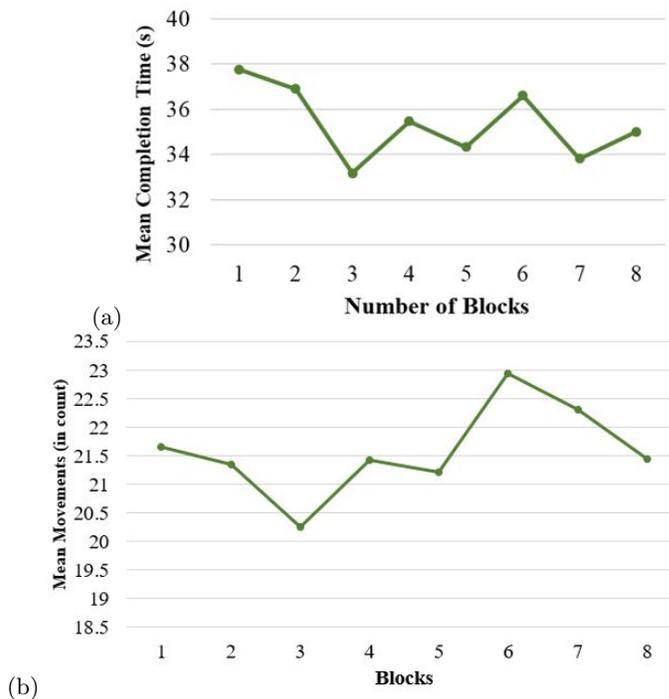


Fig. 10: Learning over eight blocks with *CameraMouse* for (a) completion time (s), and (b) movements (counts).

6 Participant Feedback

We collected participant feedback on a set of questions. Five of the 12 participants had no prior experience using *CameraMouse*. When asked about their preferred method of input, 11 of 12 participants chose the keyboard; only one participant chose the touchpad.

Participants also provided responses on two 5-point Likert scale questions. One question was on the participant’s level of fatigue with *CameraMouse* (1 = *very low*, 5 = *very high*). The mean response was 2.67, closest to the *moderate fatigue* score. The second question was on the participant’s rating of the hands-free phase of the experiment (1 = *very poor*, 5 = *very good*). The mean response was 3.25, just slightly above the *normal* score. Hence, it can be noted that the interaction with *CameraMouse* fared well.

We also asked the participants if they could name other games that they think can be played with the *CameraMouse*. Some interesting answers were given. Participants suggested games such as *Temple Rush*, *Subway Surfer*, and *Point Of View Driving* as possible candidates. However, 5 of the 12 participants thought *CameraMouse* could not be used in any other games. Participants were also asked about aspects of *CameraMouse* that they struggled with. The re-

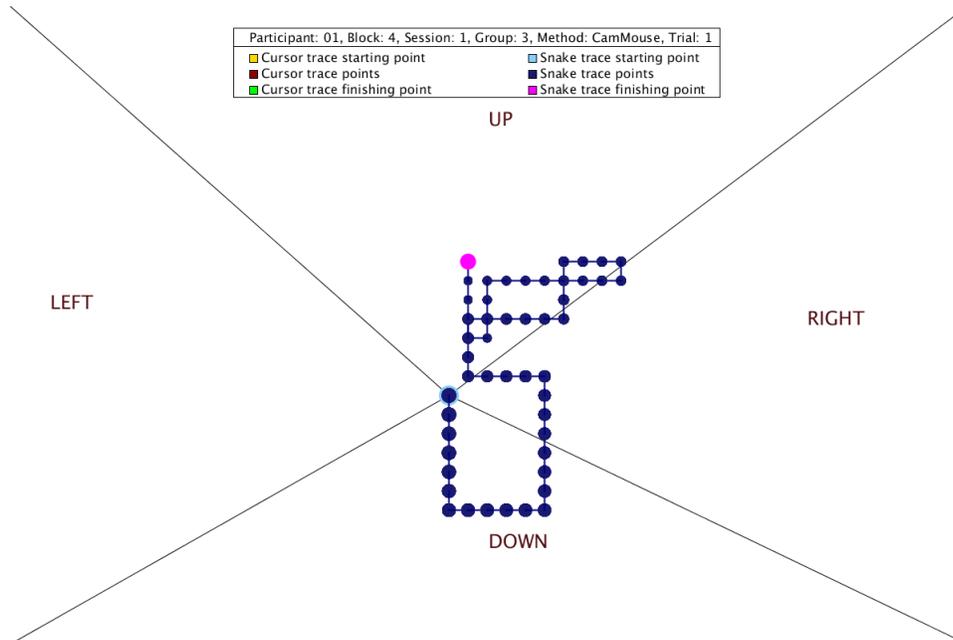


Fig. 11: Trace file for the snake's movement.

sponses were 'keeping track of the cursor', 'figuring out the required degree of head movement', 'horizontal movement', 'vertical movement', 'sensitivity of the cursor', etc. 'Keeping track of the cursor' and 'Figuring out the required degree of head movement' each received five responses. 'Sensitivity of the cursor' received nine responses.

7 Case Study

We conducted a case study along with our user study with a participant who has mild cerebral palsy. The participant is an undergraduate student. He is able to walk and has some use of his hands, but he does not have as much control in his hands as the 12 participants in the user study. Hence, he was only asked to do the hands-free session of the experiment (see Figure 15). He took part in all eight blocks of the hands-free session. The hardware, software, and procedure were the same as in the user study.

7.1 Design

This was a 1×8 single-subject design. We had two factors for the case study. Their names and levels are:

- Input method (*CameraMouse*)

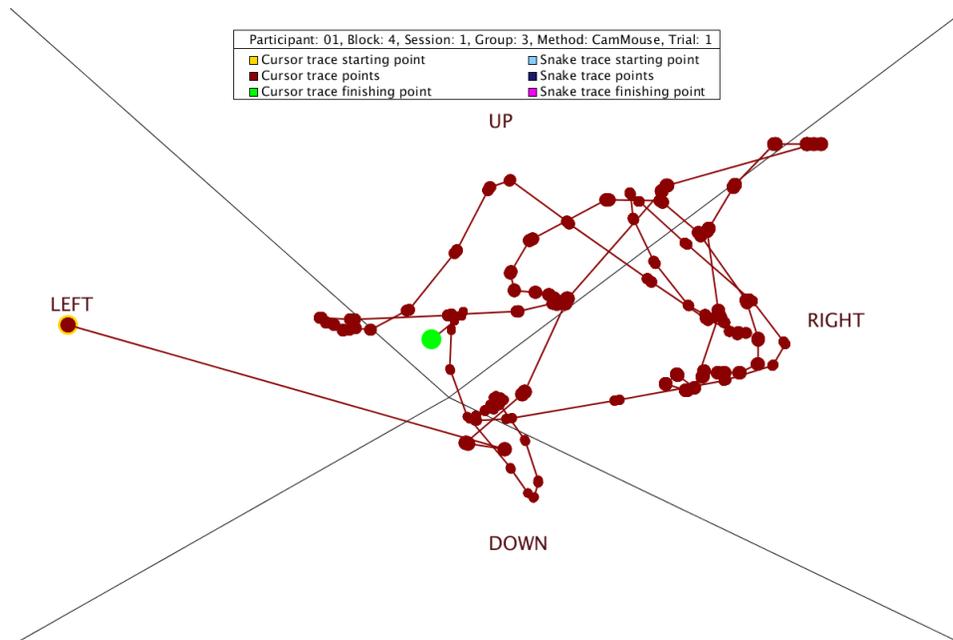


Fig. 12: Trace file for cursor movement.

- Blocks (1, 2, 3, 4, 5, 6, 7, 8)

Of course, input method is not a factor, per se, since only one level was used. It is reported as such to be consistent with the earlier user study.

The performances measures were once again score, completion time (s), and number of movements (counts).

7.2 Results and Discussion

We compared the performance measures of the hands-free user study with the performance measures in the case study. As seen in Figure 16a, the physically-challenged participant had a much lower mean score of 0.55 compared to the mean score for the hands-free session in the user study, which was 1.94. In terms of completion time, the case study had a mean completion time of 15.5 seconds whereas the user study's mean completion time for the hands free session was 21.57 seconds. See Figure 16b. The case study had a mean number of movements of 21.08, but the user study had a mean number of movements at 35.38. See Figure 16c. The physical condition of the participant of the case study obviously did not allow him to be as flexible as the non-disabled participants of the user study; thus, lower performance measures were observed during the case study.

The participant displayed very little learning in terms of completion time. See Figure 17b. But, his performance fluctuated a lot in terms of score. See Figure 17a. There was no significant learning effect observed for number of movements during the case study.

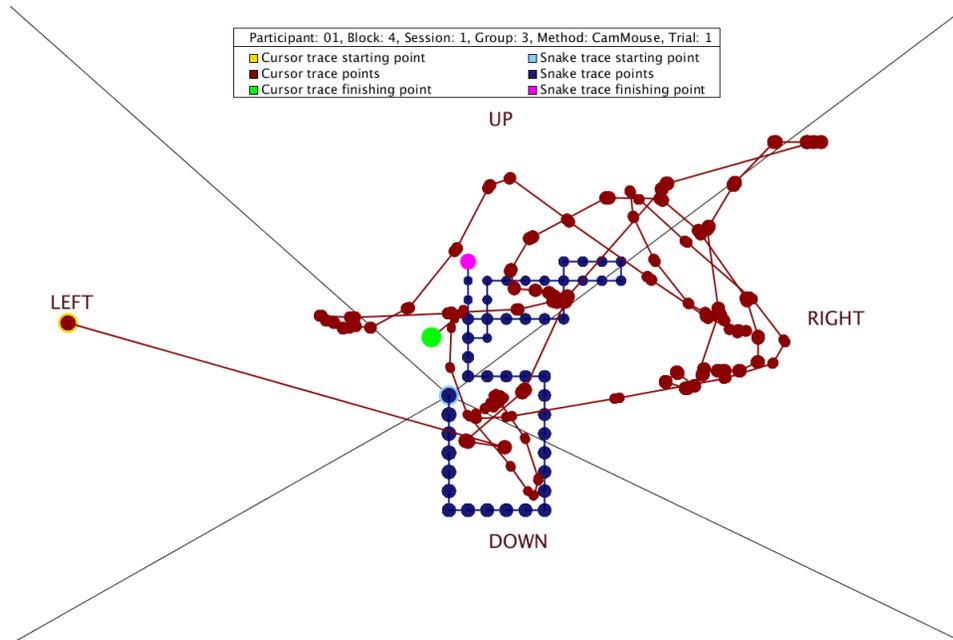


Fig. 13: Trace file for the snake's movement and cursor movement.

7.3 Case study Participant Feedback

The participant had no prior experience using the *CameraMouse* and did not mention any other game which can be played with *CameraMouse*. He affirmed he did not feel much fatigue during the session. When was asked to rate his *CameraMouse* experience on a five-point scale: (1 = *very poor*, 5 = *very good*), he gave *CameraMouse* a score of 2, which indicates *poor*. He also mentioned that the sensitivity of the cursor was a challenge. The participant also stated that as he moved farther away from the web-cam, he felt more comfortable while using *CameraMouse*.

7.4 Summary of the Case-study

We conducted this case study to identify how a physically-challenged participant would perform in our hands-free gaming experiment. We followed all standard experiment procedures for the case study and compared the performance measures in the case study with those in the user study. The performance measures in the case study were lower than those in the user-study, for all cases.

8 Conclusion

In this experiment, we compared *CameraMouse* with the touchpad and keyboard of a laptop computer to play a simple game: *Snake*. The keyboard was the best

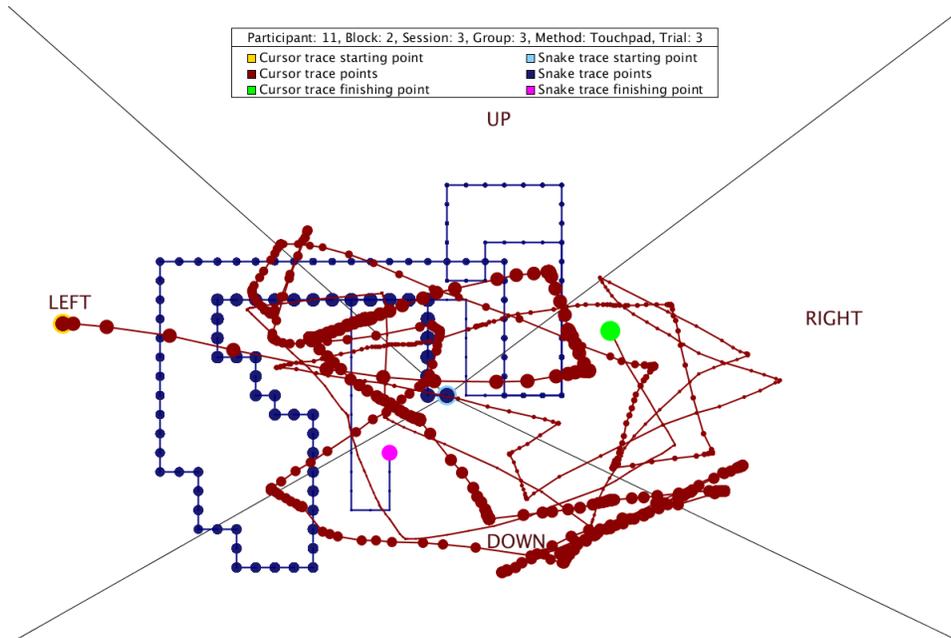


Fig. 14: Trace file for the snake’s movement and cursor movement (longer lasting trial).

performing method among the three tested. The mean score for the keyboard was 5.89; the mean score for the touchpad was 3.34; and *CameraMouse* had the lowest mean score at 1.94. The mean completion time for the keyboard was 40.79 seconds; the mean completion time for the touchpad method was 39.32 seconds; and *CameraMouse* had a mean completion time of 35.38 seconds. The mean number of movements for the keyboard was 73.90; the mean number of movements for the touchpad method was 28.18; and *CameraMouse* had a mean number of movements of 21.57. We also conducted a case study with a physically challenged participant. The performance measures showed lower values in the case study compared to the results gathered in the hands-free session.

But considering participant feedback, opportunities were noted to try *CameraMouse* in other games, such as *Temple Run*, *Subway Surfers*, etc.

Voice command, gesture control, and brain-computer research are also significant forms of hands-free interaction. In the future, the door remains open to explore these forms of hands-free interactions against facial tracking methods. It is also worthwhile to mention that accessible computing does not necessarily mean that a user has to be completely apart physically from the machine. Hence, partially hands-free methods can also be valuable for user-interaction in the accessible computing domain.

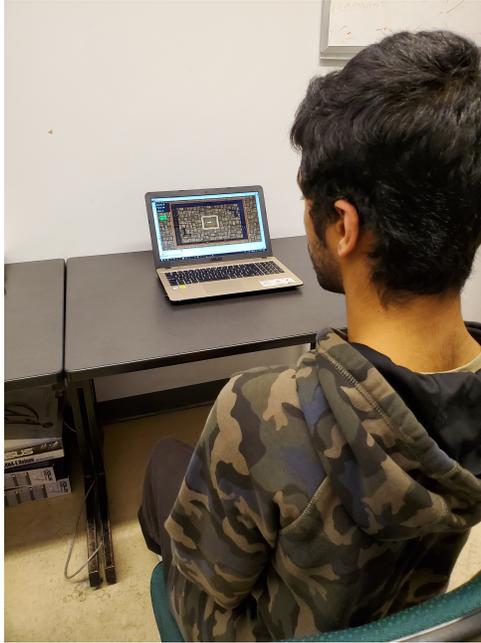


Fig. 15: Case study participant taking part in the experiment.

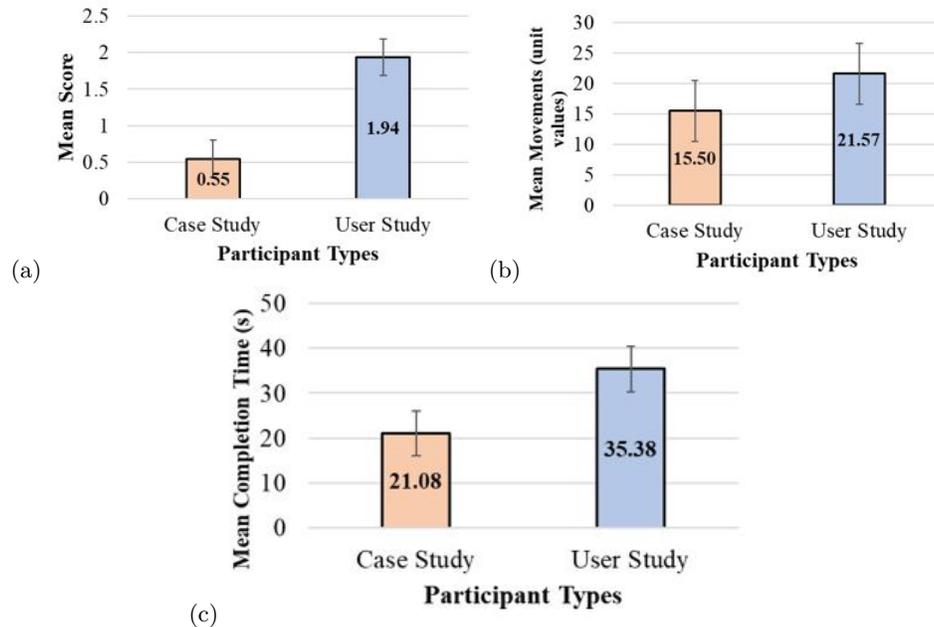


Fig. 16: Performance measure comparisons between the case study and the user study.

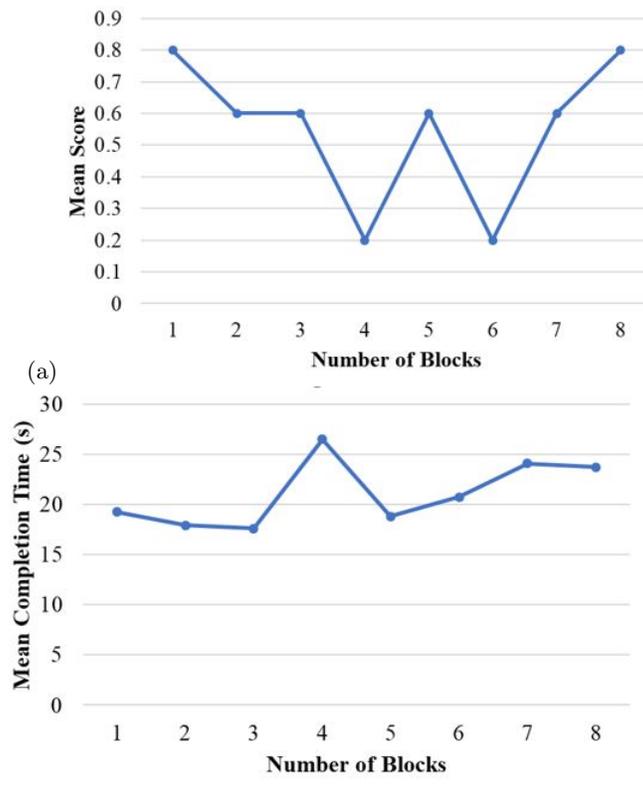


Fig. 17: Learning over eight blocks with *CameraMouse* for (a) score, and (b) completion time (s).

References

1. Betke, M., Gips, J., Fleming, P.: The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on neural systems and Rehabilitation Engineering* **10**(1), 1–10 (2002)
2. Cloud, R., Betke, M., Gips, J.: Experiments with a camera-based human-computer interface system. In: Proceedings of the 7th ERCIM Workshop "User Interfaces for All," UI4ALL. pp. 103–110. ERCIM (2002)
3. Cuaresma, J., MacKenzie, I.S.: A comparison between tilt-input and facial tracking as input methods for mobile games. In: Games Media Entertainment (GEM), 2014 IEEE. pp. 1–7. IEEE, New York, NY, USA (2014)
4. Gips, J., Betke, M., Fleming, P.: The camera mouse: Preliminary investigation of automated visual tracking for computer access. In: In Proc. Conf. on Rehabilitation Engineering and Assistive Technology Society of North America. pp. 98–100. RESNA (2000)
5. Grammenos, D., Savidis, A., Stephanidis, C.: Ua-chess: A universally accessible board game. In: Universal Access in HCI: Exploring New Interaction Environments-Proc. 11th Int. Conf. on Human-Computer Interaction (HCI International 2005). vol. 7 (2005)
6. Hassan, M., Magee, J., MacKenzie, I.S.: A fitts law evaluation of hands-free and hands-on input on a laptop computer. In: International Conference on Human-Computer Interaction. pp. 234–249. Springer (2019)
7. Kent, S.L.: The Ultimate History of Video Games: Volume Two: from Pong to Pokemon and beyond... the story behind the craze that touched our lives and changed the world. Three Rivers Press (2010)
8. MacKenzie, I.S.: Evaluating eye tracking systems for computer input. In: Gaze interaction and applications of eye tracking: Advances in assistive technologies, pp. 205–225. IGI Global, Hershey, PA, USA (2012)
9. MacKenzie, I.S.: Human-computer interaction: An empirical research perspective. Morgan Kaufmann, Waltham, MA, USA (2012)
10. Magee, J., Felzer, T., MacKenzie, I.S.: Camera mouse+ clickeraid: Dwell vs. single-muscle click actuation in mouse-replacement interfaces. In: International Conference on Universal Access in Human-Computer Interaction. pp. 74–84. Springer (2015)
11. Magee, J.J., Scott, M.R., Waber, B.N., Betke, M.: Eyekeys: A real-time vision interface based on gaze detection from a low-grade video camera. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. pp. 159–159. IEEE, New York, NY, USA (2004)
12. Roig-Maimó, M.F., Manresa-Yee, C., Varona, J., MacKenzie, I.S.: Evaluation of a mobile head-tracker interface for accessibility. In: Proceedings of the 15th International Conference on Computers Helping People With Special Needs - ICCHP 2016 (LNCS 9759). pp. 449–456. Springer, Berlin (2016)