Full Body Gestures Enhancing a Game Book for Interactive Story Telling

Felix Kistler, Dominik Sollfrank, Nikolaus Bee, and Elisabeth André

Human Centered Multimedia, Augsburg University,
Universitätsstr. 6a, D-86159 Augsburg, Germany
{kistler,bee,andre}@hcm-lab.de
http:/www.hcm-lab.de

Abstract. Game Books can offer a well-written, but non-linear story, as readers always have to decide, how to continue after reading a text passage. It seems very logical to adopt such a book to investigate interaction paradigms for an interactive storytelling scenario. Nevertheless, it is not easy to keep the player motivated during a long-winded narrated story until the next point of intervention is reached. In this paper we tested different methods of implementing the decision process in such a scenario using speech input and tested it with 26 participants during a two player scenario. This revealed that with an omitted on-screen prompt the application was less easy to use, but caused considerably more user interaction. We further added additional interactivity with so-called Quick Time Events (QTEs). In these events, the player has a limited amount of time to perform a specific action after a corresponding prompt appeares on screen. Different versions of QTEs were implemented using Full Body Tracking with Microsoft Kinect, and were tested with another 18 participants during a two player scenario. We found that Full Body Gestures were easier to perform and, in general, preferred to controlling a cursor with one hand and hitting buttons with it.

1 Introduction

With the release of the Kinect sensor[1], Microsoft has made depth sensors easily available for every home. This makes it possible to take the next step in human-computer interaction by using Full Body Gestures and Movements. However, it is a challenging task to apply this new way of interaction and to find possibilities for offering a good user experience and usability along with it.

In this paper we use the Kinect sensor to provide different types of interaction for our interactive storytelling scenario adopting parts of the story of the game book "Sugarcane Island" by Edward Packard [10].

[1] http://www.xbox.com/kinect

Various approaches for providing innovative interaction modalities in interactive storytelling have been investigated in the past. Some of these are described by way of illustration below, though none of them uses Full Body Interaction. Cavazza and colleagues presented an interactive storytelling system[3] where the user can influence the story by speaking to and giving advice to a virtual character, such as "don't be rude". They therefore have speech recognition that uses template matching.

Façade is a well-known interactive drama by Michael Mateas and Andrew Stern [8]. The user can interact by typing any kind of text message on the keyboard. Façade was further evolved further to an augmented reality version, additionally implementing speech recognition with a Wizard-of-Oz setting to provide a more natural interaction [4].

Project Geist is an augmented reality storytelling system [2], where users can perform gestures on a magic wand and speaking basic sentences that are recognized by the system.

To our knowledge, there have as yet been no scientific studies about the application of QTEs. However, good examples of games where there is extensive use of QTEs are *Fahrenheit (Indigo Prophecy)* and *Heavy Rain* from the games developer Quantic Dream [12]. To implement the QTEs, they use mouse gestures and keyboard input on the PC. On the game console, they use button and analog stick input, together with gestures performed on a game pad with an accelerometer. This means that the user is required to perform abstract actions or gestures on a controller in opposite to our implementation of QTEs.

The Kinect games currently available on the Xbox console mainly comprise sport and fitness games (e.g. *Kinect Sports*), racing games (e.g. *Kinect Joy Ride*), and party and puzzle games (e.g. *Game Party in Motion*). They all have some sort of motion or gesture interaction, but none of them concentrates on a story. Indeed, nearly all of them do not have any story at all.

Game books offer a well-written and non-linear story well-suited to an interactive storytelling scenario. Some early examples of the game book genre are described below:
In 1941 Jorge Luis Borges published a short story called *An Examination of the Work of Herbert Quain*[1]. It discusses several works by the fictional author Herbert Quain, including the non-linear novel *April March.* This novel is made up of thirteen chapters, covering nine different story lines. This is achieved through the fact that the first chapter can lead into any one of three subsequent chapters, and each of those in turn has three possible subsequent chapters. It could be considered the first example of the basic idea behind game books.

Another publication along similar lines were the educational books published under the series *TutorText* between 1958 and 1972 [14]. The idea was to enable students to learn without a teacher being present. Multiple-choice questions led to different pages depending on the chosen answer. While an incorrect answer leads to a page explaining what is wrong with the answer, a correct answer leads to a page with more information and the next questions.

2 Application and Methods

For the story line of our application, we adopted parts of the game book "Sugarcane Island" by Edward Packard [10], briefly described in section 2.1. However, unlike the book, our application is designed for two users listening to a virtual narrator and interacting at specific points to influence the story.

We have two different interaction types enabling users to do this. The first is close to the decisions that have to be made while reading the book, and is explained in section 2.2. The second adds QTEs with Full Body Gestures, and is illustrated in section 2.3. For each type of interaction, we have realized two different modes that outline different ways of implementing them.

Our application runs on the *Horde3D GameEngine* [6]. Additionally, we are using *SceneMaker 3* [9] to model and execute the story as a hierarchical finite state machine extended with multimodal scene scripts that consist of the text to be spoken including additional commands like animations or sounds.

2.1 Sugarcane Island

"Sugarcane Island" is the first book in the popular *Choose Your Own Adventure* series. It starts with a shipwreck on an expedition. Waking up on the beach of an unknown island, the reader needs to find a way to survive.

After each text section of the book, the reader has to decide how to proceed. The given choices refer to different pages in the book to read next. For example, the reader has to make a decision on page 17 of the book[2] in the following way:

Page 17: You wake up in a thatched hut. [...] You take a peek outside and observe ferocious looking natives doing a tribal dance around a fire.
You decide to flee. Go to *Page 27*.
You stay. Go to *Page 28*.
Page 27: You start up and sprint into the woods. [...]
Page 28: A little while later, some natives appear in your hut. [...]

2.2 Implementation of Decision Modes

For the decision modes, we chose to implement a part of "Sugarcane Island" that covers about one third of the text. Figure 1 visualizes that part as a decision graph. Each node of the graph refers to one page of the book. The decisions in the story are represented by an edge leading to a subsequent node. Blue edges mark modifications to the original story line. Blue edges without a target node stand for omitted decisions. The black edges constitute a tree, and therefore a story without any multiply used nodes like the one described by Borges in his short story [1]. The red edges lead to different parts of the tree, and sometimes add circle paths to it. There are a total of 12 different endings to the story as implemented. These endings can be clustered into three different types: in the

[2] Re-translated to English from the German version [10].

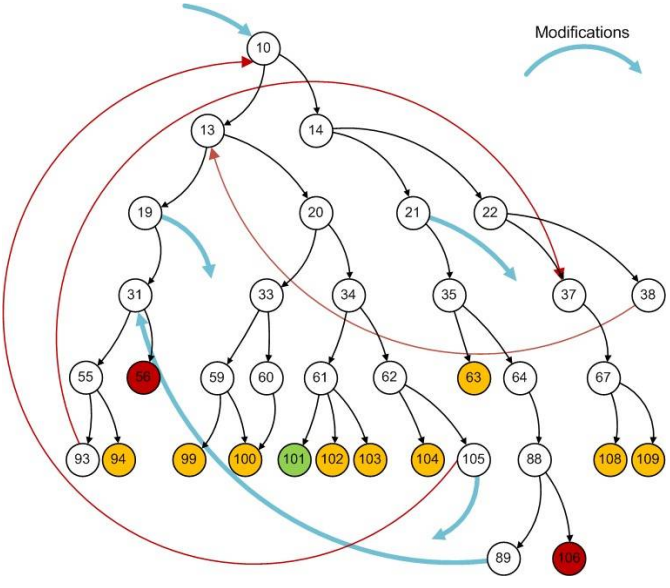

Fig. 1. Decision Graph of the implemented part of the book

only *good ending* (green node), the users return home having found a treasure. In the two *bad endings* (red nodes) the protagonist dies. The nine *neutral endings* (yellow nodes) stand for everything in between, like staying on the island or returning without a treasure.

For each node, the corresponding text section of the book is narrated by a virtual character (see Figure 2). A background image is displayed behind the character and a background sound is played, matched to the content of that text section. In addition, the character changes his facial expression depending on the content (e.g. he has a fearful expression if something unexpected happens), and at specific parts of the text short sounds are played (e.g. in one scene wild dogs appear, so the sound of barking dogs is played in parallel with this). When reaching the end of a node (a decision in the book), the narrator asks for input from the users and continues with the next node depending on the decision taken.

We have implemented two different modes for decision-making:

In the first mode (*indicated mode*), users have to speak out the decision they choose from an on screen prompt, as shown in Figure 2.
In the second mode (*freestyle mode*), we simply omit the on-screen prompt, so users have to figure out the possible answers for themselves. That may sound relatively simple, but it already gives users a sense of having more scope to take an influence on the story. In addition, users are allowed to ask for possible decisions in that mode when they cannot find one for themselves.

In the corresponding study, no actual speech recognition is used to recognize the spoken words of the users; instead, this is realized using Wizard-of-Oz.

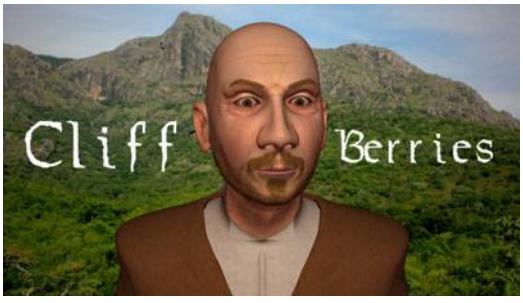

Fig. 2. The screen with decision overlays

2.3 Implementation of Quick Time Modes

To make the interaction more interesting and less predictable, we implemented another possibility for altering the story line, via the *Quick Time Events* (QTEs) that are frequently used in current video games. Whenever a QTE occurs in a video game, a symbol representing a specific action on the control device appears on screen. The user then has a limited amount of time to perform that action in order to complete the QTE successfully. Most times, successful performance of the QTE results in a particular action by the player avatar, while unsuccessful performance causes the player avatar to fail in this action. The utilization of QTEs ranges from enriching cutscenes with interactivity to using QTEs as the main gameplay mechanic.

We adopted this idea to provide another instrument for adding interactivity to the story and for bridging long text passages.

QTEs in "Sugarcane Island". Some passages of the book already contain situations that are well-suited for the application of QTEs. One example in the original text (p. 13)[3] reads as follows:

You start to climb the steep hill. It is highly exhausting and one time you loose your grip and almost fall down a rock face. But finally you arrive at the top.

The modified text part looks like this (modifications marked with bold font):

You start to climb the steep hill. It is highly exhausting and one time you ***almost*** *loose your grip.*

The QTE then starts, and when it is solved the following message is narrated:

You manage to hold on just in time and *finally you arrive at the top..*

Otherwise, the text is:

You fall down a rock face but you are lucky that you did not get hurt too badly.

If the QTE is solved, the story continues like the original version (p. 20), but if not it jumps to a different, but appropriate, node in the story graph (p. 14).

[3] Re-translated to English from the German version [10].

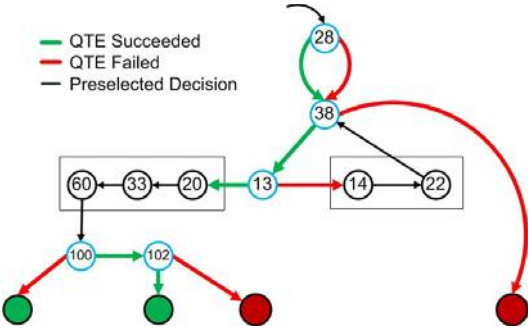

Fig. 3. Decision Graph for the QTE implementation

Figure 3 visualizes the story line implemented for testing the QTEs. As we want to concentrate on the execution of the QTEs and their user acceptance, all normal decisions are omitted and the only possibility for influencing the story are the QTEs. For the same reason, we also shortened the adopted texts in the story, trying to keep the story line consistent and to retain the cues leading to the inserted QTEs at all times.

As soon as a QTE starts in our application, a countdown appears centrally in the upper part of the screen with a symbol shown for each user representing the action requested (see Figure 4). As soon as one user solves a QTE, a mark is shown on top of the symbol, providing immediate feedback. The full QTE is solved if both users successfully complete their action before the countdown reaches zero.

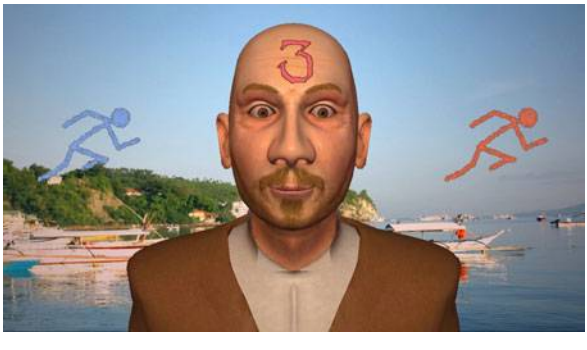

Fig. 4. Screenshot for the QTE "Run"

Implementation with Microsoft Kinect We employ the Microsoft Kinect sensor in order to provide Full Body Interaction. Kinect is an additional peripheral for the Microsoft Xbox 360 that provides a depth image in a 640x480

resolution at 30 frames per second. User tracking applied to the depth image allows users to interact with their whole body via gestures and movements.

The most important parts of the Kinect are an infrared (IR) camera and an IR laser projector. They are used to provide a depth image that is created according to the structured light principle [11].

To work with the Kinect on a PC, we are using the "OpenNI" (Open Natural Interaction) framework, "NITE" (Natural InTEraction)[4] middleware and a mod of the corresponding PrimeSensor driver[5] that supports the Kinect. Besides access to the raw data from the Kinect, this also provides user body tracking with resulting 3D positions of specific joints from the tracked users.

We employ two different modes to carry out of QTEs, both using the Kinect: in the first mode (*button mode*), each user has to press a randomly-positioned and -sized button on screen, using a cursor controlled by moving the hand. In the other mode (*gesture mode*), users need to perform gestures that are indicated on screen via one of the symbols shown in Figure 5.

Fig. 5. Quick Time Symbols

The requested user actions for each symbol are (from left to right):
Balance: Hold hands out at shoulder-height; *Kick*: Perform a kick with one leg; *Catch*: Put the hands together in front of the body; *Climb*: Move hands up and down in front of the head, as if climbing; *Left and right Hand*: Raise the left or right hand; *Run*: Move the feet up and down like running, but without moving.

Figure 6 shows two users in front of a screen, performing the QTE gestures "Kick" (left user) and "Catch" (right user). Note the Kinect placed centrally below the screen. The left user has already succeeded in the QTE "Kick", so a green tick has appeared over the corresponding symbol.

To implement the *button mode*, we take the tracked vector from one hand to the corresponding shoulder for each user. By moving this hand in front of their body on a plane parallel to the screen, both users control their own cursor displayed on-screen, similar to normal mouse input. By hovering a button for more than 1.5 seconds with the cursor, the button is activated. Alternatively, the users can perform a short and fast movement directed to the screen (push gesture) to activate the button directly.

As soon as the QTE starts, a button containing text for the requested action is shown for each user. The buttons are positioned randomly: on the right half of the screen for the right user, and on the left half for the left user. In addition, they

[4] `http://www.openni.org`
[5] `https://github.com/avin2/SensorKinect`

Fig. 6. QTE gestures performed by two users

randomly have a slightly different size for each QTE. Once a button is activated, it disappears and a tick appears to inform the user about the successful action.

The implementation for recognizing the gestures shown in Figure 5 is somewhat more complicated. However, some of them can be recognized directly from the joint positions, as they only require static postures from the user. For example, "left hand" is recognized in our system by getting the positions of the left hand and shoulder from the tracker and simply testing whether the left hand is currently above the left shoulder (y-coordinate of the hand greater than y-coordinate of the shoulder). The same method of looking at how specific joints are positioned in relation to each other is used for recognizing "right hand", "balance", "kick", and "catch". Of course, "balance", "kick", and "catch" would normally include some kind of movement, but for the recognition purposes it is sufficient to wait until the most meaningful part of that movement is performed. For example, "kick" only needs one foot being high enough above the ground, as the speed or direction of the movement are not important in our case.

To implement the other gestures, we recognize combinations of postures with specific time constraints. Figure 7 shows the recognition of the action "run" as an automaton. The states in the automaton define multiple postures that have to be fulfilled in parallel. The numbers on the black edges define the order in which the states have to be performed, in order to eventually complete the posture combination in the end node. The green edges represent the timeout for each state: if there has been no transition for 500 ms, the automaton reverts to the start node and the recognizer waits again for the first-state postures. To successfully complete "run", the user is consequentely required to move the left and right knee up and down alternately, with this being measured by the distance between the knee and hip joint. Those two postures have to be performed at least

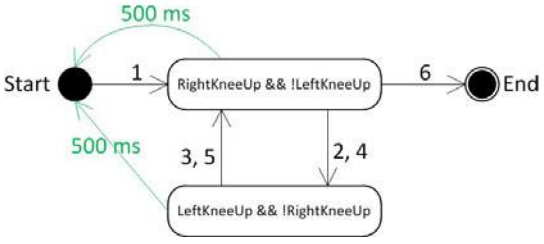

Fig. 7. Recognition automaton for running

two times and with not more than 500 ms in between. The posture combination "climb" can be realized in the same way, but in that case the user has to move his or her hands up and down in front of the head.
The implementation of the mentioned gesture and posture recognizers led to the development of the Full Body Interaction Framework [5].

3 Study and Results

We conducted two user experiments. The first one was to investigate the differences between the user experience in the *freestyle* and the *indicated mode* in a two-party storytelling setting. We further hypothesized that the *freestyle mode* would lead to more user interaction (H1). The second experiment aimed to investigate user acceptance of the QTEs in our virtual storytelling setting using Full Body Interaction with Microsoft Kinect. We assumed that the *gesture mode* would be preferred to the *button mode* (H2).

3.1 Study on Decision Modes

For the first experiment, we chose a Wizard-of-Oz design to ensure that all user decisions were recognized correctly. The participants had to sit at a table and a 50 inch plasma display was located in about two meters in front of them. A camera was installed on top of the display to capture the whole user interaction. The questionnaire for both experiments was derived from the IRIS (Integrating Research in Interactive Storytelling) Evaluation Measurement Toolkit [13]. Each statement was given on a nine-point Likert scale ranging from "strongly disagree" (-4) through "neutral" (0) to "strongly agree" (4).

An application run consists of a short introduction and the main story part. The introduction was meant to understand how to interact with the system. 26 participants were involved in the first study. We applied a "within subjects" design, and thus each group had to participate in both conditions (i.e. *freestyle* and *indicated mode*). To prevent positioning effects, we counterbalanced the order in which the conditions were encountered. In addition to the questionnaires, we collected nearly four hours of video material to objectively measure the number and duration of the user actions during the interaction with the application.

The average age of the participants was 27.2 years, and we applied two-tailed paired t-tests to show the validity of the results.

Results. The results of the questionnaire show that the users found the *indicated mode* (M: 2.9, SD: 1.2) significantly easier to use than the *freestyle mode* (M: 1.2, SD: 1.8), where they had to decide on their own from the story what to do ($t(25) = 4.7, p < 0.001, r = 0.49$). Further, they would imagine that most people would learn the *indicated mode* (M: 3.4, SD: 0.8) significantly quicker than the *freestyle mode* (M: 1.8, SD: 1.8), where they had to figure out the decisions by themselves ($t(25) = 5.1, p < 0.001, r = 0.50$). The participants also found the *indicated mode* (M: -3.2 , SD: 1.0) significantly less inconvenient to use than the *freestyle mode* (M: -1.6, SD: 1.9; $t(25) = 4.5, p < 0.001, r = 0.47$). All the other items regarding satisfaction, immediate impact of actions, influence of the decisions, and curiosity were not significantly affected by the two modes.

The whole interaction was captured by a camera on top of the display. The videos generated were subsequently annotated using the freely-available video annotation tool ANVIL [7]. In addition, the whole system actions and calls were logged in an ANVIL readable XML format. The focus of the analysis was the participants' gaze and speaking behavior during the two modes. Furthermore, we measured the overall duration of each run. The results are described below:
Gaze: In *freestyle mode* (M: 31.5 s, SD: 31.1), the participants looked significantly longer at each other than in *indicated mode* (M: 9.8 s, SD: 8.4; $t(25) = 4.3, p < 0.001, r = 0.43$). Moreover, in *freestyle mode* (M: 4.4, SD: 3.0) the participants looked significantly more often at each other than in *indicated mode* (M: 9.9, SD: 7.3; $t(25) = 4.6, p < 0.001, r = 0.44$).
Speech: In *freestyle mode* (M: 34.0 s, SD: 29.0), the participants spoke significantly longer with each other than in *indicated mode* (M: 12.2 s, SD: 9.1; $t(25) = 4.8, p < 0.001, r = 0.45$). Furthermore, in *freestyle mode* (M: 12.8, SD: 7.8) the participants spoke significantly more often to each other than in *indicated mode* (M: 6.3, SD: 3.9; $t(25) = 5.2, p < 0.001, r = 0.47$).
Duration: The longest run of a group in *indicated mode* was *5:50 min*, whereas it was *7:50 min* in *freestyle mode*. The average duration was *3:40 min* in *indicated mode* and *5:21 min* in *freestyle mode*. The 1:41 minutes longer average duration is caused by longer durations of the interactions. This indicates a deeper interaction between the participants in *freestyle mode*.

3.2 Study on Quick Time Event Modes

In the second experiment, our intention was both to discover if the *button* or *gesture mode* is preferred, and to establish the quality of our system's recognition.

18 participants were involved in the study; none of these were involved in the study of decision modes. The participants were arranged into groups of two. The two participants in a group had to stand in front of a 50 inch plasma display at a distance of between 1.5 and 3 meters. The Microsoft Kinect was installed slightly below the display.

We applied the "within subjects" design, and therefore each group had to participate in one application run of both conditions (i.e. button and gesture mode). To prevent positioning effects, we counterbalanced the order in which the conditions were encountered. An application run consists of two parts, a short

introduction and the main story part. The short introduction was intended only to familiarize the participants with the interaction with the system. After each run, the participants had to fill out a questionnaire that was again derived from the IRIS Evaluation Measurement Toolkit [13]. Each statement was given on a nine-point Likert scale ranging from "strongly disagree" (-4) through "neutral" (0) to "strongly agree" (4).

The average age of the participants was 24.7 years, and we applied two-tailed paired t-tests to show the validity of the results.

Results. The results of the questionnaire show that the participants found the gesture-based QTEs (M: 2.9, SD: 0.9) significantly easier to use than the button-based events (M: 2.0, SD: 1.9), where they had to simply point at a specific button label ($t(17) = 2.1, p < 0.05, r = 0.29$). The participants also would imagine that most people are able to learn the system with the gesture-based QTEs (M: 3.3, SD: 0.8) significantly quicker than the one without (M: 2.4, SD: 1.5; $t(17) = 2.2, p < 0.05, r = 0.35$). The *gesture mode* (M: -3.1, SD: 1.0) was considered more comfortable to use compared to the *button mode* (M: -1.3, SD: 2.1; $t(17) = 3.5, p < 0.01, r = 0.48$). The participants were significantly more satisfied with the *gesture mode* (M: 2.3, SD: 1.0) than with the *button mode* (M: 1.6, SD: 1.4; $t(17) = 2.4, p < 0.05, r = 0.28$). The *gesture mode* (M: 0.0, SD: 2.5) was also considered as significantly less inconvenient compared to the *button mode* (M: -2.3, SD: 1.7; $t(17) = 4.0, p < 0.001, r = 0.47$). Interaction in *gesture mode* (M: 3.1, SD: 1.0) was experienced as significantly more fun than in *button mode* (M: 1.0, SD: 2.3; $t(17) = 3.8, p < 0.01, r = 0.51$), and lastly the participants stared at the screen with significantly higher expectations in *gesture mode* (M: 1.4, SD: 1.6) compared to the *button mode* (M: 0.8, SD: 1.5; $t(17) = 2.2, p < 0.05, r = 0.19$).

The recognition within the button and gesture mode worked very well. The participants succeeded in 93% of all actions within the button-based QTEs (i.e. 67 out of 72). For the gesture-based QTEs the participants were even more successful, with 97% of all possible actions (i.e. 65 out of 67).

4 Conclusion

We showed that a game book like "Sugarcane Island" can be easily adopted to create an interactive storytelling scenario. In addition, we made some interesting observations regarding implementation of our interaction types: in relation to the usability of the decision implementation we found that *indicated mode* was preferred to *freestyle mode*. One may conclude that a system that is easier to use is automatically rated better on other aspects as well, but in our questionnaire the two modes (freestyle and indicated) are rated similarly in all other statements. However, if we also take the results from the video recording into account, it becomes apparent that the *freestyle mode* generated considerably more user interaction, proving our first hypothesis (H1).

One result regarding our QTE implementation is that *gesture mode* was preferred to *button mode* in terms of usability. This implies that natural gestures for

QTEs not only support better usability, but also greater comfort. Moreover, using natural gestures also made more fun. The effect sizes (r) of the results further indicate strong findings, so our second hypothesis (H2) can also be considered proven. But not only perceptions of usability, comfort and fun were improved; we also showed participants were better at solving QTEs using natural gestures than the cursor-button interaction. This indicates, that it may be worthwhile designing new and more natural ways of interaction for a Full Body Tracking system, instead of adapting the conventional point-and-click paradigm.

Acknowledgement. The work reported in this paper is partially supported by the European Commission under grant agreement IRIS (FP7-ICT-231824) and eCUTE (FP7-ICT-257666). We would like to thank Gregor Mehlmann for his help with SceneMaker.

References

1. Borges, J.L.: An Examination of the Work of Herbert Quain. In: Kerrigan, A. (ed.) Ficciones, pp. 73–78. Grove Press, 841 Broadway, New York (1962)
2. Braun, N.: Storytelling in Collaborative Augmented Reality Environments. In: Proceedings of the 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG (2003)
3. Cavazza, M., Charles, F., Mead, S.J.: Interacting with Virtual Characters in Interactive Storytelling. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems. ACM, New York (2002)
4. Dow, S., Mehta, M., Harmon, E., MacIntyre, B., Mateas, M.: Presence and Engagement in an Interactive Drama. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1475–1484. ACM, New York (2007)
5. Full Body Interaction Framework, `http://www.hcm-lab.de/fubi.html`
6. Horde3D GameEngine, `http://www.hcm-lab.de/projects/GameEngine`
7. Kipp, M.: ANVIL - A Generic Annotation Tool for Multimodal Dialogue. In: 7th European Conference on Speech Communication and Technology (Eurospeech), pp. 1367–1370. ISCA (2001)
8. Mateas, M., Stern, A.: Façade: An Experiment in Building a Fully-Realized Interactive Drama. In: Game Developers Conference: Game Design Track (2003)
9. Mehlmann, G., Gebhard, P., Endrass, B., André, E.: SceneMaker: Visual Authoring of Dialogue Processes. In: 7th Workshop on Knowledge and Reasoning in Practical Dialogue Systems (2011)
10. Packard, E.: Die Insel der 1000 Gefahren. Ravensburger Buchverlag, Ravensburg (2007)
11. PrimeSense Ltd: US Patent Application Publication No. US2010/0118123 (2010), `http://www.freepatentsonline.com/20100118123.pdf`
12. Quantic Dream, `http://www.quanticdream.com`
13. The IRIS project, `http://iris.scm.tees.ac.uk`
14. TutorText: Doubleday Series, `http://www.gamebooks.org/show_series.php?id=457`