Using Wearable Devices to Participate in 3D Interactive Storytelling

Tsai-Yen Li(✉) and Wen-Hsuan Wang

National Chengchi University, Taipei 116, Taiwan
li@nccu.edu.tw

Abstract. There exist various ways for a user to participate in a story through interactive narratives. Most previous work uses traditional user interfaces for interactions with limited modifications to story elements. It is a challenge to allow a user to participate in an interactive story through the first-person view in a 3D virtual environment. In this work, we propose to use wearable motion capture (mocap) devices to enable a user to play as a character in a 3D virtual scene and interact with the environment and other virtual characters in real time. The interactions will affect how the story develops as well as the result. In such an interactive storytelling system, we have designed methods to interpret user actions as well as to generate parameterizable animations according to the interactions. We have conducted a user study to evaluate our system by comparing a traditional controller with a wearable device. The experimental results reveal that the interaction methods we have designed are more intuitive and easier to use, compared to the controller. In addition, the users are willing to try to play with the system multiple times, which confirms the replay value of our interactive storytelling system.

Keywords: Interactive storytelling · Wearable device · 3D virtual environment · Motion capture · Character animation

1 Introduction

As computer technologies advance, there exist more and more opportunities for the audience to change their roles as observers and actively participate in a story in a 3D virtual environment. Through interactions with the virtual environment as well as other characters in a story, one can change the plot of the story or the contents of story elements on the fly. The audience will be able to break the fourth wall of narratives in various ways. In a typical 3D digital interactive storytelling setting, one can use traditional user interface devices such as a keyboard and a mouse to interact with the narrative system by selecting a story branch from a menu or navigating to a Non-Character Player (NPC) to retrieve information and make a conversation. The interaction may not be intuitive and the responses from the system may also be limited. On the other hand, the development of new sensing and VR technologies is opening up new directions for novel applications. For example, somatosensory devices such as Leap Motion, Kinect, and other wearable motion capture devices are becoming more affordable, people are starting to use these devices to develop novel applications for interactive storytelling.

A. Mitchell and M. Vosmeer (Eds.): ICIDS 2021, LNCS 13138, pp. 80–93, 2021.
https://doi.org/10.1007/978-3-030-92300-6_8

As the technologies develop, motion capture (mocap) has become a popular and affordable way to generate character animations in 3D applications. If we use the mocap devices to control the motions of a 3D avatar for the user, how to interpret these motions and generate responsive interactions becomes an important research issue. Besides, captured motions are mostly used as canned motions for NPCs when certain events are triggered or certain conditions for the story or game are satisfied. If the behaviors of NPCs cannot adapt to user interactions, the plausibility of the virtual scene, as well as the replay value of the application, will be greatly reduced. Thus, in this work, we aim to design a 3D interactive storytelling system allowing intuitive 3D avatar control and generation of responsive and appropriate NPC behaviors. We will describe how the system is designed and report the experimental results from a user study.

In the following sections, we will first describe the related work about our research. In Sect. 3, we will describe how we have designed and implemented the interactive storytelling system. Then, we will report the result of the experiment that we have designed to evaluate the system. Lastly, we conclude the paper with some remarks and future research directions.

2 Related Work

Interactive storytelling is a form of entertainment allowing the authors and audience to co-create a unique experience of the story through user interactions. 3D interactive narrative is a special form emphasizing using a 3D virtual environment to deliver a story. To realize such an application, many techniques are involved such as drama generation, authoring tools, character animation generation, scripting languages, user interface design, etc. The Oz project [1] is one of the earliest interactive storytelling systems utilizing the concept of agents to provide impromptu interactions with users. Spierling et al. [13] introduced the concept of Narrative Formalism into interactive storytelling and use a layered model to create interactive stories and contents.

Unlike text-based interactive drama creation, the challenges for 3D interactive storytelling are more on the dynamic generation of multimedia contents and the design of multi-modal user interfaces. Kistler et al. [7] and Yang and Li [15] all have proposed to use gestures to interact with the objects in the virtual environment or to select a branch in a story graph. Cavazza et al. [3] have designed a multimodal user interface to allow a user to use voice input to influence the behaviors of virtual characters as well as the plot in an immersive 3D scene. In [4], a gesture recognition module also has been implemented for interactive storytelling. Brown et al. [2] reported a study attempting to find the features of iconic gestures for retelling a story. Piplica et al. [10] presented a system for combining improvisational acting with full-body motions detected through Kinect to support the co-creation of interactive narratives. Mousas et al. [8] proposed to detect user gestures with Kinect to trigger actions of virtual characters. Rhodin et al. [11] proposed to use Kinect and Leap Motion and predefined settings of gestures and velocities to present the corresponding animation of virtual characters. Eubanks et al. [5] recently presented two studies about the investigation of how body tracking fidelity is related to avatar embodiment.

There has been much research on the automatic generation or editing of character animations. For example, Tonneau et al. [14] proposed a technique to edit existing motion clips to respond to large environmental changes on the ground. Shoulson et al. [12] proposed an animation testbed allowing a user to leverage a character's animation and navigation capabilities when authoring both individual decision-making and complex interactions. In [6], the authors used precomputed semantic information about the environment to choose appropriate animation clips for the character to reach a given goal. Mousas et al. [9] also have proposed a system called CHASE allowing a novice user to design scripts with parameters to produce the desired animation. Besides, Yang and Li [15] also proposed to use a scripting language in XML to generate the animations for different story nodes in 3D interactive storytelling.

3 System Design and Implementation

To realize our VR system, we have adopted the Unity3D game engine as our experimental platform. In terms of hardware, we have used a relatively low-cost wearable motion capture device, called Perception Neuron, to capture the motion of a player. We have also chosen the HTC VIVE Head-Mounted Display (HMD) and controller as the devices for 3D rendering, audio outputs, and control input. The architecture of our system, as shown in Fig. 1, consists of three main modules: motion interpretation, animation management, and story management. The inputs, except for the ones from the user controller, also include data from external files such as story script, interaction script, and motion database. The authors of the story are in charge of creating the story script and interaction script while the player uses body gestures and the controller to interact with the environment or the NPCs. We will describe the main modules in the following subsections.

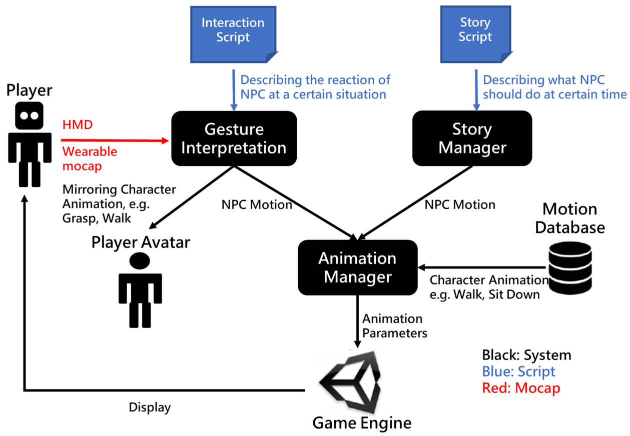

Fig. 1. System architecture.

3.1 Motion Interpretation Module

The motion interpretation module is in charge of interpreting player motions into regular gestures or specific commands for further processing. The motions are captured through a wearable device mentioned above and inputted into the system in a streaming format (BVH). Since the capture motions have many degrees of freedom, to reduce the control and computation complexity, we have only used a few parts of the body, such as hands and legs, for interpretation. Motion interpretation is needed because some of the player motions are designed to trigger certain events with interaction with the environment or NPCs. In addition, some motions, such as sitting down on a chair, cannot happen in the physical space as in the virtual space. Therefore, we have divided the inputs into two modes: direct input and command input.

Direct Input Mode. In this mode, the motions captured for the player are mirrored directly into the motions of the avatar. Therefore, the player can move his body as if he was the avatar. When certain conditions described in interaction scripts are satisfied, corresponding actions from the environment or the NPC will be triggered to respond to the player's intention for interaction. For example, in Table 1, we have listed the types of motions that will be recognized as meaningful actions and trigger events with corresponding motions by the affected objects or the NPC. For example, when the hand of a user avatar approaches an object, such as a glass or a hand of an NPC, and a grasp motion is performed, the object will be snapped into and moves with the hand (Fig. 2). Similarly, an object is released when the player ungrasps. The push motion is similar and can be used to push objects, such as doors, before entering or exiting a room. The motion of a short walk reflects a short-distance walk or position adjustment when the player moves around in the physical space.

Table 1. The types of motions recognized in the direct input mode

Motion types	Procedure to trigger an action
Grasp	Approach an object and grasp it when it turns red
Push	Touch the object and move forward. The object will move along
Wave	Raise the hand over the shoulder and wave
Short walk	Move the body, and the avatar follows

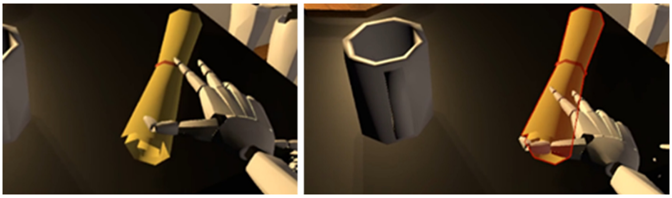

Fig. 2. The graspable object is highlighted (right) when the hand is close enough

Command Input Mode. Unlike the direct input mode where all motions are mapped to the avatar, there could be actions that the player would like to perform in the virtual world but is not physically feasible in the real world. For example, the physical space could be limited and does not allow the player to freely walk for a long distance as one may expect in the virtual world. Similarly, certain motions are not feasible simply because there are no corresponding objects in the real world. For example, when one would like to sit down on a chair in the virtual world, he/she cannot do so because there may not be a chair ready for this in the real world. As a result, we have to design special motions allowing a player to specify his intention for doing this type of motion, which we call the Command Input Mode. In the current implementation, we have designed three motions for command inputs as shown in Table 2. For the long-distance walk, since the player is not transporting his body, the avatar will move along his facing direction. To perform the sit-down motion, the player has to touch a chair first and then bend his knees to tell the system his intention for sitting down. Once the action is triggered, the player can stand up and allow the avatar to display the sit-down animation. Similarly, when the avatar is in a sit-down situation, the player can bend his knees to specify the intention for the avatar to stand up.

Table 2. The types of motions recognized in the command input mode

Motion types	Procedure to trigger an action	Corresponding animations
Walk	March on the spot by lifting legs alternatively	Avatar moves forward along the facing direction
Sit down	Touch a chair and bend the knees	Play the sit-down animation when activated. Walk to the chair if necessary
Stand up	Bend the knees when sitting	Play the stand-up animation

Animation Enforcing Mode. There are also some situations where we would like to enforce the display of certain animations to ensure the progress of a story or to simplify the interaction with the environment or NPC. In this mode, the player is still allowed to move his head to watch the display of enforced animations. In our demonstrative example, two situations may trigger this mode. The first case occurs when the story develops to a point where the player is shot by the suspect. Then a lying-down animation will be enforced even though the player does not perform this action. The second case is when the player (playing as a policeman) takes the suspect to the police car after arresting him, both need to enter the car. We choose to enforce the animations for both characters to avoid complex interactions and motion coordination which are not crucial for experiencing the story.

3.2 Interaction and Story Scripting

As depicted in Fig. 1, in our system, there are two types of scripts used to describe the interactive story and the interactions with the environment or NPCs: interaction script and story script. The interaction script is used to define how the environment or the NPCs should react to user interaction while the story script defines the story graph with branches for the interaction narratives.

Interaction Script. We have designed an interaction scripting language allowing an author to specify the reactions of an NPC when certain conditions are satisfied through the interpretation of player motions in two input modes as described above. The scripting language is an XML-based markup language, and an example is shown in Fig. 3. In the script, we define the conditions for triggering a reaction from an NPC and what kinds of responses (animation) should be taken for the interaction. Currently, there are six attributes defined for the `<InteractMovement>` tag that are used to specify the conditions and responses. For example, in Fig. 3, the interaction scripts for three actors (Suspect, Bartender, and Waiter) are described. In line 5, when the player, playing as a passerby, sits down in a good mood, the interaction movement of “give” for the suspect will be sent to the animation manager for offering a drink for 20 s. In the current system, event types of `RespondMovement` have been implemented.

```
1  <Interaction>
2      <Character tag="Suspect">
3          <InteractMovement  PlayerIdentity="Police"  PlayerMovement="all"  SelfMood="all"
           RespondMovement="shoot" Duration="20"/>
4
5          <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="sit"  SelfMood="good"
           RespondMovement="give" Duration="20"/>
6
7          <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="grab"  SelfMood="good"
           RespondMovement="beCatched" Duration="20" GrabTarget="SuspectStrongHand"/>
8
9          <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="walk"  SelfMood="bad"
           RespondMovement="walk" Duration="20"/>
10
11         <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="grab"  SelfMood="Great"
           RespondMovement="follow" Duration="100" GrabTarget="SuspectStrongHand"/>
12         <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="grab"  SelfMood="Great"
           RespondMovement="follow" Duration="100" GrabTarget="SuspectOtherHand"/>
13     </Character>
14     <Character tag="Bartender">
15         <InteractMovement  PlayerIdentity="Passerby"  PlayerMovement="sit"  SelfMood="good"
           RespondMovement="lookat" Duration="20"/>
16     </Character>
17     <Character tag="Waiter">
18         <InteractMovement  PlayerIdentity="Police"  PlayerMovement="waveHands"  SelfMood="good"
           RespondMovement="waveHands" Duration="5"/>
19     </Character>
20 </Interaction>
```

Fig. 3. An example of interaction scripts

Story Script. Unlike an interaction script which describes the responses of the environment or NPCs, the story script is used to describe the story graph consisting of story nodes connected with branches. It also includes nodes that are time-triggered, which means that some animations may start voluntarily without interactions. Thus, the storyteller can take an initiative to drive the story instead of waiting for the player to interact and trigger events. For example, the script in Fig. 4 describes that the suspect will start to move to a chair by walking for 15 s at an absolute or relative time.

```
<AnimCharacterMove StartTime="0" Duration="15"
ActorTag="Suspect" Movement="walk" destinationTag="Chair" />
```

Fig. 4. An example of story scripts

3.3 Animation Management Module

The animation management module is in charge of generating natural and responsive animations for the NPCs. In our interactive storytelling system, the animations of the NPCs could be triggered by the story management module because of the need for story development or by the motion interpretation module when the player interacts with the NPC. Since the motion interpretation module deals with real-time interactions with the player, the needs for responsive actions are usually higher. As a result, in the animation management module, we need to have a way to schedule the animations according to their priorities. In addition, the animation management module may also voluntarily insert a necessary transition motion to make the target motion feasible. For example, if the objective is for the NPC to sit down on a chair but the NPC is too far from the chair, the animation manager will issue a walk-to-chair animation for the NPC to reach the chair first.

In Fig. 5, we show an example scenario where the NPC receives animation requests from different modules, and the animation management module needs to arrange these requests according to their priorities. For example, at time `t1`, the suspect NPC received a request for a sit-down action from the story manager. But the animation manager realized that the suspect was too far from the chair and therefore inserted a walk action ahead of sit-down to reach the chair. At time `t3`, the player (policeman) walked in and was spotted by the suspect. As a result, the motion interpreter issued a shooting action, which takes the highest priority. Once the shooting is over, the suspect will resume walking to the chair and finally sit down at `t6`.

Suspect (t0)	Suspect (t1)	Suspect (t2)	Suspect (t3)	Suspect (t4)	Suspect (t5)	Suspect (t6)
	SM: sit	AM: walk	MI: shoot	AM: walk	SM: sit	
		SM: sit	AM: walk	SM: sit		
			SM: sit			

SM: Story Manager, AM: Animation Manager, MI: Motion Interpreter

Fig. 5. Animation scheduling in the Animation Manager

In addition to determining the priorities of the animations, the diversity and quality of the animations will also affect a player's immersion experience. By the diversity of animations, we mean that the responses that the player gets from NPC should vary according to the development of the story or the emotion of the NPC. According to the history of interactions with the player, we maintain three emotional states: bad, good, and great for the NPC. When a responsive action needs to be taken by the NPC, the animation manager will select an appropriate animation according to the emotional state of the NPC. Therefore, the responses a player sees may vary at different times of play.

The quality of the animation for an NPC can be considered from several aspects including the factors of cost and complexity. Since most of the animations for the NPCs are prepared and stored in the motion database, the diversity of animations will affect the cost of preparing such an interactive storytelling system. In our animation management system, we have attempted to minimize the need for creating a large animation database by reusing existing motions as much as possible. For example, many motions only focus on a certain part of the body. As such, one can decompose a motion into different parts that can be recomposed to form new animations. For example, as shown in Fig. 6, the hand gesture of the NPC does not depend on the lower body state. Thus, we can decompose the animation for the upper-body gesture and use it when the NPC is sitting on the chair.

Fig. 6. Example of decomposing animation for reuse

In addition to selecting different animation clips for different NPC emotions, the location of the user avatar, as controlled by the player, cannot be determined in advance. Thus, the animation management module may need to make a minor adjustment on the facing direction of the NPC toward the player to conduct a conversation. This kind of minor adjustment also happens when the player needs to have physical interactions with the NPC such as in the situation of arresting the suspect by grasping his hand. As shown in Fig. 7, the configuration of the hand being grasped is adjusted with inverse kinematics according to the player's hand location. We have used the built-in IK controller module in Unity3D for the implementation of this function.

Fig. 7. Example of computing hand configurations with inverse kinematics

4 Experimental Design and Results

To evaluate the system that we have designed, we have created a demonstrative scenario about an interactive story with multiple branches and endings. We have also designed an experiment with this scene and invite participants to experience the story and provide feedback through questionnaires.

4.1 Demonstrative Scenario

The demonstrative scenario is about a story in which a policeman (played by the participant) is asked to find a dangerous suspect of a crime who is reported to be present in the area around a bar. Snapshots of the scene are shown in Fig. 8. The policeman is supposed to find the suspect and arrest him if possible. However, since the suspect is dangerous and armed, it could be a good idea not to wear a police uniform and try to do the investigation privately. Therefore, in the beginning, the system will prompt the player about changing clothes in a special room in the area. If the player decides not to change clothes, he will be spotted by the suspect later when walking to the bar and got shot by the suspect. If the player chooses to change his clothes and enters the bar, he will find that one of the guests that could be the suspect is walking to the bar and will be sitting in front of the bartender. The player will be instructed to sit beside the suspect, and the suspect will start a conversation with the player and tries to offer him a drink. The conversation will release enough information for the player to confirm that the man sitting aside is the suspect. Then, he can decide to arrest the suspect by quickly grasping his hand and take him to the police car outside. If the grasping action is not fast enough, the suspect may find out and run away. In other words, depending on the history of the interactions, the story may develop into multiple endings.

Fig. 8. Snapshots of story development during the interactive storytelling (a) mission setup, (b) clothes changing, (c) suspect in a bar, (d) inviting for a drink, (e) releasing crime information, (f) arresting the suspect, (g) taking him to the car, and (h) putting him in prison.

4.2 Experimental Settings

We have invited eight subjects to participate in the evaluation. The ages of these subjects range from 22 to 28 with 4 males and 4 females, and half of them are experienced users of VR games while the other half has relatively few experiences.

The procedure for the experiment was a within-subject design and worked as follows. Each subject would experience the story twice with the VIVE controller and the wearable device, respectively. The order of the two experiences was random such that the learning effect could be canceled. Before the formal session started, we first explained to the subjects how the system worked and allowed the subjects to practice how to interact with the objects in a tutorial session through the wearable device and the VIVE controller, respectively. For the VIVE controller, we have adopted the most common ways of interacting with objects. For example, we use a touchpad to trigger teleport, a hair-trigger to grasp, and a grip to switch between sit-down and stand-up. At the end of each formal session, the subjects were asked to fill in a questionnaire about the operations, story, system functions, and immersion of their experiences.

4.3 Experimental Results

After each formal session, the subjects were asked to fill in three questionnaires for the evaluation of our system. The questions in the survey are answered on a 5-point Likert scale with 1 to 5 meanings strongly disagree to strongly agree. Three questionnaires are about the assessments on user interface operations, system and story, and immersion experience, respectively. The results (means and standard deviation) for both formal runs (controller and wearable device) are shown in Tables 3, 4 and 5. A paired two-tailed t-test has also been conducted for each question.

Table 3. Assessment of user interface operations

Questions	Controller		Wearable		
	M	STD	M	STD	t-test
1. The operation of movement is intuitive.	4.1	0.83	4.6	0.52	0.227
2. The operation of movement is smooth.	4.1	0.99	4.3	0.71	0.785
3. The operation of movement is interesting.	4.1	0.99	4.4	0.92	0.516
4. The operation of grasping is intuitive.	4.0	0.93	4.8	0.46	*0.048
5. The operation of grasping is smooth.	4.1	0.99	4.1	0.99	1.000
6. The operation of grasping is interesting.	4.1	0.64	4.6	0.52	0.104
7. The operations of sit-down and stand-up are intuitive.	3.1	1.25	4.1	0.64	*0.033
8. The operations of sit-down and stand-up are smooth.	3.8	1.39	4.4	0.52	0.217
9. The operations of sit-down and stand-up are interesting.	4.4	0.74	4.8	0.71	0.285
10. I can move my body freely.	3.8	1.39	4.5	0.76	0.303

*$p<0.05$

From the results in Table 3, we can find that the average scores for the wearable device are all higher than the controller. The variation for the wearable device is also lower than the controller. However, only the questions of intuitiveness on the grasping and sit-down and stand-up operations have achieved significant differences between the two types of interfaces. In the interview after the experiment, we also found that the scores for the controller quite depend on the prior experience of the subjects while all subjects consider the wearable devices are more intuitive to use.

Table 4. Assessment of story and system functions

Questions	Controller		Wearable		
	M	STD	M	STD	t-test
11. I can understand the story.	4.9	0.35	4.8	0.46	0.351
12. I think the story is interesting.	4.6	0.52	4.4	0.74	0.451
13. I think the progress of the story is smooth.	4.6	0.74	4.6	0.52	1.000
14. I want to experience different story plots.	4.6	0.52	4.9	0.35	0.170
15. The audial and textual prompts help me interact properly in the story.	4.9	0.35	4.8	0.46	0.351
16. I feel that the NPC is interacting with me.	4.4	0.52	4.3	0.46	0.598
17. I think I have influenced the development of the story.	4.5	0.53	4.5	0.53	1.000

*$p<0.05$

The results in Table 4 reveal that the user feedback about the story and system functions are all very positive for either interaction interface but no significant difference has been found. The subjects have found that being able to interact with the NPC and influence the development of the story is interesting.

Table 5 shows the result of assessing the immersion of the player. The subjects are more immersed in the environment with the wearable device but all enjoy playing in the scene with both input devices. The variation of the scores is higher for the controller session probably because their familiarity with the controller varies. On the other hand, from questions 28 and 30, we can find significant differences between the two interfaces, and the wearable mocap device provides a more intuitive interface allowing a player to immerse into the virtual environment more easily.

Table 5. Assessment of immersion

Questions	Controller		Wearable		
	M	STD	M	STD	t-test
18. My mood was up and down as the story develops.	3.9	1.13	4.0	1.20	0.598
19. I wanted to know how the story developed.	4.3	1.04	4.8	0.46	0.227
20. I am worried about if I can accomplish the mission.	4.0	0.93	4.1	1.13	0.598
21. I found myself in the story and would like to dialog with the virtual characters.	4.1	0.83	4.5	0.53	0.285
22. I enjoy the scene in the game.	4.3	0.71	4.4	0.52	0.351
23. I enjoy playing the game.	4.6	0.52	4.4	0.52	0.351
24. I think the operations in the game were easy to learn.	4.3	0.71	4.5	0.53	0.516
25. I am not aware of using any controller.	2.4	1.30	4.5	0.76	**0.001
26. I can move according to my will.	4.3	0.89	4.4	0.74	0.802
27. I can interact with the virtual world like in the real world.	3.3	1.16	4.1	0.64	0.087
28. I was not aware of what was happening in the real world during the play.	3.4	1.06	4.3	1.16	*0.021
29. I felt that the game was my only concern.	3.9	0.83	4.1	0.64	0.351
30. I would not stop playing the game to see what happened around me.	3.8	0.71	4.4	0.52	*0.049
31. I felt the time flies when I played.	4.4	0.74	4.6	0.52	0.170

$^{*}p<0.05$, $^{**}p<0.01$,

5 Conclusions and Future Work

There has been much research on interactive storytelling and different ways to design and experience a 3D interactive story. In this paper, we have attempted to design a 3D interactive narrative system allowing a player to wear a mocap device to participate in an interactive story. To realize such a system, in addition to using the wearable device, we also have developed a way to receive user inputs by recognizing the motion and intention of the player under system guidance and to provide plausible responses from the NPC through realistic animations. We have also conducted a study to evaluate the system from several aspects. The experimental result reveals that the interactive storytelling system has achieved the goal of allowing the players to participate in the story through interactions with both the controller and the wearable devices, respectively. The wearable mocap device provides a more intuitive way to interact with the environment or NPCs and is also easier to learn for novice users. Besides, the subjects are better engaged with the story and immersed into the environment with the wearable device and thus have a better story experience in general.

In the current system, the dialogs between the player and the NPC are all predetermined and pre-recorded. Therefore, although the story has multiple plotlines and endings, the system still cannot change the story on the fly with customized contents or dynamic dialogs. To allow this to happen, more research on dynamic drama creation and scene generation will be necessary. Besides, although the cost for wearable mocap devices is becoming affordable, the quality for stability may be compromised. We will be looking forward to the development of this type of input device for it to become more affordable and precise.

Acknowledgment. The authors would like to thank the sponsor of this research: Ministry of Science and Technology in Taiwan under contract MOST108-2221-E-004–007-MY3.

References

1. Bates, J.: Virtual reality, art, and entertainment. Presence: J. Teleoper. Virtual Environ. **1**(1), 133–138 (1992)
2. Brown, S.A., et al.: Towards a gesture-based story authoring system: design implications from feature analysis of iconic gestures during storytelling. In: Cardona-Rivera, R., Sullivan, A., Young, R. (eds.) ICIDS 2019. LNCS, vol. 11869, pp. 364–373. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33894-7_38
3. Cavazza, M., Lugrin, J.L., Pizzi, D., Charles, F.: Madame bovary on the HolodeckL immersive interactive storytelling. In: Proceedings of the 15th ACM International Conference on Multimedia, pp. 651–660 (2007)
4. Cavazza, M., Charles, F., Mead, S.J., Martin, O., Marichal, X., Nandi, A.: Multimodal acting in mixed reality interactive storytelling. IEEE Multimed. **11**(3), 30–39 (2004)
5. Eubanks, J.C., Moore, A.G., Fishwick, P.A., McMahan, R.P.: The effects of body tracking fidelity on embodiment of an inverse-kinematic avatar for male participants. In: Proceedings of 2020 IEEE International Symposium on Mixed and Augmented Reality (2020)
6. Kapadia, M., et al.: PRECISION: precomputing environment semantics for contact-rich character animation. In Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp. 29–37 (2016)
7. Kistler, F., Sollfrank, D., Bee, N., André, E.: Full body gestures enhancing a game book for interactive story telling. In: Si, M., Thue, D., André, E., Lester, J.C., Tanenbaum, T.J., Zammitto, V. (eds.) ICIDS 2011. LNCS, vol. 7069, pp. 207–218. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25289-1_23
8. Mousas, C.: Towards developing an easy-to-use scripting environment for animating virtual characters. arXiv preprint arXiv:1702.03246 (2017)
9. Mousas, C., Anagnostopoulos, C.-N.: Performance-driven hybrid full-body character control for navigation and interaction in virtual environments. 3D Res. **8**(2), Article No. 124 (2017)
10. Piplica, A., DeLeon, C., Magerko, B.: Full-body gesture interaction with improvisational narrative agents. In: Nakano, Y., Neff, M., Paiva, A., Walker, M. (eds.) IVA 2012. LNCS, vol. 7502, pp. 514–516. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33197-8_63
11. Rhodin, H., et al.: Generalizing wave gestures from sparse examples for real-time character control. In: Proceedings of ACM SIGGRAPH Asia 2015, vol. 34, no. 6, Article No. 181 (2015)
12. Shoulson, A., Marshak, N., Kapadia, M., Badler, N.I.: ADAPT: the agent development and prototyping testbed. IEEE Trans. Vis. Comput. Graph. **20**(7), 1035–1047 (2014)

13. Spierling, U., Grasbon, D., Braun, N., Iurgel, I.: Setting the scene: playing digital director in interactive storytelling and creation. Comput. Graph. **26**(1), 31–44 (2002)
14. Tonneau, S., Al-Ashqar, R.A., Pettré, J., Komura, T., Mansard, N.: Character contact re-positioning under large environment deformation. In: Proceedings of the 37th Annual Conference of the European Association for Computer Graphics, pp. 127–138 (2016)
15. Yang, C.C., Li, T.-Y.: Participating in narratives with motion-sensing technologies in a 3D interactive storytelling system. In: Proceedings of Computer Graphics Workshop, Taipei (2015)