Quantifying “Magic”: Learnings from User Research for Creating Good Player Experiences on Xbox Kinect

Kristie J. Fisher, Microsoft Studios, Redmond, WA, USA
Tim Nichols, Microsoft Studios, Redmond, WA, USA
Katherine Isbister, School of Engineering Game Innovation Lab, New York University, New York, NY, USA
Tom Fuller, Microsoft Studios, Redmond, WA, USA

ABSTRACT

In November 2010, Microsoft released the Kinect sensor as a new input device for the Xbox 360 gaming console, and more recently the “next generation” of Kinect was released in November 2013 as part of the Xbox One entertainment system. Kinect has the ability to detect multiple points of skeletal movement, differentiating among multiple simultaneous users. This ability enables users to control and interact with on-screen elements by moving their bodies in space (e.g., move characters, select menu items, manipulate virtual objects). Controllers or on-body sensors are not needed to use gesture inputs with Kinect, and Kinect can also detect speech inputs. The team at Microsoft Studios User Research (SUR) was an integral part of creating the first full-body gaming experiences for the Kinect system. During the development of Kinect, and in the more than 3 years since its initial launch, SUR has worked with game designers, programmers, and hardware developers on games and other applications that use Kinect. In this article the authors leverage data SUR has collected over the development cycles of many different games created for many different audiences to summarize the unique user experience challenges that the Kinect sensor brings to game development. The authors also propose principles for designing fun and accessible experiences for Kinect.

Keywords: Gesture, Kinect, Microsoft, Motion Gaming, Video Games, Xbox

INTRODUCTION

Video games can provide players with a wide range of experiences, from the thrill of shooting enemies in a highly-realistic combat scenario to the challenge of solving complex spatial puzzles, to the simulation of racing a Formula 1 car, to the simple joy of beating a friend in virtual Scrabble™. A common goal for all video games, though, is to either allow players to experience things that they cannot do or that do not exist in real life, or to greatly enhance the fun, reward, or challenge of real life experiences by creating a “game-ified” version of them. The Kinect (See

DOI: 10.4018/ijgcms.2014010102
Figure 1) full-body motion gaming sensor for the Xbox 360 allowed for the creation of new types of games based on experiences that had been difficult to “game-ify” in the absence of such full-body motion input technology, such as dance, fitness, and augmented reality. It also has the potential to make video games from more “traditional” genres (action, combat, racing, etc.) more immersive by allowing users to more “directly” interact with them.

THE VISION OF KINECT

Kinect was designed with a few specific goals in mind. First, Kinect was meant to expand the technical capabilities of motion gaming. While Kinect was being developed, an extremely popular motion gaming device was the Nintendo Wii. The Wii requires the player to move a handheld controller through space in order to interact with its games. This constrains the user experience in some ways, because Wii games are programmed to attend only to the location of the controller relative to the sensor, meaning that the rest of the player’s gestures are irrelevant. Typically the player uses the standard “Wii-mote” controller to interact with the system, but some games require a secondary controller accessory, which requires users to have a collection of input devices. Similarly, the Sony EyeToy, which was a motion input device for the Playstation 2 that pre-dated the Wii, allowed for some controller-free gesture input, but its functionality was extremely limited. There was therefore an opportunity to advance motion gaming to include inputs derived from full body tracking of multiple players in 3D space as well as speech inputs. In expanding the technical capabilities of motion gaming, the possibilities for player experience could also expand.

As one result of this increased technical capability, the creators of Kinect wanted using it to feel “magical.” The design philosophy behind this was that when a game removes the intermediate input device between the user and the system – the game controller – then the players’ ability to interact with games “directly” using their bodies would inherently be more immersive than traditional controller gaming experiences. Indeed, the idea that movement can enhance the engagement and emotion of players is supported by some researchers in the field of human-computer interaction (e.g., Bianchi-Berthouze, Kim, & Patel, 2007; Lindley, Le Couteur, & Berthouze, 2008).

Figure 1. The Microsoft Kinect
Kinect also had the potential to broaden the Xbox 360 audience beyond the “traditional” console gamer audience, as the target audience for most games created at Microsoft Studios for the Xbox 360 is males between the ages of 18–40. That is, with Kinect there were new opportunities to create casual gaming experiences that could appeal to whole families and to individuals who were intimidated by popular video game genres, such as shooters, and/or by the steep learning curve of the Xbox 360 controller. One of the advertising slogans for Kinect is “All you need is you,” and the official Xbox website explains, “You already know how to play” (retrieved Feb. 26, 2014 from http://www.xbox.com/en-US/kinect), which implies that no previous gaming experience is required in order for players to play and enjoy games made for the Kinect. More specifically, part of the creative vision for Kinect was that players could “intuitively” know what to do in order to play the games without being given any instructions.

THE ROLE OF USER RESEARCH AT MICROSOFT STUDIOS

The user research group was one of the teams at Microsoft Studios tasked with helping to realize the design goals for Kinect. Microsoft Studios is responsible for creating games and other entertainment content for Microsoft’s platforms, and Microsoft Studios User Research (SUR) has existed at Microsoft since 2000. The group was created during the development of the original Xbox console with the task of determining whether the games being created for the Xbox would be fun for players. Its function is to work in close partnership with game designers from the earliest stages of the production cycle to ensure that players are having the experience that the game creators intend for them to have. To do this, SUR operationally defines relevant aspects of user experience (for example, “fun,” “mastery,” “pace,” “frustration,”) and gathers behavioral, attitudinal, game telemetry, and other data from users via a variety of methods derived from academia and industry. SUR then collaborates with the game creators to iterate on the game experience in response to those data (see Pagulayan et al., 2007, for a more detailed summary of games user research work at Microsoft Studios).

SUR has been involved in the development of Kinect games from their early incubation throughout the present. In light of the new gesture- and speech-based interaction models, new types of game experiences, and new types of players that Kinect introduced to the Xbox system, SUR adapted its methodologies to provide the same type of data and insights to game development teams that it had for traditional controller games. In the remainder of this article, we first describe the learnings SUR has assimilated throughout the last 5 years regarding how the unique capabilities and limitations of Kinect affect player experience. We then describe best practices for designing games for Kinect that are most likely to create good experiences for users. Most of the findings discussed here are distilled from numerous usability studies of Kinect games conducted at Microsoft, and all games described here that were developed by Microsoft Studios have been released to the public. It is important to note, however, that specific data and detailed methodology from usability studies and other research on internal in-development titles or on competitor titles is considered Microsoft confidential. Therefore, in the remainder of the article we present our view on full-body motion gaming as industry experts, but we are able to describe only our high-level learnings from our studies.

KINECT’S UNIQUE USER EXPERIENCE CHALLENGES

Kinect presented players with entirely new interaction models, and while novelty alone can present a challenge to user experience, the inherent nature of the system presents additional challenges, especially in light of the design vision for Kinect.
Very Few Gestures are “Intuitive,” and Gestures are Hard to Teach

One of the user experience goals for Kinect was that using it should feel “intuitive.” Instructions and tutorials should be unnecessary, and players should simply know what to do as soon as they step up to use the system. In the fields of cognitive psychology and human computer interaction, the term “intuitive” typically implies that within a given system there are proper affordances to guide the user to the correct action and/or that the user has prior knowledge that he or she can apply to interact with the system (e.g., Norman, 2010, 2013). The work done by SUR during the development of many Kinect games revealed that, even with good affordances, it was generally difficult for users to correctly guess the exact gestures the system was expecting. For example, given an on-screen virtual ball and a target, it might be “intuitive” that the object of the game is to throw the ball at the target, but there are a variety of ways that one can “throw” a ball (overhand, underhand, “push”, two handed, spinning around first to get momentum, etc.). If only one of those ways is “correct” according to the input the system is expecting, then the system will only work for the handful of users who happens to have the correct “intuition.”

For example, several early Kinect games required users to initiate engagement with the game by “waving” at the Kinect sensor. The template wave gesture that the Kinect was programmed to detect was performed as follows: the user held his or her hand up parallel with the body (with the elbow bent and the open palm facing the television) and steadily moved the hand to the side, away from the body, and back, for a range about 90 degrees. However, users of all ages and levels of experience with video games and technology interpreted the instruction to wave at Kinect in a numerous, distinct ways (See Figure 2).

This inability of players to have an “intuitive” experience with Kinect’s gesture inputs is exacerbated by the inherent limitations of the Kinect system to distinguish player intentions from player inputs and by the inherent variability of individuals, as described in the section below. Because having an inherently “intuitive” system was practically impossible for most Kinect games, the focus of SUR and the game development teams when creating the first games for Kinect shifted to providing players with comprehensive gesture instruction and in-game feedback (See Figure 3).

Gesture instruction has its own challenges, however, because learning to perform a gesture precisely and accurately is very difficult for people generally (e.g., Allard & Starkes, 1991); athletes, dancers, and musicians dedicate lifetimes to mastering precise muscle movements. SUR found that creating good learning systems for Kinect not only involved having clear instructions for players, but also having relatively loose gesture input requirements to accommodate variability in how players executed the

Figure 2. Visual examples of users waving to interact with Kinect
gestures that they learned. Hinrichs and Carpen
dale (2011) came to a similar conclusion in a
field study of a “natural user interface” (NUI)
system requiring gesture inputs, that is, that
flexibility of the system to respond to multiple
types of user input is important for creating an
accessible and enjoyable user experience. We
further discuss player instruction in the Creating
Good Player Experiences with Kinect section.
Clear feedback to players regarding gesture
execution is also difficult to provide because in
order to do so the game needs to make accurate
assumptions around what the user’s intentions
are, which is difficult for Kinect, as discussed
below. A best practice developed by SUR was
that every input gesture a user executes should
have some type of feedback associated with it;
furthermore, all intended user behaviors should
have visual and/or auditory feedback. Conduct-
ing extensive user testing on Kinect games can
also reveal common types of “mistakes” people
will make, which can help to improve instruc-
tion and feedback.

User Intent is Ambiguous
and User Input is Variable

With a gamepad controller, the ways in which
user input can be variable are constrained by
the nature of the system, and the system can be
easily programmed to recognize and respond
to nearly all input variations. On the Xbox 360
controller there are 15 buttons and two analog
joysticks, and user input is comprised of which
buttons are pressed, the order in which they are
pressed, and the duration with which they are
pressed. With gesture inputs, however, there
are infinite degrees of freedom in how a user
might execute a given gesture, even when the
user has a correct understanding of what ges-
ture he or she is meant to perform. Additional
variations in the size, shape, and mobility of
human players themselves further add to the
variability of gesture inputs.

Moreover, with the console gamepad
controller, player intent almost always matches
player input. For example, if a player approaches
an in-game enemy and hits the attack button, the
game will register that the button was pressed
and will respond according to the programmed
game rules. It is safe for the game to “assume”
that when a specific button press is registered,
the user intended to execute the corresponding
action (excepting cases in which the user does
not know which actions are mapped to which
buttons or accidentally presses the wrong but-
ton). If during game development it is observed
that a player’s attempt at executing an attack
repeatedly fails, then the player inputs can be
examined within the game context to determine
why the failure occurred and what design solu-
tion, if any, is required to ameliorate it.

In the case of this attack example, perhaps
the player did not execute the attack at the ap-
propriate distance from the enemy, which may
mean that player training was ineffective or that
the game’s logic should be revised to adjust
the attack range to match player expectations.
Alternatively, perhaps the attack is designed to

Figure 3. Comprehensive gesture instruction

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grab</td>
<td>Pick up javelin by placing one arm out to side</td>
</tr>
<tr>
<td>Run</td>
<td>Run in place in straight line (no turning)</td>
</tr>
<tr>
<td>Pull arm back</td>
<td>Pull arm back while running</td>
</tr>
<tr>
<td>Forward Arm Motion</td>
<td>Move arm forward in a high, overhand throwing motion</td>
</tr>
<tr>
<td>Release/follow through</td>
<td>Follow through</td>
</tr>
</tbody>
</table>

Copyright © 2014, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
require a level of precision in execution that the player must learn to master over time. In this case there would be no “issue” to fix as long as the player is aware of this mastery aspect of gameplay. When user intentions are known (the game detects the attack button input, so it “knows” the player wants to attack), then comparing design intent to player experience can be done simply and directly in most cases. When user intentions are known (the game detects the attack button input, so it “knows” the player wants to attack), then comparing design intent to player experience can be done simply and directly in most cases.

In a full-body gesture controlled game, however, the input registered by the game may not be an accurate representation of what actions the user is actually doing or intending to do. This is a broader problem with gesture-based NUIs in that what the machine can “sense” usually does not encompass everything the user can do, and what the user is doing at a given moment may not be the input that the machine desires (Benford et al., 2005). For example, when a user is intending to swipe his or her hand from the right to the left side of the screen, the mere act of raising the hand and positioning the arm to perform the swipe may be interpreted by Kinect as an input gesture. In many cases, this might trigger an unintended action from the system (See Figure 4).

When the system responds to the user in unexpected ways, due to an incorrect interpretation of intent, players can develop a flawed mental model of how to interact with a game. For example, in the game Kinect RUSH: A Disney Pixar Adventure (2012), players control the movements of an avatar in 3D space. One of the most frequently used avatar actions was the “run” action. For the run input gesture, the speed of the avatar was controlled by the player swinging his or her arms in a run motion. In one area of the game, players were meant to perform a “swim,” gesture, which resembled a breast stroke. While the game was still in development, it was observed during testing that some players (who were young children, the target audience) would forget the correct gesture in these swim areas and would default to using the gesture they were most familiar with, run (See Figure 5). Because the run gesture and the swim gesture both involve forward arm movement, the system occasionally responded to it, but not reliably, and not in proportion to the increased intensity of players’ run gesture input, which led to a kind of “start-stop” partial progression of the on-screen avatar. In this way,
players could become frustrated or could feel like they did not have control over their avatar. However, because the system was responding at all, the users didn’t realize that they were performing the incorrect gesture.

The user experience issues caused by the variability in player inputs and the inherent limitations of the Kinect system in interpreting player intent can be mitigated somewhat by good instruction and feedback. However, these limitations should also be considered in the initial phases of conceptualizing and designing games, as discussed in the Creating Good Experiences with Kinect section of this article.

**Kinect Must Manage Players in 3D Space**

Players interact with Kinect by moving their bodies through space and/or by speaking to it, and multiple players can interact with Kinect at once. Because of this, a Kinect game has the added challenge of player management—knowing which player is which and where players are in the room and relative to one another—which controller games do not. Because Kinect was meant to appeal to families and casual gamers, being able to play with friends or family members as a group was an important capability. However, all of the issues described above with player instruction, in-game feedback, player intent, and gesture variability, are compounded when multiple users are interacting with the system. The Kinect first needs to communicate to players how to initiate interaction with it (e.g., “Player 1 – Raise hand to start!”), then it needs to determine which players are attempting to engage, and lastly it needs to ignore inputs from other individuals who may be in the room but not actively playing the game.

Once a game has begun, Kinect needs to keep track of player identity and location and to differentiate gesture inputs from each active player, which can be a technical challenge if players’ limbs or bodies overlap as they move about to interact with the game. Kinect must also keep players within the boundaries of the “playspace” (the area in which Kinect can detect players), and it must alert players when they have moved outside of the ideal boundaries (See Figure 6). All of these player management features can impact player experience and have design implications. Furthermore, in any kind of social situation, players’ attention is likely to be divided between the game and the other

*Figure 5. An example of ambiguous feedback to incorrect gestures*
individuals in the room, which creates additional constraints for providing effective instruction and feedback.

**CREATING GOOD PLAYER EXPERIENCES WITH KINECT**

The new types of game experiences enabled by Kinect are accompanied by new user experience challenges. Creating good player experiences in Kinect games requires designing games that leverage the unique capabilities of Kinect while circumventing its limitations and avoiding some of the potential issues described above. The following best practices are specific to Kinect, but are aligned with more generalized motion game design principles recently suggested by Mueller & Isbister (in press).

### In Kinect Games, the Gestures Should Be Fun in and of Themselves

Given that Kinect differentiates itself from its competitors with its 3D camera and its ability to use full-body gestures as inputs without the aid of a secondary device, players’ gestures should be at the center of any good Kinect experience:

- The gesture controls themselves should be enjoyable to execute.

In Kinect Sports Season 2 (2011), players in SUR studies reported that it was fun to make throw gestures when playing football, because it allowed them to feel like they were really playing the game in a way that went above and...
beyond the feeling of throwing a football in a controller game.

- Gesture controls should correspond to the experience of the game.

  For example, performing squats as an input gesture is appropriate for a fitness game, but probably not for a shooter game.

- Gesture controls should not cause discomfort or inappropriate fatigue during the duration of expected play sessions.

  The goal of some full-body motion games may be for players to be active or to exercise, and in this case feeling fatigued or having sore muscles after playing would be appropriate, but this should not be the case for games in which physical exertion is not a design goal. For example, in Child of Eden (2011) players stand in front of Kinect and move one or both of their arms around a screen to control the reticles of weapons used to destroy enemies. Maintaining this kind of standing position for a 30 minute gaming session can cause fatigue and back ache.

- “Enhance” the players’ experience of executing a gesture, rather than simply providing an on-screen representation of that movement.

  In Dance Central (2010), a “model dancer” avatar (rather than a representation of the player) is at the forefront of the screen so that players can feel that they are embodying a stylish dancer who is executing the moves perfectly, whether or not that is actually the case. In Kinect Disneyland Adventures (2011) there is a park “attraction” in which players control a character flying through the sky. The feeling of controlling a character who is flying is enhanced by using gesture inputs because of the unique “first person” perspective achieved when the player’s body movements control the game camera. In Puss in Boots (2011) players have partial control over an on-screen character, and when they perform sword fighting moves the on-screen character does not match player movements exactly, but rather performs exciting animations in response to players’ inputs, which enhances the feeling of combat.

- Give players the opportunity to express themselves through movement when possible.

  In Kinect Sports Season 2 (2011) the movements of the on-screen character closely match the movements that a player performs, so if after winning a player jumps up and down to celebrate, the on-screen character will do the same. Similarly, Dance Central 3 (2012) has modes that allow players to earn points through “free style” dancing.

If a Game Might Be Easier or More Fun for Players to Control with a Gamepad Controller, then it is Probably Not Appropriate for Kinect

In short, a game should have appropriate precision and timing requirements based on Kinect and player limitations. If players are asked to perform complex sequences of gestures with the same precision and timing with which they are asked to perform complex sequences of button presses on a gamepad, then they are highly likely to fail and become frustrated:

- What Kinect defines as a “successful” input should incorporate both player variability and the game context in which it must be performed.

  For example, if a gesture must be performed quickly while under pressure, then the required gesture input should be more loosely defined in the system. Kinect Star Wars (2011) requires players to perform sequences of complex
gestures in rapid succession during combat, which strains the capabilities of players and the Kinect; as a result, in SUR studies many players reported that controls could feel unresponsive. In contrast, the game Puss in Boots (2011) essentially equates amount of gesture activity (speed and frequency) with amount of damage that the player does to the enemies during combat, and it accepts as input nearly any type of arm “swipe” gesture in any direction, facilitating player success.

- Input gestures should be as distinct from one another as possible with respect to the system’s capabilities, and “big,” simple gestures should be favored over subtle, complex ones.

While still in development, Fable the Journey (2012) experimented with various types of gestures for players to use to cast spells. SUR observed that complex, multi-step gestures were often mis-interpreted by the Kinect sensor. The Kinect especially confused the inputs from gestures that required the player to bring his or her hands together in front of the chest, because Kinect detection of overlapping limbs was relatively less robust. These complex gestures were eventually abandoned for very simple spell casting gestures.

- Avoid “Kinect-ifying” an existing gamepad controller experience. Many “traditional” console games are not designed in a way that affords “translating” button presses into gesture inputs.

In Kinect Disneyland Adventures (2011), character navigation was difficult for many players in SUR studies. Moving a character through 3D space with precision (which requires 360 degree turning, quick direction changes, avoiding obstacles, etc.) is something that is easily accomplished with gamepad button or joystick inputs. With gesture inputs, in contrast, it is impossible to create a one-to-one mapping of movement, as it is with a joystick, because players must face the screen and must stay within the Kinect playspace in order to play the game (in this game, players bend the arm at the elbow and use the hand to direct the movement of the character). These limitations make it very challenging to create good 3D character navigation experiences for Kinect.

- If the game experience requires lots of graphical user interface menu navigation, leverage speech inputs as much as possible.

This applies to both Kinect games and controller games “augmented” with Kinect inputs. For example, in The Elder Scrolls V: Skyrim (2011), a detractor of fun reported by players in post-release competitor studies conducted by SUR was that the Inventory and Favorites menus were difficult to learn how to use and navigation through the menus could be slow and tedious. This could cause problems for players attempting to use the menus quickly to execute important in-game actions. The use of Kinect speech commands allows players to bypass the use of these menus.

Ensure Appropriate Player Instruction

As discussed in earlier sections, effective player instruction around gesture inputs is essential for a good Kinect user experience:

- The gesture instruction should be primarily visual in nature.

As discussed above, learning a gesture is inherently difficult, and the best way to teach is to show.

- Highlight the parts of the body that are most important for executing a gesture.

When presented with a visual demonstration of a gesture, players may not spontaneously attend to the part of the gesture that is most relevant to the input the system is expecting,
so gesture instructions should emphasize the parts of the body that are most important (See again Figure 3).

- Text or audio cues should describe important aspects of the gesture that are difficult to illustrate or emphasize visually.

For example, in Kinect Sports (2010), players are told via voice-over instructions that lifting their knees high while running in place will increase their character’s speed.

- User testing can help to determine the range of gesture inputs that the system should accept, given player variability, and can help guide the creation of effective player instruction tools.

Because children have no “real world” experience driving cars, when attempting to use a “steering wheel” gesture during the development of Kinect RUSH: A Disney Pixar Adventure (2012) and Kinect Joyride (2010), children were often observed during SUR tests to make very large gestures with their arms and to cross their hands when trying to turn. User testing helped to determine an appropriate sensitivity for steering and informed the look of the animated gesture instruction figures.

- Provide players with an opportunity to practice gestures and receive feedback without risking failure.

A universal best practice for all types of video games is providing a place for players to learn and practice game controls without the risk that their actions will result in failure. For example, in the original Halo: Combat Evolved (2001), players are given the opportunity to practice moving their character and controlling the camera before ever encountering an enemy or acquiring a weapon. Given the difficulty of learning to perform gestures, this principle is even more salient for Kinect games.

- When possible, provide feedback not just on whether or not the player is doing a gesture correctly, but also on how the player needs to adjust his or her motions.

Because mastering kinesthetic awareness and executing gestures with precision is so inherently difficult (e.g., Allard & Starkes, 1991), it is not helpful for the game to simply tell the player that their input is incorrect. Rather, it needs to tell the player what to do to correct his or her action (See again Figure 6, which shows how Kinect Adventures, 2010, instructs players on how to move back into the playspace).

- Do not require users to memorize too many gestures, and provide frequent cues to players to remind them of required gesture inputs.

Human working memory is limited (e.g., Baddeley, 1992), so games should provide users with frequent cues reminding them of the appropriate gesture to perform (especially during a user’s initial experience with the game) so that the user does not become “stuck.” For example, in the Kinect Sports (2010) hurdles event, the player must only perform two gestures, running and jumping, and there is a multi-faceted jumping cue so that players can prepare to jump and execute the jump with correct timing.

Kinect Should Leverage a User’s Social Context when Possible

Kinect games can be a better group experience compared to controller games because individual controllers are not required to accommodate each player, and because the lines between “players” and “observers” are often blurred. That is, Kinect can accommodate a “jump in, jump out” player-switching model and can detect speech inputs from any nearby individuals.

Therefore, whenever possible, games should find ways to include others in the room in the fun, even if indirectly (a design
best practice that is also supported by Reeves, Benford, O’Malley, & Fraser, 2005). Games should also strive to make the experience of watching others play enjoyable (see Figure 7 for an example). Kinect’s ability to entertain a large group by providing a fun observer experience, by “knowing” about other players in the room, and by allowing others to participate in the experience in some way is something that can contribute to the feeling of “magic” that is part of Kinect’s design vision.

Kinect “Augmentation” Should Not Be Disruptive

Some Xbox 360 controller games (e.g., Elder Scrolls V: Skyrim, 2011; Mass Effect 3, 2012; Halo: Combat Evolved Anniversary Edition, 2011) have used Kinect’s speech input capabilities to augment the game experience, for example by allowing players the option of using speech commands for menu navigation or as shortcuts to performing some in-game actions. Though speech inputs are inherently simpler and faster for players to execute than gesture inputs, for each game there exists a unique syntax for how speech commands need to be executed in order for the system to recognize them; this creates the same need for player instruction and feedback as with gesture inputs. The Xbox 360 dashboard provides a good example for how to cue users to the existence and proper use of speech commands (see Figure 8). More recently, Dead Rising 3 (2013) for the Xbox One allows players to direct non-player characters by pointing at the screen, in addition to using speech commands. Zoo Tycoon (2013) allows players the option to use Kinect to interact “directly” with the animals in their virtual zoo.

All of these types of Kinect-augmented experiences have the potential to enhance players’ gameplay by allowing them to accomplish some tasks more efficiently and/or by allowing them to interact with the game in ways that are more realistic or engaging (for example, actually speaking to a character rather than pressing a button to “have a conversation”). However, it is important that a given Kinect augmentation does not require the player to shift his or her attentional focus from the primary game experience and does not put the player at risk of failure if the player is required to put down the controller in order to interact with Kinect. The best Kinect augmentations for games explicitly inform players of their existence, have a system for teaching players how to use them, and provide players with affordances and

Figure 7. Example of how multiple players can enjoy using the Kinect
feedback that they are able to attend to given their current gameplay context. Lastly, Kinect features should be optional to accommodate players who cannot or do not want to use them.

CONCLUSION

Kinect is a unique technology with both amazing capabilities and inherent limitations. These capabilities and limitations, and how they intersect with human cognitive abilities when it comes to learning and executing gestures, need to be considered carefully when creating games. Even as the technical power of Kinect increases in the “new generation” of gaming consoles, such as the Xbox One and beyond, there are some user experience challenges that will remain despite any new technical achievements.

Specifically, players will always need clear affordances, effective instructions, and timely, informative feedback in order to properly execute gesture and speech inputs. Furthermore, gesture and speech controls that are too complex to teach players or that are too similar to one another given innate human variability and system limitations will always cause problems for the player and the system.

ACKNOWLEDGMENT

The authors would like to acknowledge all of the members of the Microsoft Studios User Experience team for their contributions to this research.
Research team, whose work contributed to the learnings described in this article. We would also like to acknowledge Microsoft Studios design director Clayton Kauzleric for his insights and feedback.

REFERENCES


---

**ENDNOTES**

1 The Xbox One was released in North America on November 22, 2013. The Xbox 360 gaming console was released in November, 2005. The Kinect sensor was released in November, 2010. According to the earnings report released by Nintendo in 2009, by the end of the 2009 fiscal year the Nintendo Wii had sold 50.39 million units since its release in November 2006 (Nintendo, 2009). The Sony EyeToy was released in 2003, and as of 2008 it had sold 10.5 million units (Kim, T., 2008).

2 The first Xbox console was released in North America on November 15, 2001.

3 More information about Studios User Research can be found at their website, www.studiosuserresearch.com

---

Kristie J. Fisher is a user researcher at Microsoft Studios User Research and holds a PhD in Cognitive Psychology from the University of Washington.

Tim Nichols leads user research on future platforms at Microsoft Studios User Research and received his PhD in Engineering Psychology from Georgia Tech.

Katherine Isbister is the Research Director of the NYU School of Engineering Game Innovation Lab—she is jointly appointed between Computer Science and the Tisch Game Center.

Tom Fuller is a Design Researcher with Xbox One, prior to which he worked at Microsoft Studios User Research for 12 years.