Editors: Larry Rosenblum
and Simon Julier

Swordplay: Innovating Game Development through VR

Michael Katzourin, Daniel Ignatoff, Lincoln Quirk, Joseph J. LaViola Jr., and Odest Chadwicke Jenkins
Brown University

Despite the many revolutionary advancements in video game technology, the basic interface between the human and the gaming system has received relatively little attention. Over the past three decades, video games have been an important driving force for the advancement of real-time computer graphics, sound, and, more recently, artificial intelligence. For example, with each new generation of console game hardware, the graphics and sound provide greater realism and the AI is more formidable. Unfortunately, the same arguments cannot be made when it comes to the advancement of user interaction and gameplay. In fact, the interfaces in today's games have become increasingly complex and difficult to use given the myriad of buttons and analog controllers on today's gamepads. Thus, potential impacts for improving the gaming experience for both serious and casual users exist through research into better gameplay and interaction techniques.

Three-dimensional user interfaces are one technology class that can bridge the divide between gameplay and natural human expression.[1] For reasons of convenience and deployment, commercial video game systems often have artificially limited input devices, such as a keyboard and mouse or the standard gamepad. Given that many of today's games call for interaction in 3D worlds, such 2D input devices don't appear to be an optimal choice. However, there is a movement toward 3D input devices for gaming, spearheaded by devices such as Nintendo's Wii console and Sony's EyeToy camera. The Wii, in particular, explores a different form of input—the system tracks the acceleration of the controller (the Wiimote) in space. The game's interpretation of this acceleration data can have almost any meaning depending on the context: swinging a simulated tennis racket, casting a fishing line, or conducting an orchestra. In all these cases, players physically move the Wiimote in space with motions analogous to the real world. This 3D interaction style provides a natural mapping from human movement to gameplay controls we could not realize with traditional controllers.

Given this movement toward 3D input devices in gaming, we explored the use of 3D user interfaces in the context of *Swordplay*, an immersive sword-fighting and spell-casting game aimed toward casual gamers. Developed as part of a course on "Innovating Game Development" at Brown University, *Swordplay* is an attempt to determine what types of interaction techniques could be useful with natural human expression in virtual 3D environments. The result is a fun, lightning-fast game intended to push the skills and dexterity of its users, similar to the popular button-mashing games of the early 1980s.

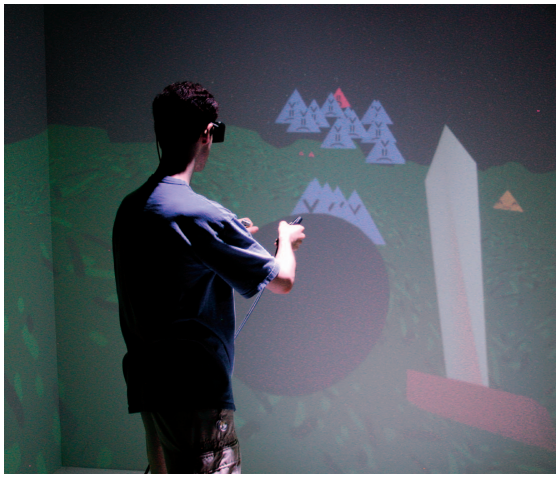
Swordplay

Swordplay is similar to arcade games in its style: the player tries to survive as long as possible using the game's first-person view (similar to ID Software's *Doom*) in a world with constantly spawning enemies (similar to Atari's *Gauntlet*). A limited number of hit points express the player's health, which diminishes as enemies attack. The player has a magic sword, a shield, and a longbow with infinite arrows with which to score points by vanquishing attacking enemies. There are three types of enemies to test the player's ability: fast/swarming, predictable/abundant, and unpredictable/explosive monsters. The player can cast magic spells to perform various special actions by drawing predefined symbols. These actions include various types of explosions, weapon enhancements, and healing. The ability to cast spells is limited by a recharging supply of energy that is consumed when spells are cast.

Swordplay is played in Brown University's surround screen VR system, an $8 \times 8 \times 8$-foot cube with three back-projected walls and a screen floor to display the virtual world (see Figure 1 on the next page). The system uses three 6-DOF trackers (position and orientation): one for head tracking to enable rendering of a stereoscopic display for the player's shutter glasses, and two for player controls (one each for the primary and secondary hands).

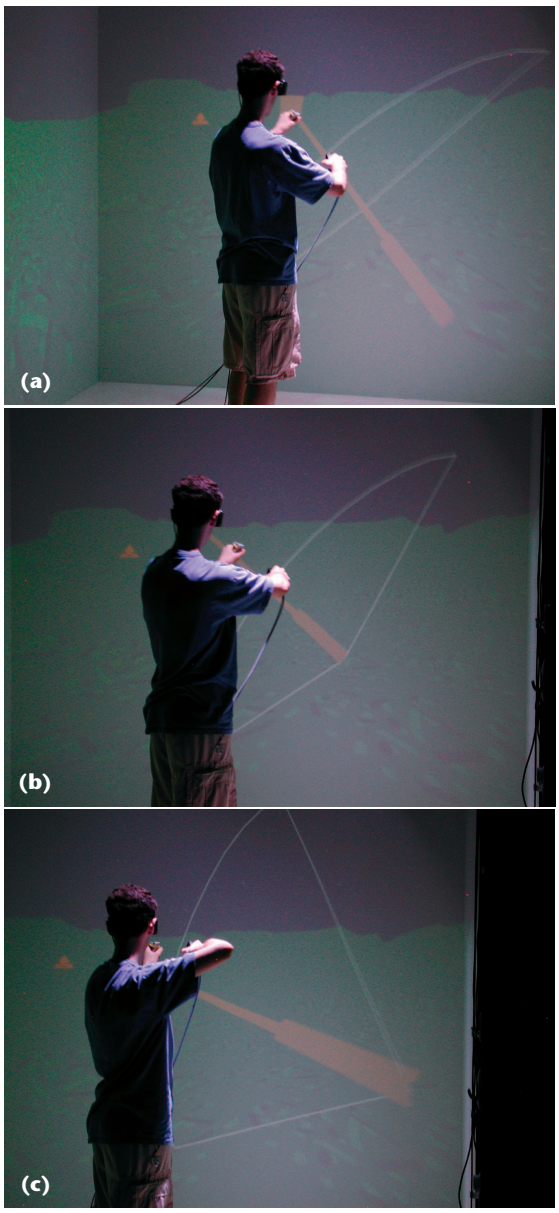
Interface

With *Swordplay*, we emphasized player control using natural human movement rather than artificial control mechanisms. With traditional input methods, the user is often required to remember tens of different button combinations that make little cognitive sense with their intended action. Instead of pressing buttons, the player's movement directly controls such natural actions as swinging the sword, firing an arrow, drawing spell



1 *Swordplay* **in Brown University's surround screen VR system. The player holds a sword and shield and prepares to fight off enemies.**

2 The player attacks several enemies with his sword.



3 A player (a) prepares to fire an arrow, (b) pulls the bow string back, and (c) aims at an enemy.

symbols, or dodging attacks. *Swordplay* only uses a few artificial controls: a single button for starting a spell or to set an arrow, and an analog stick for navigation.

Navigation

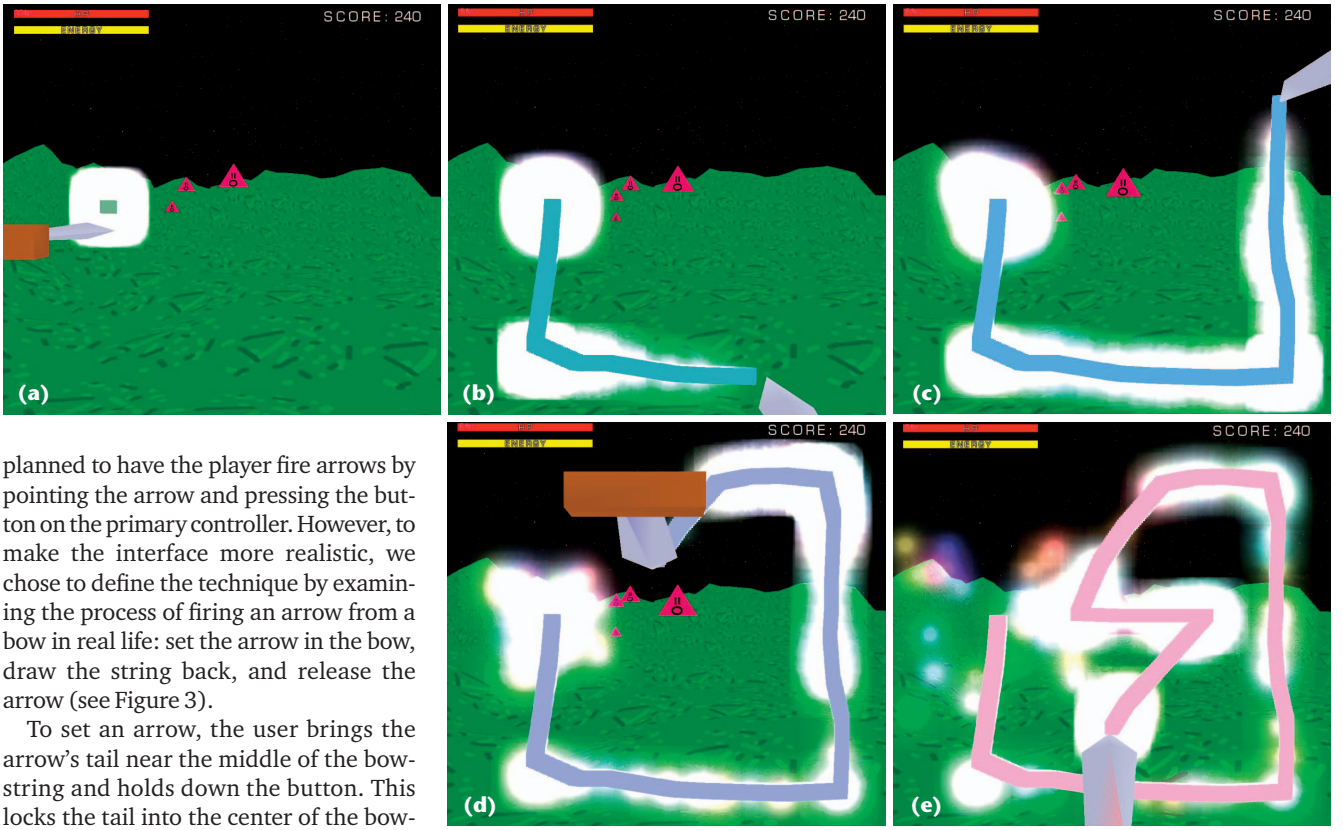
Players need to travel throughout the virtual world. We reviewed several possible schemes for movement within the world including leaning to move, turning the head momentarily to rotate the player's view, and scripted movement. We rejected scripted movement as it didn't fit the spirit of this game. Leaning and turning would be too difficult and confusing for control since players might want to look in a certain direction without moving or might lean inadvertently. Thus, we settled on a simple scheme for navigation: using an analog stick in concert with walking within the surround screen virtual environment. The analog stick moves and rotates the player within the world and is typically used for longer-range movements.

Wielding weapons

Players wield either a sword and shield or a bow and arrow, but cannot use them simultaneously. Thus, we needed a natural means to transition between these two modes. To avoid possible player confusion with an unnatural button sequence, we implemented the mode switch as a simple gesture for reaching over the player's shoulders. This gesture mimics the motion of drawing a weapon from its sheath and is similar in spirit to Mine's over-the-shoulder deletion technique.[2] We found that players easily recognized this approach and that it worked well without requiring an abstract button press.

During their use, the sword and shield are fixed to the position and orientation of their respective controllers. When the player moves or rotates a controller, the sword or shield move in the same manner. Players position the shield to block enemies and push them away, and swing the sword at enemies to damage them, dealing more damage with faster swings (see Figure 2).

In bow and arrow mode, the bow's grip is fixed in position relative to the secondary controller, and the primary controller holds the arrow's tail. Originally, we



4 A sequence of images showing a spell's evolution. The red and yellow rectangles in the upper left corner of each image show the player's hit points and spell energy.

planned to have the player fire arrows by pointing the arrow and pressing the button on the primary controller. However, to make the interface more realistic, we chose to define the technique by examining the process of firing an arrow from a bow in real life: set the arrow in the bow, draw the string back, and release the arrow (see Figure 3).

To set an arrow, the user brings the arrow's tail near the middle of the bowstring and holds down the button. This locks the tail into the center of the bowstring. To draw, the user pulls back the string by increasing the distance between the bow grip and arrow tail and aims at a target. To fire, the user releases the button making the arrow fly in the direction of the tail of the arrow toward the bow grip. The bowstring's tautness—that is, the distance between the bow's grip controller and the string controller—determines velocity. To get another arrow, a player reaches behind his head, mimicking the action of drawing an arrow from a quiver. This gesture is a one-handed version of the weapon switch gesture.

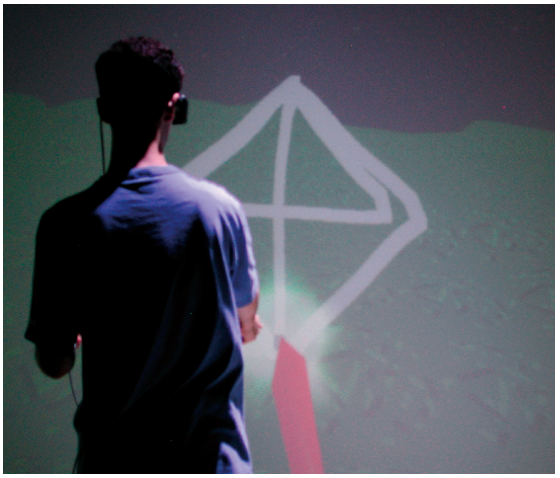
Spell casting

Many existing games let players perform both standard actions (for example, punch something or swing a sword) and special, supernatural variations on these actions (for example, perform an uppercut that explodes on contact, shoot a bouncing fireball to kill an enemy, or cast a destructive magical spell). In some games, users tell their avatars to perform one of these supernatural variants by entering some special combination of key presses. For example, to perform the exploding uppercut, the user might have to press the down key, followed by the forward key, and then the punch key. In other games, players cast a spell simply by pressing one button that is associated with that spell. With *Swordplay*, we were interested in the former approach, where the user provides a sequence of symbolic inputs to cast a particular spell.

The user casts a spell by holding down the button on the main controller in sword and shield mode and drawing a series of one of more symbols in the air (see Figure 4). One early design decision was that the user should not have to provide auxiliary input, indicating each symbol in the sequence was complete. Having the game determine the various symbols that were drawn in one continuous stroke would reduce interface complexity. Thus, *Swordplay*'s spell recognition system addresses two separate issues—the segmentation of the sword-tip positions over time into different symbols and the recognition of these symbols.

Many solutions exist for individual symbol recognition, and we explored several of them including using an open source optical character recognition program (GOCR; see http://jocr.sf.net), the statistical moments algorithm using *k*-nearest neighbors, feature-based gesture recognition, and simplified line-based gesture recognition. In the end, we decided to use the line-based approach for its simplicity and ease of implementation. The recognition system assumes that all shapes consist of straight lines that are horizontal, vertical, or at a 45-degree angle. The technique is relatively limiting in terms of what it can recognize, but proved to have good recognition accuracy for our particular problem domain.

The method transforms user input into an ordered list of vectors between each input point. Then it replaces each of these vectors with a new vector of the same length but along the closest of eight primary directions. This input categorization allows vectors in similar directions to be appended to form longer vectors used in recognition. The actual recognition process relies on a small database of exemplars (three to seven representatives for each symbol). The exemplars for each symbol are averaged and compared, using geometric



5 A player casts the multishot spell by drawing in the air with his sword. Once this spell is cast, the player can switch to the bow and arrow, and fire multiple arrows simultaneously.

distance, to each user input segment, resulting in a sequence of recognized shapes.

The language of the magical symbols in the game has a verb–adverb grammar: the game reacts with a simple spell when the user draws a single verb symbol representing some general type of spell (like a weapon enhancement spell, or explosion spell), and a more complex spell when the user appends extra symbols (that is, adverbs) to it. We were careful to keep the general effect of the adverb symbols consistent, to encourage the user to figure out new combinations experimentally.

Some spells have immediate effects in the game. For example, one spell damages all the nearby enemies. Other spells have more persistent effects, such as the multishot spell (see Figure 5), which split the arrows into three (or seven, if you append the correct adverb symbol) new arrows. To achieve persistence of some spell effects and to add strategy, the drawing the user makes to cast the spell is left in the game world for a short period of time, and is activated through collisions. Returning to the multishot example, the symbols used to cast the multishot spell remain in the air for about 15 seconds, and whenever it is hit by an arrow, it removes that arrow and adds three more in its place, each with a slightly different trajectory. Another spell, the inferno spell, also leaves its drawing in the game world for a few seconds. When the sword pierces the drawing in 3D space, a short spurt of triangles that explode and damage enemies on impact is fired along the sword's direction.

Results

We demonstrated *Swordplay* at Brown University's Arcade Day in May 2006. During the exhibition, approximately 50 people unfamiliar with the game interacted with the system. We observed how the users felt about their overall experience, specifically which parts of the game they did or didn't enjoy, and how quickly or easily users adapted to the control schemes. The sword and shield were the simplest parts of the interface to understand. Especially noteworthy was how quickly users understood this metaphor. However, user interaction with the bow and arrow was not as simple to learn. Some users were initially confused and needed a visual demonstration before understanding our instructions. However, no one took longer than two or three tries to grasp the function of the bow and arrow. Once players understood it, it received highly positive reactions, with one user exclaiming "I feel like Legolas!"

Users had no difficulty switching between sword and shield and bow and arrow modes. Users were often pleasantly surprised by how their movements would change weapons without pressing any buttons. They laughed or "wowed" in surprise when they brought their arms from behind their backs to see themselves holding different weapons.

Spell casting had mixed results. While players quickly understood how to cast spells, only a few users actually memorized the spells and used them regularly. This result was likely due to user inexperience with our symbol vocabulary, since we did not provide much in the way of verbal or visual demonstration. We are inclined to believe a proper introduction to the symbol vocabulary, similar to Harmonix's *Antigrav: EyeToy* game, would allow the user to cast spells fluently.

We were surprised with how positively our users reacted overall and with how friendly the game was toward people who weren't gamers or VR users. People who don't normally play video games or would consider themselves at most to be casual gamers were able to understand the game and controls within roughly five minutes.

Broader perspective

While a surround screen VR system is a deeply immersive environment for a video game, it is not practical for household use. The Nintendo Wii's Wiimote will allow next-generation games to use the same interaction techniques as *Swordplay*. Since the Wii has a similar input device as in our surround screen VR configuration, a *Swordplay* version could be made for the Wii.

We originally intended *Swordplay* to use physical simulation (employing the open dynamics engine) to increase physical plausibility and allow for emergent behavior. Due to computational limitations and issues with physical character control, we passed on the integration of physical simulation. With current advancements in faster physical simulation and practical character control, the integration of physics and physical avatars is on the horizon for *Swordplay*.[3]

Our spell recognition system was a good first attempt. One immediate improvement that could be realized is greater scalability in the number of symbols that can be recognized. The spell recognition could also be improved by modifying the core algorithms to handle curved lines, analyze the drawing as a 3D figure, and allow for drawing of symbols in any direction (currently users must draw at the front wall to simplify recognition). Although there is still a need for better symbol recognition methods, the current system used in *Swordplay* is functional for the casual gamer.

Swordplay demonstrates the potential for incorporating natural human movement into compelling games. To realize the full potential of gaming, innovation should

occur at the level of interfaces in addition to graphics and sound. *Swordplay* has taken one step toward innovation in this direction through the use of 3D user interfaces. ■

Acknowledgments

We thank Andrew Forsberg, Prabhat, and the Center for Computation and Visualization for their support and feedback. We thank the many guest speakers that visited our class for their insights and experiences.

References

1. D. Bowman et al., *3D User Interfaces: Theory and Practice*, Addison Wesley, 2004.
2. M. Mine, F.P. Brooks, and C. Sequin, "Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction," *Proc. Siggraph*, ACM Press, 1997, pp. 19-26.
3. P. Wrotek, O. Jenkins, and M. McGuire, "Dynamo: Dynamic Data-Driven Character Control with Adjustable Balance," *Proc. ACM Siggraph Sandbox Video Game Symp.*, CD-ROM, 2006.

Readers may contact author Joseph J. LaViola Jr. at jjl@cs.brown.edu.

Readers may contact the department editors at lrosenbl@nsf.gov and S.Julier@cs.ucl.ac.uk.