Water Ball Z: An Augmented Fighting Game Using Water as Tactile Feedback

Lode Hoste and Beat Signer
Web & Information Systems Engineering Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels, Belgium
{lhoste,bsigner}@vub.ac.be

ABSTRACT

Water Ball Z is a novel interactive two-player water game that allows kids and young adults to "fight" in a virtual world with actual physical feedback. The body movement of a player is captured via an RGB-D sensor and analysed by a 3D gesture recognition engine. In order to enable tactile feedback without the need for wearable devices, a number of water nozzles are positioned around each user's play area. The idea is to translate the input gesture of one player to the corresponding water spray hitting the other player. Besides severely reducing the risk of injury in a fight, Water Ball Z engages people in a real and fun experience where a hit is physically manifested via a water spray. Furthermore, power up moves and a live scoreboard extension bring the virtual world of Dragon Ball Z and Mortal Kombat cartoons into real (augmented) life. In addition to a detailed description of the physically augmented game, we discuss two new control parameters for mapping gesture input to haptic output which, to the best of our knowledge, are not present in existing 3D gesture recognition approaches.

Author Keywords

Gaming; haptic feedback; gestures; water; entertainment; augmented reality

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: Haptic I/O

INTRODUCTION

With the introduction of inexpensive controller-free sensors such as Microsoft's Kinect controller and new skeleton tracking algorithms, a wide variety of new applications became possible to build and explore. We present Water Ball Z, a free-air fighting game using water as a mean for tactile feedback. Water Ball Z is designed for entertainment purposes and could find a place at theme parks, fun fairs or subtropical swimming pools. The concept is loosely based on the popular Dragon Ball Z[1] cartoon where intense power up moves are key to defeat the enemy in a fight. Furthermore, computer games such as Mortal Kombat[2] illustrate how a virtual battle can look like between two players.

Virtual fighting games have always been popular. Since the introduction of the Microsoft Kinect, the transformation of the player into a powerful character has never been greater. However, one major component missing in free-air virtual fighting games is haptic feedback. In this paper, we introduce the use of multiple water nozzles to simulate the interaction with an opponent. The attacker can see the physical effects of their punches and kicks on their opponent and might be hit themselves if the opponent performs a counter attack. The haptic feedback makes the fighting game much more realistic than existing virtual counterparts.

We discuss the design of our Water Ball Z augmented fighting game and illustrate what kind of hardware can be used to construct such an augmented game using water as tactile feedback. Furthermore, we propose two new control parameters, *shoot-and-continue* and *holding-pose*, to control and simplify the gesture recognition process for the application developer. We start by presenting some related work and continue by describing the general setup and functionality offered by Water Ball Z. We proceed by providing detailed information about the necessary gesture recognition process as well as the hardware design. Last but not least, we outline future work and provide some concluding remarks.

BACKGROUND

Sodhi et al. [8] highlight that a physical feeling for virtual entities is a missing part in the emerging field of controller-free augmented reality. They introduce an air-based haptic technology called AIREAL to impart physical forces to a user in free air. Their solution generates an air vortex by aligning a number of subwoofers. However, the range from 8.5 centimetres up to one metre is not enough to be used in our fighting scenario. More traditional air compressors might represent an alternative solution for our application domain but have not been tested.

TEI'14, February 16–19 2014, Munich, Germany
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2635-3/14/02 ...$15.00.
http://dx.doi.org/10.1145/2540930.2540946.

[1] Dragon Ball Z: `http://www.dragonballz.com`

[2] Mortal Kombat: `http://www.mortalkombat.com`

Richter et al. [5] recently presented LiquiTouch as a mean for tactile feedback on touch surfaces. In their work, a nozzle mounted on a robot arm can be programmed to spray water with different pressure, frequency and rhythm. Unfortunately, the configuration of their system does not allow for large range water output. Additionally, different safety measures have to be taken in account when using much more water and a wider spray area.

The AquaTop display [3] uses speakers to agitate the water to form a tactile experience for tabletop displays. The water is mixed with white bath milk in order that a projector can display images on top of the water. The hands are tracked by a Microsoft Kinect sensor positioned above the surface. Based on the presented related work it is clear that new ways for providing haptic feedback are finding their way into interesting application domains.

Last but not least, Parés et al. [4] demonstrate a successful public installation of a water game for with they got very positive user feedback. In their installation, a camera was used to recognise people standing in a circle and holding their hands to trigger artistic fountains.

WATER BALL Z SETUP

Our proposed two-player Water Ball Z setup is illustrated in Figure 1. It consists of two circular play areas called battle stations. The size of these stations is defined by the range of the Kinect sensor denoted by (1) in Figure 1 and the water nozzle configuration (2). Each battle station requires one Kinect sensor tracking the movement of the player. The water nozzles are activated and deactivated by using solenoid valves, a wireless Arduino board and a dedicated computer running Linux (located near 3).

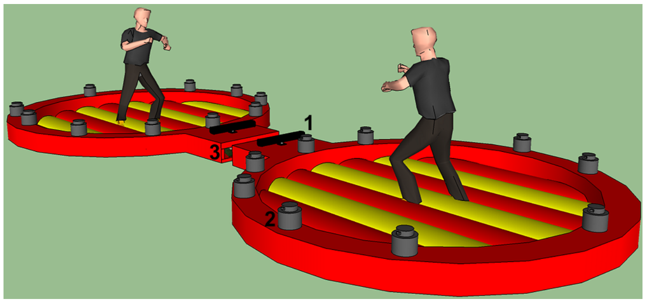

Figure 1. Water Ball Z setup

In our setup, the water sprayed out of the water nozzles shown in Figure 4 might easily reach the upper body of a player. For safety reasons, we opted for a setup where the solenoid module with the 12V electronics can operate wirelessly and fully disconnected from the electricity network by using a battery. The Kinect sensor and the computer responsible for the gesture recognition and wireless activation of water spays are protected against water by covering them with plexiglas.

We currently distinguish seven key input gestures, including a regular jab and a haymaker (wide punch) with the left and right arm, left and right foot kicks as well as a circular motion gesture. The last gesture is triggered by raising the hand to head level pointing up and performing a circular motion. All gestures result in a water spray on the opponent's battle station. For instance, a regular jab with the right hand will inflict a short water spray (100 ms) on the front left nozzle aiming to the head of the opponent. As the spray is rather short, a double jab will have the effect of spraying twice. Similarly, a haymaker is bound to the two sideway water nozzles at the opposite site. The foot kicks trigger a nozzle aiming to the back of the opponent. Finally, the circular motion gesture is a special gesture where all nozzles are enabled one-by-one in the same direction the gesture is performed. However, this action is only activated after one full circular motion has been performed. Note that we call this type of gesture a *shoot-and-continue* gesture.

Potential Extensions

Due to budget constraints our prototype is limited to the above specification. However, we foresee a number of extensions to the game:

Live Scoreboard

By providing a live scoreboard at the side of the battle stations, a real championship could be set up. A health bar showing the remaining life energy, similar to Mortal Kombat games, can be used and more complex attacking gestures could result in a large drop of the life energy shown in the health bar.

Power Up Moves

Several special gestures can be used to power up an attack. The Dragon Ball Z cartoon is famous for their power up moves and we think that they could further enrich the game. While player A can charge their next move by using a special pose, player B can launch any attack they want resulting in quite some loss of life energy for player A. On the other hand, player A is investing in a powered up attack that might cost player B a lot of energy in one hit. Visualising the progress of such a power up move could be done on the live scoreboard or by subsequently enabling the LEDs surrounding the battle station.

Defensive Gestures

The detection of defensive gestures could be useful to balance the scoreboard. In future work, we might experiment with defensive gestures to block the activation of water nozzles. We are not sure whether blocking the activation of a water nozzle is more engaging than the current format, where water is being hold off by the hands, arms and evasive moves.

Voice Input

A final extension to our game is the incorporation of multimodal input where a roar could result in a power up for the current move or a reduction of the opponent's spray time.

IMPLEMENTATION

The implementation of our Water Ball Z prototype touches three different domains, namely gesture recognition, electronic design and water plumbing. In the following we describe our work in these three domains in more detail.

Gesture Recognition

In order to detect different moves of a player, we use the existing Mudra [1] 3D gesture recognition engine. Mudra is the multimodal successor of the Midas [7] multi-touch framework and offers online gesture processing without explicit segmentation based on a gesture spotting approach [2]. The implicit segmentation of a real-time input data stream has two advantages. First, players are not forced to perform explicit segmentation poses and can therefore try to evade water sprays in many ways while trying to attack their opponent at the same time. Second, by using a precise declarative gesture definition, sprays will only be triggered when a valid move is executed. Declarative reasoning over movement in time makes the recognition system more robust to false positives. The sprays only trigger for the correct movements which improves the feeling of a real battle compared to a system that reacts to less focused movements. If necessary, the gesture rules can be modified to support more variation.

Listing 1. Circular motion gesture

```
 1 rule enter_circular_g0HandR0
 2    r = RelativeJoint { parent_joint == 12, child_joint == 15 }
 3    no { test matchEllipsoid(0.09754365, -0.45139461,
 4       0.00097420, 0.1444336, 0.5310212, 0.4756404,
 5       0, 0, 0, r.x, r.y, r.z) }
 6    test matchEllipsoid(0.09754365, -0.45139461,
 7       0.00097420, 0.1444336, 0.5310212, 0.4756404,
 8       0, 0, 0, r.previous_x, r.previous_y, r.previous_z)
 9    assert EnterEllipsoid {
10       id => ”cHandR0”, time=> r.time }
11 end
12
13 rule circularcw
14    e0 = EnterEllipsoid { id == ”cHandR0” }
15    e1 = EnterEllipsoid { id == ”cHandR1” }
16    e2 = EnterEllipsoid { id == ”cHandR2” }
17    e3 = EnterEllipsoid { id == ”cHandR3” }
18    test time:before(e0, e1, e2, e3, 1.5.seconds)
19    assert CircularCW {
20       start_time => e0.time, stop_time => e3.time }
21 end
22
23 rule(:circularcw).register(:sac, 500.ms) { |p|
24    spray(p / 100 ∗ sprays, 100.ms)
25 }
```

While developing Water Ball Z, we identified the necessity for three different types of 3D gestures: one-shot, shoot-and-continue and a holding pose. The *one-shot* gesture is a commonly used type of gesture, meaning that a single particular movement results in the corresponding single action. On the other hand, a *shoot-and-continue* (`sac`) means that a gesture should be matched at least once and that following movements which are in line with the gesture definition should be processed in an online manner. For instance, the gesture based on the circular motion of the hand requires one full rotation and from then on every consecutive online step in the gesture can trigger an action as defined in the rule provided in Listing 1. Finally, the *holding pose* type allows developers to register an event handler to a particular pose as part of a larger gesture. As long as the player holds the pose, a callback will be triggered every x milliseconds (with parameter x defined by the developer). For example, a user can hold their hands to form a virtual energy ball and when it is powerful enough (based on the elapsed time) they can unleash it towards their opponent by using a throw gesture.

A partial implementation of a clockwise (CW) circular motion gesture in Mudra is illustrated in Listing 1. The first few lines declare a rule that will be triggered whenever the right hand (id `15`) relative to the right shoulder (id `12`) enters a virtual ellipsoid. The coordinates (lines 3 to 8) are obtained by transforming, rotating and scaling the ellipsoid in a graphical editor [6]. A boolean `matchEllipsoid` function checks whether a relative joint falls inside the given ellipsoid coordinates. In case the conditions are met, on lines 9 and 10 we assert a new fact called `EnterEllipsoid` that represents a control point (characteristic point in the movement) and can be used in composite rules to form a gesture.

The following rule (lines 13 to 21) consists of a combination of four `EnterEllipsoid` control points to define a circular motion. The gesture is defined by a sequence of relative positions of the hand compared to the shoulder moving in a clockwise circle. To be more precise, line 14 to 17 expresses that a combination of four `EnterEllipsoid` facts (i.e. front, right, back and left) should be found. Further, line 18 describes that the control points should be matched in a sequential temporal relation with a maximum timeout parameter. A full implementation of the gesture set is available on our website[3].

Finally, the application logic can be added using a callback mechanism. In this case, the option `:sac` is used to express that the gesture should be completed at least once (i.e. shoot-and-continue). Subsequent intermediate steps of the gesture trigger the callback and make it behave like an online gesture. This results in a direct mapping between the location of the hand and the activation of the respective water nozzles installed at the battle station.

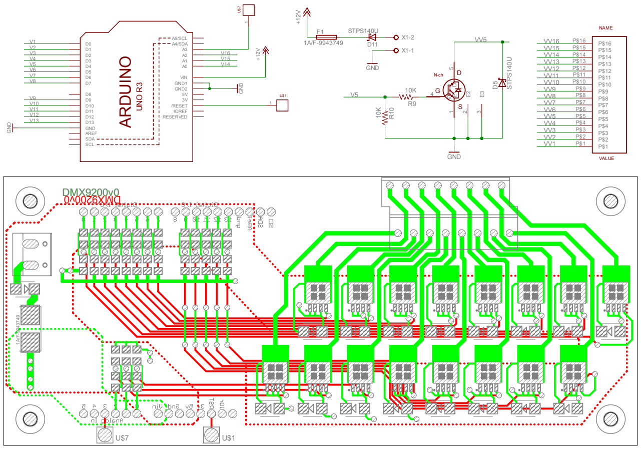

Figure 2. Electronic schematics

[3] Water Ball Z: **`http://wise.vub.ac.be/water-ball-z`**

Electronic Schema

We used a wireless XinoRF Arduino board (similar to an Arduino UNO R3 with wireless connectivity) to digitally interface the solenoid valves with our computer. However, the digital pins of an Arduino board do not provide the necessary power to control the solenoid valves. In order to solve this problem, we designed a PCB using N-channel power MOSFETs (40V/23mΩ) and 1N4001 Diodes. We can then draw 12V from the Arduino board using an external battery to couple it to the ten plastic water solenoid valves (12V-1/2" Nominal). Figure 2 shows the Water Ball Z electronic schematics. Furthermore, the complete Water Ball Z solenoid module is highlighted in Figure 3.

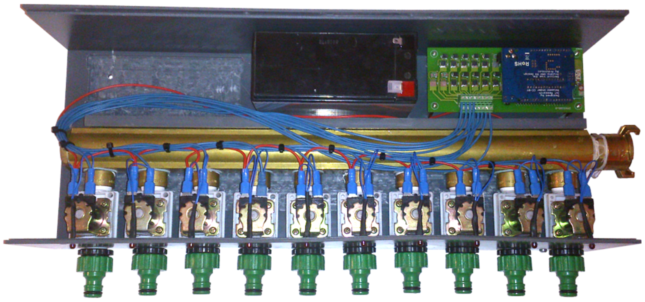

Figure 3. Water Ball Z solenoid module

Solenoid Valves and Nozzles

For our prototype, we used off-the-shelf gardening hoses in combination with plastic solenoid valves. The water input is split into multiple hoses and is then switched by the solenoid valves. The final output of each water hose is attached to a nozzle aiming at a player's front, head, back or other parts. An installation of our prototype is shown in Figure 4 and requires about 11 metres in length and 5 metres in width for a single battle station.

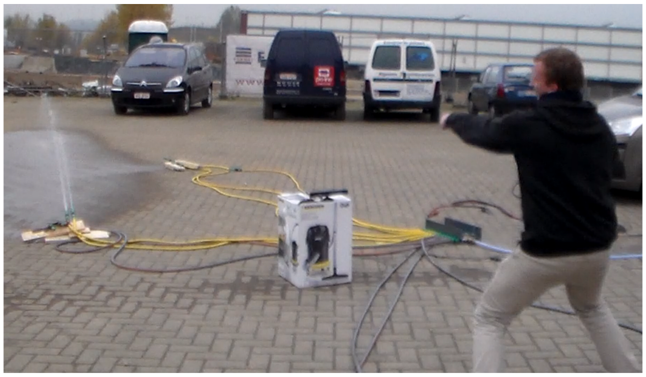

Figure 4. Tests with an early Water Ball Z prototype

Limitations

We currently use a Microsoft Kinect sensor which is not suitable for use in direct sunlight. However, more expensive models such as the Panasonic D-IMager EKL3106 can serve as a direct replacement for this hardware component. Further, in future work we plan to evaluate the effects of several options such as the live scoreboard, power up moves and blocking gestures.

CONCLUSION

We have provided an in-depth description of our novel augmented fighting game using body motion input and water sprays as haptic feedback. By using an RGB-D sensor for the tracking, there is no need for calibration or special tracking suits. In addition, the use of well-positioned water nozzles allows us to simulate the effect of a real hit without causing pain or injury to the player. Future work includes the experimentation and study of various options such as a live scoreboard, power up moves, defensive gestures or voice input. Further, we encourage gesture recognition approaches to incorporate our new shoot-and-continue and holding pose control parameters, offering software developers fine grained control over their action handlers. After various Water Ball Z sessions, we are confident that the presented ideas contribute to the blending of the virtual and the real world and that the presented water-based free air feedback can be applied to other games or application domains. Water Ball Z with its water-based tactile feedback might further stimulate novel interaction paradigms for augmented reality games and entertainment systems.

ACKNOWLEDGEMENTS

We would like to thank everybody who helped in realising Water Ball Z, in particular Christophe Scholliers, Hannes De Bondt, Wouter Heyse, Sandra Trullemans and Kevin Van Vaerenbergh. The work of Lode Hoste is funded by an IWT doctoral scholarship.

REFERENCES

1. L. Hoste, B. Dumas, and B. Signer. Mudra: A Unified Multimodal Interaction Framework. In *Proc. of ICMI 2011*, Alicante, Spain, November 2011.
2. L. Hoste, B. D. Rooms, and B. Signer. Declarative Gesture Spotting Using Inferred and Refined Control Points. In *Proc. of ICPRAM 2013*, Barcelona, Spain, February 2013.
3. Y. Matoba, Y. Takahashi, T. Tokui, S. Phuong, and H. Koike. AquaTop Display. In *Proc. of VRIC 2013*, Laval, France, March 2013.
4. N. Parés, J. Durany, and A. Carreras. Massive Flux Design for an Interactive Water Installation: Water Games. In *Proc. of ACE 2005*, Valencia, Spain, June 2005.
5. H. Richter, F. Manke, and M. Seror. LiquiTouch: Liquid as a Medium for Versatile Tactile Feedback on Touch Surfaces. In *Proc. of TEI 2013*, Barcelona, Spain, February 2013.
6. B. D. Rooms. VolTra: A Development Environment for Extracting and Defining 3D Full-Body Gestures. Master's thesis, Vrije Universiteit Brussel, 2012.
7. C. Scholliers, L. Hoste, B. Signer, and W. De Meuter. Midas: A Declarative Multi-Touch Interaction Framework. In *Proc. of TEI 2011*, Funchal, Portugal, January 2011.
8. R. Sodhi, I. Poupyrev, M. Glisson, and A. Israr. AIREAL: Interactive Tactile Experiences in Free Air. In *Proc. of SIGGRAPH 2013*, Anaheim, USA, July 2013.