# Spooky Dodgeball at a Distance

Devin Wilson
devinwilson@gatech.edu

Albith Delgado
albith@gatech.edu

Digital Media Program
Georgia Institute of Technology
Atlanta, Georgia, USA

## ABSTRACT

Spooky Dodgeball at a Distance (or SDAAD for short) is a local multi player electronic game of virtual dodgeball without graphics. Two Players use a Neurosky MindWave headset, a low-cost EEG biosensor, and a PlayStation Move controller to defend and attack in a virtual game of dodgeball. The project is an exploration of the gameplay possibilities that these two different devices could offer when used together. We're sharing 1) the setup and configuration guidelines, 2) a record of our iterative design process, 3) feedback from our testing sessions, and 4) the insights derived from these activities as design research.

## 1. INTRODUCTION

To set up the game, players need the following hardware devices: two Neurosky MindWave Mobile headsets, two PlayStation Move Controllers, and two Macintosh computers with Bluetooth functionality [1]. To run the game's source code, players must download the Processing Development Environment and a set of libraries to interface with the Neurosky and PlayStation Move devices [2]. Networking is handled with oscP5, an Open Sound Control implementation for Processing [4]. With the PS Move API, the PlayStation Move Controllers can be accessed and paired to the computers via Bluetooth [3]. One computer functions as a client and a server to the other computer's client.

## 2. HOW TO PLAY

### 2.1 Goal

In the game, players take turns throwing a virtual ball to their opponent. Successful throws break down the opponent's defense, leaving them vulnerable to the next attack by their opponent. The ultimate goal of the game is to hit your unshielded opponent with the virtual ball. Feedback in the game comes from audio cues from the server computer, visual feedback via the controllers' LEDs, and vibration from the PlayStation Move Controller.

### 2.2 Before the match

Before the match begins, players first put on their headsets, ready their devices and perform a face-off to determine who begins the game with possession of the ball. The players wait a randomly generated amount of time for their LED's light to turn white. The first person to press the controller's "Move" button after this obtains possession of the ball. Press the Move button too early or too late, and possession is given to the opposing player.

### 2.3 During the match

Ball possession is indicated by the player's LED shining blue during the match. A red LED indicates that the player does not have the ball. Players can perform several actions in the game: they can focus to raise their "attention" level, relax to raise their "meditation" level, or throw the virtual ball. The MindWave Mobile device affords the first two actions. The device calculates these values from the wearer's EEG waves. Tricks for raising the "attention" level are to read text, count, or perform mental math. To raise the "meditation" level, players may close their eyes or take deep, steady breaths. When the meditation level is high enough, a player's shield engages, which is indicated by a steady light. Conversely, a flashing light indicates that the player has no shield.

**Table 1. Player's Game State, according to PS Move's LED light**

| *If Player Has:* | *Ball* | *No Ball* |
|---|---|---|
| *Shield* | LED Steady Blue | LED Flashing Blue |
| *No Shield* | LED Steady Red | LED Flashing Red |

Players can throw the ball by swinging the PS Move Controller. A throw that is successful knocks the other player's shield off, or—if the hit player is unshielded—the thrower wins the game. A player win is indicated by a green light on controller LED. An unsuccessful throw results in a missed shot. Any throw, with the exception of a winning throw, always results in the opposing player gaining possession of the ball.

### 2.4 Accuracy Modes

The game can be played with two accuracy models to choose from: threshold-based or probability-based.

In threshold-based accuracy, the players' controllers' vibration level is either at maximum intensity or entirely off, depending on if a throw will hit or not (respectively).

In probability-based accuracy, the intensity of the controller's vibration maps to the player's attention level. The higher the player's attention, the more intense the vibration, and the better chance of a successful throw. A successful hit is based on a 100-sided die roll. If the resulting number is less than the player's attention level when attacking, the hit is successful.

## 3. CONFIGURATION OPTIONS

The game's main variables, described in the online documentation, can be configured in Processing to fit different kinds of play styles. It can be played by a single player, against a simple, computer-controlled opponent. The face-off stage of the game can be turned off completely or its maximum and minimum delay times reconfigured. The game can be configured so that

players begin with or without a shield. The accuracy models described earlier can be selected as well. Finally, players can edit the minimum gesture strength required to register a throw with the PlayStation Move Controller, as well as the Meditation and Attention Threshold values to recover player shields or to make successful hits.

## 4. ONLINE DOCUMENTATION

The project website **spookydodgeball.wordpress.com** contains the documentation, a gameplay video and links to the software needed to run this game. In addition, a record of our design iterations and insights, an explanation of the setup, configuration options and links to the project's source code repository are provided [5].

## 5. FINDINGS

Throughout this design process, we discovered the potentials and limitations that the Neurosky MindWave and the Playstation Move Controller provide game designers. The use of both of these devices brings some challenges and interesting insights. We found that despite the variable levels of accuracy and sensitivity that this inexpensive EEG reader provided, playtesters welcomed the potential to use the brain as a game input. We also learned to adapt our design to the constraints of devices like the Neurosky MindWave, and the affordances of the Playstation Move Controller. One of the more surprising constraints we discovered is that the MindWave's software was designed to be used by a single headset per computer. To work around this limitation, our game requires that the each of the two headsets be paired to one Apple computer, with the computers then being networked together. In addition, the low rate of EEG signal sampling made the 'attention' and 'meditation' levels update slowly. Moreover, these two values had an inverse relationship. Players had to take time to increase their meditation or attention values before executing an attack. We decided to encourage a slower, more methodical play style by having players exchange possession of the ball after every attack, and by setting threshold values that required effort from the player to reach them. Since throwing the virtual ball as determined by brain sensor readings would have proven inaccurate, we decided to have players attack with the PS Move Controller. Thus we capitalized on the affordances of both devices. Moreover, the novelty for players to be able to use their brainwaves in gameplay is still very high. We hope that this work will encourage other game designers to experiment with the unusual methods of input and output that we employed in this project, using the devices either together or separately in order to expand the boundaries of what digital games can be imagined as.

## 6. ACKNOWLEDGMENTS

We would like to thank Russell J. Huffman and Jessica Anderson for their vital participation and feedback during our early playtesting sessions.

## 7. REFERENCES

[1] MindWave Mobile: Brainwave Starter Kit: 2015. *http://store.neurosky.com/products/brainwave-starter-kit*. Accessed: 2015-1-30.

[2] Processing.org: 2015. *https://processing.org/*. Accessed: 2015-1-15.

[3] PS Move API (thp.io): 2012. *http://thp.io/2010/psmove/*. Accessed: 2015-1-28.

[4] Andreas Schlegel – oscP5: 2011. *http://thp.io/2010/psmove/*. Accessed: 2015-1-25.

[5] SDAAD_game-Github: 2015. *https://github.com/Albith/SDAAD_game*. Accessed: 2015-2-15.