Tap-Kick-Click: Foot Interaction for a Standing Desk

William Saunders, Daniel Vogel
Cheriton School of Computer Science, University of Waterloo, CANADA
william@williamsaunders.net, dvogel@uwaterloo.ca

ABSTRACT
Foot interaction techniques for controlling conventional desktop applications at a standing desk are described. Indirect, discrete two-foot input using combinations of spatial kicks, taps, jumps, and standing postures are tracked using a depth camera and instrumented shoes. An implemented system shows how visual feedback and interface augmentation can make foot input compatible with existing desktop applications. Application scenarios using the system demonstrate productive pure foot input breaks with real application tasks like web browsing and code debugging, as well as using feet as a secondary input channel with mouse and keyboard. An evaluation validates the usability of the approach.

Author Keywords
interaction technique; foot input; healthy computing;

ACM Classification Keywords
H.5.2. Information Interfaces (e.g., HCI): Input devices

INTRODUCTION
Using a computer while standing at an elevated desk can increase health and productivity [10]. This is largely due to a standing posture and increased physical movement. To enhance this benefit, we created *Tap-Kick-Click*, foot interaction techniques for a standing desk to enable productive "foot input only" breaks (Figure 1a) and increased physical activity by using foot input with mouse and keyboard input (Figure 1a). We build on Meyers et al.'s [16] foot-based exercise system to sort emails and photos on a large display, and we extend, realize, and validate our previous hypothetical illustration of foot input at standing desk [25].

Tap-Kick-Click primarily uses an interaction vocabulary of discrete taps and kicks aimed at virtual targets arranged in a semicircular array around each foot. Low-density targets are used for application commands (injected as keyboard shortcuts or simulated target selection) under eyes-free usage and high-density targets with indirect "foot cursors" for menu selection. Small jumps access help and, as a further probe, mildly uncomfortable foot positions are used to self-control "cyberslacking" [31] (Figure 1c). Our implemented system shows how carefully designed feedback and interface augmentations can make foot input compatible with existing desktop applications.

DIS 2016, June 04-08, 2016, Brisbane, QLD, Australia © 2016 ACM.
ISBN 978-1-4503-4031-1/16/06…$15.00 DOI:
http://dx.doi.org/10.1145/2901790.2901815

Consider a *Tap-Kick-Click* usage scenario for academic research (also demonstrated in the accompanying video):

While standing at her desk, Jane enters a paper search term with the keyboard, then lifts her hands to stretch while scrolling search results with forward and backward toe taps. She continues working "away from the keyboard" by kicking forward to enter "click mode" where visible hyperlinks are decorated with icons to convey short sequences of forward, side, and back taps to select them. Jane selects a paper link by performing a sequence, reads the abstract, and with a backward kick, adds the paper to her reference manager. She switches to her reference manager using a right kick and forward left toe tap, then opens the downloaded PDF with a forward kick. Jane skims the PDF while scrolling with her feet like the webpage. Having taken a short physical break, she reaches again for the keyboard to enter notes. While typing, her music player starts playing an annoying song, so she skips it "in the background" with a forward whole foot tap. Having accomplished some research, Jane decides to check Facebook. To help reduce procrastination, she configured her system so she has to stand in a mildly uncomfortable position while viewing certain sites. It is just enough to deter her from spending too much time on it and Jane returns shortly to her work.

Our work leverages research in foot input, sensing, and break software to design, implement, and validate a foot-based interaction technique in a fully working system for a new and thought-provoking application.

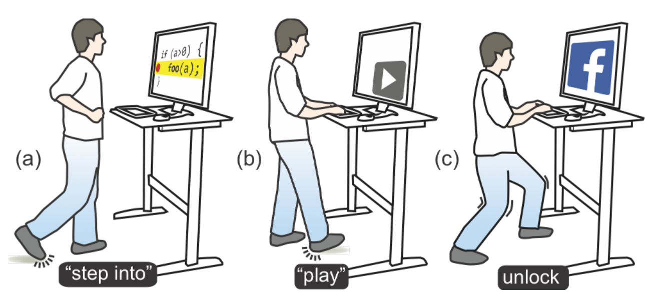

Figure 1. Tap-Kick-Click interaction enables: (a) physically active and productive "foot input only" breaks; (b) increased physical activity using foot input *with* mouse and keyboard input; (c) and, as a way to self-control cyberslacking.

To our knowledge, no previous work has combined foot input with a standing desk for command-level control of conventional desktop applications. We make five contributions:

- A set of indirect, discrete foot interaction techniques for a standing desk that enables primary foot-only application control and secondary foot input augmentation of mouse and keyboard.
- Feedback visualizations and command mappings to make foot input practical with real desktop applications.
- A tracking system using an under-desk depth-camera and shoes with IR LEDs and force sensors.
- Demonstrating how Tap-Kick-Click can be used for web browsing, academic research, and interactive debugging.
- A usability evaluation validating the system and interaction techniques applied to real application tasks.

After surveying related work, we describe the interaction vocabulary, provide implementation details for the system, illustrate how the vocabulary is applied to example applications, and present the results of a usability evaluation.

RELATED WORK

Past work motivates using standing foot input for physical activity rather than increased performance, previous foot-based systems in other domains like mobile and large displays provide inspiration for input methods, and studies examining standing foot input provide guidelines.

Foot motion follows Fitts' law, but it is slower than comparable arm movements [7,12,29] making foot-based cursor control slower and more-error prone than a hand-operated mouse [20] or trackball [19]. Although footswitches are used for audio transcription [21], using feet for desktop applications is rare for able-bodied users [6,14], especially when standing. Velloso et al. [30] provide a comprehensive survey of foot-based interaction, we highlight the most relevant work for foot input while standing.

Active Foot Input

Although foot input may be slow, feet can make interaction physically active. Foot-based games such as Dance Dance Revolution (DDR) [13] are strenuous and skilled, but exergaming systems [33] and mobile games [18] have used less strenuous taps, jumps, and kicks for mild exercise, immersion, and enjoyment [10].

These kinds of physically active movements can also be used to perform conventional computing tasks. For example, foot input can provide a break from mouse and keyboard to reduce wrist RSI, such as Berque et al.'s [3] extra foot operated mouse to control a game and scroll a Twitter feed. However, given poor foot-based cursor control while seated [19–21], discrete command invocation is likely more suitable. In their work-in-progress, Meyers et al. [16] use foot taps on a 3×3 DDR foot switch mat to make tasks like email sorting more active and enjoyable. A DDR mat uses direct input, so gaze shifts between display and monitoring foot position, and the input space is limited to about 8 to 10 single movement actions: this will not scale easily to controlling real applications. Given issues with continuous foot control and the encouraging results of Meyer et al.'s initial exploration of discrete control, our interaction vocabulary is discrete.

Subtle Foot Input

A counter-point to increasing physical activity are systems designed to be subtle and socially acceptable [23]. Crossan et al. [5] use in place toe taps for an eyes-free menu and Scott et al. [27] explore single foot, heel and toe pivoting gestures following a double toe tap demarcation. We extend in place tapping by adding a spatial dimension of virtual floor targets and by using different kinds of taps. Even more subtle input is possible with pressure. Schöning et al. [26] sensed subtle weight shifts with a Wii balance board for large display map navigation. Matthies et al.'s insoles also sense small toe movements [15], an idea expanded by Fukahori et al.'s pressure sensing socks to perform secret tasks like password entry [8]. Our goal is not to use subtle movements like toe or heel pivots since they are less physically active. Moreover, they are difficult to distinguish from shifting feet when standing and are more likely to cause false positive inputs at a standing desk. We use insole force sensors and spatial targets to make tapping more expressive and robust.

Guidelines for Standing Foot Input

Alexander et al.'s [1] elicitation study suggests additional foot gestures, including spatial taps and kicks for controlling continuous map navigation, a more specialized goal than general command invocation. We justify some design choices using this work, but we extend their elicited mappings to desktop applications and to a larger input space of spatial tapping and kicking using both feet. Alexander et al. tested recognition of dominant foot kicks in four directions and forward taps in their system implementation, and only applied on kicks and in-place taps for map navigation. Han et al. [11] develop guidelines for discrete, directional kicks: people can reliably kick in 5 forward directions over a 120° arc (24° targets) and reliably control two levels of kick velocity. Our previous work evaluating indirect foot pointing in a controlled experiment [25] provides expanded guidelines supporting the use of both feet, the near equivalence of tapping and kicking, tap and kick direction preferences, and recommended target sizes. Velloso et al.'s [29] study of foot input while seated confirms many of our findings as well. We apply these guidelines to our design.

TAP-KICK-CLICK INTERACTION

The core interaction vocabulary is a sequence of taps and kicks performed with either foot, aimed at virtual targets in a semicircular array around each foot. Alexander et al.'s study [1] found people intuitively use this kind of tapping and kicking. We supplement the core vocabulary with jumps and static foot positions for specialized applications.

This vocabulary and system are powerful enough to perform useful application tasks such as web browsing, reading, and stepping through code with an interactive debugger. The combination of a standing desk and standard desktop applications means foot input is easily interleaved with mouse and keyboard. Feet are not used for rapid or precise tasks like text entry or drawing; the goal is to provide enough capability to enable physical breaks away from the mouse and keyboard and create opportunities for increased physical activity by using feet for background tasks.

Since foot input is always available, our system enables a physical break without leaving a work task. This is a middle ground between intense task focus without any kind of break and completely off-task physical and mental breaks enabled by systems like SuperBreak [17]. There is evidence that off-task breaks may not increase productivity [24] and our own informal observations suggest workers do not always want to take an off-task break when "on a roll."

Interaction with Spatial Taps and Kicks

We use a combination of foot action and target selection as a discretized input language.

Indirect Target Selection

The center-front of each foot is the selection hotspot, represented on the display as *foot cursors* within a virtual target layout (Figure 2b). This is *indirect* input control where the user can focus on the display rather than their feet. A DDR mat or Augsten et al.'s interactive floor [2] use *direct* control where the user looks at their feet relative to actual floor targets. An indirect cursor also eliminates foot occlusion and minimizes hotspot ambiguity [2].

Targets are selected with slow-speed kicks and three types of taps: toe (front of the foot), heel (back of the foot), and whole foot (both front and heel) (Figure 4). In our previous study [25], we found kicks or taps to be nearly equivalent and people naturally perform these different taps with some preference for toe tapping. Taps are detected using heel and/or toe floor contact and kicks are identified when and where the foot reverses direction.

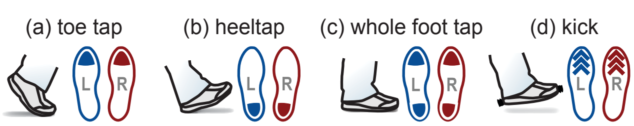

Figure 4. Foot actions with foot icons used in the interface.

"Midas-Step" Avoidance

There is no explicit mode switch to turn foot input on or off since our aim to make foot input always available to support spontaneous foot breaks and background foot input. We accomplish this with a simple counter-measure to avoid accidental foot input. Between each tap and kick, the foot typically returns to a home position to maintain balance and encourage physical activity. The diameter of each home position in motor space is 10 cm and it is dynamically adjusted to match the standing position by following short, slow foot movements (typical when adjusting stance). Combining a

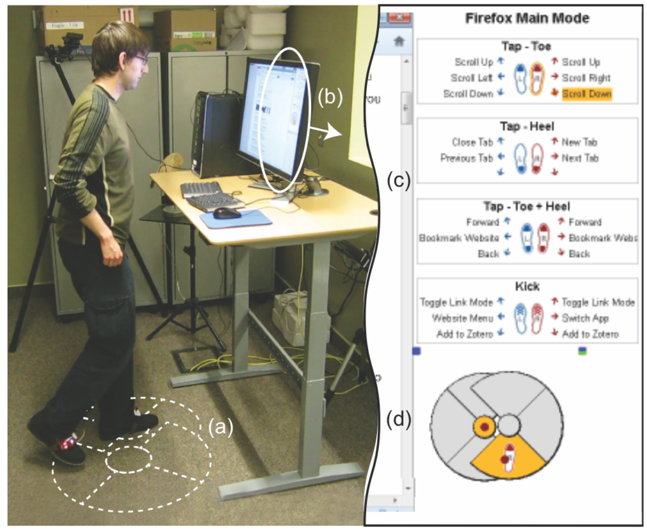

Figure 2. Indirect foot input system showing: (a) low-density virtual targets; (b) indirect feedback in always-visible sidebar with (c) cue card showing command mapping for foot action and target (d) foot cursors within virtual targets (left foot is in home position, right foot has just completed a back toe tap).

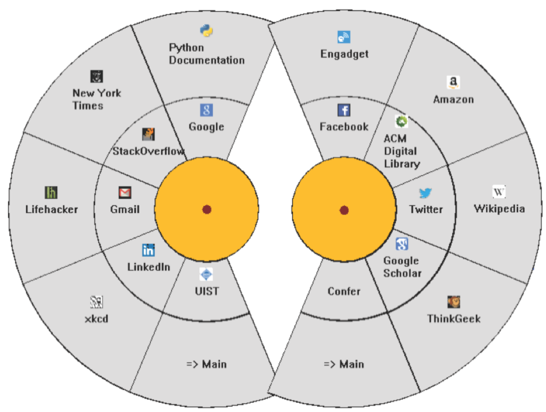

Figure 3. High density virtual targets used for foot menu.

dynamic home position and distinctive spatial tap and kick movements reduce "Midas step" inadvertent activation.

Virtual Target Layouts

Two virtual target layouts are used: low- and high-density. The *low-density layout* is for issuing application commands and selecting GUI targets with a special "click mode." With some practice, this can be done *eyes-free,* without monitoring the foot cursor allowing complete focus on the application under control. Following guidelines [25], low-density targets have an angular size of 90º and are positioned forward, outward, and backward from each foot (Figure 2a).

The *high-density layout* is used for a special-purpose foot menu where the user can fully rely on indirect feedback to select among many simultaneous actions (Figure 3). Following guidelines [25], the high-density targets have an angular size of 45º and are positioned in a contiguous outward arc from front to back of each foot in two bands, one from 10 to 20 cm and the other 20 to 30 cm.

Application Commands with Low-Density Targets

A constantly available side panel displays a foot cursor with virtual target positions (Figure 2d) and a dynamic cue card showing current foot action to command mappings (Figure 2c). The active command set matches the mode of the focused application. Each time a tap or kick is sensed, the corresponding command is sent to the application by injecting a keyboard shortcut key sequence. The most recently sensed foot action is displayed using a foot-action icon (see Figure 4) at the target location and the cue card highlights the command issued. Rich feedback is important to convey detected events and current state to spot and troubleshoot errors (e.g. a background object disrupting foot tracking) and it provides a way to learn mappings between foot actions and application commands.

The side panel does reduce space for other applications[1]. With larger displays, or multiple displays, this is less of a concern. The panel design could be refined to minimize its size, or it could be made semi transparent and become active when foot input is detected. We also plan to experiment with augmenting the system cursor with a subset of information shown in the side bar for expert users.

Command Mapping Patterns

With 4 foot actions (toe tap, heel tap, whole foot tap, kick) and 6 virtual target locations across both feet, 24 commands can be accessed in one set (examples of command sets in Figure 8). Where possible, we map frequent commands to forward targets and favour toe taps [25]. However, highly correlated mappings overrule this guideline: we initially assigned scroll-down to a forward tap based on frequency, but pilots revealed strong dislike for this mapping. Alexander et al. [1] note the importance of correlated mappings.

We distribute frequent commands across both feet for comfort and balance and map the most frequent commands, such as scroll down, to both feet so feet and be alternated. Backward heeltaps are difficult [25], so we avoid them, or use them for infrequent and irreversible commands (e.g. "delete all"). Tapping actions are "auto-repeated" when the foot is held on a target for a period of time (after 150ms, the action repeats every 400ms). This can be particularity useful for scrolling or stepping through code.

Kicks are used to change application modes. For example, in a code editor, interactive code debugging (and the corresponding debugging command set) is activated with a forward kick. Across all applications, an outward right-foot kick changes to a global command set for application switching and background application control. For example, toe taps are for application task switching (analogous to "alt-tab")

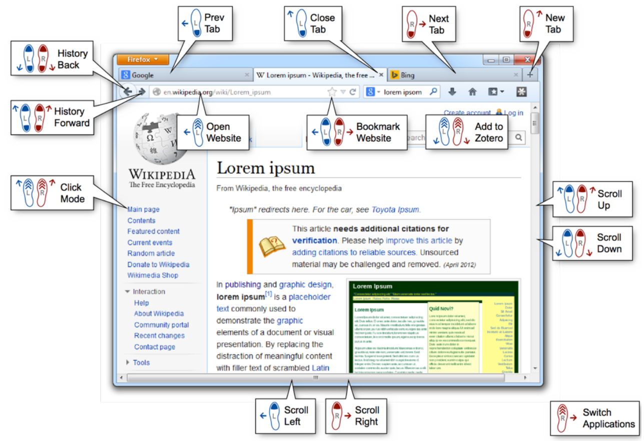

Figure 5. Help screen showing foot action to command mappings in the context of the application interface.

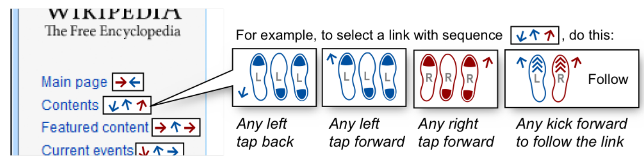

Figure 6. *Click mode* selection sequences (the right callout is only shown in help mode).

and other taps for background application control (e.g. a right whole foot tap sends 'play' to a music player and a right whole foot tap up skips to the next track).

"Clicking" on GUI Targets with Low-Density Targets

There are many cases in GUI applications where arbitrary targets must be selected and there is no direct keyboard mapping (e.g. webpage hyperlinks). Given the poor performance of foot-controlled mice and our focus on discrete input, we developed a technique called "click mode" where GUI targets are selected with short sequences of taps. Once activated with a forward kick, click mode decorates each target with icons representing a unique sequence of forward, outward, and backward taps using left or right feet. Since sequence lengths vary, we use a forward kick to accept the entered sequence and "click" on the target.

We implemented this in Firefox using a modified version of the "Hit-a-Hint" add-on. Sequences are displayed beside each link in the visible portion of the page as strips of 16px arrow icons (Figure 6). This scheme scales well, sequences up to length 3 can index 258 targets ($6 + 6^2 + 6^3$). With dense targets, the arrows occlude content, but click mode is usually activated after the desired target is visually identified. This technique could be extended to targets in any GUI application using accessibility APIs for target locations. Click mode

[1] Note the panel size is larger in the accompanying video to increase legibility in the video only.

is not as fast as using a mouse, but it is easy to learn and requires fun combinations of steps that fulfill our goal of physical activity.

Foot Menu with High-Density Targets

The high-density layout is used for a special-purpose foot menu (Figure 3). In the web browser command set, a left kick opens the menu of bookmarks. The high-density virtual targets forming the menu, and the foot cursor feedback, are displayed in the center of the screen. Each target is labelled with a menu action. With 4 foot actions and 20 targets, 80 menu items are possible, although we use 20 commands for ease of understanding.

In Place Jump for Contextual Help

Meyers et al. [16] report people found two foot jumps enjoyable, but using them for core application control is too physically demanding. Augsten at al. [2] test jumping as a target selection technique on a direct input floor, we integrate it into a vocabulary in a different input context. We use small in place two-foot jumps to activate a help system. The jumping action is intended to be easy to remember and encourage learning (avoiding exertion by consulting help too frequently). Help is provided in the form of command mappings in the context of the application interface (Figure 5). Foot action icons are shown in callouts pointing to the equivalent application command location when possible.

Static Foot Positions for Self-Regulating Cyberslacking

Static, discrete foot positions can be used as a kinaesthetic mode to unlock websites like Facebook or YouTube that can be a source of "cyberslacking" [31]. We use a mildly uncomfortable posture resembling a lunge with one foot forward and the other back (Figure 1c). The foot positions can be switched at any time.

We imagine this as "self-help" where the user chooses to block websites. The lunge posture is purposely chosen to be extreme. More natural postures could be used for other tasks, for example crossing one foot behind the other to switch between two virtual desktops. Poses and mappings must be chosen carefully to avoid false positives.

Error Recovery

An incorrect foot movement, a sensing error, or a cognitive mistake, could result in the invocation of an undesired command. Erroneous navigation commands, such as scrolling and link selection, can be reversed using additional foot input commands or by reaching for the keyboard or mouse. Most erroneous application commands can be "undone" using an application's built-in undo function. Currently, undo is not mapped to foot input, so this requires the use of the keyboard. Given the standing desk context, reaching for the keyboard or mouse should not be a significant burden.

SYSTEM

The Tap-Kick-Click interaction vocabulary is fully implemented to work with existing Windows desktop applications. The system handles foot input tracking using a depth camera and insole pressure sensors, as well as managing the side bar

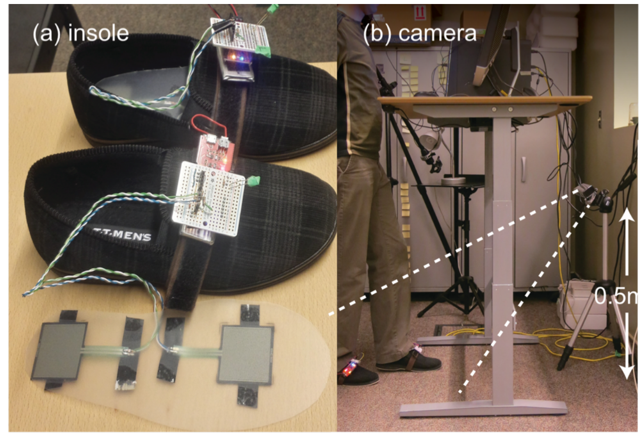

Figure 7. Sensing hardware: (a) insole with force sensors; (b) under desk Kinect depth camera for foot position tracking.

display, command mode selection, command injection, and application augmentation such as decorating links with icons for "click mode."

Sensing Foot Input

Using instrumented shoes with a depth camera makes sensing foot position and foot tap quite robust. This enabled us to focus on interaction and also minimized a confounding effect of tracking errors from our usability study.

Foot Position

To be practical for small offices, a Microsoft Kinect depth camera is mounted behind the standing desk with a clear view of the feet and lower leg (Figure 7b). Full skeletal tracking requires a less convenient mounting location and does not provide accurate foot positions. Simple depth image thresholding can identify people by their shoes [22], or track directional foot movements when seated at a desk where occlusion is not an issue [28]. For increased precision and to mitigate partial-occlusion during backward movements, we mounted 850nm IR LEDs on two slippers to locate each foot. Tracking fixed points also makes the algorithm robust to changes in foot orientation, characteristic when moving to side targets [25].

The depth of each foot is calculated using an algorithm implemented with SimpleCV with the following steps:

1. The two brightest points (the IR LEDs) are located in a downsampled 160×120px IR frame using two passes with a 3×3 25-percentile filter.
2. Using a downsampled 320×240px depth frame, the 25th percentile of depth is computed in the region surrounding each point masked by a bottom half-ring shape (inner radius 5px, outer radius 15px). This shape avoids the depth "hole" caused by a bright IR LED.
3. The depth sample is used to find the 3D location of the foot and then the 2D position by projecting down along the normal of the calibrated floor plane.

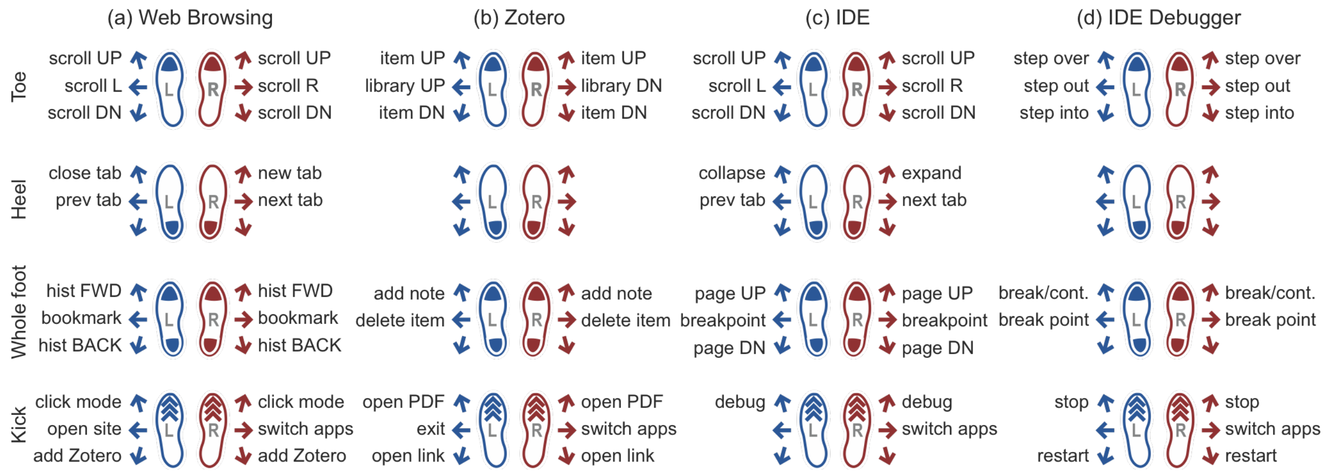

Figure 8. Example application command sets with foot action mappings represented as sidebar cue cards.

Foot Contact

Force sensitive resistors are mounted at either end of each insole to accurately differentiate between heel, toe, and whole foot tapping, between kicking from tapping, and precisely sense repeated tapping actions (Figure 7a). The sensors for each foot (and IR LED) are connected to a battery-powered Arduino and sensor readings are sent over Bluetooth to the computer system

Sensor Fusion and Action Sensing

The Arduino force and Kinect position streams are time stamped and synchronized. Commands are sent when a foot action and target position combination is sensed over a 200ms time window. Kicks are classified as reversing motions over a target without any floor contact. Additional constraints reduce false activations: kick events are only sent if the foot returns to home target without any taps over a target, and toe or heel events are only recognized after a whole foot event if the foot lifts up entirely or the toe or heel state is sensed for 500ms.

A more evolved computer vision algorithm or pressure sensitive floor could remove the requirement for instrumented shoes. For example, the GravitySpace system [4] uses a high-resolution pressure-sensitive floor to unobtrusively detect different types of foot contacts and kicks using the contact pattern of planted feet. Alternatively, the shoe-based sensing could be improved to eliminate the need for the depth camera. For example, similar “on foot” sensing already exists for fitness tracking (e.g. Nike+ Fuel, Footlogger, Sensoria Sock, SENSEable Shoes) and researchers have used shoe-based sensors for interaction in the past. For example, Paradiso and Hu [20] put sensors in shoes to create an interactive dance performance and Matthies et al.’s [15] insoles use pressure sensors to detect actions like jumping, walking, and turning for virtual environment navigation.

Command Set Selection and Command Injection

A system-level service synchronizes the set of command mappings with the current foreground application state by template matching the focused window title text. The appropriate command set is currently chosen from hand built configuration files. This could be more automated by scraping application menus using accessibility APIs. Commands are sent to the application by injecting keystrokes that simulate short-cut key sequences.

EXAMPLE TASK SCENARIOS AND APPLICATIONS

To illustrate Tap-Kick-Click utility, we implemented three representative task scenarios: general web browsing, web-based academic research, and interactive debugging. These are good examples of suitable tasks for Tap-Kick-Click interaction for three reasons: 1) they are common tasks for different types of knowledge workers; 2) the task goals do not emphasize input speed (find information, read a paper, trace through code); and, 3) they involve periods without text entry or editing. The accompanying video provides demonstrations of representative scenarios.

Primary command sets for associated applications are shown in Figure 8. These mappings leverage the results of Alexander et al.’s [1] elicitation study and natural behaviour we observed in our controlled foot input experiment [25]. To facilitate learning, some commands are universal (like kicks to switch applications) and some are application independent (like scrolling pages or items with forward and backward toe taps). Note that a large set of commands are possible, but ultimately it is the user’s choice when to use foot input. For example, they could initially use feet only for scrolling web pages or documents when reading.

Web browsing

The command set uses “click mode” for hyperlink selection and supports standard web browsing tasks (Figure 8a). Toe tapping scrolls the page, link selection is with click mode, and history navigation is maps to whole foot commands. This provides a complete method for navigating within and between web pages. Heel actions close tabs, switch between tabs, and open a new tab. When a new tab is opened, the foot menu is used to select a bookmarked website. Occasional textual search queries or form entry are accomplished by reaching out and typing on the keyboard. Other features like bookmarking a page use outward whole foot taps. The system also has special website-specific command sets. For example, a special Google search results command set makes it easily to select from a list of results without labelling every link with click mode: a forward left kick highlights the first result, lower results are selected with toe taps, and a forward right kick selects. More website-specific command mappings could be developed for web applications such as email or task management.

Web-based Academic Research

Web-based academic research typically spans multiple applications for browsing web-based databases, reading PDF papers, and storing items in a reference manager. Currently, web browsing uses the techniques above, but website-specific browsing modes could be added to streamline sites like the ACM Digital Library or JSTOR. While browsing, a PDF link is opened using click mode and the paper read using toe taps for scrolling, heeltaps for tab switching, and kicks for zooming. Using an outward right-foot kick to change to the global command, the PDF reader may be closed with a backwards kick, returning the focus to the web-based database and website command set. There, a kick back adds the paper to the Zotero reference manager. After switching to Zotero using the global command set, Zotero command mappings (Figure 8b) enable navigating between libraries and selecting reference items with toe taps and whole foot taps add or delete a note. A forward kick opens the item in a PDF reader.

Interactive Code Debugging in an IDE

Finally, Tap-Kick-Click can be used for specialized tasks within text-heavy applications, such as a coding IDE. This can augment keyboard input by using feet for code navigation (Figure 8c): toe taps to scroll and heeltaps to switch tabs and expand/collapse code blocks. Perhaps most compelling is using an outward whole foot tap to set a breakpoint, and a forward kick to start a debugging session. The debugging command set (Figure 8d) uses toe taps to *step into*, *step out of*, and *step over* code while tracing through program execution code. A whole foot tap breaks or continues execution, and kicks exit or restart the session.

EVALUATION

The goal of our evaluation is to validate the fundamental Tap-Kick-Click interaction techniques as realized in the system when used for realistic tasks. This goal spans usability and usefulness by determining if the techniques and system can be learned and used during a short session and verify applicability to key application scenarios. This initial evaluation focuses on using foot input as a primary input method and was conducted in a lab. A longitudinal field study could evaluate how the system integrates into everyday computing tasks and explore how often people choose to use their feet for primary or secondary input.

Protocol

We recruited 8 people from a university (ages 20 to 30, 4 female). One participant had used a standing desk; none were familiar with the system or foot input. The software and sensing system were as described above. Three pairs of slippers were instrumented: Women's 8½, Men's 10, Men's 12 (US sizing). These provided an adequate range of sizes for both male and female participants.

Tasks

Participants performed a sequence of five tasks to practice the foot interaction vocabulary and apply it to web browsing and PDF viewing. The IDE debugger was not included to avoid restricting participants to programmers. The experimenter introduced components and provided clarification as requested. Participants used the keyboard for text entry, otherwise only foot input.

Task 1: Practice low-density foot actions — Participants performed the foot action (toe tap, heel tap, kick, etc.) and direction (forward, side, back) as indicated by a pair of foot icons and an arrow (similar to Figure 8). All 22 action-direction combinations were performed. This task was repeated up to 4 times until 3 errors or less occurred to ensure a common level of eyes-free performance.

Task 2: Practice "click mode" — Participants navigated through 10 Wikipedia pages, following specified links.

Task 3: Web browsing — Participants selected a pair of Wikipedia pages from a list, and then navigated from the first page to the second page using links. We used the website wikispeedia.net [32], which presents tasks of this form using a version of schools.wikipedia.org.

Task 4: Practice high-density foot menu — Using a version of the bookmark menu labelled with letters, participants were shown a letter and asked to select the corresponding item using any foot tap. All 20 item locations were selected.

Task 5: Web search and PDF viewing — Participants used the bookmark menu to open the ACM Digital Library, Microsoft Academic Search, and Google Scholar to look up a specific author's citation count. After, they found a specific paper by that author; they opened it in a PDF reader, and counted the number of figures in the paper. Finally, they closed the PDF reader and all web browser tabs.

Tasks 1, 2, and 4 focus on learnability and usability of specific types of interactions, while tasks 3 and 5 focus on more general usability with realistic tasks. On average, the task portion of the study took 47 minutes (excluding setup and calibration time).

After the tasks, a subset of the NASA-TLX was administered and a questionnaire-guided interview gathered subjective feedback on each system features, and suitability for hypothetical usage scenarios. The subject feature feedback included rating each feature on a continuous scale from 1 to 5 for "how easy it was to learn and use." Given the desktop context, there is an implied comparison between foot input and mouse and keyboard input which provided participants with some relative context for their subjective feedback.

Results and Discussion

Table 1 summarizes mean questionnaire results for NASA-TLX ratings and Table 2 summarizes "how easy it was to learn and use" the six main features.

Overall

All participants completed tasks without significant difficulty. NASA-TLX ratings are near neutral with moderately high variance shown by the standard deviations. Some rated the system as more mentally demanding, others rated it as more physically demanding. Although some experimenter clarification was required (more so in Tasks 1 and 2) participants were able to recover from errors like triggering the wrong command without resorting to keyboard or mouse. Regarding learning speed, 6 participants completed Task 1 with 3 errors or less on the third task repetition. Neutral to somewhat positive mean ratings for "ease of learning and use" of system features support these observations. We did not measure "enjoyability" since it is hard to quantify, but some participants commented to the effect that they found it fun and a bit like dancing.

These overall results are encouraging given that foot input is known to be harder to learn [9], the system is quite novel, and the relatively short study time. The fact that people can learn the core of the system and accomplish non-trivial tasks in a single experiment session is a measure of success.

System Features

Four participants reported high cognitive load when looking up commands in the *sidebar*, but 3 of these said practice would make this easier. P8 said "[It is mentally demanding] in the sense that you have to think a lot; it was easy to get mixed up between the different commands sometimes. Until you get the hang of it - once I got the hang of it, it seems pretty easy."

Participants rated the *contextual help screen* as easiest to learn and use, but even after practicing with it, no one used it during the applied tasks. The current static contextual help screen may be too limited.

Click mode was rated as moderately easy to learn and use, and participants used it multiple times in Tasks 2, 3, and 5. Two participants said link annotations covered too much text, and one participant said link selection had high physical demand due to the number of actions required.

Participants found the *bookmark menu* hardest to learn and use. This is supported by a 40% error rate in Task 4, high-density menu training. There were some sensing errors related to certain types of foot contacts and backward movement directions (see below), but position sensing was quite accurate. In our controlled target selection experiment, we did not find higher error rates when target location and size simulated the high density target layout used by the bookmark menu [26]. We believe the primary reason for bookmark menu difficulty was not enough practice causing mental error (selecting the wrong item) and physical errors (making the wrong movement).

Factor	Mean (stddev)
Frustration	8.0 (4.6)
Temporal	8.2 (3.4)
Mental	8.2 (5.0)
Physical	8.3 (4.2)
Effort	10.1 (2.1)
Performance	11.8 (4.2)

Table 1: NASA-TLX Ratings: mean (with stddev) by factor on scales from 0 (best) to 20 (worst)

Feature	Mean (stddev)
Help Screen	1.9 (1.1)
App Switching	2.0 (0.8)
Sidebar Feedback	2.4 (1.1)
App Navigation	2.5 (0.8)
Click Mode	2.5 (0.9)
Bookmark Menu	3.4 (0.7)

Table 2: Subjective ratings in response to question "For each of the features of the interaction technique, rate on a scale of 5 how easy it was to learn and use." (1 is Very Easy, 5 is Very Hard).

Participants only selected each of the 20 bookmark menu locations once. In Task 1, low-density training, participants repeated the task multiple times for extra practice resulting in a comparable error rate for Task 1 after the first run (36%), but was less than 14% for 6 participants on the third repetition. P4 wrote: "Bookmark menu: easy to select wrong [item] because they are very close to each other." In spite of the theoretical advantage of central, indirect feedback with the bookmark window, selecting high density targets requires more practice to master and appears more susceptible to sensing errors.

Interaction Technique

Three participants wanted a larger interaction area or encountered sensing errors from kicking too far. We used a conservative estimate of interaction range [25], but users may prefer a custom range. Four participants reported or experienced problems with whole foot taps, especially in the back direction. Analysing logs suggests this is caused by slow movements when a heel or toe tap is sensed before a whole foot tap is completed. More conservative user-calibrated thresholds would improve this. We did not find occurrences of "Midas-Step" suggesting our avoidance rules were effective while interacting with feet.

System Usefulness

When asked if the system would be useful for hypothetical applications, participants were generally positive: 8 said it would be useful for controlling a background application; 7 for controlling a web browser; 7 as a deterrent to using applications like Facebook; and 5 for controlling a citation manager. After we explained what an interactive debugger was, only 2 participants were in favour of using the system for that purpose. One participant felt using feet would add to the already high cognitive difficulty of debugging, but recall we did not control for programming experience.

Discussion

Our evaluation shows participants were able to learn the system and perform tasks requiring multiple foot gestures within a 60-minute time frame. When they made mistakes, they were able to recover within the framework of the system. The results of the study can be summarized in the following implications for design:

- Navigation controls, link selection, application switching and feedback panel were generally well received, although with somewhat of a learning curve. There is room for additional support in teaching commands to new users, perhaps in the form of an explicit tutorial, or a training task which prompts the user with a command (go back in the web browser) and requires the user to perform the correct foot action.
- The bookmark menu in its current form is somewhat difficult to learn and use – it may be better to also use *click mode* for selecting among many items in a menu like bookmarks.
- Two-foot jumping appears to be considered an easy to perform action, but our contextual help screen may not be as useful as we hoped in practice.
- The help screen is currently not used by participants. It might need to be reworked into a more relevant form, such as a tutorial or training task for the different web browser commands.
- Some minor system changes would improve performance, including increasing the interaction range and making toe and heel tap detection more conservative to improve performance of whole foot tapping detection.

CONCLUSIONS AND FUTURE WORK

The Tap-Kick-Click system demonstrates how foot interaction techniques can effectively control conventional desktop applications at a standing desk. Foot input may not provide a clear efficiency benefit compared to mouse and keyboard, but feet can still have a role in desktop computing. Since foot input is always available, it can be used for pure foot input breaks while still continuing a primary work task like reading a paper, or it can be used to augment mouse and keyboard for background tasks like controlling a media player. If desired, it could even be used as a physical deterrent for cyberslacking.

We accomplish this with a simple, but expressive vocabulary of indirect, discrete two-foot input composed of kicks and taps, as well as standing postures supported by feedback and contextual help systems. Our evaluation shows that the core idea of Tap-Kick-Click is learnable and usable. A longitudinal field study remains future work since it relies on incremental engineering improvements and a self-supporting configuration interface to map foot input to any application. We hope that our work demonstrates how a well designed foot input system can make daily computing a little bit healthier, and perhaps, even a little more fun.

REFERENCES

1. Jason Alexander, Teng Han, William Judd, Pourang Irani, and Sriram Subramanian. 2012. Putting Your Best Foot Forward: Investigating Real-world Mappings for Foot-based Gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1229–1238. http://doi.org/10.1145/2207676.2208575
2. Thomas Augsten, Konstantin Kaefer, René Meusel, et al. 2010. Multitoe: high-precision interaction with back-projected floors based on high-resolution multi-touch input. *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ACM, 209–218. http://doi.org/10.1145/1866029.1866064
3. Dave Berque, Jimmy Burgess, Alexander Billingsley, Shan Johnson, Terri Bonebright, and Brad Wethington. 2011. Design and evaluation of persuasive technology to encourage healthier typing behaviors. *Proceedings of the 6th International Conference on Persuasive Technology: Persuasive Technology and Design: Enhancing Sustainability and Health*, ACM, 9:1–9:10. http://doi.org/10.1145/2467803.2467812
4. Alan Bränzel, Christian Holz, Daniel Hoffmann, et al. 2013. GravitySpace: Tracking Users and Their Poses in a Smart Room Using a Pressure-sensing Floor. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 725–734. http://doi.org/10.1145/2470654.2470757
5. Andrew Crossan, Stephen Brewster, and Alexander Ng. 2010. Foot Tapping for Mobile Interaction. *Proceedings of the 24th BCS Interaction Specialist Group Conference*, British Computer Society, 418–422.
6. David Dearman, Amy Karlson, Brian Meyers, and Ben Bederson. 2010. Multi-modal Text Entry and Selection on a Mobile Device. *Proceedings of Graphics Interface 2010*, Canadian Information Processing Society, 19–26.
7. Colin G. Drury. 1975. Application of Fitts' Law to foot-pedal design. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 17, 4: 368–373.
8. Koumei Fukahori, Daisuke Sakamoto, and Takeo Igarashi. 2015. Exploring Subtle Foot Plantar-based Gestures with Sock. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, ACM, 3019–3028.

http://doi.org/http://dx.doi.org/10.1145/2702123.2702308

9. Fredrick P. Garcia and Kim-Phuong L. Vu. 2011. Effectiveness of hand- and foot-operated secondary input devices for word-processing tasks before and after training. *Computers in Human Behavior* 27, 1: 285–295. http://doi.org/10.1016/j.chb.2010.08.006
10. Anne Carolyn Grunseit, Josephine Yuk-Yin Chau, Hidde Pieter van der Ploeg, and Adrian Bauman. 2013. "Thinking on your feet": A qualitative evaluation of sit-stand desks in an Australian workplace. *BMC Public Health* 13: 365. http://doi.org/10.1186/1471-2458-13-365
11. Teng Han, Jason Alexander, Abhijit Karnik, Pourang Irani, and Sriram Subramanian. 2011. Kick: Investigating the Use of Kick Gestures for Mobile Interactions. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ACM, 29–32. http://doi.org/10.1145/2037373.2037379
12. Errol R. Hoffmann. 1991. A comparison of hand and foot movement times. *Ergonomics* 34, 4: 397–406. http://doi.org/10.1080/00140139108967324
13. Johanna Hoysniemi. 2006. International survey on the Dance Dance Revolution game. *Comput. Entertain.* 4, 2. http://doi.org/10.1145/1129006.1129019
14. S.-H. Kim and D. B. Kaber. 2008. Design and Usability Evaluation of Foot Interfaces for Dynamic Text Editing. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 52, 6: 581–585. http://doi.org/10.1177/154193120805200615
15. Denys J. C. Matthies, Franz Müller, Christoph Anthes, and Dieter Kranzlmüller. 2013. ShoeSoleSense: Proof of Concept for a Wearable Foot Interface for Virtual and Real Environments. *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*, ACM, 93–96. http://doi.org/10.1145/2503713.2503740
16. Brian Meyers, A. J. Bernheim Brush, Steven Drucker, Marc A. Smith, and Mary Czerwinski. 2006. Dance Your Work Away: Exploring Step User Interfaces. *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, ACM, 387–392. http://doi.org/10.1145/1125451.1125534
17. Dan Morris, A.J. Bernheim Brush, and Brian R. Meyers. 2008. SuperBreak: Using Interactivity to Enhance Ergonomic Typing Breaks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1817–1826. http://doi.org/10.1145/1357054.1357337
18. Volker Paelke, Christian Reimann, and Dirk Stichling. 2004. Foot-based mobile interaction with games. *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, ACM, 321–324. http://doi.org/10.1145/1067343.1067390
19. Toni Pakkanen and Roope Raisamo. 2004. Appropriateness of foot interaction for non-accurate spatial tasks. *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ACM, 1123–1126. http://doi.org/10.1145/985921.986004
20. Glenn Pearson and Mark Weiser. 1988. EXPLORATORY EVALUATIONS OF TWO VERSIONS OF A FOOT-OPERATED CURSOR-POSITIONING DEVICE IN A TARGET-SELECTION TASK. *SIGCHI Bull.* 19, 3: 70–75. http://doi.org/10.1145/49108.1046356
21. G. Pearson and M. Weiser. 1986. Of Moles and Men: The Design of Foot Controls for Workstations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 333–339. http://doi.org/10.1145/22627.22392
22. Stephan Richter, Christian Holz, and Patrick Baudisch. 2012. Bootstrapper: Recognizing Tabletop Users by Their Shoes. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 1249–1252. http://doi.org/10.1145/2207676.2208577
23. Julie Rico and Stephen Brewster. 2010. Usable gestures for mobile interfaces: evaluating social acceptability. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 887–896. http://doi.org/10.1145/1753326.1753458
24. Arthur Saltzman. 1998. Computer user perception of the effectiveness of exercise mini-breaks. *Proceedings of the Silicon Valley Ergonomics Conference and Exposition*, 147–151.
25. William Saunders and Daniel Vogel. 2015. The Performance of Indirect Foot Pointing using Discrete Taps and Kicks While Standing. *GI '15 Proceedings of the 41st Graphics Interface Conference*, 265–272.
26. Johannes Schöning, Florian Daiber, Antonio Krüger, and Michael Rohs. 2009. Using hands and feet to navigate and manipulate spatial data. *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, ACM, 4663–4668. http://doi.org/10.1145/1520340.1520717
27. Jeremy Scott, David Dearman, Koji Yatani, and Khai N. Truong. 2010. Sensing foot gestures from the pocket. *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, ACM, 199–208. http://doi.org/10.1145/1866029.1866063
28. A. L. Simeone, E. Velloso, J. Alexander, and H. Gellersen. 2014. Feet movement in desktop 3D interaction. *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, 71–74. http://doi.org/10.1109/3DUI.2014.6798845
29. Eduardo Velloso, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. Interactions Under the Desk: A Characterisation of Foot Movements for Input in a

Seated Position. In *Human-Computer Interaction – INTERACT 2015*. Springer International Publishing, 384–401.

30. Eduardo Velloso, Dominik Schmidt, Jason Alexander, Hans Gellersen, and Andreas Bulling. 2015. The Feet in Human–Computer Interaction: A Survey of Foot-Based Interaction. *ACM Comput. Surv.* 48, 2: 21:1–21:35. http://doi.org/10.1145/2816455
31. Jessica Vitak, Julia Crouse, and Robert LaRose. 2011. Personal Internet use at work: Understanding cyber-slacking. *Computers in Human Behavior* 27, 5: 1751–1759. http://doi.org/10.1016/j.chb.2011.03.002
32. Robert West, Joelle Pineau, and Doina Precup. 2009. Wikispeedia: An Online Game for Inferring Semantic Distances Between Concepts. *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, Morgan Kaufmann Publishers Inc., 1598–1603.
33. Anthony Whitehead, Hannah Johnston, Nicole Nixon, and Jo Welch. 2010. Exergame effectiveness: what the numbers can tell us. *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, ACM, 55–62. http://doi.org/10.1145/1836135.1836144