Habitsourcing: Sensing the Environment through Immersive, Habit-Building Experiences

Katherine Lin, Henry Spindell, Scott Cambo, Yongsung Kim, Haoqi Zhang
Northwestern University
Evanston, IL
{katherinelin2016, henryspindell2015, scottallencambo, yk}@u.northwestern.edu, hq@northwestern.edu

ABSTRACT

Citizen science and communitysensing applications allow everyday citizens to collect data about the physical world to benefit science and society. Yet despite successes, current approaches are still limited by the number of domain-interested volunteers who are willing and able to contribute useful data. In this paper we introduce *habitsourcing*, an alternative approach that harnesses the habit-building practices of millions of people to collect environmental data. To support the design and development of habitsourcing apps, we present (1) interaction techniques and design principles for *sensing through actuation*, a method for acquiring sensing data from cued interactions; and (2) *ExperienceKit*, an iOS library that makes it easy for developers to build and test habitsourcing applications. In two experiments, we show that our two proof-of-concept apps, *ZenWalk* and *Zombies Interactive*, compare favorably to their non-data collecting counterparts, and that we can effectively extract environmental data using simple detection techniques.

Author Keywords

Habitsourcing; physical crowdsourcing; communitysensing; citizen science

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

INTRODUCTION

Communitysensing and *citizen science* enable scientists, entrepreneurs, and government agencies to access previously unavailable environmental data to study the natural world, deliver new applications, and improve public services. Existing citizen science and communitysensing projects recruit volunteers to monitor the environment [1, 6], track migration patterns and invasive species [4, 5], report city problems [20, 11], and refine maps [12, 3, 18]. But despite successes, current approaches rely on, and are thus often limited by, the number of domain-interested volunteers they are able to recruit to help contribute useful data. These limits are especially problematic in more niche domains, for which a large population of domain-interested people may not exist.

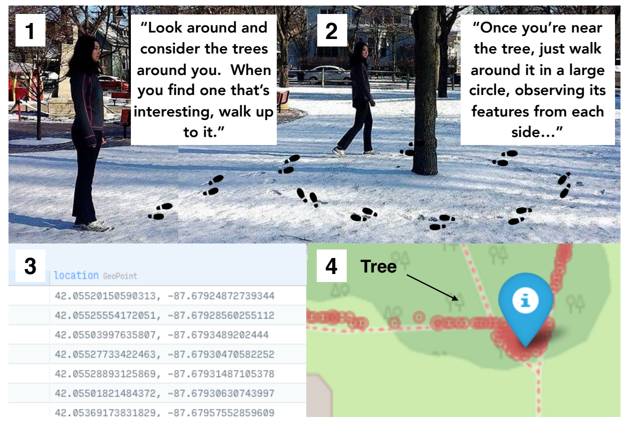

Figure 1. Habitsourcing apps add immersion to habit-building interactions by using cues to elicit user actions that produce sensor data and GPS traces which we analyze to infer useful data about objects that users encounter in the physical environment.

In this paper we introduce *habitsourcing*, an alternative approach that leverages people's daily habit-building practices to collect sensing data about the physical environment that they encounter while practicing a habit such as going for a run or taking a walk (see Figure 1). Millions of people are developing personal habits, and many already use mobile apps to support their habit-building practices. For example, popular apps such as Headspace and Zombies, Run! have been downloaded millions of times and support hundreds of thousands of active users interested in practicing mindfulness and running, respectively. Habitsourcing seeks to piggyback off the success of habit-building apps to reach a larger crowd of potential contributors who participate to support their habit, and not because they are interested in collecting data for some external need. In order to use habit-building experiences for data collection, we must tackle the design challenge of creating interactions that simultaneously support people's habit-building and collect useful data.

The main conceptual contribution of this work is *the idea of using immersive interactions embedded within existing habit-building experiences to elicit natural user actions from which we can extract useful sensing data.* We present ZenWalk and

UIST 2016, October 16 - 19, 2016, Tokyo, Japan

ACM 978-1-4503-4189-9/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984533

Zombies Interactive, two proof-of-concept iOS applications through which we demonstrate the effectiveness of habitsourcing and study its design, implementation, and use. ZenWalk is a walking meditation app that helps users practice mindfulness. Zombies Interactive is a running app that uses a zombie apocalypse storyline to entertain and motivate users to run. In both of these examples, cued interactions allow users to have a more engaging experience. Users' actions in response to these cues can then be used to derive sensing data about the physical world.

The core technical contribution of this work for designers is *sensing through actuation*, an interaction technique for collecting environmental data by cueing users to perform physical actions that are appropriate given their habit-building goals. Figure 1 shows an example ZenWalk interaction in which the user is directed to examine and walk around a tree of interest. Because we know the type of interaction cued and the time it was cued, detecting the loop to identify the tree location from GPS traces is easy with simple pattern detection and does not require advanced machine learning algorithms. Based on multiple rounds of iterative design and testing of ZenWalk and Zombies Interactive, we present later in this paper a set of design principles for effectively using sensing through actuation in habitsourcing applications.

The core technical contribution of this work for developers is *ExperienceKit*, an extensive iOS library written in Swift that makes it easy for a developer to create habitsourcing applications with few lines of code. The library is designed so that components of an experience are modular, interactions are reusable, and data is collected and categorized automatically. Our architecture also provides extensions that afford complex, context-aware interactions that are beyond the scope of our proof-of-concept applications.

In summary, our paper makes the following contributions:

- *Habitsourcing*, the idea of transforming people's habit-building activities into data collection opportunities.
- *ZenWalk* and *Zombies Interactive*, iOS applications that respectively help people practice mindfulness and run, and that indirectly sense objects in the natural world.
- *Sensing through actuation*, an interaction technique for adding immersion and enjoyment into habit-building apps to elicit user actions from which we can infer useful environmental data.
- *ExperienceKit*, an iOS library that makes it easy for a developer to build rich habitsourcing applications.
- Results of two user studies, that show (a) habitsourcing apps are as preferred or more preferred than their habit-building app counterparts; and (b) the sensing data inferred from interactions accurately represents the environment in a number of use cases.

RELATED WORK

Millions of people use mobile apps such as Zombies, Run! and Headspace to practice and build their habits. To support habit-building, some apps adopt research best practices to provide (a) a predetermined process that can lead users to a target behavior (e.g., Headspace); (b) simulated experiences within which people can immersively interact (e.g., Zombies, Run!) [9]. They also promote continual, daily practice by using episodic storyline missions and streak counters to encourage long-term engagement. In our work, we use habit-building technologies as a basis for habitsourcing applications, but focus primarily on advancing the collection of environmental data through people's habit routines.

Most participatory communitysensing and citizen science apps engage people who are already interested in the domain to contribute data. For example, birders track the presence of particular bird species [16], drivers report traffic conditions [20], bus riders report real-time transit information [21], and concerned citizens report city problems [11]. In these examples, a data contributor is most often also a data consumer; a user benefits directly from contributing and also from the collective efforts of others' contributions. In contrast, our work advances an *indirect* approach in which data is collected as a byproduct of people who participate to build a habit, to play a game [3, 12, 15, 18], or to access their phone [19, 17]. In this manner, our approach has the potential to broaden participation to a much larger set of non-domain-interested people who may nevertheless contribute data because using habitsourcing apps advance their personal habit goals.

Most closely related to our work, physical games with a purpose collect sensing data as a byproduct gameplay [3, 12, 15, 18]. One example is PhotoCity [18], which engages players to take photos in the context of a capture-the-flag game that allows its designers to reconstruct 3D models of physical spaces. A contemporary example is Google's Ingress, which pits players against rival factions to create an immersive game environment in which dedicated gamers travel out of their existing routes to take in-game actions that produce crowdsourced data, such as walking routes, as a byproduct. While some design elements from these games may be useful for habitsourcing apps, we must first and foremost support people's habit-building goals if we are to tap into the efforts and routines of millions of habit-builders who may otherwise be uninterested in contributing sensing data. For example, this steers us away from enjoyable interactions that require users to interact continually with their device, or that otherwise distract from user's habit-building goals.

Our proof-of-concept habitsourcing apps use people's motion activities to infer the locations and characteristics of physical objects in the environment without complex machine learning algorithms. Typically, making inferences about people's motion activities for the purposes of activity recognition and context detection is a challenging machine learning problem [13, 14]. This is particularly true in open-world domains, in which a system can make few assumptions about what a person is doing a priori [2]. In our setting, the problem is made much simpler because our apps suggest specific actions for users to take that occur in pre-specified timeframes. In the case of ZenWalk and Zombies Interactive, inferring user activity from sensor traces is easy with simple pattern detection and does not require advanced machine learning algorithms.

ZenWalk

Stage 1: Focus on posture and breath
"Bring your awareness to how the body moves as you walk…"

…

Stage i: Focus on environment
"Allow your awareness to become heightened and take in the world around you, seeing it as it is, without judgment or excess thought…"

Stage i+1 (*interactive): Focus on trees
"As you take in your surroundings, look around and observe any interesting trees around you… when you see one that you find interesting, go up to it and walk around it in a circle."

…

Stage n: Reflect on feelings
"Take a moment to pause, close your eyes, and reflect on how you're feeling right now…"

Zombies Interactive

Stage 1: Warm up, mission brief
"This is the advanced intel team at Abel Township. We're gathering information on zombies…"

…

Stage i: Start running, listen to story
"You're the only runner whose device is still fully functioning, so we're going to have to ask you to do some recon work…"

Stage i+1 (*interactive): Respond to events in the story
"Runner 5, our monitors show your pace has slowed and zombies in the area are gaining ground. You need to increase speed to reach a safe distance. You should pass a fire hydrant after 15 to 20 seconds, at which point you should return to regular pace. Begin sprinting now."

…

Stage n: Cool down, mission debrief
"Great job. You've are now in a safe place to stop and rest."

Figure 2. Habitsourcing apps such as ZenWalk and Zombies Interactive insert sensing opportunities in appropriate stages of a habit-building experience.

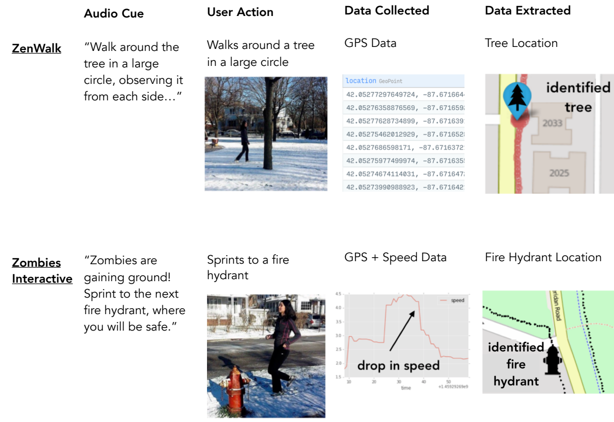

Figure 3. Examples of ZenWalk and Zombies Interactive interactions for collecting tree and fire hydrant locations. Audio cues elicit user actions from which we can collect location and motion data (e.g. GPS, accelerometer) from which we can infer object locations.

In our iterative design process, we benefitted from being able to frequently test with people who use the habit-building apps that our apps are based on. Since accessing such people locally was sometimes challenging, we made use of Reddit and found that we were able to readily recruit users who provided helpful feedback throughout the design process. Our approach follows prior work on using Reddit and other forms of social media for gathering design feedback [10, 7, 8].

HABITSOURCING

The user of a habitsourcing app is guided through the habit-building experience in a sequence of *stages*, similar to those in existing habit-building mobile apps but with the addition of *interactive stages* that provide a more immersive experience (see Figure 2). Each stage serves a purpose for the habit-building experience (e.g., help the runner warm up). A stage consists of one or more *moments*, which collectively provide audio instructions to accomplish the purpose of the stage. Within interactive stages, users hear cues to interact with physical objects in their environment. These interactions can potentially take many forms, including motion (e.g. running faster), gestures (e.g. knocking on a phone), and speech (e.g. answering a question with voice). We discuss in the next section techniques for designing such interactions and extracting useful sensing data from them.

With *ZenWalk*, a user goes through a walking meditation in stages that help them (1) focus on their breath, (2) focus on their body, (3) focus on their thoughts and feelings, and (4) broaden their awareness to their external environment in order to balance their inner and outer worlds. Because the user is focused internally in the earlier stages of the meditation, it makes sense to insert interactions later in the experience, when the user is asked to broaden their awareness to their surroundings in stage 4. In an example ZenWalk interaction, the user hears instructions to go up to a tree they find interesting, walk around it, and observe its features from each side (Figure 2, left).

With *Zombies Interactive*, a user goes through a running experience in stages that (1) help them warm up, (2) brief them with their mission in the zombie narrative, (3) motivate them to run through mission objectives, (4) guide them to cool down, and (5) obtain narrative closure to end the story. Habitsourcing interactions are inserted in the form of mission objectives in stage 3. The user hears commands from a mission control center and responds to these commands based on their current situation in order to complete objectives. As an example, a user might hear a command that informs them that zombies are nearby and directs them to sprint to the nearest fire hydrant, where they will be safe (Figure 2, right).

Sensing Through Actuation: An Interaction Technique for Collecting Data through Cued Interactions

We address in this section the challenge of designing interactions that both enhance the habit-building experience and collect environmental data. We base our approach on two observations. First, unless users receive specific action cues, it is difficult to extract any meaningful data about the environment using only users' motion data [13, 14]. Second, users use habit-building apps for achieving personal goals and not for collecting data; habitsourcing interactions that are irrelevant to their practice are unlikely to be acceptable.

Sensing through actuation is a technique for collecting environmental data by cueing users to perform physical actions that are appropriate given their habit-building goals. Figure 3 shows two example interactions. In one example, a ZenWalk user is asked to walk around a tree. The interaction helps the user develop an awareness of their outer world, directly supporting the user's goal of practicing mindfulness. At the same time, the location of the tree can be inferred using a trace of their GPS data. In the other example, a Zombies Interactive user is asked to sprint past a fire hydrant in order to escape a zombie horde. Users interact with real objects in their environment rather than imagined ones and may feel motivated to momentarily increase their pace. The fire hydrant's location can be approximated by examining where the user's speed data shows a sharp increase, indicating a sprint.

Sensing through actuation interactions aim to avoid disrupting, or even to enhance, the user's experience by using cued actions that directly support the practice of the user's habit.

The user is not required to look at their phone, because sensing through actuation interactions make inferences based on user activity (e.g. motion, speech) to collect environmental data. Thus, the user can remain aware of their physical surroundings and environmental data can be extracted by detecting motion activity patterns that occur within a short timeframe around the cue.

Design Process

We followed an iterative design process to study and address the challenge of creating interactions that are natural to habit-building and also afford collecting environmental data. A habitsourcing app's success relies on maintaining an enjoyable, effective habit-building experience that the user will want to use continually. When creating an interaction, the designer needs to consider not only the action the user is taking but also how that action fits in the context of the overall habit-building experience. In our design process we considered the following aspects of an interaction:

- The physical action that the user is prompted to do
- The narrative that the user hears
- The time during the habit-building experience at which the user does the interaction
- Other factors that may influence whether the interaction will be a good experience for the user (e.g. their locations, the interactions they have already done)
- The extraction algorithm that is used to detect environmental data from the user's physical actions, and how it works in junction with the interaction itself.

We focus in this paper on advancing interaction designs for collecting data through cued interactions, and not on solving particular data challenges. We thus chose for verification convenience to focus on common objects, including trees, fire hydrants, and buildings. In the discussion, we address how our approach can be used for data collection goals that extend beyond such common, static objects.

ZenWalk and Zombies Interactive build off existing habit-building apps. ZenWalk follows the walking meditation structure used in Headspace and suggested by meditation guides such as Wildmind. We designed interactions that took place within the stage of the meditative walk in which a user begins to shift from focusing on their own body and breath to focusing also on their external environment. For our sensing objective, we focused on interactions that identified the presence and locations of trees. Zombies Interactive builds on the contents from the missions "Jolly Alpha Five Niner" and "Scouting Mission" of the Zombies, Run! app. We sought to design interactions that fit the storylines in these missions and could be used to collect information about the presence and location of infrastructure (e.g. fire hydrants, buildings, stop signs).

In brainstorming possible interactions for both of our apps, we first considered interactions from which we can extract data and then used domain-specific constraints to narrow the design space. For example, we knew that interactions in which a user walks around an object can be used to detect object locations. To support users developing an awareness of their environment in their meditation, we then tested an interaction that asks a user to circle a tree while observing its features. Following this brainstorming process for both ZenWalk and Zombies Interactive provided us with a pool of testable interactions that had potential to provide quality data and simultaneously contribute to the habit practice.

We then devised distinct design processes for ZenWalk and Zombies Interactive to support our exploration of a set of interactions and to evaluate their effectiveness within a habitsourcing experience. For ZenWalk, it is important to test a mindfulness interaction within the context of a full meditation because interactions require users to be in a certain state of mindfulness that they reach through the guidance of previous stages of the meditation. To explore multiple designs, we tested multiple versions of ZenWalk in parallel that provided the full meditation but differed in their interaction stages. We then refined designs through iterative rounds of testing.

For Zombies Interactive, we assumed that interactions within a run are largely independent and can thus be developed separately. This allowed us to adopt a two-phase process where we test the effectiveness of interactions on their own and later test their effectiveness in the context of the overall experience. For example, in the first phase we were able to prototype and evaluate many different interactions within a single experience, composed only of interaction stages. Since interactions may only last a couple of minutes in an experience that is typically 30 minutes or longer, testing interactions in isolation allows us to learn about many designs quickly and identify interactions that work well in isolation. We then test in the context of the full experience in the second of the two phases referenced above.

We recruited testers both locally from a convenient sample and online through Reddit. For ZenWalk, we recruited 8 local testers. For Zombies Interactive, we found it difficult to recruit a large number of testers locally who were representative of our actual user population, i.e. people who were already using habit-building apps such as Zombies, Run!. We thus focused local tests primarily on gathering usability feedback on interactions from 2 local testers. We then recruited 7 testers online from around the world through subreddits */r/runner5* and */r/running* who were enthusiastic about wanting to run regularly and were already using apps like Zombies, Run!. Local testers volunteered their time; Reddit testers received $5 gift cards.

Lessons learned for designing effective interactions

Figures 4 and 5 summarize the key interactions we tested through the design process for our two apps as well as feedback and lessons we learned from each iteration. Through iterative testing of these apps, we identified the following general lessons for designing sensing through actuation interactions for habitsourcing apps:

- *The narrative needs to clearly instruct the user on what to do and when to do it.* Description and instruction for each interaction should be clear and precise so that users know exactly what they are supposed to do. For example, we found that some ZenWalk testers were not sure when they

Interaction	Example Tester Feedback	Lessons Learned
1. Observe/walk in a circle around a piece of litter	User did not enjoy observing the litter because it was not relaxing. She felt stressed thinking about all the litter in the environment.	Objects of interaction should be relevant to the habit practice (e.g. meditation)
2. Observe/walk in a circle around an interesting tree	User enjoyed observing trees as it felt natural and relaxing, but she felt bored looking at a tree for so long (a few minutes)	Details of habit practice, such as timing, should be considered
3. Observe/walk in a circle around an interesting tree, with guidance for what to observe about the tree	User enjoyed the tree observation guidance	Guidance should be provided where it would be helpful
4. Observe/walk in a circle around two interesting trees	User enjoyed looking at a greater variety of trees	Interactions should provide an interesting and varied experience
5. Stand in front of interesting tree, slowly turn in place and observe the environment	a) User was not sure how long to turn in place b) User felt awkward spinning in place on a busy street	a) Instructions should be clear and specific about what the user should do b) Interaction should feel comfortable in social context

Figure 4. ZenWalk Iterative Design Findings

Interaction	Example Tester Feedback	Lessons Learned
1. Sprint until you reach a stop sign to avoid the zombie hangout	a) Runners enjoyed being motivated to increase pace b) Runners appreciated the post-interaction audio saying "Good work, you can return to your regular pace"	a) Interactions should help the user towards their habit practice goals b) Interactions should give feedback acknowledging the user's action
2. Find somewhere to sit and rest / Find cover and stretch	Runners did not like stopping during their runs	Interactions should not ask users to perform actions that are unnatural to their habit practice
3. Double knock on your device to record tall buildings	Runners did not understand what was being asked of them by "double knocking"	Interaction instructions should be clear and easy to follow
4. Avoid zombies by taking a higher altitude route	Runners were intrigued by the possibility of their route affecting the plot	Experiences striving for immersion should consider interactions for which the user's choices have consequences
5. Sprint past ten trees to safety	Runners were taken out of the story when there were no trees nearby	Interactions centered around an object should consider the case of the object's absence

Figure 5. Zombies Interactive Iterative Design Findings

should stop circling a tree, so they stopped circling it and moved on before the interaction was over.

- *Interactions should provide feedback when the user performs the prompted action*. Interactions should acknowledge the execution of a prompted action. For example, we found that some Zombies Interactive testers wanted to hear a noise indicating that they had completed a task.
- *Designers should consider the cases in which the user is not able to perform the prompted action*. Interactions can quickly break down when a user cannot locate the object they are prompted to interact with. For example, ZenWalk testers who were prompted to find and circle a tree had a negative experience when they could not find a nearby tree.
- *Interactions should relate directly to habit-building goals*. Users who are interested in running tended to prefer interactions that involve sprinting over those that involve unrelated actions, e.g., hiding behind a bush to avoid zombies.
- *Interactions should be designed with social context in mind*. Users often felt less comfortable performing actions that seem awkward when other people were around. For example, a ZenWalk tester reported feeling uncomfortable testing an interaction that had him circle trees on busy streets when there were many people around.
- *Designers should consider the details of the habit practice*. For instance, timing is important in a meditation, so the duration of interactions requires careful tuning so that ZenWalk users do not feel rushed.

ExperienceKit: An iOS Library for Developing Habitsourcing Applications

Building habitsourcing apps requires an architecture that can track data, respond to events and user actions, and construct stages and the transitions between them. Current habit-building apps only play static audio files and lack functions to collect and analyze data. To fill this void and to support the development of habitsourcing applications broadly, we introduce *ExperienceKit*, a native iOS Swift framework and library that supports a developer building habitsourcing applications. ExperienceKit contains classes that handle the creation of interactions that can include audio files, pauses, and activation of sensors. It also simplifies the extraction and analysis of data from the habit practice by tying sensor traces to interactions. This allows developers to easily build habitsourcing apps that, without such an architecture, can require thousands of lines of code.

Design Goals

ExperienceKit is designed to meet the following goals:

- *Create habitsourcing experiences with less than 100 lines of code.* ExperienceKit handles all logic needed to manage a habitsourcing experience. Consequently, one can define a complete experience in a habitsourcing app (Figure 6), as well as the interactions included in the experience (Figure 7) using a small number of constructors and appropriate selection of parameters.
- *Modular, so that interactions and stages can be easily arranged in varying configurations to form different experiences.* With ExperienceKit, a developer can easily alter experiences by rearranging stages and the moments within each stage. A developer creating an experience with sets of individual interactions could easily experiment with the order of stages and test multiple different experiences to quickly find a desired configuration.
- *Reusable, so that interaction techniques and data collection mechanisms can be reused across interactions and apps.* Sensing through actuation interactions are modeled in ExperienceKit so that templates and techniques for interaction and data collection are easily portable between applications. For example, a developer may have a Zombies Interactive interaction with the goal of finding a building's diameter. By editing parameters for two or three lines of code, this interaction can be transformed into a ZenWalk interaction that finds the boundaries of a garden. Once an interaction technique has been created for a habitsourcing

```
class MissionViewController: UIViewController, ExperienceManagerDelegate {
    var missionTitle: String = ""
    var experienceManager:ExperienceManager!

    override func viewDidLoad() {
        super.viewDidLoad()
        CLLocationManager().requestAlwaysAuthorization()

        // Construct interactions
...
        let findFireHydrant = Interaction(moments: [findHydrantInstruct,
findHydrantCollector, Sound(fileNames: ["radio_static","you've_thrown_off","radio_static"])],
title: "Sprint to hydrant")

...
        let passTenTrees = Interaction(moments: [passTenTreesInstruct,
passTenTreesCollector, Sound(fileNames: ["radio_static","you_should_be","radio_static"])],
title: "Sprint past ten trees")

...
        let sprintToBuilding = Interaction(moments: [sprintToBuildingInstruct,
sprintToBuildingCollector, Sound(fileNames:
["radio_static","building_confirmed","radio_static"])], title: "Sprint to tall building")
        let sprintingInteractions = [findFireHydrant, sprintToBuilding, passTenTrees]

        // Construct experience
        var stages: [Stage] = []
        let stage1 = Stage(moments: [Sound(fileNames: ["radio_static", "intel_team_intro",
"radio_static", "vignette_transition", "S1M16_1"]), Interim(lengthInSeconds:
2.minutesToSeconds), Interim(lengthInSeconds: 120)], title: "Scouting",
interactionInsertionIndices: [2], interactionPool: sprintingInteractions)
...
        let stage7 = Stage(moments: [Sound(fileNames: ["S1M16_7", "mission_completed"])],
title: "Debrief")
        stages = [stage1, stage2, stage3, stage4, stage5, stage6, stage7]

        experienceManager = ExperienceManager(title: missionTitle, stages: stages)
        experienceManager.delegate = self
    }
}
```

Figure 6. Implementation of a Zombies Interactive mission using ExperienceKit. Creating the entire mission requires only 50 lines of code.

```
// Construct interactions
let findHydrantInstruct  = Sound(fileNames: ["radio_static",
"our_monitors_show", "radio_static"])

let findHydrantCollector = SensorCollector(lengthInSeconds: 30, dataLabel:
"fire_hydrant", sensors: [.Location, .Speed])

let findFireHydrant = Interaction(moments: [findHydrantInstruct,
findHydrantCollector, Sound(fileNames:
["radio_static","you've_thrown_off","radio_static"])], title: "Sprint to
hydrant")
```

Figure 7. Implementation of an ExperienceKit interaction in Zombies Interactive. This interaction instructs the user to run until they reach a fire hydrant, then collects speed and location data for 30 seconds.

experience, it is easy to adapt it to another, no matter how different the narratives are.

- *Easy to extend interaction techniques and data collection methods.* It is easy for a developer to create new interaction techniques using the ExperienceKit architecture. For example, some interactions wait for a user action to trigger a transition. Designing triggers for different actions only requires extending the condition detection code; the triggering logic that advances the experience remains the same. Using the framework, a developer can also easily create new data collection methods, e.g., that track data from new device sensors not currently included.
- *Provide affordances for inserting interactions dynamically based on the user's current location and the status of the experience.* With ExperienceKit, a developer can create interactions that are inserted opportunistically into the experience only if some specified requirements are met. These requirements may include conditions such as the user's proximity to a given location, or the time remaining in the experience. For example, this allows us to define a Zombies Interactive interaction with a certain statue, such that only users who run near the statue will receive that interaction.

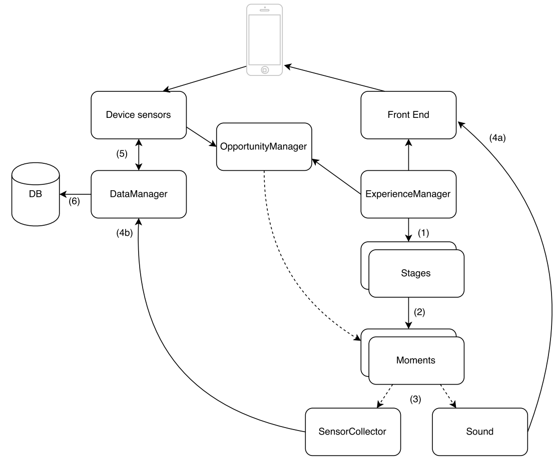

Figure 8. Architecture Diagram of Experience Kit

Architecture

The ExperienceKit architecture consists of the following major components:

- *Moments* are the building blocks of a Stage, and typically serve the purpose of advancing the narrative with audio, providing context for an interaction, or collecting data as part of an interaction. Moments are divided into subclasses that serve these different functions – the Sound subclass only plays audio files, and the SensorCollector subclass collects sensor data for a specified duration. Some Moments are *interruptible*, meaning if necessary, they will yield to other dynamic interactions that can be injected into a Stage.
- *Stages* are sequential groupings of Moments. The function of a Stage is to tie together moments according to a purpose (e.g. a number of Sounds introducing a narrative). Together, one or more Stages form a habitsourcing experience.
- *ExperienceManager* plays a collection of Stages by subscribing to events that fire when each has finished. It manages communication between Stages, DataManager, and OpportunityManager.
- *DataManager* handles toggling of device sensors, and saves data from and about the experience.
- *OpportunityManager* monitors the current state of the experience and dynamically suggests fitting interactions to be inserted into the experience, by comparing interaction prerequisites and the user's situational context.

Experience Flow

To show how the architectural components interact with one another, we walk through how the ExperienceKit architecture manages an experience and collects data via sensing through actuation. Figure 8 illustrates the process.

The ExperienceManager is passed an ordered set of Stages (1), which are each a collection of Moments (2). Some of these Moments are interactive, and some are not (3). The ExperienceManager subscribes to the events published by the current Stage, and each Stage subscribes to the events published by its Moments.

Moments are responsible for publishing a "finished" event when their task has completed, whether that task is to play a static audio file or to collect sensor data for a specified period of time. The use of a publish-subscribe pattern allows for a loose coupling between the experience and the Moments it is comprised of, as each level (Moment → Stage → experience) can make a decision to act or propagate an event up the tree.

If a Moment being played is of subclass SensorCollector, it will publish an event "sensorCollectorStarted" along with a payload containing the names of sensors to be turned on (4,5). This event is propagated up to the ExperienceManager, which will then notify the DataManager that the specified sensors need to be turned on for the duration of the current Moment. All sensor data collected during this period will be timestamped and associated with a "DataEvent" object in the database (6), that describes the source experience and interaction. This allows us to later detect objects based on device data, as it is tied to an interaction's instructions. The sensor data continues to be saved until the "sensorCollectorFinished" event is triggered. Upon hearing a Moment has "finished," the current Stage cues the next Moment in line.

Once the current Stage has heard "finished" from its final Moment, the Stage publishes its own "finished" event upon which the ExperienceManager will begin the next Stage. This continues until the experience has completed.

At certain points during the process, the experience may be interrupted by the OpportunityManager. During an interruptible Moment, the ExperienceManager will query the OpportunityManager to see if there are any context-specific interactions that can be inserted immediately. This may be the case if the user is near a point of interest, is moving at a certain rate, or otherwise meets some criteria associated with an interaction. When criteria are met during an interruptible Moment, one of the matching interactions will immediately displace the current Moment and begin playing. Once it has finished, the experience resumes as previously planned.

Implementing Zombies Interactive and ZenWalk

Zombies Interactive and ZenWalk are native iOS apps built using ExperienceKit. The baseline experience without interactions for Zombies Interactives uses the exact storyline and audio files from Zombies, Run! Missions 1 and 16. Added interactions come from invented characters injected into the story and are recorded by one of the authors in a different voice. ZenWalk is a standalone walking meditation app for which we recorded all audio ourselves, since inserting additional prompts from another voice would be disruptive to the experience. For detecting user actions and extracting sensing data, we wrote algorithms that detect objects for interactions involving sprinting by and circling an object, which we will present in the Data Collection Experiments section.

USER STUDIES

We conducted two user studies to 1) examine whether habit-sourcing apps are as enjoyable or more enjoyable than their standard habit-building counterparts without added interactions, and 2) evaluate the accuracy of the data collected from cued interactions. In the following subsections we present our interaction user studies and data collection experiments.

Interaction User Studies

Setup

We conducted within-subject user studies to compare two versions of each app: *interactive* and *non-interactive*. For Zombies Interactive, the interactive version included three interactions (sprint to fire hydrant, sprint to the tallest building, and sprint past ten trees) inserted into the narrative, and the non-interactive version included only the narrative. For ZenWalk, the interactive version included one interaction (circling a large tree) along with standard meditation guidance (e.g. focus on your breath), and the non-interactive version included only standard meditation guidance. The chosen interactions resulted from our design process. Participants were counter-balanced to use either the interactive version or the non-interactive version first, and they were asked to complete both versions within a week of each other. Each Zombies Interactive run lasted around 40 minutes, and each ZenWalk meditative walk lasted around 20 minutes.

After each participant completed both versions, they were asked to fill out a post-survey asking them about their level of experience with the habit, frequency of practicing the habit, their enjoyment of each version of the app and willingness to use similar apps in the future on a 5-point Likert scale. They were also asked to answer open-ended questions about what they liked and disliked about the experiences, as well as any external factors that affected their experiences. Participants were compensated with a $25 gift card.

For Zombies Interactive, we recruited 12 participants (10M, 2F) from running subreddits, whose ages ranged from 18 to 42 (mean: 29.92, SD: 8.90). For ZenWalk, we recruited 9 participants (5F, 4M) from meditation subreddits, whose ages ranged from 21 to 49 (mean: 31.11, SD: 8.07). Recruiting in related subreddits allowed us to reach participants resembling our target users (i.e. people interested in or already practicing the habit). 8 out of 12 Zombies Interactive participants said they currently use running apps, including MapMyRun, Strava, and Runkeeper. Two participants had previously used the Zombies, Run! app. 5 out of 9 ZenWalk participants said they currently use meditation apps, including Calm, Headspace, OMG. I can Meditate!, Stop Breathe & Think, and Insight Timer.

Results

Zombies Interactive participants found the interactive version to be more enjoyable and are more likely to use apps similar to the interactive version than the non-interactive version (see Figure 9). The mean enjoyment rating was 3.83 for the interactive version and 3.0 for the non-interactive one (Figure 9, left). A Wilcoxon signed-rank test shows a significant difference between the enjoyment ratings for the two versions ($Z = -2.08$, $p = 0.04$, $r = 0.60$). The mean likelihood of future use rating was 3.08 for the interactive version and 2.17 for the non-interactive one (Figure 9, right). A Wilcoxon signed-rank test shows a significant difference between the ratings of

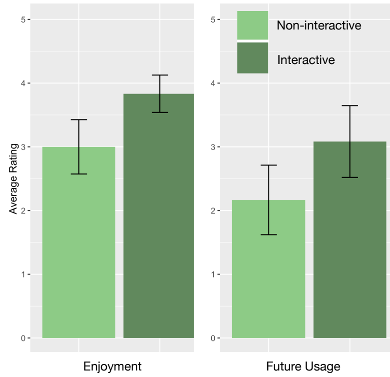

Figure 9. Zombies Interactive participants' ratings of enjoyment (left) and potential future usage (right) on a 5-point Likert scale.

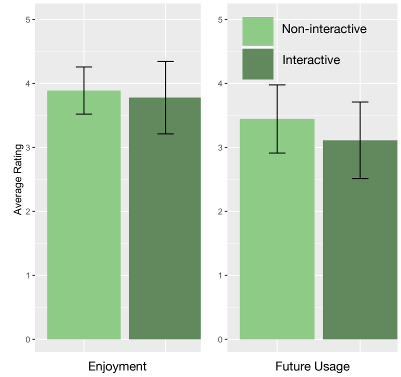

Figure 10. ZenWalk participants' ratings of enjoyment (left), potential future usage (right) on a 5-point Likert scale.

future use likelihood for the two versions (Z = -2.22, p=0.04, r = 0.64).

ZenWalk participants found the interactive version as enjoyable as the non-interactive one and are about as likely to use apps similar to the interactive version as the non-interactive one (see Figure 10). The mean enjoyment rating was 3.78 for the interactive version and 3.89 for the non-interactive one (Figure 10, left). A Wilcoxon signed-rank test shows no significant difference between the enjoyment ratings for the two versions (Z = 0.19, p = 0.82, r = 0.06). The mean likelihood of future use rating was 3.11 for the interactive version and 3.44 for the non-interactive one (Figure 10, right). A Wilcoxon signed-rank test shows no significant difference between the ratings of future use likelihood for the two versions (Z = 0.93, p = 0.53, r = 0.31). Additionally, ZenWalk participants were asked to rate how relaxing their experience. A Wilcoxon signed-rank test again shows no significant difference between the non-interactive (mean: 3.89) and interactive (mean: 3.67) versions (Z = 0.56, p=0.76, r = 0.19).

What did people enjoy about the interactive version?
Zombies Interactive participants felt more immersed in the story in the interactive version of the app. One participant commented: "I like the prompts to speed up at various places in the run. Made the experience seem more personalized and interactive. I felt more like a part of the storyline." They also found the the interactions fun and entertaining: "I enjoyed the frequent commentary; it helped the run go by quickly. I also liked the pretend situations which encouraged me to increase my pace. Sometimes, the story line seemed a little cheesy, but overall it was good." The participants also mentioned that having more specific goals motivated them to keep up their pace: "[In the interactive version], I think most the original audio was still there but with additional conversation added. I liked this better for two reasons. First it was more engaging. Second, the audio was more specific in terms of goals. Instead of things like 'run fast' it was more specific 'run past 10 trees' or 'run to the fire hydrant"'

For ZenWalk, the participants mentioned that they enjoyed connecting with nature. One participant commented: "My favorite part was when I focused on the trees. Just focusing on the different characteristics of the tree, and how it 'just was' was surprisingly very calming and even reassuring. Overall, I felt very relaxed, my mind felt like it was free of clutter, and I felt more energized and connected to the world in front of me." The participants also mentioned that they felt more aware of their surroundings: "I liked that I began to see things in a new way than I previously did, sparking my curiosity and noticing and appreciating the smaller things around me."

What did people dislike about the interactive version?
Zombies Interactive participants disliked interactions that did not make sense given their current location: "The instructions were a little weird, saying that I might see a tall building (I'm in NYC, tall buildings everywhere) or that I had to avoid trees (I was running in a park full of trees). It would be cool if there were actual GPS-based landmarks but that seems like it would take quite a lot more effort to implement." The participants who reported that they were more experienced with running tended to feel constrained by the interactions, since the interaction asked them to do things that differed from what they had planned for themselves: "I liked the additional run directives, but I would like the ability to turn them on and off, like the zombie chases in Zombies, Run!, because some days I am looking to hold a constant pace."

In ZenWalk, some participants felt self-conscious performing the interactions when other people were around: "In terms of the environment, being surrounded by trees and focusing on the trees made the walk very pleasant, whereas being surrounded by mostly houses and/or around people that I know (that might be judging me if I'm focusing on a tree or wanting to do a 'stop and chat') tended to halt the effects of the meditation." They also felt distracted from the meditation when they were not able to complete the cued action: "I couldn't circle the tree in the first walk and was near houses on a residential street in a neighborhood with lots of trees. The fact that I couldn't complete the instructions was distracting. Once I knew where the ideal environment would be, I chose a more suitable location and had a better experience."

Data Collection Experiments

Setup

We conducted data collection experiments to understand the effectiveness of our interactions for sensing the environment. We separated data collection experiments from interaction studies in order to recruit (a) representative users via Reddit to best study the effectiveness of our designs on the user experience and (b) local users so we could validate data in-person. We chose to validate data in-person rather than rely on Google Street View because sensed objects were often obstructed by other objects (e.g. cars) or missing from the map (e.g. on unmapped park trails).

As the focus of our data collection experiments is to study the effectiveness of interactions with regard to data accuracy rather than user experience, we used shortened 10-minute experiences that allowed us to more readily recruit local users who may not have wanted to participate in a full 30-minute experience. We used the same interactions as in the user study, expecting that some interactions would be more effective for data collection than others. For Zombies Interactive, we tested three sprinting interactions that respectively collected fire hydrant locations, tall building locations, and estimated tree counts between two locations. For ZenWalk, we tested one interaction, walking around and observing an interesting tree, that collected tree locations. Each run contained all three Zombies Interactive interactions in randomized order, and each walk contained three occurrences of the tree circling interaction.

We recruited 9 Zombies Interactive participants (5 female, 4 male) and 9 ZenWalk participants (7 female, 2 male) from university Facebook groups and mailing lists. Each participant received a $10 gift card as compensation. We excluded a ZenWalk participant's data due to GPS inaccuracy. We also excluded a Zombies Interactive participant's tree trace because he paused the experience, preventing us from continuing to correctly trace data.

Object Detection Algorithms and Measures

The Zombies Interactive interactions cue users to sprint past an object and to return to regular pace thereafter. To detect object locations, we use a heuristic algorithm that looks for speed readings above a user's average pace (i.e. when they are sprinting) that are followed by a drop in speed toward average pace (i.e. the point where they slow down from sprinting). From each data point in a user's speed trace within the interaction window, we identify the largest contiguous region during which the user's speed at any point in time is slower than at least one of the two previous data points. This identifies regions in which the user's speed is generally decreasing; we then filter out regions shorter than 3 seconds to avoid false readings and thus potential false positives. From the remaining regions, we identify those for which the maximum speed is above average pace and the minimum speed is below average pace. If no regions remain, the algorithm returns that no object is found. Otherwise, we choose the region with the sharpest drop in speed between the maximum and minimum speed points by comparing slopes across regions, and return the location associated with the maximum speed point.

The ZenWalk interaction we test cue users to circle around trees and to keep walking once the interaction is finished. To detect tree locations, we use a heuristic algorithm that looks for a point that is surrounded by the densest set of location readings from the user's GPS trace. To do this, we first construct a bounding box around all readings from the user's GPS trace. We iterate through points across the bounding box (discretized at a 3-meter interval) to find the point with the highest number of location readings within a set distance. We set this distance at 3 meters, which is far enough to capture all location readings from when a user may have been circling the tree. We return the location of this point as long as there are at least μ+2σ surrounding location readings (to avoid false positives). Otherwise, the algorithm determines that no object is found.

We evaluated the effectiveness of our approach for sensing the presence of objects by measuring how well our algorithms recognize when a user has correctly performed a cued interaction. As ground truth, we collected self-reports of whether the user correctly performed the interaction through a post-study survey (e.g., "did you complete the 'Increase speed until the fire hydrant' task?"). The question directly references the object, so we assumed that user reports of completing an interaction indicate that the object was there. To understand failure cases, in the same survey we asked users who did not complete an interaction for reasons why they did not or could not (e.g. "If you did not complete the the task, please elaborate on why you didn't or weren't able to.").

We then evaluated the effectiveness of our approach for accurately locating sensed objects by comparing their actual locations to the locations detected by our algorithms. Two of the authors visited these objects' actual and detected locations and measured the distance in between the two locations. We also constructed a binary measure of whether detected locations were accurate, using the following correctness criteria for each object type:

- **Tree location.** There is a tree within 6 meters of the detected tree location and the tree also matches user description.
- **Fire hydrant location.** There is a fire hydrant within 30 meters of the detected fire hydrant location. This measure captures fire hydrants across the street from the detected location.
- **Tall building location.** There is a tall building within 30 meters. This measure captures situations where buildings are across the street from the detected location.
- **Estimated tree count.** There are greater than 5 and fewer than 15 trees between the locations where the user started and stopped sprinting. We only count trees that are within the first row of trees on either side of the sidewalk.

Results

Our algorithm is accurate at recognizing when people performed the cued interactions. As Table 1 shows, our approach accurately detected 15 out of 16 interactions in ZenWalk and 19 out of 20 interactions in Zombies Interactive.

Detected locations are mostly accurate for ZenWalk but less so for Zombies Interactive. In ZenWalk, among the 15 true positive cases where our algorithm detected object locations and users reported that they did the interactions, 12 detected locations met the correctness criteria and were close to the

Interactions	TP	FN	TN	FP	Total
Tree (ZW)	15 (12)	1	7	1	24
Tree count (ZI)	6 (3)	1	0	1	8
Tall building (ZI)	6 (3)	0	0	3	9
Fire hydrant (ZI)	7 (4)	0	0	2	9
Total	34 (22)	2	7	7	50

Table 1. This is confusion matrix of our algorithm detecting whether user correctly performed interaction with ZenWalk (ZW) and Zombies Interactive (ZI). Numbers in parentheses under true positives (TP) indicate the number of objects detected whose inferred locations met the correctness criteria.

nearest actual tree locations, with a mean distance error of 4.35 meters (SD: 1.25). In failure cases, the mean distance to the actual trees was 8.1 meters (SD: 0.9) and therefore filtered by the correctness criteria. In Zombies Interactive, among 7 true positives for fire hydrants, 4 detected locations met the criteria, with a mean distance error of 18.75 meters (SD: 3.53). Among 6 true positive cases for buildings, 3 detected locations met the criteria with a mean distance error of 26.56 meters (SD: 10.71). 3 out of 6 true positive cases for tree count interactions met the criteria with a mean error of 2.66 (SD: 0.47). Figure 11(a–c) shows successful examples for interactions that resulted in the detection of a tree, a fire hydrant, and a tall building.

Figure 11(d–f) shows examples of failure cases from Zombies Interactive. Failure cases were more common in Zombies Interactive, mainly due to challenges in performing the interactions in user's surroundings or to unclear instructions. For example, in Figure 11(d), one participant was running in an urban area with very few trees, so he wasn't able to sprint past 10 trees. During our data validation, we also found that in tree-dense areas, it was unclear which trees the user counted as being passed, as there were many possible trees at varying distances from the sidewalk. In our data validation, we counted trees within the row of trees nearest to both sides of the sidewalk. Users may have counted trees differently in reality. In one fire hydrant failure case, the nearest fire hydrant was across the street from the runner's path, on which the detected fire hydrant was located, as shown in Figure 11(e). The interaction instructions may not have been specific enough in how close to the fire hydrant the user should be, so that some users thought they had completed the interaction as long as the object they sprinted past was within sight. In terms of tall buildings, many participants were running in residential areas, making it difficult for them to sprint to a building taller than 3 stories, as Figure 11(f) shows.

Our results show that clear, specific instructions from interactions are important for correctly detecting user performed interactions. Because the ZenWalk participants were asked to perform distinct alternative actions based on whether or not they were able to find a tree, we were able to correctly detect 7 out of 8 cases when users didn't perform the interaction. Since Zombies Interactive users were not given alternative instructions, they assumed that there must be an object nearby regardless of whether or not they saw one and therefore sprinted regardless with the hope of reaching the

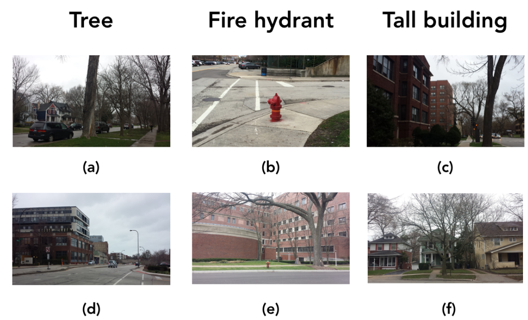

Figure 11. Successful examples of the interactions (a), (b), and (c) for data collection. Failure examples (d), (e), and (f).

presumed-present object. For such cases, since users did not actually complete the task, these detected sprints resulted in six false positives in our sprint detection algorithm.

DISCUSSION

In this paper we introduced habitsourcing as a novel concept for collecting environmental data from people's habit-building experiences. We introduced sensing through actuation as a technique for using cued interactions to elicit user actions that support the habit-building experience and also produce motion and activity traces from which we can extract environmental data. Through our user studies and data collection experiments, we found that habitsourcing apps are as preferred or more preferred than their habit-building app counterparts and, for some interactions, the inferred sensing data accurately represents the environment.

Our studies of Zombies Interactive and ZenWalk illustrate some of the challenges in both providing an engaging experience and collecting accurate data. While interactions in Zombies Interactive provided a more enjoyable habit-building experience for users, instructions for some interactions were not specific enough to provide data from which we can accurately elicit information about the environment. On the other hand, ZenWalk was not deemed more enjoyable than its habit-building counterpart, but provided data from which we were able to very accurately infer tree locations. In the rest of this section, we discuss design principles for habitsourcing that may help us to simultaneously collect useful data and enhance people's habit-building experiences. We then discuss design opportunities enabled by this work and future work to help realize them.

Design Principles

- *Keep interactions relevant to user's context, and provide guidance in fallback scenarios.* Interactions should be robust to variations in a user's context, regardless of their current proximity to an object. Providing alternative instructions can help to ensure that the user enjoys the interaction even if they are too far away from an object to perform a data collection interaction. This also ensures good data. For example, some Zombies Interactive users were confused when they were instructed to sprint to a fire hydrant when there was not one near them. In subsequent designs, we sought to reduce ambiguity by providing alternative instructions such as "If you don't see a fire hydrant, jog in place until we tell you it's safe to move on." In early tests, we found that this improves user enjoyment and also allows researchers to identify when a user takes an alternative action and ignore the data for that interaction, thereby avoiding potential false positives.

- *Design interactions with social context in mind.* Interactions should not put the user in a situation that is socially uncomfortable for them or those around them. Users often feel less comfortable performing actions that seem awkward when other people are around. For example, a ZenWalk tester reported feeling uncomfortable testing an interaction that instructed him to circle trees while on a busy street with many people around. To address this, we subsequently tested interactions that are less conspicuous, such as asking a user to stand at a tree and walk around it only if they want to. We found that these interactions are more enjoyable and can still provide data.
- *Prioritize habit-building for long term data collection.* Interactions should not disrupt the user's habit-building experience but rather enhance it. Data collection should come as a byproduct of enjoyable experiences offered by the interactions. We can afford to avoid interactions that produce good data yet are disruptive, since even though each user may only contribute a couple of data points per interaction, the benefit of the habitsourcing approach is that data collection can be slow and steady. Because people regularly practice their habits, we can collect data over time and avoid potentially burning out volunteer interest. For example, many Zombies Interactive users disliked an interaction that required them to stop and gesture with their device to report a tall building because it affected the momentum of their run. We therefore discarded this interaction, despite it providing data that was easy to analyze.
- *Match experience and preferences for users.* Interactions should be personalized so that they match the preferences of different users. For example, some ZenWalk users wanted to stay and observe a tree for longer than they were instructed to, while others would have preferred to move on to the next portion of the meditation sooner. To customize the experience according to different user preferences, we are exploring the use of *dynamic moments*, which respond in real time to user activity to allow users to interact in varied ways and for varying durations, thereby increasing user control and enjoyability. For example, to allow a ZenWalk user to stay at a tree for as long as they want, we can instruct users to look at a tree and to continue walking whenever they are ready. When the app detects that the user has begun to walk again, it can trigger the instructions for the next portion of the meditation automatically.

Enabled Design Opportunities

Habitsourcing allows us to potentially harness the habit-building practices of millions of people by supporting their practices while also using their interactions with objects in the physical environment to collect sensing data that would otherwise be difficult to gather. This allows us to tap into the efforts of users who may not care about collecting data for citizen science or communitysensing, yet are still motivated to take an action that produces valuable data as long as it immerses them in ways that help them build their habit.

While we focused in this paper on simple interactions for detecting common, static objects that are easy to verify and present in most environments, habitsourcing apps can also be used to collect more richly detailed, dynamic data of objects and events that may only be situationally available. First, habitsourcing apps can be more aware of the user-specific context. For example, apps can consider the physical location of the user so that it could collect suitable data, e.g., if the user is in downtown we may seek data about tall buildings and not ask users to circle around a tree. Second, apps can be more aware of the situational context. For example, apps can be aware of seasonally or temporally occurring events, and collect information with tailored interactions (e.g., sprint to the nearest food truck, a blossoming tree, or a street performer). Third, apps can be more aware about objects in the world. For example, apps can prompt users who are nearby a known object to interact with it so as to collect more information about the object. As one approach, we can use alternative instructions to draw users toward the object only if it exhibits certain additional characteristics (e.g., stop at the tree ahead if its leaves are needle-like; otherwise, just keep walking). By detecting which action the user takes, we can infer additional properties of objects which can help to classify them (e.g., whether the tree is coniferous).

While our user studies considered only short-term deployments, the immersive nature of habitsourcing apps has the potential to engage users long-term. In our interaction study, Zombies Interactive users noted that being able to interact with real world objects increased their enjoyability and likelihood of using similar apps in the future. Since users are likely to interact with different objects on each run or walk, habitsourcing apps may further promote longer-term use by keeping experiences fresh. Advancing designs that incorporate user-specific context, situational context, and knowledge of the physical environment can further promote long-term use with experiences that are appropriate, engaging, and relevant to habit-building. In future work we are interested in continuing to advance designs that promote long-term engagement, and to explore concrete use cases so that government agencies, entrepreneurs, and researchers may benefit from the collected data.

Future Work

Future work on habitsourcing should seek to advance both user and data collection goals. To improve the user experience, we are interested in designing more personalized, context-aware interactions that are sensitive to users' experience, previous usage, and personal goals; and that provide experiences made possible by the user's current location and surrounding context. To advance the fidelity of the collected data, we are interested in studying mechanisms for scaffolding user contributions. For example, in order to go beyond object location data to collect detailed attributes of objects, we are exploring interactions that make use of detected object locations to promote further data collection. Beyond habitsourcing, we are generally interested in the direction of *affordance-aware computing*, that by using collected knowledge of the environment, enables new opportunities for immersive interactions that advance our interest and ability to interact with the physical world.

Acknowledgements
We thank all of our participants for testing our apps and providing valuable feedback. We also thank members of the Design, Technology, and Research program and the Delta Lab for helpful discussions in the early stages of this project.

REFERENCES

1. Aoki, P. M., Honicky, R., Mainwaring, A., Myers, C., Paulos, E., Subramanian, S., and Woodruff, A. Common sense: Mobile environmental sensing platforms to support community action and citizen science. In *Ubicomp* (2008).
2. Bao, L., and Intille, S. S. Activity recognition from user-annotated acceleration data. In *Pervasive computing*. Springer, 2004, 1–17.
3. Bell, M., Reeves, S., Brown, B., Sherwood, S., MacMillan, D., Ferguson, J., and Chalmers, M. EyeSpy: Supporting Navigation Through Play. In *CHI* (2009), 123–132.
4. Bonney, R., Cooper, C., Dickinson, J., Kelling, S., Philips, T., Rosenberg, K., and Shirk, J. Citizen science: A developing tool for expanding science knowledge and scientific literacy. *BioScience 59* (2009), 977–984.
5. Brossard, D., Lewenstein, B., and Bonney, R. Scientific knowledge and attitude change: the impact of a citizen science project. *International Journal of Science Education 27* (2005), 1099–1121.
6. Conrad, C. C., and Hilchey, K. G. A review of citizen science and community-based environmental monitoring: issues and opportunities. *Environmental monitoring and assessment 176*, 1-4 (2011), 273–291.
7. Dow, S., Gerber, E., and Wong, A. A pilot study of using crowds in the classroom. In *CHI* (2013), 227–236.
8. Faste, H., Rachmel, N., Essary, R., and Sheehan, E. Brainstorm, chainstorm, cheatstorm, tweetstorm: new ideation strategies for distributed hci design. In *CHI* (2013), 1343–1352.
9. Fogg, B. J. *Persuasive Technology: Using Computers to Change What We Think and Do*, 1st ed. Morgan Kaufmann, 2002.
10. Hui, J. S., Gerber, E. M., and Dow, S. P. Crowd-based design activities: helping students connect with users online. In *DIS* (2014), 875–884.
11. King, S. F., and Brown, P. Fix My Street or else: Using the Internet to Voice Local Public Service Concerns. In *ICEGOV* (2007), 72–80.
12. Matyas, S., Matyas, C., Schlieder, C., Kiefer, P., Mitarai, H., and Kamata, M. Designing location-based mobile games with a purpose: Collecting geospatial data with cityexplorer. In *ACE* (2008), 244–247.
13. Randell, C., and Muller, H. Context awareness by analysing accelerometer data. In *The Fourth International Symposium on Wearable Computers* (2000), 175–176.
14. Ravi, N., Dandekar, N., Mysore, P., and Littman, M. L. Activity recognition from accelerometer data. In *AAAI* (2005), 1541–1546.
15. Rula, J., and Bustamante, F. E. Crowd (Soft) Control: Moving Beyond the Opportunistic. In *HotMobile* (2012), 1–6.
16. Sullivan, B. L., Wood, C. L., Iliff, M. J., Bonney, R. E., Fink, D., and Kelling, S. eBird: A citizen-based bird observation network in the biological sciences. *Biological Conservation 142*, 10 (Oct. 2009), 2282–2292.
17. Truong, K. N., Shihipar, T., and Wigdor, D. J. Slide to x: Unlocking the potential of smartphone unlocking. In *CHI* (2014), 3635–3644.
18. Tuite, K., Snavely, N., Hsiao, D.-y., Tabing, N., and Popovic, Z. Photocity: training experts at large-scale image acquisition through a competitive game. In *CHI* (2011), 1383–1392.
19. Vaish, R., Wyngarden, K., Chen, J., Cheung, B., and Bernstein, M. S. Twitch crowdsourcing: crowd contributions in short bursts of time. In *CHI* (2014), 3645–3654.
20. Waze, 2013.
21. Zimmerman, J., Tomasic, A., Garrod, C., Yoo, D., Hiruncharoenvate, C., Aziz, R., Thiruvengadam, N. R., Huang, Y., and Steinfeld, A. Field Trial of Tiramisu: Crowd-sourcing Bus Arrival Times to Spur Co-design. In *CHI* (2011), 1677–1686.