Robots, Pancakes, and Computer Games: Designing Serious Games for Robot Imitation Learning

Benjamin Walther-Franks[1], Jan Smeddinck[1], Peter Szmidt[1], Andrei Haidu[2], Michael Beetz[2], Rainer Malaka[1]

[1]Digital Media Group, [2]Artificial Intelligence Group
TZI, Universität Bremen

{bwf, smeddinck, piotr, haidu, beetz, malaka}@tzi.de

Figure 1: Many robot manipulation tasks (left) are learnt best by observing humans. Human motion can be demonstrated using virtual environments (center). Data acquisition with motion-based games makes demonstrating a fun activity (right).

ABSTRACT

Autonomous manipulation robots can be valuable aids as interactive agents in the home, yet it has proven extremely difficult to program their behavior. Imitation learning uses data on human demonstrations to build behavioral models for robots. In order to cover a wide range of action strategies, data from many individuals is needed. Acquiring such large amounts of data can be a challenge. Tools for data capturing in this domain must thus implement a good user experience. We propose to use human computation games in order to gather data on human manual behavior. We demonstrate the idea with a strategy game that is operated via a natural user interface. A comparison between using the game for action execution and demonstrating actions in a virtual environment shows that people interact longer and have a better experience when playing the game.

The definitive version was published as:
Walther-Franks, B., Smeddinck, J., Szmidt, P., Haidu, A., Beetz, M., & Malaka, R. (2015). Robots, Pancakes, and Computer Games: Designing Serious Games for Robot Imitation Learning. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 3623–3632. https://doi.org/10.1145/2702123.2702552

Author Keywords

Programming by demonstration; human computation games

ACM Classification Keywords

H.5.2 User Interfaces; H.1.2 User/Machine Systems; I.2.9 Robotics; K.8.0 [Personal Computing]: General - Games.

INTRODUCTION

HCI has moved from focusing mainly on desktop interaction to ubiquitous interaction and in particular to embodied computer artifacts. With the advent of robots in our homes, computation is moving beyond the classical understanding of human-computer interaction. The future of household robots envisions active and intelligent companions in our everyday lives with many concepts exploring anthropomorphic robots like the NAO robot [24]. Researching the humanoid form factor is important, since it is likely to be accepted by humans when they are to "live" in the same environment [13]. Besides the familiar appearance of the robot, human-like robot behavior fosters human-robot interaction and cooperation, as it requires less adjustment and learning effort by the human. Even though the degrees of freedom of the robots' limbs and joints may allow for other non- or even super-human movements, it is often a better choice to let robots act in a way that corresponds to the mental models of their users. However, programming robots to master everyday activities in a human-like fashion has proven to be a huge challenge [3]. Today, robots can only solve tasks in a narrow range of

conditions. Many seemingly mundane everyday tasks are only vaguely described and show a high level of variation. The artificial intelligence vision of understanding common sense knowledge needed for everyday activities has been unsolved for decades [19].

In recent years, prototypical solutions for household tasks have been built, for instance robots making pancakes [3] or baking cookies [5]. The necessary knowledge for solving such tasks in various environments has to be fed into the systems through knowledge acquisition and machine learning in labor- and cost-intensive processes. Since most humans are experts at such tasks, there is great potential in acquiring this knowledge for robots from human sources. If people perform the required movements, such as those for making a pancake, they can be tracked, analyzed, and then used to build models for the motion of the robot's arms and hands [20,23]. In this imitation learning approach to robot programming, the challenge lies in collecting large amounts of movement data from humans solving some problem while conditions vary systematically. Obviously this is tedious work. It is difficult to motivate people to move pans on a stove over and over again with all sorts of pan sizes and stove heights.

Using virtual environments in which the user interacts with simulated task contexts, human motion data can be generated by recording user input [16]. The advantage being that a virtual world is fully observable and demonstration is unconcerned with real world constraints or safety issues. Games can increase the motivation for performing menial tasks in such environments, potentially increasing the number of demonstrations and thus the amount of -and variation in- motion data. They also offer further benefits for imitation learning: in games, objectives can be introduced to influence players' behavior, such as eliciting their best performance. Moreover, well-balanced challenge in games increases the players' resistance to frustration. This ability to design for task failure is highly important for robot programming, and games are an easy and controlled means of eliciting data on failing behavior.

In this paper we present a human computation games approach to knowledge acquisition tools for imitation learning. Engaging people as experts of everyday activities to perform actions in an embodied game raises the motivation of demonstrators and encourages them to deliver high quality data. Freedom in game design allows for tailoring player behavior to generate desired outcomes and allows obtaining varied human motion data from successful or, equally important, unsuccessful actions.

We present a sample application using a tower defense game with full-hand motion input. The users were highly motivated to play and thus contribute their embodied knowledge. With this setup, the data acquisition for programming a robot is translated to an HCI problem of game interaction and player experience. Our study shows 1) that performing tasks in-game provides a better self-reported experience than demonstrating them in a non-game virtual environment and 2) that motion data has the quality required for successful use in robot imitation learning

With this paper we not only contribute a successful design of a free-hand motion control serious game but also demonstrate a design approach of how to leverage existing game design knowledge for increasing the effectiveness of motion data acquisition tools. Moreover, this paper is a contribution to human-robot interaction for humanoid robots in home environments. We propose to use *gameful interaction* [8] as a means for building more realistic and acceptable robotic companions.

RELATED WORK

Robot Programming

Imitation is the behavior where an individual observes and reproduces another's action. *Imitation learning* or robot *Programming by Demonstration* (PbD) [4] (also called *Learning from Demonstration* [2]) is a means of learning and developing new skills by observing how they are performed by others. It is a technique for enabling robots to autonomously perform new tasks. When observing either success or failure examples one can reduce the complexity of search spaces for learning, by either beginning the search from the observed successful solution (local optima), or by eliminating the failures from the search space.

The PbD learning problem can be separated into two fundamental phases: *gathering* examples and *deriving* policies [2]. Focusing on the gathering phase there exists a large variety of techniques for executing and recording demonstrations, ranging from human operators moving robots around which then record their own enacted movements over recording teleoperation commands to observing demonstrators execute behaviors with their own body. Furthermore, the control levels may vary from low-level actions for motion control, over basic high-level actions (action primitives) to complex behavioral actions.

In the work of Haidu et al. [11] the authors learn a failure detector model to allow a robot to recognize the point where the current action will lead to a failure. The data used for learning is collected from a virtual environment by running multiple episodes (success and failure cases) of the given action. The virtual environment built on a robotic simulator with a realistic physics engine, resulting in precise, realistic data. Having a virtual scenario gives the advantage of a fully observable world; one can completely represent the motion and the states of all simulated objects. Even if the motion sequences are comparatively simple, they can contain valuable information (e.g. for low-level motion control).

Other researchers have also been employing simulation environments [25]. While the results in terms of accuracy of the gathered data are promising, these simulators have not been designed to appeal to the general public, to be operated by naïve users, or to support motivation by

introducing game mechanics beyond the playful appeal of an interactive virtual world. In order to gather large amounts of varied data that can be used to build large-scale behavior models, such simulations must be able to run on consumer hardware and feature motivational structures to animate players to perform multiple actions.

Human Computation Games

Introducing playful and gameful elements [8] to collective intelligence applications has been a hot topic in a new emerging field over the last years: *Human computation* (HC) *games* (HCG) are a form of collective intelligence where participants are not paid but voluntarily contribute data or chunks of knowledge in game-based interaction. HC is similar to crowdsourcing, where traditional human workers are replaced with members from the public, but differs in that in HC human computers (usually also from the public) replace digital computers in domains where these struggle [21]. The first prominent examples of such *games with a purpose* [27] were used for image labeling tasks. It was shown that very simple casual games motivated a large number of users and that it was possible to aggregate their individual contributions into concerted collective intelligence. In recent years, human computation games have been developed for further labeling tasks [14], computational biology [6], question answering [1], and for many other use-cases [28].

Human Computation Games for Robot Programming

In an effort to structure human computation game design approaches, Krause and Smeddinck [14] name four categories: *Identification* (of a task that can be turned into a HC application), *Motivation* (design mechanisms for motivating members of the public to contribute time and effort), *Observation* (design methods for enacting solutions in the application and for observing the approach taken by each member of the crowd) and *Evaluation* (design methods for aggregating, analyzing and interpreting the potentially large numbers of crowd contributions); or IMOE in short. Notably, this structure is akin to the general steps of gathering (IMO) and deriving (E) in PbD. Accordingly, we argue that PbD is a special case of human computation and thus approaches from human computation games are likely applicable to PbD. Both communities can profit from research that aims at combining their approaches that have so far been discussed in almost entirely separate sub-communities. While crowdsourcing has established itself for PbD, it is rarely embedded in game contexts (see [10] for a recent overview).

Looking into the categories of typical HC tasks as part of the considerations on *identification*, Krause and Smeddinck describe four categories, each one taking advantage of a specific human ability: aesthetic judgment, making intuitive decisions, contextual reasoning, and free interaction with the physical world (mentioned as *embodiment issues*). The latter offers a matching category for the application domain of the study presented in this paper (harnessing hand movements and actions). Embodied human knowledge is implicitly at hand for humans but hard to be formalized explicitly, making this category a strong candidate for human computation. So far, however, most human computation games do not consider embodied human knowledge that is used in physical interactions with the environment. Some approaches exist in mapping applications [18,26]. However, they require substantial physical effort by all human contributors and capture a gross measure of human body location instead of intricate details of human embodied interaction.

Approaches from related areas, such as gamification, have investigated matching game mechanics to generalized tasks. Flatla et al. [9] tackle gamification for calibration tasks such as screen calibration or respiration chest strain sensor calibration. They developed a design framework for identifying calibration types and the basic tasks behind them, and how to match these tasks to basic game mechanics. Using this framework they designed three calibration games, and were able to show increased user motivation. They also discuss differences in the data quality between standard and gamified calibration. However, their approach has not yet been linked to the related area of PbD and they did not focus on embodied interaction.

DESIGN APPROACH: KITCHEN TOWER DEFENSE

The primary design goal is to motivate people to demonstrate manual tasks by transforming the experience of performing mundane manual actions (physical or simulated) into a more enjoyable activity. We lay out our strategy before discussing how target and source domains can be mapped.

Design Strategy

Our design strategy to reach this goal is situated toward the "Gaming" and "Whole" ends of the two dimensions Gaming/Playing and Parts/Whole of the design space [8].

Gaming vs. Playing

Playful tools for motion data acquisition based on simulated virtual environments have been explored in the robotics community [11,16]. They offer the benefits of a fully observable world and safety from real world constraints. However, embedding this in a rule-based game system as opposed to a mere playful environment has two distinct advantages. First, task demonstration can be designed into the game mechanics in order to pose tasks more implicitly, directed via game rules and level design. Second, the addition of game *aesthetics* [12] such as challenge and narrative can improve the experience; while the enjoyment of baking virtual pancakes will diminish over time, a well-designed game can captivate for hours.

Parts vs. Whole

Designing for the stated goal can involve either (whole) HCG or introducing game design elements (parts) into motion data acquisition tools. The latter can be achieved either by adding game elements to an existing application (gamification) or by adding purposeful elements to an

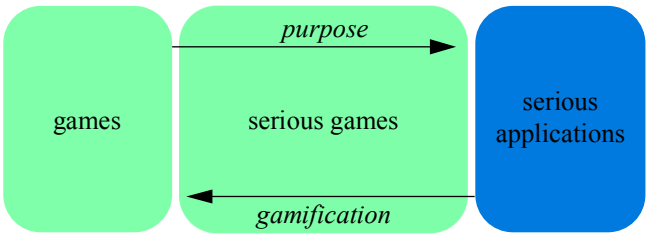

Figure 2: Serious games: increasingly purposeful games versus increasingly gamified applications.

existing game. This decision can be modeled as a third (if not fully orthogonal) dimension in the design space.

Games vs. Serious Applications

This axis reaches from one extreme, namely adding increasingly more game elements to a "serious" application, to an opposite extreme, which begins with full-fledged games that are augmented by an increasing number of "serious" elements (Figure 2). Our approach is situated toward the "games" end of this spectrum, in order to capture ordinary people and their sense of what constitutes a game. While gamification can be successfully used to increase motivation to perform otherwise boring or laborious tasks [9], we opted for creating a full game experience that uses established game mechanics for the following reasons: The approach enabled us to use whole game concepts to leverage existing design knowledge on working game mechanisms. Moreover, it increases the likelihood to create long-term motivation rather than short-lived engagement common in swiftly designed casual games. Furthermore, once a reliable game design is found for extracting a certain class of human behavior it can be used for a whole range of task domains, e.g. all workbench-based manipulation tasks.

Task Domain

Cooking is a good example of an embodied everyday activity. It includes complex and diverse manual activities that are easy to perform for (many) people but are hard to describe in a formal or algorithmic way. In robotics, cooking simulators are being investigated for knowledge acquisition [3,5,11,16]. We chose cooking as a paradigmatic scenario for an activity in the household and demonstrate how human computation games that go beyond the playfulness of simulator tools can help to gather data from ordinary persons who have no experience in robotics.

The first step in designing a game around a task domain is to identify the characteristic of the domain. We can identify four characteristics of cooking:

Characteristic 1: An enclosed, planar workspace

Characteristic 2: Well-defined manual actions

Characteristic 3: Actions operate on objects within the workspace

Characteristic 4: Active management of events

Note that these are valid for a number of worktop-based activities, such as crafts (e.g. sewing) or deskwork. The next step is to select an existing game archetype that matches these characteristics best.

Game Domain

The selection of a suitable game genre has to result in a good mapping from actions in the game domain to the desired actions in the task domain. Turn-based strategy (TBS) games reflect the above-mentioned properties very well. The player has a "godlike" position of a commander with an overview of a confined terrain or map. This relates well to *characteristic 1*, since players are able to see the available objects and the according contextual action space. TBS players perform various manipulation actions (relating to *characteristic 2*) that change the events enfolding on this map. Such actions can be recorded and their execution parameters and the contextual variables can be used for imitation learning. When enacted and chained to complex behaviors such as building up a base or city, or directing units, the manual actions operate on available objects (relating to *characteristic 3*). This results in information about workspace organization, object alignments and positioning. Lastly, in TBS, the player has control over the pace of the events unfolding via turn advancement. Relating to *characteristic 4,* the turns help with the challenging problem of chunking action sequences, which is needed for interpreting the overall user action sequences. Moreover, the game genre is suitable for embodied control with real hand gestures, which is important since we are interested in the actual trajectories of the player's arms and hands.

Other popular game genres, such as action or adventure, are focused on steering an avatar through expansive game-worlds (violating *characteristic 1*), and interaction with the game world occurs via the avatar as a mediator (violating *characteristic 2*) rather than the player manually interacting with it. Real-time games that are not turn-based lead to higher time pressure (violating *characteristic 4*), which could also have a negative influence on the resulting data.

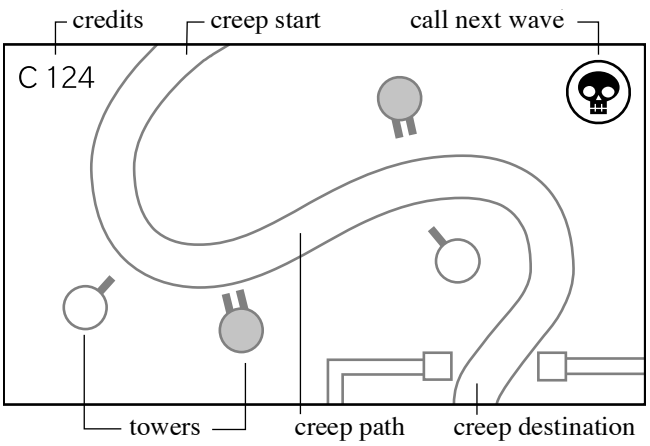

Figure 3: Schematic view of a typical level of a tower defense game with respective game elements.

intention	actions
build	*pick* type + *pick* site
upgrade	*pick* tower + *pick* upgrade
move	*pick* tower + *pick* site
demolish	*pick* tower + *pick* demolition

Table 1: General tower defense game intentions and actions.

intention	actions
place	*pick* pan + *place* on stove
add	*pick* bowl + *pour* batter
move	*pick* bowl + *place* bowl
turn	*pick* spatula + *lift* pancake + *flip* pancake

Table 2: Task intentions and actions in pancake making.

In this light we chose *tower defense games* (TD) as the game domain (Figure 3). In TD, players have to create and manage defensive structures against hordes of intruders invading their territory (also called *creeps*). These typically arrive in waves, whose arrival can be managed by the player to a certain degree. Game mechanics involve a simple (typically single-currency) economy in which construction costs are balanced against "revenue" earned by defeating enemies.

Mapping Actions

Having established a game genre that matches task domain characteristics, it is necessary to define the exact mapping from task actions to game actions. In order to arrive at a mapping we must consider the asymmetry in requirements: from the game domain we only require strict adherence to *intentions*, while a correct rendition of the task domain *actions* is crucial for gaining the right motion data. We thus need to map task domain actions to game domain intentions (Table 1). Since cooking is an activity for which a vast amount of behavior can be identified, we chose pancake making as a well-defined example also used in related research [3,11,16]. We assume a simplified procedure that uses ready-made pancake batter, reducing the task to few simple actions (Table 2). As there is no way of further operationalizing the mapping procedure, the last step is down to creative design. For the example of pancake making, we chose a mapping that relates action intentions semantically: building is conceptually related to moving things, an upgrade adds functionality, and constructions are demolished by turning them over (Table 3).

Control Interface

The goal is to tap the embodied knowledge on manual behavior that humans are not used to describing explicitly, but can better recall implicitly given a task context (e.g. describing the procedure of operating clutch, gearstick and accelerator versus performing the act of driving). This requires interfaces with as little mediation and indirection as possible. Task simulators for recording human motion thus employ virtual hand manipulation techniques using full-hand trackers. Since glove interfaces incur obtrusiveness, the best solutions for a broad audience are outside-in hand tracking systems, i.e. camera-based solutions. High-fidelity full-hand trackers have reached consumer level of maturity and support the prospects of success of outsourcing embodied manipulation knowledge acquisition through full-hand interaction games.

3DFEND: FREE-HAND CONTROL TOWER DEFENSE

As a result of these design considerations we developed *3DFEND*, the first free-hand 3D TD. The story of the game is a micro-narrative in a style found in many casual games (Figure 4). Players defend earth against hordes of robot aliens by constructing a defense system via hand motions. While input is 3D, the game follows the typical planar TD logic, i.e. creeps move along two-dimensional paths, along which towers can be placed. A large platform provides the stage on which towers can be constructed. Players operate on objects on the stage through tools, not directly via cursor. Creeps advance over the stage from left to right.

Game Controls

Free-hand control is implemented as a simplified virtual hand metaphor [7]. This takes the form of a 3D cursor to which palm translation and orientation are mapped directly. Fingers are not used, since robust low-latency tracking is not solved in the category of hand trackers under consideration. Balloon cords, shadows, and a map overlay provide depth cues. Tool selection occurs if the cursor collides with the tool before a certain timeout, which is animated with a blend to white. Figure 5 gives an overview of the interface configuration; please refer to the video figure for a more immediate illustration of the controls.

A 3D toolbar at the bottom of the screen offers three construction modules for three different tower types, a nano bot container, and a relocation/demolition tool. Tools are selected by the above procedure. Each of the four basic controls involves compound manual actions (Table 3).

Build Turret

The player can choose one of the three construction modules and place it on the stage: after a construction time of one second, the turret is ready. Constructing a turret costs different amounts of energy credits, depending on type.

intention	actions
build	*pick* construction tool + *place* on site
upgrade	*pick* upgrade tool + *pour* nano-bots
move	*Pick* demolish tool + *place* on new site
demolish	*pick* demolish tool + *lift* turret + *flip* turret

Table 3: The basic game controls of 3DFEND.

Far in the future, a wormhole opened in proximity to earth. A hostile alien race is sending troops through the wormhole. The nations of earth have sent you, the player, to construct a defense in space. As the enemy is entirely unknown, you must quickly match their strategy with specialized defense systems. Fortunately, destroyed aliens set free energy that you can harness and use to improve your defenses.

Figure 4: The game background story of 3DFEND.

Upgrade Turret
The player can choose the upgrade tool and pour *nano bots* on constructed turrets. Turrets upgrade continuously, depending on the amount of nano bots received. Upgrading costs energy credits.

Move Turret
The player can use the relocation tool to move a tower by using it like a spatula on constructed towers, once on the spatula they can be placed elsewhere on the stage.

Demolish Turret
In the same manner, lifting it with the demolition tool and turning it by more than 90° can demolish a turret. The player receives a percentage of the tower construction cost as energy credits.

A heads-up overlay provides data on the game state: current wave, lives remaining, credit costs, and resources. A 3D button on the top center can be used to call for the next wave when the player is ready.

Gameplay
Apart from the unique free-hand interaction described above, 3DFEND has the typical dramaturgy of a tower defense game: creeps move over the stage from left to right along a highlighted path. Players must build towers along the way that automatically fire at the creeps lest these reach the portal on the far right end of the stage, in which case the player loses life points. Building towers costs energy credits that can be replenished through harvesting destroyed enemies or demolishing unused towers. With each wave, creep health and number increases. Starting with the fifth wave, creeps gain a shield that increases in power with each wave. Bullet towers are ineffective against shielded enemies, while laser turrets diminish shields quickly but are ineffective otherwise. Rocket turrets diminish both enemy shield and health, but are slow and expensive. With increasing enemy numbers and power, players will need to upgrade towers to manage the incoming hordes. They will need to figure out the ideal configuration of tower count, position and level to defeat the incoming enemies.

Implementation
3DFEND was implemented using the Unity3D game authoring environment. For a hand tracker we used the *Leap Motion* sensor [17], which was integrated into Unity using the SDK plug-in.

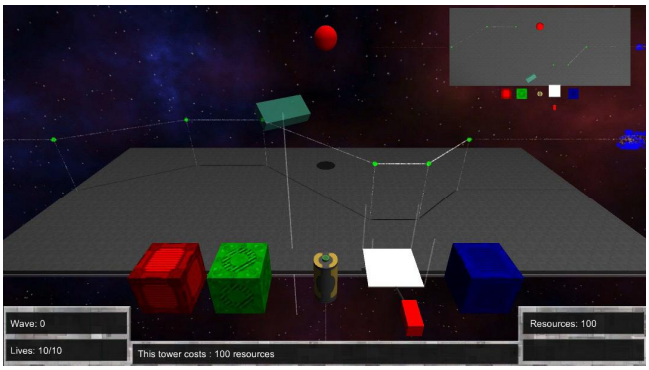

Figure 5: Screenshot of the 3DFEND game.

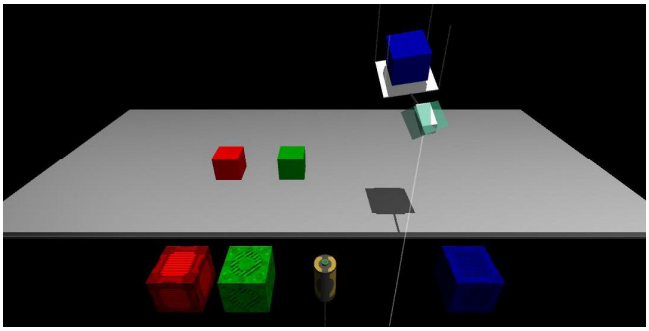

Figure 6: Screenshot of the virtual manipulation environment.

EXPERIMENT
We conducted a formal experiment to evaluate our claim that HCG can be used to increase motivation for demonstrating manual actions for robot imitation learning. Two hypotheses coin this statement:

H1. Demonstrating manipulation tasks in a game environment provides a better experience than following instructions in a playful environment.

H2. The manipulation data recorded in a game environment has at least the level of quality as motion data recorded in a playful non-game environment.

Study Design
The study design compares following explicit instructions to perform virtual manipulation tasks with performing the same tasks implicitly while playing a game. It uses a within-subjects design with an alternating order of treatments. The independent variable is *activity evocation type* with the conditions *virtual manipulation environment with instructions (VE)* and *game.*

For the *VE* condition, a virtual environment was created (Figure 6). It features the same basic manipulation controls but no game elements, namely: enemies, game overlay, or animations. Instructions were administered via text overlay. Participants were asked to place a cube, move a cube, pour on a cube, and flip a cube, and repeat this twice with varying locations. On completing these 12 actions the treatment was terminated. Attention was paid not to introduce any bias by framing (e.g. by using the words "game" or "virtual environment"): both conditions were introduced neutrally by declaring them as "prototypes".

The *game* condition consisted of playing a prototype of 3DFEND for at least 15 minutes. Pre-tests showed that twelve actions is a typical number for 15 min gameplay. Players were alerted when this time was over, but were allowed to terminate gameplay on their own accord.

Measures

To measure the subjective *player experience* we used the competence, autonomy, presence/immersion and intuitive controls subscales of the Player Experience of Need Satisfaction (PENS) questionnaire [22]. Since the scenario at hand focuses on single-player experience we dropped the relatedness scale. An interview complemented the assessment of the participant experience.

In PbD, data on demonstrating motions is used to build a model for robot behavior. It is hard to quantify what makes "good" motion data for PbD. Based on the experience gained in previous work [5,11,16], three aspects of motion data were identified to operationalize *data quality* for the purposes of assessing H2.

- A large amount of completed actions is a basic requirement for motion data for use in PbD; The VE condition as representative of state-of-the-art motion data acquisition was designed for a high frequency of demonstration samples.
- Cursor speed, height, and angle can serve as simple characteristics for comparing manipulations between this baseline and the proposed alternative.
- Spill in the pour action serves as an indicator for the precision of operation (accuracy was less consequential in the place and turn actions).

Participants

We recruited 16 participants (13 male, 3 female) from the academic milieu. Participant ages ranged from 24 to 32 (M =26.8, SD=2,32). Out of all participants, 11 considered themselves regular gamers (more than 1 hour of playing computer games a week) and all but one had experience with games and motion input in games.

Setup and Procedure

Participants sat at a table with a 24" screen and a Leap Motion sensor. First, participants were asked to fill out a demographics questionnaire. After introducing participants to the experiment procedure, they were trained on the basic controls (without instructions or game) until they felt competent and comfortable to proceed. Next they were exposed to both treatments in turn. After each treatment they were asked to fill out the questionnaire. Finally, a short interview was conducted.

Results

We used two-sided repeated measures Student's t-test on the collected data. Due to logging problems with the move action we only report on the three other actions.

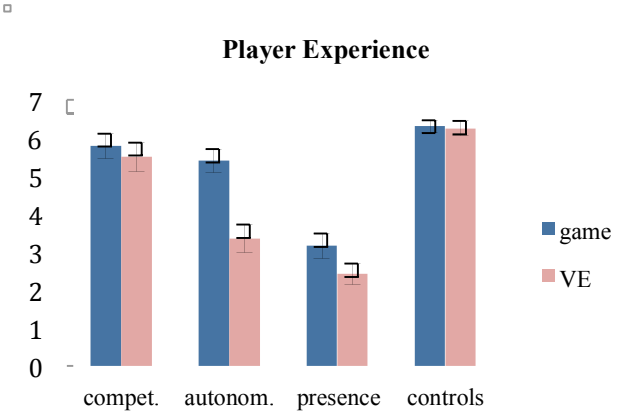

Figure 7. Player experience (PENS subscales). Error bars indicate the standard error of the mean (SEM).

Player Experience

Figure 7 displays score means and SEM of the four PENS scales for player experience. While no significant differences in the subjective assessment of *competence* ($p=0.27$) or *intuitive controls* ($p=0.69$) can be made out, both *autonomy* ($p<0.01$) and *presence/immersion* scored significantly higher for the game condition ($p=0.02$).

15 participants voluntarily extended their gameplay, on average playing the game for 30 minutes 26 seconds (VE instructions took 5 minutes 46 seconds on average).

Participant's comments showed a clear subjective preference of the game version. Some comments after the game condition were: *"I want to finish the round! What was the highest high score yet?"*, *"Where can I buy the game and where can I buy the sensor?"*, *"Can I play again?"*, *"The game is so much fun"*. Comments of the users after and during the VE condition were along the lines of: *"Do I have to do all of this?"* and *"What, again?"*.

Beside these general comments the participants also provided feedback on the graphics, the interface and the tracking system: The users would like to see more advanced and appealing graphics, more accurate hand tracking and grasping. Some of these improvements would certainly further improve the experience and usability of the system. However, the goal of our study was to establish 3DEFEND in an initial version with simple methods and tools.

action	game	VE
place	14,56	6,13
pour	16,63	4,94
turn	2,94	3,00

Table 4. Mean number of completed actions.

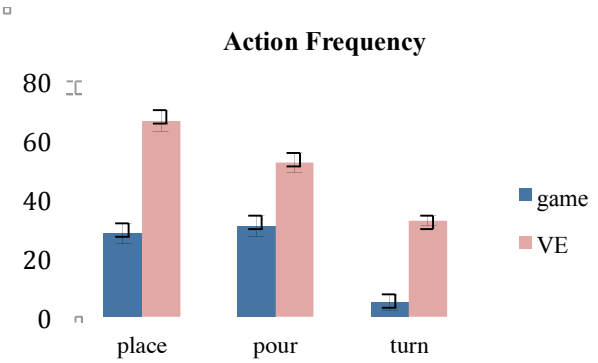

Figure 8. Action frequency of the main control actions (completed actions per hour). Error bars indicate SEM.

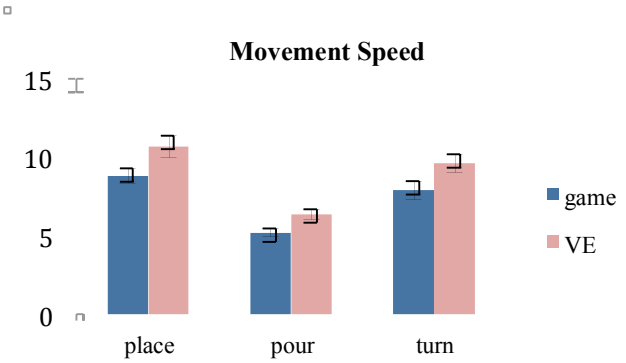

Figure 9. Movement speed in the main control actions (game units per second). Error bars indicate SEM.

Motion Data

Figure 8 shows the action frequency means extrapolated from the measured data. Significantly fewer occurrences can be measured for all actions under the game condition ($p<0.01$). The small number of turn actions under the game condition shows that game mechanics need better balancing so that these controls are more frequently used.

Figure 9 shows movement speed means per action. Movement was significantly slower for all actions ($p_{place}<0.01$, $p_{pour}<0.01$, $p_{turn}=0.01$) in the game condition. Pour height ($p=0.73$) and turn height ($p=0.15$) did not differ significantly between the conditions; neither did the pouring orientation ($p_x=0.62$, $p_y=0.36$, $p_z=0.69$).

There was significantly less content spilled in the pour action in the game condition than under instructions in the VE (5.4% versus 20.8%, $p=0.01$).

ANALYSIS

Our game provides a better experience regarding autonomy and presence. Combined with the observed motivation of participants, this supports H1 that demonstrating manipulation tasks in a game environment provides a better experience than following instructions in a playful environment. This has a direct effect on the amount of motion data that can be acquired. While games are less efficient than instructions (in our example only half as many actions per time range), the willingness of players to spend more time playing and thus demonstrating more than makes up for this: Despite lower frequency our game totaled the same amount or more actions (Table 4). The motion from both conditions had the same spatial characteristics; the same actions can be discerned from the data. However, in the game condition, manipulation speed was slower and more precise (for the pouring action). A likely explanation, corroborated by interview responses, is that players care more about the outcome of actions in the game. Such effects have been observed in other studies comparing user behavior in games with comparable non-game contexts [9]. The difference in spilling illustrates this best. When actions have outcomes that affect players this motivates them to act more carefully: Spilling valuable nano-bots wastes credits, while the VE did not penalize spill. Thus, regarding H2 we can assess that the quality of the motion data resulting from the game condition is at least the standard of the baseline VE condition. High player motivation and thus longer demonstration sessions can compensate the lower frequency of actions. Taken together with similar observations in related work we can surmise that the higher task accuracy is a strong indicator for the motivational background increasing task performance and thus quality of motion data for PbD.

DISCUSSION

We have established a HCG for collecting data on embodied knowledge of everyday activities that can be used for programming manipulation robots. In the following we touch points of discussion regarding controls, game design, and experiment design and reflect the chosen approach.

Free-hand Controls

As a first exploration of HCG for robot imitation learning and for the first 3D TD available, controls were sufficient and very usable. As suggested by participants, the free-hand controls for virtual action demonstration can be further improved. With a more literal virtual hand that includes tracked fingers, grab and release gestures could replace tool/object activation via timeout, resulting in more fluent and efficient interaction that better replicates the real world activity. Advances in camera-based low-latency hand tracking will also improve the controls. A stereoscopic display would provide better depth perception, which was an issue for some participants despite depth cues.

Physics Simulation

Game engine physics are not optimized for fully realistic simulation of physical processes. For practical use, game-based knowledge acquisition tools such as 3DFEND require more realistic simulation engines. Related research problems such as enabling interactive simulation at real-time frame rates are complementary to the problems under investigation in this work and are being researched in the robotics and computer graphics communities. Actual PbD tools will thus integrate insights on HCG for imitation learning with state-of-the-art physics simulation.

Game Design

3DFEND is the first free-hand control 3D tower defense game and still in prototype stage. Game design was minimalist, and game mechanics, assets and story should be improved in further iterations. Also, proper balancing is essential in order to make sure that no "game-breaking" strategies emerge and frequency distribution of the various actions is even. However, player feedback confirmed that the prototype was sufficient for exploring a HCG approach to acquiring motion data for PbD.

Experiment

The control condition of our experiment presented participants with instructions for basic manipulation without motivation or context. We abstracted from the task domain cooking in order to provide a generalizable, neutral counterpart to the game scenario. This includes two design choices that might have affected the aspects of experienced presence and autonomy, respectively. The first is that a context can give such scenarios intentions: baking virtual pancakes is likely more motivating and likely increases presence over moving cubes. The second is that purely following instructions subdues the creative potential and experienced autonomy of playful virtual task exploration—different players might develop different action strategies to reach the same goal. However, without the guiding effect of game mechanics and narrative, it is very difficult to control action frequency in such free designs. We leave exploring the effects of these design nuances for future work.

Scalability and Extendibility

We have illustrated that the chosen game design approach works for the cooking subtask of making a pancake. This can be easily scaled and extended to other task domains with similar characteristics. Strategy is a game genre with huge variation. For TD, more tower types, different levels, varied enemy resistances and tactics are only a few examples of how game design can model more diverse and complex actions.

Embodied Human Computation Games

HCG can incorporate embodied knowledge by capturing intricate details of motion behaviors through modern natural user interfaces. The mapping of game intentions to task domain actions is a flexible design decision. However, with embodiment issues, the mapping of user input to actions, also discussed for PbD [2], can be very direct, requiring less complex methods in the *deriving* step. Thus marrying HGC for embodied task knowledge and imitation learning is an especially promising approach.

Toward a Framework for Manual Behavior HCG

In order to bring insights gathered in our design process into a tentative framework for choosing game archetypes for manual action demonstration human computation games, we formulate questions for designers to consider.

Can every part of the game world be reached within a short time space? Large-scale locomotion is distinct to strategies of manual behavior. To capture the latter, the former should be avoided. In most tower defense games, the game world can easily fit into 1-4 screens.

Do game controls require many degrees of freedom (DOF)? High-DOF game controls might be more difficult to match than low-DOF controls. In the tower defense example, the simple click-and-select controls gave us more freedom in matching manual actions.

Can game time constraints be managed (e.g. turns, waves)? Full real-time game world simulation can be limiting, as manual actions often have a different time scale than the game world. Managed time constraints such as in turn-based gameplay gives more leeway in matching task and game domain time scales.

CONCLUSION

HCI paradigms will change dramatically in the future when the hardware moves from current display-dominated interfaces to active agents such as household robots. Programming such companions for everyday activities requires new methods for acquiring knowledge about highly variable environments and the tasks they have to perform in them. With 3DFEND we established a human computation games approach for harvesting embodied knowledge on "workbench" manipulation tasks. Our study shows that this embodied human computation game motivates users to voluntarily contribute data for imitation learning. We mapped a set of typical manual actions to a tower defense game that could easily be extended to other task domains. Our work shows that gameful human computation approaches with consumer hardware motion-based input can generate motion data for understanding human manual behavior. It provides a basis for future work on mappings from the task domain to the game domain.

REFERENCES

1. Aras, H., Krause, M., Haller, A., & Malaka, R. (2010). Webpardy: Harvesting QA by HC. In *Proc. HCOMP 2010*, (pp. 49–52). ACM.
2. Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, *57*(5), 469–483.
3. Beetz, M., Klank, U., Kresse, I., Maldonado, A., Mosenlechner, L., Pangercic, D., Ruhr, T., & Tenorth, M. (2011). Robotic roommates making pancakes. In *Proc. Humanoids 2011*, (pp. 529–536). IEEE.
4. Billard, A., Calinon, S., Dillmann, R., & Schaal, S. (2008). Robot programming by demonstration. In B. Siciliano, & O. Khatib (Eds.) *Springer Handbook of Robotics*, chap. 60, (pp. 1371–1394). Springer.
5. Bollini, M., Tellex, S., Thompson, T., Roy, N., & Rus, D. (2013). Interpreting and executing recipes with a cooking robot. In J. P. Desai, G. Dudek, O. Khatib, & V. Kumar (Eds.) *Experimental Robotics*, vol. 88 of *Springer Tracts in Advanced Robotics*, (pp. 481–495). Springer.

6. Bonetta, L. (2009). New citizens for the life sciences. *Cell*, *138*(6), 1043–1045.
7. Bowman, D. A., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2004). *3D User Interfaces: Theory and Practice*. Addison-Wesley.
8. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining "gamification". In *Proc MindTrek 2011*, (pp. 9–15). New York, NY, USA: ACM.
9. Flatla, D. R., Gutwin, C., Nacke, L. E., Bateman, S., & Mandryk, R. L. (2011). Calibration games: Making calibration tasks enjoyable by adding motivating game elements. In *Proc. UIST 2011*, (pp. 403–412). ACM.
10. Forbes, M., Chung, M. J., Cakmak, M., & Rao, R. P. N. (2014). Robot programming by demonstration with crowdsourced action fixes. In *Proc. HCOMP 2014*, (pp. 67–76). AAAI Press.
11. Haidu, A., Kohlsdorf, D., & Beetz, M. (2014). Learning task outcome prediction for robot control from interactive environments. In *Proc. Intelligent Robots and Systems*. IEEE/RSJ.
12. Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. In D. Fu, S. Henke, & J. Orkin (Eds.) *AAAI Workshop Challenges in Game Artificial Intelligence*. AAAI Press.
13. Kanda, T., Ishiguro, H., Imai, M., & Ono, T. (2004). Development and evaluation of interactive humanoid robots. *Proceedings of the IEEE*, *92*(11), 1839–1850.
14. Krause, M., & Smeddinck, J. (2012). Human Computation: A New Aspect of Serious Games., (pp. 1027–1047). In *Handbook of Research on Serious Games as Educational, Business and Research Tools: Development and Design.* IGI Global.
15. Krause, M., Takhtamysheva, A., Wittstock, M., & Malaka, R. (2010). Frontiers of a paradigm: Exploring human computation with digital games. In *Proc HCOMP 10*, (pp. 22–25). ACM.
16. Kunze, L., Haidu, A., & Beetz, M. (2013). Acquiring task models for imitation learning through games with a purpose. In *Proc. IROS, 2013*, (pp. 102–107). IEEE.
17. Leap Motion, Inc. http://leapmotion.com (last visited on 2015-01-07).
18. Matyas, S., Matyas, C., Schlieder, C., Kiefer, P., Mitarai, H., & Kamata, M. (2008). Designing location-based mobile games with a purpose: Collecting geospatial data with CityExplorer. In *Proc. ACE 2008*, (pp. 244–247). ACM.
19. McCarthy, J. (1968). Programs with common sense. In *Semantic Information Processing*, vol. 1, (pp. 403–418).
20. Pollard, N. S., Hodgins, J. K., Riley, M. J., & Atkeson, C. G. (2002). Adapting human motion for the control of a humanoid robot. In *Proc. ICRA 2002*, vol. 2, (pp. 1390–1397). IEEE.
21. Quinn, A. J., & Bederson, B. B. (2011). Human computation: A survey and taxonomy of a growing field. In *Proc. CHI 2011*, (pp. 1403–1412). ACM.
22. Ryan, R. M., Rigby, C. S., & Przybylski, A. (2006). The motivational pull of video games: A Self-Determination theory approach. *Motivation and Emotion*, *30*(4), 344–360.
23. Schaal, S. (1999). Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, *3*(6), 233–242.
24. Shamsuddin, S., Ismail, L. I., Yussof, H., Ismarrubie Zahari, N., Bahari, S., Hashim, H., & Jaffar, A. (2011). Humanoid robot NAO: Review of control and motion exploration. In *Proc. ICCSCE*, (pp. 511–516). IEEE.
25. Toris, R., Kent, D., & Chernova, S. (2014). The robot management system: A framework for conducting Human-Robot interaction studies through crowdsourcing. *Journal of Human-Robot Interaction*, *3*(2), 25+.
26. Tuite, K., Snavely, N., Hsiao, D. Y., Smith, A. M., & Popović, Z. (2010). Reconstructing the world in 3D: Bringing games with a purpose outdoors. In *Proc. FDG 2010*, (pp. 232–239). ACM.
27. von Ahn, L., & Dabbish, L. (2008). Designing games with a purpose. *Commun. ACM*, *51*(8), 58–67.
28. Yuen, M.-C., Chen, L.-J., & King, I. (2009). A survey of human computation systems. In *Proc. CSE*, vol. 4, (pp. 723–728). Los Alamitos, CA, USA: IEEE.