

# Haptic Turk: a Motion Platform Based on People

Lung-Pan Cheng, Patrick Lühne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch

Hasso Plattner Institute, Potsdam, Germany

{firstname.lastname}@hpi.uni-potsdam.de

## ABSTRACT

Motion platforms are used to increase the realism of virtual interaction. Unfortunately, their size and weight is proportional to the size of what they actuate. We present *haptic turk*, a different approach to motion platforms that is light and mobile. The key idea is to replace motors and mechanical components with humans. All haptic turk setups consist of a *player* who is supported by one or more *human-actuators*. The player enjoys an interactive experience, such as a flight simulation. The motion in the player's experience is generated by the actuators who manually lift, tilt, and push the player's limbs or torso. To get the timing and force right, timed motion instructions in a format familiar from rhythm games are displayed on actuators' mobile devices, which they attach to the player's body. We demonstrate a range of installations based on mobile phones, projectors, and head-mounted displays. In our user study, participants rated not only the experience as player as enjoyable (6.1/7), but also the experience as an actuator (4.4/7). The approach of leveraging humans allows us to deploy our approach anytime anywhere, as we demonstrate by deploying at an art festival in the Nevada desert.

## Author Keywords

Haptics; force-feedback; motion platform; immersion.

## ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

## INTRODUCTION

For a long time, the key to immersion in interactive experience and games was sought in photorealistic graphics [8]. More recently, game makers made games more immersive by requiring players to physically enact the game such as with Wii (<http://wii.com>) and Kinect [26]. With graphics and user interaction now part of many games, many researchers argue that *haptics and motion* are the next step towards increasing immersion and realism, i.e., applying the forces triggered by the game onto the player's body during the experience.

While some game events can be realistically rendered using one or more vibrotactile actuators (e.g., driving over gravel in a racing game [14]), a much larger number of gaming events result in *directional* forces, such as centrifugal forces pulling at a steering wheel or a car bumping into the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.

Copyright ©ACM 978-1-4503-2473-1/14/04...\$15.00.

<http://dx.doi.org/10.1145/2556288.2557101>

railing. Such events have been simulated using motion platforms [27]. Motion platforms are able to move one or more users around and have been used to add realism to flight simulators [22] and theme park rides.

Unfortunately, the size and weight of motion platforms tends to be proportional to what they actuate. As a result, motion platforms not only tend to be prohibitively expensive, but also large and heavy and thus stationary, limiting their use to arcades and lab environments.

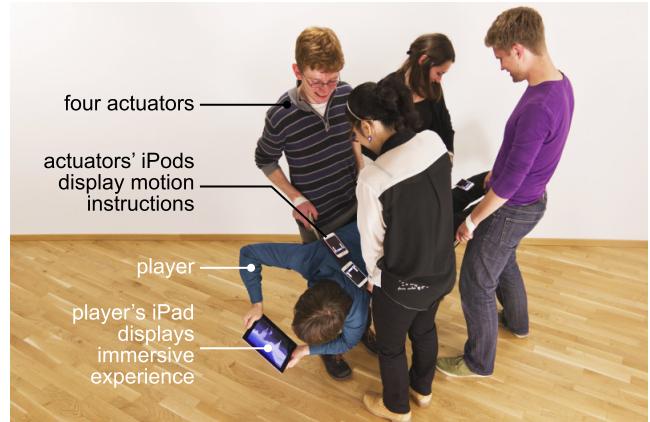


Figure 1: Haptic turk allows producing motion experiences anywhere anytime. Here, the suspended player is enjoying an immersive hang gliding game. The four *actuators* create just the right physical motion to fill in the player's experience.

In this paper, we present *haptic turk*, a software platform that allows experiencing motion anywhere there is people. Its key idea is to substitute the motors and mechanical components of traditional motion platforms with humans.

## HAPTIC TURK

Haptic turk is a motion platform based on people. The name is inspired by the 18<sup>th</sup> century chess automaton “The Turk” [20] that was powered by a human chess master.

The specific configuration shown in Figure 1 involves one *player* located in the center. The player is enjoying an immersive experience, here a first-person simulation of flying a hang-glider, running on a hand-held device (iPad). In the shown setup, the player can steer the hang-glider by tilting the iPad.

The main difference to regular video games is that the player's experience comes with *motion*—this motion is administered by *human-actuators* who *manually* lift, tilt, and push the player around. Here there are four of them.

To get the timing and force right, all actuators receive timed motion instructions in a format familiar from rhythm games (see Figure 3 for a preview). In the set-up shown in Figure 1, actuators receive these motion instructions on

their mobile devices (here iPods and iPhones) that they have attached to their player's body.

During the experience, actuators execute the motion instructions displayed on their respective device by moving the part of the player's body assigned to them. At the moment shown in Figure 1, for example, the two actuators left of the player lift up their side of the player, causing the player to be rotated ("rolled") towards the camera.

Haptic turk generates these motion instructions so as to feed into the player's experience. As actuators perform their motion instructions, they therefore contribute to the player's experience, making it richer and more immersive.

A haptic turk experience *may* incentivize actuators to perform their best by measuring and scoring how well they match their motion instructions in terms of timing, position, and force—using the inertial measurement unit (IMU) in their mobile device. However, we obtain a better player experience by scoring actuators as a group, as this encourages actuators to synchronize during cooperative moves. But finally, several participants in our user study stated that they simply enjoyed supporting the player, suggesting that the act of providing an experience to the player provides all the necessary incentive.

#### WALKTHROUGH

We now illustrate the haptic turk platform at the example of the specific configuration from Figure 1 (four actuators with mobile devices) and a hang glider experience we call *team flight*. The actual experience takes three minutes and contains 14 motion event groups (25 individual actuator movements overall). We present selected scenes that allow us to illustrate the design elements.

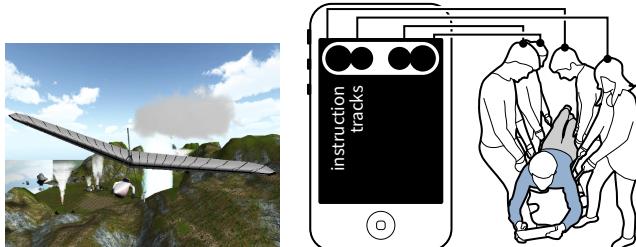


Figure 2: The *team flight* hang glider experience has started: The actuator display is blank, actuators are relaxed.

As illustrated by Figure 2, the experience starts out in calm weather and the player's hang glider is in a neutral, horizontal position. Accordingly, the *actuator displays* are blank. This instructs all actuators to stand upright, relaxed, shoulders dropped—a position that actuators can sustain with minimum effort. We design our experiences to bring actuators back to this position frequently to avoid fatigue. Note that the actuator displays show the timelines of *all* actuators so as the help actuators see "the big picture" and to synchronize their actions.

**Up/down-bars:** The player is approaching a big fan located on the right and the hang glider is about to get caught by the fan's draft. Haptic turk reflects this by rendering *up/down* motion instructions. The bars enter the actuator

display from below and travel up the screen as the hang glider is approaching the fan (Figure 3a). This leverages the visual language of, for example, *dance dance revolution* ([www.konami.com/ddr](http://www.konami.com/ddr)). As the bars reach the actuators' *bullseyes* at the top of the screen, the respective actuators execute the instructions. Motion instructions take about seven seconds to reach the bullseye. This is essential as it allows actuators to get ready so as to perform their motion *on time*, i.e., in sync with the player's experience.

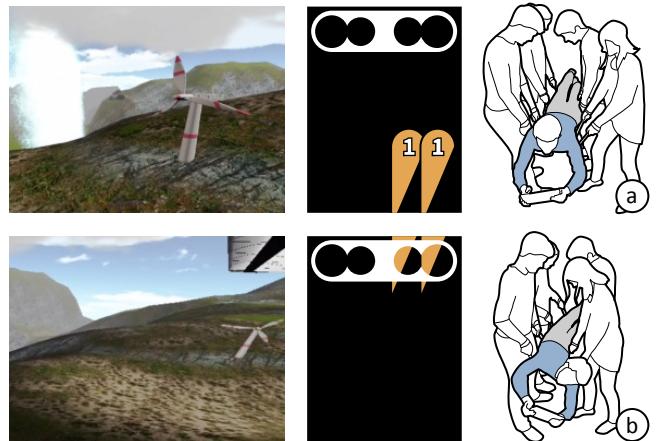


Figure 3: (a) The player is about to get caught by the draft of a fan "lift to +1" motion instructions appear, and (b) are executed as they reach the actuator's bullseyes.

The shown up/down bars in the two right columns reach the bullseye exactly at the moment the player enters the draft of the fan. The up/down bars are labeled "1"; the two actuators right of the player therefore lift their side of the player to position "1", which is waist height. This causes the player to be rotated or "rolled" to the left, as the player's hang glider is pushed sideways by the fan.

The two up/down bars have round heads, which demands actuators to move up *abruptly*, right at the moment the round head fills the round bullseye. Actuators continue to hold the player at +1 height, then return *smoothly* to the relaxed position, as demanded by the bars' diagonal tails.

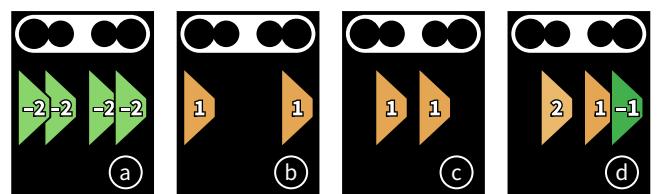


Figure 4: Up/down bar representations for (a) drop, (b) face down tilt, (c) face up tilt, and (d) up-left tilt-roll.

Up/down bars are haptic turk's most versatile design element and they are used to render the vast majority of motion events. As shown in Figure 4, up/down bars come with values from -2 to +2, with +2 'chest level', +1 'waist level', blank 'relaxed', -1 'knee level', and -2 'foot level'. Redundant color-coding makes the display more glanceable (sun orange means up; grass green means down). The main idea behind up/down bars is that they show actuators exactly

what to do, yet, with experience, allow them to see the big picture.

**Large movements & anticipation:** Figure 5 shows how haptic turk performs a large motion event. Here the player enters a tornado, which causes the player's hang glider to shoot up into the air. Haptic turk renders this by instructing all four actuators to abruptly lift the player to '+2' the chest level. To make the player's experience even more intense, haptic turk instructs actuators to first lower the player slightly in anticipation *and then* to perform the vertical '+2' move.

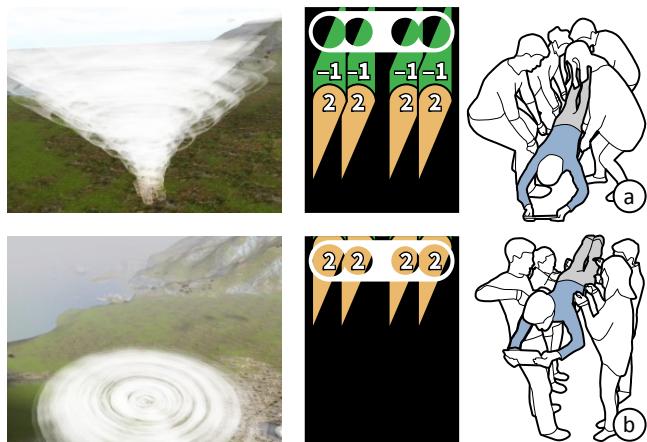


Figure 5: Right before the player enters the tornado, haptic turk lets players anticipate the move by lowering the player.

**Collisions:** In Figure 6, the player's hang glider collides with an object, here a blimp approaching from the left. Haptic turk renders the event as a pair of "bump" instructions on the left. As the bump events reach the bullseyes, the left two actuators bump the player using their thighs.

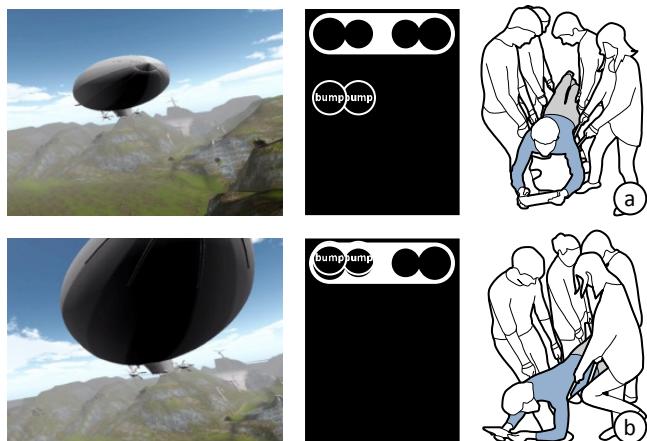


Figure 6: As the player collides with an object, here a blimp, actuators bump the player using their thighs

**Bumpy ride:** In Figure 7 the player enters turbulences, here created by a group of fans pointed at the player from different directions. Haptic turk responds by creating 'shake' instructions for all actuators, which each actuator executes as an unsynchronized vertical shake.

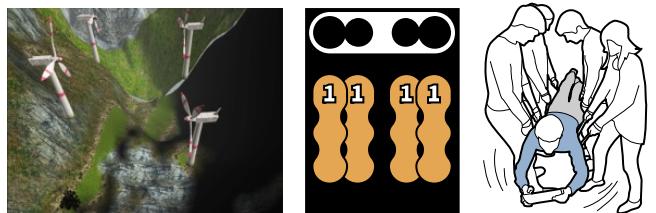


Figure 7: The fans cause an unsynchronized vertical shake

**Horizontal motion:** In Figure 8, the player is entering the field of a *very* powerful fan that propels the glider quickly out over the desert. Haptic turk emphasizes the onset of this movement with a horizontal move event, which actuators execute by taking a step into the specified direction. After all, the shown implementation of haptic turk is *mobile*, allowing actuators to walk around during the experience.

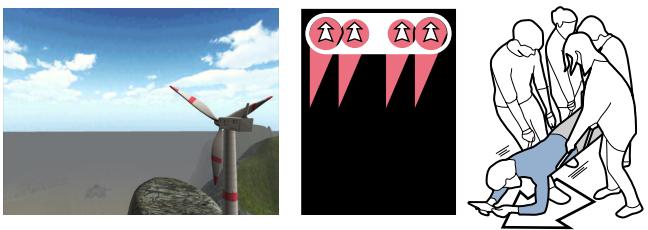


Figure 8: This fan produces a strong wind that shoots the player off horizontally, rendered as horizontal motion

**Special effects:** The concept of haptic turk is broader than just motion. Figure 9 shows two effects that we have explored, i.e., water and heat. They are administered by an additional "special effects" actuator.

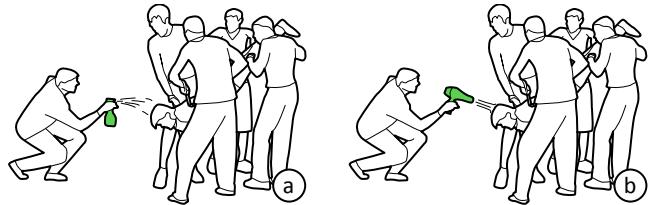


Figure 9: This special effects actuator (a) sprays water at the player as the player enters a waterfall and (b) uses a hair dryer to emulate a hot breeze, as the player crosses over into the desert terrain in Figure 8.

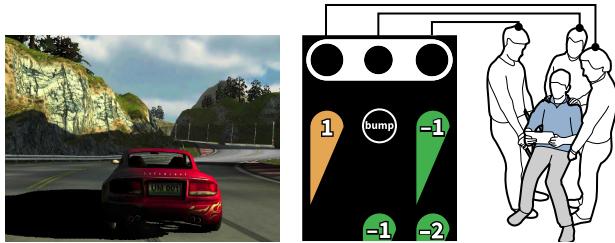
**Landing:** Finally, the player has reached the destination: a temple in the middle of the desert. As the player approaches the ground, haptic turk generates landing instructions, as shown in Figure 10.



Figure 10: To land the player, the front actuators whip up the player to +2, while the back two set down the player's feet.

These were all simple scenes, taken from an experience designed to be calm and serene, as one would expect a

hang-glider experience to be. Figure 11 shows a more action packed game sequence we created—note how the actuator display now handles multiple actions at once.



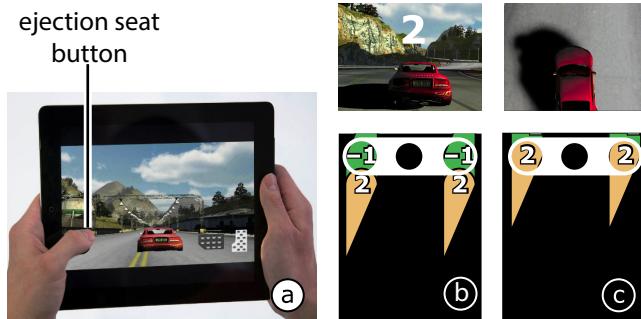
**Figure 11:** More action-packed games, such as our car racing game contains faster sequences of motion instruction. (This experience is administered by 3 actuators, thus 3 timelines)

#### GETTING THE TIMING RIGHT: PREEMPTIVE WARNING

The main technical challenge behind haptic turk is to get the timing right, i.e., to make sure that actuators provide force exactly at the moment required by the player’s experience. This is a challenge because actuators’ are inherently subject to human response time and thus delay. The mean response time for humans to react to a simple visual stimulus is 220ms [17], to which we add network delay and human movement. This duration is substantially longer than the 50ms humans tolerate in delayed haptic feedback [15].

For pose changes, we found this delay to not cause too much of an issue as long as we simulate a vehicle that moves *smoothly*. Hang gliders, for example, change their position only gradually in response to their pilots shifting their weight around. Here, haptic turk simply displays a preview of the expected goal pose as soon as the player starts to steer; this gives actuators ample time.

The true challenge comes from instantaneous events, such as the forces resulting from a user colliding with an object. Haptic turk addresses this challenge with two approaches.



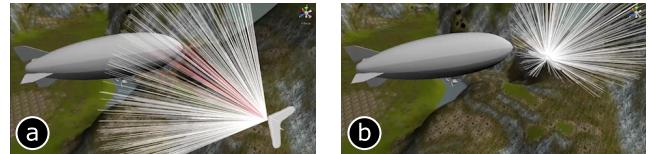
**Figure 12:** Delaying a game event using a countdown: (a) In response to the player pressing the ‘eject’ button, (b) haptic turk renders up/down bar. (c) When the countdown runs down, the actuators eject the player—well prepared.

**#1: Countdown events.** Haptic turk cannot speed actuators up, but it can slow reality down—by adding a countdown. We use this approach for all player actions where precision matters, either because they result in a high force or because they are performed by multiple actuators in synchrony. The ejection seat button in the racecar of our racing game, for example, uses a three-second countdown before it fires.

The countdown not only adds drama to the game, but, more importantly allows haptic turk to give actuators advance warning. This is essential, as it allows actuators to throw the player up in the air in synchrony.

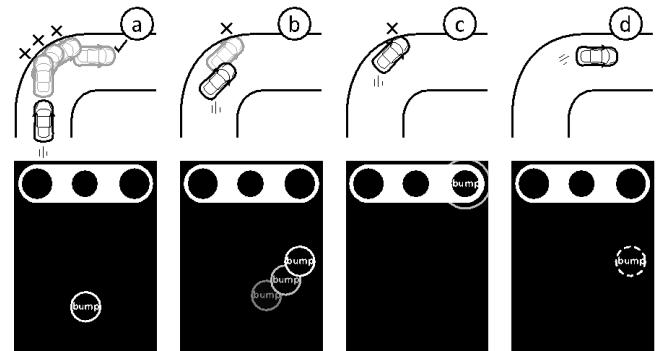
Only certain game events can plausibly be fitted with a countdown. For other events, such as those resulting from physical collisions, haptic turk creates timely responses through “anticipation”.

**#2: Anticipate collisions.** Rather than waiting for the collision to happen, haptic turk continuously checks for the *possibility* of an upcoming collision. Whenever, a collision is likely, haptic turk displays a *possible bump* event, allowing actuators to get ready. We call this haptic turk’s “pre-response” mechanism. Haptic turk probes the space the user is likely to reach in the next seconds using *probe lines* (Figure 13, inspired by imaginary reality games [2]). In team flight, for example, we send probe lines forward, as this is the only direction the glider can go. Each probe line is as long as the player can travel in the 7 seconds that actuator displays project into the future.



**Figure 13:** (a) Haptic turk’s debug view: the hang glider on the bottom right continuously sends out probe lines (~1000 on a MacBook Pro). Here some of them detect a blimp (shown in red). (b) As the player turns right to avoid the blimp, no more probe lines are reaching the blimp.

At every frame, haptic turk counts how many probe lines predict a collision. If a collision seems likely ( $>30\%$ ) haptic turk injects a motion event into the actuator displays (Figure 14a). It renders the motion instruction’s opacity so as to reflect the probability of the event, so 70% opacity represents a 70% probability. Haptic turk then continuously updates the actuator display.



**Figure 14:** (a) Haptic turk anticipates a collision with the curb and creates a bump event. (b) As the player turns, the bump event changes to reflect the new expected orientation of the bump. (c) If the car stays on the course, the collision takes place. Otherwise, (d) the bump event dissolves.

As the player navigates, the expected collision time tends to change. The motion instructions may thus speed up if the

player steers into the obstacle or they may slow down if the player turns to hit the obstacle more tangentially. At the same time, the expected collision probability and thus the opacity of the motion instructions will vary. If the probability ever drops below a threshold (<15%), haptic turk removes the bump instruction—a “false alarm”.

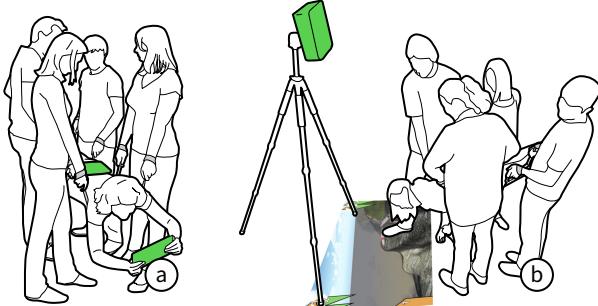
To enable probe lines, all the in-game objects have their own “colliders” and corresponding motion instruction. The tornado in the hang gliding game, for example, has a cylindrical collider. Probing it triggers the -1/+2 up/down bars shown earlier.

### ONE PLATFORM, MANY CONFIGURATIONS

The haptic turk software is a general-purpose platform for creating motion experiences based on people. In the examples shown above, we used one particular hardware set-up based on mobile devices, one for each actuator. This installation was designed with ubiquitous use in mind in that it only requires devices that users are likely to carry with themselves at all times. In the following, we show other hardware configurations we have explored.

#### Display/sensing configurations

Figure 15a shows another mobile set-up. It reduces hardware requirements further by running on two iPads—one for the player and a single shared iPad for the actuators.



**Figure 15:** (a) Haptic Turk running on two iPads (b) The walk-up installation, designed for museums etc., is designed to minimize per-user start-up time. This version integrates player and actuator display into a single projection.

Figure 15b shows one of our *walk-up* installations designed for use at tradeshows, museum exhibits, and art shows. Walk-up installations do not require participants to bring anything and we designed them to minimize per-player set-up time. The shown version integrates player display and actuator display and projects onto the floor. This version allows players to control the game using a simulated delta, which we implemented by attaching an iPod to a bar.

We also combined our walk-up installation with a head-mounted display (*Oculus Rift*, <http://oculusvr.com>) as shown in Figure 16. It allows users to look around in the virtual environment. As the head mounted display is closing out reality, it delivers a higher level of immersion than any of the other installations. We deployed this version at Burning Man 2013 (see section “Deployment”, Figure 25).



**Figure 16:** This installation uses a head-mounted display for the player experience (and projection as actuator display).

Finally, we may record haptic turk experiences and share them as “haptic turk movies”. People downloading the movie may then render the experience on any installation they have access to. While this could be done by using any video format that allows for timed annotations (e.g., mpeg 7), we feel that the most effective format is to overlay the actuator instructions directly onto the video stream (as in Figure 15b). This allows us to share on *any* medium that transports video, including web pages, file sharing services, and YouTube.

#### Mechanical configurations

All designs shown so far use four actuators to keep the player suspended. We initially had actuators hold the player directly (Figure 20), but then added the *slingshots* shown in Figure 17 to reduce actuator fatigue, increase player safety, and to soften potential proxemics issues. The shown design we made from the seat cushions of foldable chairs (*Folding stool Stockholm II* by *Lectus*), curtain bars, and linen ribbon. Slingshots can be rolled up to carry.



**Figure 17:** We made these custom slingshots to reduce actuator fatigues, increase player safety, and soften proxemics.

Figure 18 shows some of the alternative mechanical configurations we have explored. The shown designs (a, b, c) allow three experiences that require an upright player pose, such as car racing, (b, c, d) provide additional physical support to the player, thereby reducing actuator fatigue. They allow playing with fewer actuators and/or people with a wider range of physical abilities.

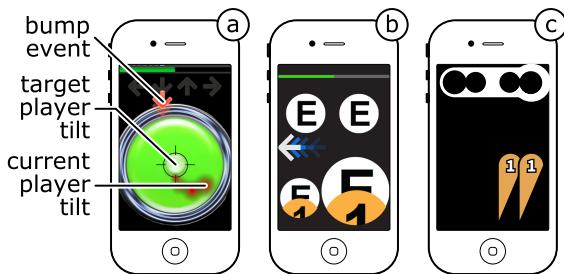


**Figure 18:** Haptic Turk configurations: (a) racing game based on two actuators, (b) chair and (c) swing allow for single-actuator use, and (d) in swimming pool.

### Actuator interfaces

The actuator interface presented earlier in the “walkthrough” section was the result of a series of iterations. Figure 19 shows two earlier designs for context. Our initial “level and arrows” design (Figure 19a) caused split attention, but most of all it caused ergonomic issues: Since this display style only indicated the desired player tilt, but no absolute height, actuators tended to spend most of their time in non-neutral postures, resulting in substantial fatigue.

We consequently switched to a model that instructed actuators using a notion of absolute height (Figure 19b). The 2x2 layout also showed the actuator instructions of up to four actuators, allowing actuators to better synchronize.



**Figure 19: Earlier versions of the actuator interface:** (a) Initial version based on the metaphor of a two-dimensional level. Only for bumping events did we use the time-line metaphor. (b) This second helped us get actuators into the neutral position, thus improved actuator ergonomics. (c) The current actuator interface. The thick bullseye highlights the actuator’s timeline.

The interface, however, was limited in that it provided actuators with too little preview of upcoming events, which is a truly essential requirement for actuator displays. We addressed this by lining up the four timelines at the top of the display and unifying all instructions to fit the timeline format—which brought us to the version shown throughout this paper (Figure 19c). It performed very well throughout studies and deployment.

### CONTRIBUTION & BENEFITS

The main contribution of this paper is a new mechanism for creating a motion platform—based on humans. We present user interfaces and a system of motion instructions, three display and four mechanical configurations, interactive experiences, and two approaches to address lag.

The main benefit of our approach is *ubiquitous availability*. (1) *Instantaneous*: Since the underlying units—people—are incredibly flexible, users may obtain new experiences simply by downloading them from a network “into the actuators”. This means that the actuators that formed a hang glider a second ago may now serve as a car or battle robot. This way, haptic turk reduces the sharing of motion experiences to the sharing of data files. (2) *Available*: Haptic turk runs on equipment that is orders of magnitude cheaper and more space-efficient than the technical equipment it emulates. In particular, the mobile versions of haptic turk could potentially reach millions of users by leveraging the existing install base of mobile devices. (3) *Everywhere*: We

have deployed haptic turk not only in the lab and the student cafeteria but also at a desert festival, i.e., a hostile environment that would make it hard to deploy an actual motion platform. A single person was able to bring the equipment on the plane.

### LIMITATIONS

Limitations of our approach include that it requires multiple people, is tiring, and that human actuators cannot rival a computerized motion platform in terms of responsiveness and reliability. In particular, a real time unpredictable event is difficult to handle in our current system. For example, if players controlling a racing game swerve and hit a wall, this hit is difficult to predict. This makes the event difficult to haptically render by our system, as it does not leave enough time to properly prepare actuators for their action. Haptic turk runs into similar issues if *multiple* events become possible at a given moment.

In addition, there are all the risks that come with motion equipment in the first place, such as motion sickness and the risk of injury. The approach may also raise proxemics considerations, the extent of which should be expected to vary across cultures. We address some of these issues with the design of slingshots that reduce actuator fatigue, increase player safety, and soften proxemics issues, but these are certainly only a partial solution.

In exchange to the proxemics issues, haptic turk delivers not only force feedback, but also a human-to-human experience that lets people interact in a new way. While we initially expected that competing to win the rhythm game would be the main incentive for actuators, the physical activity itself and, in particular, the social nature of the setup turned out to be the main driving force that made people participate in studies and experimental deployment.

### RELATED WORK

The work presented in this paper is related to interaction concepts, such as Wizard of Oz as well as motion platforms.

#### Wizard of Oz and Crowdsourcing

Haptic Turk shares some characteristics with Wizard of Oz systems, such as the aforementioned Turk [18]. Researchers use the wizard of oz method to speed up prototyping [6] and to give users new experiences [7].

Haptic Turk is different from crowdsourcing, such as Amazon’s Mechanical Turk ([www.mturk.com/mturk](http://www.mturk.com/mturk)), which focus on recruiting workers. However, haptic turk could use crowdsourcing platform such as TaskRabbit ([www.taskrabbit.com](http://www.taskrabbit.com)), CommunitySourcing [13] and Friendsourcing [3] to help recruit actuators.

#### Motion Platforms, Exoskeletons & Mobile Haptics

A wide variety of force feedback devices have been used to enhance the realism of virtual reality interactions including virtual object manipulation and motion simulation [18].

**Motion simulators** simulate motion by shaking, lifting or tilting players or groups of player sitting or standing on them. They are intensively used in driving and flight simulation for both training and entertainment purposes [22].

Most of them are based on a Stewart platform [27], which has six degrees of freedom driven by six hydraulic cylinders as actuators.

HapSeat [5] achieves motion simulation with lower cost and a more compact form factor by actuating the user's head and hands. The fact that users perceive motion mainly using their visual, auditory, vestibular, and kinesthetic systems [4,10] allows this project to limit actuation to arms and headrests.

Researchers also showed that vibrotactile feedback may generate the illusion of self-motion [25]. Tactile Brush [14] uses this haptic illusion and renders vibrotactile strokes on the user's back using a grid of actuators in the chair.

**Exoskeletons** are wearable machines not only for amplifying users' motion but also providing force feedback. There are different kinds of exoskeletons, such as [11,19] that use different actuators (pneumatic, hydraulic and strings) to help users perform six or more degrees of freedom tasks either in the real or the virtual world. They can provide large forces, but are also heavy and large because of their actuators and mechanical parts. *FlexTorque* [29] offers an exoskeleton in a portable form factor; it provides force feedback to the arm of a user playing shooting games.

**Mobile Haptics** Lopes et al. [21] proposed a mobile force feedback device, which relies on electrical muscle stimulation to actuate the user's opposing muscles to counter the current input motion. GyroTab [1] produces torque in a mobile form factor based on a gyroscopic effect. Yano et al. [30] proposed a handheld haptic device that uses a laser range finder to control a linear servomotor that touches users fingertips so that users can touch the object remotely. Han et al. [9] proposed a handheld haptic device that integrates a linear servomotor and a solenoid-magnet pair located under a membrane to simulate huge and subtle kinesthetic feedback when the finger touches the membrane. Another mobile haptic device is POKE [23], which uses an air pump and silicon membranes on the front side to poke the caller's face remotely.

## IMPLEMENTATION

We implemented the haptic turk platform and the experiences (hang glider and racing) in C# and JavaScript on the Unity3D engine (<http://unity3d.com>). We then deploy to notebook computers (we used Mac OSX, but can also deploy to Windows and Linux) and mobile devices (we used iOS, but can also deploy to Android). By switching to the professional version of Unity3D, we obtained the ability to deploy to the oculus rift.

The actuators' mobile devices sample their accelerometers and gyros 60 times a second and assess actuators' performance by comparing these readers with the received motion instructions in terms on tilt, roll, and acceleration.

Haptic turk connects devices using WiFi. Informal testing showed a mean latency of 32ms, which is faster than the earlier discussed 50ms delays that humans tolerate in a haptic response. Haptic turk achieves this with off-the-shelf

components, so we found no need for more sophisticated network synchronizations methods such as Delay Measurement Time Synchronization [24].

To run an experience, one person starts the haptic turk app as "host", which makes this person the "player" for the first round (Figure 20a). (b) All other users join the session as actuators. (c) Haptic turk indicates each actuator where to stand with respect to the player. The actuators Velcro-strap their devices to the indicated part of the player's body.



Figure 20: Setting up a haptic turk session via WiFi:  
(a) player starts a session, (b) joining actuator are assigned a position on the player's body, (c) play.

## LAB STUDY

To validate our approach, we conducted a user study. Our main objective was to verify that the haptic experience does indeed produce an enjoyable experience—for players and for actuators.

### Experience

In teams of six (player, four motion actuators, one special effects actuator) participants played the *team flight* hang glider experience, parts of which we presented in the walkthrough section.

### Interface

We ran the team flight experience on the walk-up virtual reality setup of haptic turk already shown in Figure 16, i.e., players experienced the world through a head mounted display (an Oculus Rift). Actuators saw their instruction on the computer screen in front of the player.

To assure that all participants enjoyed the same experience, we set the hang glider experience to autopilot. This limited player's interaction abilities to looking around using the oculus rift, but assured that all participants encountered the same events. It also allowed us to always complete the experience in a fixed amount of time (3min), thereby creating a controlled experience also for the actuators.



Figure 21: One team of participants playing *team flight*

Similarly, there was no actuator/actuator team scoring, so that the only reward for actuators was the opportunity to affect the player, plus the experience itself.

#### Procedure

We brought in participants in groups of 2, 3, or 4 and filled in experimenters so as to reach the 5 people required to play (plus we provided one experimenter for the water special effects). We then played one round of the team flight experience for each participant, so that each participant had the opportunity to be player exactly once.

With each new group, we provided two minutes of training during which we explained the handling of the slingshots, the actuator display and how high to lift the player for the four types of up/down bars, which track each actuator was expected to follow, and how to wear the oculus rift.

Participants played once as a player and 1-3 times as a actuator, according to their group size. Within these constraints, player/actuator order was counterbalanced. After all participants of a group were done playing and turking, all participants filled in questionnaires about their experience. We then released the group of participants and brought in the next group. Running a group of 2-4 participants took 10 to 15 minutes.

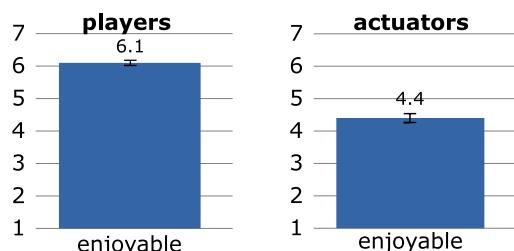
#### Participants

We recruited 14 participants (4 females) from our university. Their age ranged from 20 to 28 years (mean 23.6), BMI from 19-25 (mean 22.3). We recruited participants in groups of 2-4, i.e., participants were familiar with some of their actuators/co-actuators, but never all of them.

#### Results

Figure 22 summarizes our results.

**As players**, participants rated their experience on average as 6.1 ( $SD=0.7$ ) on a 7-point Likert scale (1=unpleasant, 7=enjoyable)—so clearly as enjoyable.



**Figure 22: Players rated gameplay as 6.1, so clearly enjoyable. Also as actuators, participants enjoyed the experience. Error bars are +/-1 standard error of the mean.**

Overall, players preferred the large motion events. Five players stated that they liked the intense motion resulting from lifting, shaking and bumping. Another player stated that he particularly enjoyed whole-body movements, such as being lifted or being swung forward—more so than being tilted and rolled. Along the same lines, three participants stated that they did not enjoy the extensive landing period in which they were tilted down. One participant described tilting as uncomfortable.

Accordingly, when asked about the most impressive moment of their player experience, 11 participants picked the intense -1/+2 boost caused by the tornado. One player expressed that “the changes in altitude were amazing and immersive”. Another player explained that he enjoyed the moment when he bumped into the blimp.

Players rated that the actuators “were pushing me at the wrong/right time” as 5.4 ( $SD=0.8$ ), i.e., they were satisfied with the actuators’ timing.

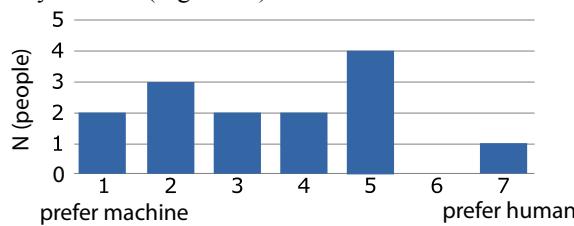
**As actuators**, participants rated the experience as actuator as less enjoyable than as players, yet still on the “enjoyable” side ( $M=4.4$ ,  $SD=1.2$ ).

Very obviously, the actuators’ experience was strongly driven by their perception of players’ experience. Actuators felt that their performance contributed to player’s experience ( $M=5.4$ ,  $SD=1.3$ ) (the “contribution” bar in Figure 22). Our observations match this. Five actuators said that they enjoyed seeing their players scream and giggle. One actuator said “it’s fun to play this with your friends and see their reactions as they fly.” And one simply stated “it’s fun to watch.” One participant would have enjoyed an even better view of what the player is experiencing. While we thought of the special effects role as being less exciting, one actuator said he would have also liked to take on that role.

The most likely reason for the lower score on enjoyment was fatigue. Seven actuators mentioned fatigue. One actuator mentioned that fatigue kicks in after two rounds of turking. Another actuator said “The person we moved was too heavy for me and I was smaller than the other players so my arms end at a lower height.” Two actuators mentioned that lifting the player to level +2 repeatedly had caused fatigue.

To learn more about proxemics, we asked participants who they would play team flight with. They indicated that they would play with friends (14/14) and family (10/14), but only 1/14 felt it was appropriate to play with the public. Given that this study had forced participants to play with a group of mostly strangers, this suggests that subjective satisfaction may improve further if experienced in a closer circle of friends and family. One participant explicitly said that she would enjoy playing haptic turk with her kids.

The human-human nature of haptic turk polarized participants. While 7/14 participants responded that they would have preferred an experience administered by a mechanical motion platform, 5/14 stated that they preferred being actuated by humans (Figure 23).



**Figure 23: Some participants preferred haptic turk, other would have preferred a mechanical motion platform.**

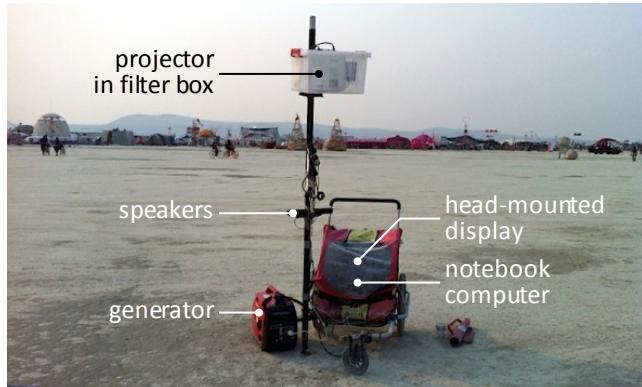
This suggests that these participants found an enjoyable quality in the human element. This is interesting, especially given that the actuators were assigned to them for the purpose of the experiment and not the friends or family that participants would have liked to play with.

**In summary,** our simple study provides some initial validation for the haptic turk concept. Most importantly, participants very much enjoyed the player experience. The actuator experience, while still enjoyable, could be improved by longer breaks and by giving the choice of who to play with.

### EXPERIMENTAL DEPLOYMENT

Encouraged by the results of the study, we decided to try an experimental deployment of haptic turk and the *team flight* experience at an art festival in the Nevada desert (burning man, <http://burningman.com>). Our goal was to learn more about the social dynamics of haptic turk—outside the lab.

We again opted for the walk-up VR haptic-turk set-up previously shown in Figure 16. As shown in Figure 24, we adapted the set-up to the hostile desert environment by placing the projector in a box equipped with air filters, etc. The entire installation (minus the generator, which we acquired on site) was transported, set up, and run by *one* of the authors, emphasizing haptic turk’s potential to deliver motion experiences anytime anywhere.



**Figure 24:** Our walk-up installation with head mounted display at the Burning Man art festival in the Nevada desert. A plastic box with fans protected the projector from desert dust.

We ran the installation only at night. While players wore the oculus rift, the projector projected a copy of the rift display with overlaid actuator display onto the desert ground, which served as actuator display (Figure 25). As before, we ran the team flight experience in autopilot and without additional incentives for actuators.

On three nights, we ran about 100 attendees. Player weights ranged from an estimated 100 pounds to a self-declared 200 pounds. While the venue did not afford running questionnaires, we observed attendees and videotaped four of the runs for further analysis.

### Players recruited actuators

Unlike a lab study, our set-up had to begin by attracting its own audience. Every run unfolded as follows. An attendee or a small group of 2-3 attendees would walk by and in-

spect the installation. There was no particular attract mode, but the projection showed the game running.

Whenever an attendee was interested in playing, they would convince their friends to tuck for them. Groups, however, were small—never large enough to play. We then encouraged attendees to recruit strangers as additional actuators. This typically took them a minute or two (“can you help us play a video game?”) and was simplified by the friendly atmosphere of the event.

While most attendees recruited to be able to play, several attendees “gifted” the experience to a friend—oftentimes their boy/girlfriend—i.e., they recruited actuators, but then stepped back and let their friend take on the role as player while they actuated.

Given the constant competition for visual attention at the festival, it was essential for us to get started as quickly as possible. To get started in about a minute, we assigned actuator positions, helped actuators into the slingshots, and taught them about how high to raise lower at the individual up/down bars. We then filled in one actuator position ourselves, which allowed us to start right away and instead coach the other actuators as the experience was unfolding.



**Figure 25:** A player wearing the oculus rift. Our installation projected the actuator interface (overlaid onto the game world) onto the desert ground. This player weighed 200 pounds.

### Observations

As during the study, the large motion events were favorites for players and actuators and led to audible expressions of joy. Upon “landing”, several players were visible taken by the immersion of the experience and typically required a few seconds to find their way back to reality.



**Figure 26:** A player entering a tornado.

Among actuators, the most popular scenes were again the large motion events, but also the rocky ride sequence in

which actuators had the opportunity to shake up the player a bit. One player reported motion sickness.

None of the actuators mentioned fatigue this time. The reason could be that, unlike our study, most attendees actuated only once, very few of them twice. This suggests that the fatigue threshold for this particular 4-person setup is around 3-5 min for this particular young athletic audience.

## CONCLUSION

We have presented haptic turk, an approach to having motion experience everywhere. With haptic turk, we tackle this challenge orders of magnitude cheaper, more space efficient, and faster-to-deploy than the technical equipment it is inspired by. By leveraging the existing install base of mobile devices, haptic turk has the potential to reach hundreds of millions of users. As a side effect, haptic turk produces an interesting new social experience.

As future work, we plan on using haptic turk to apply force feedback to players' hands and to use haptic turk for virtual reality, rehabilitation, and for rapid prototyping haptic machinery. We also plan on exploring how to give actuators a more active role, e.g., by allowing them to slip into the role of game world characters and to *control* game events.

## ACKNOWLEDGEMENTS

We thank Caro Fetzer for her help modeling the team flight 3D world.

## REFERENCES

- Badshah, A., Gupta, S., Morris, D., Patel, S., and Tan, D. GyroTab: a handheld device that provides reactive torque feedback. *Proc. CHI '12*, 3153–3156.
- Baudisch, P., Pohl, H., Reinicke, S., Wittmers, E., Luehne, P., Knaust, M., Koehler, S., Schmidt, P. and Holz, C. Imaginary Reality Gaming: Ball Games Without a Ball. *Proc. UIST '13*.
- Bernstein, M.S., Tan, D., Smith, G., Czerwinski, M., and Horvitz, E. Personalization via Friendsourcing. *TOCHI 08*, 17, 2, 6:1–6:28.
- Berthoz, A. *The Brain's Sense of Movement (Perspectives in Cognitive Neuroscience)*. Harvard University Press, 2000.
- Danieau, F., Fleureau, J., Guillotel, P., Mollet, N., Lécuyer, A., and Christie, M. HapSeat : Producing Motion Sensation with Multiple Force-feedback Devices Embedded in a Seat. *Proc. VRST '12*, 69–76.
- Davis, R.C., Saponas, T.S., Shilman, M., and Landay, J.A. SketchWizard: Wizard of Oz Prototyping of Pen-based User Interfaces. *Proc. UIST '07*, 119–128.
- Dow, S.P., Mehta, M., MacIntyre, B., and Mateas, M. Eliza Meets the Wizard-of-oz: Blending Machine and Human Control of Embodied Characters. *Proc. CHI '10*, 547–556.
- Gaggioli, A. and Breining, R. Perception and cognition in immersive Virtual Reality. *Emerging Communication: Studies on New Technologies and Practices in*, 71–86.
- Han, B., Kim, S., Lim, S., Pyo, D., and Kwon, D. Physical mobile interaction with kinesthetic feedback. *Proc. HAPTICS '12*, 571–575.
- Harris, L.R., Jenkin, H., Zikovitz, D., et al. Simulating self motion I: cues for the perception of motion. *Virtual Reality 2*, 6 (2002), 75–85.
- Hayashi, T., Kawamoto, H., and Sankai, Y. Control method of robot suit HAL working as operator's muscle using biological and dynamical information. *Proc. IROS '05*, 3063–3068.
- Heer, J. and Bostock, M. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. *Proc. CHI '10*, 203–212.
- Heimerl, K., Gawalt, B., Chen, K., Parikh, T., and Hartmann, B. CommunitySourcing: engaging local crowds to perform expert work via physical kiosks. *Proc. CHI '12*, 1539–1548.
- Israr, A. and Poupyrev, I. Tactile Brush : Drawing on Skin with a Tactile Grid Display. *Proc. CHI '11*, 2019–2028.
- Jay, C., Glencross, M., and Hubbald, R. Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *TOCHI 14*, 2 (2007).
- Kittur, A., Chi, E.H., and Suh, B. Crowdsourcing user studies with Mechanical Turk. *Proc. CHI '08*, 453–456.
- Kosinski, R.J. A literature review on reaction time. <http://biae.clemson.edu/bpc/bp/lab/110/reaction.htm>, 2012.
- Laycock, S.D. and Day, A.M. Recent Developments and Applications of Haptic Devices. *Computer Graphics Forum 22*, 2 (2003), 117–132.
- Lee, S., Park, S., Kim, M., and Lee, C.-W. Design of a force reflecting master arm and master hand using pneumatic actuators. *Proc. ICRA '98*, 2574–2579.
- Levitt, G.M., *The Turk, Chess Automation*. McFarland & Company, Incorporated Publishers, 2000.
- Lopes, P. and Baudisch, P. Muscle-Propelled Force Feedback: Bringing Force Feedback to Mobile Devices. *Proc. CHI '13*, 2577–2580.
- NASA Vertical Motion Simulator. <http://www.nasa.gov/centers/ames/research/technology-onepagers/vms.html> .
- Park, Y., Beak, K., and Nam, T. The roles of touch during phone conversations: long-distance couples' use of POKE in their homes, *Proc. CHI '13*, 1679–1688.
- Ping, S. *Delay Measurement Time Synchronization for Wireless Sensor Networks*. Berkeley, CA., 2003.
- Riecke, B.E., Schulte-Pelkum, J., Caniard, F., and Bulthoff, H.H. Towards Lean and Elegant Self-Motion Simulation in Virtual Reality. *Proc. IEEE VR '05*, 131–138.
- Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A. and Blake, A. Real-time human pose recognition in parts from a single depth image. In *Proc. CVPR '11*, 1297–1304.
- Stewart, D. A Platform with Six Degrees of Freedom. *Proc. Institution of Mechanical Engineers 180*, 1 (1965), 371–386.
- Telban, J., Cardullo, M., Telban, R.J., Wu, W., and Cardullo, F.M. Motion Cueing Algorithm Development: Initial Investigation and Redesign of the Algorithms. 2000.
- Tsetserukou, D., Sato, K., and Tachi, S. ExoInterfaces: novel exoskeleton haptic interfaces for virtual reality, augmented sport and rehabilitation. *Proc. AH '10*, 1:1–1:6.
- Yano, H., Miyamoto, Y., and Iwata, H. Haptic Interface for Perceiving Remote Object Using a Laser Range Finder. *Proc. WHC '09*, 196–201.