The RAD: Making Racing Games Equivalently Accessible to People Who Are Blind

Brian A. Smith **Shree K. Nayar**
Columbia University
New York, NY, USA
{brian, nayar}@cs.columbia.edu

ABSTRACT
We introduce the *racing auditory display (RAD)*, an audio-based user interface that allows players who are blind to play the same types of racing games that sighted players can play with an efficiency and sense of control that are similar to what sighted players have. The RAD works with a standard pair of headphones and comprises two novel sonification techniques: the *sound slider* for understanding a car's speed and trajectory on a racetrack and the *turn indicator system* for alerting players of the direction, sharpness, length, and timing of upcoming turns. In a user study with 15 participants (3 blind; the rest blindfolded and analyzed separately), we found that players preferred the RAD's interface over that of *Mach 1*, a popular blind-accessible racing game. We also found that the RAD allows an avid gamer who is blind to race as well on a complex racetrack as casual sighted players can, without a significant difference between lap times or driving paths.

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

Author Keywords
Accessibility; accessible games; audio games; sonification.

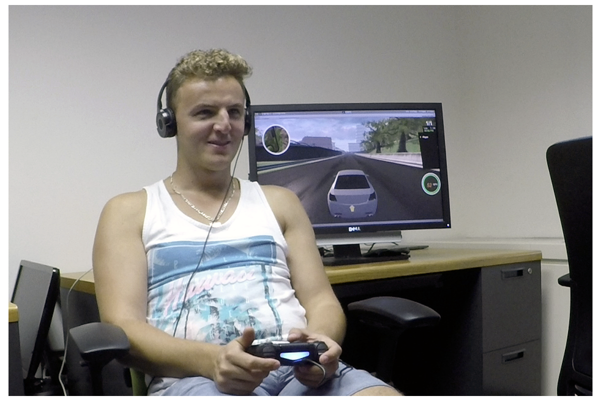

Figure 1. Study participant P8 — who is congenitally blind — playing our racing game prototype using the racing auditory display (RAD). The RAD outputs spatialized sound and works with a standard pair of headphones. Using the RAD, players who are blind can play the same types of racing games that sighted players can play with an efficiency and sense of control that are similar to what sighted players have. Our supplemental video shows P8 using the RAD with the RAD's audio included.

INTRODUCTION
Accessibility alone is not enough to make the world a fair place for people with disabilities. Even with assistive technologies, if people with disabilities cannot experience the world in the same manner as anyone else [32, 41], or even as productively as anyone else [16], the world will not yet be fair — or as we will say, *equivalently accessible*.

Imagine a wheelchair ramp leading to an entrance of a public library. Technically, the ramp would make the library accessible to people using wheelchairs. But if that ramp makes such a circuitous route on its way up that only people who need it would ever want to use it, the ramp would not make accessing the library efficient or fair [41]. Worse, suppose that the ramp leads to a separate, less handsome entrance to the library or, even worse, to a different building altogether: a smaller, adjacent library that only has the digest versions of books from the main library. These facilities would clearly not be fair for people using wheelchairs.

This situation, however, is similar to what video games are like for people who are blind. Most blind-accessible games today are either loaded with competing sources of information that players must sift through [4, 15, 29, 42], slowing down the efficiency of play, or are very simplified versions of games that sighted players would play [2, 6, 19, 20, 21, 27, 44], to the extent that the player may be doing nothing more than following orders from the game [2, 19, 20, 27, 44]. These games are technically accessible to players who are blind, but they are far from the same game that sighted players would play, and so are not equivalently accessible.

The reason is that when making an existing type of game blind-accessible, there is a fundamental conflict between preserving the game's complexity and the game's pace. Preserving the former allows players to have the same sense of control that sighted players have when playing existing games, while preserving the latter keeps the action continuous and in real-time.

Figure 2 illustrates this tradeoff, with the sense of control that the game affords to the player on the vertical axis and

CHI 2018, April 21–26, 2018, Montréal, QC, Canada
© 2018 ACM. ISBN 978-1-4503-5620-6/18/04...$15.00
DOI: http://dx.doi.org/10.1145/3173574.3174090

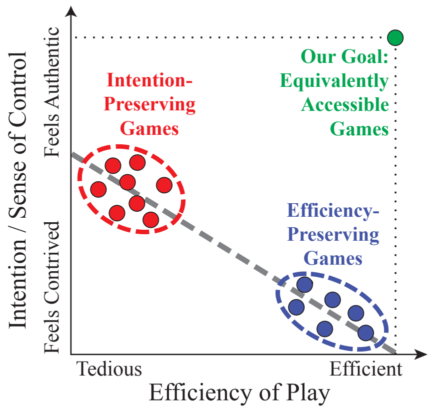

Figure 2. The intention–efficiency tradeoff. When designing blind-accessible games, game designers must choose between sacrificing each game's complexity — and by extension the player's *intention* or sense of control within the game — and the game's efficiency of play. Moreover, sophisticated actions such as cutting corners in racing games are difficult to incorporate even in intention-preserving games, so many do not feel fully authentic to play compared to what sighted players would play. Our goal is to overcome this tradeoff to help racing games become equivalently accessible to people who are blind.

the game's efficiency (pace) on the horizontal axis. For a game to be equivalently accessible to people who are blind, it should offer both, as the green dot at the top-right of the figure indicates. In practice, however, designers must sacrifice one of the two, causing existing blind-accessible games to fall into two distinct groups: what we call *efficiency-preserving games* and *intention-preserving games*.

Efficiency-preserving games, indicated by the blue dashed circle in Figure 2, are ones that sacrifice the sense of control that they afford players to keep their gameplay moving at a continuous pace. These include games such as *Blind Hero* [44], *Rock Vibe* [2], and *Blindfold Racer* [27]. They are often simplified versions of games that sighted players would play and often boil down the gameplay to a simple test of reaction speed. In *Blind Hero* and *Rock Vibe*, for example, players do not get to prepare for upcoming beats like sighted players would when playing *Guitar Hero* or *Rock Band*, which these games were based on. Rather, players are tasked with pressing buttons as soon as they feel corresponding vibration cues.

Intention-preserving games, indicated by the red dashed circle in Figure 2, are ones that sacrifice their efficiency of play to maintain more of their complexity and, by extension, give players a greater sense of control. These include *Terraformers* [42] and *Blindfold Color Crush* [29]. They are often cumbersome to play because they force players to navigate menus and process many audio cues just to understand what the current situation in the game is at any given time. Moreover, although they preserve much of their complexity and sense of control, they cannot preserve it all: complex actions such as cutting corners and performing head shots remain out of reach.

In this paper, we present the *racing auditory display (RAD)*, an audio-based user interface with the goal of overcoming this tradeoff to help racing games become equivalently accessible to people who are blind. The RAD comprises two novel sonification techniques: the *sound slider* for understanding a car's speed and trajectory on a racetrack and the *turn indicator system* for alerting players of the direction, sharpness, length, and timing of upcoming turns. Figure 1 shows a participant who is congenitally blind playing a racing game with the RAD.

We conducted two user studies to investigate whether the RAD allows players who are blind to play racing games at the same pace and with the same level of control as sighted players can. In the first study, we found that players preferred to play a racing game using the RAD over that of *Mach 1* [6], a popular blind-accessible racing game. In the second study, we found that the RAD makes it possible for a gamer who is blind to race as well on a complex racetrack as casual sighted players do. When that gamer raced using the RAD, there was no significant difference between his lap times or driving paths compared to those of casual players racing with sight.

INTENTION AND ITS ROLE IN RACING GAMES

Here, we introduce the concept of intention to describe what we mean by sense of control more precisely, and will illustrate how this concept applies to racing games. This concept can be used to examine whether a game gives players a high sense of control and, if not, how it can be changed to do so.

Intention is the process of "allowing and encouraging players to do things [within games] intentionally" [7, 8, 9]. More specifically, it is the process of "making an implementable plan of one's own creation in response to the current situation in the game world and one's understanding of the game play options" [7, 8, 9]. By breaking this definition down into parts, we can see that for a game to support intention, it must help the player perform the following three activities:

1. Understand the current situation in the game.
2. Understand what game play options are currently available.
3. Make an implementable plan of their own creation.

These activities are analogous to the three components of Yuan et al.'s game interaction model [45]. When we say that a game affords players a high sense of control, we mean that the game supports intention, which more precisely means that the game supports the player in performing each of the three activities listed above. For a blind-accessible video game to be *equivalently accessible* to people who are blind, it must support these three elements of intention without sacrificing the game's pace — overcoming the tradeoff in Figure 2 — and without simplifying the gameplay.

To support the first activity, racing games must help players understand all aspects of their current situation that are relevant to racing: their vehicle's position and orientation on the racetrack, a general sense of its current speed, the nature of any upcoming turns, etc. The game does not need to help players understand aspects of the current situation that are not relevant to racing, such as their vehicle's paint color or even its precise speed in mph/kph. (In fact, many games such as the *Grand Theft Auto* series and most of the *Mario Kart* series do not show players their vehicle's speed.) To support the third activity, racing games should make it possible for players to form strategies such as cutting corners or positioning themselves to better handle an upcoming turn.

RELATED WORK

Our work builds on two areas of research: work on developing audio navigation systems and work on developing blind-accessible racing games and driver assistance systems.

Audio Navigation Systems

Audio navigation systems help people who are blind navigate on foot from one place to another in the real world. They consist of a GPS tracker, a computing device, and a pair of headphones. Perhaps the most archetypal examples are audioGPS [17] and SWAN [43] (short for System for Wearable Audio Navigation), which both guide users from their current location to their destination via a sequence of waypoints that they must reach along the way. The user must follow a sound known as an *acoustic beacon* to travel from waypoint to waypoint until they reach their destination.

Most research in this area has focused on how to perfect these types of systems, such as discovering which type of sounds are easier to localize and follow [35] or how large each waypoint's "capture radius" should be [39, 40]. These systems, however, are unsuitable for racing games for two reasons. First, they assume that the user is walking and has the flexibility to stop and rotate to center the acoustic beacon in front of them. Second, using these systems amounts to simply following orders, while video games should afford players a high sense of control over what they are doing.

Blind-Accessible Racing Games

A number of driving systems and racing games currently employ mechanisms to assist drivers and players who are blind. Here, we will survey three blind-accessible video games and a blind driver assistance system, each employing a different user interface for driving a car on a virtual track.

Blindfold Racer (iOS, 2014)

Blindfold Racer [27] is an audio racing game developed on iOS by Marty Schultz as part of his series of blind-accessible smartphone games. In *Blindfold Racer*, players steer by rotating their mobile device left and right as they would a steering wheel. The goal is to drive to the end of a track without hitting fences on the track's sides. The player can also adjust their speed to three fixed values by swiping up or down on their device's touchscreen. The game outputs sound in stereo and pans a music track between the left and right channels as a means of displaying the car's lateral position on the track. It will play exclusively in the left channel if the player's car is adjacent to the left side of the track and vice-versa.

Treasure and animals are indicated using repeating audio samples that grow louder as the player approaches them. The player should try to center the sounds of treasure between the left and right channels to collect the treasure, and they should keep the sounds of animals panned to the left or right to avoid hitting the animals.

With respect to Figure 2, we would classify *Blindfold Racer* as an efficiency-preserving game. While *Blindfold Racer* moves at a pace that is just as fast as racing games with graphics, the three elements of player intention as described in the previous section are limited in *Blindfold Racer* compared to racing games that sighted players would play.

In *Blindfold Racer*, it is not possible for the player to anticipate upcoming turns, accelerate and decelerate in an analog manner, or perform higher level strategies such as cutting corners. In fact, the developer explains that there is no concept of vehicle physics, that tracks are modeled using a simplified geometry that requires straightaways to be in the same direction and all turns to be less than 90° [26], and that car steering is simplified so that the car will move in that straightaway direction when the mobile device is tilted to the center position [28].

Mach 1 (PC, 2003)

Mach 1 [6] is an audio-based racing game published on PC by Jim Kitchen eleven years before *Blindfold Racer* was released. The player's goal is similar that from *Blindfold Racer*, but in *Mach 1* there are no obstacles present on the track. Players accelerate, decelerate, and steer using a USB steering wheel or controller joystick, and they can press a button to have a voice speak their current lateral position on the track: a number from 1 to 100 where 1 represents the track's left edge, 100 the right edge, and 50 the center. The player should tap the button repeatedly to monitor their lateral position continuously.

As the player approaches an upcoming turn, the game will loop a predetermined sound effect in the left or right stereo audio channel depending on the direction of the turn. The sound effect starts playing quietly but grows louder as the player approaches the beginning of the turn. The game will play a thumping sound as the player reaches the turn, and the process will repeat to signify the end of the turn: a random looping sound effect growing louder followed by a thump.

Unlike *Blindfold Racer*, *Mach 1* allows players to anticipate upcoming turns and accelerate and decelerate in an analog manner. Still, players do not have full freedom to "read the road" since it is difficult to determine from the increasing volume effect exactly when a turn will begin and the game only alerts players of a single upcoming turn or straightaway at a time. As with *Blindfold Racer*, there is no concept of vehicle physics, tracks are modeled using a simplified geometry (so there is no concept of cutting corners), and car steering is simplified so that the car will move in a straightaway direction whenever the player lets go of the steering.

Top Speed Series (PC, 2004)

The *Top Speed* [11] series is a series of racing games released on PC by a team of four developers. The goal for players is the same as in *Blindfold Racer* and *Mach 1*: to reach the end of the track as quickly as possible without hitting the sides. *Top Speed 2* and *3* support multiplayer races, though the cars cannot collide with each other or interact with each other in any way. Players steer with a joystick controller as in *Mach 1*.

Like *Blindfold Racer*, a sound is panned between the left and right channels as a means of displaying the player's lateral position on the track. In the *Top Speed* series, however, that sound is the sound of the player's car's engine. A speech clip saying a phrase such as "easy left" or "hard right" will play when the player enters a turn. These phrases describe the direction and sharpness of the turn, and the player must react quickly by steering the appropriate amount. As with *Blindfold Racer* and *Mach 1*, there is no concept of vehicle physics, tracks are modeled using a simplified geometry, and car steering is simplified so that the car will move in a straightaway direction when the player lets go of the steering.

Sucu and Folmer's Haptic Steering Interface
Sucu and Folmer's haptic steering interface [34] is a driver assistance system published as a response to the National Federation of the Blind's Blind Driver Challenge [22], an initiative to make it possible for people who are blind to drive a car by themselves. The driver steers with a steering wheel and has rumble motors (in this case, PlayStation Move controllers) attached to the back of their hands.

At each time step, the system computes the location of what Sucu and Folmer call a *target point*, which is the point on the median of the track a fixed distance ahead of the driver's current position. If the car's current heading points too far away from the target point's direction, the system will vibrate the left or right rumble motors. The analogy is that of a rumble strip on the side of a highway: if the vibration is felt on the right the player should steer to the left and vice-versa.

Although the authors state that making a racing game with this system is promising future work, we feel that its current goal as a driver assistance system runs contrary to supporting intention. When drivers use this system, they must follow its orders as soon as those orders are felt and nothing more. Our interface, by contrast, is designed to support intention.

THE RACING AUDITORY DISPLAY (RAD)
In this section we introduce the *racing auditory display (RAD)*, a user interface whose goal is to help racing games become equivalently accessible to people who are blind. The RAD was designed with the principle that it should not just tell players what to do but rather give them enough relevant information to form a plan of action themselves.

The RAD comprises two novel sonification techniques: the *sound slider* and the *turn indicator system*. The *sound slider* helps the player understand their car's speed and trajectory on a racetrack while the *turn indicator system* alerts players of the direction, sharpness, length, and timing of upcoming turns well in advance of the actual turns. Together, the techniques allow players to understand aspects about the race and perform a wide variety of actions that are not possible to understand and perform in current blind-accessible racing games.

The RAD's Sound Slider
The RAD's sound slider is a novel mechanism for displaying a value within a range using spatialized (3D) sound. It is analogous to a traditional user interface slider, where the slider's track is a line segment in the 3D soundscape and the slider's handle is replaced with a virtual speaker (sound emitter). The position of the speaker on the virtual track represents the slider's displayed value, where one end of the track represents the slider's minimum value, the other its maximum value, and positions in between intermediate values. The slider's value is for display only: the speaker cannot be manually manipulated by the user like a traditional user interface slider's handle can.

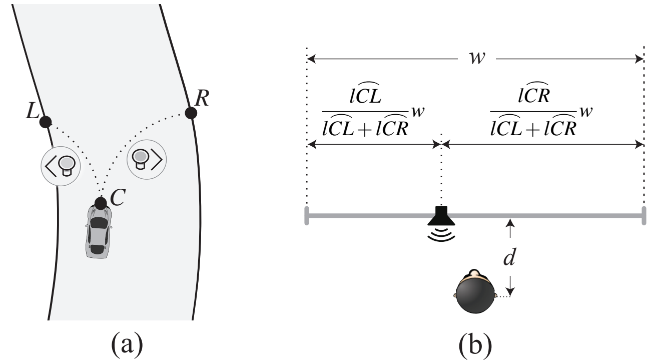

Figure 3. The RAD's sound slider. (a) Sample car pose showing what the car's trajectories would be if the player were to steer fully left or fully right. (b) Overhead view of corresponding rendered spatialized (3D) soundscape. The RAD's sound slider is a speaker emitting the car's engine noise whose lateral position in the soundscape tracks the ratio of the trajectories' lengths. The ratio represents the player's relative risk of hitting either edge of the track. In this case, the player will hear the car's engine right in front of their face but slightly to the left.

Figure 3 shows the specific sound slider configuration that we propose for blind-accessible racing games. The slider's track is a virtual horizontal bar of width w placed a distance d in front of the player's face in the soundscape. In our prototype racing game, w is 65 m and d is 12 m. The speaker emits the sound of the player's car's engine and slides left and right along the bar as the sound slider updates its value.

We explained the concept of the sound slider to our studies' participants as follows. We asked them to imagine being behind the car that they were controlling, so they could hear the sound of the car's engine right in front of their face. The car's sound will move left or right as the car becomes more at risk of hitting the track's left or right edges, respectively. When they steer, they control the car's sound directly, so if they hear the car's sound move far toward the left, they will want to steer right to bring the sound back toward the center.

If the player is in a turn and not turning nearly as sharply as they need to, perhaps because they are going too fast, the sound slider will emit a tire screeching sound adapted from [5] from the same position as the car's engine sound. This acts to warn the player that they must slow down by hitting the brake or letting go of the accelerator to properly complete the turn.

Computing the Slider Value
Figure 3 illustrates how the RAD computes the sound slider value to display to the player. Given the car's current position, orientation, and speed on the race track, the RAD computes the trajectories that the car would follow if the player was to steer fully to the left or right. It models these trajectories as circular arcs which we denote as $\widehat{CL}$ and $\widehat{CR}$, respectively. The radii of the arcs are modeled as being directly proportional to the car's current speed, where the constant of proportionality represents how sharply the car turns. Through manual tuning, we found its value in our prototype game to be roughly 1.6.

Next, the RAD finds the points at which the trajectories intersect the track's edges, then it computes the respective arc lengths $l\widehat{CL}$ and $l\widehat{CR}$ from the car's position to these points.

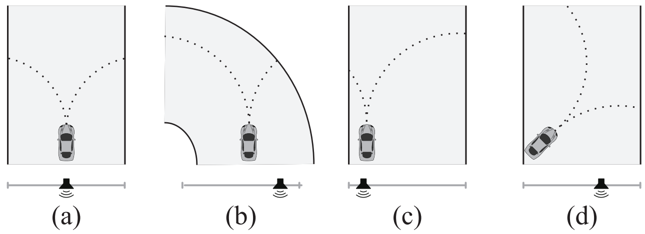

Figure 4. Four car poses and their corresponding sound slider values. Though the car's lateral position is the same between (a) and (b) and between (c) and (d), the corresponding sound slider values are very different. This is because the left and right trajectories' lengths — and therefore the relative risks of hitting the left and right sides of the track — are very different in each pair of cases.

$l\widehat{CL}$ and $l\widehat{CR}$ represent the distances the car would travel before hitting an edge were the player to steer fully to the left or right, respectively. Finally, the RAD sets the sound slider value to the following quantity, which we call the *time-to-impact ratio*:

$$\text{Slider Display Value} = \frac{l\widehat{CL}}{l\widehat{CL} + l\widehat{CR}}. \tag{1}$$

The sound slider's leftmost and rightmost positions are represented by zero and one respectively. The system will set the slider value to something different than the time-to-impact ratio in two cases. The first case is when both trajectories hit the track's left edge — which means that the player is driving toward the left edge — or when the player's car is currently off the track on the left side. The second is the analogous case for the track's right edge. In these cases, the system will set the slider value to zero and one, respectively.

From Lateral Position to Relative Risk

The algorithm described above represents a new approach for letting players know where they are situated on a racetrack. Unlike the stereo pan values in *Blindfold Racer*, *Mach 1*, and the *Top Speed* series, the sound slider's display value is *not* a direct reflection of the car's lateral position on the track. Rather, it is a function of the car's relative risk of hitting the track's left or right sides if the player wanted to. This distinction is what makes the sound slider intuitive even with the complex vehicle physics, steering behaviors, and track geometries that are present in modern racing games.

Figure 4 illustrates the benefit of updating the auditory display using our trajectory-based approach over the car's lateral position alone. The car's lateral position is the same between Figures 4(a) and (b) and between Figures 4(c) and (d), but the player's relative risks of hitting the track's left and right sides is very different between each pair. In Figure 4(b), for example, the player is much more at risk of hitting the track's right side than they are in Figure 4(a) due to the sharp left turn in Figure 4(b), and the player should be aware of this.

As another example, the car's heading in Figure 4(c) puts the car more at risk of hitting the track's left edge than its right edge, while its heading in Figure 4(d) does the opposite. The player should be aware of this as well. The sound slider's trajectory-based approach communicates these risks.

Overcoming the Intention–Efficiency Tradeoff

We argue that the RAD's trajectory-based approach to computing its sound slider's displayed value allows it to overcome the intention–efficiency tradeoff that plagues other blind-accessible racing game interfaces (Figure 2).

The reason is that this approach distills many pieces of information — the car's lateral position on the track, its heading with respect to the track's, its speed, the track's width, whether the track is about to immediately turn, and more — into a single measure that we hope is as relevant to the process of racing as all of that information put together. Moreover, it does so in a way that gives players the freedom to decide how risky they would like to race: whether they should cut corners by being close to hitting the track's inside edge or stay closer to the track's center. Sucu and Folmer's haptic driving interface, by comparison, eliminates intention by simply telling players which way they should steer at any given time.

We liken this process of distilling the many pieces of information to that of dimensionality reduction in machine learning and statistics. Dimensionality reduction is important in these fields because it boosts classification speed and removes redundancies in the representations of features. In the RAD's sound slider's case, the process reduces the amount of information that must be conveyed to the player while preserving its meaning and relevancy.

The RAD's Turn Indicator System

The RAD's turn indicator system uses spatialized (3D) sound cues to alert players of the direction, sharpness, and timing of upcoming turns and the length of in-progress turns. It works by playing a series of four beeps that trigger when the player's car crosses four corresponding and equally spaced distance markers placed ahead of the turn. The last beep is a continuous sound that begins playing just as the turn begins and continues sounding until the player completes the turn. Left and right turns are indicated by beeps emitted from the left and right ends of the sound slider's track, respectively. Overlapping turns are indicated by overlapping sets of beeps.

By using four beeps to indicate turns, the player is given enough time to recognize the beeps' rhythm and anticipate the timing of the last beep, which marks the beginning of the turn. The player can then time their steering accordingly, cutting the corner if they wish by starting to steer a little before the last beep begins sounding. The distance markers triggering the four beeps are spaced 20 m apart, giving the player 1.7 s of advance notice of the turn when they are driving at the maximum speed of 35 m/s (approximately 75 mph).

The beep sounds themselves are modified recordings of a distant engine hum, adapted from [10]. Low pitched beeps indicate soft turns, moderately pitched beeps indicate moderate turns, and high-pitched beeps indicate sharp turns. We defined soft turns as those which turn less than 0.3° per meter of track and sharp turns as those which turn more than 1° per meter of track. When a turn changes sharpness partway through, as in Turns 7a and 7b in Figure 6, the system treats each part as a separate turn and alerts the player accordingly.

Figure 5. Racing game prototype implemented in Unity.

In addition to playing the beeps, the system announces each upcoming turn's number, where Turn 1 is the track's first turn, Turn 2 the second, and so on. The number is announced at the same time as the first beep, and the goal is to help players learn the track over time as sighted players do.

Supported Actions

The RAD's sound slider and turn indicator system work together to support the following actions:

Understand the car's current speed: The sound slider's car engine sound will increase in pitch as the engine revs up, giving the player a general sense of the car's current speed.

Align the car with the track's heading: If the player's car is not aligned with the track's heading, the car engine sound will begin moving left or right on the sound slider. The player can align their car with the track's heading by steering until the engine sound stops moving.

Learn the track's layout: The turn indicator's turn number announcements help the player remember specific turns and sequences of turns.

Profile upcoming turns: The direction, sharpness, timing, and length of upcoming turns are indicated by the turn indicator beeps' left vs. right location in the soundscape, the beeps' pitch, the beeps' rhythm, and the fourth beep's duration, respectively.

Cut corners: By steering into a turn just before the turn indicator's fourth beep, the player can cut corners. The player can maintain an inside position during the turn by steering such that the engine sound moves toward the inside of the turn on the sound slider (and away from the slider's center).

Choose an early or late apex: By steering into a turn just before or after the turn indicator's fourth beep, the player can choose between taking an earlier apex or a later apex [25].

Position the car for an optimal turning path: By steering the car in a way that moves the engine sound to a desired position on the sound slider ahead of a turn, the player can position the car for a more optimal driving path.

Know when braking is needed to complete a turn: The sound slider emits a tire screeching sound when the player is going too fast in a turn to turn sharply enough.

RACING GAME PROTOTYPE

As a proof of concept, we developed a racing game using the Unity game engine (version 5.4.2) [37] and implemented the RAD in that racing game. Our prototype, shown in Figure 5, is an extension of TurnTheGameOn's Racing Game Template [23]. It features full 3D graphics and uses realistic vehicle physics from the Edy's Vehicle Physics package [13].

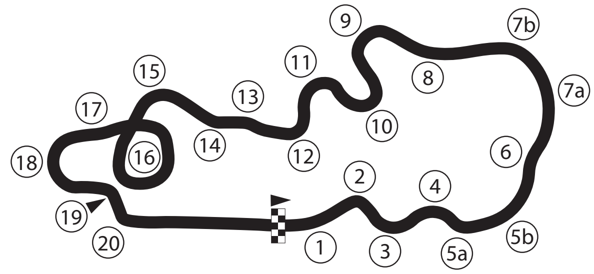

Figure 6. Circuit diagram for the racetrack used in user studies. This track is difficult and much more complex than ones in previous blind-accessible games, featuring a wide variety of turns.

Our game is played with a Sony DUALSHOCK 4 (PlayStation 4 controller) [31] and a standard pair of headphones. The controls are mapped similarly to other PlayStation 4 racing games: the left analog stick controls steering, R2 (the right analog trigger) is gas/acceleration, L2 (the left analog trigger) is brake and reverse, and R1 (the right shoulder button) is the handbrake. In case of a crash, participants could press the Triangle button to reset their car to the center of the track.

To generate spatialized (3D) sound, we enabled the simple demo spatializer provided by Unity's Audio Spatializer SDK [38]. The spatializer applies a direct head-related transfer function (HRTF) that is based on a data set generated from a KEMAR dummy-head [14].

The Racetrack

Figure 6 shows the track that we used for our user studies. The track was developed internally at Unity [36] and is much more complex than ones in previous blind-accessible games. It features a wide variety of turns: soft, moderate, and sharp turns; a long straightaway; a series of hairpin turns (Turns 9–11) that require players to slow down; a 270° turn (Turn 16); several short kinks in the track (Turns 8, 13, 14, & 17); several series of esses (Turns 1–5 & 19–20); long and gradual turns (Turns 5b & 7a); and turns that vary in sharpness as they progress (Turns 5, 7 & 17). The track is 3,641 m long, 19 m wide, and has 20 turns in total.

STUDY 1: THE RAD VS. OTHER INTERFACES

We performed a study with both blind users and sighted users wearing blindfolds to compare the RAD with *Mach 1*'s interface [6] and Sucu and Folmer's haptic steering interface [34]. These interfaces represent a broad range of design alternatives.

Our study had three goals. First, we wanted to determine how well the average person would perform with each of these user interfaces with a short amount of training. Second, we wanted to see how users would rank the three interfaces by order of preference. Third, we wanted to observe how well each interface helped players anticipate upcoming turns.

Study 1 Participants

Our study included fifteen participants. Three of them — P4, P8, and P11 — were blind their entire lives and the rest were sighted but blindfolded. Seven were age 16–25 and the rest were age 26–35; four were female and the rest were male. Our study was approved by our institution's Institutional Review Board, and parents were present with minors.

We recruited P4, P8, and P11 through Helen Keller Services for the Blind. P4 had no prior experience with racing games, while P8 and P11 had played just one audio racing game each years prior: *Top Speed* and *Blindfold Racer*, respectively. P8, however, described himself as a gamer and had played other types of audio games before, namely an RPG [12] and a first-person shooter [18]. Six of the twelve sighted participants had at least a moderate amount of experience playing video games, and the rest had very little experience. Of those with moderate experience, three would describe themselves as gamers.

We should note that participants who are sighted but blindfolded are generally not suitable proxies for participants who are blind. Silverman et al. [30] found, for example, that sighted but blindfolded participants can be biased by the initial challenge of becoming blind, therefore judging the capability of people who are blind as much less than it actually is. As a result, and as is good practice [24], we will present the results from these two groups of participants separately.

Study 1 Procedure

In the study, participants raced using each of the three user interfaces in a counterbalanced order while we observed them. Participants controlled their car using a Sony DUALSHOCK 4 (PlayStation 4 controller) [31] and wore a pair of AmazonBasics on-ear headphones [3]. All sighted participants wore blindfolds and could not see us loading the track, nor could they see what they were doing in the games. We told the participants that our team developed all three of the user interfaces. Each session lasted approximately two hours.

For both the haptic steering interface and the RAD, we had participants play our prototype racing game in which we implemented both. For *Mach 1*'s interface, however, we had participants play *Mach 1* itself. We did this because *Mach 1* uses simplified models rather than realistic designs for its tracks and steering system, and its user interface was designed with the simplified models in mind. Since we loaded *Mach 1* into a level before the study began, the participants were not aware that they were playing a previously published game.

Like other modern game controllers, the DUALSHOCK 4's rumble motors are different in size, with the left motor being significantly larger than the right motor. Since the haptic steering interface requires identical rumble motors for the user's left and right hands, however, we replaced our DUALSHOCK 4's left motor with one identical to the right motor. We clamped the motors' vibration intensity to 50% of its normal maximum to make it easier for players to distinguish between the motors.

We began each user interface trial by training each participant with hands-on instruction for 15–20 minutes on how to use the interface. We created two training tracks in our prototype — a square track with rounded corners and a figure eight track — to help the participants relate the interfaces' feedback with easily understandable shapes. We told participants to play with each interface until they understood how they worked.

We followed the three trials with a survey asking participants to rank the three interfaces from their most to least favorite, rate how well each interface helped them anticipate upcoming turns on a 20-point Likert scale in which higher values were better, and offer feedback justifying their ratings. Participants' feedback was extensive. To analyze it, we first transcribed it in full, then — via a series of repeated readings — wrote topic labels for each piece describing what it was talking about. We then tallied positive and negative opinions for each identified topic. We report these numbers along with the quotes that were most descriptive and representative of overall opinions.

Study 1 Results: Participants Who Are Blind

User Interface Ranking

P4 ranked the user interfaces from best to worst as *Mach 1*'s interface, the RAD, and Sucu and Folmer's interface, in that order. Both P8 and P11 ranked them as the RAD, Sucu and Folmer's interface, and *Mach 1*'s interface, in that order.

Awareness of Upcoming Turns

On a 20-point Likert scale in which higher values are better, P4 rated their ability to anticipate upcoming turns using the RAD, Sucu and Folmer's interface, and *Mach 1*'s interface as 8, 11, and 15, respectively. P8's ratings were 18, 11, and 7, respectively, while P11's were 5, 10, and 12, respectively. The difference is sharp between P8 and the others. Both P4 and P11 had very little experience playing video games while P8 considers himself a gamer. Although P11 rated the RAD lowest and *Mach 1*'s interface highest on this scale, she ranked the RAD as the best of the three interfaces overall and *Mach 1*'s interface the worst of the three.

Driving Performance in Our Prototype Game

Of the participants who are blind, only P8 was able to complete a full lap, and he did so with each of the three user interfaces. P8, the only self-described gamer among the three, completed our track (Figure 6) with zero major collisions on his first try with both Sucu and Folmer's interface and the RAD.

Neither P4 nor P11 could complete a full lap using any of the user interfaces, though all three participants completed our square and figure eight training tracks using both the RAD and Sucu and Folmer's interface. Recall that there were no training tracks for *Mach 1*. We should note that our track (Figure 6) resembles what one would find in a real video game and is very challenging compared to ones in existing blind-accessible racing games. Sucu and Folmer, for example, tested a basic oval and still found many crashes [33, 34].

Qualitative Feedback: Mach 1's Interface

P4 rated *Mach 1*'s user interface as his favorite because it was the only one to explicitly read out the car's lateral position and because he felt that he "had [more] time to think and react" to its cues compared to the other interfaces. This is likely because *Mach 1* does not provide continuous feedback about the car's positioning as the other interfaces do; rather, it reads the information whenever a particular button is pressed.

Both P8 and P11 found *Mach 1*'s interface to be the worst of the three, with P11 saying that it was "the hardest" and "hard to use properly." P8 said that while "it had pretty much [all of the game elements that] [he] would expect from a racing game," it was "very difficult [to use because] there are so many things going on" at the same time, including many "sounds that are not relevant." He also said that it "was difficult [. . .] knowing when you are in the turn and when you are out of the turn" because the steadily increasing sound effect volume that it employs to indicate the beginnings of turns was not precise.

Qualitative Feedback: Haptic Steering Interface

P4 considered Sucu and Folmer's haptic steering interface to be the worst of the three "mainly due to not being able to see upcoming turns." P8 and P11 ranked the haptic steering interface in between their most and least favorite, with P11 saying that she "did not get to think about how to attack the turn[s]" and that "[using] it would have been easier if there was a warning in advance, when you should start turning." Still, P11 felt that while the lateral positioning feedback "wasn't exact[ly precise], it was to the point that I [...] could kind of tell if the car wasn't in the center."

P8 said that the vibrations "didn't give much [of an] indication of how sharp [each] turn was," preventing him from making strategies such as, "I shouldn't turn too much here to avoid colliding with the [inside] wall." He felt that "the experience would be better, perhaps, by "mak[ing] the game controller vibrate more or less" in intensity depending on the sharpness of the turn. Sucu and Folmer, however, found users' performance with continuous vibration feedback to be worse than with binary (on/off) feedback [33, 34].

Qualitative Feedback: Racing Auditory Display (RAD)

P4 ranked the RAD in between his most and least favorite, saying "it is definitely better than the vibration method" (Sucu and Folmer's interface) but that he "still had a hard time" because it was "confusing to parse between the two types of sounds" (the sound slider and the turn indicator system). Both P8 and P11 considered the RAD to be their favorite interface, with P8 saying that it was "very, very logically built up [. . .] because it gave [him] an indication of how sharp the turns were [and] for how long [he was] in [each] turn."

P8 felt that distinguishing between soft, moderate, and sharp turns "worked very well with the tonality of the sound." P11, on the other hand, said that while she "got the concept, it was [. . .] harder to put the concept into use," finding the RAD "difficult to [learn] but very entertaining" to play with. She remarked that with the RAD "the feeling of the game is fast-paced," adding, "Yes, you have the time [to plan], but sometimes you might not be able to [pull it off]." P8 said that he liked how the RAD did not "constantly sa[y] 'Do this, do that,'" and followed up by saying, "After the training was done, I had the possibility of doing whatever I wanted to." These last two comments suggest that the RAD supports intention.

Study 1 Results: Sighted but Blindfolded Participants

User Interface Ranking

Figure 7 shows how participants ranked each interface from most to least favorite. Sighted participants' rankings are those without the dot patterns. Six sighted participants chose the RAD as their preferred interface, five chose the haptic steering interface, and one chose *Mach 1*'s interface. Ten out of twelve sighted participants liked *Mach 1*'s interface the least.

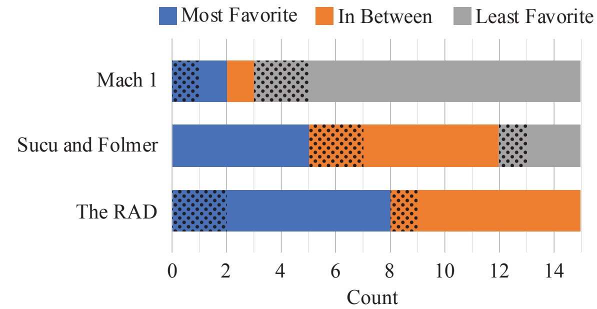

Figure 7. Participants' user interface rankings. The dot patterns indicate rankings from participants who are blind. Most disliked *Mach 1*'s interface, and eight of fifteen preferred the RAD's the most.

Awareness of Upcoming Turns

An ANOVA showed that the user interface has a significant main effect on the sighted participants' awareness of upcoming turns ($F_{2,22} = 4.83$, $p = 0.02$). Pairwise mean comparison showed that the only significant difference was between the RAD and *Mach 1*'s interface ($p < 0.05$). The mean (std. dev) ratings for this metric for the RAD, the haptic steering interface, and *Mach 1*'s interface are 13.0 (4.9), 8.8 (6.5), and 6.8 (5.5), respectively. This suggests that the RAD does a better job communicating the nature of upcoming turns for sighted players than *Mach 1*'s sound effects of increasing volume.

Driving Performance in Our Prototype Game

Ten out of twelve sighted participants were able to complete the track in Figure 6, five of which after crashing and resetting themselves many times. Their performance seemed to depend on their prior experience with video games: participants tended to perform well with both interfaces or poorly with both interfaces. All seven sighted participants with at least moderate video game experience completed the track, two of whom after crashing many times. By contrast, only three of the five participants with limited video game experience completed the track, all of whom after crashing many times.

These results suggest that both the RAD and the haptic steering interface make it possible for gamers to play racing games without sight, but neither can make a non-gamer proficient at playing racing games.

Qualitative Feedback: Mach 1's Interface

Of the three interfaces, sighted participants liked *Mach 1*'s the least in general. Though many mentioned that "it was relatively easier to understand [their] horizontal location with [this interface's spoken] numeric value[s]" (P2) than with the other interfaces' feedback, four lamented that "having numbers read to [them] took extra brain power [to process, making] it much more difficult for [them] to move forward quickly" (P5). All said that it "took [them] a while to sort out all the sounds that were going on" (P15) and that there was "too much auditory information for too long a period" (P9).

Ten felt that determining the position and length of turns was "very difficult" (P2) and that they could not determine the turns' sharpness at all because "the sound leading up to the thump which indicates [when] turn[s begin and end were] more confusing and disorienting than anything" (P13). A different set of ten felt that a "[big] difficulty was to determine the difference between the probe number [(lateral position)] and the speed of the vehicle" (P10).

Qualitative Feedback: Haptic Steering Interface

Ten sighted participants felt that this interface's vibrations were "easier to [learn and] focus on [compared to] the [other interfaces'] multiple sounds" (P10), but five felt that "turning and preparing for turns was completely out of [their] control" (P5) because "the only interaction [they] had was immediately responding to the vibrations" (P5), "conforming to the rumble indicators" (P14), or as P13 put it, "just [. . .] bouncing around from wall to wall trying to stay in the center." P5 added that she "had no idea when a turn was coming up, how sharp or long it would be, [or] whether or not it was actually a turn [she] was dealing with or simply trying to straighten [her]self out on a straightaway after a turn."

Some liked how "the rumble [being] binary [made it] really clear [to know] when you are 'good' or 'bad"' (P13) but six bemoaned the resulting lack of intention (though not using that word). Three mentioned that they would prefer having differing levels of vibration so they could tell "exactly how far [. . .] from the middle of the road" (P6) they are or "how sharp the turn was" (P1, P3). As mentioned earlier, however, Sucu and Folmer found that users crashed much more with such a system than with binary vibration feedback [33, 34].

Qualitative Feedback: Racing Auditory Display (RAD)

Eight sighted participants felt that the RAD's turn indicator system made them "well aware of the upcoming turns with their position and sharpness" (P3). Two of them, however, mentioned that the system was "sometimes confusing when [turns were] very short. . . " — in which case the fourth turn indicator beep would be very short — ". . . and/or followed immediately by another turn" (P1) — in which case the RAD would output multiple overlapping sets of beeps.

Four participants found the RAD difficult to use while two found it very natural. In particular, eight participants found it difficult to distinguish between the sound slider's engine sound and the turn indicators, with P5 mentioning that "as a full-sighted person [she is] not used to using every single sound as an informational cue and usually do[es]n't pay attention to such noises as engine volume." P2 and P5 sometimes found the RAD's sound slider "difficult to understand" (P2, P5) because "the location of the engine sound (left vs. right vs. middle) [can] change incredibly fast" as they enter sharp turns.

Seven participants mentioned that they were "almost always aware of which side of the track [they are] on" (P3) when using the RAD, with P3 adding, "[. . .] compared to both [of] the other methods where I was quite clueless." Seven participants felt that the RAD made them "fe[el] the most like [they were] racing" (P13) compared to the other interfaces. P9 found the RAD "fun and definitely the most immersive" of the three interfaces, and that with the RAD he "could actually visualize the car and its location."

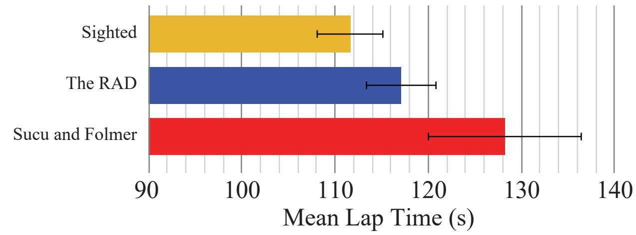

Figure 8. Mean lap times of P8 — a gamer who is blind — using Sucu and Folmer's haptic steering interface [34], P8 using the RAD, and sighted players using vision. The error bars indicate standard deviations. With the RAD, P8 races significantly better than he does using the haptic steering interface and comparably to casual players racing with sight.

STUDY 2: FIELD TEST WITH GAMER WHO IS BLIND

Our second study tests whether the RAD makes it possible for a player who is blind to race better than Sucu and Folmer's haptic steering interface does, and whether their racing performance can match that of a sighted player's.

Study 2 Procedure

In this study, we had participant P8 from our first study — our only participant that is both blind and considers himself a gamer — drive thirteen laps around the racetrack in Figure 6 using Sucu and Folmer's haptic steering interface and fourteen laps using the RAD. We recorded his lap times, full driving paths, and gameplay video of him racing as he played. The car starts at the beginning of the long straightaway in Figure 6 so that it can reach full speed by the start of the first lap. Our supplemental video shows P8's third lap ever on this track.

We then had eight sighted players (three female, five male) drive one to three laps around the track using sight as we recorded their lap times and driving paths. We used just one to three laps here because we found in a pilot study that sighted players' lap times did not improve over the course of driving 14 laps. The same was true for P8: his average lap time for his first three laps was 0.3 s faster than for his last three.

Study 2 Results

Figure 8 compares lap times for the three conditions: P8 using the haptic steering interface, P8 using the RAD, and sighted players using vision. The mean (std. dev) lap times are 128.2 s (8.2 s), 117.0 s (3.7 s), and 111.7 s (3.5 s), respectively. An ANOVA showed that the user interface has a significant main effect on the mean lap times ($F_{2,32} = 23.38$, $p < 0.0001$). Pairwise mean comparison showed that the differences were significant between every pair of interfaces ($p < 0.01$) except the RAD vs. sighted players using vision. This suggests that the RAD allowed P8 to race significantly better than the haptic steering interface did — saving an average of 11.2 s per lap — and comparably to that of players using sight. Only one of the sighted players, however, described themselves as a gamer.

Figure 9 compares typical driving paths from P8 using the haptic steering interface and the RAD, respectively. The haptic steering interface causes P8 to oscillate around the track's

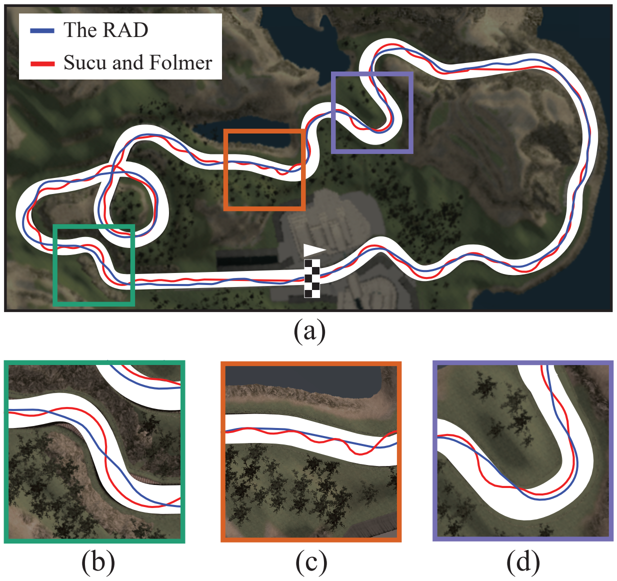

Figure 9. Sample driving paths of P8 — a gamer who is blind — using the RAD and using Sucu and Folmer's haptic steering interface [34]. We compare the paths for (a) the entire circuit, (b) an ess turn, (c) a near-straight section, and (d) a hairpin turn. P8 oscillates constantly with the haptic steering interface but drives more smoothly when using the RAD. He is also able to cut the corners in (b) using the RAD. Our supplemental video shows P8's third lap with the RAD's audio included.

center line for the entire lap, which is this interface's usual behavior since it works by vibrating the player's controller when their heading is too far away from that of a center target point [34]. By contrast, P8 drives in a much smoother path using the RAD. In Figure 9(b), for example, we see that P8 carves a nearly straight path through Turns 19 and 20 (which form an ess turn sequence) when using the RAD but follows the track's center line when using the haptic steering interface.

The mean (std. dev) driving path lengths are 3,639 m (74 m), 3,557 m (40 m), and 3,469 m (71 m) for the three respective conditions: P8 using the haptic steering interface, P8 using the RAD, and sighted players using vision. An ANOVA showed that the user interface has a significant main effect on the driving path length ($F_{2,32} = 19.21$, $p < 0.0001$). Pairwise mean comparison showed that the differences were significant between every pair of interfaces ($p < 0.05$ for the haptic steering interface vs. the RAD and $p < 0.01$ otherwise). This shows that P8 can perform shorter laps with the RAD than with the haptic steering interface (mainly by reducing oscillations), though not quite as short as laps made by players driving with sight.

HUMAN–COMPUTER INTERACTION (HCI) IMPLICATIONS

Though games especially benefit from intention, our work has broader implications within HCI. First, our definition of a sound slider is generic: a virtual speaker that indicates a value within a range by its position on a 3D line segment in the soundscape. For blind users, sound sliders can substitute for traditional UI sliders; brightness, temperature, or pressure gauges; progress bars; and any other display that displays a value within a range. They can also help users perform steering tasks in the classical sense [1] by representing a tunnel's width. Furthermore, the RAD can be used in place of AudioGPS [17] and SWAN [43] for pedestrian navigation tasks. AudioGPS and SWAN tell users know which way to walk, but the RAD can tell users how wide the path or bridge is, how much "wiggle room" they have, and whether they are in the middle or toward one side, helping them avoid oncoming foot traffic.

CONCLUSION, LIMITATIONS, AND FUTURE WORK

This paper offers a vision of how video games can go beyond just being blind-accessible to being *equivalently accessible* to people who are blind, allowing them to play with a similar sense of control (intention) and efficiency as sighted players can. To this end, we introduce the *racing auditory display (RAD)* to help racing games become equivalently accessible to people who are blind. It comprises two novel sonification techniques: the *sound slider* for understanding a car's speed and trajectory on a racetrack and the *turn indicator system* for alerting players of the direction, sharpness, length, and timing of upcoming turns.

Through a pair of empirical studies, we found that players preferred the RAD's interface over that of *Mach 1*, a popular blind-accessible racing game, and at times "felt like [they] had as much information as if [they] could see the track" (P1). We demonstrated that the RAD makes it possible for a gamer who is blind to race comparably to casual players using sight.

Still, there are several limitations to our studies and to the RAD. First, our study included just four self-described gamers and three people who are blind, so our results cannot be assumed to apply to everyone from these groups. A more thorough follow-up study targeting gamers who are blind would be needed for this. Second, the RAD relies on 3D sound spatialization. Not everyone can hear spatialized sounds correctly with off-the-shelf head-related transfer functions (HRTFs). Future games could allow players to load an HRTF from a profile so they can hear spatialized sound clearly in many different games.

Last, the RAD is not as effective with non-gamers and does not teach them "video game literacy" such as how video game vehicle handling works, nor is it effective at helping players recover from crashes or from driving off the track. A future version of the RAD could include a *Mach 1*-style probing feature for helping players learn the game mechanics and recover from crashes. We also think it would be feasible to extend the RAD to incorporate other racing game elements such as opponent vehicles, boosts, item pickups, and shortcuts.

We hope that just as user interface toolkits provide tools such as scrollbars, sliders, menus, and radio buttons that "just work" when software is published, game engines will one day include building blocks such as walls and track pieces that will "just work" with user interfaces such as the RAD or AudioGPS [17] when games are published to make all games blind-friendly.

ACKNOWLEDGMENTS

We would like to thank Russell Martello and Gus Chalkias at Helen Keller Services for the Blind (HKSB) for help recruiting participants, Daniel Sims for help conducting user studies and preparing figures, Sean Pagaduan and Janet Kayfetz for help writing, and the CHI reviewers for valuable suggestions.

REFERENCES

1. Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' Law: Models for Trajectory-based HCI Tasks. In *Proc. ACM SIGCHI Conf. Hum. Fact. in Comput. Sys. (CHI 1997)*. ACM Press, New York, NY, USA, 295–302. DOI: http://dx.doi.org/10.1145/258549.258760
2. Troy Allman, Rupinder K. Dhillon, Molly A.E. Landau, and Sri H. Kurniawan. 2009. Rock Vibe: Rock Band® Computer Games for People with No or Limited Vision. In *Proceeding Elev. Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2009)*. ACM Press, New York, New York, USA, 51–58. DOI: http://dx.doi.org/10.1145/1639642.1639653
3. Amazon.com, Inc. 2014. AmazonBasics Lightweight On-Ear Headphones. (25 November 2014). Retrieved September 17, 2017 from http://a.co/9OFo5NC.
4. Matthew Tylee Atkinson and Sabahattin Gucukoglu. 2004. The AGRIP Project (AudioQuake). (2004). Retrieved September 16, 2017 from http://agrip.org.uk.
5. audible-edge. 2009. Chrysler LHS tire squeal 04 (04-25-2009).wav. Audio file. (27 April 2009). Retrieved September 14, 2017 from https://freesound.org/s/71739/.
6. audiogames archive. 2015. Games by Jim Kitchen. (2015). Retrieved September 16, 2017 from http://www.agarchive.net/pages/devs/kitchensinc.html.
7. Doug Church. 1999a. Formal Abstract Design Tools. *Game Developer Magazine* 6, 7 (August 1999), 28.
8. Doug Church. 1999b. Formal Abstract Design Tools. Gamasutra. (16 July 1999). Retrieved August 25, 2017 from http://www.gamasutra.com/view/feature/131764/formal_abstract_design_tools.php.
9. Doug Church. 2006. *The Game Design Reader: A Rules of Play Anthology*. MIT Press, Cambridge, MA, USA, Chapter Formal Abstract Design Tools, 366–381.
10. CosmicD. 2007. engine_hum_new.wav. Audio file. (10 April 2007). Retrieved September 14, 2017 from https://freesound.org/s/33503/.
11. Leonard de Ruijter, Pieter de Ruijter, Bram Duvigneau, and Davy Loots. 2004. Playing in the Dark: Top Speed. (2004). Retrieved September 16, 2017 from http://www.playinginthedark.net/topspeed_e.php.
12. Driftwood Audio Entertainment. 2010. Entombed - An RPG Roguelike Game for the Blind and Visually Impaired. (2010). Retrieved January 2, 2018 from http://www.blind-games.com/entombed.aspx.
13. Angel "Edy" García. 2015. Edy's Vehicle Physics. (2015). Retrieved September 15, 2017 from http://www.edy.es/dev/vehicle-physics/.
14. William G. Gardner and Keith D. Martin. 1995. HRTF Measurements of a KEMAR. *J. Acoust. Soc. Am.* 97, 6 (June 1995), 3907–3908. DOI: http://dx.doi.org/10.1121/1.412407
15. GMA Games. 2005. Shades of Doom Version 1.2. (2005). Retrieved September 16, 2017 from http://www.gmagames.com/sod.html.
16. Terri Hedgpeth, John A Black Jr., and Sethuraman Panchanathan. 2006. A Demonstration of the iCARE Portable Reader. In *Proc. 8th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2006)*. ACM Press, New York, New York, USA, 279–280. DOI: http://dx.doi.org/10.1145/1168987.1169054
17. Simon Holland, David R. Morse, and Henrik Gedenryd. 2002. AudioGPS: Spatial audio navigation with a minimal attention interface. *Pers. Ubiquitous Comput.* 6, 4 (September 2002), 253–259. DOI: http://dx.doi.org/10.1007/s007790200025
18. Jeremy "Aprone" Kaldobsky. 2011. Aprone's Accessible Software and Games. (2011). Retrieved January 2, 2018 from http://www.kaldobsky.com/audiogames/.
19. Joy Kim and Jonathan Ricaurte. 2011. TapBeats: Accessible and Mobile Casual Gaming. In *Proc. 13th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2011)*. ACM Press, New York, New York, USA, 285–286. DOI: http://dx.doi.org/10.1145/2049536.2049609
20. Daniel Miller, Aaron Parecki, and Sarah A. Douglas. 2007. Finger Dance: A Sound Game for Blind People. In *Proc. 9th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2007)*. ACM Press, New York, New York, USA, 253–254. DOI: http://dx.doi.org/10.1145/1296843.1296898
21. Tony Morelli, John Foley, and Eelke Folmer. 2010. VI-Bowling: A Tactile Spatial Exergame for Individuals with Visual Impairments. In *Proc. 12th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2010)*. ACM Press, New York, New York, USA, 179–186. DOI: http://dx.doi.org/10.1145/1878803.1878836
22. National Federation of the Blind. 2013. Blind Driver Challenge. (2013). Retrieved September 12, 2017 from http://www.blinddriverchallenge.org.
23. Stephen "TurnTheGameOn" O'Donnell. 2015. Racing Game Template. (2015). Retrieved January 3, 2018 from https://www.turnthegameon.com/racing-game-template.
24. Andrew Sears and Vicki L. Hanson. 2012. Representing users in accessibility research. *ACM Trans. Access. Comput.* 4, 2 (2012), 1–6. DOI: http://dx.doi.org/10.1145/2141943.2141945
25. Seas. 2012. Cornering Technique. (2012). Retrieved September 14, 2017 from http://www.formula1-dictionary.net/cornering_tech.html.
26. Marty Shultz. 2013. Blindfold Racer: Creating the Track. Blog post. (3 December 2013). Retrieved September 12, 2017 from https://blindfoldgames.org/2013/12/03/creating-the-track/.
27. Marty Shultz. 2014a. Blindfold Racer. (2014). Retrieved September 16, 2017 from http://www.blindfoldracer.com.

28. Marty Shultz. 2014b. Blindfold Racer: It's Too Hard to Control. Blog post. (10 March 2014). Retrieved September 12, 2017 from https://blindfoldgames.org/2014/03/10/blindfold-racer-its-too-hard-to-control/.

29. Marty Shultz. 2015. Blindfold Color Crush. (2015). Retrieved September 16, 2017 from https://blindfoldgames.org/user-guides/blindfold-color-crush-user-guide/.

30. Arielle M. Silverman, Jason D. Gwinn, and Leaf Van Boven. 2015. Stumbling in Their Shoes: Disability Simulations Reduce Judged Capabilities of Disabled People. *Soc. Psychol. Personal. Sci.* 6, 4 (2015), 464–471. DOI:http://dx.doi.org/10.1177/1948550614559650

31. Sony Interactive Entertainment. 2013. DUALSHOCK 4 Wireless Controller. (2013). Retrieved September 15, 2017 from https://www.playstation.com/en-in/explore/accessories/dualshock-4-wireless-controller/.

32. Edward Steinfeld and Jordana Maisel. 2012. *Universal Design: Creating Inclusive Environments*. Wiley, Hoboken, NJ, USA.

33. Burkay Sucu and Eelke Folmer. 2013. Haptic Interface for Non-Visual Steering. In *Proc. 2013 Int. Conf. Intell. User Interfaces (IUI 2013)*. ACM Press, New York, New York, USA, 427–434. DOI: http://dx.doi.org/10.1145/2449396.2449451

34. Burkay Sucu and Eelke Folmer. 2014. The Blind Driver Challenge: Steering using Haptic Cues. In *Proc. 16th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2014)*. ACM Press, New York, New York, USA, 3–10. DOI: http://dx.doi.org/10.1145/2661334.2661357

35. T V Tran, T Letowski, and K S Abouchacra. 2000. Evaluation of acoustic beacon characteristics for navigation tasks. *Ergonomics* 43, 6 (2000), 807–827. DOI:http://dx.doi.org/10.1080/001401300404760

36. Unity Technologies. 2015. Car Tutorial (Unity 3.x only). (5 May 2015). Retrieved September 16, 2017 from http://u3d.as/1qU.

37. Unity Technologies. 2016. Unity Download Archive. (2016). Retrieved September 14, 2017 from https://unity3d.com/get-unity/download/archive.

38. Unity Technologies. 2017. Audio Spatializer SDK. (2017). Retrieved September 15, 2017 from https://docs.unity3d.com/2017.2/Documentation/Manual/AudioSpatializerSDK.html.

39. Bruce N. Walker and Jeff. Lindsay. 2004. Auditory navigation performance is affected by waypoint capture radius. In *Proc. Int. Conf. Audit. Disp. (ICAD 2004)*. Georgia Inst. Tech., Sydney, NSW, AU, 6–9. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.97.9443

40. Bruce N. Walker and Jeffrey Lindsay. 2006. Navigation Performance With a Virtual Auditory Display: Effects of Beacon Sound, Capture Radius, and Practice. *Hum. Factors* 48, 2 (June 2006), 265–278. DOI: http://dx.doi.org/10.1518/001872006777724507

41. WBDG Accessible Committee, Jordana L. Maisel, and Molly Ranahan. 2017. Beyond Accessibility to Universal Design. (15 February 2017). Retrieved August 22, 2017 from https://www.wbdg.org/design-objectives/accessible/beyond-accessibility-universal-design/.

42. T Westin. 2004. Game accessibility case study: Terraformers - A real-time 3d graphic game.. In *Proc. Fifth Int. Conf. Disabil. Virtual Real. Assoc. Technol. (ICDVRAT 2004)*. University of Reading, Reading, UK, 95–100. DOI:http://dx.doi.org/10.1.1.103.8041

43. Jeff Wilson, Bruce N. Walker, Jeffrey Lindsay, Craig Cambias, and Frank Dellaert. 2007. SWAN: System for wearable audio navigation. In *Proc. IEEE Int. Symp. Wearable Comput. (ISWC 2007)*. IEEE Press, Boston, MA, USA, 91–98. DOI: http://dx.doi.org/10.1109/ISWC.2007.4373786

44. Bei Yuan and Eelke Folmer. 2008. Blind Hero: Enabling Guitar Hero for the Visually Impaired. In *Proc. 10th Int. ACM SIGACCESS Conf. Comput. Access. (ASSETS 2008)*. ACM Press, New York, New York, USA, 169–176. DOI: http://dx.doi.org/10.1145/1414471.1414503

45. Bei Yuan, Eelke Folmer, and Frederick C. Harris. 2011. Game accessibility: A survey. *Univers. Access Inf. Soc.* 10, 1 (2011), 81–100. DOI: http://dx.doi.org/10.1007/s10209-010-0189-5