Enhancing Orientation and Mobility Skills for Persons with Visual Impairments through VR Games: Introducing the EdKit Unity Package

José M. L. de Moraes Júnior[1][0009−0008−3635−9759], Agebson Rocha Façanha[2][0000−0002−3615−2765], Bruno C. da Silva[1,2][0000−0002−4479−6042], Windson Viana[1][0000−0002−8627−0823], and Joel dos Santos[3][0000−0001−7234−613X]

[1] Federal University of Ceará, Brazil
martoniojunior@alu.ufc.br,windson@virtual.ufc.br
[2] Federal Institute of Education, Science and Technology of Ceará
agebson@ifce.edu.br
[3] Federal Center for Technological Education Celso Suckow da Fonseca (Cefet/RJ)
jsantos@eic.cefet-rj.br

Abstract. The integration of digital technologies, particularly Virtual Reality (VR) games, into Orientation and Mobility (OM) training, has shown significant potential in enhancing the rehabilitation process for People with Visual Impairments (PVI). These technologies offer immersive and interactive experiences that can foster greater autonomy and independence. Despite the potential benefits, VR games specifically designed for OM training are limited. Tools and code libraries that facilitate the development of such games are also scarce. Addressing this gap, this work presents 'EdKit', a custom Unity package that combines gestures, multimodal feedback, and in-game analytics to help integrate Left-Right Discrimination concepts into VR games for PVI. This package also supports OM analytics, enhancing the educational experience for both students and instructors.

Keywords: Orientation and Mobility · Virtual Reality · VR · Gestures · Multisensory Feedback.

1 Introduction

According to the "World Report on Vision" publication from the World Health Organization, there are 2.2 billion people worldwide with some kind of visual impairment [18]. People with Visual Impairments (PVI) have vision functions affected that could limit their interaction with the world and day-to-day activities. PVI could have difficulties determining their location in the environment they're in, as well as spatial relations between themselves and people, objects, and other reference points [11]. Orientation and Mobility (OM) is a field of training that focuses on enabling PVI to navigate their environment safely and independently, aiding in cognitive, motor, and emotional development [18, 10].

Developing OM concepts with PVI allows them to use other senses and residual vision to train their skills related to space perception [13], analyzing tactile and audio cues to perform spatial tasks such as navigation, route planning, and localization [22]. By developing OM skills, we allow individuals to understand and interact with the world with more accuracy and confidence. OM skills also allows them to build a mental representation of the spaces they're in or will navigate on it [1]. One such concept is Left-Right Discrimination (LRD), which is the propensity that humans have of defining a preference to use one side of the body more than the other through the development of neural, psychological, and motor skills [14]. Some activities involve the identification of body parts, learning to use the limbs as reference points, understanding how to use their smaller body parts such as fingers, and executing crossed commands (e.g. "Put your right hand on your left shoulder."). Other example is discerning left and right using their body as a reference point (egocentric) or projecting their orientation system into external references (allocentric) [24, 14, 7, 6].

Assistive technologies make it possible to improve spatial cognition through artifacts designed for activities that develop more than one spatial skill at once. Combined with technologies such as Virtual Reality (VR), it's possible to create engaging experiences that encourage the practice of OM exercises in an immersive and playful environment that is fully controllable. VR gaming can effectively substitute real-world games for cognitive tasks within a limited spatial environment, yielding comparable performance and user experience metrics in both settings [2]. It allows the player to explore new situations without the dangers present in the real world. Researchers had shown OM skills acquired can be transferred between virtual and real environments [22, 12, 5].

As such, integrating VR games for PVI in a OM practice model allows for more immersive experiences in virtual environments. When combined with concepts of game analytics, they offer a new tool for OM instructors to analyze the student's performance while playing the game. OM instructors could plan future learning strategies, either in a physical or remote practice model, with the latter being a opportunity space for new research interventions [4]. It is crucial to create and make available libraries and frameworks that aid developers in creating accessible VR games for PVI. In this context, this paper presents an in-progress work of "EdKit", a custom Unity package that works as an abstraction layer that unifies player gestures, feedback mechanisms (e.g. sound, haptics), and game analytics that can be used in VR games for PVI.

2 Related Works

In the literature, we find some examples of research that have designed and developed VR games for PVI [8]. However, few immersive solutions focus on the development of left-right discrimination in these applications [20, 25, 16, 9].

When talking about VR games for PVI, some studies have designed and implemented solutions for the audience [8]. However, even when expanding the definition to include virtual and hybrid environments, as well as mixed-reality solutions, only a few have made explicit considerations about the development of left-right discrimination skills in their solutions [20, 25, 16, 9].

Virtual Showdown [25], for instance, introduces a digital adaptation of Showdown, a sport similar to air hockey where players using blindfolds must hit the ball into their opponent's goal. The game was implemented as a hybrid environment solution, combining Kinect v2, Nintendo Switch Joy-Con, cardboard, and a table to define the physical and virtual space of a Showdown match to make the game more playable while staying recognizable. The game employs Kinect to track the player's position in the table. To help with inferring depth, the game uses panning and audio spatialization to indicate to players where the ball is coming from. Also, the game adopts multi-modal feedback through Joy-Con, both for when the player hits the ball, but also to lead the player's hands back to the table in case they drift out of it. The authors also note how participants obtained better scores when the ball came from the same direction as their dominant hand or from a neutral location such as the center of the table.

Following this hybrid environment approach, Kim et al. [9] present a prototype for adapting badminton to be playable between people with and without visual impairments. The prototype uses rackets augmented with a wireless sensor and substitutes the physical shuttlecock with a virtual one, represented through localized audio output that players must interpret and swing accordingly. Apple Swipe [16], in other hand, offers a game for mobile devices based on the game Fruit Ninja, where the goal is to slice the fruits and avoid obstacles to get the highest score possible. Each object has a specific sound that is played at the position they appear on the screen. Thus, it offers the possibility to train left-right discrimination. The work highlights the importance of using existing conventions to provide a better gameplay experience for PVI. It includes how the device is held, interaction mechanisms, interface navigation, and using text-to-speech.

The solutions presented here are unavailable both in binary and source code. This reinforces the opportunity and importance of making custom packages available to help developers create games working with the concepts they present. To the best of our knowledge, this is the first work designed as an open-source Unity package that offers a tool for developers to embed left-right discrimination concepts into their applications. As such, our proposal follows an approach similar to the work of Di Maria et al.[3], but applying it to OM and LRD field. In other words, this works attempts to create an open-source solution that acts as a starting point for creating games for OM and LRD. We designed it to be modular, reusable and able to be extended by other developers, which could create new poses and gestures, implement their own affordances or expand the analytics system. With this, we hope the package can be used in any game made in Unity that works with OM and LRD concepts in an explicit or implicit manner.

3 Our proposal

After the literature review, we began our research by interviewing one orientation and mobility instructor and three specialists in the development of OM applications in VR and based on the requirements gathered in this process, we designed Edkit, which was created as a traditional Unity package. The choice of game engine was made based on solutions identified in Section 2.

The system architecture is defined by three main components:

- **Gesture Model**: defines how the system determines the position of objects and how to identify a gesture from the player's position and use it.
- **Feedback**: defines representations for an affordance in the game and how it can be applied to generate multimodal feedback.
- **Analytics**: keeps track of the events that happened in the game and stores them in logs available to OM instructors.

These modules were chosen through a domain analysis process of the solutions listed in Section 2, with the aim of offering a complete package for LRD games for OM with the least amount of elements required. Figure 1 shows an initial design for how EdKit would be applied in a game. Game developers can use the entire package or just parts of it in other related contexts, such as games for PVI that are not designed for VR in mind. It allows developers to focus on other development features, such as gameplay or accessibility features.

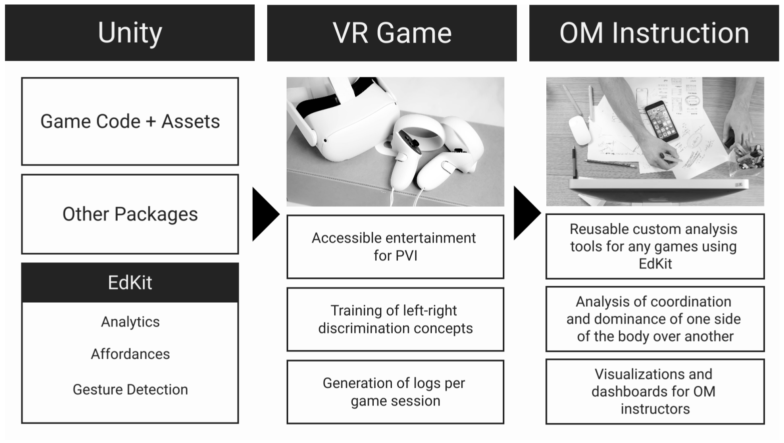
Unity
Game Code + Assets
Other Packages
EdKit
Analytics
Affordances
Gesture Detection
VR Game
Accessible entertainment for PVI
Training of left-right discrimination concepts
Generation of logs per game session
OM Instruction
Reusable custom analysis tools for any games using EdKit
Analysis of coordination and dominance of one side of the body over another
Visualizations and dashboards for OM instructors

Fig. 1. EdKit - Basic Architecture. Source: The authors.

3.1 Gesture Model

For the gesture model, we encapsulate spatial references in order to determine relationships in a scene. First, we created Axis enumerators to describe the delta in a spatial relationship between two objects (such as the distance vector from A to B) in a discrete format: inspired by Kim et al.[9] and Wedoff et al.[25], the axis is separated into 3 regions as described in table 1, with an extra case for a

Axis	Negative Value Outside Deadzone	Value Inside Deadzone	Positive Value Outside Deadzone
Axis.X	Left	Center	Right
Axis.Y	Below	Neutral	Above
Axis.Z	Back	Body	Front

Table 1. Specification for Axis enumerators. Source: The authors.

Null Object in inconclusive or invalid comparison cases (e.g. position of objects in a 2D space does not use the Z axis).

Orientation is a structure containing the relative position and rotation of an object in relation to another in a determined moment in time. Together with Unity's Bounds structure, we create a normalized representation of the relative position or rotation, offering a description for the relative position. The type can be used as a reference point to place other Transform components as new Orientations in relation to itself. With 3 Orientations, one can define a Placement structure for any game actor (e.g. players, enemies) and identify poses.

Poses are an interpretation of Placement that uses a "Bring Your Own Algorithm" approach, where a programmer defines the detection algorithm for each Orientation structure and returns a score to define the accuracy for a pose. This approach was chosen to allow more flexibility for implementation, in which you can choose how you want to evaluate the pose, be it via a common algorithm, using an ML model or using a custom-made algorithm.

The scoring function of a pose should return a positive value when there's some resemblance to the actual pose in one or more orientations and a negative value or zero when the pose does not match at all. To create a pose, script a new instance of the Pose structure or create a new PoseData object, defining reference Bounding Boxes to determine the pose's score. The score ranges from negative (outside the box) to between zero (edge of the box) and one (center of the box). Pose Events store the actor's placement, the identified pose, and the timestamp when it occurred.

Gestures are an interpretation of a sequence of pose events, returning a score based on the poses' accuracy. Just like Poses, they can be created via scripting or using a GestureData object. It analyzes a sequence of Poses, Gesture Events storing the list of Pose Events, the Gesture identified, and the timestamp. *Gestures* were conceived as a way to create gestures by reusing the implementation from pose detection and composing together already existing poses.

Tracker is a component that is responsible for converting referenced Transforms[4] in a scene into a Placement structure. As the Tracker deals with the information stored in the Transform and not directly with the input of the motion devices, developers can create custom components to control the Game Object directly or indirectly, which can be useful for Non-Playable Character (NPC) or for improving the quality of tracking when using imprecise motion devices. The component uses 4 reference Transforms related to an actor to perform its

[4] Transform: represents position, rotation and scale of a GameObject in Unity[21].

operations: head (or the actor's body), left hand (or equivalent), right hand (or equivalent) and the actor's respective origin (e.g. XR Origin).

The conversion process happens through a coordinate system change for the actor's positions and rotations to a new one, which has it's origin composed by the camera's orientation related to the XZ plane and the origin's up vector, defining an egocentric reference to an actor. Since Tracker is based on the player's orientation in the scene, this allows PVI to perform gestures even if they drift away during a play session without worrying about repositioning themselves in the physical space. One of these components is a Recognizer: a component that handles gesture detection by receiving Placements and, through a pose buffer and a evaluation process, recognize poses and gestures and fire events it's respective events. While the component was made to be wired together to the Tracker's event, both components can be used standalone.

The poses and gestures evaluated in this component are managed by a custom data structure called GestureEvaluator. This structure registers gestures and their associated poses. While a Recognizer only includes methods based on ScriptableObject to meet serialization needs for UnityEvents, the component includes a public GestureEvaluator property. This allows developers to register gestures and poses through scripting. Figure 2 illustrates how a player's position in a VR space can be interpreted as a gesture.

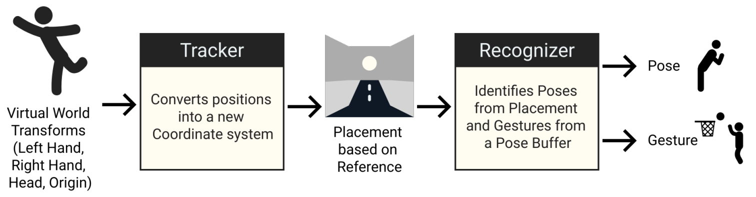
Virtual World Transforms (Left Hand, Right Hand, Head, Origin)
Tracker
Converts positions into a new Coordinate system
Placement based on Reference
Recognizer
Identifies Poses from Placement and Gestures from a Pose Buffer
Pose
Gesture

Fig. 2. EdKit - Gesture Recognition. Source: The authors.

With poses and gestures, developers can create VR games that allow players to perform actions without pressing buttons. This system offers flexibility in selecting and using gestures and supports effective unit testing. Although the Tracker and Recognizer are designed to work together, they can also be used separately for custom recognition or tracking systems.

3.2 Feedback

For the feedback system, we have a structure inspired by the [16, 25, 9] in which developers use Affordances to represent the game's feedback. In general, they're a depiction of multimodal feedback that is applied simultaneously through distinct channels. In EdKit, Affordances can have up to three data representations: **Sound**, which can be an AudioClip, an AudioMixer or an Audio effect to be played; **Text**, which can be a normal String type, a localized component or a

String that's encapsulated into a contextual type (e.g. TableEntry); and **Visual**, which can be a Color, a Material, a Shader, an Animation or a Mesh.

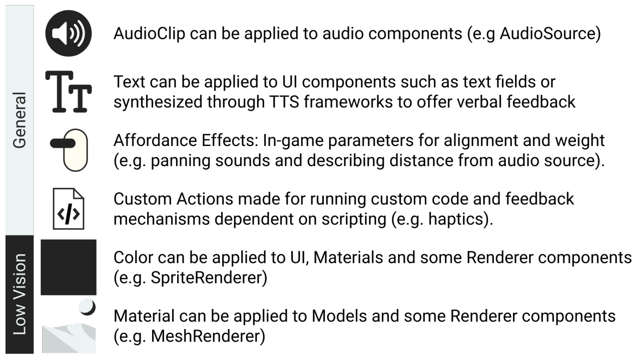
General
Low Vision
AudioClip can be applied to audio components (e.g AudioSource)
Text can be applied to UI components such as text fields or synthesized through TTS frameworks to offer verbal feedback
Affordance Effects: In-game parameters for alignment and weight (e.g. panning sounds and describing distance from audio source).
Custom Actions made for running custom code and feedback mechanisms dependent on scripting (e.g. haptics).
Color can be applied to UI, Materials and some Renderer components (e.g. SpriteRenderer)
Material can be applied to Models and some Renderer components (e.g. MeshRenderer)

Fig. 3. EdKit - Affordances. Source: The authors.

Affordances can either be executed directly in code or via an Affordance Receiver, a component that gets the representations of an Affordance and fires events for each representation. In both, developers pass in an Affordance Effect to control how it gets executed, based on these parameters: **Alignment**, three-dimensional normalized vector which describes a spatial relation; **Weight**, three-dimensional normalized vector to describe the impact of the effect in each axis, with values going from 0 (no impact) to 1 (max impact); **Duration**, floating number describing how long should the Affordance last for in seconds. When the duration is set to -1, it keeps playing until it is replaced. When set to 0, it triggers the update method only once.

3.3 Analytics

For tracking game events to build the Analytics model, the system logging is inspired by the PROV-O ontology [23][15], taking advantage of serialization to store entities, agents and activities, together with interfaces to transform any data into provenance structures. We designed it to log OM metrics such as position of the player's hands, time spent to execute an activity, whether the player moved more it's left or right hand or the accuracy of interacting with objects coming from your left or your right. Everything is stored in a Session structure, together with information such as date and time when the game session was played. As each game has it's own set of data types and structures that can be logged as attributes, it's best to provide a solution that's both well structured (logs are stored in sessions using the PROV-O ontology) and flexible (attributes can use any serializable type).

The lifecycle of a Session works as described in Figure 4: To start, the developer opens a session passing identifiers for the user and scene. Then, events are

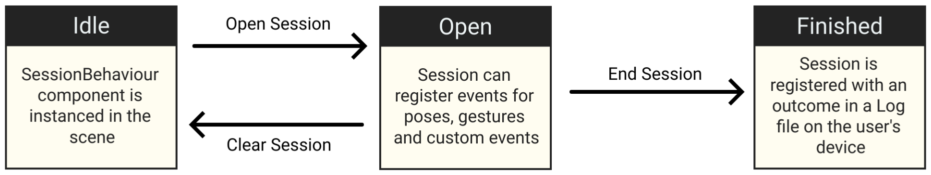
Idle
SessionBehaviour component is instanced in the scene
Open Session
Clear Session
Open
Session can register events for poses, gestures and custom events
End Session
Finished
Session is registered with an outcome in a Log file on the user's device

Fig. 4. EdKit - Analytics Session. Source: The authors.

registered using any of the methods declared. When the session ends, the developers closes the session with the session's outcome, which then gets saved into a log file that's stored on the player's device and can be extracted and interpreted by custom tools, such as dashboard visualizations [17].

4 Final Considerations

In this work, we introduced EdKit, a custom Unity package for developing VR games for PVI. Our package provides a basic API for gesture detection based on the player's reference points, along with a structure for multimodal feedback and an analytics model for games with in-game events stored in a log that can be read by custom tools. EdKit is available as an open-source package on Github[5].

As part of this work, we plan to: (a) apply the package in "Ed", a VR game prototype to teach left-right discrimination concepts through motion gestures that are accessible for PVI to use, (b) improve the package's experience by creating custom editors, (c) improve the API of the package to be more robust and (d) develop more documentation resources for the package.

The project described in (a) is part of a smaller set of proof-of-concepts that are being developed with the goal of testing and refining the package's API and determining how useful it is in the development process. We also plan to evaluate the library with game developers with an approach inspired by Piccioni et al.[19]. For (b), we plan to use UnityEditor's API to create custom Property Drawers and Inspectors for the types already implemented to allow fine-tuning parameters directly from the Unity Editor and have a better experience in doing so. Due to limitations related to how the Unity Editor works, not all types implemented in EdKit will be supported in it, which limits the use of the package by non-programmers and requires creating custom authoring mechanisms.

Finally, (c) and (d) are part of the instantiation process of the package, as more solutions such as (a) are developed and requirements and limitations arise to make EdKit easier to use, as well as offering more utility API for faster implementation.

Acknowledgments. This research was partially funded by CNPQ, under grant number 409184/2021-7.

[5] Source code at: https://github.com/MartonioJunior/EdKit

References

1. Andrade, R., Waycott, J., Baker, S., Vetere, F.: Echolocation as a means for people with visual impairment (PVI) to acquire spatial knowledge of virtual space. ACM Transactions on Accessible Computing (TACCESS) **14**(1), 1–25 (2021)
2. Choueib, A., Berkman, M.İ.: Comparing Performance and Experience in VR vs. Real-World Through a Puzzle Game. In: International Conference on Videogame Sciences and Arts. pp. 72–85. Springer Nature Switzerland (2023)
3. Di Maria, G., Stacchio, L., Marfia, G.: Unity-vrlines: Towards a modular extended reality unity flight simulator. In: International Conference on Entertainment Computing. pp. 241–250. Springer (2023)
4. Dove, G., Fernando, A., Hertz, K., Kim, J., Rizzo, J.R., Seiple, W.H., Nov, O.: Digital Technologies in Orientation and Mobility Instruction for People Who are Blind or Have Low Vision. Proceedings of the ACM on Human-Computer Interaction **6**(CSCW2), 1–25 (2022)
5. Façanha, A.R., Darin, T., Viana, W., Sánchez, J.: O&M indoor virtual environments for people who are blind: A systematic literature review. ACM Transactions on Accessible Computing (TACCESS) **13**(2), 1–42 (2020)
6. Gonçalves, D., Piçarra, M., Pais, P., Guerreiro, J., Rodrigues, A.: "My Zelda Cane": Strategies Used by Blind Players to Play Visual-Centric Digital Games. In: Proceedings of the 2023 CHI conference on human factors in computing systems. pp. 1–15 (2023)
7. Hoogsteen, K.M., Szpiro, S., Kreiman, G., Peli, E.: Beyond the cane: describing urban scenes to blind people for mobility tasks. ACM Transactions on Accessible Computing (TACCESS) **15**(3), 1–29 (2022)
8. Karaosmanoglu, S., Kruse, L., Rings, S., Steinicke, F.: Canoe vr: An immersive exergame to support cognitive and physical exercises of older adults. In: CHI Conference on Human Factors in Computing Systems Extended Abstracts. pp. 1–7 (2022)
9. Kim, S., Lee, K.p., Nam, T.J.: Sonic-badminton: audio-augmented badminton game for blind people. In: Proceedings of the 2016 CHI Conference extended abstracts on human factors in computing systems. pp. 1922–1929 (2016)
10. Kreimeier, J., Götzelmann, T.: Two Decades of Touchable and Walkable Virtual Reality for Blind and Visually Impaired People: A High-Level Taxonomy. Multimodal Technologies and Interaction **4**(4), 79 (2020)
11. Lahav, O.: Virtual reality systems as an orientation aid for people who are blind to acquire new spatial information. Sensors **22**(4), 1307 (2022)
12. Lahav, O., Schloerb, D.W., Srinivasan, M.A.: Newly blind persons using virtual environment system in a traditional orientation and mobility rehabilitation program: a case study. Disability and Rehabilitation: Assistive Technology **7**(5), 420–435 (2012)
13. Loomis, J.M., Klatzky, R.L., Giudice, N.A.: Representing 3d space in working memory: Spatial images from vision, hearing, touch, and language. Multisensory imagery pp. 131–155 (2013)
14. Lussac, R.M.P.: Considerações psicomotoras sobre a lateralidade e respectivos apontamentos acerca da capoeira. Caminhos da educação: diálogos culturas e diversidades **4**(1), 01–13 (2022)
15. Marques, F., Lignani, L., Quadros, J., et al.: Probee: A provenance-based design for an educational game analytics model. Tech Know Learn (2024). https://doi.org/10.1007/s10758-024-09758-x

16. Mohd Norowi, N., Azman, H., Wahiza Abdul Wahat, N.: Apple Swipe: A Mobile game Apps for Visually Impaired Users Using Binaural Sounds. In: Proceedings of the Asian CHI Symposium 2021. pp. 196–201 (2021)
17. Moraes, J.M., Do Carmo F, P., Araújo, M.d.C.C., Costa, B., Amorim, M., Façanha, A.R., dos Santos, J., Viana, W.: Dashboards to support rehabilitation in orientation and mobility for people who are blind. In: Proceedings of the 29th Brazilian Symposium on Multimedia and the Web. pp. 6–10 (2023)
18. Organization, W.H.: World report on vision. World Health Organization (2019)
19. Piccioni, M., Furia, C.A., Meyer, B.: An empirical study of api usability. In: 2013 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement. pp. 5–14. IEEE (2013)
20. Simões, D., Cavaco, S.: An orientation game with 3d spatialized audio for visually impaired children. In: Proceedings of the 11th Conference on Advances in Computer Entertainment Technology. pp. 1–4 (2014)
21. Technologies, U.: Unity - manual: Unity user manual 2022.3 (lts), https://docs.unity3d.com/Manual/index.html, (Last Accessed: 9 June 2024)
22. Thevin, L., Briant, C., Brock, A.M.: X-road: virtual reality glasses for orientation and mobility training of people with visual impairments. ACM Transactions on Accessible Computing (TACCESS) **13**(2), 1–47 (2020)
23. (W3C), W.W.W.C.: PROV-O: The PROV Ontology, https://www.w3.org/TR/prov-o/, (Last Accessed: 27 May 2024)
24. Watemberg, J., Cermak, S., Henderson, A.: Right-left discrimination in blind and sighted children. Physical & Occupational Therapy in Pediatrics **6**(1), 7–19 (1986)
25. Wedoff, R., Ball, L., Wang, A., Khoo, Y.X., Lieberman, L., Rector, K.: Virtual showdown: An accessible virtual reality game with scaffolds for youth with visual impairments. In: Proceedings of the 2019 CHI conference on human factors in computing systems. pp. 1–15 (2019)