“Lend Me a Hand” – Extending the Reach of Seated VR Players in Unmodified Games Through Remote Co-Piloting

Felix J. Thiel[*]
University College London

Anthony Steed[†]
University College London

Abstract

Playing virtual reality (VR) games while seated can be challenging because many games are designed for a mobile, standing player. Sitting down not only limits mobility but also the reach of the player. This is particularly problematic if the player does not have a choice in whether they play seated or standing because it then becomes an accessibility issue. We have taken the concept of co-piloting (controls shared across multiple input devices and players) and applied it to VR with the aim of extending the reach of the VR player. To demonstrate the feasibility of the concept, a software prototype was developed that allowed a second, remote, player to take over the VR player’s controllers and help them reach out. Because the prototype manipulates the SteamVR middleware and not the game, it can be used with any application that runs on SteamVR. A small formative study was conducted to inform further research and proof the concept.

Index Terms: Human-centered computing—Accessibility—Accessibility system and tools; Human-centered computing—Collaborative and social computing—Collaborative and social computing theory, concepts and paradigms—Computer supported cooperative work

1 Introduction

With the emergence of 6-Degree-of-Freedom (6-DoF) tracking of head and hands in consumer virtual reality (VR) devices, games in which the player can move freely in their tracked space have been very popular. While this might certainly increase the engagement of players, there are many reasons why a player might not be able to stand up such as temporary injuries, mobility impairments (i.e. from disability or injury), lack of fitness, fatigue, or simply a desire for comfort.

While a new problem for VR designers, issues of accessibility have been widely explored in other media. Operating systems and TVs are both not designed for all kinds of impairments but both provide tools that make them more accessible to users with specific conditions. Examples of this are high contrast display modes for visually impaired computer users or audio descriptions for deaf viewers. In VR, however, very little in terms of guidelines and tools exist that a seated player could use to make their existing games accessible to them and create a comparable experience to standing players.

We propose an accessibility tool that can move the player’s virtual hand to give them the reach they cannot achieve physically. This has been attempted before with techniques such as Go-Go [13], HOMER [3,4], and RNL [16], but all of these come with additional controls that could interfere with the game’s controls or are not themselves usable with by some types of seated player (e.g. with variable control over movement). To complement these efforts, we propose the concept of *co-piloting* where the controls are split between the player and a second person. Without altering the game’s design, the second player can then assist the VR player remotely.

2 Related Work

While the concept of co-piloting in VR as such has not been of much focus in research, co-piloting is a common assistive tool for game consoles. We thus give a broad overview of accessibility in games and VR, before turning to prior work on co-piloting and reach extension.

2.1 Accessibility and Video Games

With a survey from 2018, Beeston et al. [2] tried to capture an image of the population of disabled players. In cooperation with the organisation “Able Gamers”, they sent out a questionnaire and received 154 responses. They found that their participants do not differ much from able players in terms of play times and preferences.

Dalgleish [5] reviews accessibility issues in regular video games for players with motor impairments. He concludes that the dimensionality of the controls required by games has increased dramatically and with it the demands on the player and their ability to manipulate the controllers. Systems such as the Microsoft Kinect resist his classification due to their highly variable and context-dependent input. He states that most of the accessibility efforts haven been limited to control re-mappings and hardware modifications. While they have their merits, he points out that remapping may be insufficient alone and that hardware modifications are often expensive. As an alternative, he proposes approaching the issue through game design and suggests incorporating asymmetric designs and controls to embrace the diversity of the players. The puzzle game *Keep Talking and Nobody Explodes* [14] is mentioned by the author as one example of such a design.

2.2 Accessibility and Virtual Reality

Mott et al. [11] explored the accessibility of VR for people with limited mobility. While their focus is mostly on hardware related issues such as weight, button placement, and cord management, they also advocate for alternative customisable input methods and a design concept that they call “Interdependence”. They found that a common strategy of impaired players was to have someone else help them with the hardware. However, they also found that the hardware design does not acknowledge or facilitate that and simple tasks such as putting on the headset become more difficult with help than without. While their use case is focused on hardware, co-piloting also expresses interdependence because it facilitates and highlights the collaboration between the player in VR and their helper.

Gerling et al. [7] did a survey of 25 wheelchair users to gather insights on their motives and experiences when engaging in VR. They derived design implications that they used to implement three VR games which were evaluated with 18 participants. The author’s three design implications were flexible control schemes, considering the wheelchair when designing interaction, and avoiding mandatory representation of disability. The design implication of a flexible and adaptable control scheme is not only important because of different preferences of the players, but also because the player’s impairments could also affect other parts of their body as well. Tools such as

[*] e-mail: felix.thiel.18@ucl.ac.uk

[†] e-mail: a.steed@ucl.ac.uk

our remote co-piloting could be part of a larger tool-set that allows players to fit their controls to their needs.

Zhao et al. developed SeeingVR [17] to address the issue that current VR applications do not support people with low vision (i.e. not blind, but also not correctable by glasses). It is a set of 14 tools that provides visual and audio augmentations to enhance a VR application for people with low vision. The players can select, adjust, and combine these tools according to their needs. A majority of them works with existing VR applications *post hoc*, while the remaining require the developer to include a Unity toolkit in their application. While they address a different accessibility issue of VR than our project, there are similarities. In particular, both are meant as accessibility tools to be used on top of existing VR games and at the user's demand. Another commonality that our concept shares with a part of SeeingVR is that it can work without any support from the original developer.

2.3 Co-Piloting

The idea of splitting controls of a video game across different input devices to make them more accessible is not a novel one. Recent Xbox systems, including the Xbox One and the Xbox Series X, have a feature built-in that allows two controllers to act as one so that two players can take over the role of a single player with split controls. Iacopetti et al. [6, 9] followed a similar approach, with the difference that their device allowed the second means of input to be a set of accessible buttons. Their device connected to the PlayStation 2 or PlayStation 3 and advertised itself towards the console systems as a single controller while it actually connected to a regular gamepad and a set of accessible input devices. Although their commercial spin-off "Jpemulator" is now defunct, this concept was recently picked up by Microsoft with their "Xbox Adaptive Controller" (XAC). The XAC allows the user to plug in a variety of accessible input devices which the controller then maps to the input of a regular Xbox gamepad. The use of standard connectors allows for a wide range of commercial and home-built solutions to make console- and PC-gaming accessible to a wider population of users.

This concept of splitting controls has also been applied to VR: WalkinVRDriver [1] is a toolkit which offers a range of assistive tools to help disabled people interact with virtual environments. Among virtual controllers and temporary adjustments in height and reach, it also allows a second player to alter the position of the VR player by using a regular Xbox gamepad. WalkinVRDriver is a good example of how a third-party accessible toolkit for VR could look like. Unfortunately, there does not seem to exist a published evaluation on its effectiveness. Compared to our proposed approach, it comes with a few limitations. For once, it requires the second player to be at the same location as the VR player while our concept works remotely as well. Furthermore, with WalkinVRDriver, the second player can only control the positions of the full player (i.e. head and both hands) and not of the individual hands. However, individual offsets between virtual and physical hands can be set temporarily through WalkinVRDriver's user interface.

2.4 Solving the Problem of Reach in VR

A common approach to extending the players reach in VR is to amplify their physical motion before applying it to the virtual hand. This makes the virtual hand travel further than the physical and allows for grabbing things that were previously out of reach. One well-known example of this is the Go-Go technique by Poupyrev et al. [13] where the movement of the arm is amplified non-linearly after it passes a certain point. A similar approach is used by Wentzel et al. [16] who also use a non-linear mapping to extend the reach, but limit its extent to retain body ownership. Another approach is the HOMER (Hand-centred Object Manipulation Extending Ray-casting) technique and its variations by Bowman et al. [3, 4]. With HOMER, the user points a ray at an object and when they initiate the grabbing, the virtual hand will move to the remote object. By using buttons, the user can then "reel it in".

While they all extend the reach of the user, these techniques add additional controls that increase the complexity of the application and can interfere with the existing controls. They risk overwhelming or frustrating the user. HOMER also has the additional drawback that it requires knowledge about the objects in the scene and whether they can be interacted with. This information is not easily obtainable outside of the game and makes HOMER unsuitable to be used as an accessibility tool compatible with general VR applications. With our approach, the additional controls are operated by a second player, so the control complexity for the first player does not increase. It also does not interfere with the existing control scheme because it moves the virtual hand as a real human would.

3 Concept

Our prototype applies the concept of co-piloting to VR. It allows a second player, the co-pilot, to extend the reach of the pilot, the player in VR. The aim of the concept is not to provide a multi-user experience but to have two players acting as one. The concept is meant to be applied at the system level without any need to adapt the games themselves.

3.1 User Experience: Pilot

The pilot is the player inside of the virtual environment and their user experience remains mostly unchanged. However, when they encounter a situation where their reach is insufficient, they can verbally request help. This help will be provided by one of their hands "detaching" itself from the location of their physical hand to float towards the requested target. The pilot can then perform the desired action before the virtual hand returns to the location of their physical hand.

3.2 User Experience: Co-Pilot

The co-pilot is the assisting player and does not need to be in a virtual environment. They receive a video stream that shows the image displayed in the pilot's headset and use an input device such as a keyboard or an Xbox game controller. Normally, they have no control over any of the virtual hands of the pilot but through a button press, they can take over one of them. From that point onwards, they can manipulate the position of the selected hand through their input device. In our prototype, the co-pilot can use the keyboard through the keys WASD for horizontal movement and the keys QE for vertical movement, or the Xbox controller with the two analogue sticks (left for horizontal, right for vertical movement). All movements are relative to the pilot's view direction. To release the virtual hand that is currently selected, the co-pilot can either press the same button again or press the button to control the other hand. In either case, the hand that was controlled will drift back to the location of the physical hand of the pilot.

3.3 Merging Inputs

When the co-pilot controls one of the virtual hands, the pilot does not completely lose control over it. This started as a technical limitation of the implementation but was kept as it turned out that it allowed for better cooperation. Instead of moving the actual virtual object, the implementation changes the reference frame of the controller. This means that the pilot can still move the controller, but this new movement will be according to the new reference point. This allows the pilot to compensate for the inaccuracy of the co-pilot's controls. Because it is designed to extend the reach of the user, the controls for the co-pilot move the hand relatively quickly, at 1 m/s. This comes at the cost of precision.This is further complicated because the co-pilot's screen does not display stereo images and they thus have no depth perception causing more frequent over- and undershooting. However, through the pilot still retaining control, they can compensate for this imprecision and perform delicate movements themselves. As such, the co-pilot only needs to navigate the hand into a rough area around the target and the pilot can perform the actions that require more precision themselves. This was found to make the interaction both faster and less frustrating for both parties.

4 Implementation

Our prototype was developed in Unity (2019.4.10f1) with large parts written in C#, but C++ was also used for the library that handles the manipulation itself. A rough overview of the structure of our prototype can be seen in Fig. 1.

4.1 Networking

To allow the co-pilot to be remote of the pilot, a network connection over the internet between the pilot and the co-pilot needs to be established. To avoid firewall issues, we decided that the two clients would not connect directly to each other but that both would establish a connection to a server that acts as a simple relay. Once more than one TCP connection is established, the server broadcasts all incoming messages to all other connected clients. The relay server was written in Node while the client-side networking code was written in C#.

4.2 Video Streaming

For the video streaming, we started with a remote desktop application[1] for Unity by Hecomi that was published under the MIT licence. We then fitted it to our needs through exchanging the networking layer with our own implementation and adding functionality for remote control and manipulation of the SteamVR environment (see the following sections). Hecomi's implementation was chosen because it already provided much of the functionality we wanted and used the Nvidia hardware encoder (NVENC) for the video encoding. Although this would require an Nvidia card to work, we hoped that the hardware encoding would help with keeping the latency down. Indeed, we achieved latency values of less than one second with other users in the UK during prototype testing.

4.3 Remote Control

On the side of the co-pilot, the player can use either their keyboard or an Xbox controller as their input device. The input from the keyboard and the controller can be directly read through Unity interfaces and is bundled up in regular intervals and sent to the pilot's client via the network. There, the message is unpacked and the new state is passed to the C++ library that handles the manipulation of the virtual environment.

4.4 SteamVR Manipulation

All the functionality to manipulate the virtual environment is bundled into one C++ library. This library holds control over what controls are active at that moment, their behaviour, and, most importantly, communicates with the API of the "OpenVR Input Emulator"[2]. The input emulator allows us to ignore the specific games and instead manipulate the SteamVR middleware. This makes it possible to build manipulations that affect every game that uses SteamVR without the need to change them. The input emulator achieves that by hooking itself into SteamVR libraries and offers an API to other applications that is used by our manipulation library. One obvious drawback is that the input emulator might no longer function when SteamVR itself is updated. This occurred during the period we developed and ran the study and thus our application was limited to SteamVR versions 1.13.10 and below at the time of writing.

Our manipulation library determines the behaviour of the manipulation. It keeps track of which hand is being manipulated and how large the current offset is between the real and the manipulated position. When a command to release the hand is sent by the co-pilot, the manipulation library will decrease the offset automatically until both positions are the same again. This allows the hand to smoothly transition back to the pilot's control without instantaneously snapping back. Whenever an offset needs to be changed, whether it be because the co-pilot is giving commands to move it or because they gave the command to release it, the manipulation library interfaces with the API of the input emulator and changes the "world from driver offset". This offset describes the offset between the coordinate system of the tracking system and the coordinate system of the virtual environment. Usually, this offset is set on calibration and its purpose is to align the player's physical space with the virtual one. For devices that are handled by the same tracking system, these offsets are also usually the same, however, SteamVR stores one offset for each device and thus makes it possible for us to change the offsets for the devices independently. Through manipulating these offsets, the virtual hand of the pilot can be moved independently of their real one without completely depriving the pilot of their control over it.

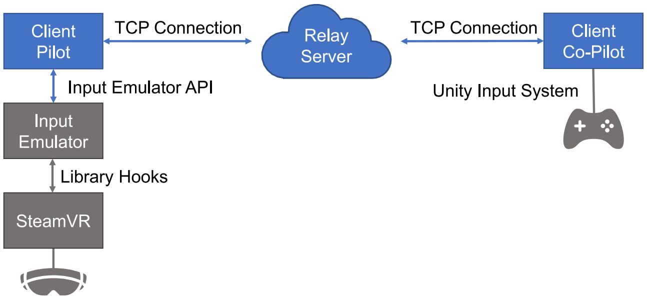

Figure 1: Diagram outlining the structure of our prototype. Blue components were developed by us. Grey components were developed by a third party.

5 Formative Study

To proof our concept and inform further research, we conducted a small formative study with four participants and the game *Job Simulator* [12]. This particular game was chosen because it is simple enough for a new player to get into quickly, offers interactions close and far in a fairly balanced ratio, and is not timed so pilot and co-pilot have time to coordinate where necessary. We want to note that this game is actually one of the more accessible VR games because it offers subtitles for players who are hard of hearing and a "Shorter Human Mode" which shrinks the environment and brings everything closer. For the sake of the experiment, these accessibility options were disabled.

5.1 Participants

Six participants were recruited for the experiment, but two dropped out due to technical issues. Of the remaining four participants, three were male and one preferred to not disclose their gender. Two were in the age bracket 22 - 25, one in 30 - 33, and one in 46 - 49. Due to current COVID-19 restrictions, it was not possible to perform this experiment in our laboratory and instead people were recruited that had a compatible VR system at home. Unfortunately, this also limited the number of available participants and three of the six were fellow researchers from our or other institutions. All participants reported having extensive experience with VR and video games, which was compatible with our goal of supporting games players in these situations. All participants were able-bodied, however, as abled and disabled players alike encounter the issue of a reduced reach when playing VR sitting down, the results should be valid for

[1] https://github.com/hecomi/UnityRemoteDesktopDuplication

[2] https://github.com/matzman666/OpenVR-InputEmulator/releases

both populations.

5.2 Procedure

Before the experiment, the experimenter and the participant would connect via a VOIP application and set up the experiment on the participant's computer. This meant installing the Input Emulator, our prototype, and the game *Job Simulator*. After the installation, the participant was given a questionnaire that captures basic demographic information such as age, gender, and previous VR experience.

The experiment itself was conducted in three passes. In each, the participant played the office level of *Job Simulator* in which they had to perform a set of tasks set in an office cubicle. As the player does not need to leave the cubicle, everything is in easy reach for a standing player but seated players will have some difficulty. The three passes differed in the posture of the player and the available assistance. In the first pass, they played the level as it was intended so they were standing upright and could move around a few steps. This was so they could familiarise themselves with the game and the environment. In the second pass, they were asked to play the game while sitting down and without any assistance. This was done so they could experience how a person forced to sit would experience the game and make them aware of the difficulties that come with a seated position. In the third and final pass, they were assisted by the experimenter who assumed the role of the co-pilot. The participant could then ask the experimenter at any time to help them reach for something out of their range. This pass was to introduce the participant to our application and get feedback on its usefulness.

Each of the three passes was followed by a short break out of VR with a semi-structured interview. The experimenter would ask the participants to comment on their experience with a focus on any difficulties and frustrating moments that they may have encountered. In the third and last interview, a few additional questions were asked about their thoughts on the remote co-piloting.

After the experiment, the experimenter and the participant would remove the experiment software and the details for the £50 compensation in form of an Amazon voucher would be arranged.

6 Results

6.1 Frequency of Use and Purposes

Three of the four participants made frequent use of the assistance, while the last only used it once. The average number of uses was 10.25 with a maximum of 21 requests and a minimum of 1. The most frequent actions for which they requested assistance for was to reach for the little printer attached to the cubicle that triggers the next task in the level (see Fig. 2) and when the player had to hand a candy bar to a co-worker in the neighbouring cubicle (see Fig. 3). These two actions are the ones furthest away from the player in that level. Another commonality between the participants was that they never used the assistance when they could do it themselves. Often, they would try and then ask for assistance when their range was insufficient. One participant, in particular, stated later in the interview that they also based their decision on whether they thought it would be faster if they do it themselves.

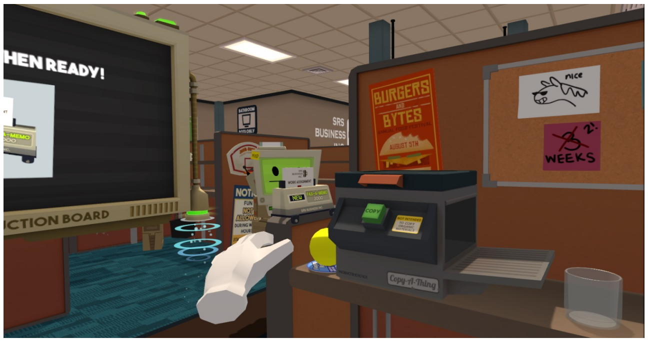

Figure 2: View at the printer from the perspective of a seated player.

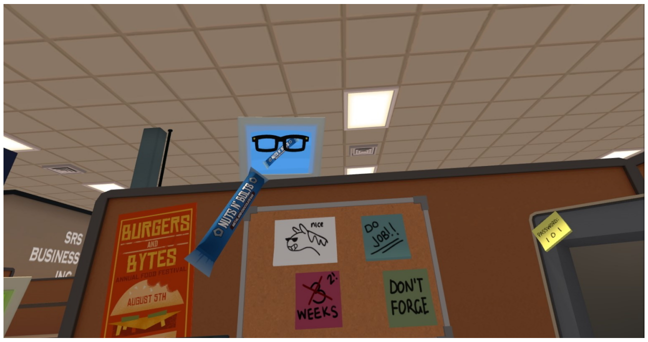

Figure 3: View at the co-worker from the perspective of a seated player.

6.2 User Engagement and Communication

One aspect that we were interested in was how the participants would engage with the assistance and their co-pilot and what the communication would look like. In testing of an early prototype of co-piloting, where the co-pilot would control the player's position in the game *Space Pirate Trainer*, the majority of the players did not engage with each other at all because they perceived shooting the robots and moving to avoid getting hit by laser rays as two separate tasks that required little coordination. In this experiment, however, the pilot needs to let the co-pilot know what they want to have help with for the assistance to be effective. Most of the participants phrased their need for help as a question along the lines of “Could you...” or “Can you ...”. However, one of the participants seemed to perceive the assistance as more of a supernatural power and even came up with a catchphrase, saying “Magic Hand!” nearly every time they wanted assistance while pointing at the object of interest.

6.3 Interview Results

None of the participants reported serious symptoms of cybersickness at any stage of the experiment though one participant reported feeling light eye strain after the second and third pass.

Most participants recognised and acknowledged the issue of a reduced reach when sitting down after the second pass. The reported issues were reaching for objects far away, turning on a chair to reach objects behind, and picking up objects from the floor. While they rarely perceived the actions as impossible, they found them more difficult and in one case more exhausting. They also reported visual issues, with the instructions and non-player character being obstructed by level geometry when sitting down. Despite the reported difficulties, only a few said they were really frustrated. However, some admitted that they would have been if the issues would have occurred more often. As the source of frustration, they gave the inability to pick things up. One participant clarified that the particular source of frustration would be the game expecting them to perform the action while not giving them the means to do it. Except for one, all found their enjoyment impaired when compared to the first pass.

The third pass generally improved the experience. The participants reported fewer difficulties and less frustration and commented occasionally that the assistance allowed them to perform actions that were difficult or impossible in the previous pass.

When asked whether they found the assistance useful, all participants stated that they found it particularly useful when reaching for objects that they should not have been able to reach from their seated position. One also further commented that they found the knowledge alone that they could ask for help nice already.

On the question of whether they felt the interaction as cooperation or a take-over of the co-pilot, the participants gave very mixed feedback. One found it cooperative, but not as it would usually be in games where there are two entities with individual goals. Another said they perceived it as cooperative, but would still stop interacting until the co-pilot completed their interaction. One participant felt that the experimenter took over, and one felt it was cooperative because the hand would only move when they were asking for it.

When asked for ideas for further improvements, the participants asked for a higher movement speed of the hand so it could reach its target quicker and support for finer movements so it could help with tasks that require more precision. These two suggestions are to some degree contradictory and will likely require the co-pilot to be able to adjust either their control sensitivity or speed.

Two of the participants also had additional comments regarding the experiment. One of them commented that they think they would use the system a lot more with somebody that knows what they mean. They said that with a voice activation system, people would be used to it not working correctly and only use it when they have to. They also think that people would get used to remote co-piloting quickly if either the game or one of the players require it. A second participant wondered whether they should be surprised that the independent movement of the hand did not cause any breaks in embodiment or presence for them. They thought it might be about them being the one that makes the conscious decision when and where it moves. They were also not sure how they would respond to an automated system, because they were using a lot of contextual commands such as "Could you do it again, please" which requires context that is easier to understand for a person.

7 Discussion

7.1 Human Assistance vs. Automated Assistance

Without prompting by the experimenter, three of the four participants seemed to assume that human co-pilot would be replaced later on in the development. One assumed that it would be additional controls for the pilot and two assumed that an automated system would take over on command. One participant explicitly mentioned voice activation as a mechanism to request help. Indeed, there are concerns about an approach that requires two human players. For example, if the pilot is really dependent on the help, they cannot play when the second player is not available or unsuited for the role. This was reflected in the feedback from the participants as well, with one stating that the usefulness of the technique depends very much on how willing and engaged the co-pilot is. However, a second human player also has major benefits. Both of the two participants that assumed that the co-pilot would later be automated brought up concerns about the effectiveness of such a system. One mentioned that they think that people would use the system with voice activation way less than people with a familiar human player because a human would be quicker in understanding what they want. The second participant made a similar statement, saying that they were using a lot of commands such as "Could you do it again, please" that were dependent on what happened in the past and the game's context. This contextual behaviour was also seen with other participants with one simply pointing in the general direction of the object of desire and shouting "Magic Hand!". A human co-pilot that sees the pilot's view and knows the game can react to this much faster than a simple automated system. Developing an artificial co-pilot that is aware of the game's context is likely to be a very hard problem, especially when the goal is to use it independent of specific games.

7.2 Playfulness and Enjoyment

While playfulness and enjoyment appear to be minor aspects, they are still important because accessible games still need to be fun. All participants claimed to enjoy the experience, though this was a bit dampened in the second pass due to their limitation in reach and mobility. Apart from the self-reported enjoyment, which could be affected by acquiescence bias, we could also observe them in their actions. With the assistance, the participants enjoyed the game as they would in the first pass, messing up the virtual workplace, and pursuing actions that did not progress them into the game (such as photocopying a wall clock thirty times). They even involved their co-pilot in these actions, for example by asking them to help place an object on the head of their virtual boss. While this will need to be confirmed with a larger study and participants in both roles, it appears that the assistance does not impact the enjoyment of the game negatively.

7.3 Presence and Embodiment

One observation made by one of our participants was that they were wondering whether they should be surprised that they did not feel a break in embodiment and presence when the co-pilot took control over their hand. This was surprising because they are losing autonomy over their virtual body to some degree when the Co-Pilot takes over. They suggested that this might be because requesting assistance is a conscious decision they make. Another comment that could be related to this observation came from another participant. They said that they considered the co-pilot as part of the game mechanic and not a second autonomous actor. Accepting the assistance as part of the game that only happens on-demand, might preserve the sense of presence and embodiment. However, this will need to be examined more closely in a future experiment.

7.4 The Role of the Co-Pilot

How the players inter-operate and what roles they assume is of particular importance because the games played with a co-pilot will not be designed to be played by two players. The goal of the co-piloting concept is to facilitate one player helping another, not performing the task for them. This experiment cannot answer most of these questions because the role and the game were both fixed for the experiment. Also, in order not impact their behaviour the experimenter did not engage in conversation. However, some early implications can be drawn from the participants' responses and behaviour. One would be how they interacted with the experimenter when requesting assistance. For example, three of the four participants usually asked politely for help and often thanked afterwards. The fourth participant, however, treated the assistance as some kind of supernatural ability and came up with a catchphrase to shout whenever they needed help. In a later comment, they explained that they forgot about the human part of the co-pilot and saw them as another game element because everything else would destroy their immersion. They also stressed that, in their eyes, the value of the co-pilot is highly dependent on them being willing to take on this assistive role and that an ill-suited helper would reduce the fun and the effectiveness drastically.

7.5 Limitations

7.5.1 Body Height and Natural Reach

One limitation of our experiment that we became aware of during the study was not capturing the size of the participant. The office level of Job Simulator was selected, because, it promised a range of actions at varying distances. We expected the participants to struggle with the actions far away and maybe even the ones at medium range when unassisted. However, one participant performed all tasks with ease and used the assistance only once. In the post-experiment interview, they stated that they had no issues because they are nearly 2 meters tall and have long arms. Should the frequency of use be considered as a metric for future experiments, the height of the participants needs to be captured as well.

7.5.2 Participant Selection

The recruitment of participants was hindered by the current unavailability of our lab facilities and other local COVID-19 restrictions.

This made it necessary to recruit participants who already have a VR-system at home and were willing to install experimental software on it. Naturally, this led to a very small and homogeneous participant pool. Most participants were male and most were either a VR-researcher themselves or friends of one. Together with the small number of participants, this could potentially skew the significance of the results. As described in the previous subsection, we noticed that body height can play a role in how often participants make use of the assistive technique. Considering that, a participant pool made up mostly of men, who are on average taller than women, could skew the results towards a lower average number of uses.

8 Conclusion and Future Work

We have introduced a concept for a remote assistive technique that can extend the reach of seated VR players through the help of another player to make VR games more accessible. We implemented a prototype and performed a small formative study to show its feasibility and gather impressions for further research directions. While limited, the study was successful in demonstrating the usefulness and feasibility of the concept.

In addition to that, we also identified a range of directions for future work. The most obvious one is about the technical way forward. While the concept is relatively straightforward, the implementation is not. A low-latency network connection needs to be established and the communication with the virtual environment is currently done through library hooks that are not officially supported, and thus the implementation is susceptible to changes. Indeed, at the time of writing our implementation is not compatible with the latest version of SteamVR, though users are able to downgrade this tool easily. This could be addressed through the industry exposing an interface for accessibility tools that allows manipulating the positions of individual devices. We are hopeful that upcoming full implementations of OpenXR [10] would provide such an interface. Combined with Steam Remote Play [15], which offers streaming of voice, game video, and controls, most of the prerequisites for an integration of remote co-piloting are already implemented in common PC games platforms and thus only needed to be pieced together.

Closer interaction with the industry could also open other ways to improve remote co-piloting. At the moment, the interface of our prototype is mostly just the video stream and indicators whether a hand is controlled or not. Acquiring more information about the state of the game or the VR system is likely to be complex and each game might require to be targeted individually. With a deeper integration in the VR or game platforms though, the co-pilot could be supplied with additional information. One example of this could be depth information. Over- and undershooting are issues that arise because the image that is shown to the co-pilot is just monoscopic 2D. However, with additional depth information, that image could be displayed on a stereoscopic display or a feature could be implemented that makes the virtual hand snap to the location of the closest object in its path. Both could help to prevent the over- or undershooting and make the assistance faster and more meaningful. Alternatively to retrieving the depth information directly from the VR-system, it could also be calculated from the received video stream as works such as the one by Gordon et al. [8] demonstrated.

Another direction that requires further research is the roles of the players and how they find and understand them. In this experiment, game and role were fixed for the participant and they only experienced the pilot role. It is mostly unknown how this concept would hold up when both roles are filled with participants and they encounter an unknown game. Another dimension worth exploring is how the experience changes when both players know each other very well compared to when they are total strangers. One last direction to further explore is the impact that remote co-piloting might have on the sense of presence and embodiment. Even though the co-pilot controls part of the pilot's body, one of our participants pointed out that they did not feel a break in embodiment or presence. This effect was unexpected by us and should be examined closer in the future.

Acknowledgments

The authors wish to thank Sebastian Friston and David Walton for their invaluable help in building the network implementation and testing of the prototype. We also thank David Swapp for his support throughout the project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 856998.

References

[1] G. Bednarski. https://www.walkinvrdriver.com/, 2020.

[2] J. Beeston, C. Power, and M. Barlet. *Accessible Player Experiences (APX): The Players*, pp. 245–253. 06 2018. doi: 10.1007/978-3-319-94277-3_40

[3] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, I3D '97, p. 35–ff. Association for Computing Machinery, New York, NY, USA, 1997. doi: 10.1145/253284.253301

[4] D. A. Bowman, D. B. Johnson, and L. F. Hodges. Testbed evaluation of virtual environment interaction techniques. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, VRST '99, p. 26–33. Association for Computing Machinery, New York, NY, USA, 1999. doi: 10.1145/323663.323667

[5] M. Dalgleish. There are no universal interfaces: how asymmetrical roles and asymmetrical controllers can increase access diversity. *G—A—M—E Games as Art, Media, Entertainment*, 1(7):3–4, 2018.

[6] L. Fanucci, F. Iacopetti, and R. Roncella. A console interface for game accessibility to people with motor impairments. In *2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*, pp. 206–210, 2011.

[7] K. Gerling, P. Dickinson, K. Hicks, L. Mason, A. L. Simeone, and K. Spiel. Virtual reality games for people using wheelchairs. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–11. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3313831.3376265

[8] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8976–8985, 2019. doi: 10.1109/ICCV.2019.00907

[9] F. Iacopetti, L. Fanucci, R. Roncella, D. Giusti, and A. Scebba. Game console controller interface for people with disability. In *2008 International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 757–762, 2008.

[10] Khronos Group. https://www.khronos.org/openxr/, 2020.

[11] M. Mott, J. Tang, S. Kane, E. Cutrell, and M. R. Morris. "i just went into it assuming that i wouldn't be able to have the full experience": Understanding the accessibility of virtual reality for people with limited mobility. In *ASSETS 2020*. ACM, October 2020.

[12] Owlchemy Labs. https://jobsimulatorgame.com/, 2020.

[13] I. Poupyrev, M. Billinghurst, S. Weghorst, and T. Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, p. 79–80. Association for Computing Machinery, New York, NY, USA, 1996. doi: 10.1145/237091.237102

[14] Steel Crate Games. https://keeptalkinggame.com/, 2020.

[15] Valve Corporation. https://store.steampowered.com/remoteplay, 2020.

[16] J. Wentzel, G. d'Eon, and D. Vogel. *Improving Virtual Reality Ergonomics Through Reach-Bounded Non-Linear Input Amplification*, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2020.

[17] Y. Zhao, E. Cutrell, C. Holz, M. R. Morris, E. Ofek, and A. Wilson. Seeingvr: A set of tools to make virtual reality more accessible to people with low vision. In *CHI 2019*. ACM, May 2019.