

Central Lancashire Online Knowledge (CLOK)

Title	Designing Accessible Visual Programming Tools for Children with Autism Spectrum Condition
Type	Article
URL	https://clock.uclan.ac.uk/id/eprint/32981/
DOI	https://doi.org/10.1007/s10209-021-00842-y
Date	2021
Citation	Zubair, Misbahu Sharfuddeen, Brown, David, Bates, Matthew and Hughes-Roberts, Thomas (2021) Designing Accessible Visual Programming Tools for Children with Autism Spectrum Condition. Universal Access in the Information Society. ISSN 1615-5289
Creators	Zubair, Misbahu Sharfuddeen, Brown, David, Bates, Matthew and Hughes-Roberts, Thomas

It is advisable to refer to the publisher's version if you intend to cite from the work.
<https://doi.org/10.1007/s10209-021-00842-y>

For information about Research at UCLan please go to <http://www.uclan.ac.uk/research/>

All outputs in CLOK are protected by Intellectual Property Rights law, including Copyright law. Copyright, IPR and Moral Rights for the works on this site are retained by the individual authors and/or other copyright owners. Terms and conditions for use of this material are defined in the <http://clock.uclan.ac.uk/policies/>



Designing accessible visual programming tools for children with autism spectrum condition

Misbahu S. Zubair¹ · David J. Brown² · Thomas Hughes-Roberts³ · Matthew Bates²

Accepted: 31 August 2021
© The Author(s) 2021

Abstract

Visual programming tools allow users to create interactive media projects such as games and animations using visual representations of programming concepts. Although these tools have been shown to have huge potential for teaching children, research has shown that they may not be accessible for children with cognitive impairments, including those with autism spectrum condition. Therefore, this study proposes a set of recommendations for the design of accessible visual programming tools for children with autism spectrum condition. Semi-structured interviews with experts ($n = 7$) were conducted and thematically analysed to identify initial recommendations. A second set of semi-structured interviews with a subset of the initial experts ($n = 3$) were then conducted to validate and produce a final set of recommendations.

Keywords Accessibility · Design · Visual programming · Autism · Learning disability

1 Introduction

Autism spectrum conditions (ASC), sometimes referred to as autism spectrum disorders (ASD), are diagnosed by the presence of social, communication and interaction difficulties, together with repetitive patterns of behaviour and restricted interests [3, 4]. The individual conditions that make up ASC (e.g. autistic disorder, Asperger's disorder, and pervasive developmental disorder) used to be considered as distinct conditions or disorders, however, the latest Diagnostic and Statistical Manual of Mental Disorders (DSM-5) consolidated them into ASD [3]. According to the new manual, the symptoms of these disorders “represent a single continuum of mild to severe social communication and restrictive repetitive behaviours/interest”. ASC is common among individuals with a learning disability, so much so that there has been a debate on whether the two conditions are distinct [33]. The United Kingdom's Department of Health even includes individuals with ASC as part of those

with learning disabilities as long as they do not have average or above-average intelligence [15].

Technology-based interventions have long since been used for individuals with disabilities, including those with ASC. A major advantage offered by this approach compared to traditional approaches is flexibility, i.e. it can be easily customised to meet the varying needs of users, and be adjusted automatically based on each user's interactions [2]. The emergence of smartphones and tablets has led to increased interest in technology-mediated education and interventions for individuals with ASC. This is because, in addition to the advantages offered by previous technological platforms, the mobility of such mobile devices extends the learning environment from classrooms to homes, which promotes learning [2]. Another important contributor to the popularity of technology-based intervention is the reported appeal they have to children with ASC compared to traditional intervention approaches [2, 18].

Using these modern technological platforms, interactive applications have been utilised to provide interventions for children with ASC. They have been used for improving communication skills [1, 11, 26], social skills [1] collaboration skills [39], and academic skills [24].

A technology-based approach used to teach skills and educational content to typically developing children that is not being fully utilised for children with ASC is content creation using Visual Programming Tools (VPTs). Visual

✉ Misbahu S. Zubair
mszubair@uclan.ac.uk; misbahu.zubair2015@my.ntu.ac.uk

¹ University of Central Lancashire, Preston, UK

² Nottingham Trent University, Nottingham, UK

³ Liverpool John Moores University, Liverpool, UK

programming allows users to “Specify a program in a two (or more) dimensional fashion” [32]. During the last two decades, VPTs (e.g. Scratch and Pocket Code) have been created specifically for children to learn to program and to create content related to their interests, e.g. animations, interactive stories and games. These VPTs represent programming concepts as visual objects (e.g. blocks or bricks) that can be put together to create a program, and include media resources that children can use to create programs, e.g. character sprites or models, background images or landscapes, and sounds.

This study builds on the findings of an accessibility evaluation of Scratch (that showed the existence of accessibility issues in the VPT) [40], by gathering a set of recommendations to guide the creation of accessible, usable and engaging VPTs for children with ASC, specifically those that also have a learning disability. Recommendations were gathered and then validated with experts by collecting data using semi-structured interviews and subsequent thematic analysis of these data. References to children with ASC made in the remainder of this paper includes the target group of this study, i.e. children with ASC and learning disability; the target group are also referred to as “users”, especially when presenting the gathered recommendations.

1.1 VPTs for children

This section briefly introduces the main VPTs found in the literature and/or used in practice.

Scratch is the most popular VPT for children based on findings in the literature. Using Scratch, children aged between 8 and 16 years can learn to program while creating “interactive, media-rich projects” such as animations or games [28]. Scratch has a rich media library for images and sounds, an inbuilt sound recorder, a painter for creating custom images, and allows the importation of existing media files. Each Scratch project consists of 2D media files controlled by a script made up of colourful command blocks snapped together. It has a single user interface with multiple panes to ease navigation, provides an error-free programming environment, i.e. there is no wrong way of snapping blocks that fit together, makes data concrete by displaying variables on-screen during execution, and visualises execution by highlighting blocks that are executing at runtime.

Pocket Code is a Scratch inspired mobile VPT targeted at children between 13 and 18 years old. Using block-based programming metaphors and a media library, it provides a platform for creating animations and games while learning to program [25]. Being mobile-based, Pocket Code also allows users to create programs that take advantage of mobile sensors such as compass and accelerometer.

Alice is an innovative three-dimensional VPT created by the Stage 3 Research Group at Carnegie Mellon University

[12]. It allows programming novices to create virtual worlds with animations, interactive narratives, and games. Projects in Alice are made up of 3D objects, each object holds its private properties (e.g. height and width) and member methods. At the start of a project, objects are added to the virtual world and positioned based on the needs of the user. Once objects are initialised, the program code is added using a drag and drop smart editor.

Kodu is a VPT created specifically for young children to learn to program through individual independent exploration [27]. It is built within a real-time 3D game and includes features to assist users in creating content such as a terrain editor, layout tools, character menus, etc. It reduces complexity by handling physics, collision detection and camera control. Kodu is packaged with several sample worlds that the user can edit, interactive lessons, and an online community for sharing worlds with other Kodu users. Programming in Kodu, similar to Scratch, is block-based, where blocks of instructions are put together using a “when - do” approach. In the “when” slot sensors are declared to determine a condition, and in the “dot” slot, the actions that should be performed are listed.

1.2 VPTs as learning and teaching aids for neurotypical children

Due to the graphical representation of programming concepts provided by VPTs, the most obvious application for children is in the context of learning programming visually instead of using text-based programming languages. A study by Meerbaum-Salant et al. [29] showed that middle school students with no previous programming experience achieved a reasonable understanding of Computer Science concepts using Scratch. The findings of another study by Fokides [17] with 138 5th grade students using Kodu to create games also showed that students were able to learn basic game design and programming concepts as they authored their first games.

Furthermore, proof of the suitability of VPTs for learning programming has been found when comparing the level of students’ programming abilities and misconceptions based on the programming language used to learn [30]. More than 200 10–12-year-old students from four schools participated in this study and were divided into six subgroups. Every two subgroups were assigned a separate programming language to learn to program with, i.e. two subgroups used Logo, another two subgroups used Python, and the last two subgroups used Scratch. Pre- and post-tests were used to assess the participants’ understanding of sequencing and loops. The results showed that misconceptions about loops were minimised when visual block-based Scratch was used compared to when text-based Python and Logo were used.

Another context in which VPTs have been applied is in learning specific academic subjects. Ke [22] implemented a design-based approach to learning mathematics with 64 middle school students. Participants worked within design groups and played several educational mathematics games before creating their own games using Scratch. The evaluation demonstrated improved attitudes towards mathematics by the students after participating in the game making process.

Storytelling is also a context in which VPTs are used. Burke and Kafai [10] investigated the possibility of creating Scratch computer programs as a way of developing the creative and storytelling abilities of children. Middle school children aged between 11 and 14 years took part in the study, and 60% of participants felt their storytelling skills improved.

In addition to learning programming, computational concepts, academic subjects and improving creativity, additional benefits of using VPTs have been reported. For example, Sáez-López et al. [37] found that the use of VPTs also resulted in increased motivation, fun and enthusiasm on the part of the children.

1.3 VPTs as learning and teaching aids for children with ASC

A study by Munoz et al. [31] shows that the computational thinking skills of children with ASC could be improved by the use of VPTs. The study involved seven children aged between 11 and 15 years participating in a five week workshop for creating games using Scratch. Results showed encouraging improvements in the computational skills of the participants, especially considering the fact that they had no prior programming knowledge.

Social skills of children with ASC could also be improved with the help of VPTs as reported by Eiselt and Carter [16]. These researchers conducted an 8-week programming course with eight children with high functioning ASC aged between 9 and 15 years which aimed to teach programming and social skills through game making. By the end of the study, improvements in programming knowledge and social interactions was observed. It is, however, interesting to note the findings reported by Bossavit and Parsons [5] from their study with two teenagers with ASC programming a game of their choice using KODU. The findings report the demonstration of problem-solving and programming skills by the participants, but very little interaction and collaboration between the two participants.

Overall, the literature was found to contain few studies on the use and benefits of VPTs by children with ASC, and the few that exist involve individuals with no reported associated learning disability. However, the accessibility evaluation of Scratch conducted by Zubair et al. [40] observed

seven participants with learning disabilities, five of whom were also diagnosed with ASC, using Scratch to create stories; and also interviewing their class teachers after these observations had taken place. Findings from the observations revealed that the participants struggled with a number of user-interface, and cognition-related accessibility issues. Findings from interviews with teachers revealed that these issues were caused by the VPT's poor visual labelling, lack of constraints, and the unavailability of templates to scaffold and guide users.

Although Bossavit and Parsons [5], Eiselt and Carter [16], and Munoz et al. [31] did not report any accessibility issues, very little is known about the cognitive abilities of the users that participated in these studies. It is possible that unlike the 5 participants with ASC that participated in the study conducted by Zubair et al. [40], the participants in the three studies above do not have an associated learning disability. This means that they may not face the same difficulties in using VPTs as those with ASC and related learning disabilities.

1.4 Accessibility recommendations for designing interactive applications for children with ASC

None of the studies that proposed accessibility recommendations for designing interactive applications for children with ASC found in the literature are targeted towards designing VPTs; however, recommendations have been proposed for designing accessible serious games [38], websites [9, 35], mobile applications [13], virtual reality (VR) applications [7, 19], tangible user interfaces [36] and other interactive applications [14, 23, 34].

Various methodologies were used in gathering these guidelines including literature reviews [7, 9, 13, 19, 38], engaging stakeholders [35], extending existing guidelines [23], combination of literature review and stakeholder engagement [34], and from research experiences [14, 36].

Despite the differences in the types of applications targeted, and the methods used by these studies, several recommendations were common among most or all of them. For example [9, 13, 34, 35] all recommend having visual objects as an alternative means of presenting information and [9, 13, 23, 35] recommended using simple straightforward language with no jargon, acronyms, etc. This shows that although there are accessibility recommendations that are context-specific, there are also important accessibility recommendations that address important needs of the target group and remain valuable across several contexts.

1.5 Aim of the study

This study aims to gather and propose a set of recommendations for designing accessible and engaging VPTs

for children with ASC, specifically those that also have a learning disability. In doing so it is hoped that more children with ASC will be able to use VPTs, with potential benefits reported to include improved engagement and collaboration [21].

2 Gathering recommendations

The following section describes the qualitative study designed to gather design recommendations for accessible VPTs for children with ASC.

2.1 Participants

Teachers with experience in working with children with ASC, and researchers with both practical experience with children with ASC as well as published works in areas related to their characteristics, needs, or their use of technological interventions were recruited as experts. They were recruited from existing links with special education needs schools and academic researchers in the relevant fields, and by searching online for academic experts with relevant experience. Seven experts (E1–E7) agreed to participate (three males and four females; two teachers and five academic researchers). Their professions, roles and experiences are presented in Table 1.

2.2 Method

All seven participants were interviewed between May–August 2018. Ethical approval was sought and received from Nottingham Trent University's Ethics Committee before participants were recruited and interviews were conducted. Steps were also taken to ensure anonymity and confidentiality of participants. Six of the interviews were conducted face to face, and one was conducted via video conferencing (Skype). All interviews were semi-structured, lasted between 30 and 60 min and were audio-recorded with the permission of the interviewee.

Before the start of each interview, the interviewer demonstrated the main features of VPTs by creating an example animation program using Scratch. Then the interview was conducted with questions asked around the observed behaviours and difficulties associated with using VPTs identified by a previous study by the authors that evaluated the accessibility of Scratch for children with learning disabilities and ASC [40]. Specifically questions were asked around: the occurrence of difficulties, i.e. why an observed difficulty occurred and whether similar difficulties are known to occur in other contexts; mitigating difficulties, i.e. how can an observed difficulty be mitigated and how are similar difficulties in other contexts mitigated; significance of behaviours, i.e. whether an observed behaviour is associated with any of the observed difficulties. The researcher was flexible and opportunistic in following up on relevant and interesting points as they arose, and by using the points raised by experts to decide the sequence in which behaviours and difficulties

Table 1 Interviewed experts, their professions and experience

Expert	Profession	Experience
E1	Researcher	Associate professor in education with research interests within the field of special education needs and disabilities including the use of technology to provide personalised learning experiences for those with disabilities. Has additional experience from working in special needs schools and as a language therapist for children with communication difficulties
E2	Researcher	Senior researcher with more than 10 years research experience in the area of assistive technologies, and strong interests in the use of virtual reality by children with autism spectrum condition
E3	Teacher	Assistant head teacher at a school for students aged 3–18 years with severe learning difficulties, profound a multiple learning difficulties and autism spectrum condition. Also has more than 20 years experience in the special education needs field
E4	Researcher	Professor in the field of learning disabilities with more than 30 years research experience in the field. Expertise in the design and evaluation of technology for students with a range of cognitive impairments and autism including virtual environments, serious games and robotics
E5	Researcher	Associate professor with research interests in the behaviour of autistic individuals, the use interactive applications (e.g. video games) as a method of intervention or to understand the behaviours of individuals with ASC. Has received research funding from bodies including US National Science Foundation and the UK National Institute for Health Research, and has collaborations in several countries
E6	Researcher	Associate professor with research interests in autism in education. Over 20 years research experience in various university research centres in multiple countries, and has worked in special needs schools
E7	Teacher	Classroom teacher with several years of experience at a school for students aged 3–18 years with severe learning difficulties, profound and multiple learning difficulties and autism spectrum condition

(observed from the aforementioned previous study) were discussed.

2.3 Analysis

Recorded interviews were first transcribed and then analysed using thematic analysis [8]. Specifically, theory-driven thematic analysis was used to analyse the interview transcripts. In this approach, the researcher codes and creates themes to answer specific questions that are being asked of the data [8]. Codes are used to denote basic parts of the data that are interesting and relevant to the issue being researched, and themes describe meaningful patterns in the data. The question that was asked to emphasise and identify (for coding) experts' recommendations for the inclusion or exclusion of features to improve accessibility was "What features should visual programming tools have or avoid to ensure that they are accessible, usable and engaging for children with ASC?".

All excerpts of the interview transcripts that directly referenced a recommendation for the inclusion or exclusion of a feature were coded, as were excerpts describing features/approaches that worked or did not work well for the same target group either applied to other software applications, or in real life. This approach was helpful in highlighting recommendations explicitly suggested by experts, as well as justification for making recommendations based on insights provided by experts.

After coding the interview transcripts, the first round of theme creation was performed by grouping together related codes to form potential themes. Coded extracts for each theme were then checked for consistency with each other, and the theme. This led to the elimination of redundant extracts, coded extracts that did not fit their codes or theme, and ultimately led to the creation of nine consistent and meaningful themes. A final pass at reviewing the themes, their codes and extracts was conducted. This time all codes and extracts were found to be consistent with their themes. Themes were named to highlight the main concept associated with their coded extracts; for example, the created theme that contained coded extracts emphasising the importance of personalisation features in making VPTs accessible was named "Personalisation".

QSR Nvivo 12 was used to conduct this data analysis. It was used to transcribe audio interview data, code data, and store themes.

Each theme and its related recommendations are discussed next. Excerpts from the transcripts related to each theme are provided to show the opinions of experts in their own words.

2.4 Findings

The nine themes identified from the transcripts cover expert recommendations on various matters ranging from the type of hardware platform most suitable for VPTs targeted at children with ASC, to the type of content that should be provided within the VPT and how it should be presented. For each theme discussed in this section, relevant excerpts from the transcripts are provided. A complete list of gathered recommendations organised by themes and their descriptions is also provided in Table 2.

2.4.1 Mobile device compatibility

The use of VPTs requires constant and continuous interaction with visual representations of programming logic (e.g. blocks and bricks), and objects that are being programmed (e.g. character and background images). These interactions, as pointed out by E6, may not be straightforward for all users with ASC, especially using a mouse as an input device.

They need to be learning the link between what they're doing... some of them play video games and they're very clear that what they do with their hands has an effect on what happens on the screen, but with other children, making that link, particularly if they're going to use a mouse can be difficult (E6)

A potential solution to this problem is making VPTs compatible with and accessible on mobile touch screen devices such as smartphones and tablets. This could greatly reduce the distress caused by difficulties in interacting with other platforms that offer less intuitive input techniques. In addition to the directness of interaction and intuitiveness offered by touch screen devices, E4 points out that most young people including those with ASC own and use smartphones and tablets, and are comfortable interacting with objects on a touch screen.

A lot of young people use iPads obviously, and they get used to controlling things with their finger. And that conceptual link between you touching it and it doing something is a much shorter jump isn't it, it's much easier to understand what's happening there. Whereas with the mouse when you have to do the double click on things, some of that will be difficult for some children. Even if you tell them they have to click twice, they leave it so long that it does not work as a double click (E4)

The need for an accessible interaction method cannot be overstated because as explained by E5, interaction difficulties can cause a ripple effect leading to difficulties in other areas.

Table 2 Recommendations gathered from analysing the interviews

Theme	Recommendation	Description
Mobile device compatibility	Ensure mobile device compatibility	VPTs should be compatible with and accessible on mobile devices, especially smartphones and tablets in order to allow easier access and interactions for children with ASC including those with motor difficulties
Engaging users	Integrate content known to interest children with ASC Suggest popular content to unmotivated users	Provide users with a diverse set of media and templates covering as many topics known to interest children with ASC as possible. For example, provide templates and media related to various forms of transportation, e.g. a space rocket launch project template, sprites and models of planets, astronauts, etc Content suggestions should be made to users (especially those barely interacting with the VPT) by suggesting content that is popular among other users with similar profiles
Information presentation and visualisation	Represent information visually using icons/symbols	Information within VPTs should be represented or supported with visual symbols/icons. For example, visual symbols or icons should be used to label objects or to support the objects' text labels throughout the user interface. This should include labels on buttons, tabs, panes, programming elements (blocks, bricks), etc
Sounds	Make sounds optional, turn off by default and allow users to control volume	The VPT should be usable with or without sounds. Sounds (including feedback sounds and program sounds) should not be audible unless explicitly turned on by the user, and volume control should be provided for users to adjust their sound level
Restrictions and limitations	Limit the choices of media elements available to users to a relevant and manageable set Limit the choices of programming elements available to users to a usable, manageable and relevant set Limit the features available to the user to those required to achieve the user's goal and are within the user's cognitive abilities	Users should be provided with a manageable and relevant subset of media items (e.g. characters, backgrounds, etc.) to work with, based on the goal of their project. For example, when creating a "space racing game", the choice of background images or landscapes options can be limited to only those related to outer space Provide users with a usable, manageable and relevant subset of programming elements (blocks, bricks, etc.) for their projects. For example, only a small subset of basic programming blocks should be available to a user with ASC and severe learning disabilities and ensuring that only blocks that reposition objects are available when the user is creating a script for moving an object VPTs usually have numerous tools and features for performing various tasks. Only those features that the user is capable of using and support the achievement of the user's goal should be available to the user. For example, a user that requires constant scaffolding should be restricted from using the "create blank/empty project" feature. Features that become subjects of obsessive behaviour to the extent that they stand in the way of goal achievement should also be restricted to allow the user to move on to achieve their goal

Table 2 (continued)

Theme	Recommendation	Description
Scaffolding	Provide templates for projects	VPTs should provide templates for creating a wide range of projects (e.g. games and animated stories). The templates should provide a visual structural scaffold for users by guiding them through the various stages of creating a project. For example, template for a story should guide users to choose characters, backgrounds, and actions for each character. They should also allow users to view a potential version of their end product
	Provide programming elements at higher levels of abstraction	VPTs should provide programming elements at different levels of abstractions for users with different abilities. For example, visual programming elements for “move along x” and “move along y” can exist at the low level. However, for those users that may be unable to create a script using these two programming elements to represent jumping, high-level programming elements “jump forwards” and “jump backwards” should be made available
Goal orientation	Design to ease, support and encourage success and goal achievement	Personalisation, restrictions, limitations, visualisations and scaffolding should all be applied in a way that helps and encourages users to achieve their goal(s) without having to perform unnecessary/inaccessible actions or tasks
Personalisation	Use profiles to store information about users	VPTs should have user profiles for storing relevant personal information about users (e.g. interests, capabilities and difficulties) and users’ interaction histories (e.g. frequency of programming blocks’ usage, properties of programs created). User profiles should be automatically updated with each use of the tool
	Personalise based on user profile and interaction history	VPTs should configure their user interface, apply the right restrictions, limitations, choose the right level of programming abstraction, etc. based on users’ profiles. As the user’s profile evolves, the tool should also reconfigure itself to keep up with the changing needs of the user.
Managing changes	Notify users before making any changes	Users should be made aware of any change or changes due to personalisation that will affect the way the VPT looks or functions before said changes are made. The notifications should be subtle and simple to comprehend
	Implement changes in small and manageable steps	Changes should be made in small manageable steps that can be handled by users. Drastic changes with major impacts should not be implemented at once
	Keep track of changes applied for a user	The VPT should keep a history of changes made to a user’s configuration. This should be available for a user to view visually as a form of journey tracker

For example, if you have difficulty performing mouse operations then maybe you're also going to have difficulty structuring and sequencing, not because you fundamentally have difficulty structuring sequencing but because you can't click the mouse in order to tell the computer what to do (E5)

2.4.2 Engaging users

Individuals with ASC are known to have restrictive interests; these interests may seem random, peculiar and sometimes strange. However, experts believed some core areas of interests are common among individuals with ASC, and integrating content related to these areas of interest within VPTs could potentially lead to higher engagement levels and motivate users to create programs around their areas of interests. E3 and E6 had this to say on the subject:

I mean there are some things that sort of generally float their boats in terms of ASC. I'm sort of being very generalistic here, for example, Thomas the Tank Engine, you know, they like the repetition, they like that it's not unpredictable and it's a very closed world (E3)

So there are people who have weird and wonderful interests but there are some things that a lot of young people are interested in and it's a bit different for boys and girls. So, for younger children, we know Thomas the Tank Engine is you know' they actually love Thomas. For older boys, a lot of boys still like trains but not Thomas. With girls, they tend to love animals, you know. There are probably a number of core things that you could perhaps say: 'we know that a lot of boys like these things and a lot of girls like these things and maybe we'll focus on integrating those' (E6)

In addition to providing content related to known core areas of interests of children with ASC, another suggestion made by E3 is to use content receiving high levels of interest from users within a VPT as a potential choice (to build projects around) for other users that are not motivated.

If a child is motivated by one character, maybe you could take that and get rid of the others and use it as a choice for somebody that is less motivated (E3)

2.4.3 Information presentation and visualisation

The use of visual icons and symbols instead of text, or to support text, was a key suggestion by experts. This is due to the reading and reading comprehension difficulties, and preference for visual means of communication associated with children with ASC. E6 expressed concern about having more text labels than visual labels in VPTs:

I mean there is the issue of having to be able to read for meaning, it's not just reading the words, it's reading and understanding what it is, you know. If I read that, what will I have to do? So you have to have a certain level of reading comprehension, don't you? I know there are other online programs that are directed at young people with a certain level of intellectual ability and reading ability. So if it's something that's supposed to be accessible to everyone perhaps there needs to be a bit less writing and more visual kind of representations (E6)

E4 also shared similar concerns about users needing to read and comprehend textual labels in order to use a VPT. To illustrate the difficulty of the task, E4 compared it to reading a foreign language that one cannot speak:

Think of it as if you were a native Italian speaker looking at this (scratch interface in English) and didn't know any English how much of it could you understand?

Therefore, as E6 mentioned, just like other applications targeted at children with disabilities in general, VPTs for children with ASC should present information visually, with as many visual symbols and icons as possible to make it easy for users to understand without the need to read text or text labels. E1 also made the same recommendation as seen in the excerpt below:

You could have a version that minimized the need to read or eradicated the need to read by putting symbols on it. You can have a run icon, you can have a jump icon and lots of other things (E1)

2.4.4 Sounds

ASC is commonly associated with high sensory sensitivities, and affected children can be highly sensitive to sounds. Therefore, E4 expressed concern over the use of sounds in VPTs:

Is auditory feedback necessary? Because I don't think at the moment that's a good idea, it may well be that the auditory system could be overloaded for some people with ASC (E4)

Since sounds may not cause distress for all children with ASC and could be useful in content creation and program execution, the recommendation proposed here is to make sounds optional for the use of VPTs and only played when turned on. Therefore, feedback should also be provided through other means, e.g. visual feedback for those that are sensitive to sounds and therefore have sounds turned off.

2.4.5 Restrictions and limitations

Experts unanimously believed that the numerous choices (e.g. sprites, backgrounds, sounds, blocks) and open-ended scenarios (e.g. blank projects) presented by VPTs can be overwhelming for children with ASC. This is true even in their everyday activities as explained by E3:

When you are working with autistic students and you ask them an open question they struggle with that. So for example, if I was to say “how do you feel?” there are sort of many different answers you can give. But if I say “do you feel happy or sad today?” it sort of closes it down (E3)

According to E4, another negative effect of having this level of choice comes in the form of children with ASC performing repetitive tasks as a way of controlling their arousal levels.

There’s a lot of theory around arousal levels and emotional disengagement in autism. A lot of the things observed [by the researcher] are attempts by the person to control their level of arousal. You are confronting somebody with a lot of information [within the VPT], and their return to doing repetitive tasks or something else is their attempt in saying “I can do this, let’s just carry on doing this” (E4)

Therefore, experts suggested the use of restrictions or limitations (when and where appropriate) as a way of providing safe and manageable options for children with ASC. This could potentially lead to them being comfortable enough to try out new things without getting overwhelmed. E6 suggests how this approach can be applied to providing a manageable number of media elements (e.g. story characters) to use in projects within a VPT:

So, you wouldn’t make 50 characters available, you maybe make five available, so they’d just pick from a really small number so that they’re not getting overwhelmed by the sheer number and not knowing what to pick. If there’s a small number, they can pick on what they like the look of and then quickly get into doing something with it (E6)

In addition to media elements such as characters, limitations or restrictions can also be applied to other aspects of VPTs such as tools, features, programming elements, etc. However, E1 cautioned that there might be a downside to this as having too many restrictions may prevent engagement.

So, the educator in me thinks I wouldn’t want the restrictions to be there all the time, because I would

be wanting to support them to do something more, but I would also want to be having it open enough so that they can get engaged in the first place (E1)

Therefore, a balance should be struck where restrictions exist to prevent users from getting overwhelmed and reacting negatively, but not too much to prevent users from engaging with the VPT.

2.4.6 Scaffolding

Experts believed that children with ASC may have difficulties with creativity and imagination, sequencing events, understanding and using programming elements (even if they are visual). They recommended the use of scaffolding to guide the users through the various stages of program creation, from media selection to program logic specification. As E6 explains below, individuals that require scaffolding to perform tasks that require sequencing and structuring (e.g. writing a story) without the use of a VPT will surely require scaffolding from the VPT if they are to use it to perform the same task independently.

Sequencing and structuring may well be things that those children have difficulty with when they’re writing a story by hand in a book, you know, some children have that difficulty. So again, there might be a need for a bit more scaffolding for children who might have that difficulty. If you just sit them in front of a piece of software and ask them to create a story, that will be really difficult (E6)

Children with ASC also find it difficult to visualise imaginary things, settings or situations as pointed out by E3:

A lot of students struggle with imagination. [Creating content using VPTs] is equivalent to say role-play, unless you’ve actually got something in front of them, some props, then they will struggle just to make a story out the sky (E3)

Scaffolds can also help children with ASC to avoid this difficulty by showing them what a possible outcome can look like, hence making it easy for them to create.

Having a sense of direction in the sense of reward, not reward but a clear idea of what an outcome might look like... I think sometimes young learners in general, but especially autistic groups, if you show them what the end goal is going to look like, they can better see the journey. And I think if you can provide scaffolding, thus enabling better structure and sequencing and things like that. I think that will better enable them to stay on track (E2)

Another potential benefit of having a scaffold as seen in the extract above is keeping users on track to achieve their goal and preventing them from getting lost in unrelated tasks.

Based on insights gathered from experts, we recommend the use of templates that can guide users through program creation, as well as programming specific scaffolds through the provision of highly abstracted programming blocks that represent scripts for performing popular actions.

2.4.7 Goal orientation

Findings show that experts believe designing VPTs with specific goals in mind can make it easier to identify the aspects of VPTs that need to be eliminated, restricted, automated or greatly simplified to allow children with ASC to focus on achieving goals and obtaining feedback. For example, E6 had this to say about eliminating reading activities to get users to focus on programming:

You want them to try and do something quickly you know, so you don't want them to have to read things and be concentrating, it's not a reading exercise it's a programming exercise. So if they're trying to read and work out what they have to do, it slows things down to the extent that they lose interest, and then you have lost purpose... If the purpose of the activity is to learn programming using the program then it needs to be easily accessible so they can do it quickly because then they'll get feedback, "wow I can make that work" and then they will want to do something more complicated (E6)

This approach could be a way of reducing frustration, increasing motivating and goal achievement, and encouraging actual visual programming. This, in turn, can lead to continued use of the VPT. However, success needs to be achieved quickly for this to occur as E4 mentions below:

If you don't understand [visual programming], more exposure is going to help you with that. But then you're not going to get more exposure unless you get some degree of success. One thing we tried to do when we were designing stuff was trying to make sure that people could actually get something done, there will always be some degree of challenge but they always got some degree of success because if they don't, they just give up (E4)

2.4.8 Personalisation

Children with ASC face varying degrees of difficulties, and experts stressed the need to acknowledge this when designing VPTs: "*you need some differentiation built-in*" (E2). For example, when discussing the appropriate application of

limitations and restrictions, E1 suggested making decisions based on the needs of each child: "*If I knew a child I would make different decisions according to the child.*" (E1). E6 also made a similar recommendation:

You have to work at the pace of the child because for some children they might stick at only having five symbols (programming blocks) for quite a long time whereas another child might get it straight away and the next week they want to be writing a story. So you might have to make more available for them more quickly (E6)

Personalisation can also be applied in prioritising the child's interest when providing content to create programs with. E7 believed that this can help increase engagement:

I think if it's following their interest a bit more I think that's going to make it more interesting to them, isn't it? It's picking up on something they know about and want to learn about. I think that's another additional problem with probably all of them, is finding the things that float their boat, which is a massive issue, which is why you know personalizing it is probably going to help with that difficulty (E7)

For personalisation, a VPT would require a knowledge of the characteristics of the child using it, and make decisions based on the characteristics and needs of the child. Therefore, the recommendations proposed here are to have user profiles for storing user information and then to personalise based on the contents of the profile.

2.4.9 Managing changes

Most of the recommendations discussed in the themes above propose some form of modification to the user interface or logic of VPTs to improve accessibility. Implementing these modifications will undoubtedly produce changes that will be noticeable by the user. Although children with ASC are known for their difficulties with dealing with change, experts suggested informing them about modifications to VPTs before they occur as a way of reducing the difficulty:

A characteristic of people with autism is that change can be something that's really challenging but if they know it's going to come and the connotations of the change are good ones then they can find it easy to cope with it (E4)

Some children will hate it if you change it from the way it was originally but actually in my experience you can, you can warn them, you can explain why you can present it as a good thing. So, I wouldn't say that it would be wrong to do it, but I think it's how you present it (E1)

It can be seen from the excerpts above that although children with ASC characteristically do not respond well to unexpected changes, they deal well with changes that they are made aware of, and understand the benefit of. Therefore making changes slowly and not drastically can make it easier for children with ASC to accept the changes as explained by E2:

If you can scaffold it and gently move things from one to two to three to four, you know, be clear about where they are in the journey, I think that the expectations can be managed well enough that it shouldn't represent a major problem (E2)

The recommendations proposed for helping children with ASC handle changes made to VPTs as a result of personalisation are: to present notifications whenever a change is due to be made; to avoid making major changes at an instance; and to have a feature that keeps track of changes applied for each child and presents it in a visual manner when requested.

3 Validating recommendations

Some of the initial recommendations gathered were explicitly stated by experts, while others were proposed by the authors to address concerns raised by experts. Therefore, a validation exercise was conducted to confirm the validity of all proposed recommendations. The validation study and its findings are presented in this section.

3.1 Participants

The same approach used to recruit participants in the initial study was used for participant recruitment in this study. Three experts, all of whom had participated in the initial study (E1, E4 and E6), agreed to participate.

3.2 Method

A qualitative approach to validation was employed using semi-structured interviews as the data collection method due to the success of this approach in the initial study and the richness of the data gathered. Since ethical approval was already secured for the research before the start of the initial study, no new ethics permission was required.

All interviews took place on May 2019, were held face to face, lasted between 30 and 45 min and were audio-recorded with the permission of the interviewee. Before the start of each interview, each expert was given a copy of the proposed recommendations. Then a semi-structured interview approach was adopted to investigate the validity of the recommendations previously derived within each theme. Experts were asked whether each recommendation could

improve VPTs' accessibility to children with ASC. Based on the expert's response, the interviewer then followed up on any interesting points raised by asking questions to either validate (e.g. how would it affect the experience of a user with ASC?), correct existing recommendations (e.g. how can the recommendation be improved?), or even to derive additional recommendations (e.g. what would you recommend instead?).

3.3 Analysis

The interviews were first listened to, to identify interesting concepts and then transcribed before the transcripts were coded. Coding was done by examining the data for opinions, suggested modifications, additions or eliminations. When any of these were identified, it was coded and assigned to the theme it relates to (no code was found to require a new theme). Then the codes in each theme followed a similar process as specified above. Findings showed agreement on the validity of all recommendations, two recommendation additions, no eliminations, and some suggested modifications.

QSR Nvivo 12 was used to conduct this data analysis. It was used to transcribe audio interview data, store notes, code data and update themes.

3.4 Findings

The findings of the validation exercise are discussed in this section. Excerpts from the interview transcripts are used to show experts' opinions about the recommendations in their own words. The changes made to the gathered recommendations as a result of the findings are presented in Table 3. Newly added recommendations and their descriptions, and updated recommendations for initial recommendations are presented in italics.

3.4.1 Mobile device compatibility

There was a general agreement by experts on the need for VPTs to be available on mobile devices:

It would be fantastic if it could be very portable, for children who can engage on phones that would be great. I think as a minimum it should be on some kind of tablet (E1)

If you can give them the opportunity to create something where they haven't got to be holding a pen or a mouse and they can create an animation or whatever it's going to be, I think that would be really important because we don't want to set up an alternative means of doing something that still has challenges for them. I think most children understand drag and drop because they do it all the time in games and things (E6)

Table 3 Changes to recommendations as a result of validation

Theme	Recommendation	Description
Engaging users	Integrate content known to interest those with ASC	In addition to the initial description: <i>where possible, make use of such media items on splash screens, lock screens, etc. to capture the attention of users</i>
	Suggest popular content to unmotivated users	In addition to the initial description: <i>this should only be applied for users that are not resistant to new information</i>
Information presentation and visualisation	Represent information visually using icons/symbols	In addition to the initial description: <i>restrictions should focus on features and not media content or programming elements, even if the user is fixated on them</i>
Goal orientation	<i>Provided templates should scaffold towards projects appropriate for teaching relevant skills to children with ASC</i>	<i>Templates should be designed for projects that teach children with ASC relevant skills such as communication and collaboration</i>
Personalisation and Customisation	<i>Store personal user information and preferences</i>	<i>VPTs should have user profiles for storing relevant personal information about users (e.g. interests, capabilities and difficulties). An automated user modelling test can be used to collect user data for initialising the user's profile, otherwise, the data can be entered manually by a caretaker</i>
	<i>Record users' interaction histories for personalisation</i>	<i>VPTs should record users' use of the tool (e.g. frequency of programming blocks' usage, properties of programs created). This record should be automatically updated with each use of the tool</i>
	<i>Support user customisation, i.e. manual selection of preferences</i>	<i>The VPT should also allow manual setting of preferences as a way of overriding automatic personalisation. For example, a user should be able to choose font size and colour</i>

3.4.2 Engaging users

Experts welcomed the recommendation for integrating content that is known to interest children with ASC within VPTs. E6 also suggested making the most out of this type of content to capture the attention of users:

Make sure that some of their favourite things are on the first screen that they see, I think that would be really important. If it's sharks or rainbows or unicorns you know, whatever their thing is "my favourite thing is going to be behind the screen, so I need to keep going" (E6)

However, with regard to making popular content suggestions to unmotivated users, E4 warned that this might not be suitable for those that have highly restrictive interests:

It may well be that in their profile you get a question about their resistance to new information. I can imagine scenarios where if you're really into Thomas the Tank Engine and somebody keeps saying you should use my little pony, then it won't go down very well (E4)

Considering these findings, the descriptions for the recommendations related to the integration of engaging content were modified.

3.4.3 Information presentation and visualisation

Although all experts agreed on the need for visualisation using icons/symbols, concerns over the choice of using new or existing symbols/icons to use were raised:

"Regardless of what they've been used to, some children will be able very quickly to pick up a new [visual] language, a new set of icons, they'll just quickly figure it out. And others obviously might want it to be something *maro familiar*" (E1)

E4 believed that using existing sets will be best since children with ASC will already be familiar with them, and if they are not, then they get the chance to learn a symbol set that will be useful to them in other contexts. This also helps to avoid putting the children through the unnecessary task of learning to understand the visual language in order to be able to use the tool. However, the use of an existing symbols set might come with certain Intellectual Property and Rights restrictions, and different children might be experienced in the use of different sets, which may mean having VPTs support multiple sets and to record individual preferences within profiles. Therefore, E6 favoured the idea of having new sets created and used for VPTs and argued that as long as the children are interested, they will be able to learn the new symbol set:

I found when we have introduced', whether it's for communication devices or programming, you know when kids started getting into the Lego programming and model making things because they're engaged with it and they are enthused, they can quite quickly learn the rules and the symbols that go with it. So I think if you have a new set and they're excited to use it, they will be able to learn (E6)

Considering these findings, we propose the use of existing visual/iconic languages where possible and only use new ones in cases where the use of existing ones is not possible due financial, legal or other constraints.

3.4.4 Sounds

The need for making all sounds optional for the use of VPTs was also agreed on by experts.

3.4.5 Restrictions and limitations

Experts all agreed on the need for restrictions and limitations to ensure users remain focused and not overwhelmed:

I think it's really important because when you give too many choices, it might look great, but they just don't know where to begin. It's linked to their executive function challenges, being able to differentiate between what's important and not important, and how to organize things (E6)

However, when imposing restrictions to help users get out of repetitive cycles, E4 advised against imposing them on content used to create programs even if the user is fixated on that content:

I think if they are only fixated on some topics then you should exploit that and remember that you are trying to get them to move up the levels of coding (E4)

The corresponding recommendation was updated to reflect the additional information gathered.

3.4.6 Scaffolding

All experts agreed on the need for structure using templates and the provision of scaffolds, to ensure children with ASC with different needs and abilities can program using VPTs:

It's an opportunity to really engage in a different way with something they've never done and it might', it should be highly motivating especially if it scaffolds at the right level (E1)

The use of templates is a good idea, particularly for the children having moderate or severe learning difficulties (E6)

3.4.7 Goal orientation

This recommendation was also well received by experts. E6 mentions that this design approach can also serve as a way of giving caregivers ideas on how VPTs can be used by children with ASC, thus encouraging them to introduce the use of the tools at home or in class:

I mean people pick out particular programs and tools because they think it will help with a particular goal. So this would be another way of getting the caregivers interested in picking this software (E6)

E6's insight led to the addition of a new recommendation for ensuring VPTs are goal oriented.

3.4.8 Personalisation

Recommendations concerning personalisation were greatly supported by all experts. However, a concern was raised by both E6 and E1 regarding the time and precision required to accurately set up profiles:

The user profile presumably is something that a teacher or parent or someone else might be able to put in as opposed to the child themselves right? So all of these things will take a certain amount of setting up (E6)

To tackle this issue and to improve the accuracy of the information within profiles, E1 suggested using a user-modelling test for children with ASC to automatically generate their profiles. Additionally, E1 suggested the provision of a feature that allows caretakers or the users themselves to further customise automatically generated profiles. This is useful especially in cases where the applied automatic personalisation is not producing the expected positive results.

3.4.9 Managing changes

Experts all agreed that these recommendations are necessary for managing the frustrations that children with ASC face when dealing with change.

4 Discussion

The aim of this study is to propose recommendations for designing accessible, usable and engaging VPTs for children with ASC, specifically those that also have a learning disability. A literature review was carried out, but no recommendations were found for designing accessible and usable

VPTs for children with ASC or children with cognitive impairments in general. In response to this identified gap, the first part of the current study gathered recommendations by conducting semi-structured interviews with seven experts and using thematic analysis to analyse the interviews. This led to the proposal of 16 recommendations categorised under nine themes identified from this thematic analysis. However, not all the recommendations were proposed by the experts themselves, some were proposed by the authors to mitigate concerns raised by the experts. To ensure the validity of both recommendations by the experts and the authors, a validation exercise was conducted using semi-structured interviews with three experts. Findings from this validation exercise ensured the validity of all gathered recommendations and also provided new data that led to the addition of two new recommendations. Several initial recommendations were also updated to further clarify their meaning. This was mainly done by rewording or adding content to their descriptions, although in one case one recommendation was broken into two separate recommendations. This increased the total number of proposed recommendations to 19, while the number of themes remained unchanged.

The first recommendation proposed suggests making VPTs compatible with mobile touch screen devices. This recommendation aims to take advantage of the widespread ownership and use of such devices by the target group. Interviewed experts believe that the familiarity that most children have with this mode of interaction would make it easier for the target group to understand the interactions required to make use of VPTs. This could also reduce the difficulties associated with clicking and dragging objects using a mouse and other less intuitive input devices faced by those with fine motor skills difficulties. Accessibility recommendations for designing other types of applications for the target group have also proposed utilising mobile devices for improving accessibility [9, 13].

Although making VPTs accessible on mobile devices can make it easier for children with ASC to access and use them, access does not always imply usage. One way of getting the attention of children with ASC is by taking advantage of their interests [6]. Therefore, the experts recommended integrating content related to the topics known to interest children with ASC within VPTs, and in clear sight of users, to encourage interaction and programming of projects. A similar recommendation has been made for designing accessible virtual reality applications for the target group [7]. Davis et al. [14] also recommended accommodating special interests of children with ASC when designing interactive applications as well as avoiding content associated with any fears the users may have.

VPTs are known for visually representing programming constructs and program execution. However, for these tools to be accessible for children with ASC, visualisation must

be extended to the labelling of all user interface elements. Findings from this study show that experts were concerned about the accessibility of VPTs depending on the abilities of children with ASC to read and understand textual information. To address this, experts recommend the use of visual icons/symbols in place of text, or together with text to convey the purpose of user interface elements including visual programming elements (blocks or bricks). This recommendation was found to be popular in the literature [9, 13, 34, 35] as it can improve the accessibility of any application, especially if the visual icons/symbols used are familiar to the users [9, 23, 35].

Avoidance of default audio feedback and playback within VPTs is another recommendation made by experts, as auditory hypersensitivity is common among individuals with ASC [20]. However, complete elimination of sounds from VPTs is not recommended since audio feedback might be useful and even entertaining to some children. The recommendation is to have all sound off by default, make the VPT usable without sound, and only play sounds when a user explicitly declares that preference as recommended by Bozgeyikli et al. [7] when designing accessible VR applications.

VPTs targeted at children offer collections of media objects, visual programming elements, and in some cases, additional tools/features for editing, creating, storing and sharing media, programming elements and projects. While this may be advantageous for encouraging creativity in neurotypical children, experts warned that children with ASC might find having numerous choices overwhelming. This can affect their ability to pick out programming elements as well as media objects to program. Therefore, limiting these choices to a set that is manageable and useful (given the user's goals) reduces the possibility of overwhelming the user and makes it easier for them to make choices. Although meaningful usage of VPTs requires a subset of visual programming elements and media objects, other tools/features available within the VPT may not be entirely necessary depending on the nature of the project being created. Such tools can be removed to declutter the user interface and provide fewer options for the user. Similarly Bozgeyikli et al. [7] and Davis et al. [14] recommend preventing users with ASC from performing certain tasks or accessing certain features by removing the features or taking over control of the applications at certain times.

Experts believe that users with ASC may require scaffolding to program within VPTs. An important observation made by one expert (E6) is that users that require scaffolding to perform a task without the use of technology will not be able to perform that same task with the use of interactive technology unless the technology itself provides scaffolding. Therefore, the use of templates within VPTs is recommended to provide guidance and structure to build projects around. To further scaffold the users' programming experience, the

provision of higher abstract level programming elements is also recommended. For example, some users might be able to create a script that makes a character jump using visual programming elements for moving along the x and y -axis. However, not all users will be able to accomplish this, and in this case, the provision of a “jump” visual programming element that abstracts the movement along x and y -axis will be more suitable. Although scaffolding can be important for the accessibility of all interactive applications, these proposed scaffolding recommendations are unique compared to the scaffolding recommendations found in the literature, e.g. providing relevant examples [35] and providing multimedia instructions [9, 34]. This is due to the creative nature of VPT use compared to other applications, e.g. websites.

Designing VPTs to be goal-oriented instead of “sandbox” in nature is also recommended by experts. This will allow users to quickly access and use features that are relevant to their goal [9, 14, 23], program an executable project, obtain feedback, and become more motivated. Without providing goal-oriented focus, VPTs may become platforms where children with ASC spend most of their time on secondary or low-level tasks.

Support for each individual user’s unique characteristics, interests and goals should be offered within VPTs. The use of profiles is recommended for storing user information as a route to personalisation. The user profile should be a live data store created and updated either manually or automatically by logging relevant user interactions. The VPT should then make decisions based on the contents of the user profile on how other recommendations should be applied, e.g. the restrictions to apply to programming elements. A recommendation is also proposed for allowing the user or a teacher to explicitly specify preferences for overriding those automatically set by the tool, in cases where the changes made by the tool do not suit the user. Accessibility recommendations for designing for those with ASC propose personalisation and customisation of various features such as text size, colour and font [9, 23, 34], characters and environments [7, 38], and number of elements within the interface [9].

Personalising to meet and subsequently keep up with changing needs, interests and capabilities of users will undoubtedly lead to changes that affect how the tool looks and functions. Individuals with ASC are characterised as being resistant to change, and by their preference for consistency and predictability. Therefore, it is proposed that users are made aware of any changes that are going to happen due to personalisation beforehand. Additionally, the provision of a feature that presents a visual history of personalised changes applied for a user is recommended. Only Khowaja and Salim [23] proposed a similar recommendation, while Britto and Pizzolato [9] recommended restricting interface changes and proposed keeping users informed about the status of the system and providing constant feedback to users.

The complete set of proposed recommendations are provided in Table 4. They are presented in a descriptive manner intentionally to ensure their applicability to VPTs of different nature (e.g. those that support block-based programming or those that have a game based programming environment), and to allow flexibility in the way they are applied by designers and developers. However, to avoid ambiguity, concrete examples have been provided (where it appears appropriate) to show how the recommendations can be applied to existing VPTs; however, these examples are in no way meant to be exhaustive.

5 Limitations

Like any research investigation, this study has its limitations. The interviews were mainly driven by discussions around the outcomes of an earlier study, detailing the behaviours and difficulties faced by five children with ASC while using Scratch 2.0. However, we believe the recommendations that were gathered as a result of discussing these observations can be generalised to making all VPTs accessible. Secondly, although all experts that were interviewed had extensive experience in designing interventions for children with ASC (either traditional or technology-based), not all were familiar with VPTs before participating in this study (although three were very experienced in using VPTs). To address this, a Scratch programming demonstration was shown at the start of all interviews.

6 Conclusion and future work

In spite of the popularity of VPTs for use by neurotypical children, and the reported benefits of using them in a range of learning contexts, there is still a lack of research on their use for children with ASC, especially those with a learning disability. Following on from previous research evaluating the accessibility of VPTs for children with learning disability including those also diagnosed with ASC [40], this work aimed to propose a set of recommendations for designing accessible, usable and engaging VPTs for children with ASC. An initial set of recommendations were gathered from interviewing seven experts before being updated after validation by interviewing a subset of the previous seven experts (three in total). Finally, a set of 19 validated accessibility recommendations for designing accessible VPTs for children with ASC and learning disability are proposed. Although these do not represent an exhaustive set, they have all been validated by a set of experts.

Empirical studies that evaluate the accessibility of VPTs designed with the proposed recommendations need to be conducted in order to fully validate the recommendations.

Table 4 Complete set of proposed recommendations

Theme	Recommendation	Description
Mobile device compatibility	Ensure mobile device compatibility	VPTs should be compatible with and accessible on mobile devices, especially smartphones and tablets in order to allow easier access and interactions for children with ASC including those with motor difficulties
Engaging users	Integrate content known to interest children with ASC	Provide users with a diverse set of media and templates covering as many topics known to interest children with ASC as possible. For example, provide templates and media related to various forms of transportation, e.g. a space rocket launch project template, sprites and models of planes, astronauts, etc. Where possible, make use of such media items on splash screens, lock screens, etc. to capture the attention of users
	Suggest popular content to unmotivated users	Content suggestions should be made to users (especially those barely interacting with the VPT) by suggesting content that is popular among other users with similar profiles. This should only be applied for users that are not resistant to new information
Information presentation and visualisation	Represent information visually using icons/symbols	Information within VPTs should be represented or supported with visual symbols/icons. For example, visual symbols or icons should be used to label objects or to support the objects' text labels throughout the user interface. This should include labels on buttons, tabs, panes, programming elements (blocks, bricks), etc. Existing symbol/icon sets should be used where possible, otherwise a new, easy to understand set of symbols/icons can be created and used
Sounds	Make sounds optional, turn off by default and allow users to control volume	The VPT should be usable with or without sounds. Sounds (including feedback sounds and program sounds) should not be audible unless explicitly turned on by the user, and volume control should be provided for users to adjust their sound level

Table 4 (continued)

Theme	Recommendation	Description
Restrictions and limitations	Limit the choices of media elements available to users to a reasonable and meaningful set	Users should be provided with a manageable and relevant subset of media items (e.g. characters, backgrounds, etc.) to work with, based on the goal of their project. For example, when creating a “space racing game”, the choice of background images or landscapes options can be limited to only those related to outer space
	Limit the choices of programming elements available to users to a reasonable and meaningful set	Provide users with a usable, manageable and relevant subset of programming elements (blocks, bricks, etc.) for their projects. For example, only a small subset of basic programming blocks should be available to a user with ASC and severe learning disabilities and ensuring that only blocks that re-position objects are available when the user is creating a script for moving an object
	Limit the features available on the interface to the ones that are needed to achieve the user’s goal, and restrict access to features that control redundant things	VPTs usually have numerous tools and features for performing various tasks. Only those features that the user is capable of using and support the achievement of the user’s goal should be available to the user. For example, a user that requires constant scaffolding should be restricted from using the “create blank/empty project” feature. Features that become subjects of obsessive behaviour to the extent that they stand in the way of goal achievement should also be restricted to allow the user to move on to achieve their goal. Restrictions should focus on features and not media content or programming elements, even if the user is fixated on them
Scaffolding	Provide templates for projects	VPTs should provide templates for creating a wide range of projects (e.g. games and animated stories). The templates should provide a visual structural scaffold for users by guiding them through the various stages of creating a project. For example, a template for a story should guide users to choose characters, backgrounds, and actions for each character. They should also allow users to view a potential version of their end product
	Provide programming elements at higher levels of abstraction	VPTs should provide programming elements at different levels of abstractions for users with different abilities. For example, visual programming elements for ‘move along x’ and ‘move along y’ can exist at the low-level. However, for those users that may be unable to create a script using these two programming elements to represent jumping, high-level programming elements “jump forwards” and “jump backwards” should be made available

Table 4 (continued)

Theme	Recommendation	Description
Personalisation and customisation	Store personal user information and preferences	VPTs should have user profiles for storing relevant personal information about users (e.g. interests, capabilities and difficulties). An automated user modelling test can be used to collect user data for initialising the user's profile; otherwise, the data can be entered manually by a caretaker
	Record users' interaction histories for personalisation	VPTs should record users' use of the tool (e.g. frequency of programming blocks' usage, properties of programs created). This record should be automatically updated with each use of the tool
	Personalise based on user profile and interaction history	VPTs should configure their user interface, apply the right restrictions, limitations, choose the right level of programming abstraction, etc. based on users' profiles. As a user's profile evolves, the tool should also reconfigure itself to keep up with the changing needs of that user
	Support user customisation, i.e. manual selection of preferences	The VPT should also allow manual setting of preferences as a way of overriding automatic personalisation. For example, a user should be able to choose font, size and colour
Goal orientation	Design to ease, support and encourage success and goal achievement	Personalisation, restrictions, limitations, visualisations and scaffolding, should all be applied in a way that helps and encourages users to achieve their goal(s) without having to perform unnecessary/inaccessible actions or tasks
Managing changes	Provided templates should scaffold towards projects appropriate for teaching relevant skills to children with ASC	Templates should be designed for projects that teach children with ASC relevant skills such as communication and collaboration
	Notify users before making any changes	Users should be made aware of any change or changes due to personalisation that will affect the way the VPT looks or functions before said changes are made. The notifications should be subtle and simple to comprehend
	Implement changes in small and manageable steps	Changes should be made in small manageable steps that can be handled by users. Drastic changes with major impacts should not be implemented at once
	Keep track of changes applied for a user	The VPT should keep a history of changes made to a user's configuration. This should be available for a user to view visually as a form of journey tracker

Therefore, as future work, we intend to implement these recommendations into an existing VPT and evaluate its accessibility for children with ASC. The first set of recommendations that will be implemented as part of this future work are those proposed within the personalisation theme, and will mainly be applied to scaffolding and limitations related to programming elements. An ontology will be used to represent programming elements categorised based on complexity and purpose, and how they should be made available to users depending on their characteristics (as represented in their profiles). Three personae created for children with ASC by the authors will be used to create the user profiles that will be used in the ontology.

Declarations

Conflict of interest This study was funded by PTDF Nigeria. The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Alessandrini, A., Cappelletti, A., Zancanaro, M.: Audio-augmented paper for therapy and educational intervention for children with autistic spectrum disorder. *Int. J. Hum. Comput. Stud.* **72**(4), 422–430 (2014). <https://doi.org/10.1016/j.ijhcs.2013.12.001>
- Allen, M.L., Hartley, C., Cain, K.: iPads and the use of apps by children with autism spectrum disorder: do they promote learning? *Front. Psychol.* **7**, 1305 (2016). <https://doi.org/10.3389/fpsyg.2016.01305>
- American Psychiatric Association: Diagnostic and Statistical Manual of Mental Disorders (DSM-5). American Psychiatric Association (2013)
- Baron-Cohen, S., Scott, F.J., Allison, C., Williams, J., Bolton, P., Matthews, F.E., Brayne, C.: Prevalence of autism-spectrum conditions: UK school-based population study. *Br. J. Psych.* **194**(6), 500–509 (2009)
- Bossavit, B., Parsons, S.: From start to finish: teenagers on the autism spectrum developing their own collaborative game. *J. Enabling Technol.* **11**(2), 31–42 (2017). <https://doi.org/10.1108/JET-02-2017-0004>
- Boyd, B.A., Conroy, M.A., Mancil, G.R., Nakao, T., Alter, P.J.: Effects of circumscribed interests on the social behaviors of children with autism spectrum disorders. *J. Autism Dev. Disord.* **37**(8), 1550–1561 (2007)
- Bozgeyikli, L., Rajj, A., Katkoori, S., Alqasemi, R.: A survey on virtual reality for individuals with autism spectrum disorder: design considerations. *IEEE Trans. Learn. Technol.* **11**(2), 133–151 (2018). <https://doi.org/10.1109/TLT.2017.2739747>
- Braun, V., Clarke, V.: Using thematic analysis in psychology. *Qual. Res. Psychol.* **3**(2), 77–101 (2006). <https://doi.org/10.1191/1478088706qp063oa>
- Britto, T., Pizzolato, E.: Towards web accessibility guidelines of interaction and interface design for people with autism spectrum disorder. In: ACHI 2016: The 9th International Conference on Advances in Computer-Human Interactions, pp. 1–7 (2016)
- Burke, Q., Kafai, Y.B.: Programming & storytelling: opportunities for learning about coding & composition. In: Proceedings of the 9th International Conference on Interaction Design and Children, IDC '10, pp. 348–351. ACM, New York (2010). <https://doi.org/10.1145/1810543.1810611>
- Chien, M.E., Jheng, C.M., Lin, N.M., Tang, H.H., Tael, P., Tseng, W.S., Chen, M.Y.: iCAN: a tablet-based pedagogical system for improving communication skills of children with autism. *Int. J. Hum. Comput. Stud.* **73**, 79–90 (2015). <https://doi.org/10.1016/j.ijhcs.2014.06.001>
- Cooper, S., Dann, W., Pausch, R.: Alice: a 3-d tool for introductory programming concepts. *J. Comput. Sci. Coll.* **15**(5), 107–116 (2000)
- Dattolo, A., Luccio, F.L.: Accessible and usable websites and mobile applications for people with autism spectrum disorders: a comparative study. *EAI End. Trans. Amb. Syst.* **4**(13), e5 (2017). <https://doi.org/10.4108/eai.17-5-2017.152549>
- Davis, M., Dautenhahn, K., Powell, S., Nehaniv, C.: Guidelines for researchers and practitioners designing software and software trials for children with autism. *J. Assist. Technol.* **4**(1), 38–48 (2010)
- Department of Health: Valuing people: a new strategy for learning disability for the 21st century. Tech. rep, Department of Health (2001)
- Eiselt, K., Carter, P.: Integrating social skills practice with computer programming for students on the autism spectrum. In: 2018 IEEE Frontiers in Education Conference (FIE), pp. 1–5. IEEE (2018). <https://doi.org/10.1109/FIE.2018.8659252>
- Fokides, E.: Students learning to program by developing games: results of a year-long project in primary school settings. *J. Inf. Technol. Educ.: Res.* **16**, 475–505 (2017). <https://doi.org/10.28945/3893>
- Goldsmith, T.R., LeBlanc, L.A.: Use of technology in interventions for children with autism. *J. Early Intensive Behav. Interv.* **1**(2), 166–178 (2004). <https://doi.org/10.1037/h0100287>
- Herrera, G., Vera, L., Sevilla, J., Portalés, C., Casas, S.: On the development of VR and AR learning contents for children on the autism spectrum: from real requirements to virtual scenarios. In: Augmented Reality for Enhanced Learning Environments, pp. 106–141. IGI Global (2018)
- Hughes-Roberts, T., Brown, D., Boulton, H., Burton, A., Shopland, N., Martinovs, D.: Examining the potential impact of digital game making in curricula based teaching: initial observations (2020). [10.1016/j.compedu.2020.103988](https://doi.org/10.1016/j.compedu.2020.103988)
- Hughes-Roberts, T., Brown, D., Boulton, H., Burton, A., Shopland, N., Martinovs, D.: Examining the potential impact of digital game making in curricula based teaching: initial observations. *Comput. Educ.* **158**, (2020). <https://doi.org/10.1016/j.compedu.2020.103988>
- Ke, F.: An implementation of design-based learning through creating educational computer games: a case study on mathematics learning during design and computing. *Comput. Educ.* **73**, 26–39 (2014). <https://doi.org/10.1016/j.compedu.2013.12.010>

23. Khowaja, K., Salim, S.S.: Heuristics to evaluate interactive systems for children with autism spectrum disorder (ASD). *PLoS One* **10**(8), e0136977 (2015)
24. Knight, V., McKissick, B.R., Saunders, A.: A review of technology-based interventions to teach academic skills to students with autism spectrum disorder. *J. Autism Dev. Disord.* **43**(11), 2628–2648 (2013). <https://doi.org/10.1007/s10803-013-1814-y>
25. Koitz, R., Slany, W.: Empirical comparison of visual to hybrid formula manipulation in educational programming languages for teenagers. In: *Proceedings of the 5th Workshop on Evaluation and Usability of Programming Languages and Tools, PLATEAU '14*, pp. 21–30. ACM, New York (2014). <https://doi.org/10.1145/2688204.2688209>
26. Lorah, E.R., Parnell, A., Whitby, P.S., Hantula, D.: A systematic review of tablet computers and portable media players as speech generating devices for individuals with autism spectrum disorder. *J. Autism Dev. Disord.* **45**(12), 3792–3804 (2015). <https://doi.org/10.1007/s10803-014-2314-4>
27. MacLaurin, M.B.: The design of Kodu: a tiny visual programming language for children on the Xbox 360. *SIGPLAN Not.* **46**(1), 241–246 (2011). <https://doi.org/10.1145/1925844.1926413>
28. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The scratch programming language and environment. *Trans. Comput. Educ.* **10**(4), 16:1–16:15 (2010). <https://doi.org/10.1145/1868358.1868363>
29. Meerbaum-Salant, O., Armoni, M., Ben-Ari, M.M.: Learning computer science concepts with scratch. *Comput. Sci. Educ.* **23**(3), 239–264 (2013). <https://doi.org/10.1080/08993408.2013.832022>
30. Mladenović, M., Boljat, I., Anko, A.: Comparing loops misconceptions in block-based and text-based programming languages at the k-12 level. *Educ. Inf. Technol.* **23**(4), 1483–1500 (2018). <https://doi.org/10.1007/s10639-017-9673-3>
31. Munoz, R., Villarroel, R., Barcelos, T.S., Riquelme, F., Quezada, A., Bustos-Valenzuela, P.: Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access* **6**, 63880–63889 (2018). <https://doi.org/10.1109/ACCESS.2018.2877417>
32. Myers, B.A.: Taxonomies of visual programming and program visualization. *J. Vis. Lang. Comput.* **1**(1), 97–123 (1990). [https://doi.org/10.1016/S1045-926X\(05\)80036-9](https://doi.org/10.1016/S1045-926X(05)80036-9)
33. O'Brien, G., Pearson, J.: Autism and learning disability. *Autism* **8**(2), 125–140 (2004)
34. Pavlov, N.: User interface for people with autism spectrum disorders. *J. Softw. Eng. Appl.* **7**(02), 128–134 (2014). <https://doi.org/10.4236/jsea.2014.72014>
35. Raymaker, D.M., Kapp, S.K., McDonald, K.E., Weiner, M., Ashkenazy, E., Nicolaidis, C.: Development of the AASPIRE web accessibility guidelines for autistic web users. *Autism in Adulthood* **1**(2), 146–157 (2019). <https://doi.org/10.1089/aut.2018.0020>
36. Sitdhisanguan, K., Chotikamthorn, N., Dechaboon, A., Out, P.: Using tangible user interfaces in computer-based training systems for low-functioning autistic children. *Pers. Ubiquitous Comput.* **16**(2), 143–155 (2012). <https://doi.org/10.1007/s00779-011-0382-4>
37. Sáez-López, J.M., Román-González, M., Vázquez-Cano, E.: Visual programming languages integrated across the curriculum in elementary school: a two year case study using scratch in five schools. *Comput. Educ.* **97**, 129–141 (2016). <https://doi.org/10.1016/j.compedu.2016.03.003>
38. Tsikinas, S., Xinogalos, S.: Design guidelines for serious games targeted to people with autism. In: Uskov, V.L., Howlett, R.J., Jain, L.C. (eds.) *Smart Education and e-Learning 2019*, pp. 489–499. Springer, Singapore (2019)
39. Zancanaro, M., Giusti, L., Bauminger-Zviely, N., Eden, S., Gal, E., Weiss, P.L.: Noproblem a collaborative interface for teaching conversation skills to children with high functioning autism spectrum disorder. In: *Playful User Interfaces*, pp. 209–224. Springer (2014). https://doi.org/10.1007/978-981-4560-96-2_10
40. Zubair, M.S., Brown, D., Hughes-Roberts, T., Bates, M.: Evaluating the accessibility of scratch for children with cognitive impairments. In: Antona, M., Stephanidis, C. (eds.) *Universal Access in Human-Computer Interaction, Methods, Technologies, and Users*, pp. 660–676. Springer, Cham (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.