Understanding Digital Game Authoring Practices of People with Intellectual and Developmental Disabilities using Scratch

Hasani Gayani Kariyawasam Peradorapage
Department of Human-Centred Computing
Monash University
Melbourne, Vic, Australia
hasani.kariyawasamperadorapage@monash.edu

Kirsten Ellis
Department of Human-Centred Computing
Monash University
Melbourne, Vic, Australia
kirsten.ellis@monash.edu

Swamy Ananthanarayan
Department of Human-Centred Computing
Monash University
Melbourne, Victoria, Australia
swamy.ananthanarayan@monash.edu

Abstract

Digital game authoring has been made accessible to novice creators through block-based programming environments such as Scratch. However, the understanding of how people with intellectual and developmental disabilities (IDD) engage in digital game authoring remains under-explored. We conducted a three-month qualitative study with ten adults with IDD and four support coaches, enrolled in a digital game-making program using Scratch at a disability support organisation. Data were collected through semi-structured interviews with participants and support coaches, a game-model mapping activity, researcher observations, and analysis of Scratch game artefacts created by participants. Participants created 41 Scratch games across seven game types, with coaches scaffolding the entire authoring process. All participants demonstrated creative agency in their games yet encountered difficulties with idea generation and debugging at some point during the process. Based on the findings, we propose five design considerations to guide the development of a more inclusive block-based game authoring platform such as Scratch.

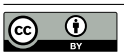
CCS Concepts

• **Human-centered computing** → **Empirical studies in HCI**; **Empirical studies in accessibility**.

Keywords

game authoring, Scratch, digital games, intellectual disability, design considerations

ACM Reference Format:
Hasani Gayani Kariyawasam Peradorapage, Kirsten Ellis, and Swamy Ananthanarayan. 2026. Understanding Digital Game Authoring Practices of People with Intellectual and Developmental Disabilities using Scratch. In *The 28th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '26), October 25–28, 2026, Vila Nova de Gaia, Portugal.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3797867.3829040



This work is licensed under a Creative Commons Attribution 4.0 International License.
ASSETS '26, Vila Nova de Gaia, Portugal
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2521-0/26/10
https://doi.org/10.1145/3797867.3829040

1 Introduction

Digital game authoring is a creative process [35] in which designers combine rules, mechanics, and systems of interaction between game elements to facilitate meaningful play experiences [62]. Rules explain how a game system functions, mechanics determine the actions through which players make meaningful choices, and systems of interaction govern how game elements respond to one another [62]. Unlike playing, game creation requires a deeper engagement of system analysis and troubleshooting, which develops ownership that can instil confidence in creators [36, 68]. Prior work has shown that game-authoring positively influences problem-solving skills, social interaction, learning ownership [8], motivation [52], and decision-making [1].

While block-based programming environments such as Scratch [50] have made digital game authoring more accessible to creators with no prior programming knowledge, limited research has explored game authoring by people with IDD, who may experience difficulties with learning, memory, and attention [57]. IDD encompasses intellectual disability (ID), characterised by significant limitations in intellectual functioning and adaptive behaviour [24, 53], alongside developmental disabilities (DD), referring to impairments in physical, learning, language, or behavioural areas [54]. Existing work on digital games for people with IDD has focused on using games as intervention tools to teach workplace skills [10, 64], social integration skills [34], and relationship boundaries [17]. As a result, people with IDD have been mostly positioned as players of games designed for them, rather than as creators of their own games [60]. Where they have been involved in creation, participation has typically been limited to play-testing [21, 31, 48, 63] and personalisation based on sensory profiles [69]. This highlights the need for a deeper understanding of how people with IDD author digital games.

In this paper, we employ Scratch [50] to understand the game authoring practices of people with IDD. Scratch has been successfully adopted in programming interventions for children with IDD [20, 25], including storytelling [71] and computational thinking [70]. The participants in our study were already familiar

with Scratch, having engaged with it through a game-making program at a local disability support organisation (DSO) — a program initiated in response to participants who had been playing games and expressed interest in creating their own. To our knowledge, only limited research has explored how people with IDD use Scratch, and no research has examined how adults with IDD engage in digital game authoring. This study addresses these gaps by investigating the following research questions:

RQ1: How do people with IDD engage in game authoring using Scratch, and what types of games do they produce?

RQ2: What challenges do people with IDD encounter during game authoring, and what enablers and barriers shape their engagement?

RQ3: How can the Scratch environment be designed to better support people with IDD during game authoring?

We conducted a three-month qualitative study across two sites of the DSO that runs Scratch-based game-making programs for adults with IDD, with sessions of three hours each. The study included 10 participants and 4 support coaches. We observed game authoring sessions over 60 hours on-site and conducted in-person semi-structured interviews with participants and coaches. We also explored how participants conceptualised their games, identified learning strategies, creative choices, accessibility barriers, emotional responses, and support needs involved in game authoring. Building on our research questions and data collected, this paper makes three contributions to human-computer interaction and accessibility research:

(1) We present the results of an empirical study of how people with IDD conceptualize and engage in digital game authoring using Scratch, characterising their authoring practices and the games they produced.
(2) We discuss the challenges and enablers that people with IDD encounter during the game authoring process.
(3) We propose five design considerations for block-based game authoring platforms such as Scratch to better support game creation by people with IDD.

2 Related Work

We begin by reviewing existing work on digital games developed for people with IDD, then explore how Scratch has been used by other communities to create digital games, and finally examine how people with IDD have engaged with Scratch.

2.1 Digital Games for People with IDD

People with IDD have largely been positioned as players of games designed for them rather than as creators. Research has mainly focused on serious games, designed with a primary purpose beyond entertainment [61], to teach vocational skills such as workplace navigation [64] and cleaning tasks [10], daily routine skills such as brushing teeth [34], and social skills including relationship boundaries [17]. A meta-analysis of eight randomised controlled trials has also confirmed improvements in adaptive and cognitive skills through such approaches [59]. Games have also been developed for treating phobias [27], and for training cognitive skills [47, 49]. Additionally, full-body embodied interaction games have also been developed [13, 26]. Leisure-oriented and inclusive play remains comparatively under-explored [61].

Despite the majority of work focusing on games as intervention tools, some studies have involved people with IDD in the creation process, though the depth of involvement varies widely. Most commonly, participants served as play-testers of pre-designed prototypes [14, 48, 63]. Some studies however offered decision-making over specific elements. For example, in PigScape [26], children customised graphics, difficulty levels, created narration, and gesture controls, while Anthony et al. [2] engaged post-secondary students with learning differences in identifying accessibility requirements and improving existing prototypes during a 60-minute workshop. Others provided participants a more direct role in shaping design. For example, Ringland et al. [69] tailored games to individual neurodiverse children through rounds of paper and digital prototyping based on each child's sensory profile and interests.

These studies show that people with IDD are predominantly positioned as players or play-tester of games rather than creators, leaving limited understanding of how they engage in and experience game creation themselves.

2.2 Digital Game Authoring using Scratch

Scratch is a block-based programming environment developed by the MIT Media Lab, which enables users to create games, animations, and stories, and facilitates sharing and remixing (modifying a copy of another project) within an online community [43, 50]. Scratch-based game authoring has been predominantly studied with children and adolescents across coding workshops [4, 56], school classrooms [6, 55], online communities [3, 16], and in underrepresented countries [44, 45], with research on adults remaining limited to university students [32, 40] and adults with histories of opioid addiction [19].

When authoring games, remixing was a common strategy, where students modified replicas of familiar games like Flappy Bird [51] by changing aesthetics (characters, backgrounds) and sounds (songs) [66]. Students also used code-blocks from the Scratch community projects [16] and posted questions related to their work in the Scratch community [3]. Further, the authoring process was scaffolded by researchers with real-time assistance [30, 45, 51, 66].

Challenges arose at every stage. At ideation, participants were uncertain about what ideas were feasible, compounded by Scratch's blank canvas interface, which left them reliant on mentors to find starting points [19, 37]. During development, implementing game logic proved difficult [16, 30]. Cheng et al. [16] found that participants struggled with tasks like changing object speeds, and user-defined variables. During a four-day workshop, students encountered issues with initialising sprites and mainly avoided complex logic such as repeat-until blocks, list data structures, and cloning, likely due to limited familiarity [30]. These challenges also carried emotional responses, such as confusion when mental models were contradicted, frustration when code did not work as expected, and boredom and disengagement when difficulties persisted [55, 56]. A recurring challenge was the gap between what participants wanted

to create and what they were able to implement. In cases where implementations were too complex, researchers stepped-in, to implement those for the students [19, 45].

This suggests that game authoring in Scratch is a cognitively demanding process, which requires sustained scaffolding, structured starting points, and real-time support.

2.3 Scratch Experiences for People with IDD

Scratch has also been the dominant platform for programming interventions with people with cognitive disabilities and IDD [20, 25], used in contexts such as storytelling [71] and computational thinking [33, 39, 70]. Two studies have specifically involved people with IDD in Scratch-based game authoring. Dorsey et al. [23] engaged children with autism aged 9-10 in game creation across five sessions of one-and-a-half hours each, where they were asked to create a video game in which they were the lead character. Children from the lower support needs displayed stronger interest in the game design activity. However, game authoring was only one component of the study, which also included storytelling and robotics activities. Guedes et al. [28] engaged adults with ID in creating games using Scratch and Makey Makey, finding that greater creative control led to enhanced engagement and sense of ownership, though game authoring was again one activity within a broader making workshop. The accessibility of Scratch has been examined through user evaluations in which children with cognitive disabilities aged 13–14 created stories [71] and through expert heuristic inspections [72].

These studies highlight that independent game authoring by people with IDD remains under-explored. While prior work suggests that creative and game-making activities can foster enhanced engagement, sense of ownership, and sustained interest [23, 28], it remains unclear how adults with IDD independently engage in Scratch-based game authoring, and what enables or constrains this process.

3 Methods

In this paper, we use the term IDD to refer to our participant group, as it encompasses the range of disabilities of participants we worked with at the DSO. The research was conducted in collaboration with a local DSO that supports over 500 adults with IDD across 20 sites within a single geographic region. The DSO's support coaches deliver programs across a wide range of interests, skills, and employment goals to help people with IDD achieve their ambitions. GameCraft is a structured, coach-facilitated digital game-making program initiated in 2024 at the request of participants who had been playing games and expressed interest in learning to make their own. The program has been running for approximately two years using Scratch, delivered five days a week across two sites, with sessions typically supporting 1–3 participants each. We have an ongoing engagement with the DSO through supporting the delivery of this program, and the present study was embedded within the existing program rather than introduced as a separate research activity.

We conducted a three-month study within the GameCraft program to better understand how people with IDD authored games using Scratch, facilitated by the organisation's support coaches. The first author attended all sessions, observing the process and providing technical help only when coaches requested assistance. The study involved 10 participants and 4 coaches, drawing on multiple data sources: session observations, semi-structured interviews, game-model mapping activity, and Scratch-game artefacts created by participants, which were analysed to examine the game authoring process.

Ethical approval for the study was obtained from our university Human Ethics Committee and from the DSO. To obtain informed consent from participants, we provided an easy-read information sheet with images, verbal explanations, and a short video providing an overview of the study purpose. Prior to the start of each session, coaches reminded participants about the research study, to ensure they had an understanding of their participation. Participants were also informed that they could withdraw at any time without affecting their ongoing participation in the GameCraft program. Study materials and interview questions were also adapted to participants' communication preferences, including visual supports, simplified language, and additional time when needed.

3.1 Participants and Coaches

3.1.1 ***Participants.*** Our study included 10 people with IDD enrolled in the GameCraft program at two sites (Table 1). All program participants (n=14) were invited to take part in the study, and 10 consented. Participants (7 males, 2 females, 1 non-binary) ranged in age from 20 to 40 years (M = 28.5, SD = 6.11) and had a range of disability backgrounds, including autism spectrum condition (n=4), intellectual disability (n=2), and co-occurring conditions such as intellectual disability with cerebral palsy (n=1), intellectual disability with autism spectrum condition (n=2), and visual disability with autism (n=1). Disability backgrounds were self-reported. All participants had prior experience using computers to browse or play online games. Five participants were based at each site. Scratch experience, gained through the GameCraft program, varied: 1 year (n=3), 8 months (n=4), 3-4 months (n=2) and 2 months (n=1). During the program, participants worked individually on their own Scratch projects using a computer connected to the internet. Participants exhibited a range of language and reading capabilities, from full-sentence communication (n=5) to short sentences (n=1) and limited word use (n=4). Most participants (n=7) did not require reading or spelling assistance, while (n=3) did. Therefore, we gathered more quotes from some participants while observing the behaviours of the others. All participants (n=10) took part in at least four game-authoring sessions, while 9 completed the game-model mapping activity.

3.1.2 ***Coaches.*** The study also included four support coaches (2 male, 2 female) employed by the DSO (Table 2), with experience working with people with IDD ranging from 2 to over 5 years. Coaches facilitated GameCraft sessions and supported participants throughout the game authoring process. None had formal programming experience; while C1 and C2 developed their Scratch knowledge through the program itself, C3 and C4 had only used Scratch two to three times. At Site 1, C3 and C4 alternated weekly as both had joined the program that year. At Site 2, C2 ran all sessions independently and had facilitated the program for one year. C1, who initiated the program and ran it at Site 1 for over a year, was

not teaching in the current semester, but was interviewed given his foundational role in establishing the program.

3.2 Researcher Positionality

Participants and coaches were central to this study, and it is equally important to acknowledge how the researchers' own positionality shaped the study's design and the interpretation of participants' experiences. All authors maintain ongoing relationships with the organization where the study was conducted, and these engagements informed the study's design and its integration into the organization's existing GameCraft program. Prior to the research, the first author volunteered regularly at the organization, attending sessions and supporting coaches and participants during game authoring activities.

The last two authors have a long-standing collaboration with the disability support organisation, contributing resources, materials, and guidance for inclusive technology initiatives. Their sustained engagement provided valuable contextual understanding of participant routines, support needs, and organisational practices, while also requiring reflexivity regarding how familiarity might shape participant behaviour. The research team's expertise in HCI, accessibility, and qualitative fieldwork informed the study's use of observations, semi-structured interviews, the game-model mapping activity, and analysis of Scratch artefacts. None of the authors identify as having an ID, so we approached data collection and analysis with this outsider perspective in mind, examining observations alongside insights from coaches and centering participants' own accounts of their interactions.

3.3 Program Structure

The program ran five days a week, with 2 days at Site 1 and 3 days at Site 2, supporting 1-3 participants per day. During the program, three hours were spent on game authoring using Scratch, which formed the basis of our data collection, followed by 90 minutes of Xbox gaming, which was not part of the study. While there were no scheduled breaks within this three-hour block, participants could pause and resume working as needed. At the beginning of some sessions, coaches selected a YouTube tutorial for all participants to follow, but on most days, participants were encouraged to create a game they preferred. Participants were usually encouraged to start a new game after completing a project, as a result, they often worked on the same project for 2–3 consecutive weeks. However, some participants started a new project each session. For example, during the program, participants first opened the devices provided by the DSO and logged into their Scratch accounts. Coaches would then ask, whether they had any work remaining from the previous week, or what they wanted to work on that day. Participants then proceeded to work on their individual projects. The coach moved around the room to support each participant as needed. During this time, the first author observed participants as they engaged in the game creation process and recorded field notes.

3.4 Interviews and Artefacts

3.4.1 ***Coach Interviews.*** Semi-structured interviews in two sessions, at the beginning and end of the study period for all coaches, with each coach's total interview time lasting approximately 30–40 minutes. These interviews explored coaches' perspectives on how participants engaged in game creation, the support coaches provided, and the challenges they observed.

3.4.2 ***Participant Interviews.*** Semi-structured interviews were conducted with participants to understand how they started making their games, the details of their games, what facilitated their process, the challenges they encountered, and how they navigated those challenges. Interviews ranged from 5 to 12 minutes, depending on each participant's communication style. Interviews were conducted only with participants willing to communicate verbally and were recorded only with consent.

3.4.3 ***Observations and Artefacts.*** The first author recorded field notes during all 20 game-making sessions across both sites, totalling over 60 hours of observation. Notes focused on how participants started their games, selected tutorials or assets, added and tested blocks, responded to errors, and sought support from coaches. A detailed summary of each session was written at the end of each day to track changes over time. Scratch game artefacts were also collected at the end of each day, totalling 41 projects by the end of the study period.

3.5 Scratch Game-Model Mapping Activity

A Scratch game-model mapping activity was conducted twice with each participant in the middle weeks of the study, to capture how they conceptualised game-making. The game-model mapping activity used a set of printed cards (Figure 1), each measuring 8 × 6 cm for ease of handling, featuring images paired with simple word labels (three words maximum) in large text, making them easy to read, pick up, and place on a desk. Cards were categorised into characters, backgrounds, controls, actions, objects, and endings. Colours were used to distinguish categories, with the exception of characters and backgrounds. The colour selection was inspired by Scratch's block categories, though not a direct mapping, as some cards represented higher-level functions requiring multiple Scratch blocks in practice. Participants were not explicitly informed of this categorization. Initial card content was determined through analysis of games participants had produced and preferences mentioned during interviews and sessions, so the activity was grounded in characters and themes meaningful to them rather than unfamiliar content. These were then presented to participants, who were asked to suggest any additions or removals. Two participants suggested cards such as "hint," "increase speed," "decrease speed," and "score," which fell outside the provided categories. These were incorporated as additional cards. The final set included 131 cards: 35 characters, 19 backgrounds, 13 controls, 26 actions, 26 objects, 7 endings, and 4 additional cards. During the activity, participants were asked to select cards they would use to create a game in Scratch. They were informed that any number of cards could be selected, and there were no right or wrong answers. After selecting the cards, participants were first asked whether they understood what each card meant. They explained what each represented, before describing the mapped game. Images of all completed models (n=9; one participant opted out) were captured, and the entire process was video recorded.

Table 1: Participant Demographic Information

Participant ID	Site	Program Day	Age	Gender	Self-Reported Disability	Scratch Experience	Language Capabilities
P1	Site 1	Day 1	26	M	Autism Spectrum	1 year	Full Sentences
P2	Site 1	Day 1	25	M	Autism Spectrum	1 year	Full Sentences
P3	Site 1	Day 1	21	M	Autism Spectrum	1 year	Limited Words
P4	Site 1	Day 2	20	M	Intellectual Disability & Autism Spectrum	8 months	Full Sentences
P5	Site 1	Day 2	31	M	Intellectual Disability & Cerebral Palsy	8 months	Limited Words
P6	Site 2	Day 3	32	Non-binary	Visual Disability & Autism Spectrum	2 months	Full Sentences
P7	Site 2	Day 3	26	F	Intellectual Disability	3–4 months	Limited Words
P8	Site 2	Day 3	40	F	Intellectual Disability	3–4 months	Limited Words
P9	Site 2	Day 4	34	M	Autism Spectrum	8 months	Full Sentences
P10	Site 2	Day 5	30	M	Intellectual Disability & Autism Spectrum	8 months	Short Sentences

Table 2: Coach Demographic Information

Coach ID	Site	Gender	Disability Experience	Scratch Experience Through Program
C1	Site 1	M	5+ years	1 year
C2	Site 2	M	2 years	1 year
C3	Site 1	F	5+ years	3 months
C4	Site 1	F	3 years	3 months

3.6 Data Analysis

The collected data were analysed using reflexive thematic analysis [11, 12], adopting a hybrid deductive–inductive approach to identify, analyse, and interpret patterns throughout the dataset. The first author led the analysis, beginning by familiarising themselves with all data sources: observation notes, interview recordings and transcripts, game-model mapping video recordings, and Scratch artefacts. Every saved Scratch game was opened, played, and inspected, then analysed using structured set of features including game type, player controls, sprites used, scoring, rules, win/lose conditions, and sound use. Initial coding of interview transcripts was conducted by the first author using open coding at the sentence and paragraph levels, to inductively identify meaningful patterns. These codes were reviewed and refined with the during weekly meetings, involving all three authors, who examined the codes and underlying data together. Where interpretations differed, the transcripts and videos were revisited and discussed, and the codes refined to reflect these discussions. The dataset was coded iteratively across multiple rounds of refinement and consolidation. Observation field notes were initially coded in alignment with the research questions, focusing on how participants engaged with game creation, the resources and strategies they drew on, and the challenges they encountered. These observation codes were similarly reviewed with the team and revised accordingly.

As coding progressed, codes from interviews and observations were compared and consolidated, with patterns from both sources mutually informing and refining the developing themes. Characteristics of mapped game models and Scratch artefacts were then examined alongside these themes, extending participants' conceptual understanding and the structural features of their games. Less verbal participants were given additional time in interviews and used alternative methods to respond (e.g., visual aids, including cards from the model-mapping activity, or pointing at the screen). Their perspectives were also captured through observation notes and artefacts from the model-mapping activity. This process produced approximately 20 code clusters, such as "agency through customisation of assets" and "lack of debugging support" which were grouped under broader categories such as "Creative decision-making and asset customisation" and "Code comprehension and literacy gaps". Related codes were then consolidated into candidate themes. Following the development of initial themes, all data sources were revisited to assess alignment, examining instances that both supported and challenged each theme. Through this process, we arrived at four final themes: (1) Emerging Skills in Authoring, (2) Coaches Scaffolding, (3) Creative Agency, and (4) Barriers to Authorship.

4 Findings

This section presents findings from observations, interviews, game-model mapping activity, and Scratch artefacts. Our analysis identified key insights related to participants' approaches to acquiring ideas and procedural knowledge, creative choices, the barriers they encountered, and the role coaches played in mediating the authoring process. We organise our findings into four thematic sections: Emerging Skills in Authoring, Coach Scaffolding, Creative Agency, and Barriers to Authorship.

4.1 Emerging Skills in Authoring (RQ1)

Participants displayed varying levels of capabilities during the program, shaped by a range of factors including their distinct intellectual and developmental disabilities, personal interests, and individual creative goals. All coaches reported that this variation

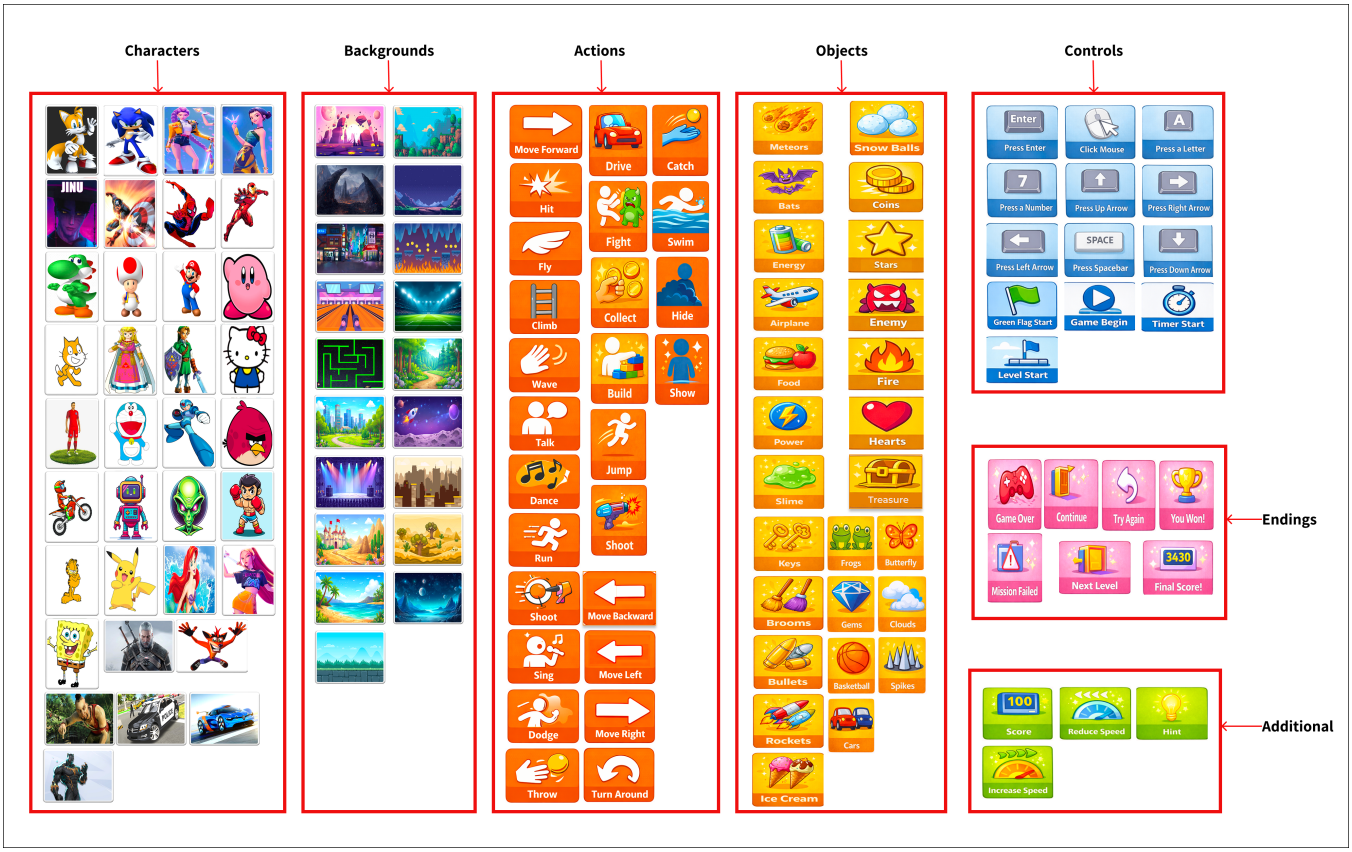


Figure 1: All cards used for the Scratch game-model mapping activity, categorised under characters, backgrounds, actions, objects, controls, endings, and additional categories.

influenced the pace at which participants' skills developed and the amount of support participants needed. Some participants were able to develop more games compared to the others. C1 noted that some participants produced *"three to four times more games than the other clients"*. C4 adapted sessions based on group composition, noting that *"it's difficult to come up with one thing for the whole group"*. During the program, this variation was evident: two participants (P5, P10) explored the platform without purposeful building, unless prompted by the coach, with P10 also requiring the coach to spell search terms. P7 similarly required coach assistance to search for tutorials, but many participants (n=6) searched for tutorials on their own. This variation in support needs shaped every aspect of the game authoring process as evident across all themes. Participants' skills as game authors developed through this co-produced process, with coaches scaffolding tutorials, debugging, and idea generation throughout — even as participants took on more of the authoring process themselves over time.

Coaches structured the program to incrementally introduce complexity as participants authored games. It began with simple designs and gradually progressed to more complex tasks. As C1 explained, *"we start off with a very simple design, and then learn that and play with that... and then we'll step up to something with more complex coding"*, with new elements added gradually. This approach was necessary to manage the increasing difficulty as complexity grew. YouTube tutorials were the primary entry point to scaffold the authoring process for both participants and coaches. P4 described tutorials as central to his workflow: *"I like to follow by just doing... looking at YouTube 'cause that actually does help"*. During the program, 7 participants searched YouTube for tutorials at the start of game-making sessions. Over time, coaches aimed to reduce this dependency, encouraging participants to *"try and have a go without a tutorial"* (C2), though tutorials continued as a dominant scaffolding mechanism throughout the program.

A significant variation was visible in the participants' tutorial engagement, based on instruction-following practices. Four participants (P3, P4, P6, P8) frequently replicated the tutorials precisely. For example, P6 was precise about the spacing between the characters, while two (P3, P6) followed the pacing of the video without skipping. By contrast, others engaged more selectively: some (P1, P5) skipped through videos to locate specific code-blocks, and P2 kept the audio at the lowest volume, focusing on visuals only, copying the code-blocks. One participant (P9) used tutorials as a modifiable base, adding blocks on his own without following the tutorial exactly. However, coaches noted that some tutorials were too fast or used complex explanations that participants had difficulty following without coach support. One Coach highlighted that

many tutorials were inaccessible due to being text-heavy, noting that *"a lot of our guys don't read"*(C4).

Over time, participants drew on different methods to acquire ideas and procedural knowledge. This included duplicating the code-blocks from previous working projects while only customising the visual and audio assets (Figure 4). Four participants (P1, P2, P3, P6) frequently demonstrated this practice to maintain familiarity and manageability. Two participants (P3, P9) referenced codes from multiple sources, including previous projects, and YouTube. P3 searched Scratch community projects to understand specific concepts when stuck, describing how he was looking for *"the project that explains gravity"* to fix an issue in his own game, yet he did not copy from it. P1 described employing multiple approaches when creating as *"I would either think about it, search it up, or just do a list of my own"*. Two participants (P2, P4) often observed what their peers were working on and sometimes asked them about the games they were developing. P2 noted that he would ask his peers *"Oh! How did you do that?"*. While some coaches considered these approaches as helpful, C4 noted that such approaches could possibly lead them to not create original work, as they were *"not actually doing anything from scratch"*.

All coaches emphasised that long-term continuous practice was important for consolidating game authoring skills, as they observed participants improve over the period of the program. Six participants searched for their own tutorials by typing *Scratch games* on YouTube without coach prompting, and P2 noted that searching for his own tutorials *"makes me more independent"*. C2 described that participants *"are just working by themselves on it. I don't really have to give a lot of support,"* with some progressing to initiating sessions on their own by logging into their Scratch accounts. C1 similarly reported that participants *"learn themselves where they can go to find the information they need to know, rather than having someone else do it for them"*. This progression was further shaped by how participants developed understanding through long-term use of the blocks. Block recognition for many participants (n=4) developed through repeated exposure to the concept through use, rather than explicit concept explanation. P1 and P4 also described using colour and text to recognise blocks. Similarly, C2 noted that asking participants to find *"a dark orange block that says forever"* was more effective than naming the block's function alone. Recreating the same game repeatedly helped build foundational understanding as participants familiarised themselves with the platform, structure, and code-blocks. C1 described how exposure to multiple game types built a repertoire of reusable structures, with participants having *"learned three or four different coding styles"*. However, P2 expressed a desire for more variety, stating *"I'm tired of doing the same thing now"*. Participants' understanding of code was primarily grounded in observable game behaviour rather than abstract programming concepts. When explaining their games, participants consistently described on-screen game behaviour rather than the underlying code logic. C1 described this pattern: participants knew how the code *"reacts within your visual frame, like the characters and the background"*.

4.2 Coach Scaffolding (RQ2)

Coaches took on multiple roles to scaffold the game-making process, including suggesting ideas, explaining tutorials in simpler language, guiding debugging, and providing emotional reassurance. These roles were not fixed; coaches dynamically adjusted their support based on each participant's needs.

All coaches prioritised participant enjoyment and comfort over technical understanding. C2 described his *"number one priority"* as ensuring participants were comfortable and enjoying the program, and C1 similarly noted he wanted participants to *"learn something and get a bit of fun out of it as well"*. Two coaches began each session by discussing what games participants were playing and what they would want to make, while others used completed examples to build confidence, believing that seeing a finished product would motivate participants. Participants described a similar approach; P4 mentioned *"I go on YouTube on my phone and just go, 'Okay, I can do that'"*.

Coaches provided support at different levels depending on individual participant needs. Five participants only needed brief initial guidance, and they continued to work on their own after that. C2 noted, *"once I've guided them through what they need to do and how to do it, they pretty much just work on it by themselves"*. For participants (P6, P7, P8) who found tutorial content difficult to follow, coaches watched videos alongside them and explained in simpler terms: *"I would basically translate what the video is saying, but in a more easier and less complex way"* mentioned C2. P7, for example, started the tutorial and waited for the coach to provide instructions for each action. Coaches also put tutorials on a shared screen for the group to follow, and acknowledged that support had to be *"very individualised"*(C4) ranging from simple verbal to visual explanations using pictures. The highest level of support coaches provided was sitting beside P10, who had reading difficulties, to help him identify the code-blocks to use.

When participants encountered breakdowns, most (n=7) attempted to fix problems on their own before requesting help. P1 explained: *"I would try to do it on my own, though watching a tutorial or having somebody by my side would help out sometimes,"* and some (n=3) mentioned revisiting tutorial steps as their first step in the debugging process. Six participants cross-checked their current code-blocks against previous working projects to identify mistakes. However, self-debugging was not always successful. P1 noted that *"it doesn't always work sometimes,"* and upon failure, participants turned to coaches, developing a consistent hierarchy of assistance. At this stage, coaches prompted self-reflection, asking *"So why do you think the game's not working?"*(C2) and guided participants to check back through reference material. If the error continued, coaches prompted collaborative debugging, as C2 mentioned *"it's not clear what the problem is until we play around a little bit with the coding,"*. In certain instances, when coach-supported debugging also failed to fix the issue, participants abandoned their games. C2 noted this: *"it's come to a point when we stopped working on it 'cause it just wouldn't work"*.

Emotional support from coaches was consistent throughout the authoring process. Coaches provided positive feedback, for example *"Well, you did do it, mate. That was all your hard work"*(C2). When

improvements were needed, suggestions were framed as encouragement: *"maybe next time, if you wanted to, you could add this to make it more challenging"*(C2). Coaches also noted that participants looked forward to reassurance, with C2 noting that they *"always want that second reassurance just to make sure the game and the code's working how they wanted it"*. Five participants notably prioritised coach reassurance during the program and continued to work confidently after the coach confirmed that their work was good and functioning correctly. P7, for example, paused her tutorial after each step to wait for the coach's confirmation before proceeding.

Notably, all coaches conducted this program without formal training in Scratch. C2 noted that in his first year, he *"didn't really know what Scratch was,"* and C4 acknowledged: *"I'm not very good myself. I only know basics in Scratch"*. C1 described the learning process as entirely improvised: *"we've just been learning as we go"*.

4.3 Creative Agency (RQ1)

4.3.1 ***Game Conceptualisation and Artefact Characteristics***. With long-term exposure and adaptive support from coaches, participants began to demonstrate creative agency in their authoring process. This was visible in how they conceptualised games, personalised assets in the games they created, developed ownership and pride in their work, and sought social validation from coaches and peers.

Participants expressed their game conceptualisation in distinct ways during the authoring process and game-model mapping activity. Most participants (n=6) described their process as starting with characters before mechanics. P4 and P7 consistently chose their game themes and environments first, with P4 noting he had a full concept in mind before beginning. However, P2 explained a mechanics-first approach: *"I just think about the mechanics, I guess, and, like, what kind of game we're going for."* P1 mentioned he needed interactive elements: *"whether it's a missile, whether it's a little dot, or it's something"* for him to consider something as a game. All participants derived their game concepts from games or films rather than generating original designs; P1 acknowledged: *"I'm basically listing what Crash Bandicoot is about"*.

During the game-model mapping activity, participants card selections varied widely. The number of cards selected ranged from 6 (P8) to 24 (P9) from a total of 131 cards, with only 2 participants (P6, P9) selecting across all 5 categories and suggesting their own cards. These self-suggested cards were all game system elements, such as score, lives, and speed. All participants selected a character card first. Four (P1, P2, P5, P8) did not select any ending cards. P7 selected the most control cards (6) with only 1 action card. Notably, 8 participants started describing their games from the main character's perspective. Participants conceptualised games with different complexities. Five participants (P1, P2, P4, P7, P10) used cards to discuss win conditions and action sequences, focusing on individual game elements rather than the relationships between them. For example, P7 explained that in her game the Mario character would win at the end and could climb a ladder and move left and right, but did not explain how these actions related to the winning condition (Figure 2). P8 focused on emotional and thematic elements rather than game rules and interpreted "hearts" as an emotional element saying: *"everyone loves it"* (Figure 3) while others who used the same card, considered it as *"lives"*, in their games. Two participants (P6, P9) described more quantified mechanic, such as coin-to-life exchange and enemy damage through lives reduction and game-over conditions. For example, P6 described a mechanic in which the player character could buy a life using 100 collected coins.

The Scratch artefacts also reflected these conceptualisation patterns. Across the three-month study, all participants created at least four games, totalling 41 projects. The largest category (n=14) consisted of interactive stories created without following a tutorial or reference, built around narrative and scene rather than formalised rules or mechanics (Table 3). Eight participants produced at least one project of this type, and all referred to them as games, reflecting a conceptualisation of games centred on narrative and scene rather than formal rules or win/lose conditions. Games created with tutorial assistance fell into six types: shooters (n=6), catch-and-collection (n=6), maze, pong and paddle-ball, flappy-style, and dodging games. Most participants (n=7) showed a preference for a specific game type, creating more than one game of the same type. For example, P1 created two shooter games out of his four total games. These tutorial-based games shared standardised structures: the space-bar was the universal action button, arrow keys controlled movement, and loss-based endings were triggered by collision. For example, in every shooter game, the player was positioned on the left with vertical-only movement as multiple enemies approached from the right. However, two participants incorporated mouse position to control player movement, though these were still based on tutorials. Win-based endings only appeared in four projects. Some participants (n=3) began incorporating new features such as timers, and scoring appeared in most games, usually incrementing by 1 point per event. Overall, creations varied considerably in complexity, from a three-block project producing a spinning, meowing character to a multi-sprite shooter incorporating cloning, user-defined variables, broadcast messaging, and difficulty progression.

4.3.2 ***Creative Agency through Personalisation and Ownership***. All participants personalised their games by modifying characters, backgrounds, and sounds while copying the code-blocks from previous projects or new tutorials. These personalisations varied substantially, from simple asset substitutions where participants (P3, P5) selected different characters and backgrounds while keeping the tutorial code-blocks, to more intentional choices (P9), such as changing single-coloured balls to multiple colours, and completely altering the game's visual theme. Some participants (n=2) expressed personal views through their game choices. P2 selected AI as his enemy character, symbolising this by using a tablet with a face, while P8 maintained a consistent K-pop demon hunters theme across all of her games. Eight participants deleted default sprites and replaced them with preferred characters at the start of every project. P1 described modifying existing game concepts creatively, taking the Tomb Raider character and asking *"why not let her shoot zombies?"*. Notably, personalisation occurred across participants with a wide range of support needs: P3, who closely followed tutorial instructions, still customised assets when using previous projects, while P5, who required the highest level of scaffolding, also wanted to change the falling balls to stars.



Figure 2: Set of cards used by P7 to map a game-model she intended to create in Scratch. As described by P7, the game includes two characters and a castle background as the setting. The game is initiated using the green flag. The Mario character can climb a ladder, with climbing triggered by pressing the spacebar. Both characters can move horizontally using the left and right arrow keys, and vertically using the up and down arrow keys.

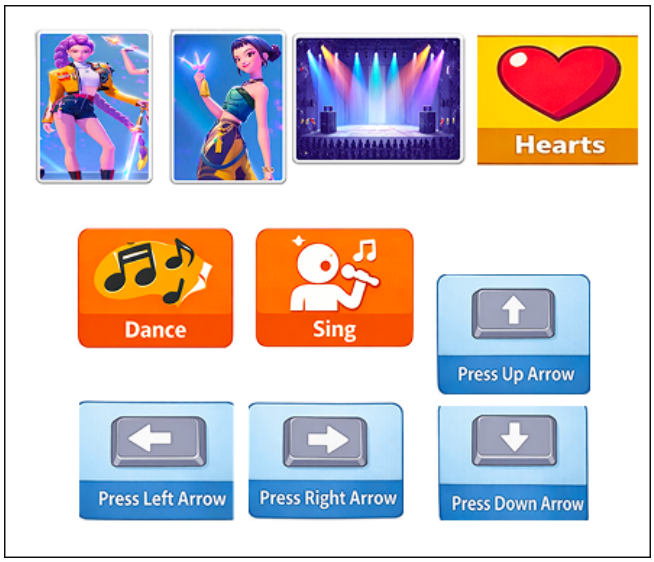


Figure 3: Set of cards used by P8 to map a game-model she intended to create in Scratch. As described by the participant, the game includes two characters performing on a stage, where they dance and sing. The characters can move horizontally using the left and right arrow keys, and vertically using the up and down arrow keys. A hearts element is included to represent that others love the characters.

Table 3: Types of games created by participants across two sites over three months

Participant ID / Types	Shooting	Maze	Catch Collection	Pong	Flappy-style	Dodging	Interactive stories	Total (Games)
P1	2	-	-	-	1	-	1	4
P2	1	-	-	1	2	-	-	4
P3	-	2	-	1	-	-	1	4
P4	1	-	2	-	-	-	2	5
P5	-	-	1	-	-	-	3	4
P6	-	2	-	-	-	-	1	3
P7	-	-	1	1	-	1	1	4
P8	-	-	-	-	-	1	4	5
P9	-	1	1	-	-	2	1	5
P10	2	-	1	1	-	-	-	4
Total (Game Types)	6	5	6	4	3	3	14	41

Sound was another significant area of personalisation. Eight participants used sounds in at least 2 of their games. Two participants (P1, P2) searched for music online because they could not find it in Scratch. P2 spent over 30 minutes selecting music by himself for a single game, uploading it and renaming it *"BEST GAME OVER THEME EVER"*. P9 adjusted sound frequencies to match different characters, and P3 combined over 20 background sounds into a music remixing section entirely on his own, even voice-recording himself for one of his maze games. However, some participants (n=2) used sound elements less consistently. P4 mentioned *"but*

sometimes I don't really use sounds," while in contrast, P2 noted *"I guess more music. I'd like to continue putting that in the games"*.

When intended mechanics could not be implemented, participants (n=4) adapted creatively rather than abandoning their work. P8 reframed an accidentally flipping image as a dancing mechanic, and P4 shifted from animating bat wings to adding sound effects when he encountered difficulties with implementing the movements. Some participants (n=3) accepted failing behaviours as features of their games. P6 noted *"as long as it works, I am fine with it"*. Coaches encouraged this pattern: *"their own concepts, but they're using basic tools to make it all work"*(C1).

Seven participants verbally claimed ownership of their games and asked coaches to playtest after completion to receive validation. Some even sought peer validation. For example, P7 asked a friend from another program to try her game during a break. P9 expressed excitement when he added a new lives system, mentioning that it made his game feel more complete. P7 exclaimed *"C2, I did it this time!"*, once she completed a game, and participants who fixed their own bugs showed their games to others, saying *"I did this by myself"*. Many (n=8) took pride in their work and were protective of their creative decisions, and preferred to keep their original ideas rather than incorporate suggested changes.

Coaches mentioned that this ownership was meaningful; C1 noted: *"it gives them a big sense of pride that they've completed something"*. Coaches aspired for participants to eventually create fully original games, with C1 expressing hope that one day it could be *"your own visual design and your own code"*. Yet C4 acknowledged uncertainty about whether participants would be capable of creating fully original games, noting *"whether or not they'll do that, I don't know"*. Playtesting provided social validation that reinforced creative engagement, with C2 noting that positive peer feedback *"makes them smile"*.

4.4 Barriers to Authorship (RQ2)

Despite the growing skills and creative agency demonstrated in some areas of the game authoring process, participants continued to face barriers that limited their authorship. These barriers stemmed mainly from limitations in the platform itself, including a lack of support for understanding code-block functions. These limitations were further influenced by the varying support needs across the participants. When encountering these challenges, participants demonstrated a variety of responses, including debugging strategies such as deleting and restarting code and disengagement.

The first barrier participants encountered was generating game ideas independently. Eight participants encountered difficulties in deciding what type of game to create without the coach input. Coaches responded by offering structured choices: *"I might throw two or three suggestions, and if they say, 'Yeah, let's do it,' we'll do it"*(C2). Participants identified idea generation as one of their main challenges; P1 noted: *"one is trying to think of any fresh ideas"*. Even when participants (P1, P2, P4, P7) had a broad theme in mind, the platform offered no structured way to convert it into a playable game concept. For example, P4 wanted an ocean-themed game with sharks, boats, and bats, but found it difficult to decide what the player was supposed to do. Four participants (P6, P7, P8, P9) changed game direction mid-development, without having settled on one concept from the start. Four participants (P2, P5, P7, P10) consistently waited for coaches to tell them what to make before beginning. C2 described this dependency: *"it's just a bit hard to come up with a game just like that"*.

The absence of debugging support within Scratch further limited participant authorship. Many (n=5) described their challenge as coding: *"mostly the code,"*(P2), which actually reflected difficulty debugging. P1 noted it as *"figuring out the coding or figuring out the problems"*. Coaches also reported that not all participants understood the logic: *"a few clients understand the logic and why we put the codes together"*(C2). This gap manifested with participants being aware that their games were not working as expected, but challenging to identify without support, where the error had occurred. Six participants encountered difficulties locating the specific code block causing an error, even when coaches attempted to guide them through debugging. For example, P6 had the game-over screen appear before the start of the game but kept making changes to the character sprites, instead of the game-over sprite. P9 knew his meteor sprite was not disappearing after being hit, but kept checking multiple blocks without being able to identify which specific block caused it. P2, after copying a game from a previous project, mentioned: *"I got all the codes correctly, I don't know why but this is not working"*.

This difficulty was partly rooted in how participants reused code-blocks. Four participants (P1, P2, P4, P6) when copying code-blocks from previous projects, sometimes mixed up the sequence of blocks or mismatched values to new characters. When participants attempted to diagnose errors, some (n=3) misattributed them to unrelated causes. P1 asked whether a movement error could be caused by image size rather than code logic, and P4 attempted to fix broken movement by changing character appearance. Beyond locating and diagnosing errors, five participants had difficulty explaining how their own implemented game mechanics worked. P1 pointed at characters and said *"he has to shoot to win"* without explaining how the scoring was determined, and P9 responded *"I don't know, this is my game, I made it with this tutorial"*.

These gaps in platform support resulted in participants developing multiple debugging strategies. Eight participants tested their games only after adding all the blocks when creating games using a previous project as reference. This resulted in further difficulties in identifying which code block had caused a problem. Upon encountering an error, 4 participants (P2, P4, P6, P8) deleted entire code blocks and restarted their work entirely. P2 removed blocks randomly from the main sequence, ending up deleting most of his code-blocks, and P8 deleted her work and restarted from the beginning. Even when the coaches encouraged collaborative debugging strategies or cross-checking references, some participants stopped working on the problem completely. P1 resisted comparing his code against reference codes, insisting, *"I have done it all correctly"*.

Another major platform constraint participants experienced was imported assets creating functional limitations: C2 noted *"when you get an image off Google, you can't make that character move or run,"*, and two participants (P4, P7) were unable to find assets in the Scratch library that matched their intended game concepts. Coaches further noted the lack of built-in tutorials: *"there's not so much on creating games. It's always we've gotta go to YouTube to do it"* (C2). C4 noted the need for audio-based guidance for non-readers

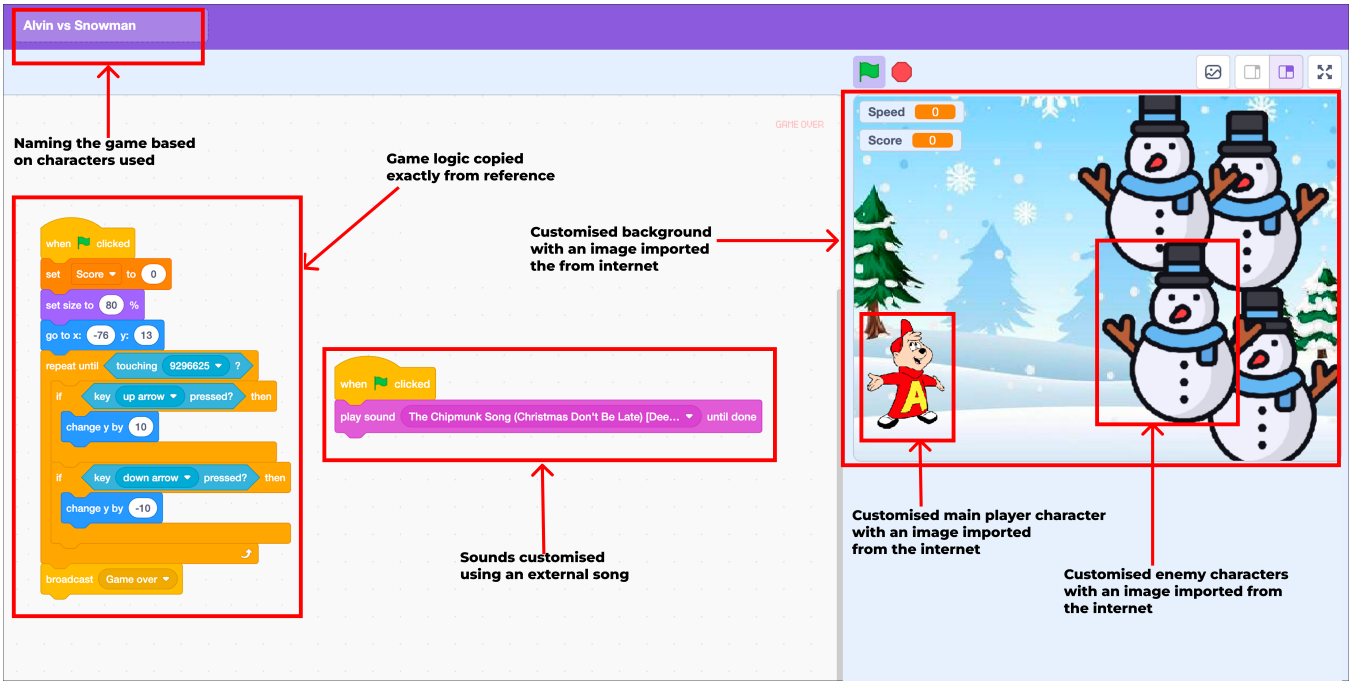


Figure 4: Annotated screenshot of a Scratch game created by a participant, highlighting key aspects of the authoring process. The game title is derived from the characters used. Core game logic is directly copied from a reference, while visual elements are customised through imported background and character images. Audio is also customised using an external song.

to follow in built tutorials: *"if you can't read, you can hear"*. Coaches also noted the need for a structured set of templates to learn gaming from start to finish within the platform. Participants also expressed openness to support through templates and examples saying *"the template, I guess, is good"* (P2)

When these challenges persisted, participants responded with frustration and disengagement. Five participants (P1, P2, P3, P6, P9) stepped away after three to four failed attempts to fix an issue, and five participants (P1, P2, P7, P8, P9) stopped working and turned to socialising or other activities when stuck. P1 described this as *"either I would try to fix it up myself... or I would just give up and just do something else,"* and P2 similarly noted: *"try to fix it. If it doesn't work, give up "*. C1 observed that participants *"tend to drop off quicker if it's too hard"*. In contrast, P4 demonstrated a persistent behaviour of working until the error was fixed.

However, when barriers were resolved, either on their own or with coach support, participants exhibited contrasting emotional responses; P2 described satisfaction noting *"okay, that's done, cool"* and P1 described emotional relief after completing a game.

5 Discussion

Our study provides empirical insights into how people with IDD engaged in game authoring using Scratch over a three-month period. All participants successfully built their own games, yet encountered challenges related to code understanding and idea generation. We discuss our findings in terms of authoring practices and challenges, addressing RQ1 and RQ2, then present design considerations to address RQ3.

5.1 Game Authoring Practices and Challenges

Our findings revealed that participants mainly used YouTube tutorials and existing projects as references when they creating games using Scratch. All participants used YouTube tutorials when attempting new game types because they offered step-by-step instructions that Scratch lacked. Over time, participants started expressing their creative agency by customising visual and audio assets, while copying reference code-blocks as they are, which indicated similarities to with prior work on neurotypical children [51, 65, 66]. For example, in the study [66] children replaced the default Flappy Bird characters and environments with themes from popular cultures, and updated sounds of the original games by selecting specific songs to match the themes [66]. The assets participants chose reflected personally meaningful creative choices, as most (n=8) drew on characters and backgrounds familiar to them rather than in-built platform assets, similar to prior work where neurotypical children used arcade style of games like Super Mario Bros [65]. Although only assets were modified, participants still claimed ownership over their games. Cheng et al. [16] noted that remixing can constrain the originality of games, yet for our participants, creative agency emerged through it. Additionally, our participants did not use any Scratch community projects when remixing, unlike prior studies on neurotypical children [51, 65] and did not share their work publicly either. Instead, participants sought recognition by asking coaches and peers to play-test their games, indicating a preference for sharing within familiar social circles. Additionally, our three-month study revealed participants' emerging skills in authoring over time,

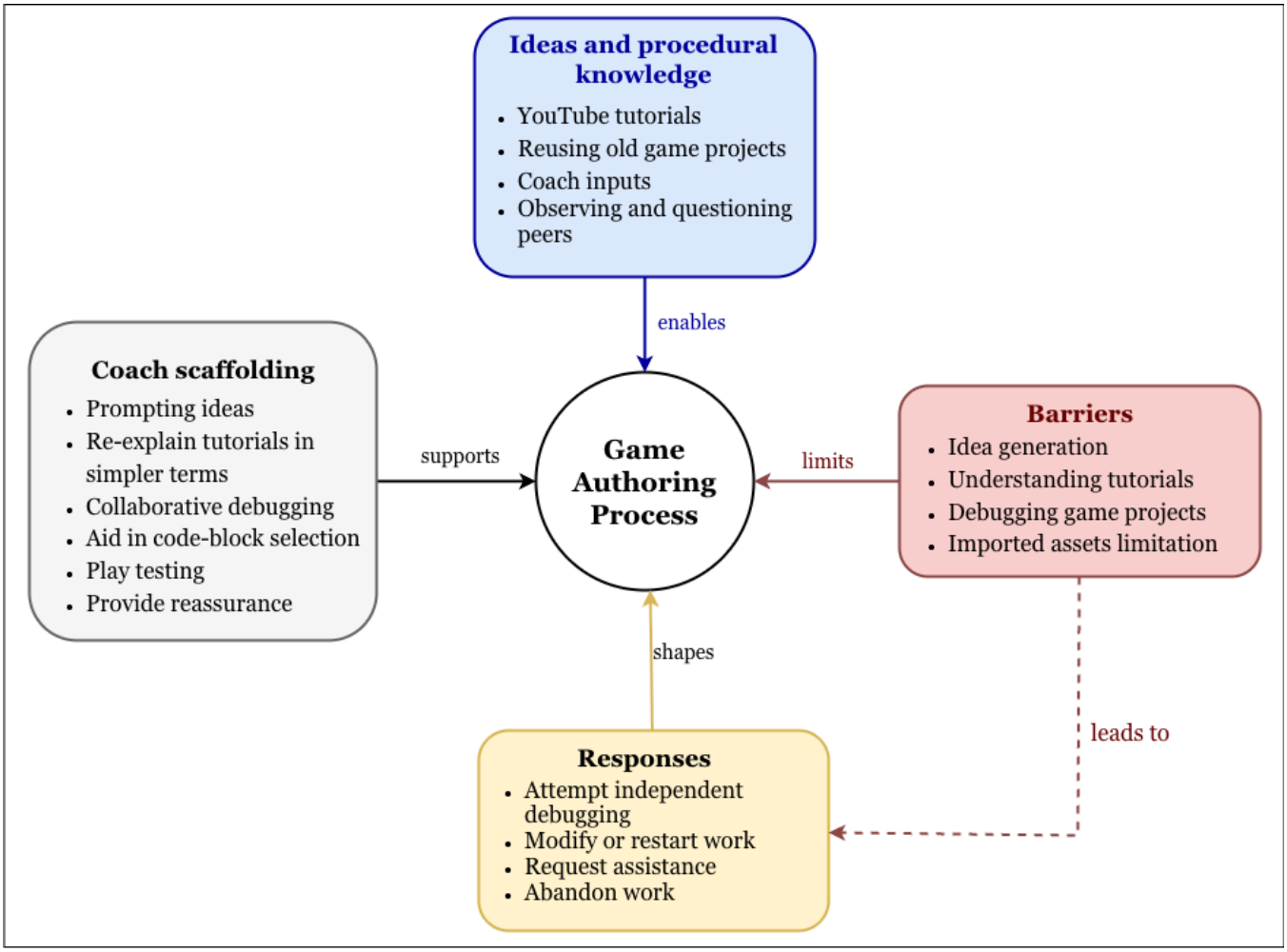


Figure 5: An overview of the game authoring process, showing how participants acquire ideas and procedural knowledge which enables authoring, coach scaffolding supports it, barriers limit it, and participant responses to those barriers shape its continuation

with most (n=6) progressing from relying on coaches to initiating tasks on their own.

Our findings also revealed how block recognition and understanding of game logics developed over time. Due to repetitive engagement, some participants (n=4) started recognising blocks using colour followed by text labels rather than functional meaning. Similarly, one participant who had reading difficulties also managed to identify the block category using colours, however, he had difficulty identifying a specific block within the category using text labels, echoing a previously documented barrier in Scratch for users with cognitive disabilities [71].

Coach dependency appeared most clearly across two recurring challenges: idea generation and debugging. Most participants (n=8) had difficulties generating game ideas at the start of sessions on their own, which resonates with challenges documented in prior work [19, 37]. A potential reason for this was the absence of a separation between design and build phases in Scratch, which directs users to the build editor without a structured design phase, leaving participants to make building decisions before establishing what they wanted to create. This also contributed to participants changing game direction mid-development, consistent with the sandbox nature of Scratch documented in prior work [71], which provides an open-ended environment with few restrictions on what users can create. When participants (n=7) encountered errors, participants first attempted to resolve those themselves by cross-referencing their code-blocks against prior working projects, but had difficulties identifying where the error occurred as their understanding of game logic was grounded entirely in observable outcomes rather than programming constructs. Coaches guided them through collaborative debugging and provided continuous reassurance that their work was on track; yet when coaches' support also failed to resolve the issue, participants stepped away from their work. This suggests that, without platform-level debugging support, error resolution depended entirely on coach availability and knowledge.

Unlike prior work where game authoring was scaffolded by researchers with domain expertise [30, 45, 51, 66], support coaches in our study had no formal programming experience yet carried the entire scaffolding responsibility, a gap that points to the need for structured support embedded within the authoring environment itself.

The design considerations follow draw on both authoring practices and challenges encountered by participants who had at least two months of prior Scratch experience. Given the diversity of language, reading, and support needs across our participant group (Table 1), they are intended to accommodate this range of needs.

5.2 Design Consideration 1: Structured Visual Planning for Idea Generation

Participants (n=8) had difficulties generating game ideas independently, as Scratch presents a building environment before users have established what they want to create. When they experienced difficulties in ideating what to build, participants (n=7) searched for tutorials using general terms like "Scratch Game", unless coach suggested a specific game type. However, during the game model mapping activity, participants were presented with 131 card choices across categories, including characters, backgrounds, objects, actions, controls, and endings, and they were able to ideate a game without any input from coaches. This illustrates that the barrier was generated from the absence of structured format to support idea generation. This is consistent with evidence that people with IDD have difficulty integrating information in open-ended tasks without external support, and that visual aids can substantially improve the quality of reasoning in this population [5].

ChatScratch [15] addressed a similar barrier by using an AI-augmented interactive storyboard that provided visual cues to help children aged 6–12 deconstruct their project into characters, scenes, and events, then automatically imported the created assets into Scratch. However, the selected assets appeared in the project without any associated code-blocks, leaving users with a blank editor, reintroducing the same open-ended barrier the storyboard aimed to resolve. The authors note that ChatScratch was not developed with game creation in mind. Prior work [29] similarly found that presenting activities through pictures and visual elements allows people with IDD to understand abstract concepts in concrete terms, supporting planning and sequencing before a task begins.

Block-based game authoring environments like Scratch for people with IDD should therefore consider incorporating a visual planning interface that appears prior to the build interface, presenting image-based suggestions paired with simple text labels across categories such as characters, backgrounds, and actions. Participants could then select and arrange their choices on a storyboard, which would then translate directly into a functional game template, ensuring users enter the editor with a workable starting point rather than a blank canvas.

5.3 Design Consideration 2: On Screen Step-by-Step In-Platform Guidance

All participants used YouTube tutorials at some point in their creation process because they offered step-by-step instructions that the authoring environment lacked. However, switching between the tutorial and the Scratch editor disrupted their workflow. These tutorials were also frequently too fast, too text-heavy, or used vocabulary that participants (n=4) found difficult to follow without support, requiring coaches to translate content into simpler terms. Participants (n=4) often mismatched values or misarranged sequences while copying code-blocks and needed assistance from coaches to fix their problems. Some participants (n=5) even sought continuous confirmation from coaches that their work was progressing correctly before they could proceed.

Prior work with autistic children [18] has found that dependence on human prompting can be reduced by embedding structured guidance directly within a tool, thereby promoting independence. Similarly, for people with IDD, research has found that effective technological support can be provided through embedded step-by-step prompting combined with multimodal instruction for each step, presented immediately after the previous step is completed [41].

Prior work [46] also indicates audio support can further enhance the platform's usability as it can help non-visual readers understand and navigate block-based code without relying solely on visual interpretation, a consideration particularly relevant for the people with IDD who require reading assistance.

Therefore, to support more self-directed game authoring for this population, full game templates with built-in step-by-step instructions that indicate which block to use, read the instructions aloud, and show where to place it, with each completed stage automatically verified before proceeding, should be considered within block-based authoring platforms such as Scratch.

5.4 Design Consideration 3: Personalising Learning Pathways with Incremental Complexity

Personalisation of assets was the primary mode of authorship among our participants, with all participants customising characters and sounds rather than writing original code. In contrast to the Use-Modify-Create framework [42] that positions "Modify" as the transitioning step towards independent creation, our findings suggest that for people with IDD, it may instead be a sustained and complete mode of engagement, consistent with prior work discussing personalization as a primary driver of creative participation [9, 28]. Furthermore, our participants did not use any Scratch community projects as starting points for personalisation, as they found it difficult to locate which blocks to modify. Barrett and Israel [7] propose pre-scaffolded template versions as one of multiple entry points that allow users to explore and develop skills that match their current level, increasing autonomy over learning.

Progressive disclosure [67], where complexity is revealed gradually so users can master one step before the next is introduced rather than presenting all options at once, should be considered when designing block-based authoring environments for people with IDD. Therefore, block-based game authoring platforms such as Scratch should consider including a library of genre-varied game templates, where each template first providing options to personalise characters, backgrounds, and sounds as the primary and default authorship layer, then offering modification opportunities such as changes to

movement and speed, and progressively introduces more complex logic for those who wish to explore further. Personalisation thus remains the main mode of authorship, while deeper engagement remains available to those who are ready for it.

5.5 Design Consideration 4: Intent Compatibility of Imported Assets

Most participants (n=7) used character and background images imported from the internet in their games. However, as these were imported as static sprites with a single costume, they could not be animated, disrupting their creative intent during development and forcing unplanned compromises. Producing animation in Scratch requires users to manually draw multiple costume variations using the paint editor and connect code-blocks to cycle through them, which requires additional technical knowledge beyond what is typically needed for basic game authoring.

Since participants based their games on characters and references from films and games they already knew, the availability of personally relevant assets was central to how they exercised authorship. ChatScratch [15] addresses a related barrier by integrating AI image generation, allowing users to draw or describe assets that are then generated and imported into the editor. However, this does not address the problem participants encountered: they wanted to animate characters they had already found, not create new ones.

Therefore, block-based game authoring platforms such as Scratch should consider facilitating an option to automatically convert any imported static images into multiple costume variations similar to the variations already available for built-in sprites.

5.6 Design Consideration 5: Structured Intention-Driven Debugging Support

Debugging was a prominent challenge among our participants, as most (n=6) had difficulty specifying the block that triggered the game to malfunction. Though they understood their game was not functioning as intended, they had difficulty determining which code-block caused such behaviour because they explained their game logic, through game visuals rather than how the blocks functioned, indicating that block-logic comprehension for this population is grounded in observable outcomes rather than conceptual understanding. This aligns with challenges documented among novice creators more broadly, where in block-based programming, locating bugs is substantially harder than fixing them, reflecting difficulties to bridge between observed behaviour and the corresponding code [38].

NuzzleBug [22] was developed as an extension for Scratch to improve debugging effectiveness by presenting questions like "why a sprite did not move" or "why a condition did not trigger" when users click on a specific sprite in their games, and provides visual answers that direct them to the responsible block. However, it requires users to first form a hypothesis about what went wrong in order to select the correct question. However it is noted, that even when NuzzleBug provides correct answers, learners may require further help to comprehend faults and navigate to the necessary fix [22].

For people with IDD, debugging support should remain grounded in visible game behaviour throughout, avoiding any requirement for users to reason about code directly. To support this, block-based game authoring platforms like Scratch should consider using a custom tracer similar to Nuzzlebug to first identify the context of the game using blocks used and sprites present. The environment could then prompt visual options representing different possible game behaviours based on the blocks used in the created game. Users could select the option that matches their intention. The environment would then provide multimodal guidance while highlighting the specific blocks to be modified one at a time, guiding them to the corresponding code change without requiring independent navigation of the editor.

6 Limitations and Future Work

Our study revealed that while Scratch supported the game creation process for participants, it presented considerable challenges related to usability and the lack of structured scaffolding. Several limitations should be noted when interpreting our findings. All participants had been using Scratch for at least two months prior to the study, meaning we were unable to examine how a person with IDD would engage with game authoring as a first-time Scratch user. The game-model mapping cards were drawn from prior games participants had made, as well as preferences mentioned during interviews and program sessions, so this familiarity may have influenced certain card combinations participants mapped. Coaches were also present throughout the authoring process, meaning our findings reflect this specific support context and may not extend to how participants would engage with game authoring by themselves or under other support arrangements. Additionally, participants worked independently rather than collaboratively, leaving open the question of how peer interaction might shape the authoring process. Finally, screen recording was not possible due to privacy constraints around the use of organisational devices, which limited our ability to capture a complete record of participants' authoring activity on their computer. A promising direction for future work is to involve people with IDD more directly in the design of block-based authoring tools like Scratch, through participatory methods that accommodate diverse scaffolding needs. Approaches such as co-design workshops, scaffolded choice-making, and guided prototyping could enable participants to inform key design decisions such as the categories within the visual planning interface, the types of game templates offered, and the complexity of step-by-step instructions. Following participatory design efforts, longer deployments will be needed to evaluate how the proposed features perform in practice and how participants' familiarity and confidence develop over time. Examining engagement across different settings and support contexts will further help ensure that authoring tools are genuinely tailored to the needs of this population.

7 Conclusion

In this paper, we presented the game authoring practices of people with IDD using Scratch, over a three-month period, as part of a GameCraft program at a disability services organisation. Through semi-structured interviews, a game-model mapping activity, observation data, and Scratch game artefacts produced by participants, we explored how people with IDD engage in game authoring and what types of games they produce (RQ1), what challenges

they encounter and what enablers and barriers shape their engagement (RQ2), and what design considerations can be proposed to better support game authoring in Scratch for this population (RQ3). In response to RQ1, our findings show that participants used YouTube tutorials and existing projects to acquire ideas and procedural knowledge. Over time, participants relied less on coaches, with most progressing to finding their own resources. Participants also developed their understanding of game logic based on visible game behaviour. This authoring process also enabled participants to express creative agency, develop their skills as authors, and experience a sense of ownership, pride, and social recognition in their work. Regarding RQ2, our findings revealed that participants encountered challenges in two main areas: idea generation and debugging. At the start of the authoring process, participants often had difficulty deciding what type of game to create and which mechanics to include, and they relied on suggestions from coaches or YouTube tutorials. Similarly, during debugging, participants often had difficulty locating the specific blocks that caused their games not to function as intended. Beyond these challenges, coach mediated support was essential in further enabling the authoring process. Coaches played multiple roles, including supporting idea generation, interpreting tutorial content in simple language, providing reassurance, and guiding debugging. Regarding RQ3, we derived design considerations that could be applied to game authoring platform like Scratch to better enable the authoring process for people with IDD. These include: structured visual planning interfaces, embedded step-by-step in-platform guidance, personalised learning pathways with incremental complexity, automatic animation compatibility for imported assets, and intention-driven debugging support grounded in visible game behaviour. As one of the first studies to examine digital game authoring by adults with IDD, this work extends human-computer interaction research on inclusive participation by repositioning people with IDD as active authors of digital games rather than passive players or end users. We call for the development of inclusive authoring platforms that not only accommodate but actively strengthen the creativity and independence of people with IDD.

Acknowledgments

We thank the participants and coaches from our disability service partner for their time and insights. This research was supported by the Australian Government through the Australian Research Council (Project LP210300465) and by Monash University Faculty of Information Technology International Postgraduate Research Scholarship. Claude was used for checking grammar, clarity, sentence structure refinement and proof-reading.

References

[1] Mete Akcaoglu and Matthew J Koehler. 2014. Cognitive outcomes from the Game-Design and Learning (GDL) after-school program. *Computers & Education* 75 (2014), 72–81.

[2] Lisa Anthony, Sapna Prasad, Amy Hurst, and Ravi Kuber. 2012. A participatory design workshop on accessible apps and games with students with learning differences. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 253–254.

[3] Cecilia R Aragon, Sarah S Poon, Andrés Monroy-Hernández, and Diana Aragon. 2009. A tale of two online communities: Fostering collaboration and creativity in scientists and children. In *Proceedings of the seventh ACM conference on Creativity and cognition*. 9–18.

[4] Abbas Attarwala. 2024. Engaging Middle Schoolers in Game Programming: A Scratch-Based Workshop Experience. *Journal of Computing Sciences in Colleges* 39, 6 (2024), 70–78.

[5] Rebecca Bailey, Paul Willner, and Simon Dymond. 2011. A visual aid to decision-making for people with intellectual disabilities. *Research in developmental disabilities* 32, 1 (2011), 37–46.

[6] Jackie Barnes, Amy K Hoover, Borna Fatehi, Jesus Moreno-Leon, Gillian Smith, and Casper Harteveld. 2017. Exploring emerging design patterns in student-made climate change games. In *Proceedings of the 12th international conference on the foundations of digital games*. 1–6.

[7] Joanne Barrett and Maya Israel. 2023. Scaffolding block coding through multiple entry points. In *Proceedings of the 18th WiPSCE conference on primary and secondary computing education research*. 1–2.

[8] Ahmet Baytak and Susan M Land. 2010. A case study of educational game design by kids and for kids. *Procedia-Social and Behavioral Sciences* 2, 2 (2010), 5242–5246.

[9] Filip Bircanin, Laurianne Sitbon, Maria Hoogstrate, Ahmed K Abbas, Alieh Hajizadeh Saffar, and Margot Brereton. 2025. Beyond the Buckets of Support: Designing for Agency and Interaction in Personalised Disability Systems. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–19.

[10] Núria Bonet-Codina, Ariel von Barnekow, and Dani Tost. 2015. IntegraGame: a real-life inspired serious game for social and professional training of people with intellectual disability. In *Proceedings of the 3rd 2015 Workshop on ICTs for improving Patients Rehabilitation Research Techniques*. 10–13.

[11] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative research in sport, exercise and health* 11, 4 (2019), 589–597.

[12] Virginia Braun and Victoria Clarke. 2025. Reporting guidelines for qualitative research: A values-based approach. *Qualitative Research in Psychology* 22, 2 (2025), 399–438.

[13] Nuno Castelhano and Licínio Roque. 2017. Lessons from designing a game to support playfulness in multisensory stimulation environments. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. 57–68.

[14] Yao-Jen Chang, Ya-Shu Kang, Yao-Sheng Chang, Hung-Huan Liu, Yu-Ling Chiu, and Chia Chun Kao. 2016. Designing a Kinect2Scratch game to help teachers train children with intellectual disabilities for pedestrian safety. In *Proceedings of the 18th international ACM SIGACCESS conference on computers and accessibility*. 269–270.

[15] Liuqing Chen, Shuhong Xiao, Yunnong Chen, Yaxuan Song, Ruoyu Wu, and Lingyun Sun. 2024. ChatScratch: An AI-augmented system toward autonomous visual programming learning for children aged 6-12. In *Proceedings of the 2024 CHI conference on human factors in computing systems*. 1–19.

[16] Ruijia Cheng, Sayamindu Dasgupta, and Benjamin Mako Hill. 2022. How interest-driven content creation shapes opportunities for informal learning in Scratch: A case study on novices' use of data structures. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[17] Samantha Conde. 2020. Establishing a serious game on relationship boundaries for people with developmental disabilities. In *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. 1–3.

[18] Meg Cramer, Sen H Hirano, Monica Tentori, Michael T Yeganyan, and Gillian R Hayes. 2011. Classroom-based assistive technology: collective use of interactive visual schedules by students with autism.. In *CHI*, Vol. 11. 1–10.

[19] Sandra Danilovic, Kenny Chee, and Michelle Skop. 2024. Playful resilience: empowering recovery through autobiographical game-based storytelling in the opioid epidemic. *Proceedings of the ACM on Human-Computer Interaction* 8, CHI PLAY (2024), 1–35.

[20] Elaine Cristina Juvino De Araújo and Wilkerson L Andrade. 2021. A systematic literature review on teaching programming to people with cognitive disabilities. In *2021 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–8.

[21] Elena de la Guía, María Lozano, and Victor R Penichet. 2013. TrainAb: a solution based on tangible and distributed user interfaces to improve cognitive disabilities. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 3039–3042.

[22] Adina Deiner and Gordon Fraser. 2024. NuzzleBug: Debugging block-based programs in scratch. In *Proceedings of the 46th IEEE/ACM International Conference on Software Engineering*. 1–13.

[23] Rayshun Dorsey and Ayanna M Howard. 2011. Examining the effects of technology-based learning on children with autism: a case study. In *2011 IEEE 11th International Conference on Advanced Learning Technologies*. IEEE, 260–261.

[24] Fifth Edition et al. 2013. Diagnostic and statistical manual of mental disorders. *Am Psychiatric Assoc* 21, 21 (2013), 591–643.

[25] Sara Fiori, Glenn Strong, and Jonathan Dukes. 2025. How Are We Teaching Programming to Students with Intellectual Disabilities? A Systematic Review of the Literature. In *Proceedings of the 25th Koli Calling International Conference on Computing Education Research*. 1–12.

[26] Yulia Gizatdinova, Vera Remizova, Antti Sand, Sumita Sharma, Kati Rantanen, Terhi Helminen, and Anneli Kylliäinen. 2022. PigScape: An embodied video game for cognitive peer-training of impulse and behavior control in children with ADHD. In *Proceedings of the 24th International ACM SIGACCESS Conference*

on Computers and Accessibility. 1–4.

[27] Marta Goyena, Matteo Dal Magro, Martina Merolli, David Barbero García, Carlos Cortés, Marta Orduna, Ainhoa Fernández-Alcaide, María Nava-Ruiz, Jesus Gutiérrez, Pablo Perez, et al. 2025. Immersive cognitive training for job integration with people with intellectual disabilities. In *Proceedings of the 2025 ACM International Conference on Interactive Media Experiences*. 346–349.

[28] Leandro S Guedes, Jacquie Johnstone, Kirsten Ellis, and Monica Landoni. 2024. Creative technologies in action: Empowering individuals with intellectual disabilities. In *International Conference on Computers Helping People with Special Needs*. Springer, 193–200.

[29] Gillian R Hayes, Sen Hirano, Gabriela Marcu, Mohamad Monibi, David H Nguyen, and Michael Yeganyan. 2010. Interactive visual supports for children with autism. *Personal and ubiquitous computing* 14, 7 (2010), 663–680.

[30] Amy K Hoover, Jackie Barnes, Borna Fatehi, Jesús Moreno-León, Gillian Puttick, Eli Tucker-Raymond, and Casper Harteveld. 2016. Assessing computational thinking in students' game designs. In *Proceedings of the 2016 annual symposium on computer-human interaction in play companion extended abstracts*. 173–179.

[31] Aarij Mahmood Hussaan, Karim Sehaba, and Alain Mille. 2011. Helping children with cognitive disabilities through serious games: project CLES. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. 251–252.

[32] Yavuz Inal and Nergiz Cagiltay. 2024. Empathy development in digital accessibility through real-life practices in a programming course: a case study. In *Proceedings of the 13th Nordic Conference on Human-Computer Interaction*. 1–7.

[33] Maya Israel, Quentin M Wherfel, Jamie Pearson, Saadeddine Shehab, and Tanya Tapia. 2015. Empowering K–12 students with disabilities to learn computational thinking and computer programming. *TEACHING Exceptional Children* 48, 1 (2015), 45–53.

[34] Javier Jiménez Honrado, Gabriel Sebastián Rivera, Jose A Gallud, and Victor M R. Penichet. 2023. Serious games to facilitate social integration for children with autism spectrum disorder. In *Proceedings of the XXIII International Conference on Human Computer Interaction*. 1–8.

[35] Yasmin B. Kafai and Quinn Burke. 2015. Constructionist Gaming: Understanding the Benefits of Making Games for Learning. *Educational Psychologist* 50, 4 (2015), 313–334. PMID: 27019536. arXiv:https://doi.org/10.1080/00461520.2015.1124022 doi:10.1080/00461520.2015.1124022

[36] Yasmin B Kafai and Quinn Burke. 2015. Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist* 50, 4 (2015), 313–334.

[37] Yasmin B Kafai, Quinn Burke, and Chad Mote. 2012. What makes competitions fun to participate? The role of audience for middle school game designers. In *Proceedings of the 11th international conference on interaction design and children*. 284–287.

[38] ChanMin Kim, Jiangmei Yuan, Lucas Vasconcelos, Minyoung Shin, and Roger B Hill. 2018. Debugging during block-based programming. *Instructional Science* 46, 5 (2018), 767–787.

[39] MiJeong Kim, JaMee Kim, and WonGyu Lee. 2025. Intellectual disabilities and programming: Improving computational thinking-based problem solving. *Education and Information Technologies* 30, 9 (2025), 12101–12141.

[40] Maren Krafft, Gordon Fraser, and Neil Walkinshaw. 2020. Motivating adult learners by introducing programming concepts with scratch. In *Proceedings of the 4th European Conference on Software Engineering Education*. 22–26.

[41] Giulio E Lancioni, Nirbhay N Singh, Mark F O'Reilly, and Jeff Sigafoos. 2025. Possible assistive technology solutions for people with moderate to severe/profound intellectual and multiple disabilities: considerations on their function and long-term role. *International Journal of Developmental Disabilities* 71, 7 (2025), 1110–1116.

[42] Irene Lee, Fred Martin, Jill Denner, Bob Coulter, Walter Allan, Jeri Erickson, Joyce Malyn-Smith, and Linda Werner. 2011. Computational thinking for youth in practice. ACM Inroads, 2 (1), 32–37. *DOI: https://doi. org/10.1145* 1929887 (2011).

[43] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. 2010. The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)* 10, 4 (2010), 1–15.

[44] John H Maloney, Kylie Peppler, Yasmin Kafai, Mitchel Resnick, and Natalie Rusk. 2008. Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. 367–371.

[45] Roger Meintjes, Thembelani Makapela, Micyla Hobbs, and Nandipha Jantjies. 2023. Child-to-Child Public Health Messaging through a Portable Craft Tech Interactive in Rural South Africa. In *Proceedings of the 22nd Annual ACM Interaction Design and Children Conference*. 210–221.

[46] Aboubakar Mountapmbeme, Obianuju Okafor, and Stephanie Ludi. 2022. Accessible blockly: An accessible block-based programming library for people with visual impairments. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–15.

[47] Samiullah Paracha, Gill Hagan-Green, Lynne Hall, and Kate Macfarlane. 2022. Special iApps: Play-based Learning for Children with Severe Intellectual Disabilities. In *Proceedings of the 10th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*. 100–106.

[48] Juhyeong Park, Sunok Lee, and Sangsu Lee. 2021. MODU: a story-based empathy expression training game for children with mild intellectual disability and borderline intellectual functioning. In *Proceedings of the 20th Annual ACM Interaction Design and Children Conference*. 615–619.

[49] Rusyaizila Ramli and Kristo Radion Purba. 2023. The development of a game for cognitive remediation therapy (CRT) to improve attention span and memory among children with learning disabilities. In *2023 19th IEEE International Colloquium on Signal Processing & Its Applications (CSPA)*. IEEE, 1–6.

[50] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.

[51] Gabriela T Richard and Yasmin B Kafai. 2015. Making physical and digital games with e-textiles: a workshop for youth making responsive wearable games and controllers. In *Proceedings of the 14th international conference on interaction design and children*. 399–402.

[52] Judy Robertson and Cathrin Howells. 2008. Computer game design: Opportunities for successful learning. *Computers & Education* 50, 2 (2008), 559–578.

[53] Robert L Schalock, Sharon A Borthwick-Duffy, Valerie J Bradley, Wil HE Buntinx, David L Coulter, Ellis M Craig, Sharon C Gomez, Yves Lachapelle, Ruth Luckasson, Alya Reeve, et al. 2010. *Intellectual disability: Definition, classification, and systems of supports*. ERIC.

[54] Robert L Schalock, Ruth Luckasson, and Marc J Tassé. 2019. The contemporary view of intellectual and developmental disabilities: Implications for psychologists. *Psicothema* 31, 3 (2019), 223–228.

[55] Kshitij Sharma, Sofia Papavlasopoulou, and Michail Giannakos. 2021. Faces don't lie: Analysis of children's facial expressions during collaborative coding. In *FabLearn Europe/MakeEd 2021-An International Conference on Computing, Design and Making in Education*. 1–10.

[56] Kshitij Sharma, Sofia Papavlasopoulou, Serena Lee-Cultura, and Michail Giannakos. 2021. Information flow and children's emotions during collaborative coding: A causal analysis. In *Proceedings of the 20th annual ACM Interaction Design and Children Conference*. 350–362.

[57] Abha Shree and Prakash C Shukla. 2016. Intellectual Disability: Definition, classification, causes and characteristics. *Learning Community-An International Journal of Educational and Social Development* 7, 1 (2016), 9–20.

[58] Melinda R Snodgrass, Maya Israel, and George C Reese. 2016. Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education* 100 (2016), 1–17.

[59] Anna Sochocka, Jakub Mirocha, and Rafał Starypan. 2020. Serious games as an aid in the development of people with intellectual disabilities. *Bio-Algorithms and Med-Systems* 16, 1 (2020), 20190055.

[60] Carla Sousa. 2020. Empowerment and ownership in intellectual disability gaming. *International Journal of Film and Media Arts* 5, 1 (2020), 14–23.

[61] CARLA SOUSA, José Carlos Neves, and Manuel Damásio. 2021. Research in serious games for people with intellectual disability: A meta-analysis study. In *13th International Conference on disability, virtual reality and associated technologies*. Universidade Lusófona, p–43.

[62] Katie Salen Tekinbas and Eric Zimmerman. 2003. *Rules of play: Game design fundamentals*. MIT press.

[63] Juan C Torrado, Ida Wold, Letizia Jaccheri, Susanna Pelagatti, Stefano Chessa, Javier Gomez, Gunnar Hartvigsen, and Henriette Michalsen. 2020. Developing software for motivating individuals with intellectual disabilities to do outdoor physical activity. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*. 81–84.

[64] Javier Torrente, Ángel Del Blanco, Pablo Moreno-Ger, and Baltasar Fernández-Manjón. 2012. Designing serious games for adult students with cognitive disabilities. In *International Conference on Neural Information Processing*. Springer, 603–610.

[65] Giovanni Maria Troiano, Dylan Schouten, Michael Cassidy, Eli Tucker-Raymond, Gillian Puttick, and Casper Harteveld. 2020. Ice paddles, CO2 invaders, and exploding planets: how young students transform climate science into serious games. In *Proceedings of the annual symposium on computer-human interaction in play*. 534–548.

[66] Veena Vasudevan, Yasmin Kafai, and Lei Yang. 2015. Make, wear, play: remix designs of wearable controllers for scratch games by middle school youth. In *Proceedings of the 14th international conference on interaction design and children*. 339–342.

[67] Katherine Vergara and Valeria Herskovic. 2025. Boxy Board: Supporting Novice Learners Through a Progressive Disclosure Approach for Physical Computing. *International Journal of Human–Computer Interaction* (2025), 1–22.

[68] Nienke Vos, Henny Van Der Meijden, and Eddie Denessen. 2011. Effects of constructing versus playing an educational game on student motivation and deep learning strategy use. *Computers & education* 56, 1 (2011), 127–137.

[69] Ben Wasserman, Derek Prate, Bryce Purnell, Alex Muse, Kaitlyn Abdo, Kendra Day, and LouAnne Boyd. 2019. Vrsensory: designing inclusive virtual games

with neurodiverse children. In *Extended abstracts of the annual symposium on computer-human interaction in play companion extended abstracts*. 755–761.

[70] Aman Yadav, Maya Israel, Emily Bouck, Alexis Cobo, and John Samuels. 2022. Achieving CSforall: Preparing special education pre-service teachers to bring computing to students with disabilities. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education-Volume 1*. 196–201.

[71] Misbahu S Zubair, David Brown, Thomas Hughes-Roberts, and Matthew Bates. 2018. Evaluating the accessibility of scratch for children with cognitive impairments. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 660–676.

[72] Misbahu S Zubair, David J Brown, Thomas Hughes-Roberts, and Matthew Bates. 2023. Designing accessible visual programming tools for children with autism spectrum condition. *Universal Access in the Information Society* 22, 2 (2023), 277–296.