Chapter 3

Game Accessibility

Getting Started

Thomas Westin, Ian Hamilton and Barrie Ellis

CONTENTS

3.1 Introduction 37
  3.1.1 Brief History of Game Accessibility and Guidelines 38
  3.1.2 Game Accessibility Guidelines 42
  3.1.3 Applying Game Accessibility Guidelines 42

3.2 Example 43
  3.2.1 Barriers 43
  3.2.2 Working With Guidelines 44
  3.2.3 Summary 48

3.3 Concluding Remarks 49
References 50

3.1 INTRODUCTION

Disability occurs through mismatched interactions, between a person’s abilities and the requirements of their environment. This can be particularly vexing when barriers present difficulty performing day-to-day tasks. This is known as the social model of disability. The social model was developed in the 1970s by disabled people to replace the medical model of disability, which views disability as a personal attribute.

Barriers do not only cause difficulty through interaction with permanent physical conditions; they can also be temporary (e.g. a broken arm, a headache), situational (e.g. playing in bright sunlight, while holding a baby or a beer), or have simple differences. Barriers in games are put there by
designers and developers, often unknowingly and often unnecessarily. Accessibility is the process of avoiding or removing these barriers.

There are two main approaches to accessibility in general: inclusive design is a pragmatic approach that considers what is possible in practice based on available resources and other limitations, as opposed to universal design (Benyon, 2019). An important way in which games differ significantly from other industries is that some barriers are desirable and necessary to give the game its rules and challenge, and without rules and challenge, the experience would not be a game. Which barriers are necessary is specific to each individual game. This makes game accessibility an optimisation process, identifying and avoiding or removing as many unnecessary barriers as possible while maintaining an enjoyable experience.

By being aware of design alternatives and knowing how to apply them, designers can make choices for inclusive game design and accessibility early on in the design process. This is essential. Retrofitting can be much more expensive and difficult, reducing both the scope of what can be considered and the effectiveness of solutions. This chapter aims to help game designers get started. However, this chapter would not be possible to write without the work of many people over many years, so we will begin with a historical perspective. After this, an introduction to current guidelines is presented followed by guidance of how to work with them.

3.1.1 Brief History of Game Accessibility and Guidelines

Esther played against Tom, but the two of them were so clumsy, and the speed of the ball so relatively fast, that they had no chance of stopping it, or of scoring points against each other except by chance.


This passage, taken from the novel Skallagrigg (Horwood, 1987), imagines a game of Pong between two disabled characters. Esther with cerebral palsy and Tom with Down syndrome. Both were fascinated by ‘television games’, but swiftly frustrated by the barriers. This experience mirrors the on-going struggle of disabled gamers wherever gaming technology offers few concessions to access needs. Happily, people (like you?) have long sought to find and share ways to remove barriers to enjoyable play.

In the 1970s, blind computer operators played text games such as Lunar Lander with touch-typed input and an Optacon hand-held scanner for
reading (Wikipedia, 2020). This device converted computer printouts to a jostling array of raised pins possible to read with a finger. From the 1980s onwards, blind gamers would more commonly use home micros and external speech synthesisers to play the likes of BBC Micro Football Manager. Some mainstream games became blind accessible unintentionally through sound alone by players such as Jordan Verner, completing N64 Zelda Ocarina of Time using in-game sound and spoken help guides. Equally impressive was blind gamer Brice Mellen defeating Ed Boon at his own game, Mortal Kombat.

Throughout this period to current times, a small sub-sector of both hobbyist and professional audiogames have arisen: games written for (and often by) blind people designed to be played through audio alone. Some of these games aimed to be a rich experience for sighted and unsighted alike to bring players together, such as the Taito supported ‘Space Invaders for Blind’ in 2003.

The advent of iOS in the late 2000s brought with it a significant boost in accidentally blind-accessible games due to native apps’ compatibility by default with system level text to speech, as seen in Hanging with Friends (Frum, 2011). At time of writing the industry is going through another fundamental shift. Software engines have always posed barriers via the incompatibility of their output with text to speech software. However, both Unity and Unreal have announced their work to solve this. Mass market games from Minecraft to The Division 2 are implementing voiced menus, driven in large part by CVAA legislation. (FCC, n.d.). Over and above this, AAA games such as Madden and Killer Instinct have put significant effort into blind-accessible gameplay.

Deaf people rarely found themselves completely shut out of video games in the early days. Arcade games such as Williams’ 1980 classic Defender disadvantaged players if they were unable to hear panicked sounds of human abduction, but few games were unplayable. That changed in the 1990s with CD-ROMs and full motion video. Games with spoken dialogue without subtitles often became impossible to follow. Games such as Zork: The Grand Inquisitor put this right with a subtitles option. Subtitling reached a tipping point in 2008, when the first Assassin’s Creed game launched with none at all, to some public uproar. As a result, subtitling in-game dialogue became a publisher level requirement for all Ubisoft games, which in turn was an important step in pushing adoption across the wider industry. Despite this, there is still much room for improvement in their structure,
comprehensiveness and presentation. Games also took a greater interest in conveying sound effects and mood music, critical for building atmosphere and giving people a fuller experience. To this day, games rarely have captions or visual equivalents for important non-speech audio, locking deaf or hard of hearing players out from a fuller experience. One notable early exception is Valve’s captioning for sound effects and important musical cues in *Half Life 2* and *Portal*.

Physically and learning-disabled players have often benefited from standard features, such as remapping and difficulty options. The 1972 Odyssey Magnavox featured a speed control for all ball games to slow play to a crawl, hugely beneficial for motor, sight and cognitive accessibility. Atari, at their peak, used what was almost a standard connection method for joysticks, leading to a massive range of controllers shared across many systems. Atari also included a ‘Special Feature’ option in some of their most successful VCS console games such as *Missile Command* making play far easier. This accessibility feature was promoted on their boxes allowing people to know about it prior to purchase. Upfront communication of accessibility functionality is something still rare to find today. But although these mass market considerations are also beneficial for accessibility, often people need more, particularly with the increasing motor and cognitive demands that games and standard controllers have presented over the years.

Where more was needed, people have often turned to customising and scratch building controllers and software or asking for accessibility updates. Ken Yankelevitz (Figures 3.1 and 3.2), Brilliant Computing,

![FIGURE 3.1 Two children playing Bowling using a fingertip controller (left player) and mouth control (right player). (Photo via Lynn Given [née Yankelevitz]. With permission.)](image-url)
OneSwitch.org.uk, Evil Controllers and many others have long served physically disabled gamers with custom controllers. Some large gaming companies such as Nintendo, Namco and most recently Microsoft with the Xbox Adaptive Controller have also delved into this world.

Historically, players have often had to resort to many methods of ‘cheating’ and hacking to open up access to otherwise unplayable games, including those lacking meaningful difficulty adjustments. This has steadily become far more difficult to do with walled garden gaming platforms and a fear of on-line ‘cheating’. Game accessibility has long been a game of catch up. Updating a game, however, has never been easier, thanks to the shift from physical media to digital distribution. Building accessibility in from the outset is much easier too, due to the rapid growth of information and knowledge sharing. Much of this improvement of knowledge is thanks to the web. Accessible gaming groups and people such as Audyssey, Without Wheels, the IGDA’s Game Accessibility Special Interest Group, Funka, BBC, OneSwitch, UA-Games, Eelke Folmer, AbleGamers and SpecialEffect were some of the early pioneers in building and sharing game accessibility expertise and information (OneSwitch, 2009). Something the web also helped facilitate was the refinement and spread of more structured game accessibility design help (OneSwitch, 2019).

There has also been progress at system level. The inclusion of accessibility software tools in computer operating systems became more common in the late 1980s and 1990s. Pushed by the Trace Center, Wisconsin on
the back of disability rights in education campaigning, the likes of IBM, Apple and Microsoft saw the benefits of including text to speech, magnifiers, on-screen keyboards, sticky keys and so on. More people could use their machines. Slowly, in the late 2010s, these types of features would surface in the likes of the PlayStation 4, Xbox One and Nintendo Switch. All of these things and more, driven by tireless advocacy on many fronts, have led to the current state of affairs in the gaming world. A time when it is not unrealistic to expect reasonable accommodations being made for humans of a very wide range of abilities.

3.1.2 Game Accessibility Guidelines
This chapter aims to get designers started with a freely available web resource of Game Accessibility Guidelines (GAG). The GAG is a living document with advice, examples of accessibility in games and quotes from disabled people. The production of GAG was a group effort by a core team consisting of developers, specialists and academics and validated through a broad pool of developers and gamers. The resource has been continuously updated since the release in 2012, based on developer input, gamer input, advances in technology and new examples of good practices.

The guidelines are divided into three main categories: Basic, Intermediate and Advanced, and sub-divided into Motor, Cognitive, Vision, Hearing, Speech and General. The Basic guidelines are design alternatives that are easy to apply for most game mechanics. The Intermediate guidelines remove barriers for many people but may take some more resources to implement and may not be applicable for all game mechanics. The Advanced guidelines may be complex to apply and are most relevant for specific niche groups of people. To learn more, please see the GAG website.*

3.1.3 Applying Game Accessibility Guidelines
To get started with the basics we’ll use a simple racing game as an example. A complete analysis of how individual features could be implemented is beyond the scope of this chapter; instead, we will focus on the process involved.

The process will be as follows: 1) familiarisation with the guidelines; 2) evaluate and plan for guidelines relevant for the game mechanics; 3) prioritise and schedule guidelines earlier that you estimate have the largest

* http://gameaccessibilityguidelines.com
production impact if considered later in the design process; 4) implement and test with disabled people; and 5) inform users about the game accessibility features.

### 3.2 EXAMPLE

#### 3.2.1 Barriers

For the purposes of this discussion, let’s consider a basic racing game, with a simple main menu consisting of the following items: New game, Load saved game, Options (containing screen resolution and volume), Exit. Even the relatively simple set of information and interaction in that initial menu can present barriers. For example:

1. Information presented solely through visuals.
2. Presentation of text, including how it is affected by the background it appears over.
3. Inconsistency and complexity of language, including clarity of outcomes.
4. Complexity of menu structure and interaction.
5. Interface elements are hard to select, due to size, motion or having to hold a button down for a period of time, or inaccessible using the standard gameplay controls.

Once past the menus, the game itself may also present barriers. For example:

1. Information presented solely through visuals, perhaps also with low contrast.
2. Difficulty distinguishing the car from the track and other cars (colour, shape etc).
3. Complexity of controls make them hard to learn and use.
4. Requiring both interactions and decision making at very fast speeds.
5. Information about laps, position, speed etc hard to see or understand.
6. The player has difficulty restarting the game.
So even within the scope of a simple menu structure and mechanic there is clearly significant opportunity for increasing the base of players who are able to have an enjoyable experience.

3.2.2 Working With Guidelines

The first step, before working on the game, is to already have some familiarity with best practice guidelines. Next, again before any work starts, consider which are relevant to your game; which elements of your design might present barriers relating to player motor, vision, hearing, speech and cognitive abilities. Reviewing the guidelines again will help with this. The site offers a downloadable Excel checklist format.*

Remember though that guidelines are only a generalised framework; you are the creative problem solver and guidelines are a tool, a source of information to help drive that problem solving, not standards or hard constraints. You may identify barriers specific to your game that are not covered by guidelines or want to figure out alternative solutions; when doing this the key principles to consider are: 1) communicate information in more than one way; and 2) allow flexibility in how players experience your game. Most accessibility barriers can be resolved using these two high level principles.

Ideally you should also involve disabled people to identify barriers they may face with your specific game; this can even take place in the form of formative user research before any work is done, based on previous/competitor products. As with user research in general, a caveat is that the information is specific to the individuals you test with, but if used together with guidelines, your own assessment and where necessary expert advice, the different methods of identifying barriers complement each other well. The more information you have, the more robust your solutions will be.

Next, whittle down the list according to which items are relevant to the concepts you have in mind and which are possible given the constraints of the technology and budget and timescales you are working with. Also think about which barriers are necessary and unnecessary – which barriers are an essential part of what makes the game enjoyable, and which get between players and the kind of emotional experience you want them to have. For example, 'Do not make precise timing essential to gameplay’ might not be a reasonable thing to aim for in a racing game. For this game you may end up with something like the following:

* [http://gameaccessibilityguidelines.com/excel-checklist-download/]
• Basic guidelines:
  o Ensure that all areas of the user interface can be accessed using the same input method as the gameplay.
  o Allow the game to be started without the need to navigate through multiple levels of menus.
  o Use an easily readable default font size.
  o Use simple clear language.
  o Use simple clear text formatting.
  o Provide high contrast between text/UI and background.
  o Ensure that all settings are saved/remembered.
  o Allow controls to be remapped/reconfigured.
  o Include tutorials.
  o Avoid flickering images and repetitive patterns.
  o Ensure no essential information is conveyed by colour alone.
  o Ensure no essential information is conveyed by sounds alone.
  o Offer a wide choice of difficulty levels.
  o Provide separate volume controls or mutes for effects, speech and background/music.

• Intermediate guidelines:
  o Support more than one input device.
  o Make interactive elements that require accuracy (e.g. cursor/touch-controlled menu options) stationary.
  o Ensure that all key actions can be carried out by digital controls (pad/keys/presses), with more complex input (e.g. analogue, speech, gesture) not required and included only as supplementary/alternative input methods.
  o Avoid/provide alternatives to requiring buttons to be held down.
- Allow the game to be started without the need to navigate through multiple levels of menus.
- Provide gameplay thumbnails with game saves.
- Provide separate volume controls or mutes for effects, speech and background/music.
- Ensure screen reader support.
- Provide an option to adjust contrast.
- Allow the font size to be adjusted.
- Include assist modes such as auto-aim and assisted steering.
- Provide a manual save feature.
- Provide an autosave feature.
- Include an option to adjust the game speed.

Note: ‘Advanced’ is not covered here as they are usually niche complex considerations.

The next step is to prioritise and schedule. The guidelines already suggest some priorities, the basic/intermediate/advanced categorisation is based on a balance of how many people benefit; how much of an impact it makes on the experience of people who benefit the most; and typical difficulty/effort of implementation. So, in general aim to meet the basic items first, although prioritisation will also be affected by your own production issues and technologies, publishers/platform/legal requirements and so on.

There will be items that just aren’t achievable, considerations that aren’t possible due to mechanics, budget, technology or even just internal politics, particularly if it’s only the first or second game that you’re considering accessibility on. **Do not let this discourage you.** View accessibility as an optimisation process rather than a bar to hit/miss; every single thing you’re able to do simply makes your game more enjoyable for more people.

Although what you aim for will likely change quite a bit as development progresses, having a list upfront means you’re in a position to identify which items need to be considered before any design or development work starts. So, spend some time considering dependencies and scheduling, which items can be addressed through a design decision versus feature
development. Which items need to be considered before any wireframing or design, which items in place for prototyping and so on. Also consider who in your team will be working on each item, assigning specific individuals/disciplines owners where applicable (and ideally a single person with overarching responsibility too). A proven strategy can be to produce separate small lists for specific teams or individuals of only considerations that are relevant to them, and get their buy-in on this general approach (with recognition that it will likely evolve) before their work starts. Having this to refer back to can be extremely valuable, particularly in more fraught later stages of development.

Also look at which considerations you can treat as success criteria of a feature, rather than distinct backlog items in their own right. For example, you may have an item on your backlog for addressing text size. Instead, have ‘adequate text size?’ as a requirement that must be met for any UI backlog item before it can be moved across into the ‘done’ pile. This is hard to do when you're starting out, but as your company gains more experience with accessibility you should be able to identify opportunities. This should all give you a clear picture of where in the backlog and scheduling items need to sit. There may be items that seem like a given, so not worth considering. For instance, keeping interactive elements stationary, or avoiding the need to hold down buttons. Even if an item seems like a given, it’s still worth keeping an eye on. Issues can easily creep in later in production as the game evolves.

So, you have a plan of what to consider and when to consider. Next comes implementation. Keep your original plan handy, refer back to it regularly (at key milestones at a minimum) and adapt as necessary. Guidelines give you a good initial framework to base your solutions on, but they are often purposely high level to allow flexibility in your solution to fit the unique barriers and constraints and opportunities presented by your game. The best solutions come from empowered designers and developers rather than generalised specifications, and there is also great opportunity for innovation and moving the industry forwards. So, if you feel like there is a different approach that would suit your game well, take the guideline as a description of the problem space and go ahead and push the boundaries. Often the solutions you will find are ones that make the game better for all players, but it is critically important not to rely on assumptions. Always remember that you have tools to ensure your approach works the way you intend, to gain more information to feed into your problem solving – user
research, expert review, wide scale player feedback like betas and social media and forum requests.

Even when designing precisely according to an existing specification or an example from an existing game, still validate. If your budget doesn’t stretch to formal user research, you can make use of cheaper methods like posting a mock-up or prototype on social media and asking for feedback. If you’re asking for people to give up significant amounts of their time to help make your game better you should be compensating them for their time, but if it’s just a case of some quick feedback through an email or tweet many people are very happy to help you out for free. Whichever method you’re using, do all that you can to make people feel like their feedback is welcome and wanted. There are lots of people who feel reluctant to speak up, having been discouraged through past experiences with both developers and fellow gamers, so any efforts you can make here are important.

Once you are getting closer to launch and have a more solid picture of what kind of considerations will be in the game, think about how this will be communicated to players. You cannot take it for granted that players will know about what you have considered or know to look through menus to find options. You can let people know in-game through tutorials, loading screen hints, start-up menus etc, and also let people know publicly through social media, blog posts, feature lists on digital store-fronts and press kits, and through features being present in demo builds at conventions. Aim to let people know at least some information before the game launches, so they can get in on pre-released and pre-launch hype as much as everyone else, safe in the knowledge that they will indeed be able to play.

3.2.3 Summary

- Consider accessibility from an early a stage as possible, ideally before any design or development work starts.

- Think about all the barriers your game may present for people with varied motor/cognitive/hearing/vision/speech ability, which of those barriers are essential for making your game enjoyable and which get between players and the kind of emotional experience you want them to have.

- Think about communicating information in more than one way and giving players flexibility in how they experience your game.
• When planning, prioritising and implementing do not be discouraged by not being able to achieve everything you want to – every consideration no matter how small makes a positive difference to your game.

• Treat guidelines as a tool and source of information, combine with other methods where possible and feel free to go beyond them.

• Assign owners, ensure different teams/disciplines know which considerations they are responsible for.

• Think about accessibility as a success criterion for existing backlog items.

• Involve disabled people both as a way of identifying barriers and validating solutions.

• Keep referring back to and adapting your plan throughout development.

• Give players the information they need to make informed purchase decisions.

3.3 CONCLUDING REMARKS

Hopefully, you now have a good idea of how you can work with the Game Accessibility Guidelines for your next or even ongoing game project. Remember, it is always better to try and do something than nothing, and if you don’t make it right the first time, just continue and you will make it better the next time. Think of it this way: even though game accessibility has been around since the 1970s, it was not until recent years that it was considered industry wide, so you are in good company when taking your first steps. We are all in this journey together, all of our efforts will help each other. Everything you do no matter how small doesn’t just benefit your own game, it helps push the industry in general to a better place.

If you want to learn more about game accessibility, see related resources at the IGDA Game Accessibility SIG* (GA-SIG) website†. You are also welcome to join the GA-SIG, simply by signing up to the mailing list or social media channels on the website. That way you will get in touch with many experts who are active in this field.

* SIG: Special interest group
† https://igda-gasig.org/
REFERENCES


