



**ACCESSIBLE™
GAMES
INITIATIVE**



**entertainment™
software
association**

Tags and Requirements

Accessible Games Initiative

**Version March 2025. Please check accessiblegames.com
to make sure you're referencing the latest version.**

Table of Contents



Auditory Features

<u>Multiple Volume Controls</u>	4
<u>Mono Sound</u>	6
<u>Stereo Sound</u>	7
<u>Surround Sound</u>	8
<u>Narrated Menus</u>	9
<u>Chat Speech-to-Text & Text-to-Speech*</u>	12

Gameplay Features

<u>Difficulty Levels</u>	14
<u>Save Anytime</u>	16

Input Features

<u>Basic Input Remapping</u>	18
<u>Full Input Remapping</u>	20
<u>Stick Inversion</u>	23
<u>Playable without Button Holds</u>	24
<u>Playable without Rapid Button Presses</u>	26
<u>Playable with Keyboard Only</u>	28

*This tag contains both auditory and visual features and is included in both categories.

Table of Contents



<u>Playable with Mouse Only</u>	29
<u>Playable with Buttons Only</u>	30
<u>Playable with Touch Only</u>	32
<u>Playable without Motion Controls</u>	33
<u>Playable without Touch Controls</u>	34

Visual Features

<u>Chat Speech-to-Text & Text-to-Speech*</u>	35
<u>Clear Text</u>	37
<u>Large Text</u>	40
<u>Large & Clear Subtitles</u>	43
<u>Color Alternatives</u>	46
<u>Camera Comfort</u>	48

***This tag contains both auditory and visual features and is included in both categories.**

Tag

Multiple Volume Controls

Player-Facing Description

Separate volume controls are available for different types of sounds.

- The volume can be changed separately for music, speech, sound effects, background audio, text-to-speech audio, accessibility audio cues, and voice chat.
- All game sounds can also be changed at once using one volume control.

Developer-Facing Requirements

Goal:

Let players adjust the volume of each type of sound so they can reduce distracting sounds and focus on sounds that they need to hear.

Requirements:

- Provide separate volume controls for each type of audio used in the game:
 - Sound effects (noises that communicate important information or that help players perform, such as footsteps approaching, gunshots offscreen, bushes rustling to indicate that there is a hidden creature).
 - Speech (character and narrative voices that communicate important information).
 - Ambient audio (background effects like whistling winds that don't impact players' decisions).
 - Music (background soundtrack or effects that don't influence players' decisions).

Tag

Multiple Volume Controls (cont.)

Developer-Facing Requirements

- Voice chat (doesn't apply if chat is only handled externally, or only by platform-level services).
- Menu narration (text-to-speech audio).
- Accessible audio cues (extra audio to help with accessibility such as directional indicator sounds for players with low vision).
- Provide a single control to change the volume of all game sounds at once.

Tips & Context:

- If the game contains multiple types of sound effects that are important, consider offering separate channels for each. For example, consider having separate controls for gunshot volume and footstep volume or for engine volume and tire volume. This will allow players to distinguish them from each other and from less important sounds covered by the general sound effects channel. Separate channels let players lower the volume of sounds they find overwhelming or that impact conditions like post-traumatic stress disorder (PTSD), and helps players focus on sounds that improve their ability to succeed or compete.
- Some games might also benefit from creating separate channels for active and passive in-game speech. For example, reducing the volume of background speech or making it more distinguishable from traffic noise lets players focus on important character speech.

Tag

Mono Sound

Player-Facing Description

Lets you play with mono audio.

- The same audio will be sent to all channels (e.g., both left and right headphones), effectively providing a single, combined audio channel.

Developer-Facing Requirements

Goal:

Provide mono audio for players who have difficulty hearing sounds from one side.

Requirements:

- Let players convert stereo and surround sound audio to mono audio that's sent to all channels.
- The option to turn this on through system preferences does not meet the requirements. Players must be able to choose mono sound within the game itself.

Tips & Context:

- Compressing all audio into a single channel meets the requirements, but it can make it harder for players to distinguish important audio cues. Therefore, it's useful to produce a specific "mono mix" with sounds balanced specifically for mono playback.
- It's useful to communicate that mono sound is on within the game itself (e.g., in the settings menu).

Tag

Stereo Sound

Player-Facing Description

Lets you play with stereo audio.

- Sounds communicate how far to the left or right they are coming from.
- Sounds will not communicate whether they are coming from above, below, ahead, or behind you.

Developer-Facing Requirements

Goal:

Provide stereo audio that helps players know which direction the game sounds are coming from.

Requirements:

- For sounds that have in-game locations, provide audio that communicates whether they are to the left or right of the player.
- The game can automatically turn this setting on or off, based on system preferences.

Tips & Context:

- It's useful to communicate that stereo sound is on within the game itself (e.g., in the settings menu).

Tag

Surround Sound

Player-Facing Description

Lets you play with surround sound.

- Sounds communicate where they are coming from, which may include any direction.

Developer-Facing Requirements

Goal:

Provide surround sound that helps players know precisely which direction the game sounds are coming from.

Requirements:

- Provide audio that communicates the direction of sounds that have in-game locations.
- When relevant to gameplay, this audio must provide more directional information than stereo sound could communicate.
- The game can automatically turn this setting on or off, based on system preferences.

Tips & Context:

- It's useful to communicate that surround sound is on within the game itself (e.g., in the settings menu).
- Surround sound might include outputting with 3.1, 5.1, or 7.1 surround sound or similar technologies, and/or virtually created surround, spatial, binaural, or similar technologies.

Tag

Narrated Menus

Player-Facing Description

Lets you use screen readers or voice narration for menus and notifications.

- Screen readers can access all menus, or the game provides similar functionality.
- Interactions and context changes are controlled by you and announced through narration.
- You can move through menus one item at a time, rather than steering a cursor.

Developer-Facing Requirements

Goal:

Help players understand and use menus and notifications without needing to see the text.

Requirements:

- Either provide narration for all menus and notifications (including non-interactive text) or support platform screen readers.
- Narrate headings and subheadings.
- Non-decorative images should have a brief narrated textual equivalent that summarizes what the image intends to communicate (alt text).
- For interactive elements, narrate their name, role, and state or value (such as “Accept, button,” “Full screen, checkbox, checked,” and “Volume, slider, 60%”).

Tag

Narrated Menus (cont.)

Developer-Facing Requirements

- Narrate what the new context is (such as the title of a new screen) whenever the player's context changes. This includes when the player initiates the change (such as moving to a new screen or changing which element is currently highlighted) and also when it happens automatically (such as a transition out of a loading screen or when the contents of a screen significantly change).
- Narrate significant changes that the game itself makes to information on the screen (such as a countdown timer, button prompt, or text updating that shows how many players were found in matchmaking).
- When the player moves to a new element, stop the narration of the previous element and start narration of the new element.
- Let players navigate all interfaces by shifting focus directly from one UI element to the next by a single action (through button/keypress, a single stick direction press, or a single touch-based action), without having to steer a cursor (such as a mouse pointer or moving a cursor using an analog stick).
- Don't narrate any content in menus that's purely decorative, used only for visual formatting, or is not presented visually.

Tag

Narrated Menus (cont.)

Developer-Facing Requirements

Tips & Context:

- These settings should be available to players before they're needed, like before gameplay and navigating game menus.
- Less important information that may clash/overwhelm can be deprioritized (prevented, limited, modified) in favor of other narration, or communicated by other means (e.g., audio cues). Examples include autosave notifications and countdowns.
- Good practice for headings/subheadings is to narrate their level. The main heading on the page is a level 1 heading, subheadings under that are level 2, further levels under those are level 3.
- Include information in your storefront listings about which languages are available for narration, either alongside other localization information or as part of the game's feature listing text.
- It is good practice to let players cancel or repeat a narration through an input action such as pressing a button or key.
- It is good practice to let players adjust the speed of narration (some listen at 400%).
- If specific screen readers have been tested, the game should provide a list of compatible screen readers for players.

Tag

Chat Speech-to-Text & Text-to-Speech*

Player-Facing Description

Lets you use text-to-speech and speech-to-text for game chats with other players.

- Text chats can be narrated out loud in real-time.
- Voice chats can be read as a text transcript in real-time.

Developer-Facing Requirements

Goal:

Make sure your social experiences are available to everyone by letting players communicate with each other using text-to-speech and speech-to-text for all game chats.

Requirements:

- If the game lets players communicate with each other using text, support text-to-speech so that players can hear the conversation narrated in real-time.
- If the game lets players communicate with each other using voice, support these options:
 - Speech-to-text so that players can read a text transcript in real-time.
 - Outgoing text-to-speech so that players can send a message to the voice chat.
- Support text-to-speech for any text-based canned messages (such as “Hello” and “Nice job”).

**This tag contains both auditory and visual features.*

Tag

Chat Speech-to-Text & Text-to-Speech* (cont.)

Developer-Facing Requirements

- Show speech-to-text messages in the same on-screen location as text chat so that players can easily read them together. That location must allow players to access the past four messages either by displaying them all at once, allowing players to scroll back to view the last four messages, or allowing players to expand the message view to see the last four messages.

Tips & Context:

- Quick chat or canned chat on their own do not meet these requirements.

**This tag contains both auditory and visual features.*

Tag

Difficulty Levels

Player-Facing Description

Lets you select from multiple difficulty options, including at least one option that reduces the intensity of the challenges.

- Differences between difficulty levels are also described.

Developer-Facing Requirements

Goal:

Give players options for the game's difficulty so they can choose an option that better matches their abilities and preferences.

Requirements:

- Provide multiple difficulty levels preset options.
- Provide at least one difficulty level that significantly reduces the intensity of the challenges.
- Describe the difference between the difficulty level options, such as which variables are changed (e.g., number of enemies, character health) and how they are changed (e.g., 50% more, slightly increased). See Tips & Context for examples.

Tips & Context:

- Examples of variables that can be changed to significantly affect the intensity of challenges:
 - Enemy or player health
 - Enemy or player damage
 - Number of lives before having to restart
 - Resource availability
 - Enemy AI

Tag

Difficulty Levels (cont.)

Developer-Facing Requirements

- Providing granular control over settings is strongly recommended to let players match their abilities and preferences to the intended game experience.
- When naming or describing different difficulty options, avoid:
 - Indicating or explicitly stating that one option is better than another (such as, “This mode is the intended experience”).
 - Naming or describing the difficulty levels in a way that demeans players (such as, “Baby mode”).

Tag

Save Anytime

Player-Facing Description

Lets you manually save your progress at any time.

Exceptions:

- The game is saving or loading.
- When saving could result in game-breaking scenarios or blocked progress, such as during death animations.

Developer-Facing Requirements

Goal:

Give players control over when they save their progress so they can safely stop playing when they need to.

Requirements:

- Let players manually save progress at any time during gameplay, with two exceptions:
 - The game is saving or loading.
 - When creating a save, and then loading from it, could result in game-breaking scenarios or blocked progress, such as saving during death animations or guaranteed failure states.
- Provide at least one (1) manual save slot that is separate from auto-save slots.
- Do not let auto-saves override manual saves.

Tag

Save Anytime (cont.)

Developer-Facing Requirements

- Quick-saves qualify as a type of manual save.
- To meet these requirements, the player must be loaded into exactly the same spot on resume even if it's in the middle of an active scene such as a boss fight or a race.

Tips & Context:

- Auto-saves may be created at any time.
- While a single slot is enough to meet these requirements, you should offer multiple slots if possible (e.g., players might want to create a new slot if they're unsure whether they're saving in a circumstance they can progress from).

Glossary:

- Manual saving means that players can choose to save their progress at any time.
- Quick-saves are a single reserved save slot that can be overwritten by a single input at any time without the need to reach a save or check point.

Tag

Basic Input Remapping

Player-Facing Description

Lets you rearrange the button controls.

- Button controls can be swapped or rearranged by some other method.
- The “Full Input Remapping” tag lets you remap all game controls, not just button controls, and remap them by choosing which action is performed by which input.

Developer-Facing Requirements

Goal:

Let players rearrange the digital controls so they can play.

Requirements:

- Provide a way to rearrange digital controls. Rearrangement must be available for all supported inputs (such as keyboard, mouse, controller, and virtual on-screen controller). Any method is permitted, including simple swaps of buttons (such as swapping A with B).
- These requirements **do** apply to:
 - Digital input interactions carried out using analog inputs such as sticks, triggers, touch-based surfaces, and motion-based functionality.
- These requirements **do not** apply to:
 - Inputs that can’t be remapped due to reserved system functions, such as the home or share button
 - Menu controls
 - Analog input interactions (see Tips & Context)

Tag

Basic Input Remapping (cont.)

Developer-Facing Requirements

- Offering a choice of preset control schemes is useful but, on its own, doesn't meet these requirements. Nor does relying on system-level remapping. Remapping must be provided by the game itself.

Tips & Context:

- Button prompts should ideally update to reflect any changes in mappings.
- For remapping both analog and digital controls, see the “Full Input Remapping” tag.

Glossary:

- Input interactions are either digital or analog and are independent of the device used to provide that input to the game.
 - Digital input interactions result in binary input (0 or 1, active or inactive), such as pressing a controller face button providing an active/inactive input, a trigger pull past a fixed threshold providing an active/inactive input.
 - Analog input interactions result in non-binary inputs, such as deflecting a stick to specific x- and y-axis values, or a trigger pull providing variable inputs within a 0-255 range.
- Supported inputs are the inputs that are identified on the game's store page or website as being supported. If no inputs are indicated as being supported, then the game's input support can't be evaluated and the game can't meet the requirements.

Tag

Full Input Remapping

Player-Facing Description

Lets you choose which action in the game is assigned to which control.

- All game controls can be remapped for all directly supported input methods, e.g. keyboard, mouse, controllers, and virtual on-screen controllers.
- Controller stick functionality can be swapped.
- The “Basic Input Remapping” tag lets you remap only button controls, and remap by simple methods like button swap.

Developer-Facing Requirements

Goal:

Let players fully remap game actions so they can play in a way that meets their specific needs and preferences, including the need for actions on the same input by default to be remapped separately.

Requirements:

- Provide a way to reassign which game action is controlled by each input for all supported inputs, whether analog or digital (such as keyboard, mouse, controller, and virtual on-screen controller).
- If your game uses either or both analog sticks of a controller, provide a way to swap the functions of the sticks.
- These requirements **do not** apply to:
 - Inputs that can't be remapped due to reserved system functions, such as the Home or Share button
 - Menu controls

Tag

Full Input Remapping (cont.)

Developer-Facing Requirements

- Typical basic gamepad controllers allow for analog input interactions from sticks, triggers, touch-based surfaces, and motion-based functionality. These may also be used by a game for digital input interactions.
- Relying on system-level remapping does not meet these requirements. Remapping must be provided by the game itself.

Tips & Context:

- These requirements are generally achieved by providing a list of actions and letting players choose which input should execute each action on the list.
- Button prompts should update to reflect any changes in mappings.
- For remapping only digital controls, see the “Basic Input Remapping” tag.

Glossary:

- Input interactions are either digital or analog and are independent of the device used to provide that input to the game.
- Digital input interactions result in binary input (0 or 1, active or inactive), such as pressing a controller face button providing an active/inactive input, a trigger pull past a fixed threshold providing an active/inactive input.
- Analog input interactions result in non-binary inputs, such as deflecting a stick to specific x- and y-axis values, or a trigger pull providing variable inputs within a 0-255 range.

Tag

Full Input Remapping (cont.)

Developer-Facing Requirements

- Supported inputs are the inputs that are identified on the game's store page or website as being supported. If no inputs are indicated as being supported, then the game's input support can't be evaluated and the game can't meet the requirements.

Tag

Stick Inversion

Player-Facing Description

Lets you change how direction inputs such as thumbsticks affect game movement in the up and down and left and right directions.

- Examples of these directional inputs include thumbsticks and flight sticks.

Developer-Facing Requirements

Goal:

Let players hold their input devices upside down or in other directions when playing.

Requirements:

- For analog inputs with both X and Y axes (such as left and right controller sticks or a flight stick), let players invert each input separately.
- This applies to any stick that is used by default and any stick that actions can be remapped onto.

Glossary:

- Analog input interactions result in non-binary inputs, such as deflecting a stick to specific x- and y-axis values, or a trigger pull providing variable inputs within a 0-255 range.

Tag

Playable without Button Holds

Player-Facing Description

Lets you play without button holds.

- The game doesn't require digital inputs (like keys or buttons) to be held.
- Some analog inputs (like sticks and triggers) may still require holds.

Developer-Facing Requirements

Goal:

Let players avoid digital input holds that might be impossible or uncomfortable for them.

Requirements:

- Either don't include digital input holds in your game, or offer an alternative:
 - A single input press instead of an input hold.
 - A toggle instead of an input hold.
 - A double-press or double-tap of an input instead of an input hold.
 - Remap the digital input hold to an analog input hold.
- These requirements **do** apply to:
 - All digital input holds that are required for gameplay.
 - Digital inputs holds using inputs that are typically used for analog interactions, such as sticks, triggers, touch-based surfaces, and motion-based functionality.

Tag

Playable without Button Holds (cont.)

Developer-Facing Requirements

- These requirements **do not** apply to:
 - Digital input holds that do not block gameplay. For example, the requirements do not apply to holding a button to skip a cutscene since players can advance by letting the cutscene play out.
 - Analog input interactions. See Tips & Context for more information.

Tips & Context:

- If a game has continuous analog inputs for moments like holding a left stick in a direction to move a character, developers are still encouraged to provide alternatives, but this is not part of the requirements for this tag.

Glossary:

- A hold is considered to be any input with a required duration. Some examples include holding a controller's face buttons to swap equipped weapons, open a container, make a menu selection, or complete a quick-time event (QTE).
- Input interactions are either digital or analog and are independent of the device used to provide that input to the game.
- Digital input interactions result in binary input (0 or 1, active or inactive), such as pressing a controller face button providing an active/inactive input, a trigger pull past a fixed threshold providing an active/inactive input. Analog input interactions result in non-binary inputs, such as deflecting a stick to specific x- and y-axis values, or a trigger pull providing variable inputs within a 0-255 range.
- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable without Rapid Button Presses

Player-Facing Description

Lets you avoid repetitive button actions like button mashing and quick-time events.

Developer-Facing Requirements

Goal:

Give players ways to avoid repetitive button actions that might be impossible or uncomfortable for them.

Requirements:

- Either don't require repetitive button actions in your game or offer an alternative.
- This requirement applies to:
 - All rapid button presses, such as those required for executing combos (e.g., hitting X, then hitting Y within one second to execute a combo).
- Examples:
 - Let players avoid sequences of one or more precisely-timed button presses (e.g., actions that require players to hit X within one second then Y within one second).
 - Let players avoid rapid repeated tapping actions (e.g., hitting X 10 times within two seconds to lift a heavy object).

Tag

Playable without Rapid Button Presses (cont.)

Developer-Facing Requirements

Tips & Context:

- It is good practice to avoid requiring rapidly repeated inputs of any type, such as waggling sticks to escape an enemy's grab.
- Some events require multiple simultaneous inputs. Simultaneous inputs are not part of the requirements for this tag, but it is still good practice to avoid these when possible.

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable with Keyboard Only

Player-Facing Description

Lets you play using only your keyboard.

- The game can be played with a keyboard alone, without any other devices.

Developer-Facing Requirements

Goal:

Support players who can use only a keyboard or who prefer playing that way.

Requirements:

- Let players play the game using only a keyboard.

Tips & Context:

- This keyboard-only option also makes the game more accessible to players using hardware that maps to keyboard inputs (such as accessibility switches).

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable with Mouse Only

Player-Facing Description

Lets you play using only your mouse.

- This also lets you play using adaptive tech that maps to mouse inputs.

Developer-Facing Requirements

Goal:

Support players who can use only a mouse or who prefer playing that way.

Requirements:

- Let players play the game using only a 2-button computer mouse.

Tips & Context:

- Mouse support also makes the game more accessible to players using hardware that maps to mouse movements (such as eye-trackers or head pointers).

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable with Buttons Only

Player-Facing Description

Lets you play using only buttons where the amount of pressure doesn't affect the controls.

- The game and menus can be controlled using only digital inputs (like buttons or keys).

Developer-Facing Requirements

Goal:

Support players who can't easily adjust how much physical pressure they put on input controls or who prefer to use digital input methods.

Requirements:

- Let players play the game and navigate all interfaces using only digital inputs (such as d-pad, face buttons) instead of analog inputs (such as analog triggers, thumb sticks, and mouse).
- These requirements **do** apply to:
 - Digital interactions using analog inputs such as sticks, triggers, touch-based surfaces, and motion-based functionality.
 - If analog triggers have simple on-off functionality (such as to fire or not fire) rather than an analog range (such as controlling how fast or slow the player is moving), then they count as a digital input.

Tag

Playable with Buttons Only (cont.)

Developer-Facing Requirements

Tips & Context:

- Digital input is more easily mapped to some types of assistive technology like accessibility switches.

Glossary:

- Input interactions are either digital or analog and are independent of the device used to provide that input to the game.
 - Digital input interactions result in binary input (0 or 1, active or inactive), such as pressing a controller face button providing an active/inactive input, a trigger pull past a fixed threshold providing an active/inactive input.
 - Analog input interactions result in non-binary inputs, such as deflecting a stick to specific x- and y-axis values, or a trigger pull providing variable inputs within a 0-255 range.
- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable with Touch Only

Player-Facing Description

Lets you play using only touch controls.

- Players are not required to use any type of non-touch controls, such as buttons or analog sticks.

Developer-Facing Requirements

Goal:

Support players who can use only touch-based controls or who prefer playing that way.

Requirements:

- Let players play using only touch input like a controller touchpad or a touchscreen on a mobile device.

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable without Motion Controls

Player-Facing Description

Lets you play without using motion controls.

Developer-Facing Requirements

Goal:

If motion controls are used, players can turn them off or map them to something else.

Requirements:

- Let players play without motion controls such as gyroscopes or accelerometers, and without the automatic detection of player movement.

Tips & Context:

- Motion controls can be a useful accessibility feature for some players.

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Playable without Touch Controls

Player-Facing Description

Lets you play without using touchpads or touchscreens.

Developer-Facing Requirements

Goal:

If touch input is used, players can turn it off or map it to something else.

Requirements:

- Let players play without resistive, capacitive, or other touch screen or touchpad input.

Tips & Context:

- Touch input can be a useful accessibility feature for some players.

Glossary:

- Playable means that a player can complete the game without being blocked. Optional game content may be inaccessible.

Tag

Chat Speech-to-Text & Text-to-Speech*

Player-Facing Description

Lets you use text-to-speech and speech-to-text for game chats with other players.

- Text chats can be narrated out loud in real-time.
- Voice chats can be read as a text transcript in real-time.

Developer-Facing Requirements

Goal:

Make sure your social experiences are available to everyone by letting players communicate with each other using text-to-speech and speech-to-text for all game chats.

Requirements:

- If the game lets players communicate with each other using text, support text-to-speech so that players can hear the conversation narrated in real-time.
- If the game lets players communicate with each other using voice, support these options:
 - Speech-to-text so that players can read a text transcript in real-time.
 - Outgoing text-to-speech so that players can send a message to the voice chat.
- Support text-to-speech for any text-based canned messages (such as “Hello” and “Nice job”).

**This tag contains both auditory and visual features.*

Tag

Chat Speech-to-Text & Text-to-Speech* (cont.)

Developer-Facing Requirements

- Show speech-to-text messages in the same on-screen location as text chat so that players can easily read them together. That location must allow players to access the past four messages either by displaying them all at once, allowing players to scroll back to view the last four messages, or allowing players to expand the message view to see the last four messages.

Tips & Context:

- Quick chat or canned chat on their own do not meet these requirements.

Tag

Clear Text

Player-Facing Description

Text in menus, control panels, and settings is a reasonable size. You can adjust the contrast.

- Text is at a reasonable size relative to the device's screen resolution and typical viewing distance.
- Font is less stylized or can be changed to a less stylized option (e.g., sans serif).
- Text has, or can be adjusted to, a reasonable contrast against all backgrounds.

(See “Large & Clear Subtitles” tag for subtitle text options.)

Developer-Facing Requirements

Goal:

Make the text in your game easier to read for as many players as possible.

Requirements:

- Minimum default text height in PC and console games:
 - 4K: 52 pixels
 - 1080p: 26 pixels
 - 720p: 17 pixels
- Use a sans-serif font or provide the option to use one.
- Use a text color that has a minimum contrast ratio of 4:5:1 against all backgrounds. Providing the option of using an opaque background container that ensures a minimum contrast ratio of 4:5:1 between the text and the background is sufficient to meet these requirements.

Tag

Clear Text (cont.)

Developer-Facing Requirements

- These requirements **do** apply to:
 - All front-end menu text in the game such as menus, settings, maps, and inventory screens. This includes text in full caps and other text without ascenders/descenders.
 - All heads-up display (HUD) text such as on-demand radial menus, health bars, waypoints, or diegetic UI.
- These requirements **do not** apply to:
 - Subtitles (covered in "Large & Clear Subtitles" tag).
 - Writing that appears in-world such as billboards, newspapers, or clothing.
 - Text that is purely decorative such as binary numbers used as an aesthetic device on the background of a menu.
 - Screens outside of menus and gameplay such as credits.

Tips & Context:

- Avoid fonts that have very large or very small x-height. If you use a font with a small x-height, consider using larger font height (ascender to descender height) than you normally would to make up for the smaller lowercase letterforms.

Tag

Clear Text (cont.)

Developer-Facing Requirements

- Text must always be measured on the basis of ascender to descender, not just cap height. There are three ways to do this:
 - Measure other text in the game that is in the same font and size and has descenders.
 - If that is not possible, input some text in your design software in the font and size that has ascenders and descenders, and then measure that.
 - If neither of those is possible, estimate cap height to be around 75%, e.g., 20px full caps = 26px asc-desc.
- Text settings should be available to players before they're needed, like before gameplay and navigating menus.
- For dialogue text, see the “Large & Clear Subtitles” tag.

Glossary:

- Sans-serif font is a letter or typeface without serifs. Serifs are short lines stemming from and at an angle to the upper and lower ends of the strokes of a letter.
- Text height is the total distance between ascender (top of lower case “h”) and descender (bottom of lower case “p”).

Tag

Large Text

Player-Facing Description

Lets you use a large font size for text in menus, control panels, and settings.

- Text can be in a large size relative to the device's screen resolution and typical viewing distance.

(See “Large & Clear Subtitles” tag for subtitle text options.)

Developer-Facing Requirements

Goal:

Help everyone comfortably read the text in your game by either providing a very large font size or letting players adjust the size.

Requirements:

- Either set the default text height to the following minimum size or give players a way to adjust it to this size on PC and console games:
 - 4K: 76 pixels
 - 1080p: 38 pixels
 - 720p: 25 pixels
- These requirements **do** apply to:
 - All front-end menu text in the game (such as menus, settings, maps, and inventory screens). This includes text in full caps and other text without ascenders/descenders.
 - All heads-up display (HUD) text (such as on-demand radial menus, health bars, waypoints, or diegetic UI).

Tag

Large Text (cont.)

Developer-Facing Requirements

- These requirements **do not** apply to:
 - Subtitles (covered in "Large & Clear Subtitles" tag).
 - Writing that appears in-world (such as billboards, newspapers, or clothing).
 - Text that is purely decorative (such as binary numbers used as an aesthetic device on the background of a menu).
 - Screens outside of menus and gameplay (such as credits).

Tips & Context:

- Avoid fonts that have very large or very small x-height. If you use a font with a small x-height, consider using larger font height (ascender to descender height) than you normally would to make up for the smaller lowercase letterforms.
- Text must always be measured on the basis of ascender to descender, not just cap height. There are three ways to do this:
 - Measure other text in the game that is in the same font and size and has descenders.
 - If that is not possible, input some text in your design software in the font and size that has ascenders and descenders, and then measure that.
 - If neither of those is possible, estimate cap height to be around 75%, e.g., 20px full caps = 26px asc-desc.

Tag

Large Text (cont.)

Developer-Facing Requirements

- Text settings should be available to players before they're needed, like before gameplay and navigating menus.
- For dialogue text, see the “Large & Clear Subtitles” tag.

Glossary:

- Text height is the total distance between ascender (top of lower case “h”) and descender (bottom of lower case “p”).
- x-height is the height of a lowercase “x” relative to a lower case “h.”

Tag

Large & Clear Subtitles

Player-Facing Description

Subtitles are available for all dialogue.

- Text is at a reasonable size relative to the device's screen resolution and typical viewing distance.
- The subtitle background transparency can be adjusted.
- Subtitles don't overlap with important game elements.
- Font is less stylized or can be changed to a less stylized option (e.g., sans serif).
- This tag covers spoken game dialogue only and doesn't include text displayed for other audio such as speaker tone or environmental sounds that are typically included in captions.

Developer-Facing Requirements

Goal:

Players can use subtitles to follow all the game's dialogue and have options to make the subtitles easier to read.

Requirements:

- Provide subtitles for all spoken content.
- Don't position the game HUD elements behind or over subtitles.
- Identify who's speaking or make that an option for players to turn on.
- Minimum default text height for 1080p for PC and console games: 32p.

Tag

Large & Clear Subtitles (cont.)

Developer-Facing Requirements

- Provide the option to scale the text height up to at least 46 pixels at 1080p on PC games.
- For all other resolutions, scale the numbers accordingly to remain equivalent to the 1080p sizes.
- Use a sans-serif font or provide the option to use one.
- Provide the option of a background container with adjustable transparency. Include the option to choose a solid black background with white text and no transparency.
- Not every word must be subtitled. Practices such as prioritization systems that selectively display NPC dialogue and editing of repeated words are permitted.

Tips & Context:

- It's recommended to show a maximum of 40 characters per line and 2 lines on-screen at a time.
- Avoid fonts that have very large or very small x-height. If you use a font with a small x-height, consider using larger font height (ascender to descender height) than you normally would to make up for the smaller lowercase letterforms.

Tag

Large & Clear Subtitles (cont.)

Developer-Facing Requirements

- Text height must always be measured on the basis of ascender to descender. You can't just measure cap height. There are three ways to do this:
 - Measure other text in the game that is in the same font and size and has descenders.
 - If that is not possible, input some text in your design software in the font and size that has ascenders and descenders, and then measure that.
 - If neither of those is possible, estimate cap height to be around 75%, e.g., 20px full caps = 26px asc-desc.
- Subtitles should be on by default, with an option to be able to turn them off. If they're off by default, let players turn them on before they are first needed (e.g., before gameplay and cutscenes).
- Fonts that are less stylized are generally easier to read, e.g., fonts that have consistent letter sizing, stroke width, and aren't slanted or crooked.
- Distinguishing speakers by color alone is permitted but would exclude the game from the "Color Alternatives" tag.

Glossary:

- Sans-serif font is a letter or typeface without serifs. Serifs are short lines stemming from and at an angle to the upper and lower ends of the strokes of a letter.
- Text height is the total distance between ascender (top of lower case "h") and descender (bottom of lower case "p").
- x-height is the height of a lowercase "x" relative to a lower case "h."

Tag

Color Alternatives

Player-Facing Description

Color is not used to communicate important information or can be adjusted.

- Shape, pattern, icons, or text is used to communicate information instead of color.

Developer-Facing Requirements

Goal:

Don't rely only on color to communicate or differentiate important information. Not everyone can see colors in the same way.

Requirements:

- Either don't use color to communicate or differentiate important information, or offer at least one of the following:
 - Provide other visual cues like shape, pattern, icons, or text to represent the colors' meanings.
 - Provide settings that let the player change the colors of color-dependent elements.
- These requirements apply to information provided through text, such as subtitles. For example, only using color to communicate the identity of the subtitle speaker doesn't meet the requirements for this tag.
- Providing full-screen filters that affect the whole palette doesn't meet these requirements (like shifting all blues in the game towards green).

Tag

Color Alternatives (cont.)

Developer-Facing Requirements

Tips & Context:

- Providing pre-set color schemes for specific types of colorblindness can be useful. However, players may also want the ability to freely choose which colors to use.

Tag

Camera Comfort

Player-Facing Description

There are no camera effects that may cause discomfort or harm (e.g., nausea, headaches) or those effects can be turned off or adjusted.

- ‘Camera effects’ include, but are not limited to: shaking, swaying, bobbing, motion blur, camera speed, and forced narrative-based movement.

Developer-Facing Requirements

Goal:

Let players avoid camera movements and visual effects that might cause discomfort or harm.

Requirements:

- If your game has any player-controlled camera movement, let players decrease the camera movement speed. This includes effects that mimic camera movement, such as vection or a distortion of world proportions.
- Either don’t use these visual effects in your game, or let players turn them off or adjust them:
 - Screen shake effects.
 - Bob effects (swaying the camera or weapon up and down as a character walks).
 - Motion blur (e.g., visual blurring or streaking when moving quickly).
 - Arm sway (e.g., swaying the camera in any direction as a player is aiming).

Tag

Camera Comfort (cont.)

Developer-Facing Requirements

- Narrative-based camera movement (like in dream sequences or scenes where characters are disoriented, or automatically turning the camera to face a point of interest).
- These requirements apply to camera movement during gameplay.
- These requirements do not apply to camera movement during cutscenes and other cinematics.

Tips & Context:

- This is not an exhaustive list of all that can be done to mitigate simulation sickness.
- One way to provide player control for narrative-based camera movement is to require their interaction. For example, a crash is heard during a scene and the player is prompted to press Y to look or investigate.
- Sliders to control intensity can be useful. However, to meet the requirements, the slide must go to zero or have an option to turn the effect off.



ACCESSIBLE™
GAMES
INITIATIVE



entertainment™
software
association

601 Massachusetts Avenue NW | Suite 300 | Washington, DC 20001

www.accessiblegames.com

© 2025 Entertainment Software Association