# Universal design

## WCAG 2.1(<https://www.w3.org/TR/WCAG21/#intro>)

To achieve the highest level of usability possible, the program aims to comply with the relevant WCAG 2.1 principles (<https://www.w3.org/TR/2018/REC-WCAG21-20180605/#requirements-for-wcag-2-1>) . These guidelines are intended to make content more accessible to meet the needs of the widest possible range of users.

## Principle 1: Perceivable(<https://www.w3.org/TR/WCAG21/#perceivable>)

Information and user interface components must be presentable to users in ways they can perceive.

### 1.1 Text alternatives(<https://www.w3.org/TR/WCAG21/#text-alternatives>)

The main goal is to have alternatives to non-text content by giving options such as braille, speech or symbols or simpler language. Since our program is only accessible through a command line interface this point is non-applicable for the project

### 1.2 Time-based Media

Providing alternatives for time-based media such as audio clips or videos by giving alternatives like cations or sign language. This point is non-applicable since our program only involves text.

### 1.3 Adaptable

Main goal is to create content that can be offered in multiple ways without losing structure or information.

### Success Criterion 1.3.4 Orientation (<https://www.w3.org/TR/WCAG21/#orientation>)

There are limited options available for customization on a command line interface, but one way we can impact the user experience is text wrapping. By making sure the text doesn’t go out of the window size we avoid confusion and make it easier for the user to understand what is being presented on the screen.

### 1.4 Distinguishable

Make it easier for users to see and hear content.

### Success Criterion 1.4.1 Use of Color (<https://www.w3.org/TR/WCAG21/#use-of-color>)

The command line interface colour scheme is based on the user’s terminal and is normally a plain black or blue background and plain white or green text colour. This is not something we are able to change in our program but is something that can be changed based on the terminal the user choses to use.

### Success Criterion 1.4.4 Resize text (<https://www.w3.org/TR/WCAG21/#resize-text>)

The command line interface allows users to resize their text without the loss of content or functionality. Either by manually changing the setting of zoom, or by using shortcuts like Ctrl and +/- on Windows or Command and +/- on Mac OS.

### Success Criterion 1.4.10 Reflow (<https://www.w3.org/TR/WCAG21/#reflow>)

The program does not require users to scroll when presenting information. New information displayed will always emerge bottom up on the CLI. Whenever new text appears the CLI will always scroll down to the button where the new information is displayed.

## Principle 2: Operable (<https://www.w3.org/TR/WCAG21/#operable>)

User interface components and navigation must be operable.

### Success Criterion 2.1.1 Keyboard (<https://www.w3.org/TR/WCAG21/#keyboard>)

All functions of the program are available only through the command line interface. Therefore, the keyboard will be the only way to operate through the program.

### 2.2 Enough time (<https://www.w3.org/TR/WCAG21/#enough-time>)

Provides users enough time to read and use content. Since our program has no time limit and the history is easily accessible by scrolling up this point is non-applicable to our project.

### 2.3 Seizures and physical reactions (<https://www.w3.org/TR/WCAG21/#seizures-and-physical-reactions>)

Since the program only displays text, this is a non-issue.

### 2.4 Navigable (<https://www.w3.org/TR/WCAG21/#navigable>)

Navigation is done by entering commands with a keyboard and does not work without one. Relevant information about navigation is displayed on screen so that the user always knows how to move around in the program. To make the program more initiative for the user the different commands in the program is usually the first letter of the option.

### 2.5 Input Modalities (<https://www.w3.org/TR/WCAG21/#input-modalities>)

The program only works with input from a keyboard and therefore this point is non-applicable.

## Principle 3: Understandable (<https://www.w3.org/TR/WCAG21/#understandable>)

Refers mostly to applications that use media such as video and images, which we do not. Language in the program has changed after we got feedback in our user tests about hard-to-understand terminology.

## Principle 4: Robust (<https://www.w3.org/TR/WCAG21/#robust>)

# Don Norman's Principles of interaction design

**Visibility:** Our program is made to be used only in text format in a command line interface. Since we are making the program, the information displayed for the user will all be relevant. Available options are described on screen in a simple understandable way. Information regarding how to navigate the program is written in more detail in the user manual.

**Feedback**: After every action taken by the user the program will give feedback indicating if the entered data is accepted/declined or if something went wrong. Options users have to go forward will also be presented after each action is taken. Since the program is only available in a command line interface, the history is always easily accessible for the user by scrolling up.

**Constraints:** Since we are making the program ourselves, we have full control over which actions the user can take at any given place in the program. The program will constrain the user by only allowing input that we have predetermined as acceptable.

**Mapping:** A command is linked to each option in the different menus. The user will be shown which options are available at any given moment and easy keyboard presses performs each option.

**Consistency:** For the sake of simplicity and improving the user experience similar actions have the same input through the program.

**Affordance:** Since our program is only available in a command line interface, the prior knowledge a user has with a CLI will determine how easily they pick up on the program. The user manual will help users understand how to use the program, but the program will also display all options available at any given moment.

WCAG 2.1

With a console app, following WCAG 2.1 Principle 1 - Perceivable will be difficult regardless of how we do it. Most of it , if not everything, refer to media, images or graphical user interface layout, which a command line interface does not have.

WCAG 2.1 Principle 2 - Operable and WCAG 2.1

WCAG 2.1 Principle 4 - Robust refers to how user agents may extrapolate and use or parse data from HTML, which is entirely non-applicable in this case, as our product is a standalone executable written in C++.