

# BEST PRACTICES

**View online:** <https://www.construct.net/en/make-games/manuals/construct-3/tips-and-guides/best-practices>

## Back up regularly!

Neither the hardware nor software in your computer is perfect. Computers fail and software can crash. Back up your projects to protect yourself from losing work. It is essential to also maintain off-site backups. If all your backups are in the same computer or saved to disks all in the same building, catastrophic events like fire, flood, theft, software/hardware failure, or cyber attacks/related malware like ransomware, can all cause you to lose all your work and backups together.

**Cloud Save** is a good way to save your work where it is safe in case of disaster. However it is wise to keep secondary backups anyway, in case you lose access to your account, or the service has an outage or even shuts down. Keeping additional backups is particularly important when saving to local files or folders on your system. Construct can help you do this by automatically making backups. See the *Save & backup* section of the [Settings dialog](#). Check *Periodically back up active project*, and choose the location and backup interval. For example you could set up an automatic save to the same location as the project every 10 minutes, or select a local backup folder (where supported by the browser) to save backups to.

*Saving an extra copy to a local backup folder isn't a safe backup when using local file/folder saves: it's just another copy on the same system, and any disaster that renders the system inaccessible will cause you to lose both your work and its backups. It's best to use this option if you're already using Cloud Save (so there's a secondary copy on your system), or if the local folder is automatically copied to the cloud (such as a local Dropbox folder which will automatically upload all files in the folder to your cloud account).*

The advice to back up regularly is not specific to Construct. It is vital to adopt this practice for any work on a computer which is important to you. Do not wait until you've lost work before starting to do this. People lose work regularly from having poor backup practices. Don't be one of them!

## Test on multiple platforms, browsers and devices

It is essential to test your project works as intended across a range of different systems. While Construct exports are based on the HTML5 standard which in theory is implemented the same on all platforms, in practice there are variations between browsers and devices (e.g. in performance, features, text rendering, etc). You should install a range of browsers on every device you have available and test with them all to ensure your project will work well for everyone. **Remote Preview** Paid plans only can help with this, especially since you can get anyone in the world to help test with their devices. You may also need to make test exports to

check how your project works as published, since app containers like Cordova (for Android and iOS) can have differences too.

## Support touchscreen devices

Many users browse the web with touchscreen devices. Where applicable, you should design your project to also support touch input. Often you can simply use the Touch plugin instead of the Mouse plugin.

## Avoid superseded features

Some features exist mainly for backwards compatibility reasons and should be avoided in new projects in favor of newer features that essentially replace them. See [Superseded features](#) for a list of such features to avoid.

## Recommended file formats

You may wish to prepare artwork and audio in other software before importing to Construct. These are the formats we recommend.

### Images

Use **32-bit PNG** (Portable Network Graphics) while preparing images. Be sure to select 32-bit if you are given a choice; the 8-bit or lower versions may degrade quality. 32-bit PNGs are lossless and fully support alpha-channel transparency. Note some images such as Microsoft Paint do not support PNG transparency. You may need to use a different editor instead, such as [Paint.NET](#) on Windows.

You can choose different export formats like JPEG or WebP inside Construct to reduce the size of your finished project. However when importing you should still stick to 32-bit PNGs if possible, and leave Construct to recompress them when exporting. Construct does a lot of optimisation on export for you. It is unlikely that any third party tools or services will be able to beat Construct's existing lossless optimisations, unless they degrade the image quality. Remember there is no point optimising images before importing them to Construct since it stores them in projects as 32-bit PNGs with default compression settings; they are only optimised on export.

### Audio

Use **16-bit PCM WAV** while preparing audio. These are typically .wav files, but note that not all .wav files are 16-bit PCM. Importing a 16-bit PCM .wav file to Construct will automatically encode it to WebM Opus. PCM WAV files are lossless, ensuring there is no quality degradation while you prepare your audio files. Allowing Construct to perform the encoding ensures the encoding is only done once (so there is no unnecessary degradation), and that the correct format is used for support across a wide range of platforms.

### Fonts

Use fonts in **WOFF** (Web Open Font Format) format wherever possible. Fonts in other formats such as TTF may or may not work depending on specific details of browser support and the features used in the font file. WOFF was designed specifically for web browsers and so is the best supported format for web-based technology such as Construct.

## Security

**Never enter usernames or passwords in to events.** These will be visible in plain text in exported Javascript, and malicious users will very quickly be able to take control of the account. If you need to connect to something like a database, write a server-side script that talks to the database, then connect to the URL of the server.

## Performance

Many people worry about performance but just end up wasting their time. There is a specific way that you ought to approach performance questions and issues - see the guide on [Performance Tips](#) for more about that.

## Memory use

Sometimes people accidentally create projects that waste large amounts of memory due to not understanding the way things like imgs are handled in memory. See the guide on [Memory usage](#) for more about that.

## Download size

It's a good idea to at least check your project's download size, and try to reduce the size of any particularly large files. This can be important for platforms that impose a file size limit, or for people who have slow connections. See the guide on [Download size](#) for more about that.