# To-Do

## Pretty Essential

**Once Hugo supports “js.Build with page bundles”**, move all the game code from *assets* to *js/js\_game* inside the project itself (where it belongs).

Example: now the Wondering Witches code is in “assets/wondering-witches/js\_game/” => move it to “content/wondering-witches/js\_game”. Also update the one *main.js* file that references it there.

*The Pirate Games* => now it has a huge bundle (with Phaser, jsPDF, jsZIP and FileSaver all in there) => should probably just integrate it with PQ\_CANVAS and load that

**Experiment** with adding animations/videos to **rules** (board game mostly, but could also be done for video games.) => Use <https://motioncanvas.io/docs/> for this. It’s open source, free, uses Canvas/Web, and does animation through coding and vectors.

**Fonts:** make all LOCAL. Rewrite older games to load them via FontFace API.

## Games

**Starry Skylines**. Make a *physical* version:

* Give each element a drawing.
* Print its properties (description, random number, etcetera) on the card.
* Generate ~5 pages of these cards as usual.

**Keebble**. Add different sets of letter values (for Dutch, Spanish, German, …) which you can select in the settings. This is applicable to all games. (Base game has “Scrabble Expansion”, Knickknack has the values built-in, and Domino uses the values for probability of using symbol.)

**Wie is de Trol?** Less text on the page, a bit overwhelming now. Less text, more GIF explanation ( / let the game stand on its own) is a good idea in general.

## Layout/Speed

* Give images the correct fixed width/height => as per Google PageSpeed recommendation
* **The wavy pattern** of the footer is not showing up on Chrome mobile? (Probably still has to do with content-visibility?)
  + Metadata-block could be loaded via content-visibility as well. Or is it already?
* **Overview pages** => make them look more distinct from game banners. (A different background pattern? Maybe even unique icons/patterns per category, if I can manage?)
* **Entry Banner** => Add fading (dark) background gradient at the *top* of entry banners (with an image)? Like the old website design, it would help separate them and give it a sense of depth.
  + Already implemented this … but it doesn’t look good, because the gradient can only be a straight line … whilst every divider has the regular curvy pattern.
  + I’d need to add an absolute div, placed at the top, with the curvy mask.
* **Narrow Footer** => add more buttons (latest project, random project, etc.)
* Allow playing a **.webm** video when hovering mouse over an entry-banner? (If defined, of course.)
* Keep watch for AVIF images that are *too* compressed/hazy
  + Can I compare filesizes in Hugo and only pick the smallest one??

# Interactive Rules

Generalize as much as possible into **rules-style.scss**.

* Add proper anchor names to sections, tables, images and interactive examples => so you can click and move around
* Decorations? Around page number?
* Build on my running header/footer code => full size, easily allow placing text or images
* Now the folding/unfolding is kinda hacky (with the "jump to full height then collapse"). How to solve?
* Easy way to turn heading counters on/off or add an icon/image.
* Double-view => Make wider (full width?) on wide-screen?
* The "cycle through images" thing (from interactivity below)
* DOUBT: If random board generation should be included with the rules, or stay on the main page

# Pandaqi Studio

## Usage

Adding a new game:

* Create a new folder in the root “content” with the game name.
* Ensure **gamepage:true**, set **category** to boardgame/videogame.
* Write title, description, perhaps short description
* Usually need three images: thumbnail, header, and banner => place there, convert to WebP, link
* Favicon is *always* favicon.png in root folder
* Ensure metadata is correct (num players, platforms, price, etcetera
* Fill the page however I like

Updating CSS:

* Everything goes via modules in the “modules” folder => these files have an underscore prefixed.
* Critical always loads. Style always loads *after* the rest of the page. Style-Boardgame autoloads on boardgames, Style-Videogame autoloads on videogames.
* Try to use “content-visibility: auto;” on as many places as possible, but also check if it breaks anything.

Header/Banner Images:

* Small => 0.3\*width
* Small-medium => 0.35\*width
* Medium => 0.4\*width
* Big => 0.4\*width
* A good width is probably 1920 pixels. The CSS is made *slightly* smaller (e.g. 0.29 instead of 0.3) to ensure we never get ugly borders/gaps if things don’t fit perfectly.

Check if the AVIF files do not become *too* washed out and hazy. (Otherwise just leave them out => the system automatically checks if they exist)

The website has a **PWA** (Progressive Web App) enabled. If things aren’t updating, clean that cache. (This is done by simple renaming the cache it uses in the serviceworker, called pwa.js.)

## Shared Elements (Studio, Blog, Tutorials)

The sketchy buttons/masked links. Both their underlying texture and their SASS.

The wavy underlines. (Built into browsers, no need for shared SASS.)

The fonts are exactly the same: **Dosis** and **Raleway.**

The structure for a narrow header with only a few buttons (or no header at all), and then a footer with all the other info.

## Javascript Building

Each page **automatically** loads the PQ\_CANVAS suite. You can load more optionally.

* **words (true/false)** to load PQ\_WORDS
* **phaser (true/false)** to load Phaser
* **tools (true/false)** to load PQ\_TOOLS

Each game can draw JS from four sources.

* **extraJS (true/false):** uses the content of the */js/* folder on the same page. (Usually board generation.)
* **extraJSGame (true/false)**: uses the content of the */js\_game/* folder, but never published to the same page.
* **sharedJS (string):** looks for *assets/games/<string>/js\_shared/*. Used for shared libraries across multiple games/spin-offs. Can be published on the page or not (**sharedJSOnlyPublish**).

Each of these will bundle the files by default. Also enable the build by adding **Build: true** after it. (For example, **extraJSBuild: true**.) This looks for *main.js* and builds that using JavaScript module rules.

To make them recognize the folders, add an empty *index.md* inside, and unlist it everywhere else.

These folders just reference the *actual* code in *assets/games/gamename/…* => in the future, once Hugo supports it, move all these files to their own game projects.

**Note:** board generation should be wrapped inside something to protect for Phaser not existing. Once the *onload* event fires, *then* you actually create the Phaser class.

* By default, PQ\_CANVAS looks for an object called “BoardGeneration”.
* But it’s better to *specifically* set the class with PQ\_CANVAS.PHASER.setGenerationClass(obj).

## Lessons Learned

**Lesson Learned:** when using **content-visibility: auto;** also set a **contain-intrinsic-size**. It doesn’t have to be correct, it just prevents page jumps.

**Lesson Learned:** some browsers (such as Chrome) still require **-webkit** prefix for **mask-images**.

**Lesson learned:** GoHugo has “safeCSS” function for inserting CSS styles via its code (partials/shortcodes).

**Lesson learned:** this site cannot use partialCached, because both header and footer are wildly dynamic and based on rest of the page. And besides that, each website is different.

**Lesson learned:** Using “&mdash;” in titles doesn’t work anymore => use “ | ” instead, or just copy-paste an actual mdash?

# Modules

## Shortcodes

* **headerimage =>** @params “img”, “class”
* **embedvideo =>** @params “bg”, “vid” (urls, no ext), “class”, “col” (border color, hex), “thickness” (border thickness, including unit)
  + common classes are “limit-width” (to center it without extending full width), “darkened” (to darken the background), “rounded” (for rounded corners)
* **screenshot-gallery =>** @params “img1” “img2” “…” (urls, no ext)
* **buylink-container =>** @params class (optional extra class
* **buylink =>** @params “url”, “metadata” (comma-seperated list), inner content represents link text
* **section-centered =>** @params “heading”, “html” (optional; if true, the content is not treated as markdown, but pure html)
* **section-flex =>** @params “media” (image or video), “is\_video” (true/false)
  + Horizontal flex container. The media is displayed on one side. Inner content is displayed on the other side
  + Add class “reversed” to switch order.
* **boardgame-intro =>** @params “heading”, “img” (big header image)
  + => inner content is the blurb
  + => metadata (playing time, player count, etc.) is read from frontmatter
* **boardgame-settings =>** container for a settings/configuration block
  + **setting-seed**
  + **setting-playercount** => @params min, max, def
  + **setting-enum** => @params id, tekst, values (comma separated list)
  + **setting-checkbox** => @params id, text
* **figure** => @params “url”
* **video** => @params “url”, “autoplay” (default true), “loop” (default true), “controls” (default false)
* **review-container** => simply gets class “review-container”
  + **review =>** @params “stars” (number of stars), “author” (optional), inner content is the actual review

**IDEA:** Use *icons* for the platforms displayed in metadata. (Shorter, more visual, cleaner.)

## Partials

* Show-media => can show both images and video
  + Show-image => @params img\_url
  + Show-video => @params video\_url
* Games-list => simply lists/paginates entry banners
  + Entry-banner
* Metadata => lists metadata (tags, categories, date, etc.) of an entry

## Important frontmatter

Designate page type:

* Game pages should have **gamepage: true**
* Boardgames should *also* have **boardgame: true**
* Anything without these is a regular (info) page

Basic taxonomy:

* Date
* Categories
* Tags

Basic text:

* title => actual title, used everywhere for reference
* headerTitle => shown in tab (”<head>” section)
* blurb => shown within entry banner
* meta => stuff like playing time, number of players, complexity, etc.

Asset work

* googleFonts => url to *all* Google Fonts used
* extraCSS => true/false
* extraJS => list
* requiredJS => list (general assets used by many pages)

## Entry Banners

Many things can be set in the *frontmatter* to influence the entry banner for a project

Images:

* fullHeaderImg => used throughout the website whenever we need a *single* image to represent a project; also the *default* for the boardgame-intro (if you don’t override it)
* headerImg => background (optional; default bg is white)
* headerThumb => logo/thumbnail shown on non-text-side (optional; just centers the text otherwise)
* headerDarkened => darkens the background image, nothing else

Flex:

* entryBannerNonTextClasses
  + Common ones: “no-shadow” (removes default drop-shadow filter behind it)
* entryBannerTextClasses
  + Common ones:

Colors:

* bgColor => blurred background behind text
* bgColorLink => background on big button
* textColor
* textColorLink

# Notes/Backups

Notes (Golden Rules of Landing Pages):

* No navigation (people will click through, get distracted, clutters the page)
* Big, wide images that show off your game
* Start with trailer or animated GIF (that shows game as quickly as possible)
  + HOLY GRAIL: (autoplay) trailer inside a big wide header image!
* Put download buttons above the fold/as high as you can => the page was designed to make people download/buy, so use it in that way
* Add some movement, but keep it simple otherwise => very simple but clean landing pages are known to sell better
* Add a press kit; maybe not at the top, but somewhere it's easy to find
* Add quotes/reviews as early as you can (and make them look nice and like your game is absolutely amazing)
  + Social proof is really powerful
* When you add screenshots, make sure they are from different settings and don't look very similar. (Maybe even different color palette.)