🟢 MORDETHRHEDAN — TUTORIEL D'INSTALLATION

Ce tutoriel installe l'overlay dans **OBS Studio**, de zéro jusqu'au premier live, puis habille ta **chaîne Twitch**. Chaque page a sa **fiche** : tu peux suivre une fiche seule, sans lire le reste.

> 💡 Pour tout voir en direct : ouvre `index.html` dans ton navigateur (et `design/moodboard.html` pour essayer les couleurs). La version à lire confortablement de ce tutoriel est `TUTO.pdf`.

SOMMAIRE

1. Ce qu'il y a dans le dossier
2. Avant de commencer (config, la police Dyer, réglages OBS, les 3 gestes de base)
3. Les scènes, fiche par fiche : Démarrage · Pause · Fin · Cam seule · Contenu · Jeu · Speedrun
4. Les sources à la carte : Alertes · Chat · Objectif · Cadre néon · Compte à rebours · Fond · Cadre de la cam
5. Les transitions : Balayage · Volets
6. Brancher le chat et les alertes (StreamElements, journal, objectif, sons des alertes)
7. Habiller la chaîne Twitch
8. Tester sans être en live
9. Personnaliser (couleur selon le jeu…)
10. Dépannage

1. CE QU'IL Y A DANS LE DOSSIER

```
mordethrhedan/
├── reglages.html       ← LA page pour changer les textes et réglages (elle écrit mes-reglages.js)
├── config.js           ← les valeurs par défaut (remplacé à chaque mise à jour de l'overlay)
├── mettre-a-jour.cmd   ← met l'overlay à jour (double-clic), en gardant tes réglages : voir 2.1
├── fabriquer-images.cmd ← fabrique SANS OBS le planning de la semaine, les images du kit, les vidéos de transition (double-clic) : voir 7
├── mes-reglages.js     ← TES réglages, écrits par reglages.html (apparaît au 1er enregistrement ; à garder)
├── index.html          ← aperçu de tout
├── TUTO.md / TUTO.pdf  ← ce tutoriel
├── TUTO-JDR.md / .pdf  ← le tuto de la scène JDR (parties de jeu de rôle)
├── pupitre-jdr.html    ← le pupitre JDR : les PV pendant la partie, dans un dock d'OBS (voir TUTO-JDR)
├── CONCEPT.md / .pdf   ← le résumé du projet (DA, choix, ce qu'il reste à faire)
├── scenes/             ← les overlays de scène (une page = une scène OBS)
├── sources/            ← les éléments à poser où tu veux (alertes, chat, cadre…)
├── sons/               ← tes propres sons d'alerte, si tu veux (facultatif, voir 6.8)
├── transitions/        ← les transitions (+ videos/ : prêtes pour OBS)
├── chaine/             ← kit de chaîne Twitch (kit.html + export/ : les PNG)
├── assets/polices/     ← les polices (locales, sans internet), dont Dyer
├── outils/             ← les scripts (vidéos, images du kit, PDF) et actualiser-obs.lua pour OBS : voir 9
├── design/moodboard.html
└── css/  js/           ← le moteur (pas besoin d'y toucher)
```

Toutes les pages font **1920 × 1080**. Le fond à facettes est **découpé** là où ta webcam, ton jeu ou ton image doivent apparaître.

2. AVANT DE COMMENCER

2.1 Remplir les réglages avec `reglages.html`

1. Dans le dossier de l'overlay, double-clique sur `reglages.html` : la page s'ouvre dans ton navigateur (**Edge** ou **Chrome**).
2. Vérifie au minimum, dans **La chaîne** : le **nom affiché** et ton **identifiant Twitch** (celui de l'adresse `twitch.tv/…`) ; dans **Couleur et fond**, la couleur des cadres.
3. Dans **Objectif** : choisis ce que la barre affiche (**followers** ou **abonnés**) et ton objectif. Les vrais nombres de la chaîne viennent de StreamElements (section 6.6).
4. Clique 💾 **Enregistrer mes réglages** (en bas, ou **Ctrl + S**).
5. **La première fois**, une fenêtre s'ouvre : choisis le **dossier de l'overlay** (celui qui contient `reglages.html`), puis **Sélectionner le dossier**. Le navigateur demande s'il peut modifier les fichiers : clique **Modifier les fichiers** (ou **Autoriser**). La page écrit alors tes réglages dans le fichier `mes-`

`reglages.js`, à côté de `config.js` : rien à copier à la main. Les fois suivantes, elle s'en souvient et enregistre directement (au plus, le navigateur redemande l'autorisation).

6. Dans OBS : **clic droit sur la source › Actualiser** pour voir le changement.

Un **point** • à côté d'un réglage veut dire qu'il a changé et n'est pas encore enregistré. La section **Alertes** montre un aperçu de chaque alerte avec tes textes, et la section **Tester** ouvre les pages en mode test. **Où vont tes réglages ?** `config.js` contient les **valeurs par défaut** de l'overlay ; `mes-reglages.js` contient **seulement ce que tu as changé**, appliqué par-dessus. En haut de la page, un encadré dit combien de réglages perso tu as. Remettre un réglage à sa valeur d'origine le retire de `mes-reglages.js`.

> ℹ️ Avec **Firefox**, la page ne peut pas écrire dans le dossier : elle **télécharge** `mes-reglages.js`, à mettre dans le dossier de l'overlay (à la place de l'ancien). Préfère Edge ou Chrome.

À la main (sans la page) : ouvre `config.js` avec le **Bloc-notes** (clic droit › *Ouvrir avec* › *Bloc-notes*). Garde les guillemets `"…"` autour des textes et la virgule `,` en fin de ligne, enregistre, puis **Actualiser** dans OBS. Attention : un changement fait à la main dans `config.js` sera perdu à la prochaine mise à jour ; dans `reglages.html`, il est gardé.

Quand tu reçois une nouvelle version de l'overlay

En un clic (Windows) : OBS › **Outils › Scripts** › `actualiser-obs.lua` (voir 2.9) › bouton **Mettre à jour l'overlay**. Ou, OBS fermé, double-clique sur `mettre-a-jour.cmd` dans le dossier de l'overlay.

- L'overlay télécharge sa dernière version sur le serveur des overlays, et l'installe à la place de l'ancienne.
- **Ton** `mes-reglages.js` **n'est jamais remplacé** : tes réglages sont gardés. Les nouvelles fonctions arrivent avec leurs valeurs par défaut.
- L'ancienne version est d'abord copiée dans `sauvegardes\<date>` (les 3 dernières sont gardées) : si tu avais modifié un fichier à la main, il est là.
- Avec le bouton d'OBS, les sources s'actualisent et les webcams se replacent toutes seules. Sous la case, OBS affiche la version installée et le résultat.

La page de téléchargement (le zip complet, le tuto, « Ce qui a changé ») : l'adresse de `reglages.html` › **Mises à jour**.

À la main (sans le bouton) : copie les fichiers du nouveau zip par-dessus les anciens (remplacer). `mes-reglages.js` n'est pas dans le zip : il reste en place. Puis, dans OBS, actualise les sources.

> 🔁 **La toute première fois seulement** (si tes réglages étaient encore dans l'ancien `config.js`) : **avant** de remplacer les fichiers, fais une copie de ton `config.js` (ex. sur le bureau). Après la mise à jour, ouvre `reglages.html` › 📥 **Reprendre les réglages d'un ancien config.js** › choisis cette copie : tes réglages reviennent dans le formulaire (marqués •). Clique **Enregistrer**, c'est fini : ils sont maintenant dans `mes-reglages.js`.

2.2 La police Dyer : rien à installer

La police de *Beyond Good & Evil* est **déjà incluse** dans l'overlay (dans `css/theme.css` lui-même) : tout ce qui passe à l'écran l'utilise (titres, textes, chat, alertes, objectif, transitions, kit de chaîne), dès l'actualisation d'une scène, sans attendre. Les lettres accentuées (é, è, à, ç…), qui manquaient au fichier

d'origine, ont été ajoutées. Inutile de l'installer dans Windows. Les pages d'outils (`reglages.html` , `index.html`) restent dans une police simple, plus lisible.

2.3 Régler le canevas d'OBS (une seule fois)

Paramètres › Vidéo : résolution de base `1920 × 1080` , résolution de sortie `1920 × 1080` .

2.4 Geste A — ajouter une source Navigateur

1. Sélectionne la scène, puis dans **Sources** : **+** › **Navigateur**.
2. Donne un nom à la source, puis **OK**.
3. Coche **Fichier local**, **Parcourir**, choisis le fichier `.html` indiqué dans la fiche.
4. **Largeur :** `1920` , **Hauteur :** `1080` .
5. Coche les cases indiquées dans la fiche.
6. **OK**.

2.5 Geste B — placer une source au pixel près

1. Clique sur la source (webcam, jeu, image, widget), puis **Ctrl + E**.
2. **Position** : x et y de la fiche.
3. **Type de zone de délimitation** : *Mettre à l'échelle à l'extérieur de la zone*.
4. **Taille de la zone de délimitation** : largeur et hauteur de la fiche.
5. Coche **Rogner à la zone de délimitation**, puis **Fermer**.

> 💡 **Voir les chiffres directement dans OBS** : `reglages.html` › **Options des scènes** › coche **Afficher la taille et la position des zones**, puis **Enregistrer** et actualise. Chaque zone (webcam, contenu, jeu) affiche sa taille et sa position : tu les recopies dans Ctrl + E. Décoche ensuite. (Ou, pour une seule source : `?zones=1` dans l'adresse, geste C.)

2.6 Geste C — ajouter une option dans l'adresse

1. Dans la source Navigateur, **décoche** *Fichier local*.
2. Colle l'adresse complète dans **URL**, avec les options après un `?` :

```
file:///G:/Projets/Overlay/mordethrhedan/scenes/jeu.html?cam=bas-droite&couleur=rouge
```

Plus simple, et pour de bon : `reglages.html` **› Options des scènes.** Le coin de la webcam de la scène Jeu (ou « Pas de webcam ») et le chat de chaque écran s'y règlent une fois pour toutes, sans toucher aux adresses dans OBS. Une option écrite dans l'adresse d'une source passe **avant** ce réglage (pratique pour une source particulière).

2.7 Astuce — une seule source d'alertes pour toutes les scènes

1. **Scènes › +** : crée une scène **« Global — Alertes »** et ajoutes-y `sources/alertes.html` (fiche 4.1).
2. Dans chaque autre scène : **Sources › + › Scène** › « Global — Alertes », tout **en haut** de la liste.

> 🔑 **Règle d'or** : dans OBS, ce qui est **en haut** de la liste s'affiche **par-dessus**. L'overlay est **au-dessus** de la webcam, du jeu, de la manette et de LiveSplit (il les encadre et arrondit leurs coins) ; le **fond** est tout en bas. Ton widget de succès, lui, se met au-dessus de l'overlay.

2.8 Le chat : déjà dans les scènes

Les scènes Démarrage, Pause, Fin, Cam seule et Contenu contiennent **déjà** leur chat : **tu n'as rien à ajouter.** En Speedrun, il est coupé par défaut pour laisser voir LiveSplit (voir 3.7 pour l'afficher d'un clic). La source séparée `sources/chat.html` sert seulement pour une scène à toi, ou pour placer le chat autrement : dans ce cas, ajoute `?chat=0` à l'adresse de la scène pour retirer celui qui est intégré, puis ajoute la source séparée.

Quand tu changes de scène, le chat **réaffiche les derniers messages** (ceux des 10 dernières minutes, réglable dans `reglages.html` › **Le chat**) : il ne repart pas à vide.

2.9 Actualiser toutes les sources d'un coup (après un changement de réglages)

OBS n'a pas de bouton pour actualiser toutes les sources Navigateur : l'overlay en fournit un, sous forme de petit script OBS.

1. OBS › **Outils › Scripts** › onglet **Scripts** › **+** › choisis `outils/actualiser-obs.lua` (dans le dossier `mordethrhedan`).
2. À droite : le bouton **Actualiser toutes les sources Navigateur**, et la case **Actualiser tout seul les sources de l'overlay quand les réglages changent** (cochée) : après **Enregistrer** dans `reglages.html`, les sources se mettent à jour en 2 secondes.
3. **Fermer** : le script reste installé. Raccourci clavier possible : **Paramètres › Raccourcis clavier** › « Actualiser toutes les sources Navigateur ».

> ⚠️ Une page actualisée repart de zéro (un compte à rebours recommence).

Mettre à jour l'overlay (le même script)

Le bouton **Mettre à jour l'overlay (tes réglages sont gardés)** installe la dernière version (voir 2.1, « Quand tu reçois une nouvelle version »). Sous le bouton : la version installée, et le résultat de la dernière mise à jour.

Refaire les vidéos de transition et les images du kit (le même script)

Les **vidéos de transition** et les **images du kit Twitch** sont des fichiers déjà fabriqués : après un changement de **couleurs**, du **nom** ou des **textes du kit** dans `reglages.html`, elles gardent l'ancien rendu (la page des réglages te le rappelle quand tu enregistres). Deux boutons les refont avec tes réglages :

- **Refaire les vidéos de transition (avec tes couleurs)** : refait `transitions/videos/*.webm`, puis OBS recharge tout seul tes transitions Stinger de l'overlay. Compte 1 à 2 minutes par vidéo ; pendant ce temps, ces transitions sont coupées (un changement de scène se fait sans animation).
- **Refaire les images du kit Twitch** : refait `chaine/export/*.png` (profil, bannière, hors-ligne, panneaux, emotes, badges) ; le dossier s'ouvre à la fin, il reste à les envoyer sur Twitch (section 7).

OBS reste utilisable pendant ce temps. Le résultat s'affiche sous les boutons (rouvre la fenêtre des scripts pour le voir) et dans **Journal des scripts**.

> **La première fois**, il faut **Node.js** (et **ffmpeg** pour les vidéos). S'ils manquent, le message le dit : **reclique sur le bouton**, ils s'installent tout seuls (2 à 5 minutes, avec l'installeur de Windows), puis ça continue.

Placer les webcams tout seul (le même script)

Dans la même fenêtre **Outils › Scripts**, le script a aussi :

- le bouton **Placer les webcams sur toutes les scènes (et les ajouter là où elles manquent)** : dans chaque scène qui affiche une page de l'overlay (`scenes/jeu.html`, `contenu.html`, `cam-seule.html`…), ta webcam est mise **pile dans sa zone**, à la bonne taille, et rognée. Si une scène n'a pas encore de webcam, elle y est ajoutée, juste sous l'overlay ;
- la case **Replacer tout seul les webcams quand les réglages changent** (cochée) : tu changes la webcam de la scène Jeu dans `reglages.html` › **Options des scènes**, tu enregistres… et dans OBS la webcam se déplace en même temps que son cadre.

Pour que le script reconnaisse ta webcam, **son nom doit contenir « cam »** (ex. « Webcam ») ; elle peut être rangée dans le groupe de la cam (fiche 3.6). Une carte d'acquisition de console n'est jamais déplacée (sauf si son nom contient « cam »). Une scène réglée **Pas de webcam** : la webcam y est cachée, puis réaffichée quand tu la remets. Les positions viennent de `js/zones.js` (les mêmes que dans les fiches ci-dessous) ; si ton canevas est en 1440p, le script fait la conversion tout seul.

> Le rognage automatique demande **OBS 30.1 ou plus récent**. Avec un OBS plus ancien, si la webcam déborde de son cadre : clic droit › **Transformer** › **Rogner** à la main (geste B).

3. LES SCÈNES, FICHE PAR FICHE

Trois dispositions, toutes avec le **même fond** qui se voit là où tu n'as mis aucune source :

Disposition	Pages	En bref
Grande	Démarrage · Pause · Fin · Cam seule	pseudo en haut, grand cadre, ligne des **derniers événements** en bas, chat à droite
Petite	Contenu · Speedrun	manette ou cam en haut à gauche, LiveSplit (ou chat) en bas à gauche, pseudo + grand cadre + derniers à droite
Jeu	Jeu	jeu plein écran, cadre au ras des bords, pseudo dans un petit encadré ; **le cadre de la cam est une source à part**

> 💎 **Le fond, tout en bas de chaque scène** : ajoute `sources/fond.html` en **dernière** position (geste A). Il est **synchronisé** avec celui de l'overlay (même motif, mêmes éclats au même instant) : là où tu ne mets pas de source, on voit le fond, parfaitement raccord.

> 🗞️ **La ligne des derniers événements** (dernier follow, dernier sub, dernier raid, série de visionnage) se remplit toute seule : voir **6.1 bis**.

3.1 ⏳ Démarrage — `scenes/demarrage.html` (disposition Grande)

À quoi ça sert : « Ça commence bientôt » et le compte à rebours dans le grand cadre (« C'est parti ! » à zéro), posés sur ta source (image, cam…).

Dans OBS :

1. **Scènes › +** : « Démarrage ».
2. **+ › Scène** › « Global — Alertes » (si tu utilises les alertes de l'overlay ; si tu gardes tes alertes Streamlabs, mets plutôt ta source d'alertes Streamlabs).
3. Geste A avec `scenes/demarrage.html`, case **Actualiser le navigateur quand la scène devient active** cochée.
4. **+ › Image** (ou ta webcam) › « Ta source ». Geste B : position `45`, `130` · taille `1380` × `776`.
5. Geste A avec `sources/fond.html` › « Fond ».
6. Ordre final : `Alertes` · `Overlay` · `Ta source` · `Fond`.

Changer le texte : `reglages.html` › **Écrans avec un grand titre** (ou `?message=Je%20arrive` dans l'adresse).
Options : `?minutes=5` (10 min par défaut) · `?heure=20:30` (heure fixe : le compteur arrive à zéro à 20 h 30 ; aussi dans `reglages.html` › **… ou heure fixe**, prioritaire sur les minutes) · `?chat=0` · `?derniers=0` · `?couleur=rouge`

3.2 🧉 Pause — `scenes/pause.html` (disposition Grande)

À quoi ça sert : « Petite pause en cours » dans le grand cadre, posé sur ta source.

Dans OBS : comme le Démarrage (étapes 1 à 6) avec `scenes/pause.html` ; ta source en `45`, `130` · `1380` × `776`.
Changer le texte : `reglages.html` › **Écrans avec un grand titre** › Pause.

3.3 👋 Fin — `scenes/fin.html` (disposition Grande)

À quoi ça sert : « Merci d'être passés ! » dans le grand cadre.

Dans OBS : comme le Démarrage avec `scenes/fin.html` ; ta source en `45`, `130` · `1380` × `776`.

3.4 🎙️ Cam seule — `scenes/cam-seule.html` (disposition Grande, sans message)

À quoi ça sert : le « blabla » : ta webcam (ou n'importe quel contenu) dans le grand cadre, le chat à droite.

Dans OBS :

1. **Scènes › +** : « Cam seule ».
2. Tes alertes en haut (voir 3.1, étape 2).
3. Geste A avec `scenes/cam-seule.html`.
4. **+ › Périphérique de capture vidéo** › « Webcam » (ou une capture de fenêtre). Geste B : position `45`, `130` · taille `1380` × `776`.

5. Geste A avec `sources/fond.html` › « Fond ».
6. Ordre final : `Alertes` · `Overlay` · `Webcam` · `Fond`.

3.5 📱 Contenu — `scenes/contenu.html` (disposition Petite)

À quoi ça sert : une fenêtre ou un jeu dans le grand cadre, ta webcam (ou ta manette) en haut à gauche, le chat en bas à gauche.

Dans OBS :

1. **Scènes › +** : « Contenu ».
2. Tes alertes en haut.
3. Geste A avec `scenes/contenu.html`.
4. Ajoute tes sources, puis geste B pour chacune :

Source	Position (x, y)	Taille (l × h)
Fenêtre / jeu	`460`, `128`	`1415` × `796`
Webcam ou manette (en haut à gauche)	`45`, `45`	`380` × `285`

5. Geste A avec `sources/fond.html` › « Fond ».
6. Ordre final : `Alertes` · `Overlay` · `Webcam` · `Fenêtre` · `Fond`.

Le chat : en bas à gauche par défaut. Pour mettre autre chose à sa place : `reglages.html` › **Options des scènes** › « Contenu : le chat » décoché (la case devient un trou, comme pour LiveSplit en 3.7).

3.6 🎮 Jeu — `scenes/jeu.html` + `sources/cam.html`

À quoi ça sert : le jeu en plein écran, un cadre au ras des bords (ses coins arrondis sont bouchés par le fond, raccord avec le reste), ton pseudo dans un petit encadré posé sur le cadre, et ta cam dans un coin. **Le cadre de la cam et ton pseudo sont une source à part** (`sources/cam.html`) : tu la mets dans un **groupe** avec ta webcam (et ton compteur de morts, etc.) pour tout masquer d'un seul raccourci pendant une cinématique : cam, cadre et pseudo disparaissent ensemble.

Dans OBS :

1. **Scènes › +** : « Jeu ».
2. Tes alertes en haut.
3. **Le groupe de la cam** :
    1. Geste A avec `sources/cam.html` › « Cadre cam ».
    2. **+ › Périphérique de capture vidéo** › « Webcam ». Geste B : taille `320` × `300`, position selon le coin (tableau ci-dessous).
    3. Ajoute tes autres widgets à masquer en même temps (compteur de morts…).
    4. Sélectionne-les tous (Ctrl + clic), **clic droit › Grouper la sélection**, nomme le groupe « Cam ». Dans le groupe, `Cadre cam` doit être **au-dessus** de `Webcam`.
4. Geste A avec `scenes/jeu.html` › « Overlay ».
5. **+ › Capture de jeu** › « Jeu ». Geste B : `0`, `0` · `1920` × `1080`.

6. Geste A avec `sources/fond.html` › « Fond ».
7. Ordre final : `Alertes` · `Cam` (le groupe) · `Overlay` · `Jeu` · `Fond`.
8. **Le raccourci** : **Paramètres › Raccourcis clavier**, cherche « Cam » : règle une touche pour **Afficher « Cam »** et une pour **Masquer « Cam »** (ou la même pour les deux). La cam, son cadre, ton pseudo et tes widgets disparaissent ensemble.

Changer de coin : `reglages.html` › **Options des scènes** › « Jeu : la webcam ». Le cadre de la cam **et** le pseudo suivent (pseudo au-dessus quand la cam est en haut, en dessous quand elle est en bas). Déplace ensuite ta webcam :

Coin	Position de la webcam (x, y)
`haut-gauche` *(défaut)*	`56`, `72`
`haut-droite`	`1544`, `72`
`bas-gauche`	`56`, `708`
`bas-droite`	`1544`, `708`

> ⚠️ Si tu écris le coin dans l'adresse (`?cam=bas-droite`) au lieu des réglages, mets **le même** sur `scenes/jeu.html` et sur `sources/cam.html`.

Position perso (au pixel près) : `reglages.html` › **Options des scènes** › **La webcam de la scène Jeu** › **Position perso** (X, Y, largeur, hauteur, ou fais glisser la cam sur le plan) ; ou dans l'adresse : `?cam=1200,700,420,236`. **Le plus simple pour la poser dans OBS** : le bouton **Placer les webcams** du script `actualiser-obs.lua` (voir 2.9), qui suit aussi la position perso.

Options : `?cam=0` sur `scenes/jeu.html` (pas de webcam : pas de groupe, le pseudo est alors affiché par l'overlay, en haut à gauche) · cadre sans le pseudo : `reglages.html` › **Options des scènes** › « Jeu : ton pseudo sur le cadre de la cam » décoché (ou `?pseudo=0` sur `sources/cam.html`) · `?couleur=rouge`

3.7 ⏱ Speedrun — `scenes/speedrun.html` (disposition Petite)

À quoi ça sert : comme le Contenu, pour le speedrun : ta manette (ou ta cam) en haut à gauche, **LiveSplit** en bas à gauche, le jeu dans le grand cadre.

Dans OBS :

1. **Scènes › +** : « Speedrun ».
2. Tes alertes en haut.
3. Geste A avec `scenes/speedrun.html` › « Overlay ».
4. Ajoute tes sources, puis geste B pour chacune :

Source	Position (x, y)	Taille (l × h)
Jeu	`460`, `128`	`1415` × `796`
Manette ou webcam	`45`, `45`	`380` × `285`
LiveSplit	`45`, `360`	`380` × `675`

5. Geste A avec `sources/fond.html` › « Fond ».
6. Ordre final : `Alertes` · `Overlay` · `Manette` · `LiveSplit` · `Jeu` · `Fond`.

Le chat à la place de LiveSplit, d'un clic : ajoute `sources/chat.html` **au-dessus** de l'overlay, avec l'adresse `…/sources/chat.html?place=petite` (geste C). Il se pose pile dans la case du bas : l'**œil** d'OBS (ou un raccourci, comme en 3.6 étape 8) l'affiche par-dessus LiveSplit ou le cache.
Pour l'avoir toujours : `reglages.html` › **Options des scènes** › « Speedrun : le chat ».

3.8 🎲 JDR — `scenes/jdr.html` (les parties de jeu de rôle)

Tout est dans le tuto à part : `TUTO-JDR.pdf` (les personnages, la scène dans OBS, le pupitre, le dépannage). En bref :

À quoi ça sert : tes parties de jeu de rôle. La table de jeu en grand (CodexVTT ou un autre site), un **cadre par personnage**
dessous (son image, son nom, ses PV, une ressource, « à terre », « son tour »), le nom de la campagne et le chat à droite.
Les personnages se règlent dans `reglages.html` › **Scène JDR** ; leurs PV se changent pendant la partie dans le **pupitre**
(`pupitre-jdr.html`, un dock d'OBS : TUTO-JDR, chapitre 4).

Dans OBS (détails : TUTO-JDR, chapitre 3) :

1. **Scènes › +** : « JDR ». Tes alertes en haut.
2. **Geste A** avec `scenes/jdr.html` › « Overlay JDR ».
3. **+ › Capture de fenêtre** › « Table » : la fenêtre de ton navigateur ouvert sur la table de jeu, **sous** l'overlay.
4. **Geste B** sur la table : position `60`, `60`, taille `1440` × `810` (1440p : `80`, `80`, `1920` × `1080`).
5. **+ › Image** pour chaque personnage, **sous** l'overlay, dans la case de son portrait (positions : TUTO-JDR, 3.4).
6. **+ › Capture audio d'application** › Discord (pour que le stream entende les joueurs).

Options : `?chat=0` · `?titre=…` · `?apercu=1` (la zone de la table et les cases des portraits) · `?test=1` (une table de démonstration)

4. LES SOURCES À LA CARTE

À poser **en plus**, dans n'importe quelle scène. Position et taille avec `?x=&y=&l=&h=` (pixels 1920 × 1080).

4.1 🔔 Alertes — `sources/alertes.html`

À quoi ça sert : follows, abonnements, bits, raids et dons, avec un carillon doux (les gros événements font pulser le halo).

Dans OBS :

1. Dans la scène « Global — Alertes » (voir 2.7) : geste A avec `sources/alertes.html`.
2. Coche **Contrôler l'audio via OBS** et règle le volume dans le mélangeur audio.

Options : `?position=haut` (défaut), `centre` ou `bas` · `?test=1`
Nécessite StreamElements (section 6).

4.2 💬 Chat — `sources/chat.html`

À quoi ça sert : le chat seul, en cartes, dans son cadre « verre ».
Options : `?x=40&y=75&l=430&h=930` · `?nu=1` (sans cadre) · `?disparition=60` · `?test=1`

4.3 📊 Objectif — `sources/objectif.html`

À quoi ça sert : la barre d'objectif (followers ou abonnés : `reglages.html` › Objectif, ou `?objectif=sub`).
Options : `?x=560&y=960&l=800&h=100` · `?test=1`

4.4 ▢ Cadre néon — `sources/cadre.html`

À quoi ça sert : un cadre néon seul pour encadrer n'importe quoi (widget, image…). La zone donnée est l'**intérieur** du cadre.

Dans OBS :

1. Place ton widget et note sa position et sa taille (Ctrl + E).
2. Geste A avec `sources/cadre.html`, puis geste C avec ces chiffres : `?x=40&y=40&l=380&h=290`.

Options : `?verre=1` (intérieur fumé) · `?apercu=1`

En image (au lieu d'une source Navigateur) : le même cadre en PNG transparent, plus léger pour OBS, ou pour l'utiliser ailleurs. Dans PowerShell, dans le dossier de l'overlay, avec la taille de ce que tu encadres :

```
node outils/cadre-image.mjs 380x290
```

L'image arrive dans `assets/cadres/cadre-380x290.png` (ajoute `verre` après la taille pour l'intérieur fumé). Dans OBS : **+** › **Image** › ce fichier. L'image a **50 px de marge** de chaque côté (pour le halo) : place-la 50 px plus haut et plus à gauche que ce qu'elle encadre. Elle garde la couleur du moment : si tu changes de couleur dans `reglages.html`, refais-la.

4.5 ⏲ Compte à rebours — `sources/compte-a-rebours.html`

À quoi ça sert : un compte à rebours seul, en police néon, à poser où tu veux.
Dans OBS : geste A, case **Actualiser le navigateur quand la scène devient active** cochée, puis geste C.
Options : `?minutes=10&x=0&y=0&l=1920&h=1080&taille=260`

4.6 💎 Fond — `sources/fond.html`

À quoi ça sert : le fond à facettes seul, **tout en bas de chaque scène** (fiches 3.x) : il se voit là où il n'y a pas de source. Il est synchronisé avec le fond des overlays (même motif, même respiration au même instant).
Options : `?couleur=rouge` · `?graine=12` (autre motif) · `?fixe` (sans animation). Si tu mets une option sur l'overlay d'une scène, mets la même sur son fond.

4.7 🎥 Cadre de la cam — `sources/cam.html`

À quoi ça sert : le cadre néon de la webcam de la scène Jeu **et ton pseudo** (dans son petit encadré sur le grand cadre), en source séparée, pour les mettre dans un **groupe** avec la webcam et les masquer d'un

raccourci (fiche 3.6). Ses coins sont bouchés : la webcam, rectangulaire, ne dépasse pas du cadre arrondi. **Options** : `?cam=bas-droite` (même coin que la scène Jeu ; par défaut celui des réglages) · `?x=56&y=72&l=320&h=300` (taille libre) · `?pseudo=0` (sans le pseudo) · `?apercu=1`

5. LES TRANSITIONS

Les vidéos sont **déjà prêtes** dans `transitions/videos/` (fond transparent). Pour chacune :

1. Panneau **Transitions de scène** › **+** › **Stinger**, donne-lui un nom.
2. **Fichier vidéo** : `transitions/videos/balayage.webm` ou `volets.webm`.
3. **Type de point de transition** : *Temps (millisecondes)* ; **Point de transition** : `800 ms`.
4. **OK**.

Une transition par scène : clic droit sur la scène › **Remplacer la transition**.

Transitions dans une autre couleur, dans PowerShell :

```
cd G:\Projets\Overlay\mordethrhedan
node outils/generer-transitions.mjs couleur=rouge
```

Crée `balayage-rouge.webm` et `volets-rouge.webm`.

6. BRANCHER LE CHAT ET LES ALERTES

Deux choses différentes :

Quoi	Comment ça arrive	À installer
Le chat	l'overlay lit ton chat Twitch directement	**rien** : ton identifiant Twitch dans les réglages suffit
Les alertes et l'objectif (follows, abonnements, bits, raids, dons)	par **StreamElements**, un service gratuit (par internet)	**rien** : un compte StreamElements et son jeton, une fois (6.2)

> 🔔 **Tu préfères garder tes alertes Streamlabs ?** C'est possible : garde ta source d'alertes Streamlabs en haut de chaque scène, à la place de `sources/alertes.html`. Le jeton StreamElements (6.2) reste nécessaire pour la ligne des derniers événements et la barre d'objectif : les deux services cohabitent sans souci.

6.1 Le chat

1. `reglages.html` › **La chaîne** › **Identifiant Twitch** : celui de ton adresse `twitch.tv/…`. Enregistre.
2. C'est tout : dans les scènes, le chat s'affiche dès qu'un message arrive.

Les **emotes Twitch** s'affichent en image ; les **bots** (liste dans `reglages.html` › **Le chat**) et les **commandes** qui commencent par `!` sont cachés ; un message **supprimé par un modo**, ou ceux d'un spectateur **banni**, disparaissent aussi de l'overlay ; en changeant de scène, les messages des 10 dernières minutes sont

réaffichés.
Il ne montre pas les messages envoyés **avant** l'ouverture d'OBS, ni les emotes des extensions 7TV, BTTV ou FFZ.

6.1 bis La ligne des derniers événements

Case	D'où ça vient	À régler
Dernier sub	annonces du chat Twitch (abonnement, réabonnement, cadeau)	**rien**
Dernier raid (pseudo · nombre de viewers)	annonces du chat Twitch	**rien**
Série de visionnage (pseudo · nombre de streams d'affilée)	annonces du chat Twitch, quand un spectateur partage sa série	**rien**
Dernier follow	**StreamElements** (un follow ne passe pas dans le chat) ; sub et raid arrivent aussi par là	ton jeton StreamElements, une fois (6.2)

Le dernier follow : rien de plus à faire une fois StreamElements branché (6.2) : c'est le même jeton que pour les alertes et l'objectif.

Les cases se souviennent de leur valeur d'un live à l'autre, et toutes les scènes affichent la même chose. Les titres des cases se changent dans `reglages.html` › **Derniers événements**. Pour enlever la ligne sur une scène : `?derniers=0`.

> ⚠️ **Pas encore vérifié sur un vrai live** (les annonces Twitch en simulation, `?test=1`, et StreamElements avec un faux serveur seulement).

6.2 Brancher StreamElements (une seule fois, 3 minutes)

StreamElements est un service **gratuit**, par internet : **rien à installer** sur le PC, rien à lancer avant le live.

1. Va sur **streamelements.com** › **Login** › **Twitch**, connecte-toi avec **ton compte de streamer** et autorise. (Déjà un compte StreamElements ? Connecte-toi simplement.)
2. Ouvre la page de ton compte : **https://streamelements.com/dashboard/account/channels** (ou : ton avatar en haut à droite › **Account** › onglet **Channels**).
3. Clique **Show secrets** (« Afficher les secrets »), puis le bouton **copier** à côté de **JWT Token** : une très longue suite de lettres et de chiffres.
4. Ouvre `reglages.html` › **StreamElements** › colle-le dans **Ton jeton StreamElements**, puis 💾 **Enregistrer**.
5. Dans OBS : **clic droit sur la source d'alertes › Actualiser** (et sur les scènes, pour le bandeau et l'objectif).

> 🔒 **Ce jeton est un secret** (c'est la clé de ton compte StreamElements) : ne le montre pas en live, ne l'envoie à personne. Il reste dans `mes-reglages.js`, sur ton PC : les mises à jour de l'overlay ne l'emportent jamais. S'il a fuité : même page › **Show secrets** › bouton pour en refaire un, puis recolle le nouveau.

✅ **À chaque live** : rien à faire. L'overlay se branche tout seul à l'ouverture d'OBS, et se rebranche tout seul si internet coupe un moment.

6.3 Vérifier que l'overlay est bien branché

OBS n'a pas de console (F12) : l'overlay a donc son propre **journal**, affiché directement dans la source.

1. Dans OBS, double-clic sur la source d'alertes.
2. Décoche **Fichier local** et colle dans **URL** l'adresse de la page suivie de `?journal=1` (geste C) : `file:///G:/Projets/Overlay/mordethrhedan/sources/alertes.html?journal=1`
3. **OK** : un panneau sombre apparaît en haut à gauche de l'écran. Tu dois y lire ✅ **Connecté à StreamElements**, puis 📊 **Compteurs de la chaîne** avec ton nombre de followers et d'abonnés.
   - « ⚠️ Pas de jeton StreamElements » : colle ton jeton dans `reglages.html` (6.2).
   - « ❌ StreamElements refuse le jeton » : il est mal copié (il manque un bout ?) ou il a été refait : recopie-le (6.2).
   - « … connexion à StreamElements » qui reste seul : pas d'internet sur le PC, ou un pare-feu bloque `astro.streamelements.com`.
   - « ⚠️ StreamElements désactivé » : coche « Se connecter à StreamElements » dans `reglages.html`.
4. Chaque événement reçu s'y ajoute (ex. `follow → follow · Pseudo`), avec **les données brutes** reçues de StreamElements en dessous.
5. Une fois vérifié, **retire** `?journal=1` de l'adresse (sinon le panneau reste à l'écran pendant le live).

Le journal marche sur toutes les pages (`?journal=1`), et aussi avec `?test=1` pour voir passer les fausses alertes.

6.4 Événement par événement

Une fois StreamElements branché, **rien à régler par événement** : l'overlay les écoute tous. Les textes se changent dans `reglages.html` › **Alertes** (avec un aperçu).

Événement Twitch	Alerte
Follow	**Nouveau follow** — rejoint l'aventure
Abonnement	**Nouvel abonné** — s'abonne à la chaîne
Réabonnement	**Réabonnement** — est abonné depuis X mois
Abonnement offert	**Abonnement offert** — offre un abonnement à…
Pluie d'abonnements	**Pluie d'abonnements !** *(le halo pulse)*
Bits	**Bits !** — envoie X bits
Raid	**Raid !** *(le halo pulse)*
Don	**Merci pour le don !** — offre X
Objectif atteint	**Objectif atteint !** *(le halo pulse)*

Abonnements, abonnements offerts et bits demandent une chaîne **affiliée** ou **partenaire**. Les alertes passent **une par une** (file d'attente) : rien n'est perdu pendant un raid.

6.5 Les dons (facultatif)

Twitch ne gère pas les dons en argent. Avec StreamElements, ils sont **déjà branchés** : ta page de dons est `streamelements.com/<ton pseudo>/tip` (à mettre dans un panneau de ta chaîne, « Soutenir »). Pour recevoir l'argent : streamelements.com › **Revenue** › **Tipping settings**, puis relie ton compte PayPal (ou un autre moyen proposé).

Chaque don arrive tout seul dans l'overlay, avec son montant. *(Les dons faits par un autre service, Ko-fi, Streamlabs ou Tipeee, n'arrivent pas : passe par la page de dons StreamElements.)* Pour l'habiller aux couleurs de l'overlay (bannière et fond) : section 7.

6.6 L'objectif (followers OU abonnés, avec les vrais nombres)

Une seule barre, qui affiche au choix **les followers** ou **les abonnés**. Chacun a son nom, son objectif et son compteur : on bascule quand on veut, la barre est tout de suite juste.

A. Choisir ce que la barre affiche

1. `reglages.html` › **Objectif** › **Ce que la barre affiche** : **Les followers** ou **Les abonnés**.
2. Pour chacun : le **nom de l'objectif** et l'**objectif à atteindre** (ex. `50`).
3. 💾 **Enregistrer**, puis actualise les sources dans OBS.

Une seule source peut aussi afficher l'autre compteur sans toucher aux réglages : ajoute `?objectif=sub` (ou `?objectif=follow`) à son adresse.

B. Les vrais nombres de la chaîne (automatique avec StreamElements)

Rien à faire de plus : dès que StreamElements est branché (6.2), l'overlay lit le **vrai total** de followers et d'abonnés de ta chaîne, à l'ouverture de chaque page puis à chaque changement. Rien n'est modifié sur Twitch : il ne fait que lire.

Vérification : ajoute `?journal=1` à la source d'objectif (6.3). Une ligne **« 📊 Compteurs de la chaîne »** doit apparaître avec tes nombres, juste après « ✅ Connecté à StreamElements ».

> Les abonnés : seulement si la chaîne est **affiliée** ou **partenaire** (sinon il n'y en a pas). La case **Les vrais nombres de la chaîne, depuis StreamElements** (`reglages.html` › **Objectif**) doit rester cochée (c'est le cas par défaut).

Sans StreamElements : décoche cette case et mets ton vrai nombre dans **Followers : nombre de départ** (ou **Abonnés : nombre de départ**) ; le changer remet ce compteur à cette valeur. L'overlay compte alors seulement ce qui arrive pendant qu'OBS est ouvert.

6.7 Tester les vraies alertes

- **Sans Twitch** : `?test=1` sur une page (fausses alertes toutes les 9 secondes, sans toucher au vrai compteur), ou `reglages.html` › **Tester**.
- **Pour de vrai** : demande à un ami (ou à un deuxième compte) de suivre la chaîne, OBS ouvert.

> ⚠️ **Pas encore vérifié sur un vrai live** : si une alerte montre « Quelqu'un » ou « ? », fais une capture d'écran du journal (6.3) pour faire corriger l'overlay.

6.8 Les sons des alertes

Chaque alerte a **son propre son**, pour savoir ce qui se passe à l'oreille, même en pleine partie :

Alerte	Le son
Follow	deux notes de carillon
Abonnement	trois notes qui montent
Réabonnement	quatre notes en zigzag
Abonnement offert	trois notes aiguës rapides
Pluie d'abonnements	une cascade de huit notes
Bits	deux petits « tling » cristallins
Raid	une basse qui monte puis quatre notes
Don	un accord doux et long
Objectif atteint	quatre notes et un grand accord

Les écouter : `reglages.html` › **Sons des alertes**, bouton ▶ à côté de chaque alerte.

Mettre ton propre son (un mp3, wav ou ogg, court de préférence) :

1. Copie ton fichier dans le dossier `sons/` de l'overlay, par exemple `sons/follow.mp3`.
2. `reglages.html` › **Sons des alertes** : dans la case de l'alerte, écris `sons/follow.mp3`. Clique ▶ pour vérifier.

3. **Enregistrer**, puis dans OBS : clic droit sur la source des alertes › **Actualiser**.

Écris `aucun` dans une case pour que cette alerte reste silencieuse ; vide la case pour revenir au son de l'overlay.
Le volume général et le bouton « son » sont dans `reglages.html` › **Alertes** ; dans OBS, le volume se règle aussi dans le mélangeur audio (case **Contrôler l'audio via OBS** de la source des alertes).

7. Habiller la chaîne Twitch

Tous les visuels sont dans `chaine/` : ouvre `chaine/kit.html` pour les voir. Les images prêtes à envoyer sont dans `chaine/export/`. Ils prennent la même couleur que l'overlay.

Visuel	Fichier	Où l'envoyer sur Twitch
Photo de profil	`profil.png` (800 × 800)	Tableau de bord des créateurs › Paramètres › Chaîne › **Marque** › Photo de profil
Bannière de profil	`banniere.png` (1200 × 480)	… › **Marque** › Bannière de profil
Écran hors-ligne	`hors-ligne.png` (1920 × 1080)	… › **Marque** › Bannière du lecteur vidéo
Panneaux de bio	`panneau-a-propos.png`, `panneau-planning.png`, `panneau-regles.png`, `panneau-materiel.png`, `panneau-commandes.png`, `panneau-soutenir.png`, `panneau-mes-creations.png`, et les réseaux : `panneau-discord.png`, `panneau-youtube.png`, `panneau-tiktok.png`, `panneau-instagram.png`, `panneau-x-twitter.png` (320 × 160)	Ta chaîne › onglet **À propos** › **Modifier les panneaux** › **+**
Page de dons StreamElements	`dons-banniere.png` (640 × 200) et `dons-fond.png` (1920 × 1080)	streamelements.com › **Revenue** › **Tipping settings** (pas à pas ci-dessous)
Emotes	`emote-gg`, `pb`, `reset`, `gold`, `coeur`, `manette` (-112, -56, -28)	Tableau de bord › **Récompenses des spectateurs** › Emotes *(affilié ou partenaire)*
Badges d'abonné	`badge-mois-1`, `-3`, `-6`, `-9`, `-12` (-72, -36, -18)	Tableau de bord › **Récompenses des spectateurs** › Badges d'abonné

Les panneaux, pas à pas :

1. Va sur ta chaîne, onglet **À propos**, active **Modifier les panneaux**.
2. **+** › **Ajouter un panneau texte ou image**.

3. **Image** : choisis le PNG du panneau. **Description** : écris le texte (qui tu es, le planning, les règles du chat…).
4. **Envoyer**, puis recommence pour les autres panneaux.

Les panneaux des réseaux (Discord, YouTube, TikTok, Instagram, X) : à l'étape 3, remplis aussi **Lien de l'image** avec l'adresse du réseau (ton invitation Discord `https://discord.gg/…`, ta chaîne YouTube…) : un clic sur l'image y mène. Tu n'as pas un de ces réseaux ? Retire son nom dans `reglages.html` › **Kit de chaîne Twitch** › Panneaux de bio. Un autre réseau (Twitch, Kick, Bluesky) : ajoute son nom, il prend son logo.

Le panneau Planning de la semaine : dans `reglages.html` › **Kit de chaîne Twitch** › **Planning**, écris une ligne par jour : `Mercredi | 20h30`, ou avec le jeu : `Mercredi | 20h30 | Minecraft`. Juste en dessous, choisis la **semaine** dans le calendrier, puis clique 🖼️ **Fabriquer l'image du planning** : la page enregistre le planning, OBS fabrique l'image en quelques secondes et la page te la montre. Elle est rangée dans `chaine/export/plannings/` (ex. `planning-2027-semaine-08.png`, avec « Semaine 8 · du 22 au 28 février » sous le titre) : **un fichier par semaine, les anciens restent**. Sur Twitch : ta chaîne › **À propos** › **Modifier les panneaux** › panneau Planning › change l'image.
Il faut **Edge ou Chrome**. L'image est fabriquée par **OBS** s'il est ouvert avec le script `actualiser-obs.lua` à jour (après une mise à jour de l'overlay, recharge-le : Outils › Scripts › ↻) ; **sans OBS** (pour préparer tes streams à l'avance), double-clique sur `fabriquer-images.cmd` dans le dossier de l'overlay quand la page te le propose : il prend la demande et la page affiche l'image. Tu peux aussi le lancer seul : un petit menu propose le planning d'une semaine (tape son numéro, ou Entrée pour cette semaine), les images du kit ou les vidéos de transition. Planning vide = le panneau titre « Planning ».

Le panneau Mes créations : dans sa description, mets tes liens (site, boutique, portfolio…), et dans **Lien de l'image** le principal. Tu préfères « Mes sites » (avec un globe) ? Renomme-le dans la liste des panneaux de `reglages.html`.

Le panneau Commandes : dans sa description, liste les commandes du chat, une par ligne avec ce qu'elle fait (ex. `!discord` : le lien du Discord · `!planning` : les jours de live). Elles se créent dans streamelements.com › **Chatbot** › **Chat commands** (le bot StreamElements doit être sur ta chaîne : **Chatbot** › **Join channel**). L'overlay cache déjà ces messages et les réponses du bot dans le chat de l'écran.

La page de dons StreamElements, pas à pas :

1. streamelements.com › **Revenue** › **Tipping settings**.
2. Dans l'apparence de la page : **Banner** (bannière) → `chaine/export/dons-banniere.png` ; **Background** (fond) → `chaine/export/dons-fond.png`.
3. **Save**, puis ouvre `streamelements.com/<ton pseudo>/tip` pour voir le résultat. Le formulaire de don se pose au milieu du fond : le décor est sur les côtés exprès.
4. Mets ce lien dans la description du panneau **Soutenir**.

Refaire les images (autre couleur, textes changés dans `reglages.html` › **Kit de chaîne Twitch**, dont le titre de la page de dons) : OBS › **Outils › Scripts › Refaire les images du kit Twitch** (2.8). Ou dans PowerShell :

```
cd G:\Projets\Overlay\mordethrhedan
node outils/exporter-chaine.mjs
```

> Les noms des menus Twitch changent parfois un peu : cherche « Marque » ou « Récompenses des spectateurs » dans le tableau de bord.

8. TESTER SANS ÊTRE EN LIVE

Option	Effet
`?test=1`	faux chat et fausses alertes qui défilent
`?apercu=1`	affiche les zones des sources
`?journal=1`	affiche le journal : connexion à StreamElements et derniers événements reçus (6.3)
`?objectif=sub` / `?objectif=follow`	la barre d'objectif affiche les abonnés / les followers, quel que soit le réglage (6.6)
`?chat=0`	retire le chat intégré à la scène
`?cam=0`	*(Jeu)* pas de webcam : le cadre de la cam disparaît
`?cam=1200,700,420,236`	*(Jeu)* webcam à une position perso : x, y, largeur, hauteur (pixels 1920 × 1080)
`?minutes=0.5`	*(démarrage)* compte à rebours de 30 s
`?couleur=rouge`	essaie une autre couleur

Le mode test **ne modifie pas** le vrai compteur de l'objectif.

9. PERSONNALISER

Presque tout se règle dans `reglages.html` (2.1), sans toucher au code.

Je veux changer…	Où
Le coin ou la position exacte de la webcam, pas de webcam, le chat d'un écran	`reglages.html` › Options des scènes
Une ambiance (Halloween, Noël…)	`reglages.html` › Couleur et fond (voir ci-dessous)
La couleur partout	`reglages.html` › Couleur et fond, ou `config.js` › `couleur` (`vert`, `rouge`, `bleu`, `violet`, `orange`, `cyan`, `jaune`, `rose` ou `"#FFD400"`)
La couleur d'une seule scène	`?couleur=rouge` dans l'adresse de l'overlay (pratique pour une scène « Jeu rouge »)
Le halo	`reglages.html` › Couleur et fond (léger, moyen, fort)
Le motif du fond / son animation	`reglages.html` › Couleur et fond
Les titres des écrans	`reglages.html` › Écrans avec un grand titre
Le vocabulaire des alertes (avec aperçu)	`reglages.html` › Alertes
Le son de chaque alerte, ou ton propre fichier	`reglages.html` › Sons des alertes (voir 6.8)
Les visuels de la chaîne	`reglages.html` › Kit de chaîne Twitch (puis `node outils/exporter-chaine.mjs`, voir « Les scripts »)
Changer l'objectif affiché, ou le compter à la main	`reglages.html` › Objectif (6.6)

Dans les textes des alertes, `{nom}`, `{montant}`, `{mois}`, `{nombre}` et `{destinataire}` sont remplacés automatiquement.

Changer d'ambiance (Halloween, Noël…) en un clic

1. Ouvre `reglages.html` › section **Couleurs** — la couleur des cadres et des éclats y est aussi, dans « Couleur et fond ».
2. Clique 🎃 **Halloween** (ou 🎄 **Noël**), ou change une couleur à la main (le nuancier, ou un code comme `#FF7A1A`). ↺ remet la couleur d'origine d'une seule couleur ; ↺ **Couleurs d'origine** les remet toutes.
3. **Enregistrer**, puis actualise les sources dans OBS (ou laisse faire le script de la section 2) : toutes les scènes, sources et alertes prennent ces couleurs.

Créer ta propre ambiance (ex. Batman) et la garder

1. Dans `reglages.html` › **Couleurs**, règle les couleurs comme tu veux (nuancier ou code).
2. Sous **Mes ambiances**, tape un nom (ex. `Batman`) puis clique 💾 **Sauvegarder ces couleurs** : l'ambiance est écrite tout de suite dans `mes-reglages.js`, avec son nom et ses couleurs. Même nom qu'une ambiance existante = elle est remplacée (la page demande confirmation).
3. Elle apparaît ensuite en bouton : **un clic** remet toutes ses couleurs, puis **Enregistrer** pour que l'overlay les prenne. × la supprime.

Sauvegarder une ambiance ne change pas les couleurs de l'overlay : seul **Enregistrer** le fait.

Les **vidéos de transition** et les **images du kit de chaîne** sont déjà fabriquées : pour qu'elles prennent les nouvelles couleurs, refais-les : OBS › **Outils › Scripts** › **Refaire les vidéos de transition** puis **Refaire les images du kit Twitch** (2.8) ; `reglages.html` te le rappelle quand tu enregistres. Et quand tu reviens aux couleurs d'origine, pareil. (Sans OBS : les scripts ci-dessous.)

Les scripts du dossier `outils/`

Ces scripts **refont les fichiers « fabriqués »** : vidéos de transition, images du kit, PDF… On ne s'en sert qu'après un changement (couleurs, textes du kit, tuto modifié). Tout le reste (réglages, textes, alertes) se fait dans `reglages.html`, sans script.

Une seule fois : installer Node.js (le programme qui lance les scripts `.mjs`)

1. Va sur le site officiel **nodejs.org**, télécharge la version **LTS** et installe-la (tout laisser par défaut, *Suivant* jusqu'au bout). Ou, dans PowerShell : `winget install OpenJS.NodeJS.LTS`
2. Pour **refaire les vidéos de transition**, il faut aussi **ffmpeg** : dans PowerShell, `winget install Gyan.FFmpeg`.
3. Ferme puis rouvre PowerShell.

Lancer un script

1. Ouvre le dossier `mordethrhedan` dans l'Explorateur Windows.
2. Clic droit dans un espace vide du dossier › **Ouvrir dans le Terminal** (ou tape `powershell` dans la barre d'adresse du dossier, puis **Entrée**).
3. Tape la commande du tableau, puis **Entrée**. Le script dit ce qu'il fait, puis rend la main.

Script	À quoi il sert	Quand	Commande
`actualiser-obs.lua`	bouton « Actualiser toutes les sources Navigateur » dans OBS, bouton « Placer les webcams » (et replacement automatique), et actualisation automatique quand tes réglages changent	une fois, à installer dans OBS (section 2)	*pas de commande :* OBS › Outils › Scripts › +
`fabriquer-images.cmd`	fabrique sans OBS : le planning d'une semaine (et répond au bouton des réglages), les images du kit, les vidéos de transition	quand OBS est fermé	double-clic (dossier de l'overlay)
`mettre-a-jour.cmd`	met l'overlay à jour depuis le serveur (double-clic), en gardant tes réglages ; c'est aussi ce que fait le bouton d'OBS	quand une nouvelle version est annoncée	
`generer-transitions.mjs`	refait les vidéos `transitions/videos/*.webm` (Stinger)	après un changement de couleurs	`node outils/generer-transitions.mjs` *(ffmpeg nécessaire)*
`exporter-chaine.mjs`	refait les images `chaine/export/*.png` (profil, bannière, panneaux, emotes, badges)	après un changement de couleurs ou des textes du kit	`node outils/exporter-chaine.mjs`
`generer-pdf.mjs`	refait `TUTO.pdf` et `CONCEPT.pdf` depuis les `.md`	après une modification de `TUTO.md` ou `CONCEPT.md`	`node outils/generer-pdf.mjs`
`cadre-image.mjs`	fabrique le cadre néon en image PNG transparente, à la taille voulue (4.4)	pour une source Image au lieu de la page	`node outils/cadre-image.mjs 380x290`
`capturer.mjs`	fait une capture PNG d'une page (pour vérifier une animation, ou l'envoyer)	pour vérifier	`node outils/capturer.mjs "scenes/jeu.html?test=1"`
`polices-locales.mjs`	copie dans `assets/polices/` les polices du thème (pour ne plus dépendre d'internet)	seulement si on change de police (déjà fait)	`node outils/polices-locales.mjs` *(internet nécessaire)*

Les scripts se servent de **Microsoft Edge** en coulisses (déjà installé avec Windows) : rien d'autre à installer.

10. DÉPANNAGE

Le bouton « Fabriquer l'image du planning » ne fait rien
→ La page dit ce qui bloque, sous le bouton. « Personne n'a pris la demande » : double-clique sur `fabriquer-images.cmd` (sans OBS) ; ou, dans OBS, recharge le script `actualiser-obs.lua` (Outils › Scripts › ↻) et vérifie que son **Dossier de l'overlay** est bien celui que tu as choisi dans `reglages.html`. Il faut Edge ou Chrome (pas Firefox).

« Mettre à jour l'overlay » ne marche pas
→ Le message est sous le bouton (OBS › Outils › Scripts) et dans **Journal des scripts**. « Serveur injoignable » : vérifie ta connexion, ou l'adresse dans `reglages.html` › **Mises à jour**. « Certains fichiers n'ont pas pu être remplacés » : OBS les utilise (souvent les vidéos de transition) → ferme OBS, double-clique sur `mettre-a-jour.cmd`. Si le dossier de l'overlay a des accents dans son chemin et que le bouton ne fait rien : utilise `mettre-a-jour.cmd`. Rien n'est perdu : l'ancienne version est dans `sauvegardes`.

Mes réglages ont disparu après une mise à jour de l'overlay
→ Ton `mes-reglages.js` a peut-être été remplacé ou supprimé : il ne doit **pas** faire partie des fichiers que tu copies. S'il te reste une copie, remets-la dans le dossier de l'overlay.
→ Si tes réglages étaient encore dans l'ancien `config.js` (avant `mes-reglages.js`) : `reglages.html` › 📥 **Reprendre les réglages d'un ancien config.js** › choisis une copie de cet ancien fichier, puis **Enregistrer** (section 2.1).

Page blanche / rien ne s'affiche
→ Vérifie `1920` × `1080` sur la source, puis clic droit › **Actualiser**. Si tu viens de modifier `config.js`, vérifie les guillemets et les virgules.

Les textes ne sont pas dans la police Dyer
→ Clic droit sur la source › **Propriétés** › **Actualiser le cache de la page actuelle** (OBS garde parfois l'ancien `css/theme.css`).

Le chat n'affiche rien
→ Vérifie l'identifiant Twitch dans `reglages.html` › **La chaîne** (l'identifiant exact, en minuscules). Le chat n'affiche que les messages envoyés **après** l'ouverture de la page, et il a besoin d'internet.

Les alertes ne s'affichent pas
→ Le jeton StreamElements est-il collé dans `reglages.html` (6.2) ? Regarde le journal (6.3, `?journal=1`) : il dit ce qui coince (pas de jeton, jeton refusé, pas d'internet). Puis actualise la source d'alertes.

Les réglages ne changent rien dans OBS
→ Après **Enregistrer** dans `reglages.html`, il faut actualiser les sources : clic droit › **Actualiser**, ou le script `actualiser-obs.lua` (section 2) qui le fait tout seul.

Pas de son sur les alertes
→ Coche **Contrôler l'audio via OBS** sur la source d'alertes et vérifie son volume dans le mélangeur.

Une alerte s'affiche mal (pseudo manquant, « ? »…)
→ OBS n'a pas de console (F12) : ajoute `?journal=1` à l'adresse de la source d'alertes (décoche *Fichier local*, colle l'adresse de la page suivie de `?journal=1`). Un panneau affiche la connexion à StreamElements (« ✅ Connecté ») et chaque événement reçu avec ses données brutes : fais-en une capture d'écran pour faire corriger l'overlay, puis retire `?journal=1`.

Une ancienne image s'affiche encore après une modification

→ Dans le navigateur : **Ctrl + F5** (recharge en ignorant la mémoire du navigateur).

→ Dans OBS : clic droit sur la source › **Propriétés** › **Actualiser le cache de la page actuelle**.

La cam ne tombe pas pile dans le cadre

→ Le plus simple : OBS › **Outils › Scripts** › `actualiser-obs.lua` › **Placer les webcams sur toutes les scènes** (le nom de ta webcam doit contenir « cam »). Sinon :

→ Refais le geste B, et vérifie que la webcam est bien **sous** l'overlay.

Bon live ! 🎮