Patagrain — Concept

> Résumé vivant de la demande. **Mis à jour à chaque échange** : nouvelle demande, choix validé, changement d'avis. Le haut du fichier décrit l'état actuel ; le journal en bas garde l'historique.

La chaîne

Streamer	Patagrain (l'utilisateur lui-même)
Identifiant Twitch	`patagrain` — **à confirmer** (Twitch n'a pas confirmé que la chaîne existe)
Ce qui est streamé	Jeux vidéo, et de temps en temps des parties de jeu de rôle
Écran / canevas OBS	2560×1440 (sortie conseillée 1920×1080, Lanczos) — les pages s'adaptent
Alertes	StreamElements (par internet) : compte + jeton à coller dans reglages.html
Réseaux	Aucun pour le moment

Direction artistique

- **Univers** : médiéval-fantaisie décalé — le **bouffon du roi** (pas l'insulte !) mélangé au **jeu de rôle / donjons**. Style **flat**, dans l'esprit du logo.
- **Logo** : écriture gothique calligraphiée, l'épée fait le « t » (manette retirée), **le chapeau du bouffon posé sur le « n »** — **validé** (l'ancien logo est gardé dans `design/archives/`). **Un seul chapeau partout** — dessiné pour être porté : largeur du crâne, pointes anguleuses bleues, pointe centrale noire soulignée d'un fin liseré gris clair et terminée par un **d20**, couronne dorée courbe, grelots d'or. Généré par `outils/generer-bouffon.mjs` (bouffon, `assets/chapeau.svg`, logo).
- **Le bouffon (mascotte)** : dessiné d'après ses photos, style flat — cheveux longs raides châtain clair juste sous les épaules, raie au milieu, lunettes rectangulaires noires, yeux vert-gris, collier de barbe fin, anneau à l'oreille gauche, grain de beauté ; **habit mi-bleu mi-noir** (buste et personnage en pied identiques), chapeau noir et bleu. `js/bouffon.js` (buste, tête pour les emotes, personnage en pied articulé), `assets/bouffon.svg`, `assets/bouffon-pied.svg`.
- **Emblème** : d20 low-poly avec un « P », en 3 variantes (bleu royal, **bleu glacier** — remplace « rouge bouffon » —, nuit étoilée) — pour la photo de profil et les petits formats.
- **Palette** : **« Royal bleu & or »** (validée) — bleu Patagrain `#3A9AD9`, bleu nuit `#152238`, or `#F5B82E`, et **bleu glacier** `#8FD0F5` **à la place du rouge** (badge « en direct » en texte foncé, fanions, dés). Plus aucun rouge, sauf le feu de camp de la Pause qui garde de vraies couleurs de feu.
- **Polices** : titres Grenze Gotisch, texte Nunito — **en local** dans `assets/polices/` (plus besoin d'internet).

- **Motifs** : fanions, grelots, dés (d4/d6/d8/d20), chapeau, épée, blason, parchemin. **Arlequin (damier) à éviter en grand** : jugé trop flashy, les transitions sont en aplats unis avec une touche d'or.
- **Ton / vocabulaire des alertes** : cour du roi — « Nouvel aventurier » (follow), « Adoubement » (sub), « Chevalier fidèle » (resub), « Présent royal » / « Largesse royale » (cadeaux), « Tribut au bouffon » (bits), « Une horde débarque » (raid), « Offrande royale » (don).
- **Sons** : tintement de grelots synthétisé, petite fanfare pour les gros événements.

Ce qui est demandé

- **Écrans** : Starting soon (« Le spectacle va commencer », compte à rebours « Jet d'initiative »), Pause (« Repos court », feu de camp, chat « La taverne »), Fin (« Fin de la session », carte « Prochaine quête »).
- **Scènes** : Cam seule (l'écran « blabla » : accueil, cam en grand + chat), Contenu (contenu, cam et chat côte à côte), Jeu (jeu plein écran, petite cam dans le coin de son choix, chat en transparence, barre fine).
- **Sources** : alertes, chat, bandeau d'infos, objectif « Guilde des aventuriers » (l'épée avance), cadre cam (chapeau au-dessus, dés en bas).
- **Transitions** (toutes avec le bouffon) : **rideau** — le bouffon machiniste tire la corde (Starting soon → Cam seule, vers la Fin) ; **coup d'épée** — le bouffon bretteur tranche l'écran (Cam seule → Jeu) ; **jet de dé** — le bouffon équilibriste marche sur le dé qui roule (Jeu ↔ Pause, Contenu). La page de grimoire est abandonnée.
- **Chaîne Twitch** : bannière, écran hors-ligne, panneaux de bio, emotes, badges d'abonné — fait (`chaine/`), à valider.
- **Le bouffon sur chaque écran** (validé, intégré) : Starting soon = numéro de cirque (jongle, poirier, salut) et, à zéro, il lance un d20 qui retombe sur 20 ; Pause = il rêve au coin du feu d'un d20 qui tombe sur 1, se réveille, grille un chamallow au bout des flammes, le croque devant sa bouche et se rendort ; Cam seule et Contenu = il dépasse du haut de la carte du chat (tête, chapeau, deux mains), ses grelots tintent à chaque message ; Alertes = la carte descend du plafond sur deux cordes, lui sur une troisième, une main levée, il salue de l'autre ; Jeu = le chapeau posé sur la cam est le sien, il sort la tête toutes les 60 s et à chaque follow ; Fin = coucou, courbette, saut de joie, bulle qui alterne ses phrases. Ses cheveux de dos ne bougent qu'en réaction à ses mouvements. Briques : `css/bouffon.css` , `js/numeros.js` ; réglages `config.js › bouffon` , `?bouffon=0` par page.
- **Réglages sans toucher au code** : `reglages.html` , un formulaire (par écran, avec aperçu des alertes) qui réécrit `config.js` (Edge/Chrome : écriture directe ; sinon téléchargement). Le modèle du fichier est dans `js/reglages.js` : tout nouveau réglage doit y être ajouté.
- **Chat et bandeau intégrés aux scènes** (décision) : gardés dans les scènes (placés au pixel près à côté des trous, le bouffon accoudé au chat) ; `?chat=0` / `?bandeau=0` pour les retirer et utiliser `sources/chat.html` / `bandeau.html` à la place. Le chat garde en mémoire ses messages des 10 dernières minutes (`chat.memoireMinutes`) : il ne repart pas à vide en changeant de scène.
- **Contraintes** : tout en français, facilement éditable.

Décisions prises

- 2026-09-29 — Palette Bouffon royal + Grenze Gotisch par défaut (modifiable en une ligne dans `theme.css`).
- 2026-09-29 — Damier d'arlequin retiré des transitions (trop chargé) ; reste en petit dans quelques liserés (à atténuer si demandé).
- 2026-09-29 — Pages dessinées en 1920×1080 et mises à l'échelle : nettes en 1440p.
- 2026-09-30 — Scène « blabla » renommée `cam-seule` (harmonisation entre overlays).
- 2026-09-30 — Le bouffon et son chapeau sont noir et bleu, et **ce chapeau est le même partout** (logo compris).
- 2026-09-30 — **Logo refait validé** (chapeau du bouffon sur le « n »).
- 2026-09-30 — Palette **Royal bleu & or** retenue : plus de rouge (emblème rouge → bleu glacier).
- 2026-09-30 — Les transitions bouffon **remplacent** rideau, épée et dé ; la page de grimoire est retirée.

État d'avancement

Élément	État
Moodboard (+ pistes de logo `design/logos.html`)	✅
Écrans et scènes (avec le bouffon)	✅
Page de réglages `reglages.html`	✅ (écriture du fichier testée seulement en génération, pas le clic « Enregistrer » dans Edge)
Options des scènes (webcam ou pas, coin, chat, bandeau, bouffon) dans `reglages.html`	✅ (vérifié dans le navigateur, pas dans OBS)
Un son par alerte + sons perso (`reglages.html` › Sons des alertes)	✅ (vérifié dans le navigateur, pas dans OBS)
Webcam placée toute seule dans OBS (script `actualiser-obs.lua`) + position perso en Jeu + Options des scènes en tableau	✅ navigateur + faux OBS ; pas testé dans le vrai OBS
Sources	✅
Transitions bouffon (vidéos Stinger : rideau 1400 ms, épée 700 ms, dé 1750 ms)	✅
Kit chaîne Twitch (`chaine/`, 54 PNG dans `chaine/export/`)	✅ (à valider)
TUTO.md	✅ (coordonnées en 1080p et 1440p ; §6 : chat, StreamElements, chaque événement, dons, objectif)
Scène JDR (`scenes/jdr.html`, moteur commun `js/partie.js`) + CodexVTT (mode spectateur, `js/codex.js`)	✅ **testé en vrai par le streamer** (OBS + CodexVTT)
Cadres des personnages (scène JDR) + pupitre des PV	✅ testé dans un navigateur ; ⬜ pas encore dans OBS
Testé dans OBS / avec le vrai StreamElements	⬜ (testé avec un faux serveur StreamElements)

À faire / questions ouvertes

- ☑ **Scène JDR — « La Malédiction de Pangée »** (tuto à part : `TUTO-JDR.md` / `.pdf`) : disposition **A** (table CodexVTT 1500 × 844 ; cartes des héros dessous ; à droite le titre de la campagne puis le chat, l'arène des jets passe par-dessus le chat pendant un jet). Héros dessinés dans le style du bouffon (`outils/generer-heros.mjs` → `js/heros.js` : Patagrain MJ sans chapeau, Zokhran, Namerya, Kenai, Trayss en 3 formes — **forme choisie dans le tableau des réglages**, plus automatique —, Ba Heng Ze, Dvergrost, et l'aventurier encapuchonné pour « Autre » / les PNJ). Table de jeu réglée (`jdr.places` : MJ + 6 places, personnage, nom, sous-titre, pseudo Discord) ; un perso CodexVTT sans place s'ajoute en

bout de table. CodexVTT (lecture seule) : jets joués par le héros (élan tout de suite, dénouement quand les dés se posent, résultat `dureeResultat` s), PV / à terre, coups et soins, tour de combat, PNJ (jets publics). Discord : bot « voix » du VPS (salons permis `SALONS`, seulement quand la scène JDR est ouverte dans OBS). Branchement par le moteur commun `js/partie.js` (Patagrain ne fait que dessiner). **Testé en vrai par le streamer, ça marche.**

- ☐ **Ensuite, dans l'ordre** (décidé le 2026-10-14) : 1. la session d'harmonisation finit les scènes JDR de John Von Gurt et d'Ambries (puis on envoie tout sur GitLab) ; 2. la **fiche d'apparence** commune des personnages (dessinée par chaque overlay à son style ; `heros.js` = le style Patagrain) ; 3. la **liste des commandes du chat** pour le JDR (sondages, bénir, maudire…). L'**overlay musique** = la prochaine version. Vue joueurs de CodexVTT : **annulée**.
- ☑ **Les Grelots — spectacles à la demande** (monnaie StreamElements de la chaîne : les « Grelots ») : un spectateur dépense des grelots avec une commande StreamElements à coût ; le bot répond dans le chat, l'overlay repère la réponse et le bouffon joue un spectacle par-dessus la scène. **Fil rouge : la rivalité avec DARKTAGRAIN**, le bouffon maléfique (même dessin, habit violet et noir, yeux rouges, rictus). **Prix validés** : **10** la peau de banane (posée par Darktagrain ; Patagrain glisse, se relève furieux et le poursuit) · **25** la tarte à la crème (lancée par Darktagrain ; Patagrain s'essuie, goûte, « Miam ! », Darktagrain enrage) · **50** la sérénade (au **banjo**, il chante faux… un grelot lui tombe sur la tête) · **75** le jet du destin (d20 géant, vrai tirage : danse + confettis sur 20 ; sur 1, dé sur le pied, cloche-pied, chapeau sur les yeux ; pouce levé 11-19, haussement d'épaules 2-10) · **100** la potion mystère (au hasard : géant, minuscule, vert ou ballon, puis « pouf ») · **125** pierre-feuille-ciseaux contre Darktagrain (au hasard, la pierre bat les ciseaux, les ciseaux battent la feuille, la feuille bat la pierre ; égalité = on rejoue ; gagnant qui danse, perdant qui trépigne ou boude) · **150** le coffre au trésor (au hasard : légendaire 20 % — couronne du roi, épée légendaire ; commun 45 % — chope, fromage ; nul 35 % — vieille chaussette, poisson pas frais) · **175** la catapulte (pile ou face : pile, Patagrain envoie Darktagrain ; face, l'inverse — puis atterrissage au hasard parmi 7 : sur ses pieds, la tête dans la paille, écrasé contre l'écran, la mare, la bouse, chez les Dames du Lac [des sirènes : longs cheveux, coquillages, rien de visible ; cœurs pour Patagrain, « Ouste ! » pour Darktagrain], à travers le toit de la taverne [il ressort une chope à la main] ; raté = le lanceur rigole, réussi = il enrage) · **200** le duel (Darktagrain désarmé s'enfuit « chercher son dragon ») · **300** le dragon, **Darktagrain sur son dos**. **Commandes mystère** : une par prix, `!spectacle10` … `!spectacle300` (StreamElements : coût fixe par commande, pas de montant libre ; pas de `!duel`, déjà pris par le duel de points SE) ; le bot répond « 🔔 ${user} jette N grelots dans le chapeau du bouffon… 🎭 » : l'overlay repère cette phrase exacte dans le chat (seul le bot compte) et joue le spectacle dans la source **Alertes**, dans la même file que les alertes. `!essai <spectacle>` dans le chat = gratuit pour le streamer (modos en option). Réglages : `config.js › grelots` + `reglages.html` › Les Grelots ; TUTO § 6.9. Démos dans le moodboard (boutons pour forcer chaque résultat). Briques : `js/grelots.js`, `js/dragon.js`, `css/bouffon.css` § 7, `Chat.ecouter` (js/chat.js). **Pas encore vérifié sur un vrai live** (testé avec un faux chat et un faux bot).
- ☐ Confirmer l'identifiant Twitch exact (`reglages.html` › La chaîne).
- ☐ Brancher StreamElements (TUTO 6.2 : compte + jeton dans reglages.html).
- ☐ Regarder les écrans avec le bouffon en continu (rythme, taille) et dire ce qui va ou pas.
- ☐ Vérifier les premiers vrais événements StreamElements avec le journal (`?journal=1`).
- ☐ Valider le kit de chaîne Twitch (`chaine/kit.html`) puis l'envoyer sur Twitch (TUTO §7).

Journal

- **2026-09-29** — Logo fourni, direction bouffon + JDR, liste des écrans et scènes. Moodboard (3 palettes, 3 polices, motifs, vocabulaire). Dés redessinés en low-poly, emblème d20 décliné, logo finalisé.
- **2026-09-29** — Transitions rendues plus sobres (plus de damier). Construction complète : écrans, scènes, sources, 4 transitions exportées en vidéo, guide et tuto. Adaptation 1440p.
- **2026-09-30** — Harmonisation avec les autres overlays : dossier `Overlay/patagrain/`, fichiers communs, `cam-seule.html`.
- **2026-09-30** — Kit de chaîne Twitch créé (profil, bannière, hors-ligne, 4 panneaux, 6 emotes, 5 badges). TUTO réécrit fiche par fiche, en PDF ; `index.html` devient une vitrine.
- **2026-09-30** — Retours : ses couleurs sont le **noir et le bleu** (nouveau thème à proposer) ; transitions dé, épée et rideau validées, **page à remplacer** par un bouffon animé à son image (6 photos fournies).
- **2026-09-30** — Moodboard : ajout du thème **Noir & bleu** et du **bouffon** (mascotte + transition), à valider avant production.
- **2026-09-30** — Retours sur le bouffon : chapeau qui flottait, yeux vert-gris, collier de barbe (pas un bouc), cheveux plus courts ; en Noir & bleu, le logo ne doit plus avoir de rouge ; envie d'animations travaillées (bouffon qui marche sur le dé, bouffon qui donne un coup d'épée). → Bouffon corrigé, version en pied articulée, transitions « équilibriste » et « bretteur » dans le moodboard, logo Noir & bleu.
- **2026-09-30** — Retours : chapeau à élargir et poser sur la tête (sans la bande), plus de joues roses, 2e oreille, barbe plus fine, bretteur tronqué ; demande d'une version « rideau » avec le bouffon qui tire une corde. → Corrigé, transition « machiniste » ajoutée au moodboard.
- **2026-09-30** — Décisions : palette Bouffon royal conservée, bouffon et chapeau noir et bleu **partout** (logo, icônes, cadre cam, kit), chapeau du bouffon élargi (plus large que la tête). Transitions bouffon produites (`transitions/rideau|epee|de.html` + vidéos), page de grimoire retirée (config, vitrine, tuto, vidéo). Kit de chaîne réexporté.
- **2026-09-30** — Retours : chapeau « calque » (pas posé sur la tête) → redessiné pour être porté (calotte, pointes attachées, couronne courbe) ; puis plus large et moins rond, **d20 au bout de la pointe centrale** ; buste aux mêmes couleurs que le personnage en pied ; logo à revoir → refait avec ce chapeau ; palette à revoir → 2 variantes sans rouge proposées. Vidéos de transition et kit de chaîne régénérés.
- **2026-09-30** — Retours : chapeau à la largeur du crâne (élargi puis réduit de 10 px par côté) ; palette Royal bleu & or validée, plus de dés ni d'emblème rouges ; liseré de la pointe noire : bleu jugé moche → gris clair discret. Générateur du bouffon rangé dans `outils/generer-bouffon.py`. Vidéos, kit et docs régénérés.
- **2026-09-30** — Logo refait validé (« il est top ! ») ; révérence retirée du moodboard (inutilisée).
- **2026-09-30** — Générateur du bouffon porté en Node (`outils/generer-bouffon.mjs`, sortie identique à l'ancienne version Python, supprimée) pour suivre la convention commune.
- **2026-09-30** — Polices passées en local (`assets/polices/`) ; logo à contour crème (`assets/logo-contour.svg`, utilisé en scène Jeu) ; 4 emotes à sa tête (bouffon, clin d'œil, mort de rire, choqué) ; expressions « endormi » et « sans chapeau » ajoutées au dessin ; correction : les facettes claires du mini-d20 du logo s'affichaient en noir. Idées du bouffon sur chaque écran → 6 démos dans le moodboard, à valider.

- **2026-09-30** — Retours sur les démos du bouffon : cirque trop discret (amplifié, démos agrandies), Pause → histoire du rêve (d20 qui tombe sur 1, réveil, chamallow), alertes → une main sur la corde, décalé sur le côté, Jeu → c'est son propre chapeau qui est sur la cam, Fin → plus de mouvement. Cam seule validée. Vitrine : ajout des aperçus chat, bandeau et cadre cam ; transitions rejouées en boucle.
- **2026-09-30** — Retours sur la démo Pause : le chamallow se voyait derrière (sa tête et ses cheveux le cachaient quand il croquait) → le bras droit passe devant la tête, le chamallow glisse du bâton et arrive devant sa bouche ; bâton allongé pour que le chamallow grille bien au bout des flammes. « Le fond de ses cheveux ne bouge pas » → les cheveux de dos se balancent en permanence et rebondissent aux sauts, au sursaut, en jonglant et à chaque message (`.bf-cheveux` , dessin régénéré).
- **2026-09-30** — Retours : le chamallow n'était pas devant sa bouche → bras recalculé, il arrive pile sur la bouche et y reste plus longtemps (bouchée en deux temps). « Les cheveux qui pendent au rythme du chapeau, c'est bizarre » → plus de balancement permanent (il avait le même tempo que le chapeau) : les cheveux ne réagissent qu'à ses mouvements. Animations validées (« top ! ») → **intégrées aux vrais écrans**, aux alertes et à la scène Jeu. Question « chat et bandeau en dur ou en .html ? » → gardés dans les scènes, options `?chat=0` / `?bandeau=0` , et mémoire du chat entre scènes. Demande d'une page pour gérer la config → `reglages.html` . TUTO complété (chat, Streamer.bot, chaque événement, dons, objectif, tests). Corrigé au passage : l'épée de l'alerte Raid débordait de sa case. Outils communs : variable `NAVIGATEUR` (un autre navigateur qu'Edge), recopiés dans les 4 overlays.
- **2026-09-30** — Retours : « le chapeau se décale de la tête » (Cam seule, Contenu) → corrigé : sur le buste, le chapeau n'avait pas de pivot et tournait autour du coin du dessin quand ses grelots tintaient (jusqu'à 11 px de décalage, 0,3 px maintenant). « La page de réglages doit être dans chaque overlay, un seul reglages.html à la source qui prend le thème et le config de chacun ? » → oui : `reglages.html` + `js/reglages.js` sont maintenant des fichiers communs (`_modele/` , copiés dans les 4 overlays), aux couleurs du thème de chacun ; chaque overlay n'a que ses libellés (`js/reglages-champs.js`). La page ne remplace plus tout le fichier : elle modifie seulement les valeurs changées dans `config.js` (commentaires gardés) et refuse le config.js d'un autre overlay.
- **2026-09-30** — Objectif réglé sur les abonnements : une pluie d'abonnements offerts compte maintenant pour tous ses cadeaux (avant : 0). PDF lisibles hors ligne (police du thème si Nunito manque), régénérés.
- **2026-09-30** — Emotes 7TV / BTTV / FFZ : pas prévues (les chaînes n'utilisent pas ces extensions).
- **2026-09-30** — Retours Patagrain : plus de chapeau sur la cam en Cam seule et Contenu (le bouffon est déjà sur le chat) et plus d'en-tête (logo + « En direct ») : le logo remplace la plaque « Patagrain » sous la cam, cam remontée (y 100) ; scène Jeu : le bouffon sort environ toutes les 3 min (au hasard ± 25 %) et à chaque follow, et il a maintenant une vraie disparition (élan, plongeon sous le chapeau qui tangue) ; bandeau : cases au choix dans les réglages (dons/bits, abonnés…) ; kit : photo de profil = le bouffon (emblème gardé en autre choix), écran hors-ligne = le bouffon endormi debout, panneau « Matériel » ajouté. Commun : « pas de F12 dans Interagir » → journal à l'écran `?journal=1` (connexion Streamer.bot + événements bruts) ; client Streamer.bot en copie locale (plus besoin d'internet pour les alertes) ; script OBS `outils/actualiser-obs.lua` (actualiser toutes les sources, et tout seul quand config.js change) — non testé dans OBS.
- **2026-09-30** — Retour : « le logo près de la cam c'est bien, mais le cadre jaune était bien » → le logo est maintenant DANS la plaque dorée sous la cam. Réglage du bandeau renommé avec des noms communs à tous les overlays (`bandeau.follow` , `abonne` , `soutien` , `objectif` , `ceSoir`).

- **2026-09-30** — Demande : changer les couleurs depuis les réglages (ex. thème Halloween, orange à la place du bleu), sur tous les overlays → `config.js › couleurs` + `js/couleurs.js` (commun) ; `reglages.html` › Couleurs : nuanciers et ambiances en un clic (🎃 Halloween, 🎄 Noël, ↺ Couleurs d'origine). Limite : le logo, le chapeau des cams et l'emblème sont des images et gardent leurs couleurs. Vidéos de transition et kit à refaire après un changement de couleurs.
- **2026-09-30** — TUTO mis à jour : section « Les scripts » (installer Node.js et ffmpeg, lancer un script, tableau de tous les scripts dont actualiser-obs.lua).
- **2026-09-30** — Demande : une option `?cam=0` (pas de webcam) → ajoutée (plus de cadre de cam, le chat récupère la place). Puis « cam et tout, on devrait pouvoir le régler depuis les réglages » → `config.js › options` + `js/options.js` (commun) ; `reglages.html` › **Options des scènes** : le coin de la webcam de la scène Jeu (ou « Pas de webcam »), la webcam de Contenu, le chat, le bandeau et le bouffon de chaque scène. Une option écrite dans l'adresse d'une source OBS passe avant.
- **2026-09-30** — Bug signalé : en Cam seule, remettre le chat (et les couleurs) dans les réglages ne marchait pas après actualisation → `js/options.js` ajoutait `?chat=0` à l'adresse, et l'actualisation d'OBS rechargeait cette adresse modifiée. Corrigé : les options ajoutées par les réglages sont notées (`depuisReglages=…`) et retirées au chargement suivant. Couleurs : l'écriture et l'affichage vérifiés (Halloween puis couleurs d'origine) ; pas testé dans OBS.
- **2026-09-30** — Réglages : **« Mes ambiances »** (section Couleurs) — nommer les couleurs affichées puis 💾, gardées dans `config.js › ambiances` (ex. Batman) ; un clic les remet, × les supprime. TUTO §9 mis à jour.
- **2026-09-30** — Standardisé dans les 4 overlays : **heure fixe** du compte à rebours (`demarrage.heure` / `?heure=20:30` , réglages › Démarrage › « … ou heure fixe ») et **étiquettes de placement** (`afficherZones` / `?zones=1` , réglages › Options des scènes). TUTO §2 et §3.1 mis à jour.
- **2026-09-30** — Demande : des sons différents selon l'alerte, pour les reconnaître en jouant → `js/son.js` : un son par alerte (follow, abonnement, réabonnement, abonnement offert, pluie d'abonnements, bits, raid, don, objectif), dans le thème (tableau dans TUTO 6.8). Et ses propres fichiers possibles : dossier `sons/` + `config.js › alertes.sons` ; `reglages.html` › **Sons des alertes** avec ▶ pour écouter (« aucun » = silence). Vérifié dans le navigateur (sons rendus hors ligne, fichier perso joué) ; pas écouté dans OBS.
- **2026-10-01** — Demande : « quand je donne un nouveau dossier, son config est écrasé » → réglages perso séparés : `config.js` = valeurs par défaut (remplacé à chaque mise à jour), `mes-reglages.js` = ce que le streamer change dans reglages.html (jamais livré, gardé lors d'une mise à jour, `.gitignore`). Migration unique : reglages.html › « 📩 Reprendre les réglages d'un ancien config.js ». TUTO §2.1 (« Quand tu reçois une nouvelle version »). Pas testé dans OBS.
- **2026-10-01** — Demandes : « la cam de l'overlay bouge mais pas la source caméra dans OBS », « un script qui place toutes les cams sur toutes les scènes », « Options des scènes trop fouillis », « la cam de Jeu à un X/Y précis, en gardant les préréglages, et voir leurs coordonnées ». → `js/zones.js` (les zones de la webcam, seule source des chiffres) + `Options.cam()` (commun) ; scène Jeu : préréglage, position perso `{ x, y, l, h }` (`?cam=x,y,l,h`) ou pas de webcam. `reglages.html` › Options des scènes refait : un tableau scènes × éléments avec interrupteurs, et un bloc « La webcam de la scène Jeu » (préréglages avec coordonnées, position perso, plan où l'on fait glisser la cam, zones fixes des autres scènes). Script OBS `actualiser-obs.lua` : bouton « Placer les webcams sur toutes les scènes (et les

ajouter là où elles manquent) » + replacement automatique quand les réglages changent. Testé hors OBS (faux OBS, 1440p, groupes, carte d'acquisition non touchée) ; **pas testé dans le vrai OBS**.

- **2026-10-01** — Retours : en scène Jeu, le logo n'était pas dans la plaque dorée sous la cam (comme Contenu et Cam seule) → plaque + logo remis sous la cam de Jeu, et le petit logo en haut d'écran retiré (doublon ; il ne reste que sans webcam). Starting soon : « trop de dés » → retirés les deux d20 autour de « ce qu'on fait ce soir » et celui devant « Jet d'initiative » (le d20 que lance le bouffon à zéro et les dés flottants du fond restent).
- **2026-10-02** — Demande : « ne plus envoyer de zip à la main ; un serveur sur mon VPS ; un push = mis à jour ; et quand eux modifient ? » → dossier `serveur/` (à la racine du dépôt) : le VPS publie une page de téléchargement + un zip et un `version.json` par overlay, refaits à chaque push (webhook GitHub ou GitLab, + vérification horaire), installés par `serveur/installer.sh` (notice : `serveur/LISEZMOI.md`). Côté overlay : bouton **Mettre à jour l'overlay** dans le script OBS `actualiser-obs.lua` (ou `mettre-a-jour.cmd`) : dernière version installée, `mes-reglages.js` gardé, ancienne version dans `sauvegardes\`. `reglages.html` › **Mises à jour** (adresse du serveur). TUTO 2.1, 2.8, scripts, dépannage. Testé : installation en conteneur, webhook, mise à jour avec PowerShell 7 ; **pas testé** : sur le vrai VPS, sous Windows, dans OBS.
- **2026-10-02** — Choix : le dépôt publié par le serveur sera sur GitLab (https://gitlab.com/Bastien.Lambour/overlays, branche main), webhook GitLab. `serveur/installer.sh` et `serveur/LISEZMOI.md` réglés dessus par défaut.
- **2026-10-02** — Le VPS héberge déjà d'autres sites (Nginx, certificat bastien-lambour.fr) → le serveur des overlays sera sur **https://overlays.bastien-lambour.fr** (même certificat, 80 → 443), dans `/var/www/overlays` (`depot/` privé, `site/` servi) ; `installer.sh` n'ajoute qu'un fichier Nginx et ne touche à aucun autre site. `miseAJour.adresse` = cette adresse. À faire : l'enregistrement DNS `overlays` → 217.154.115.223.
- **2026-10-02** — Demande (et test du bouton « Mettre à jour ») : sur Pause, le d20 du rêve tombait toujours sur 1 → tirage **au hasard** (1 à 20) : il dort tant que ça ne fait pas 1 ; sur un 1, réveil en sursaut, chamallow, croque, puis il se rendort (≈ un réveil toutes les 2-3 min ; `?test=1` : le 2e lancer fait 1). Vérifié dans le navigateur (tirages variés, réveil sur 1).
- **2026-10-02** — Demande : « une barre d'objectif pour les follows aussi ? avec les vrais nombres de Twitch, pas à la main ; plutôt un interrupteur : la barre affiche les followers OU les abonnés, chacun son compteur et son objectif » → **une seule barre**, `config.js › objectif` : `affiche` ("follow" ou "sub"), `automatique`, et `follow` / `sub` (titre, cible, départ chacun) ; les deux compteurs tournent toujours (bascule = barre tout de suite juste) ; `?objectif=sub|follow` sur une source. Les vrais totaux viennent de **Streamer.bot** : action « Overlay – Compteurs » (`outils/streamerbot-compteurs.cs`, à coller une fois, TUTO 6.6), lancée par l'overlay à la connexion et toutes les 5 minutes. Anciens réglages (`objectif.type/cible/depart`) repris tout seuls. Vérifié dans le navigateur (follow/sub, mode test, ancien mes-reglages.js) ; **pas testé** : le code C# dans un vrai Streamer.bot (pas de Streamer.bot ici).
- **2026-10-02** — Demande : « au lieu de Streamer.bot, StreamElements ? » puis « oui, on remplace Streamer.bot ! » → **Streamer.bot retiré**, tout passe par **StreamElements**, par internet (rien à installer) : `js/evenements.js` (commun) se branche à `wss://astro.streamelements.com` avec le jeton du compte (« JWT Token », `reglages.html` › StreamElements, gardé dans mes-reglages.js), écoute `channel.activities` (follow, sub, réabonnement, cadeau, pluie de cadeaux annoncée une seule fois, bits, raid, dons) et `channel.session.update` (vrais totaux followers / abonnés pour l'objectif, lus aussi

au branchement), se rebranche tout seul. Plus d'action C# ni de client Streamer.bot ; les anciens réglages streamerbot sont oubliés au prochain Enregistrer. TUTO §6 réécrit (6.2 brancher StreamElements, 6.3 journal, 6.5 dons par la page de dons StreamElements, 6.6 vrais nombres automatiques). Testé dans le navigateur avec un **faux serveur StreamElements** (chaque type d'événement, compteurs, objectif atteint, reconnexion) ; **pas testé** avec le vrai StreamElements ni dans OBS.

- **2026-10-02** — Demande : « un bouton pour regénérer les transitions et tout, depuis les réglages ? » → impossible depuis une page web (un navigateur ne lance pas de programme) : choix **depuis OBS**. Script OBS (`outils/actualiser-obs.lua`) : boutons **Refaire les vidéos de transition** et **Refaire les images du kit Twitch** (`outils/refaire.ps1` , en arrière-plan ; installe Node.js et ffmpeg au 2e clic s'ils manquent ; les transitions Stinger de l'overlay sont décrochées pendant la fabrication puis rechargées ; le dossier du kit s'ouvre à la fin). `reglages.html` le rappelle après un changement de couleurs, du nom ou des textes du kit. TUTO 2.8 (+ §5, §7, §9). Testé avec PowerShell sous Linux (faux cmd / winget / explorer : vraies vidéos et images refaites) et un faux OBS ; **pas testé sous Windows ni dans un vrai OBS**.
- **2026-10-06** — Demande : « un fond et une bannière pour la page de dons StreamElements (640×200 et 1920×1080), régénérables comme le kit, et un panneau de bio Commandes » → kit : nouveau groupe **Page de dons StreamElements** (`dons-banniere.png` , `dons-fond.png` : le bouffon qui rit, pièces d'or, « Offrande au royaume » ; le milieu du fond reste libre pour le formulaire), textes dans `config.js › chaine.dons` (titre, texte ; reglages.html › Kit), panneau **Commandes** ajouté à `chaine.panneaux` . TUTO §7 : pas à pas StreamElements (Revenue › Tipping settings) et description du panneau Commandes.
- **2026-10-06** — Nouvelle demande : animations déclenchées par les points StreamElements de la chaîne (les **Grelots**), en paliers : chute, duel contre le bouffon maléfique, dragon. Démos faites dans le moodboard (bouffon maléfique et dragon dessinés). Pause : chamallow au-dessus du feu, « Z » vers la gauche, cheveux de dos qui suivent la tête.
- **2026-10-06** — Grelots : 4 spectacles ajoutés entre la chute et le duel (tarte à la crème, sérénade, jet du destin, potion mystère), prix fixés : 10 / 25 / 50 / 75 / 100 / 150 / 300. La sérénade se joue au banjo (pas au luth). Démos faites dans le moodboard.
- **2026-10-06** — Grelots : fil rouge de **rivalité avec Darktagrain** (le bouffon maléfique). Banane posée par Darktagrain (10), tarte lancée par lui (25), nouveaux paliers pierre-feuille-ciseaux (125) et coffre au trésor (150, objet légendaire / commun / nul), duel déplacé à 200 (Darktagrain s'enfuit chercher son dragon), dragon à 300 avec Darktagrain sur son dos. 175 : idée à choisir.
- **2026-10-06** — Grelots 175 : **la catapulte** retenue, avec un pile ou face avant (qui envoie qui) et 7 atterrissages au hasard (pieds, paille, écran, mare, bouse, Dames du Lac, taverne). Ordre du chifoumi confirmé.
- **2026-10-06** — Catapulte : les Dames du Lac deviennent des **sirènes** (jolies, coquillages, tout public) ; taverne recadrée (bulles devant le décor, « CRAC ! » sur le toit, sortie par la porte visible).
- **2026-10-06** — Grelots branchés : commandes mystère `!spectacle10` … `!spectacle300` (StreamElements ne sait pas faire « !spectacle MONTANT » : coût fixe par commande ; `!duel` déjà pris), repérage de la réponse du bot dans le chat (`Chat.ecouter`), spectacles joués par la source Alertes, `!essai <spectacle>` pour le streamer, réglages et TUTO § 6.9.

- **2026-10-06** — Pendant un spectacle des Grelots, le bouffon des écrans Starting soon, Pause et Fin s'éclipse en fondu, puis revient à la fin (note « spectacle en cours » dans la mémoire commune des pages, renouvelée chaque seconde par la source Alertes).
- **2026-10-06** — Réglages des Grelots rangés en **tableau** (une ligne par spectacle : actif, commande, prix, réponse du bot) au lieu de 40 champs à la suite ; nouveau type de section « tableau » dans le moteur commun des réglages.
- **2026-10-06** — Demande : « un panneau Discord pour la bio Twitch, et les réseaux classiques » → panneaux **Discord, YouTube, TikTok, Instagram, X (Twitter)** ajoutés à `chaine.panneaux` ; chaque nom de réseau prend son logo (fichier commun `chaine/reseaux.js`, logos simplifiés d'une couleur, aux couleurs du thème ; aussi Twitch, Kick, Bluesky). TUTO §7 : lien de l'image vers le réseau.
- **2026-10-06** — Le bouffon des écrans ne s'éclipsait pas dans OBS pendant les spectacles (les sources d'OBS ne semblent pas partager leur mémoire en direct) : plan B, chaque écran (Starting soon, Pause, Fin) lit lui-même le chat et cache son bouffon pendant la durée maximale du spectacle ; la note de la source Alertes reste prioritaire quand elle passe (fin exacte).
- **2026-10-07** — Nouvelle idée : overlay pour streamer les parties de JDR (portraits illustrés animés quand un joueur parle sur Discord, interaction du chat par sondages et commandes). Discussion de faisabilité, rien de construit.
- **2026-10-07** — JDR : le streamer fournit les illustrations de ses personnages et leurs fiches (campagne « La Malédiction de Pangée ») ; 1 MJ + 4-5 joueurs variables ; D&D / Daggerheart / COF2 ; jeu sur Discord + CodexVTT. Moodboard JDR fait (portraits, cadres, deux styles, scènes, sondage, commandes, jets de dés, écran d'attente). Question d'un accès « overlay » en lecture seule envoyée à la session CodexVTT.
- **2026-10-07** — Réponse de la session CodexVTT : l'overlay peut écouter une partie en lecture seule si CodexVTT ajoute un mode « spectateur » (clé par partie, révocable par le MJ ; namespace Socket.io /spectator isolé avec son propre CORS ; jets publics seulement, résumé des fiches : nom, portrait, PV, stress). Rien n'est fait : à valider par le streamer, prévu après la version 3.24.0 de CodexVTT.
- **2026-10-07** — JDR : « tes dessins ne sont pas terribles » → abandon du portrait dessiné ; les illustrations du streamer parlent façon marionnette de papier (mâchoire découpée dans l'image qui descend, bouche sombre derrière), réglé pour les 5 personnages. Moodboard JDR : la page défile de nouveau (theme.css bloquait le défilement).
- **2026-10-07** — La marionnette « fait flipper » : retirée du moodboard (le code reste dans js/portraits.js, désactivé). Je ne peux pas générer d'images moi-même : portraits = illustrations du streamer (rebond + lueur quand on parle) ; pour une bouche qui bouge, 2e image « bouche ouverte » à générer par son IA (consigne fournie).
- **2026-10-07** — Le streamer fournit des bustes « bouche fermée / bouche ouverte » (Namerya, Kenai, Trayss, Zokhran) faits par son IA : le damier dessiné est retiré par un nouvel outil `outils/detourer.mjs`, les deux images alternent quand le personnage parle (moodboard JDR, bouton « Portraits »).
- **2026-10-07** — Son IA n'arrive pas à faire les autres bustes : retour aux portraits dessinés (vectoriels), mais en styles assumés. Zokhran redessiné d'après son buste dans trois styles à choisir (moodboard JDR § 02, `js/dessins-jdr.js`) : A encre et aplats, B sceau d'or, C vitrail ; bouche qui parle et clignement. Une fois le style choisi : dessiner le MJ, Namerya, Kenai, Trayss, Ba Heng Ze, Dvergrost.

- **2026-10-07** — Styles retenus : **A encre et aplats** et **C vitrail** (plus de détails demandés : rides, broderies, rivets, plombs, lumière dans le verre). Toute la table dessinée dans les deux styles (`js/dessins-jdr.js`) : Zokhran, Namerya, Kenai, Trayss, Ba Heng Ze, Dvergrost ; bouche qui parle, clignement. Nouvel état **« à terre » à 0 PV** (yeux fermés, bandage, couleurs ternies, buste qui penche, bandeau « À terre ») et barre de PV sous les portraits (moodboard JDR § 02 bis). Le portrait du MJ reste à faire (description ou photo à fournir).
- **2026-10-07** — JDR : mise en page **A (bandeau du bas)** retenue. Les héros sont refaits **dans le style de Patagrain, sur le squelette du bouffon** (nouveau générateur `outils/generer-heros.mjs` → `js/heros.js`) : Patagrain en **maître du jeu** (sans chapeau ni collerette, robe bleu nuit à galons d'or, pendentif d20, livre), Zokhran (hachoir), Namerya (bâton), Kenai (épée et bouclier), Trayss (arbalète), Ba Heng Ze (masse), Dvergrost (hache) ; expressions « parle » et « à terre » (bandage). Idée validée à étudier : **les jets de dés en action** — branché à CodexVTT, chaque jet est joué par le héros en bas à droite (athlétisme réussi : il court et saute ; raté : il s'étale ; attaque qui touche, 20 naturel, 1 naturel…) avec une carte du jet. Démo dans le moodboard JDR § 06 (5 jets). Les gestes du bouffon (marcher, sauter, tomber, frapper, danser) servent tels quels.
- **2026-10-07** — JDR : cadre **carte de héros** retenu. **Trayss a trois formes** (dragon, entre les deux, humaine) : dessinées dans le style Patagrain (`trayss`, `trayss-mi`, `trayss-humaine`), choix de la forme affichée (boutons dans le moodboard, plus tard un réglage, voire automatique depuis CodexVTT). Bibliothèque d'une quinzaine de gestes validée (à faire selon ce que CodexVTT pourra envoyer). Besoins détaillés envoyés à la session CodexVTT (jets publics avec compétence, réussite/échec, critiques, Espoir/Peur ; PV/stress des PJ ; tour d'initiative ; forme de Trayss), en attente de sa réponse. Commandes du chat : une liste complète à proposer plus tard.
- **2026-10-07** — Réponse de CodexVTT (faisabilité, rien de lancé) : mode spectateur ≈ 2 à 3 jours, prévu en 3.25.0 (après la 3.24.0 villes/donjons), si le streamer valide. Jets publics déjà presque complets (personnage, formule, résultat, critiques, avantage, Espoir/Peur ; à ajouter : id de fiche, type de jet, libellé en français) — **réussite/échec connus seulement si un DD est saisi** au moment du jet ; PV et stress des PJ en temps réel (petit ajout) ; tour d'initiative et dégâts/soins reçus (petits ajouts) ; forme de Trayss : pas dans CodexVTT → réglage côté overlay (ou petit champ « forme » à ajouter si voulu).
- **2026-10-07** — Le streamer veut le petit champ « forme active » dans CodexVTT (Trayss change de dessin toute seule) et valide le geste « sans verdict » quand aucun DD n'est saisi. Transmis à la session CodexVTT (lancement du mode spectateur à confirmer par lui).
- **2026-10-07** — Retouches : Trayss dragon « ressemblait à une chèvre » → tête de draconide (crâne large, long museau plat, crocs, collerettes, cornes rabattues vers l'arrière) ; toque de Zokhran « serrée » → plus large et bouffante. **Bibliothèque de gestes des jets** faite (`css/bouffon.css` § 8 + moteur `js/jets.js`) : courir, grimper, se cacher, crocheter/fouiller, lire, observer, parler, menacer, danser, lancer un sort, tirer, frapper, parer (sauvegarde), se soigner, garde (initiative), encaisser ; verdicts réussi / raté / 20 naturel / 1 naturel / sans DD / Espoir-Peur ; le geste se choisit d'après le libellé du jet. Banc d'essai dans le moodboard JDR § 06 (faux jets au format CodexVTT).
- **2026-10-07** — **Scène JDR construite** (`scenes/jdr.html`, mise en page A) : table CodexVTT en grand (trou transparent x 40 · y 36 · 1360 × 765), cartes de héros en bas, chat, arène des jets en bas à droite (invisible tant qu'aucun jet ne se joue) ; `css/jdr.css`, `js/jdr.js`. Réglages : « Scène JDR » (nom de la campagne, éteindre ceux qui se taisent) et le tableau **« la table de jeu »** (Présent · Joueur · Personnage · Identifiant Discord) → `config.js › jdr.places` ; le tableau des réglages sait maintenant afficher une

liste à choisir (moteur commun). Compétences de CodexVTT lues dans son code (D&D 5e, traits Daggerheart, caractéristiques COF2) : toutes rattachées à un geste. TUTO § 3.7. Pas encore branché : Discord (qui parle), CodexVTT (jets, PV).

- **2026-10-12** — Demandes : « générer le panneau Planning depuis les réglages » et « un panneau Mes créations / Mes sites » → le panneau **Planning** devient le planning de la semaine (jours, heures, jeu en option : « Mercredi | 20h30 | Minecraft ») dès que `reglages.html › Kit › Planning` est rempli (vide = panneau titre comme avant) ; panneau **Mes créations** ajouté (pinceau ; « Mes sites » = globe, via `chaine/reseaux.js`). Titres longs des panneaux ajustés.
- **2026-10-12** — **CodexVTT branché** (mode spectateur livré par CodexVTT, protocole `codex-web/docs/SPECTATEUR.md`) : client Socket.io copié en local (`js/socket.io.min.js` 4.8.3), nouveau `js/codex.js` (objet `Codex`, WebSocket seulement, reconnexion 5 s → 1 min, panneau `?journal=1` en haut à droite). La scène JDR joue les jets dans l'arène, montre les PV sous les cartes (« à terre » à 0), les coups (–9 rouge) et soins (+5 vert), « ⚔ Son tour » en combat, la forme de Trayss (nom du portrait actif : Dragon / Entre deux / Humaine) ; personnages retrouvés par leur nom ; jets d'inconnus ignorés ; clé refusée ou révoquée signalée. Réglages : `jdr.codex` { actif, adresse, cle } — la clé est un secret (mes-reglages.js seulement). Testé avec un faux serveur CodexVTT (bonne clé, mauvaise clé, révocation) ; pas avec le vrai, ni dans OBS. TUTO §3.7 complété.
- **2026-10-12** — Question : « dans les réglages, tout est propre et organisé ? » → section **Kit de chaîne Twitch** rangée (une aide en tête qui renvoie au bouton OBS, textes, page de dons, panneaux, planning) ; ancien champ **Réseaux** retiré (plus utilisé : les logos viennent de la liste des panneaux ; `reglages.js` l'oublie dans les anciens réglages).
- **2026-10-12** — Question : « les personnages sont préfaits ? et si je change de joueur ? » → 3 pistes proposées (créateur de personnage dans les réglages, portrait de la fiche CodexVTT, dessin sur demande) ; **choix : le portrait de la fiche CodexVTT**. Fait : personnage **« Autre »** dans le tableau des joueurs (+ colonnes « Nom du personnage » et « Sous-titre », vides = ceux du dessin) ; sa carte prend le portrait de sa fiche (`character.portrait`), sinon un **aventurier encapuchonné** (nouveau dessin `aventurier` dans `outils/generer-heros.mjs`, cape à la couleur de la carte, tirée de son nom) qui joue aussi ses jets dans l'arène (le portrait de la fiche en médaillon sur la carte du jet). Testé avec le faux CodexVTT.
- **2026-10-12** — « Trayss humaine ressemble beaucoup à Namerya » → Trayss (formes humaine et entre-deux) redessinée : **queue de cheval haute** avec **mèche violette** (au lieu des cheveux longs détachés), peau plus mate sans joues roses, visage plus anguleux, sourcils froncés, trait d'eye-liner, écailles violettes sur la pommette, anneaux d'argent aux oreilles. Namerya inchangée. À valider.
- **2026-10-12** — Qui parle sur Discord : StreamKit (simple mais ne transmet rien à l'overlay : seulement un halo posé par-dessus, sans bouche ni estompage, fragile) écarté → **un bot Discord sur le VPS** (choix du streamer). Fait : `serveur/voix/` (`voix.mjs` : discord.js + @discordjs/voice, suit le MJ (SUIVRE) dans les salons vocaux, micro coupé, repère qui parle à l'arrivée du son sans le déchiffrer ; WebSocket `wss://…/voix` avec une clé ; mode `--simulation`), `installer-voix.sh` (jeton demandé, clé générée, service `overlays-voix` + relance quand le dépôt change, `/voix` ajouté à Nginx/Caddy par `serveur/installer.sh`), LISEZMOI. Overlay : `js/voix.js` (objet `Voix`, adresse par défaut = celle des mises à jour + /voix, `?journal=1`), `jdr.discord` { actif, adresse, cle } dans les réglages, colonne « Discord (pseudo ou nom) ». Testé : chaîne bot en simulation → scène JDR (cartes animées, mauvaise clé refusée), jeton invalide refusé proprement. Pas testé : le vrai Discord, l'installation sur le

VPS, OBS. Seule partie du projet avec des bibliothèques npm (la voix Discord est chiffrée de bout en bout et change souvent).

- **2026-10-12** — CodexVTT précise le timing des dés 3D (le jet arrive au lancer avec son résultat, les dés se posent 1 à 2 s après) → l'arène joue l'**élan** tout de suite et le **dénouement** 1,5 s après (`Jets.monter(zone, { denouement: 1500 })`) : la carte montre un dé qui roule, puis le résultat et le verdict ; saut, chute, critique, parade, tir et PV soignés attendent eux aussi. Vérifié par captures (normal + ?test=1).
- **2026-10-12** — Demande : « un bouton dans les réglages pour régénérer le planning, choisir la semaine (« planning semaine 8 ») sans effacer celui qu'on a » → sous le champ Planning de `reglages.html` : calendrier de semaine + 🖼️ **Fabriquer l'image du planning** (la page enregistre, dépose `outils/demande.txt` ; le script OBS lance `refaire.ps1 -Quoi planning -Semaine …` et répond dans `outils/demande-resultat.txt` ; la page montre l'image). Une image par semaine dans `chaine/export/plannings/` (sous-titre « Semaine 8 · du … au … »), les anciennes restent.
- **2026-10-12** — Demandes : un **tuto complet** de la scène JDR et du bot (à faire, après le choix de la disposition) ; **retirer le chat** de la scène JDR et réorganiser → 2 dispositions proposées (A : table plus grande 1500 × 844, cartes plus petites ; B : table et cartes inchangées, colonne de droite = ordre du combat + arène plus grande — conseillée), **en attente du choix** ; le bot ne doit te suivre **qu'en JDR** → fait : `SALONS` (salons permis, par nom ou identifiant) et `AVEC_OVERLAY` (il n'entre que si la scène JDR est ouverte dans OBS, ressort 2 min après), testé avec un faux Discord ; **résultats affichés trop peu de temps** → fait : le résultat reste `jdr.dureeResultat` secondes (4 par défaut, réglable) une fois les dés posés, 2 s au plus si d'autres jets attendent ; **les PNJ du MJ** sans animation → question posée à CodexVTT (les jets publics des PNJ arrivent-ils, avec un drapeau « PNJ » ?), en attente.
- **2026-10-12** — PNJ : CodexVTT confirme que les jets PUBLICS du MJ depuis une fiche de PNJ ou de créature arrivent déjà (nom et id de la fiche ; jamais d'événement 'character' pour elles) ; les jets privés n'arrivent jamais. Fait : un jet dont la fiche n'est pas celle d'un joueur est joué dans l'arène par l'**aventurier encapuchonné aux couleurs des PNJ** (`jdr.pnj` { jouer, couleur : bordeaux par défaut }, dans les réglages), avec le nom du PNJ. Testé avec le faux CodexVTT (un Gobelin). CodexVTT propose un champ `isNpc` dans 'roll' (plus sûr, et utile pour les jets sans fiche du MJ) : **feu vert à donner par l'utilisateur dans la session CodexVTT** ; l'overlay le lit déjà s'il arrive.
- **2026-10-12** — Disposition **A** retenue (table 1500 × 844, cartes dessous ; à droite le titre de la campagne avec l'emblème, l'ordre du combat écarté : « on ne l'a pas forcément »). Puis le **chat revient** sous le titre (demande du streamer) : pendant un jet de dés, l'arène passe par-dessus et le chat s'efface, puis revient. Textes et effets de l'arène à l'échelle du plus petit côté (arène haute et étroite) ; le verdict sur sa propre ligne. **Tuto complet** de la scène et du bot : `TUTO-JDR.md` / `.pdf` (scène OBS de A à Z, table de jeu, CodexVTT, gestes, PNJ, bot Discord de la création à la vérification, avant chaque partie, test, dépannage) ; TUTO §3.7 raccourci et renvoie à lui ; lien dans index.html. **PV pas synchronisés** : erreur 400 de CodexVTT à l'enregistrement de la fiche (champ save des sorts) → transmis à la session CodexVTT. Idée d'une **vue joueur de CodexVTT** à mettre dans OBS (carte avec brouillard, images montrées par le MJ) → faisabilité demandée à CodexVTT.
- **2026-10-12** — CodexVTT : bug des PV corrigé (sauvegarde des sorts importés en objet → refus de tout le PATCH) ; nouveau dans 'roll' : isNpc et player (les jets du MJ sans fiche sont joués comme des PNJ). **Vue des joueurs pour OBS : feu vert du streamer, cadrage (a) automatique sur les jetons des**

joueurs + (b) « cadrage stream » posé par le MJ → transmis à CodexVTT (≈ 3 à 5 jours) ; côté overlay, TUTO-JDR à adapter à la livraison (source Navigateur à la place de la capture de fenêtre).

- **2026-10-12** — Retour : « le bouton du planning ne fonctionne pas » + « le fabriquer SANS OBS ouvert » → correctif (la page ne redemandait la permission du dossier qu'après l'enregistrement, ce que le navigateur peut refuser : elle vérifie d'abord si elle l'a déjà) et message clair quand personne ne prend la demande ; nouveau `fabriquer-images.cmd` (double-clic, `outils/fabriquer.ps1`) : traite la demande de la page sans OBS, ou menu planning d'une semaine / images du kit / vidéos de transition.
- **2026-10-14** — Scène JDR **migrée vers le moteur commun** `js/partie.js` (fait par la session d'harmonisation pour tous les overlays ; copie de `_modele`) : CodexVTT (jets, PV, coups, tour de combat, PNJ, rattachement des jets aux persos) et Discord (qui parle) passent par `Partie` ; Patagrain ne fait plus que dessiner. Rien ne change à l'écran : la table reste celle des réglages (`jdr.places`, même format : rien de perdu), la carte du MJ reste, les cartes s'affichent même sans CodexVTT. Nouveau : un personnage de CodexVTT sans place s'ajoute en bout de table (portrait de sa fiche) ; réglage « Mon personnage » (`jdr.monPerso`, vide : tu es le MJ) ; `?test=1` = la démo commune. Testé : démo, faux CodexVTT (PV, coup, soin, à terre, forme de Trayss, tour, PNJ, perso ajouté, portrait), bot voix en simulation ; pas dans OBS ni avec les vrais services.
- **2026-10-14** — Retours du streamer : **testé en vrai, ça marche** (scène JDR dans OBS, CodexVTT, bot Discord sur le VPS). **Forme de Trayss automatique retirée** (propre à un personnage : choisie dans le tableau des réglages, colonne Personnage). **Vue joueurs de CodexVTT annulée.** Ordre de la suite : finir l'harmonisation (scènes JDR de John Von Gurt et d'Ambries), puis la **fiche d'apparence** commune des personnages, puis la **liste des commandes du chat** JDR ; l'**overlay musique** = la prochaine version. L'envoi sur GitLab attend la fin de l'harmonisation.
- **2026-10-14** — Demande (via la session d'harmonisation) : aligner le design de la scène JDR sur les autres overlays → **cartes horizontales sur toute la largeur** sous la table (le héros à gauche dans son médaillon ; nom, sous-titre · joueur, barre de PV et « 20 / 20 PV » à droite), largeur min(420, partagée), étiquettes « ⚔ Son tour » et « Toi » (jdr.monPerso, carte un peu plus haute), le nom rétrécit au lieu d'être coupé (le sous-titre jusqu'à 70 % puis « … »). La table garde sa taille (1500 × 844) : rien à refaire dans OBS. L'arène (le héros joue son jet au-dessus du chat) reste le modèle des scénettes. Vérifié de 6 à 8 cartes (démo + faux CodexVTT).
- **2026-10-15** — « Pourquoi j'ai autant de cartes que de persos ? » : depuis la migration, chaque personnage de joueur de la table CodexVTT sans place s'ajoutait en bout de table. Corrigé : **seules les places des réglages ont une carte** ; l'ajout devient un réglage, décoché par défaut (`jdr.ajouterAutres`, reglages.html › Scène JDR). TUTO-JDR 3.1 et 3.4.
- **2026-10-15** — Règle précisée par le streamer : **les cartes = seulement les personnages et joueurs Discord choisis dans le tableau ; les scènes (jets joués dans l'arène) = TOUT personnage qui lance un dé**, avec ou sans portrait. Fait : le jet d'un personnage de joueur sans carte est joué par son héros dessiné s'il en a un (même nom), sinon par l'aventurier encapuchonné à sa couleur ; les PNJ restent en cape bordeaux. Testé avec le faux CodexVTT (Elwen sans carte : son jet est joué, 6 cartes).
- **2026-10-15** — La « rustine » (jet sans carte retrouvé par mon propre code) est remplacée par la version propre du moteur commun : `Partie.lire(jet).inconnu` (le personnage de joueur sans carte, avec son nom, son portrait et son apparence, gardé à part par Partie). Patagrain le joue par son héros dessiné du même nom (`JDR.dessinDuNom`), sinon par l'aventurier encapuchonné, avec le portrait de sa fiche sur la carte du jet. Les PNJ : cape bordeaux. Tout autre jet est joué aussi.

- **2026-10-15** — Harmonisation des 4 overlays : **page de réglages** refaite pareil partout (un réglage par ligne, sommaire des sections en haut, réglages avancés repliés, page plus large ; un tableau trop large devient des fiches : plus de défilement de côté). Scène JDR : **seuls les personnages et joueurs choisis ont une carte**, mais **tous les jets sont joués** (un personnage sans carte apparaît avec son apparence ou son portrait). Oreille droite des personnages réparée. Lien « Créateur de personnage » dans la vitrine. Ajout des fichiers communs des personnages (créateur de personnage au style « Conte », provisoire ; la scène JDR garde ses héros dessinés).
- **2026-10-15** — Scène JDR branchée sur la **boîte à outils de CodexVTT** (`js/codex-stream.js`) : l'overlay ne dessine plus lui-même les personnages, il demande le dessin à la boîte à outils (apparence enregistrée dans CodexVTT, sinon le catalogue). Nouveau réglage **Style des personnages** (reglages.html › Scène JDR) : Néon, Sticker, Écusson, Vitrail, Conte ou Aplat. Les héros dessinés restent. Les personnages sans héros sont dessinés au style **Conte** (nouveau : aplats, fin contour, lumière douce, liseré doré). Demande : le dessin **descend en bas de la carte**, fait : médaillon posé sur le bord du bas, héros en buste, plus grands.
- **2026-10-15** — Demande : « plus de dessins à la main dans l'overlay, refaire les héros depuis le créateur, et même les scénettes depuis CodexVTT ». Fait : le **squelette du bouffon et tous ses gestes** sont devenus le moteur commun de la boîte à outils de CodexVTT (le bouffon s'en sert aussi : `css/squelette.css` chargé avant `css/bouffon.css` , qui ne garde que chapeau, grelots, décors ; vérifié : il bouge exactement comme avant). Les **héros sont refaits avec les traits** (fiches d'apparence : nouveaux éléments toque rouge, tablier, kazoo, épée magique, tenue de cuir, robe de mage, taille nain, cape en peau d'ours, collier de griffes, barbe en anneau ; hache corrigée), dessinés en buste sur les cartes et **en pied dans l'arène** (même mise en scène qu'avant, maintenant dans la boîte à outils). Retirés : `js/heros.js` , `js/dessins-jdr.js` , `js/jets.js` , `outils/generer-heros.mjs` , `design/moodboard-jdr.html` . Le bouffon (mascotte) reste dessiné ici.
- **2026-10-15** — « Le MJ ne ressemble plus au bouffon, et les bordures jaunes ne sont pas mon style » : le style **Conte** est refait comme le bouffon (aplats de couleur sans aucun contour, seuls les traits du visage au trait fin) ; le MJ a des **lunettes carrées** et son **grain de beauté**.
- **2026-10-15** — « Pourquoi mon ancien style était plus fin ? » : les anciens héros montraient la tête ET le torse (sur le squelette du bouffon), les nouveaux un gros buste. Fait : les cartes montrent le personnage **à mi-corps** (squelette, de la tête à la taille), et les traits du visage sont plus fins (yeux, sourcils, nez, bouche, cou, épaules). Le MJ a une seule boucle d'oreille, à droite. Le créateur de personnage est retiré de l'overlay : il est dans CodexVTT.
- **2026-10-19** — Retours : « Aplat et Conte se ressemblent » → **Conte supprimé**, Aplat devient le style du bouffon (« conte » reste compris) ; cheveux du MJ (raie au milieu, longues mèches, sa vraie barbe, d'après l'ancien dessin) ; un **trou** dans ses cheveux et des **mèches détachées** (Namerya, Zokhran) corrigés pour toutes les coupes ; les formes de Trayss visibles dans la page des codes. Connexion à la pré-prod : le journal affiche maintenant « clé refusée » quand la clé vient d'un autre CodexVTT (chaque CodexVTT a sa propre base).
- **2026-10-20** — Les personnages des cartes et de l'arène sont maintenant dessinés et animés par **CodexVTT lui-même** (`js/codex-stream.js` = le lecteur de CodexVTT) : les retouches de dessin se font dans CodexVTT. Le bouffon garde son propre moteur de gestes (`css/squelette.css`), inchangé.
- **2026-10-21** — Scène JDR : l'**arène des jets est retirée** de l'overlay : les jets sont animés en grand sur la table par une **source OBS de CodexVTT** (« Animations des jets », TUTO-JDR chapitre 5), vue des

spectateurs, pas des joueurs. Plus de **liste de héros préfaits** dans les réglages : on écrit le nom du personnage comme dans CodexVTT (les anciens réglages restent compris ; la forme de Trayss vient de CodexVTT). L'ancien **bot Discord** est retiré (réglages, tuto, VPS) ; « qui parle » reviendra par Discord StreamKit. TUTO-JDR réécrit, plus court.

- **2026-10-21** — Scène JDR **harmonisée** dans les 4 overlays (Patagrain comme les autres) : la scène n'a plus que le décor, le titre de la campagne et le chat. La table (capture), les **cartes des personnages** et les **animations des jets** viennent de **CodexVTT**, en sources OBS à part, au thème de la chaîne ; les personnages, le MJ et les sous-titres se choisissent dans CodexVTT (Sources pour OBS). Réglages JDR : seulement le nom de la campagne. Tuto (fiche JDR) réécrit pas à pas. Le tableau des places des réglages est retiré (la liste se fait dans CodexVTT). TUTO-JDR réécrit.
- **2026-10-21** — **Qui parle** (Discord StreamKit, sans bot, facultatif) : quand un joueur parle sur Discord, le personnage de sa carte parle aussi. Une source Discord posée sur les cartes, avec un CSS fabriqué par CodexVTT d'après les identifiants Discord des joueurs. Pas à pas dans le tuto (fiche JDR). Pas encore essayé sur le vrai Discord.
- **2026-10-21** — Scène JDR : la table et les cartes ont maintenant **la même taille dans les 4 overlays** (table 1440 × 810, cartes 1440 × 150 en 1080p), pour que le CSS « Qui parle » de CodexVTT tombe juste partout. Le titre et le chat se sont décalés un peu à droite. Tutos mis à jour.
- **2026-10-21** — Scène JDR : **la même disposition dans les 4 overlays** (table en 60, 60 ; cartes en 60, 896 ; titre et chat à droite en x 1540) : les mêmes chiffres dans tous les tutos, et une scène OBS faite pour un overlay marche pour un autre.
- **2026-11-02** — CodexVTT ne sert plus les overlays (décision de l'utilisateur). Scène JDR refaite sans lui : la table de jeu capturée (CodexVTT ou un autre site), et **les cadres des personnages dessinés par l'overlay** (image, nom, sous-titre, PV, une ressource au nom réglable, « à terre », « son tour »), un cadre par place cochée dans `reglages.html`. Les PV se changent pendant la partie dans le **pupitre** (`pupitre-jdr.html`, un dock d'OBS). Tutos JDR réécrits. Pas encore essayé dans OBS.
- **2026-11-03** — Les images des personnages se posent **dans OBS** : la case du portrait de chaque cadre est vide et transparente, on y met une source Image (position et taille dans le tuto JDR, section 3.4, et sur `scenes/jdr.html?apercu=1`). La colonne « Image » des réglages est retirée.