**Cahier des charges - Pauvrathon**

**🎯 Résumé exécutif**

**Pauvrathon** est une application web permettant aux viewers non-abonnés de participer aux subathons Twitch en ajoutant du temps au timer via un système de clics et de mini-jeux, offrant ainsi une alternative gratuite à la participation payante.

**📋 Objectifs du projet**

**Objectifs principaux**

* Permettre aux viewers sans moyens financiers de participer aux subathons
* Créer de l'engagement communautaire autour des streams
* Générer de la visibilité pour les streamers via les mini-jeux

**Bénéfices attendus**

* **Pour les viewers** : Participation gratuite aux subathons
* **Pour les streamers** : Plus d'engagement, visibilité, moments de divertissement
* **Pour la plateforme** : Découverte de nouveaux streamers en subathon

**🏗️ Architecture technique**

**Stack technologique**

* **Frontend** : React (responsive, tous appareils)
* **Backend** : Supabase (PostgreSQL + temps réel + authentification)
* **APIs externes** : API Twitch
* **Hébergement** : À déterminer (Vercel/Netlify + Supabase)

**Authentification**

* Connexion via compte Twitch (viewers et streamers)
* Pas de système de comptes séparés

**🔍 Fonctionnalités détaillées**

**1. Détection automatique des subathons**

**Mécanisme**

* Recherche automatique toutes les 15 minutes via API Twitch
* Mots-clés : "subathon" dans le titre du stream
* Cache des résultats pendant 10-15 minutes pour optimiser les appels API
* Suppression automatique des streamers offline après quelques heures

**Gestion des rate limits**

* Système de cache Redis/mémoire
* Requêtes par batch (100 streamers par appel)
* Monitoring des appels API
* Fallback sur données en cache si limite atteinte

**2. Interface Viewer**

**2.1 Navigation principale**

Organisation en 3 sections :

* **Comptes suivis en pauvrathon** : Streamers suivis actuellement en subathon
* **Découvrir** : Liste de tous les streamers en subathon avec filtres (langue, etc.)
* **Historique** : Statistiques personnelles de participation

**2.2 Page de participation**

**Éléments visuels :**

* Photo de profil du streamer (cliquable)
* Compteur de clics actuel
* Informations sur le subathon

**Mécanisme de clics :**

* Clic sur la photo de profil
* Compteur affiché en temps réel
* Tous les 50 clics : déclenchement d'un mini-jeu aléatoire

**2.3 Système de mini-jeux**

**Mini-jeux disponibles :**

* Devine le chiffre (1-100) avec indications
* Pendu
* Autres jeux existants en HTML/CSS/JS

**Intégration :**

* Conversion des jeux existants en composants React
* Durée maximale : 2 minutes
* Timeout d'inactivité : 2 minutes → fermeture automatique
* Échec : compteur de clics remis à 0
* 3 échecs consécutifs → compteur remis à 0

**Validation :**

* Réussite du mini-jeu obligatoire pour valider le temps
* Temps ajouté : 0-60 secondes (configurable par streamer)
* Ajout automatique et immédiat au timer

**2.4 Historique personnel**

**Vue par streamer :**

* Liste des streamers auxquels le viewer a participé
* Statistiques par streamer : clics totaux, jeux gagnés/perdus

**Vue chronologique :**

* Historique détaillé de toutes les sessions
* Temps total ajouté à tous les subathons

**3. Interface Streamer**

**3.1 Dashboard de configuration**

**Paramètres configurables en temps réel :**

* Plage de temps ajouté (0-60 secondes)
* Nombre de clics avant mini-jeu
* Activation/désactivation du pauvrathon
* Timer actuel (saisie manuelle pour MVP)

**Paramètres par défaut :**

* 30 secondes de temps ajouté
* 50 clics avant mini-jeu
* Sauvegarde de l'historique des modifications

**3.2 Overlay en temps réel**

**Éléments visibles sur stream :**

* Indicateur "Pauvrathon en cours"
* Nombre de clics actuels
* Derniers participants
* Temps récemment ajouté

**Spécifications techniques :**

* Intégration OBS via source navigateur
* Mise à jour instantanée via Supabase Realtime
* Interface responsive

**3.3 Statistiques**

* Participants en temps réel
* Temps total ajouté
* Historique des sessions

**4. Gestion des sessions**

**4.1 Persistance des données**

* **Pas de sauvegarde** des clics en cours (refresh = remise à 0)
* Sauvegarde uniquement des résultats validés
* Sessions temporaires perdues à la fermeture du navigateur

**4.2 Gestion des streamers offline**

* Disparition immédiate de la liste de découverte
* Maintien en liste pour les streamers suivis avec mention "Offline"
* Impossibilité de participer si streamer offline

**🛡️ Sécurité et anti-triche**

**Mesures de sécurité**

* **Détection de rythme suspect** : analyse des patterns de clics
* **Validation par mini-jeu** : impossible d'automatiser complètement
* **Timeout d'inactivité** : 2 minutes sur mini-jeu
* **Échecs multiples** : remise à 0 après 3 échecs

**Accès universel**

* **Tous les viewers peuvent participer** (subs et non-subs)
* Pas de restriction par statut d'abonnement
* Principe d'égalité d'accès

**📊 Spécifications techniques détaillées**

**Base de données (PostgreSQL via Supabase)**

**Tables principales**

-- Streamers

streamers (

id, twitch\_id, username, display\_name,

is\_live, subathon\_active, current\_timer,

time\_range\_min, time\_range\_max, clicks\_required,

created\_at, updated\_at

)

-- Viewers

viewers (

id, twitch\_id, username, display\_name,

created\_at, last\_seen

)

-- Sessions de participation

participation\_sessions (

id, viewer\_id, streamer\_id, clicks\_count,

games\_won, games\_lost, time\_added\_total,

session\_start, session\_end

)

-- Historique des temps ajoutés

time\_additions (

id, viewer\_id, streamer\_id, time\_added,

game\_type, created\_at

)

**API Endpoints**

**Pour les viewers**

* GET /api/streamers/following - Streamers suivis en subathon
* GET /api/streamers/discover - Découverte avec filtres
* POST /api/participate/click - Enregistrer un clic
* POST /api/participate/game-result - Résultat de mini-jeu
* GET /api/history/personal - Historique personnel

**Pour les streamers**

* GET /api/streamer/config - Configuration actuelle
* PUT /api/streamer/config - Modifier la configuration
* GET /api/streamer/stats - Statistiques temps réel
* POST /api/streamer/timer - Mettre à jour le timer

**Temps réel (Supabase Realtime)**

* Subscription aux changements de clics
* Mise à jour instantanée des overlays
* Notifications de nouveaux participants

**🎨 Design et UX**

**Principes de design**

* Interface simple et intuitive
* Responsive design (mobile/desktop/tablette)
* Thème cohérent avec l'univers Twitch
* Feedback visuel immédiat sur les actions

**Parcours utilisateur type**

1. **Viewer** : Connexion Twitch → Découverte → Clic sur streamer → Participation
2. **Streamer** : Connexion Twitch → Configuration → Activation → Monitoring

**📅 Planning de développement**

**Phase 1 - MVP (4-6 semaines)**

* [ ] Setup projet React + Supabase
* [ ] Authentification Twitch
* [ ] Détection automatique des subathons
* [ ] Interface de participation basique
* [ ] Conversion des mini-jeux existants
* [ ] Dashboard streamer simple
* [ ] Timer manuel (saisie)

**Phase 2 - Améliorations (2-3 semaines)**

* [ ] Overlay temps réel
* [ ] Statistiques détaillées
* [ ] Filtres de découverte
* [ ] Historique personnel
* [ ] Optimisations performance

**Phase 3 - Fonctionnalités avancées (2-4 semaines)**

* [ ] Timer automatique (parsing OBS)
* [ ] Système anti-triche avancé
* [ ] Analytics approfondies
* [ ] Notifications push
* [ ] Classements et récompenses

**🔧 Maintenance et évolution**

**Monitoring**

* Surveillance des appels API Twitch
* Métriques de performance
* Logs d'erreurs et de sécurité

**Évolutions prévues**

* Intégration d'autres plateformes de streaming
* Nouveaux types de mini-jeux
* Système de récompenses avancé
* API publique pour développeurs

**💰 Considérations budgétaires**

**Coûts d'hébergement estimés**

* **Supabase** : Gratuit jusqu'à 500MB, puis ~25$/mois
* **Frontend** : Gratuit (Vercel/Netlify)
* **API Twitch** : Gratuit dans les limites
* **Total estimé** : 0-50$/mois selon l'usage

**Monétisation future (optionnelle)**

* Système de donations pour le développeur
* Fonctionnalités premium pour streamers
* Partenariats avec des marques gaming

**🎯 Conclusion**

Pauvrathon représente une innovation dans l'écosystème Twitch en démocratisant la participation aux subathons. Le projet allie engagement communautaire, accessibilité et divertissement tout en restant techniquement réalisable avec les technologies modernes.

L'approche par phases permet un développement progressif avec validation utilisateur à chaque étape, minimisant les risques et maximisant les chances de succès.