Université Catholique de Lille
Faculté de Gestion, Économie & Sciences
École du Numérique

Projet de Conception de Logiciel

TwiTok Bot

Bot Twitch pour la création automatique de vidéos Tik Tok

Presenté par:

Akil Wael
Aliligali Amir
Hebbinckuys Hugo
Kilito Yazid

Presenté à:

Mousin Lucien

Septembre 2024

Table des Matières

1. Introduction

1.1. Contexte

Le projet *TwiTok Bot* consiste en la création d'un bot automatisé qui permet aux streamers de Twitch de générer des vidéos Tik Tok à partir de leurs clips Twitch. Ce bot sélectionnera les clips en fonction de critères prédéfinis tels que le titre, la catégorie du stream (gaming, chatting), la date, la durée et le nombre de vues. Ensuite, il les transformera en vidéos optimisées pour Tik Tok, facilitant ainsi la création de contenu court pour les réseaux sociaux.

1.2. Client

Le client principal du projet est le streamer Twitch qui souhaite accroître sa présence sur les réseaux sociaux en générant du contenu Tik Tok de manière simple et rapide. Ce bot pourrait également intéresser des agences de marketing et des créateurs de contenu cherchant à automatiser la production de vidéos courtes adaptées aux plateformes sociales.

1.3. Historique

Avec l'explosion des plateformes de streaming comme Twitch et des réseaux sociaux basés sur des vidéos courtes comme Tik Tok, les créateurs de contenu sont constamment à la recherche de moyens pour transformer leurs performances en direct en clips engageants. Actuellement, cette tâche nécessite des outils manuels ou des éditeurs vidéo externes, un processus souvent long et coûteux. Le projet *TwiTok Bot* vise à résoudre ce problème en offrant une solution automatisée et efficace.

2. Description du Projet

2.1. Objectifs

Le projet *TwiTok Bot* a pour objectif de fournir une solution automatisée pour:

- Créer des vidéos Tik Tok à partir des clips Twitch sans intervention manuelle.

- Optimiser la sélection de clips en fonction de critères comme la durée, le format, et le nombre de vues.
- Réduire le temps de production de contenu et permettre aux streamers de maximiser leur visibilité sur les réseaux sociaux.

2.2. Fonctionnalités

Le bot devra inclure les fonctionnalités suivantes:

- **Extraction automatique des clips**: Sélection des clips basés sur le nombre de vues, la catégorie du stream, la durée, et d'autres critères pertinents.
- **Paramétrage du bot**: Les utilisateurs pourront définir des critères spécifiques comme la plage de dates, la durée maximale des clips, et les catégories de streams pour les vidéos Tik Tok.
- **Génération et édition vidéo**: Le bot doit être capable de générer automatiquement des vidéos au format Tik Tok (9:16, durée maximale de 60 secondes), avec possibilité d'ajuster les transitions, découper les clips, et ajouter du texte ou des effets.
- **Interface utilisateur intuitive**: Une interface simple et intuitive permettra aux streamers de configurer le bot, de consulter les clips générés, et de télécharger les vidéos prêtes à être publiées.
- **Notification et accès**: Une fois les vidéos prêtes, les utilisateurs seront notifiés et pourront visionner ou télécharger les vidéos depuis l'interface.

2.3. Contraintes

- **Contraintes techniques**: Twitch ne disposant pas d'une API officielle pour la récupération des clips, la solution impliquera du web scraping, ce qui peut affecter la performance. Le traitement vidéo pour la conversion au format Tik Tok demandera également des ressources importantes, notamment si le montage inclut des effets.
    - **Solution proposée**: L'utilisation de threads permettra d'optimiser les processus les plus gourmands en temps, tels que le montage vidéo. Toutefois, l'utilisation de threads dépendra des capacités de la machine de l'utilisateur.
- **Contraintes de calendrier**: Le projet doit être livré en six semaines, selon le découpage des sprints présenté dans la section *Déroulement du Projet*.

3. Architecture Logicielle

3.1. Structure du Projet

Le bot se composera des modules suivants:

- **Module d'extraction de clips**: Ce module récupérera automatiquement les clips Twitch basés sur les critères configurés par l'utilisateur.
- **Module de traitement vidéo**: Ce module convertira les clips sélectionnés en vidéos adaptées pour Tik Tok.
- **Module d'interface utilisateur**: Ce module permettra aux utilisateurs de configurer le bot et d'accéder aux vidéos générées.

3.2. Technologies Utilisées

- **Langages de programmation**: Python pour le backend et le traitement vidéo.
- **Bibliothèques**: MoviePy pour l'édition vidéo, Flask pour l'interface utilisateur.
- **Système de contrôle de version**: Git pour le suivi du code source et des versions.

4. Déroulement du Projet

4.1. Phases du projet

Le projet se déroulera en plusieurs phases, chacune comprenant des tâches spécifiques et des livrables:

1. **Développement du module de montage vidéo** (23 Sept 24 - 08 Oct 24, 15 jours)
   - Objectif: Créer et implémenter le module de montage vidéo qui ajustera les clips Twitch dans un format compatible avec Tik Tok (9:16).
   - Livrable: Module de montage vidéo fonctionnel avec tests basiques.
2. **Intégration de l'API Tik Tok** (08 Oct 24 - 18 Oct 24, 10 jours)
   - Objectif: Intégrer l'API Tik Tok pour permettre la génération de vidéos prêtes à être publiées directement depuis le bot.
   - Livrable: API Tik Tok fonctionnelle avec documentation.

3. **Intégration de l'API Twitch** (18 Oct 24 - 26 Nov 24, 39 jours)
    - Objectif: Créer le module d'extraction de clips en utilisant l'API Twitch, ou via web scraping si nécessaire.
    - Livrable: API Twitch fonctionnelle avec tests unitaires pour la récupération des clips.

4. **Tests et débogage** (26 Nov 24 - 04 Dec 24, 8 jours)
    - Objectif: Tester et déboguer les modules précédents (montage vidéo, API Tik Tok, API Twitch).
    - Livrable: Rapport de tests et corrections des bugs identifiés.

5. **Optimisation** (04 Dec 24 - 17 Dec 24, 13 jours)
    - Objectif: Optimiser une étape critique du processus, comme le montage vidéo ou la récupération des clips.
    - Livrable: Version optimisée du bot avec des améliorations en performance.

6. **Conception de l'interface utilisateur** (17 Dec 24 - 07 Jan 25, 21 jours)
    - Objectif: Finaliser la conception de l'interface utilisateur, permettant aux utilisateurs de configurer le bot et d'accéder aux vidéos générées.
    - Livrable: Interface utilisateur fonctionnelle et ergonomique.

7. **Présentation finale** (07 Jan 25)
    - Objectif: Présenter le projet finalisé en 7 minutes avec une démonstration des principales fonctionnalités.
    - Livrable: Démo du bot et vidéo Tik Tok générée.

5. Conventions de Nommage

Les conventions de nommage suivantes seront appliquées afin de garantir la cohérence et la lisibilité du code:

1. **Classes**: Nommage en *CamelCase*, chaque mot commençant par une majuscule (ex.: VideoProcessor, ClipManager).

2. **Variables**: Utilisation de la notation *camelCase*, avec des noms descriptifs (ex.: clipDuration, streamCategory).

3. **Fonctions**: Les noms des fonctions utiliseront *camelCase* et commenceront par un verbe (ex.: extractClips(), generateVideo()).

4. **Constantes**: Utilisation de la notation *SNAKE_CASE* en majuscule pour les constantes (ex.: MAX_CLIP_DURATION, DEFAULT_VIDEO_RATIO).

5. **Listes/Collections**: Les noms des collections seront au pluriel (ex.: clipsList, selectedStreams).

6. **Fichiers**: Utilisation de la notation *snake_case* en minuscule pour les fichiers (ex.: video_processor, user_interface).

6. Diagrammes

6.1. Diagramme de Classe

Un diagramme de classe sera utilisé pour visualiser la structure du projet, y compris les interactions entre les différents modules (extraction de clips, montage vidéo, interface utilisateur).

6.2. Diagramme de Gantt

Un diagramme de Gantt détaillant les phases du projet, les ressources allouées, et les délais sera fourni pour suivre l'avancement des tâches.

7. Tests

7.1. Tests Unitaires

Des tests unitaires seront mis en place pour chaque fonctionnalité clé du projet. Chaque module (extraction des clips, génération vidéo, intégration API) sera testé individuellement pour s'assurer que toutes les fonctionnalités fonctionnent correctement. Les tests incluront des scénarios courants et des cas limites afin de garantir la robustesse du système.

7.2. Tests d'Intégration

Les tests d'intégration vérifieront que tous les modules du projet fonctionnent ensemble de manière harmonieuse. Par exemple, une fois les clips extraits via l'API Twitch, ils seront automatiquement transmis au module de génération vidéo pour création. Les tests d'intégration couvriront la chaîne complète, de l'extraction des clips jusqu'à la génération finale des vidéos prêtes à être téléchargées.

7.3. Tests de Performance

Des tests de performance seront menés sur les opérations les plus exigeantes, comme le montage vidéo et le scraping des clips via Twitch. L'objectif est de garantir que les processus s'exécutent dans un délai raisonnable, même sur des machines avec des capacités réduites. Les tests évalueront également la charge sur le système lors de l'utilisation des threads.

7.4. Tests de Régression

À chaque modification du code, des tests de régression seront effectués pour s'assurer que les nouvelles fonctionnalités n'affectent pas les fonctionnalités existantes. Ces tests garantiront que le système reste stable et fonctionnel après chaque mise à jour.

8. Documentation

8.1. Documentation Technique

Une documentation technique complète sera fournie, comprenant des descriptions détaillées des modules, des API utilisées, des algorithmes d'extraction et de traitement des clips, ainsi que des informations sur l'architecture du bot. La documentation inclura également les conventions de codage et les procédures de test.

8.2. Documentation Utilisateur

Une documentation utilisateur claire et concise sera incluse pour guider les streamers dans l'installation, la configuration et l'utilisation du *TwiTok Bot*. Cette documentation expliquera comment configurer les paramètres du bot, consulter les clips générés, et télécharger les vidéos Tik Tok.

9. Livrables

À la fin du projet, les livrables suivants seront fournis:

- **Code source complet** du bot, incluant les modules d'extraction de clips, de génération vidéo, et d'intégration des API, disponible sur GitHub.

- **Vidéo Tik Tok générée automatiquement** à partir d'un stream Twitch, démontrant les capacités du bot.
- **Documentation complète**: Documentation technique (architecture, modules, tests) et documentation utilisateur.
- **Rapport final de tests**: Résultats des tests unitaires, d'intégration et de performance.
- **Présentation finale**: Démo en direct lors de la présentation finale du 7 Janvier 2025, avec explication du processus et résultats obtenus.