

# Watch Party

## UE4 - Technique de Programmation II

Lagarde Valentin, Kindbetter Arnaud, Sénéchal Maé

Master 1 DS2E-SE  
Université de Strasbourg

2024/2025

# Plan de la présentation

- 1 Présentation du projet
- 2 Fonctionnalités
- 3 Implémentation technique
- 4 Connexion et modes d'accès
- 5 Démonstration
- 6 Difficultés rencontrées et solutions
- 7 Évolutions et améliorations
- 8 Conclusion

- **Watch Party** : Application Python permettant d'organiser des soirées de visionnage de vidéos YouTube
- **Caractéristiques principales** :
  - Synchronisation en temps réel des vidéos pour tous les participants
  - Contrôles centralisés (lecture, pause, repositionnement)
  - Interface conviviale et intuitive
  - Chat intégré pour communiquer pendant le visionnage

# Fonctionnalités principales

- Mode hôte et mode client
- Contrôle de lecture (play, pause, repositionnement)
- Synchronisation automatique entre tous les participants
- Chat en temps réel
- Thème clair/sombre
- Connexion via réseau local ou Internet (avec ngrok)

# Interface principale

Watch Party

Configuration

Mode:

Adresse serveur:  Port:  Nom d'utilisateur:

Contrôle vidéo

URL YouTube:

Aller à (sec):

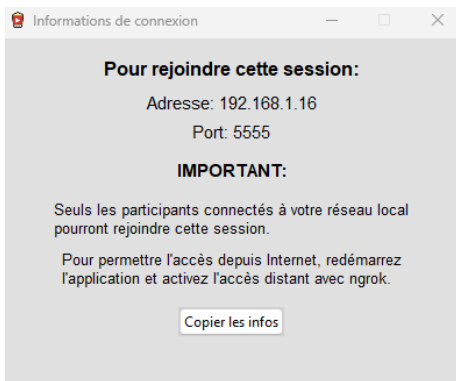
☒ Correction automatique de désynchronisation Seuil (sec):

Chat

[Système] Serveur démarré sur le port 5555  
[Système] Connecté en tant qu'hôte : User457  
[Système] === INFORMATIONS DE CONNEXION ===  
[Système] Tunnel ngrok créé avec succès!  
[Système] Les participants peuvent se connecter avec:  
[Système] Adresse: 4.tcp.ngrok.io  
[Système] Port: 12283

- **Python** comme langage principal
- **Tkinter** pour créer l'interface graphique (fenêtres, boutons...)
- **Exécution de plusieurs tâches en parallèle** pour gérer le réseau et l'interface en même temps
- **Connexions réseau** pour la communication entre les utilisateurs
- **Selenium** pour contrôler automatiquement le navigateur YouTube
- **Échange de données** en format simple (JSON)
- **ngrok** pour la connexion via Internet

# Mode réseau local



- Connexion en mode réseau local :
  - Adresse IP locale (192.168.x.x)
  - Port standard (5555)
  - Limité aux participants sur le

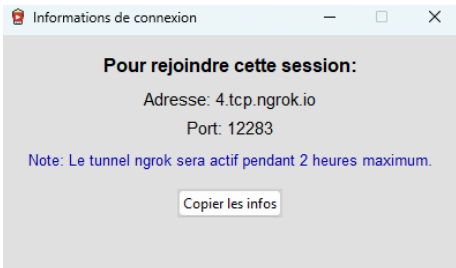
- **Avantages :**

- Configuration simple
- Performance optimale
- Pas de limitation de durée

- **Limitations :**

- Uniquement pour utilisateurs du même réseau
- Nécessite la configuration du pare-feu

# Mode connexion Internet (ngrok)



- Connexion via Internet avec ngrok :
  - Adresse ngrok publique (\*.tcp.ngrok.io)
  - Port attribué dynamiquement
  - Tunnel actif pendant 2 heures maximum

- **Avantages :**

- Accessible depuis n'importe où
- Aucune configuration de routeur requise
- Partage facile via copier-coller

- **Limitations :**

- Limité à 2 heures en version gratuite
- Nécessite une connexion Internet stable



- ❶ Démarrage en mode hôte
- ❷ Configuration de ngrok pour l'accès distant
- ❸ Connexion d'un client
- ❹ Chargement d'une vidéo YouTube
- ❺ Contrôles de lecture
- ❻ Chat entre utilisateurs
- ❼ Changement de thème
- ❽ Synchronisation automatique

[DÉMONSTRATION EN DIRECT]

- **Format de l'application :**

- *Initialement*, nous voulions développer un site web
- Mais HTML et CSS sortaient du programme du cours centré sur Python
- Nous avons donc opté pour une application de bureau en Python pur

- **Prérequis pour les utilisateurs :**

- Tous les utilisateurs doivent avoir Python installé
- Installation des bibliothèques nécessaires requise
- Configuration pas aussi simple qu'une application web classique

- **Synchronisation précise** entre différents clients
  - *Solution* : Système qui vérifie et corrige régulièrement les décalages
- **Gestion des connexions/déconnexions**
  - *Solution* : Système qui détecte quand un utilisateur se déconnecte
- **Contrôle du navigateur YouTube**
  - *Solution* : Utilisation de Selenium pour automatiser les actions sur YouTube

- **Création d'une interface utilisateur intuitive**
  - *Solution* : Organisation claire et logique des éléments d'interface
- **Configuration de ngrok**
  - *Solution* : Interface simplifiée pour la gestion du token et du tunnel

- **Conversion en application web** (idée initiale)
  - Plus accessible (pas besoin d'installer Python)
  - Interface utilisateur améliorée avec HTML/CSS
  - Meilleure compatibilité avec différents appareils
- **Support d'autres plateformes vidéo** (Netflix, Prime Video, etc.)
- **Système de salles multiples**
- **Partage d'écran intégré**
- **Authentification des utilisateurs**
- **Communication audio/vidéo entre participants**
- **Amélioration de l'interface utilisateur** (design adaptatif)
- **Mode sans ngrok** avec configuration automatique du routeur

- **Apprentissages clés**

- Approfondissement des connaissances en Python
- Découverte de nouvelles bibliothèques
- Gestion de projet de bout en bout

- **Aspects les plus intéressants**

- Contrôle automatique de YouTube
- Création du système de synchronisation
- Assemblage des différentes parties du projet

- **Difficultés surmontées**

- Gestion des connexions réseau
- Contraintes liées à l'utilisation exclusive de Python
- Amélioration des performances

Merci de votre attention !

Des questions ?