



STAGE 2A  
PROJET EIRBALLOON

---

## Mise en place de la régie de diffusion

---

*Elèves :*  
IGNATOWICZ NIGEL

*Enseignants :*  
Guillaume FERRE  
Anthony GHIOTTO

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Description de la chaîne d'information</b>	<b>2</b>
<b>3</b>	<b>Setup du PC</b>	<b>2</b>
3.1	Caractéristiques du PC . . . . .	3
3.2	OBS Studio . . . . .	3
3.2.1	Fonctionnement en bref . . . . .	3
3.2.2	Fonctions utilisées . . . . .	4
<b>4</b>	<b>Récupération et affichage des données de vol</b>	<b>4</b>
4.1	Données capteurs . . . . .	4
4.1.1	Script Python . . . . .	5
4.1.2	Affichage des données . . . . .	5
4.2	Données GPS . . . . .	5
4.2.1	Script Python . . . . .	5
4.2.2	Affichage des données . . . . .	5
4.3	Timeline . . . . .	6
4.3.1	Script Python . . . . .	6
4.3.2	Affichage . . . . .	6
4.4	Compte à rebours . . . . .	7
4.4.1	Script Python . . . . .	7
4.4.2	Affichage . . . . .	7
<b>5</b>	<b>Ajout des webcams USB</b>	<b>8</b>
5.1	Description . . . . .	8
5.2	Affichage . . . . .	8
<b>6</b>	<b>Ajout de la caméra IP</b>	<b>8</b>
6.1	Installation . . . . .	8
6.2	Affichage . . . . .	9
<b>7</b>	<b>Acquisition de l'analyseur de spectre</b>	<b>9</b>
7.1	Installation . . . . .	9
7.2	Affichage . . . . .	10
<b>8</b>	<b>Talkie Walkie</b>	<b>10</b>
8.1	Installation . . . . .	10
8.2	Diffusion . . . . .	10
<b>9</b>	<b>Récapitulatif et diffusion</b>	<b>11</b>
9.1	Récapitulatif . . . . .	11
9.2	Diffusion . . . . .	11

## 1 Introduction

Dans le cadre du projet Eirballoon, il a été choisi de réaliser une diffusion en direct de l'évènement sur la chaîne YouTube de l'ENSEIRB-MATMECA afin d'en faire profiter le plus grand nombre malgré la situation sanitaire actuelle. De plus, cela permet de garder une trace de l'événement destinée notamment aux futurs projets ballons.

Pour réaliser cela, une certaine quantité de matériels sont nécessaire. L'ensemble des moyens mis en oeuvre seront détaillés dans ce document. La diffusion sera réalisée depuis le PC du club radio de l'ENSEIRB-MATMECA, équipé de 4 écrans Full HD.

## 2 Description de la chaîne d'information

Le schéma suivant présente une vue d'ensemble simplifiée de l'installation finale (matérielle et logicielle) :

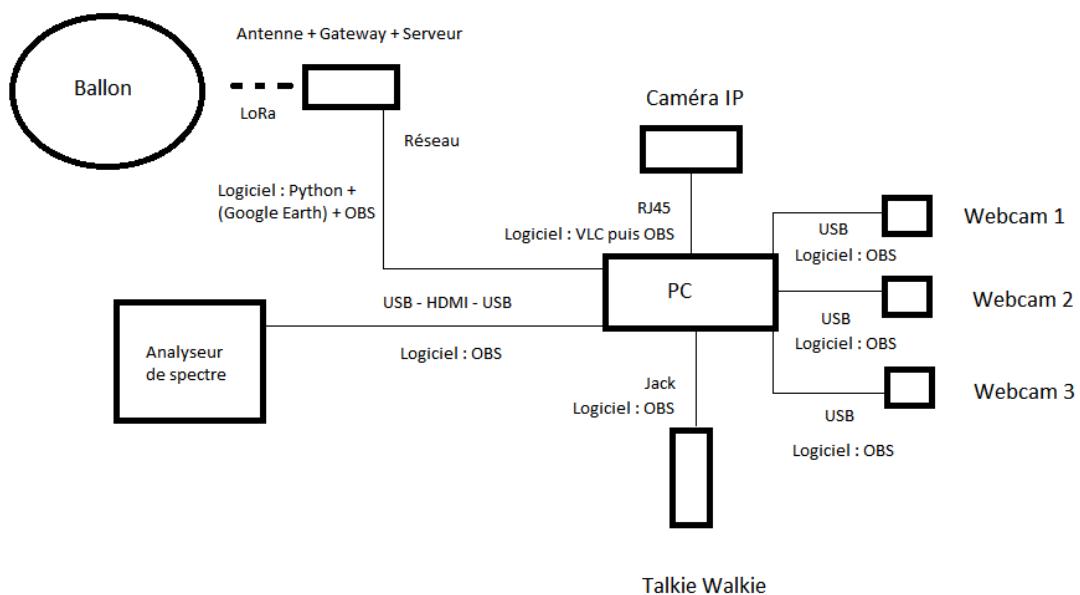


FIGURE 1 – Schéma de l'installation

L'installation comprend 3 caméras de type Webcam USB, une caméra sur IP PoE, un analyseur de spectre, un talkie walkie ainsi que la nacelle du ballon-sonde, qui enverra des données de télémesure en permanence lors du vol.

## 3 Setup du PC

Le PC utilisé possède des caractéristiques particulières qui lui permettent de réaliser cette régie.

### 3.1 Caractéristiques du PC

Il s'agit d'un PC de bureau équipé de 4 écrans full HD, et d'un processeur Intel Core i7 de 8ème génération. En effet, les 4 écrans offrent un avantage considérable dans la réalisation de cette régie, de même que le processeur, qui offre une assez bonne puissance de calcul.

Côté logiciel, le PC tourne sous Ubuntu (ce n'est pas obligatoire, en réalité la réalisation de la régie est optimisée pour Windows, mais cela fonctionne, avec cependant quelques instabilités), et est équipé d'une série de logiciels indispensables :

- OBS studio, qui permet de diffuser le contenu sur YouTube ;
- VLC Media Player, qui permet de visualiser la caméra sur IP ;
- Google Earth Pro, qui permet de visualiser les données GPS ;
- Python, pour faire tourner les différents scripts ;
- Un navigateur internet, pour afficher YouTube, les données JSON et Grafana ;

Une fois ces logiciels installés, il est important de vérifier que la version des pilotes installés est compatible avec tous ces logiciels (notamment les pilotes graphiques), autrement des problèmes critiques pourraient survenir, rendant impossible l'utilisation de la régie.

### 3.2 OBS Studio

Le point central de la régie est OBS Studio. En effet, c'est ce logiciel qui permet de rassembler toutes les sources, de les coordonner, les mettre en forme et les diffuser en streaming.

#### 3.2.1 Fonctionnement en bref

Ce logiciel permet de créer différentes scènes, dans lesquelles se trouvent les sources audio et vidéo :

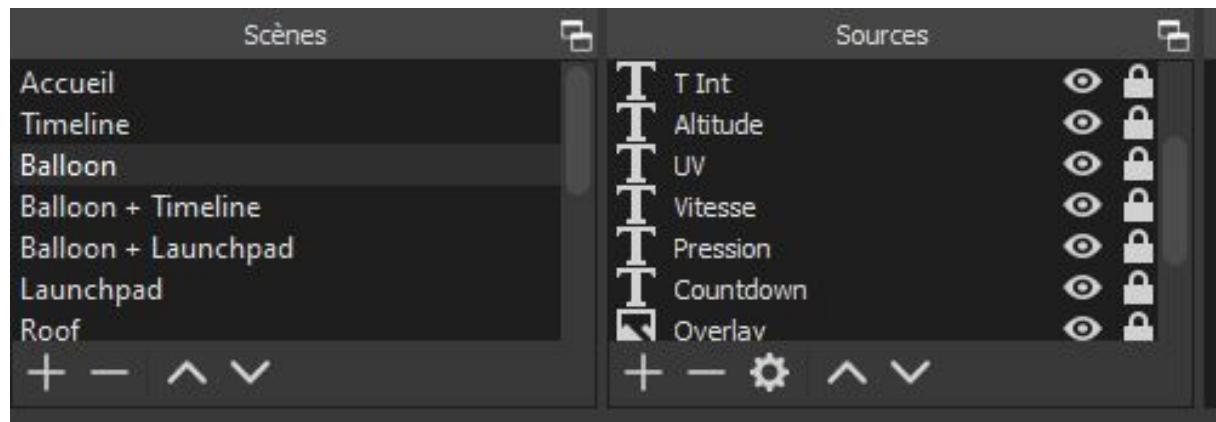


FIGURE 2 – Aperçu des scènes et des sources sur OBS

Ainsi, le logiciel permet de faire la transition entre les différentes scènes afin de proposer différents points de vue par exemple.

### 3.2.2 Fonctions utilisées

Afin de faciliter la gestion de la régie, le mode studio (qui facilite la transition entre les scènes) et le mode affichage multi-écran (qui facilite la navigation entre les scènes) sont utilisés.



FIGURE 3 – Mode studio et aperçu multi-écrans sur OBS

Ainsi, les différentes scènes sont composées de plusieurs sources de différents types. En revanche, certaines sources sont communes à toutes les scènes :

- L’overlay (de type image), qui comporte l’ensemble des logos de nos partenaires ;
- Les données capteurs (type fichier texte), qui diffuse les données envoyées par les capteurs en temps réel ;
- Un fond (de type image), qui évite l’écran noir en cas de dysfonctionnement d’une caméra ;
- Le temps de mission (de type fichier texte), qui permet d’afficher le temps avant le décollage ou le temps écoulé depuis le décollage ;
- Le talkie walkie (de type entrée audio), qui retransmet les communications de la mission ;

Enfin, dans chaque scène sont ajoutées les sources souhaitées (une caméra, une carte d’acquisition, ou une fenêtre d’application par exemple).

## 4 Récupération et affichage des données de vol

Les données de vol sont envoyées en direct via le protocole LoRa et sont reçues sur nos antennes puis nos Gateways. A partir de là il est possible de récupérer les données pour les diffuser.

### 4.1 Données capteurs

Les valeurs des données des capteurs (hors latitude et longitude) sont affichées en direct à l’écran :

UV	Pression	Température Intérieure	SNR : 9 dB	Temps de Mission	Température Extérieure	Vitesse	Altitude
0.042	1.009	27	mW/m²	T+ 0:46:41	°C	0.0	34
	Bar				°C	km/h	m

FIGURE 4 – Affichage des données de vol

Pour cela elles sont dans un premier temps récupérées via un script Python puis diffusées.

#### 4.1.1 *Script Python*

Le script Python utilise la bibliothèque JSON afin de lire les données des capteurs mises à disposition sur un serveur local au format JSON.

Une fois récupérées, ces données sont parsées afin de séparer chaque donnée selon son type (altitude, pression, UV, etc...), puis écrites dans un fichier texte. Ainsi chaque donnée est écrite sur un fichier texte séparé, et ce sont ces fichiers textes qui seront affichés. Concernant la vitesse du ballon, elle est calculée par le script à partir de la latitude, la longitude et l'altitude puis écrite dans un fichier.

#### 4.1.2 *Affichage des données*

Afin d'afficher et diffuser les données capteurs, chaque fichier texte généré via Python est ajouté comme source sur les scènes du logiciel OBS.

La police, ou la couleur du texte peuvent être changés via le logiciel OBS. La valeur se met automatiquement à jour en fonction de la valeur écrite dans le fichier qui est lu.

## 4.2 Données GPS

Les données GPS (latitude, longitude, altitude) sont également transmises par le module LoRa afin de connaître la position du ballon. La position en direct du ballon est également retransmise via Google Earth Pro

#### 4.2.1 *Script Python*

Le script Python chargé de récupérer les données du module LoRa au format JSON récupère également les données GPS. Cependant au lieu d'être écrites dans des fichiers texte pour être diffusées (sauf l'altitude), celles-ci sont écrites dans un fichier KML (qui est un format dérivé du XML lisible par Google Earth). Ainsi, une fonction python est chargée d'écrire le fichier KML en entier à chaque fois que de nouvelles données sont reçues. L'ensemble des positions envoyées par le GPS sont également écrites dans le fichier KML afin de tracer le trajet du ballon au fur et à mesure qu'il se déplace.

#### 4.2.2 *Affichage des données*

Afin d'afficher les données en direct via Google Earth, un premier fichier KML doit être ouvert sur le logiciel. Dans ce fichier se trouve une redirection vers le fichier KML généré par le script Python. Cette redirection permet de maintenir l'affichage à jour au fur et à mesure que de nouvelles données sont reçues. La période de rafraîchissement peut être définie via les propriétés du fichier directement depuis l'interface de Google Earth Pro.

Pour afficher la carte de Google Earth, une source de type Capture de fenêtre peut être ajoutée sur OBS, afin de diffuser la fenêtre de Google Earth en direct.



FIGURE 5 – Affichage de la position du ballon sur Google Earth puis OBS

Sur cette image, le ballon est représenté par le logo Eirballoon, et le trajet du ballon est tracé en bleu. Ces attributs sont modifiables directement dans le fichier KML.

### 4.3 Timeline

La timeline du vol permet de voir où se situe le ballon par rapport au début et à la fin de son trajet, ainsi que par rapport à certains événements marquant du vol (éclatement du ballon, déploiement du parachute...). De même une échelle d'altitude est donnée, afin de comparer l'altitude actuelle du ballon avec des points connus (comme l'altitude type d'un A380, d'un Concorde, l'entrée dans la stratosphère, etc...).

#### 4.3.1 Script Python

Afin de faire cette timeline et cette échelle des altitudes, une image est générée par un script Python. Cette image représente une courbe sur laquelle se déplace un point représentant le ballon à sa position actuelle vis à vis du trajet complet. Pour réaliser les calculs, une prévision de l'apogée du ballon est estimée grâce aux outils de prévision, afin que l'échelle soit respectée, et que par exemple le point "éclatement du ballon" se trouve à la bonne altitude sur la courbe. L'apogée est ainsi écrite manuellement dans un fichier texte afin d'être lue par le script Python générant la courbe.

#### 4.3.2 Affichage

Afin d'afficher tout cela sur OBS, un template est ajouté en tant que source de type image sur OBS. Par dessus sont superposées les courbes générées par Python, également sous forme de source de type image :



FIGURE 6 – Affichage de la timeline sur OBS

Ainsi le point se déplace au fur et à mesure que le ballon avance dans son périple.

### 4.4 Compte à rebours

Le compte à rebours permet d'afficher le temps restant avant le décollage ainsi que le temps écoulé après le décollage.

#### 4.4.1 Script Python

Pour réaliser le compte à rebours, un script Python est utilisé. La date et l'heure exacte prévue du lâché est rentrée manuellement dans un fichier texte, et le script calcule la différence de temps entre l'heure actuelle et l'heure du lâché (grâce à la bibliothèque "time"), et cette différence est écrite dans un fichier texte, précédée de "T-" ou T+" en fonction de si l'on se trouve avant ou après le décollage.

#### 4.4.2 Affichage

Le fichier texte est affiché grâce à OBS en ajoutant une source de type fichier texte. Celui se met en permanence à jour de façon à pouvoir observer le décompte en direct :



FIGURE 7 – Affichage du compte à rebours

## 5 Ajout des webcams USB

### 5.1 Description

Les webcams utilisées sont des webcams USB Full HD standards trouvables dans le commerce. L'une se trouve sur le toit, en direction des antennes, une se trouve sur le toit en direction du parking afin d'observer le décollage et la dernière se trouve dans le club radio. Chaque caméra est reliée directement au PC via une rallonge USB de longueur suffisante (10m, 15m et 5m). Les câbles passent par la fenêtre si les caméras se trouvent sur le toit. Il faut également penser à les protéger des intempéries.

### 5.2 Affichage

Afin de les afficher sur OBS, les webcams doivent être ajoutées en tant que source de type Périphérique de capture vidéo sur les scènes où les webcams doivent être affichées. La résolution, le nombre de FPS ou même la colorimétrie peuvent être réglés directement via les propriétés du périphérique sur OBS.



FIGURE 8 – Affichage webcam en direct

## 6 Ajout de la caméra IP

Une caméra sur IP est installée sur le toit en direction du rotor de l'antenne Yagi-Uda utilisée. Elle permet d'observer la rotation du rotor en direct.

### 6.1 Installation

Cette caméra fonctionne en PoE (Power over Ethernet), ce qui signifie que son alimentation se fait par câble RJ45, le même que celui qui sert au transfert du flux vidéo.

Pour cela, la caméra est reliée par câble RJ45 sur un injecteur PoE. Le PC du club radio est lui même relié en RJ45 sur ce même injecteur PoE afin de pouvoir lire le flux vidéo. Une fois l'installation faite, le flux vidéo peut être récupéré en connaissant l'adresse IP de la caméra.

## 6.2 Affichage

Pour afficher le flux vidéo de la caméra, il faut ouvrir un flux réseau avec le logiciel VLC Media Player. Ce flux réseau permet d'afficher la vidéo en connaissant l'adresse IP de la caméra. Pour ce faire, l'IP peut directement être rentrée dans VLC ou bien écrite dans un fichier .asf lisible par VLC.

Une fois cela fait, une source de type capture de fenêtre peut être ajoutée sur OBS, afin d'afficher la fenêtre de VLC en direct sur OBS. Le flux réseau peut également être directement lu sur OBS (en ajoutant une source de type Source Vidéo VLC), mais cela ne semble pas fonctionner sous Linux et cette version d'OBS.



FIGURE 9 – Affichage du retour de la caméra IP

# 7 Acquisition de l'analyseur de spectre

## 7.1 Installation

L'analyseur de spectre du club radio tourne sous Windows 7. Il est donc possible de récupérer son flux vidéo via USB et de réaliser une acquisition depuis le PC du club radio.

Pour ce faire, un adaptateur USB vers HDMI est installé sur un port USB de l'analyseur de spectre. Côté PC, un périphérique de capture avec un port HDMI en entrée et un port USB en sortie est branché sur le PC. Enfin un câble HDMI de 3m est installé entre les deux pour transmettre le flux.

## 7.2 Affichage

Une fois l'installation faite, il suffit d'ouvrir le périphérique sur OBS en ajoutant une source de type Périphérique de capture vidéo, comme pour les webcams. Soit cela fonctionne normalement, soit le périphérique est introuvable (il est en réalité détecté comme un périphérique de capture audio par le système) et le PC doit alors être redémarré jusqu'à ce que tout fonctionne.

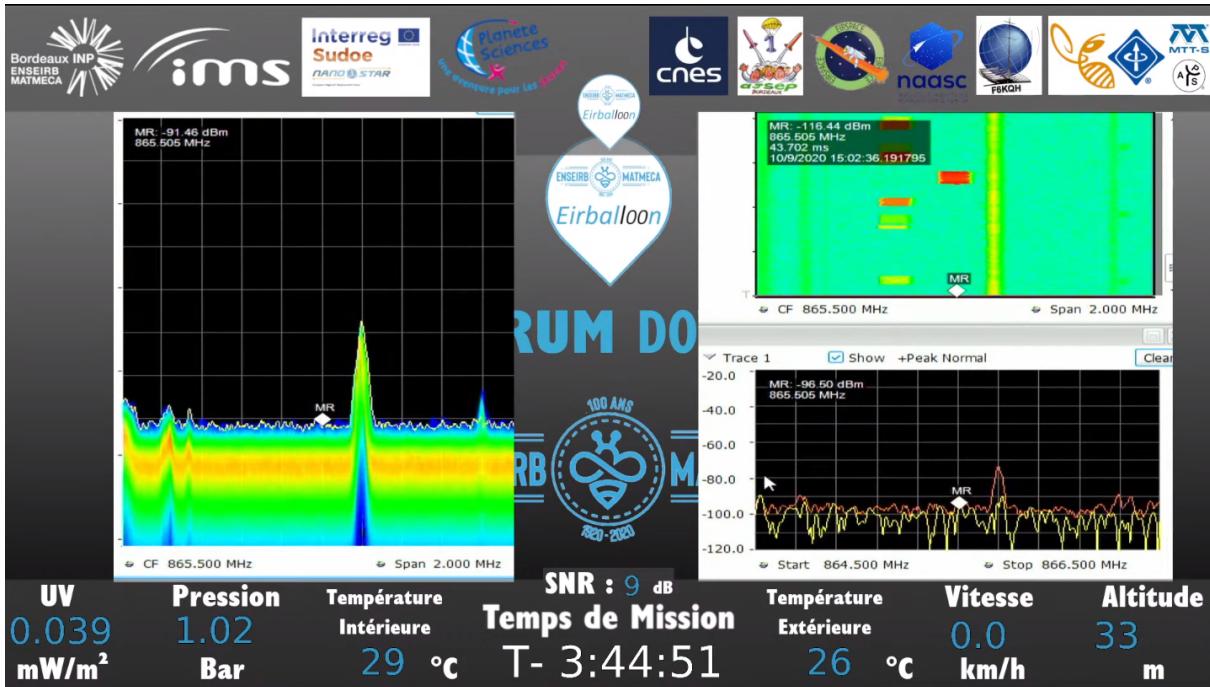


FIGURE 10 – Acquisition de l'analyseur de spectre

# 8 Talkie Walkie

Lors de l'événement, l'équipe communiquera par talkie walkie. Les communications seront également retransmises en direct.

## 8.1 Installation

La sortie haut parleur du talkie walkie est de type mono Jack 2,5mm mais l'entrée son du PC est de type stéréo Jack micro + haut parleur 3,5mm. Il faut donc mettre un convertisseur Jack 2,5mm mâle vers Jack 3,5mm femelle en sortie de talkie walkie et un séparateur Jack 3,5mm au niveau du PC afin de séparer l'entrée micro de la sortie casque. Entre les deux, un câble Jack fait maison permettant de passer d'un son mono à un son stéréo a été installé afin de transmettre le flux audio.

## 8.2 Diffusion

Pour retransmettre le flux audio sur OBS, il faut ajouter une source de type Entrée Audio sur chacune des scènes devant retransmettre les communications de l'équipe. Ainsi les communications pourront être transmises en direct sur Youtube.

## 9 Récapitulatif et diffusion

### 9.1 Récapitulatif

Pour récapituler, chacune des sources vidéos évoquées font l'objet d'une scène à part entière afin de les diffuser en plein écran. De même, des scènes sont prévues pour montrer deux sources vidéos en vis à vis. Il a donc fallu créer autant de scènes distinctes que de combinaisons de deux sources vidéos possible afin de pouvoir alterner les angles de caméra.

### 9.2 Diffusion

Afin de diffuser le stream, il faut disposer d'une chaîne YouTube (ou autre) habilitée à faire des diffusions en direct. Une fois cela fait, la clé de stream peut être récupérée directement depuis YouTube, afin d'être rentrée dans OBS.

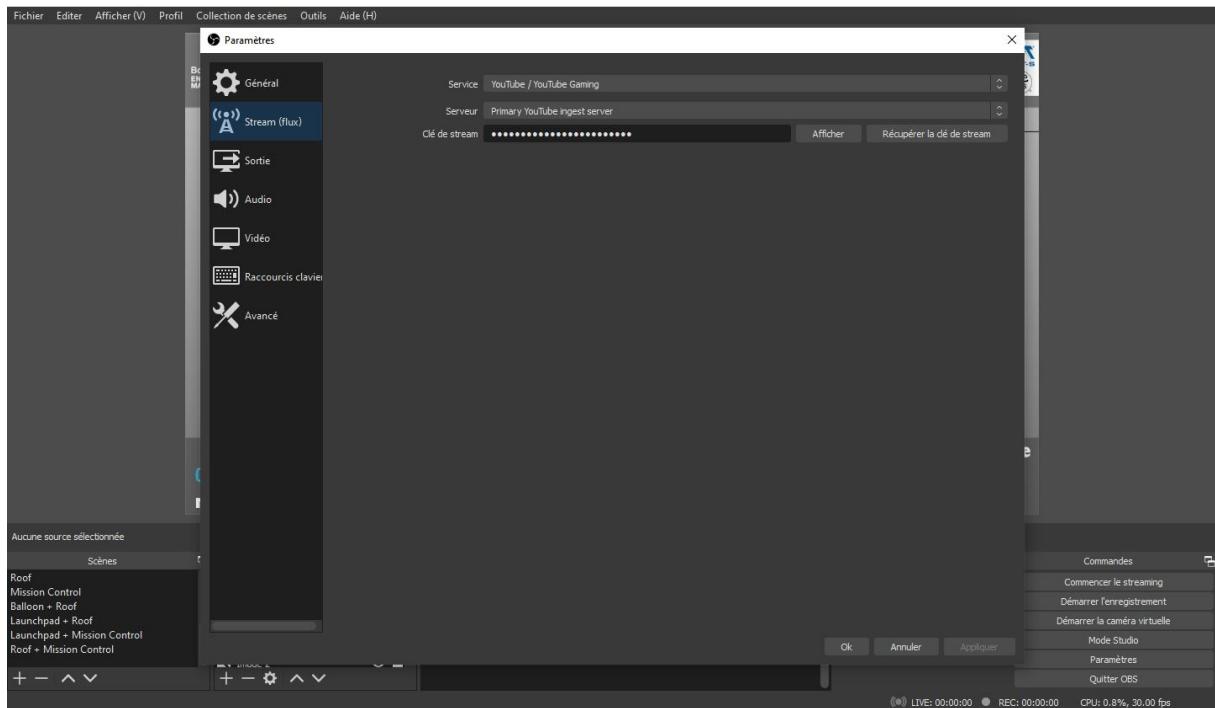


FIGURE 11 – Interface permettant de rentrer la clé de stream

De même la qualité, la résolution et le débit du stream peuvent être réglés dans les paramètres de stream via OBS. Une fois tout cela fait, il suffit de cliquer sur "Commencer le streaming". La vidéo est alors diffusée sur YouTube.