

# Projet S2

## Découpage tâches

---

### SOMMAIRE

#### Régie Audio

---

- [Tâche A1](#) – Prise de son et intégration des flux audio dans MAX
- [Tâche A2](#) – Mise en place synthèse sonore MAX/MSP
- [Tâche A3](#) – Mise en place effets sonores MAX/MSP
- [Tâche A4](#) – Mise en place de la table de mixage
- [Tâche A5](#) – Module envoi réseau du flux audio
- [Tâche A6](#) – Mise en place du module de pilotage
- [Tâche A7](#) – Intégration des différents modules de la régie audio

#### Régie Vidéo

---

- [Tâche V1](#) – Module réception réseau du flux audio
- [Tâche V2](#) – Routage des flux audio vers les pistes vidéo
- [Tâche V3](#) – Module d'analyse audio
- [Tâche V4](#) – Recherche image/vidéo et intégration des flux vidéo dans MAX
- [Tâche V5](#) – Mise en place effets vidéo MAX/JITTER
- [Tâche V6](#) – Mise en place du mixage
- [Tâche V7](#) – Intégration des différents modules de la régie vidéo

#### Interface

---

- [Tâche I1](#) – Pilotage de la régie Audio
- [Tâche I2](#) – Pilotage de la régie Vidéo

## Tâche A1 – Prise de son et intégration des flux audio dans MAX

---

Objectif : enregistrer et intégrer les sons dont vous avez besoin pour votre application.

Documentation :

- [Tuto prise de son avec mixette et enregistreur](#)
- lecteur audio : cf. démo-TS

Ressources :

- lecteur audio : cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A1,
- Prise de son de vos éléments sonores. Prenez une photo par élément sonore enregistré afin de garder une trace des conditions d'enregistrement,
- Découpage des différents extraits en n fichiers indépendants au format WAVE avec le codec PCM 16 bits,
- Créer une abstraction lecteurAudio. Afin de n'avoir qu'un seul code pour tous vos lecteurs, vous définirez un argument à votre abstraction MAX qui correspondra au nom du fichier avec l'extension. Vous définirez vous-même les commandes de ce lecteur audio (start, pause, reprise, stop, loop),
- Instancier autant de fois votre abstraction lecteurAudio que de sons utilisés dans votre application,
- Tester vos abstractions dans le main de la régie Audio,
- Fusionner la branche A1 avec la branche principale.

Résultat attendu : une abstraction lecteurAudio.maxpat.

Entrée : paramètres de contrôle du lecteur audio

Sortie : 1 flux audio mono

Remarque :

- respecter les règles de nommage de vos fichiers audio (cf. projetS2RèglesNommageFichierAudio.pdf)
- pas de post-prod, tous les effets doivent être faits avec MAX/MSP

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche A2 – Mise en place synthèse sonore MAX/MSP

---

Objectif : l'objectif de cette tâche est de synthétiser un bourdon (drone en anglais)

### Documentation :

- projetS2MapEffetsAnalyseSynthèseAudio.pdf
- accords.pdf
- noteVsFrequence.pdf

### Ressources :

- mainSyntheseAudio.maxpat
- syntheseFM.maxpat
- syntheseSoustractive.maxpat
- splitAccord.maxpat

### Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A2,
- Tester les deux abstractions syntheseFM et syntheseSoustractive,
- En fonction du rendu sonore, choisir l'une de ces deux synthèses,
- Réglage des paramètres de votre synthèse pour obtenir le son souhaité,
- Choix de l'accord,
- Synthèse de l'accord,
- Tester l'abstraction dans le main de la régie Audio,
- Fusionner la branche A2 avec la branche principale.

Résultat attendu : une abstraction syntheseSonore.maxpat

Entrée : paramètres de contrôle de la synthèse

Sortie : un flux audio mono ou stéréo

### Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

### Remarque :

Si vous le souhaitez, vous pouvez faire changer les accords au cours du temps.

## [SOMMAIRE](#)

## Tâche A3 – Mise en place effets sonores MAX/MSP

---

Objectif : L'objectif de cette tâche est de mettre en oeuvre deux effets audio.

Documentation :

- projetS2MapEffetsAnalyseSynthèseAudio.pdf

Ressources :

- mainEffetAudio.maxpat

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A3,
- Choisir deux effets audio parmi la liste des effets audio proposés,
- Écrire l'abstraction de l'effet 1,
- Appliquer cet effet à un flux audio dry,
- Écrire l'abstraction de l'effet 2,
- Appliquer cet effet à un flux audio dry,
- Tester vos abstractions dans le main de la régie Audio,
- Fusionner la branche A3 avec la branche principale.

Résultat attendu : deux abstractions effetAudio1.maxpat et effetAudio2.maxpat.

Entrée : un flux audio dry / paramètres de contrôle de l'effet  
Sortie : un flux audio wet mono ou stéréo

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche A4 – Mise en place de la table de mixage

---

Objectif : L'objectif de cette tâche est de mixer tous les flux audio générés en un seul flux audio stéréo. Le flux audio stéréo est diffusé par les hauts-parleurs de la régie Audio.

Documentation : -

Ressources :

- utiliser l'objet [live.gain~ @channels 1] pour un flux audio mono
- utiliser l'objet [live.gain~ @channels 2] pour un flux audio stéréo

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A4,
- Écrire l'abstraction mixageAudio,
- Sortir le flux audio mixé,
- Tester votre abstraction dans le main de la régie Audio,
- Fusionner la branche A4 avec la branche principale.

Résultat attendu : une abstraction mixageAudio.maxpat

Entrée : n flux audio dry / paramètres de contrôle du mixage

Sortie : un flux audio stéréo

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche A5 – Module envoi réseau du flux audio

---

Objectif : L'objectif de cette tâche est d'envoyer un flux audio par le réseau à destination de la régie vidéo.

Documentation :

- cf. démo-TS

Ressources :

- cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A5,
- Écrire l'abstraction moduleEnvoiAudio.maxpat,
- Tester votre abstraction dans le main de la régie Audio,
- Fusionner la branche A5 avec la branche principale.

Résultat attendu : une abstraction moduleEnvoiAudio.maxpat

Entrée : 1 flux audio mono / paramètres de contrôle du module d'envoi

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche A6 – Mise en place du module de pilotage

---

Objectif : L'objectif de cette tâche est de mettre en place le module de réception des données OSC envoyées par l'interface afin de piloter les différents modules de la régie Audio.

Documentation :

- cf. démo-TS

Ressources :

- cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A6,
- Écrire l'abstraction `receptionOSC.maxpat`,
- Tester votre abstraction dans le main de la régie Audio,
- Fusionner la branche A6 avec la branche principale.

Résultat attendu : une abstraction `receptionOSC.maxpat`

Sortie : n flux de contrôle

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche A7 – Intégration des différents modules de la régie audio

---

Objectif : L'objectif est d'intégrer dans le main de la régie Audio tous les modules des tâches précédentes. Normalement si tous les modules ont été écrits conformément aux différents descriptifs des tâches, le main et le schéma fonctionnel de la régie Audio devraient être similaires.

Documentation : -

Ressources : -

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche A7,
- Écrire le main conformément au schéma fonctionnel,
- Tester l'ensemble,
- Fusionner la branche A7 avec la branche principale.

Résultat attendu : compléter le main de la régie Audio

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)



## Tâche V1 – Module réception réseau du flux audio

---

Objectif : L'objectif de cette tâche est de mettre en place le module de réception du flux audio envoyé par la régie Audio.

Documentation :

- cf. démo-TS

Ressources :

- cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V1,
- Écrire l'abstraction moduleReceptionAudio.maxpat,
- Tester l'intégration de ce module dans le main de la régie Vidéo,
- Fusionner la branche V1 avec la branche principale.

Résultat attendu : l'abstraction moduleReceptionAudio.maxpat

Sortie : 1 flux audio mono

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche V2 – Routage des flux audio vers les pistes vidéo

---

Objectif : L'objectif est de router les flux audio réceptionnés par le réseau vers la ligne vidéo souhaitée.

Documentation : -

Ressources :

- routerAudio.maxpat

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V2,
- Écrire l'abstraction routerAudio.maxpat,
- Tester l'intégration de ce module dans le main de la régie Vidéo,
- Fusionner la branche V2 avec la branche principale.

Résultat attendu : une abstraction routerAudio.maxpat

Entrée : n flux audio / paramètres de contrôle du routage

Sortie : n flux audio

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche V3 – Module d'analyse audio

---

Objectif : L'objectif de cette tâche est de mettre en place un ou plusieurs module(s) d'analyse audio. Plusieurs modules d'analyse de flux audio sont proposés : détection du pitch, détection d'enveloppe et analyse spectrale.

Documentation :

- projetS2MapEffetsAnalyseSynthèseAudio.pdf

Ressources :

- mainAnalyseAudio.maxpat

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V3,
- Comprendre le principe de chaque module d'analyse audio,
- Écrire les modules d'analyse audio de votre application,
- Tester l'intégration de ce(s) module(s) dans le main de la régie Vidéo,
- Fusionner la branche V3 avec la branche principale.

Résultat attendu : une ou plusieurs abstractions analyseAudio.maxpat

Entrée : 1 flux audio mono

Sortie : donnée de contrôle de l'effet vidéo

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche V4 – Recherche image/vidéo et intégration des flux vidéo dans MAX

---

Objectif : L'objectif de ce module est de produire les flux vidéos dry de votre application.

### Documentation :

- cf. démo-TS

### Ressources :

- cf. démo-TS

### Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V4,
- Choisir les médias (image et/ou vidéo) que vous voulez utiliser dans votre application,
- Créer une abstraction lecteurVideo.maxpat. Afin de n'avoir qu'un seul code pour tous vos lecteurs, vous définirez un argument à votre abstraction MAX qui correspondra au nom du fichier avec l'extension. Vous définirez vous-même les commandes de ce lecteurVideo (start, pause, reprise, stop, loop),
- Instancier autant de fois votre abstraction lecteurVideo que de médias utilisés dans votre application,
- Tester vos abstractions dans le main de la régie Vidéo,
- Fusionner la branche V4 avec la branche principale.

Résultat attendu : une abstraction lecteurVideo.maxpat.

Entrée : paramètres de contrôle du lecteur vidéo

Sortie : 1 flux vidéo

### Remarque :

format des médias vidéo: Quicktime ou MPEG-4

format des médias image : JPEG ou PNG

pas de post-prod, tous les effets doivent être faits avec MAX/JITTER

### Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche V5 – Mise en place effets vidéo MAX/JITTER

---

Objectif : L'objectif de cette tâche est de mettre en oeuvre autant d'effets vidéo que de médias utilisés.

### Documentation :

- projetS2MapEffetsVidéo.pdf

### Ressources :

- mainEffetVideo.maxpat

### Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V5,
- Choisir les effets vidéo parmi la liste des effets vidéo proposés,
- Écrire autant d'abstractions que d'effets vidéo,
- Appliquer chaque effet à un flux vidéo dry,
- Tester vos abstractions dans le main de la régie Vidéo,
- Fusionner la branche V5 avec la branche principale.

Résultat attendu : plusieurs abstractions effetVideo.maxpat.

Entrée : un flux vidéo dry / données de contrôle de l'effet

Sortie : un flux vidéo wet

### Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)

## Tâche V6 – Mise en place du mixage

---

Objectif : L'objectif de ce module est de mixer tous les flux vidéo wet en un seul flux vidéo. Le flux vidéo ainsi généré est diffusé par la régie Vidéo.

Documentation : -

Ressources :

- assemblage.maxpat

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V6,
- Écrire l'abstraction mixageVideo.maxpat,
- Tester l'abstraction dans le main de la régie Vidéo,
- Fusionner la branche V6 avec la branche principale.

Résultat attendu : une abstraction mixageVideo.maxpat

Entrée : n flux vidéo dry / paramètres de contrôle du mixage

Sortie : 1 flux vidéo

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche V7 – Intégration des différents modules de la régie vidéo

---

Objectif : L'objectif est d'intégrer dans le main de la régie Vidéo tous les modules des tâches précédentes. Normalement si tous les modules ont été écrits conformément aux différents descriptifs des tâches, le main et le schéma fonctionnel de la régie Vidéo devraient être similaires.

Documentation : -

Ressources : -

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche V7,
- Écrire le main conformément au schéma fonctionnel,
- Tester l'ensemble,
- Fusionner la branche V7 avec la branche principale.

Résultat attendu : compléter le main de la régie Vidéo

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)

## Tâche I1 – pilotage de la régie Audio

---

Objectif : L'objectif de cette tâche est d'écrire le module de pilotage de la régie Audio.

Documentation :

- cf. démo-TS

Ressources :

- cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche I1,
- Définir les paramètres de la régie Audio à piloter,
- Définir l'interface de pilotage de ces paramètres,
- Écrire le module d'envoi des données OSC correspondantes,
- Tester l'envoi des données OSC à la régie Audio,
- Fusionner la branche I1 avec la branche principale.

Résultat attendu : compléter le main de l'interface

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

## [SOMMAIRE](#)



## Tâche I2 – pilotage de la régie Vidéo

---

Objectif : L'objectif de cette tâche est d'écrire le module de pilotage de la régie Vidéo.

Documentation :

- cf. démo-TS

Ressources :

- cf. démo-TS

Ce qu'il y a faire :

- À partir de la branche principale, créer la branche I2,
- Définir les paramètres de la régie Vidéo à piloter,
- Définir l'interface de pilotage de ces paramètres,
- Écrire le module d'envoi des données OSC correspondantes,
- Tester l'envoi des données OSC à la régie Vidéo,
- Fusionner la branche I2 avec la branche principale.

Résultat attendu : compléter le main de l'interface

Historique :

04/04/17 : version initiale

03/05/17 : ajout consignes Git

[SOMMAIRE](#)