

**DEPARTEMENT INFORMATIQUE - IUT 2 GRENOBLE**



Année Universitaire 2022-2023

**MEMOIRE DE STAGE**

---

**DÉVELOPPEMENT DE LOGICIELS FAVORISANT LA  
COMMUNICATION DES PERSONNES EN SITUATION DE HANDICAP**

**Laboratoire Informatique de Grenoble**



---

**Présenté par**

**Yanis Girardin**

**Jury**

**IUT : M. PORTET**

**IUT : Mme. LAGET-KAMEL**

**Société : M. SCHWAB**

**Déclaration de respect des droits d'auteurs**

Par la présente, je déclare être le seul auteur de ce rapport et assure qu'aucune autre ressource que celles indiquées n'ont été utilisées pour la réalisation de ce travail. Tout emprunt (citation ou référence) littéral ou non à des documents publiés ou inédits est référencé comme tel.

Je suis informé qu'en cas de flagrant délit de fraude, les sanctions prévues dans le règlement des études en cas de fraude aux examens par application du décret 92-657 du 13 juillet 1992 peuvent s'appliquer. Elles seront décidées par la commission disciplinaire de l'UGA.

A, Grenoble

Le, 28/04

Signature

A handwritten signature in black ink, appearing to read "Yanis Girardin". The signature is fluid and cursive, with a long horizontal line extending from the end of the main stroke.

## **Remerciements**

Je remercie M. SCHWAB de m'avoir donné l'occasion de vivre cette expérience de stage et d'en avoir été le responsable en s'assurant qu'il se passe dans les meilleures conditions.

Je remercie M. ARRIGO de m'avoir accompagné et soutenu sur les tâches qui m'ont été attribuées durant ce stage.

Je remercie Paul SODE, Noha BOUTEMEUR, Aubin MOURAS et Hugolin BOUHINEAU, qui ont été dans le même bureau que moi, pour leur entraide et leur solidarité.

Je remercie M. PORTET, mon tuteur IUT, de m'avoir accompagné durant ma période de stage.

## Résumé

---

Ce rapport explique ce que j'ai pu vivre et produire durant ma période de stage. Mon stage a eu lieu au Laboratoire Informatique de Grenoble, spécialisé dans la recherche et le développement des différents domaines de l'informatique. J'ai été introduit au sein du [Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole \(GETALP\)](#)\*, pour travailler sur l'un de leurs projets : [InterAActionBox](#)\*, un projet qui a pour but de favoriser l'apprentissage et la communication des personnes en situation de handicap cognitif, du plus petit au plus grand âge. Il regroupe plusieurs applications de [Communication Alternative et Augmentée](#)\*, dont [InterAActionScene](#)\*, une application permettant à l'utilisateur de créer des [scènes](#)\* et des images pour interagir avec elles. Mes missions portaient essentiellement sur cet application et consistaient à lui développer de nouvelles fonctionnalités.

Pour travailler sur ce projet, l'équipe GETALP\* utilise [Discord](#)\* pour communiquer entre eux. Pour gérer l'avancement du développement du projet, ils utilisent [Github](#)\*. Quant au développement du projet en lui-même, ils utilisent l'[Environnement de Développement Intégré \(IDE\)](#)\* IntelliJ développé par JetBrains.

Durant ce stage, mes tâches à faire m'étaient communiquées par Jordan ARRIGO, sur [Discord](#)\*. Le matin, j'annonçais mon planning de la journée, et le soir, je faisais un récapitulatif de ce qui a été fait durant cette journée. De plus, nous faisions une visio-conférence une fois par semaine pour faire un rapport hebdomadaire.

Ma première tâche était de me former au framework [Angular CLI](#)\* et à [Karma/Jasmine](#)\* grâce à des tutoriels disponibles sur internet. En effet, ces deux dernières sont des nouvelles technologies pour moi. Ainsi, j'ai pu analyser et comprendre l'application [InterAActionScene](#)\*, qui est développée avec ces langages, afin de commencer le développement de celle-ci.

Les premières réalisations à faire sur [InterAActionScene](#)\* étaient tirées des [issues](#)\* de sa page [Github](#)\*. Lorsque toutes ces issues fut réglées, j'identifiais des bugs à corriger et des nouvelles fonctionnalités intéressantes à implémenter dans [InterAActionScene](#)\*. Une fois fait, j'en faisais un rapport à Jordan ARRIGO pour échanger et apporter une décision finale sur ces idées. Ainsi, je pouvais commencer à développer celles qui ont été validées.

Tout cela m'a permis d'implémenter plusieurs nouvelles fonctionnalités dans l'application [InterAActionScene](#)\*, comme améliorer les interactions avec les images, notamment grâce à des zones cliquables appelées "hotspots\*" ou le panel de dessin, améliorer la gestion des [scènes](#)\* et des images, etc... Les fichiers de tests ont eux aussi été modifiés pour vérifier le bon fonctionnement de chaque modification que j'ai pu apporter.

Ce stage m'a permis d'apprendre les nouvelles technologies utilisées dans mes missions ainsi qu'améliorer ma maîtrise de celles que je connaissais déjà. Toutes les missions ont consolidé mes compétences en développement Web et ont élevé mon intérêt pour ce domaine. De plus, cette expérience en entreprise a été très plaisante, et me motive à m'insérer dans ce monde et à continuer mes études en alternance.

## Table des matières

---

<b>I. Introduction.....</b>	<b>7</b>
I.1. Présentation de l'entreprise.....	7
I.2. Présentation du stage.....	7
<b>II. Initiation à Angular CLI et Karma/Jasmine.....</b>	<b>9</b>
II.1. Présentations.....	9
II.2. Les Tutoriels.....	9
II.3. Bilan.....	10
<b>III. Analyse de l'existant.....</b>	<b>11</b>
III.1. Technologies et logiciels utilisés.....	11
III.2. L'application InterAActionScene.....	11
III.3. La construction du code.....	12
<b>IV. Réalisations.....</b>	<b>13</b>
IV.1. Gestion de projet.....	13
IV.2. Nouvelles fonctionnalités.....	13
Navigation entre les images grâce aux hotspots.....	14
Ajout et modification du panel de dessin.....	16
Pouvoir renommer les scènes et les images indépendamment.....	17
Pouvoir organiser nos scènes et les images librement.....	17
Suppression de tous les hotspots d'une image.....	18
Remplir automatique le nom à la création d'un objet.....	19
Mise à jour du tutoriel.....	19
IV.3. Tests.....	20
Supprimer toutes les scènes et toutes les images d'un coup.....	20
Navigation entre les images grâce aux hotspots.....	20
Ajout et modification du panel de dessin.....	20
Pouvoir renommer les scènes et les images indépendamment.....	20
Pouvoir organiser nos scènes et nos images librement.....	20
Suppression de tous les hotspots d'une image.....	20
Remplir automatiquement le nom à la création d'un objet.....	20
<b>V. Conclusion.....</b>	<b>21</b>
<b>Annexes.....</b>	<b>22</b>
<b>Glossaire.....</b>	<b>32</b>
<b>Sitographie.....</b>	<b>34</b>

## Table des figures

---

Figure 1 : Interface principal de InterAActionScene.....	11
Figure 2 : Barre des tâches avec le bouton “Recommencer”.....	13
Figure 3 : Fenêtre de confirmation.....	14
Figure 4 : Diagramme de classe de Hotspot avant et après modifications, créé avec Lucid.....	14
Figure 5 : Fonction PlayHotspot, qui exécute l'action du hotspot lors de son clique.....	16
Figure 6 : Nouveau panel de dessin de InterAActionScene.....	17
Figure 7 : Fenêtre de renommage avant modifications.....	17
Figure 8 : Fenêtre de renommage après modifications.....	17
Figure 9 : Boutons zoom et dézoom.....	18
Figure 10 : Les actions de hotspots après modification.....	19

## I. Introduction

---

### I.1. Présentation de l'entreprise (rédigé avec Noha B. et Paul S.)

Le stage s'est produit au Laboratoire Informatique de Grenoble (LIG), un laboratoire de recherche créé le 1er janvier 2007, dirigé aujourd'hui par Noël DE PALMA, Sihem AMER YAHIA, Patrick REIGNIER et Pascale POULET. D'après son livret de 2021 [1 et 2], le LIG regroupe environ 450 chercheurs, enseignants-chercheurs, doctorants et personnels en soutien à la recherche. Il est de plus en partenariat avec l'Université Grenoble-Alpes (UGA), l'INP (Institut Polytechnique) de Grenoble, le Centre National de la Recherche Scientifique (CNRS) et l'Institut National de Recherche en Informatique et en Automatique (INRIA). Le laboratoire est étendu sur trois sites : un sur le campus, au sein du bâtiment IMAG (Institut d'informatique et Mathématiques Appliquées de Grenoble), un sur le campus Minatec et un autre à Montbonnot. Le LIG se veut un acteur actif à la recherche et au développement des différents domaines de l'informatique. Il comporte au total 22 équipes portant différentes recherches et projets selon cinq axes thématiques de recherche, qui sont les suivants : Génie des logiciels et des systèmes d'informations ; Méthodes formelles, modèles et langages ; Système intelligents pour les données, les connaissances et les humains ; Systèmes interactifs et cognitifs ; Systèmes répartis, calcul parallèle et réseaux. Sur l'[organigramme](#) disponible en annexe, nous pouvons voir les liens entre les différents acteurs du LIG [5].

### I.2. Présentation du stage

Dans le cadre de mon stage, j'ai été intégré dans le Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole ([GETALP\\*](#)) menant des recherches sur les Systèmes intelligents pour les données, les connaissances et les humains. Ses recherches sont centrées sur le développement de la communication et du traitement de l'information multilingue [3].

Un des projets de cette équipe nommé [InterAActionBox\\*](#) a pour but de favoriser l'apprentissage et la communication des personnes en situation de handicap cognitif, du plus petit au plus grand âge. Plus particulièrement, il intègre un ordinateur ou une tablette permettant des interactions oculaires et tactiles, un système d'exploitation qui gère [InterAActionBox\\*](#) et plusieurs applications de Communication Alternative et Augmentée ([CAA\\*](#)). La [CAA\\*](#) représente l'ensemble des techniques et outils permettant aux personnes ayant des difficultés pour communiquer de s'exprimer ou de comprendre les autres, selon EpsilonMelia [4]. Les interactions oculaires sont d'abord testées au LIG grâce à un [oculomètre\\*](#), plus particulièrement avec le Tobii Eye Tracker 5, qui mesure les mouvement des yeux (cf. [Annexe 19](#)).

Le sujet de mon stage portait sur l'une de ces applications appelée [InterAActionScene\\*](#), permettant à l'utilisateur de créer des [scènes\\*](#), qui sont des objets représentant des listes d'images. Il peut donc leur attribuer des images et d'y placer dessus des dessins ou des [hotspots\\*](#). Ces [hotspots\\*](#) sont des zones que l'on peut placer et qui produisent des sons choisis par l'utilisateur à leur clique. Les objectifs du stage ont été définis par un [planning](#) disponible en annexes. Le planning est défini en une première phase correspondant à la formation de nouvelles technologies, pour mon cas, utilisées pour le projet [InterAActionBox\\*](#). Sa deuxième phase

consiste à implémenter de nouvelles fonctionnalités, à les tester et à corriger des bugs de l'application.

Ce rapport expliquera dans un premier temps comment j'ai été initié à une technologie qui m'était encore inconnue. La formation à cette technologie était prévu pour pouvoir travailler ensuite sur [InterAActionScene\\*](#), application étant développée avec [Angular\\*](#). Il vous fera part ensuite d'une analyse de l'existant portant sur les technologies et logiciels utilisés pour mener à bien le projet [InterAActionBar\\*](#). Elle portera aussi sur l'application [InterAActionScene\\*](#) expliquant comment se présente cette dernière. Enfin, vous pourrez découvrir les différentes réalisations que j'ai pu produire sur [InterAActionScene\\*](#), comme l'implémentation de nouvelles fonctionnalités et l'écriture de tests.

## II. Initiation à Angular CLI et Karma/Jasmine

---

### II.1. Présentations

Angular CLI\* est un framework\*, soit un ensemble de composants logiciels, qu'on appellera d'ailleurs dans la suite du rapport “Angular\*\*”. D'après Wikipédia et le Journal du Net [8], on peut comparer un framework\* à un “patron”, qui établit les fondations d'un logiciel. Les frameworks\* ont pour but de faciliter le développement d'une application en fournissant une structure pour organiser et gérer le code d'un projet. Cela permet d'accélérer le processus de développement, de créer une application maintenable et de simplifier les tâches courantes des développeurs. Le framework\* Angular\* [6 et 7] est basé sur le langage de programmation TypeScript\*, un sur-ensemble de JavaScript\*. JavaScript\* est “principalement connu comme le langage de script des pages web” selon MDN Web Docs [9] et TypeScript\* existe pour pallier les lacunes de JavaScript\*, en permettant un typage des variables, la création de classes\* et d'interfaces ainsi que l'import de modules. Angular\* est un framework\* utilisé pour le développement web. Sa première version a été lancée en septembre 2016. Il est développé par Google et la communauté Angular\*. Ce framework\* cherche à fluidifier l'expérience utilisateur en permettant de créer des applications web évitant au plus les actualisations de page. Les applications développées avec Angular\* suivent l'architecture MVC\*. Cette architecture est composée de trois modules ayant leur propre responsabilité. L'un contient les données de l'application (le modèle), un autre contient les interfaces utilisateur (les vues), et le dernier gère les actions des utilisateurs et interagit avec le modèle et les vues.

Karma/Jasmine\* est un framework\* de tests utilisable avec Angular\* depuis sa création. Jasmine permet d'écrire des tests unitaires sous un format facilement lisible et compréhensible pour tout le monde. Karma, lui, est un outil orchestrant l'exécution des tests Jasmine. Il permet d'ouvrir un navigateur pour pouvoir afficher la trace des tests exécutés, mais aussi de les afficher (d'après CodeCraft [10]).

### II.2. Les Tutoriels

Cette initiation m'a été obligatoire pour pouvoir travailler sur l'application InterActionScene\*, et a donc été ma première tâche. Pour me former à Angular\*, j'ai suivi le tutoriel *Tour of Heroes* [11], disponible sur le site officiel de Angular\*. Il enseignait les bases d'Angular\*, et apprenait comment développer entièrement une application. Ainsi, le tutoriel m'a permis de créer une application permettant d'afficher une liste de héros fictifs (voir Annexe 5), sur lesquels il est possible de cliquer dessus pour obtenir leurs informations (voir Annexe 4). Les 5 premiers héros peuvent aussi être affichés dans le mode “Dashboard”, visible sur l'Annexe 3. Des fonctionnalités comme la recherche ou l'ajout de héros sont aussi disponibles comme l'on peut le voir sur l'Annexe 3 et l'Annexe 5.

Concernant Karma/Jasmine\*, j'ai lu le tutoriel disponible sur LogRocket [12], apprenant comment fonctionnent les tests unitaires de Karma/Jasmine\* avec des exemples.

*II.3. Bilan*

L'apprentissage de ces nouvelles technologies m'a été enrichissant et efficace, puisque j'ai pu ensuite comprendre aisément le code [Angular\\*](#) de l'application [InterAActionScene\\*](#), sa structure, ainsi que ses tests existants. Il s'est aussi avéré très utile, lorsque j'ai dû commencer à développer moi-même en [Angular\\*](#) de nouvelles fonctionnalités pour [InterAActionScene\\*](#).

### III. Analyse de l'existant

#### III.1. Technologies et logiciels utilisés

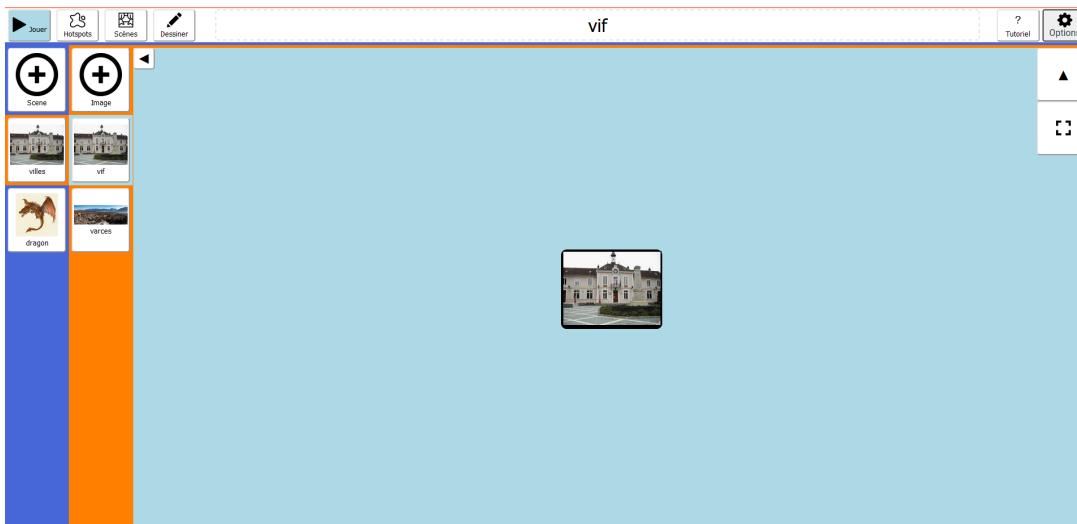
Deux outils sont utilisés pour gérer le projet [InterAActionBox\\*](#). Comme outil de communication, nous utilisons le logiciel [Discord\\*](#), déjà mis en place lors de mon arrivée. [Discord\\*](#) permet de créer des serveurs pouvant regrouper plusieurs personnes. Ces serveurs sont composés de canaux textuels et vocaux. Ainsi, nous les utilisons pour pouvoir communiquer si besoin. Comme outil de gestion de versions, l'équipe [GETALP\\*](#) utilise [GitHub\\*](#). Cet outil permet d'héberger et de gérer des projets informatiques et leurs développements. Il utilise le célèbre outil de gestion de versions [Git\\*](#). Pour pouvoir gérer le développement de [InterAActionBox\\*](#), l'équipe [GETALP\\*](#) a donc publié ses applications de [CAA\\*](#) sur cet outil, dont [InterAActionScene\\*](#).

Pour le développement des applications de [InterAActionBox\\*](#), nous utilisons l'environnement de développement intégré (IDE)\* IntelliJ, développé par JetBrains. Un IDE\* est une “application logicielle qui aide les développeurs à développer efficacement leurs codes logiciel”, d'après Amazon [13].

Concernant [InterAActionScene\\*](#), comme dit précédemment, l'application est développée en [Angular\\*](#). Pour les tests, le framework\* [Karma/Jasmine\\*](#) est utilisé.

#### III.2. L'application InterAActionScene

Comme expliqué en [Introduction](#), l'application de [CAA\\*](#) [InterAActionScene\\*](#) permet à l'utilisateur de créer des [scènes\\*](#), et les composer de plusieurs images. Elle est développée sous [Angular\\*](#) et ne comporte qu'une seule interface, comme le montre la Figure 1 ci-dessous.



**Figure 1 :** Interface principal de InterAActionScene

Sur cette interface, nous pouvons voir un grand espace bleu. C'est ici que les images sélectionnées par l'utilisateur sont affichées. Dans les coins supérieurs de cet espace, il y a trois

boutons placés. Le bouton  permet de cacher la liste des **scènes\*** et des images (voir [Annexe 7](#)). Le bouton  permet de cacher la barre des tâches (voir [Annexe 8](#)). Dans la zone latérale comportant les **scènes\*** et les images, y est placé deux boutons “Scene” et “Image”, comme on peut le voir sur l'[Annexe 7](#). Ces boutons permettent d’ajouter soit une **scène\*** soit une image. Sur l'[Annexe 8](#), nous pouvons apercevoir des boutons sur la barre des tâches. Le bouton “Options” fait apparaître les options de l’application (cf. [Annexe 9](#)). Le bouton “Tutoriel” affiche le tutoriel de l’application. À la gauche de la barre des tâches sont disposés quatre boutons. Le bouton “Dessin” permet à l’utilisateur de dessiner sur l’une de ces images grâce à un panel de dessin affiché au clique du bouton (cf. [Annexe 10](#)). Les deux boutons à droite des couleurs représentent un bouton “Gomme”, permettant à l’utilisateur de gommer ses dessins, et un bouton “Tout effacer” permettant à l’utilisateur d’effacer tout son dessin en un clique. Le bouton “Scène” permet de gérer les **scènes\*** de l’utilisateur (cf. [Annexe 11](#)). Il affiche au centre de la zone d’affichage des boutons permettant de cacher une image ou une **scène\***, de supprimer des images, d’en renommer et d’exporter ou d’importer des **scènes\*** et des images dans un **fichier JSON\***. Le bouton “Hotspots\*” permet à l’utilisateur d’exécuter des actions sur les **hotspots\*** tel qu’en ajouter, en modifier ou en supprimer (cf. [Annexe 12](#)). Lorsque l’on clique sur le bouton “Ajouter”, trois nouveaux boutons apparaissent à sa droite, permettant à l’utilisateur de choisir la forme de son **hotspot\***. Il pourra ensuite le dessiner avec la souris et commencer sa création. Lorsqu’il voudra modifier ou supprimer un **hotspot\***, il devra cliquer sur les boutons respectifs puis sur son **hotspot\*** pour exécuter l’action. Pour finir, le bouton “Jouer” permet de jouer un **hotspot\*** en cliquant dessus.

### *III.3. La construction du code*

Sous **Angular\***, l’application suit l’**architecture MVC\***. Elle est composée de plusieurs **composants\*** (cf. [Annexe 13](#)) représentant des éléments propres de l’application (Menubar pour la barre des tâches, scene-display pour la zone d’affichage plus la zone latérale, etc...). Ces **composants\*** sont chacuns constitués d’un **template\***, auquel est attaché leur fichier **CSS\***, de leur **classe\*** et de leur fichier de tests (cf. [Annexe 14](#)). Leur **classe\*** regroupe les attributs et les fonctions de l’élément, qui définissent son comportement. La structure de l’application dispose de **services\*** (cf. [Annexe 15](#)), qui regroupent les données métiers propres à l’application (les **scènes\***, les images, etc...). De cette manière, l’**architecture MVC\*** est composée des **templates\*** représentant la vue, des **classes\*** des **composants\*** représentant les contrôleurs, et des **services\*** représentant le modèle.

## IV. Réalisations

---

### IV.1. Gestion de projet

Durant le stage, les tâches à faire m'étaient communiquées par Jordan ARRIGO sur [Discord](#)\*. Les discussions et les rapports les concernant avaient lieu sur le même outil. Pour faire part de la progression de mon travail, j'annonçais le matin mon planning de la journée, et je communiquais le soir ce qui a été fait. De plus, nous faisions une visio-conférence une fois par semaine pour faire un rapport hebdomadaire.

Les premières réalisations étaient tirées des [issues](#)\* de la page [Github](#)\* de [InterAActionScene](#)\* (voir [Annexe 16](#)). Les [issues](#)\* sont des messages avertissant un problème aux développeurs leur permettant de savoir quelles modifications ou corrections doivent être effectuées sur leur projet. Après avoir réglé les [issues](#)\* nécessaires, j'ai fais une identification de fonctionnalités, pour chercher des nouveaux apports intéressants à [InterAActionScene](#)\*. J'en ai fait ensuite un rapport à Jordan pour que l'on puisse échanger et apporter une décision finale à ces idées. Ainsi, je pouvais commencer à implémenter les fonctionnalités validées par nos échanges. D'ailleurs, c'est comme ceci qu'on a pu construire mon planning sur la fiche de liaison de l'IUT (voir [Annexe 1](#)). On remarquera dans la suite du rapport que le planning initial a été accompli plus tôt que prévu. Alors, je recommençais à identifier des bugs à corriger et des modifications à apporter à l'application lorsque mes réalisations en cours étaient toutes terminées.

Lorsqu'une de ces réalisations était terminée, j'envoyais mes modifications sur [Github](#)\* par une [Pull Request](#)\*, qui représente une nouvelle version du code attendant d'être ajoutée ou non au projet initial par son administrateur.

### IV.2. Nouvelles fonctionnalités

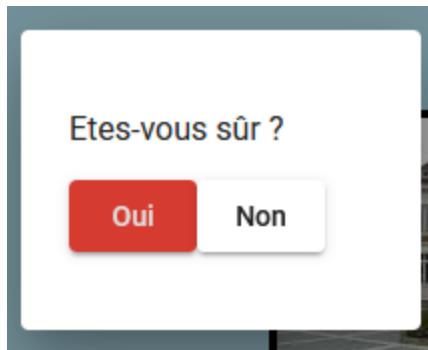
#### Supprimer toutes les [scènes](#)\* et toutes les images d'un coup

Si l'utilisateur possédait plusieurs [scènes](#)\* et plusieurs images et qu'il souhaitait tout supprimer, il devait les supprimer une par une. Cela prenait énormément de temps s'il en avait beaucoup. Pour y remédier, j'ai décidé d'ajouter un bouton "Recommencer" dans la barre des tâches comme le montre la Figure 2.



Figure 2 : Barre des tâches avec le bouton "Recommencer"

Ainsi, cliquer dessus permet d'afficher une petite fenêtre, que l'on peut voir sur la Figure 3 ci-après, demandant la confirmation de l'utilisateur.

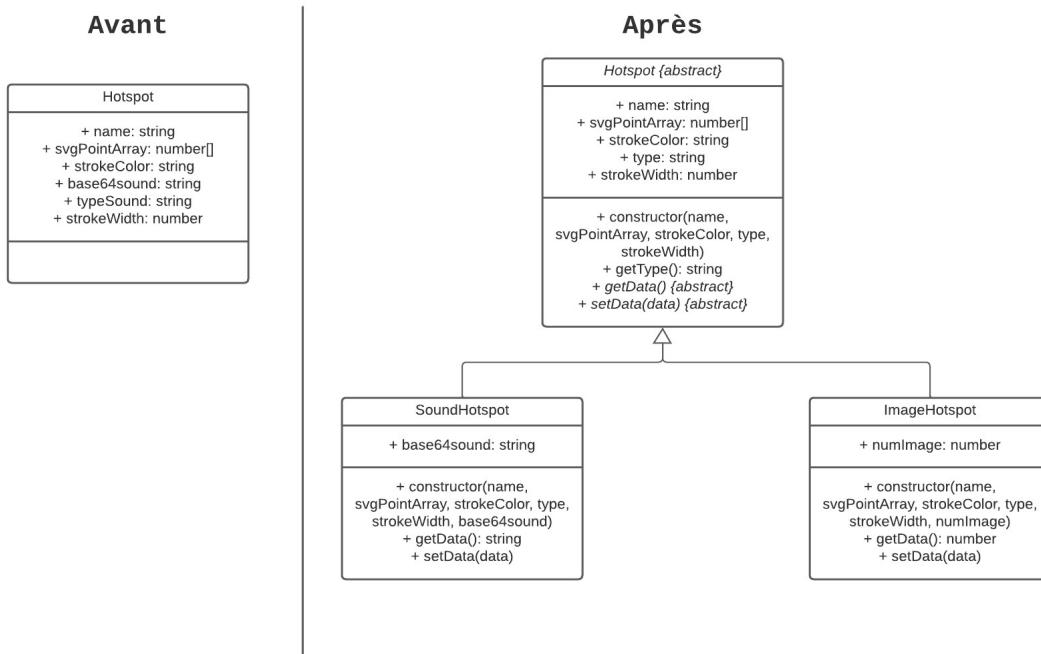


**Figure 3 :** Fenêtre de confirmation

Une fois que l'utilisateur aura cliqué sur “Oui”, toutes les [scènes\\*](#) et toutes les images se verront réinitialisées.

#### Navigation entre les images grâce aux [hotspots\\*](#)

Les [hotspots\\*](#) de [InterAActionScene\\*](#) permettent de jouer des sons lorsque l'on clique dessus. Désormais, ils permettent aussi de naviguer entre les images de l'utilisateur. En effet, la [classe\\*](#) [Hotspot\\*](#) a été revue et transformée. De base, la [classe\\*](#) [Hotspot\\*](#) était une simple [classe\\*](#) avec des attributs. Elle a donc été changée en une [classe abstraite\\*](#) pour que d'autres [classes\\*](#) puissent hériter d'elle. Ainsi, deux [classes filles\\*](#) ont été créées : [SoundHotspot\\*](#) et [ImageHotspot\\*](#). Ces deux [classes\\*](#) héritent donc des caractéristiques communes à tous les [hotspots\\*](#) et ont en plus les caractéristiques qui leur sont propres. On peut schématiser ces trois [classes\\*](#) grâce à un diagramme de [classe\\*](#) comme le montre la Figure 4 ci-dessous.



**Figure 4 :** Diagramme de classe de Hotspot avant et après modifications, créé avec Lucid [14]

Le diagramme de la Figure 4 montre que les deux [classes filles\\*](#) partagent les attributs de leur [classe mère\\*](#), à savoir name, svgPointArray, strokeColor, type et strokeWidth ainsi que son

constructeur\* et sa méthode\* getType. Néanmoins, elles ont comme caractéristiques propres leur attribut base64sound et numImage ainsi que leur constructeur\* et leurs méthodes\* getData et setData.

Lorsque l'on dessine un hotspot\* pour le créer (cf. [L'application InterAActionScene](#)), une fenêtre s'ouvre, demandant à l'utilisateur de renseigner ses caractéristiques. Suite aux modifications de la classe\* Hotspot\*, cette fenêtre devait être mise à jour. Elle doit permettre à l'utilisateur de choisir quel type de hotspot\* il souhaite créer. Comme le montre l'[Annexe 17](#), la fenêtre contient désormais deux onglets cliquables "Ajouter un son" et "Ajouter une référence" contenant la caractéristique type du hotspot\* à renseigner. Quant à sa couleur, l'épaisseur de son contour et son nom, les champs à renseigner restent identiques peu importe l'onglet sélectionné. La manière dont un hotspot est créé, une fois toutes les informations réunies, a été revue pour qu'elle puisse correspondre aux nouvelles classes Hotspot\*, SoundHotspot\* et ImageHotspot\*.

Pour mettre à jour la fonctionnalité permettant de modifier les hotspots\*, les améliorations apportées sont similaires. La fenêtre de modification de hotspot\* possède désormais les mêmes onglets que ceux de la fenêtre de création de hotspots\*. Les champs de la couleur du hotspot\*, de la taille de son contour et de son nom sont aussi identiques. Par contre, un bouton "Redessiner", fonctionnalité propre à la modification de hotspot\*, est disponible (voir [Annexe 18](#)). La fonction appelée lors du processus de modification a été changée. Désormais, elle détecte si l'utilisateur veut transformer son hotspot\* en SoundHotspot\* ou ImageHotspot\*. Si son nouveau hotspot\* est d'un type différent de l'ancien, alors la fonction détruit l'objet pour en recréer un nouveau, avec les informations données par l'utilisateur. La simple modification de cet objet n'est pas suffisante, puisqu'ils font partie de deux classes\* différentes. Or, le hotspot\* est simplement modifié seulement si l'utilisateur ne veut pas changer son type.

Pour jouer l'action d'un hotspot\*, l'utilisateur doit cliquer dessus. Ainsi, une fonction attachée à cet événement est appelée, et exécute les instructions pour jouer l'action du hotspot\*. Auparavant, elle jouait simplement le son du hotspot\*. Maintenant qu'il existe deux types de hotspots\*, la fonction vérifie d'abord le type du hotspot\* cliqué, puis joue le son ou modifie l'image affichée selon ce dernier, comme le montre ci-après la Figure 5. On remarque alors l'identification du hotspot grâce aux conditions `if (... instanceof ...)` suivies des instructions correspondantes.

```

5+ usages  ± Yansarr +2 *
PlayHotspot(hotspot: Hotspot) {
    if (hotspot instanceof SoundHotspot) {
        if (hotspot.type === "soundAudio") {
            this.audioPlayer.src = hotspot.base64sound;
            this.audioPlayer.volume = this.settingsService.VOLUME;
            this.audioPlayer.load();
            this.audioPlayer.play();
        } else if (hotspot.type === "writeAudio") {
            let speak = new SpeechSynthesisUtterance(hotspot.base64sound);
            speak.lang = this.languageService.activeSpeechSpeakerLanguage;
            speak.volume = this.settingsService.VOLUME;
            window.speechSynthesis.speak(speak);
        }
    } else if (hotspot instanceof ImageHotspot) {
        this.sceneDisplayService.selectedImage = hotspot.numImage;
        this.scenesService.updateScenes();
        this.sceneDisplayService.UpdateDimensions();
    }
}

```

Figure 5 : Fonction PlayHotspot, qui exécute l'action du hotspot lors de son clique

[InterAActionScene](#)\* utilise l'[Indexed DB](#)\*, une base de données du navigateur utilisable par les applications web. À chaque fois que l'utilisateur modifie ses [scènes](#)\* et ses images, elles sont enregistrées dans cette base de données. Cependant, lorsque l'on enregistre ces objets dans l'[indexed DB](#)\*, ils se retrouvent “dénaturés”. Plus précisément, les [scènes](#)\*, les images et les [hotspots](#)\* sont enregistrés comme de simples [Objects](#)\* (type de donnée [JavaScript](#)\* permettant de stocker un ensemble de clé/valeur, selon MDN Web Docs [15]). Ils ne sont donc plus considérés comme une [Scene](#)\*, une Image ou un [Hotspot](#)\*, qui sont des classes de [InterAActionScene](#)\*. Au démarrage de l'application, celle-ci charge les objets enregistrés comme des [Object](#)\*, et non pas comme une [Scene](#)\*, une Image ou un [Hotspot](#)\*. Lorsque l'utilisateur voulait alors cliquer sur un [hotspot](#)\*, il ne se passait rien, puisque la fonction [PlayHotspot\(\)](#) ne reconnaissait pas l'objet comme un [SoundHotspot](#)\* ou un [ImageHotspot](#)\*. De ce fait, la fonction chargeant les objets au démarrage a été mise à jour, et [instancie](#)\* désormais les objets à partir de leur [classe](#)\* respective.

#### Ajout et modification du panel de dessin

Le panel de dessin a été modifié. Comme on le voit sur l'[Annexe 10](#), le panel proposait 6 couleurs à l'utilisateur, et lui permettait de gommer ou d'effacer entièrement ses traits. À présent, l'utilisateur peut aussi choisir la taille de son crayon et de sa gomme, et peut sélectionner 4 nouvelles couleurs : le marron, le jaune, le violet et le rose. (voir Figure 6 ci-après)



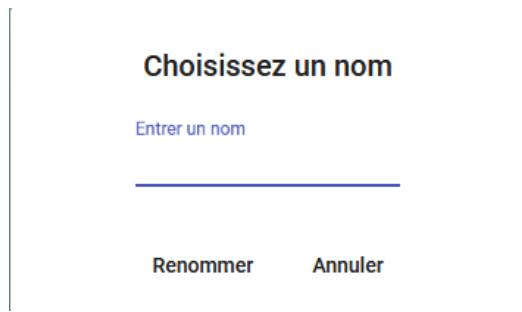
**Figure 6** : Nouveau panel de dessin de InterAActionScene

Les 4 couleurs ont été simplement ajoutées dans la liste des couleurs, tandis que les boutons de taille ont été groupés à part, sur la gauche des couleurs. Cette manière de choisir la taille de son crayon et de sa gomme a été décidée selon des questions d'accessibilité et de facilité d'utilisation, en lien avec le public cible de [InterAActionBox\\*](#). C'est pour cela que l'on choisit la taille du crayon et de la gomme grâce à des gros boutons, facilement compréhensible par leur visuel.

Cliquer sur une couleur appelle une fonction, avec comme paramètre le nom de la couleur, qui change celle du crayon en conséquence. Il a seulement été nécessaire de modifier la fonction de sorte qu'elle puisse reconnaître les 4 nouvelles couleurs, et ainsi modifier celle du crayon. En parallèle, cliquer sur un bouton de taille sauvegarde celle-ci, et l'applique à la taille du trait du crayon ou de la gomme juste avant son tracé.

#### Pouvoir renommer les [scènes\\*](#) et les images indépendamment

Jusque là, l'utilisateur n'avait pas la possibilité de renommer ses [scènes\\*](#). Désormais, il le peut. Pour rappel, l'utilisateur doit cliquer sur le bouton “Scène” de la barre des tâches, et ensuite cliquer sur le bouton “Renommer” qui s'affiche au centre de l'interface (cf. [Annexe 11](#)). Ainsi, une petite fenêtre s'affiche et permet à l'utilisateur de renommer l'image qui est sélectionnée. Pour qu'il puisse renommer sa [scène\\*](#) sélectionnée, un nouveau champ a été ajouté à cette intention dans cette fenêtre. Les Figures 7 et 8 ci-dessous le montrent :



**Figure 7** : Fenêtre de renommage avant modifications



**Figure 8** : Fenêtre de renommage après modifications

L'utilisateur a donc le choix soit de renommer sa [scène\\*](#), soit de renommer son image, soit les deux.

#### Pouvoir organiser nos [scènes\\*](#) et les images librement

Si l'utilisateur importe une petite image, elle sera affichée tel quel dans l'application, ce qui peut rendre sa visibilité difficile. Ainsi, une fonctionnalité de zoom et de dézoom ont été implémentées. Deux boutons représentatifs ont été ajoutés au-dessous du bouton permettant de cacher la barre des tâches et celui permettant d'afficher l'application en plein écran, à droite de l'interface, comme le montre la Figure 9 ci-après.



**Figure 9** : Boutons zoom et dézoom

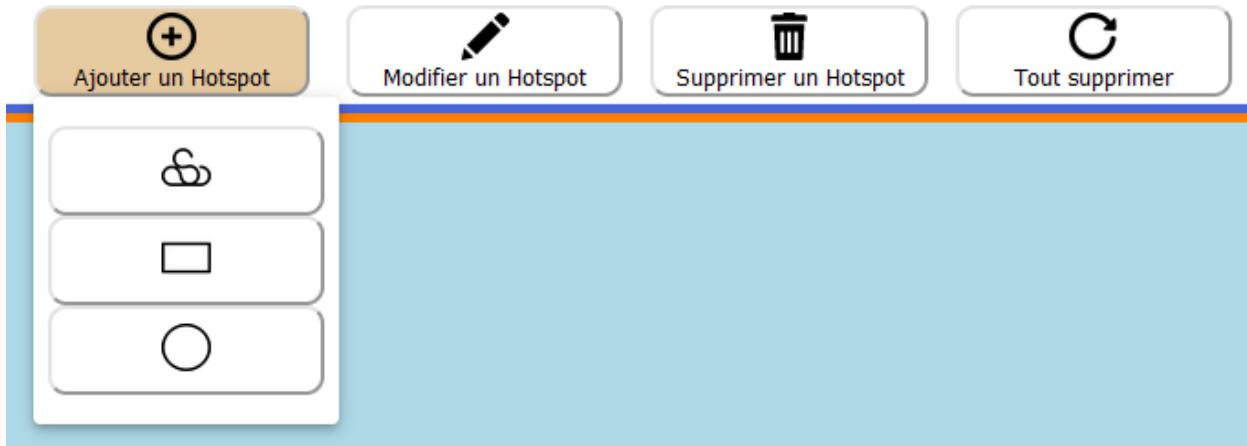
Cliquer sur ces boutons appelle une fonction qui agrandit ou diminue les dimensions de l'image sélectionnée, permettant à l'utilisateur d'améliorer la visibilité des images les plus petites.

De plus, l'utilisateur a maintenant la possibilité d'organiser l'ordre de ses images et de ses [scènes\\*](#) dans leur liste (cf. [Annexe 7](#)) comme il le souhaite. Pour ce faire, il lui faut seulement procéder à un glisser-déposer et déplacer l'objet à l'emplacement où il le souhaite. Une fonction a été ajoutée à cet événement pour que l'ordre de la liste soit modifié en conséquence.

À ce niveau-là, toutes les fonctionnalités envisagées dans le planning ont été terminées plus vite que prévu (cf. [Annexe 1](#)). Alors, lorsque toutes les réalisations étaient terminées, j'analysais [InterAActionScene\\*](#) pour trouver des bugs à corriger et des fonctionnalités à améliorer, modifier ou rajouter. C'est ainsi que j'ai pu implémenter les suivantes.

#### Suppression de tous les hotspots d'une image

Pour permettre à l'utilisateur de supprimer tous les [hotspots\\*](#) d'une de ses images, j'ai décidé de procéder de la même manière que pour [supprimer toutes les scènes en un coup](#). J'ai donc ajouté un bouton "Tout supprimer" dans la liste des actions des [hotspots\\*](#) (cf. [Annexe 12](#)). Cependant, j'ai remarqué à son insertion qu'il n'y avait plus assez de place, en utilisant des petits écrans, pour ajouter ce bouton, car les éléments de la barre de tâches se chevauchaient. Cette dernière devait alors être modifiée pour pouvoir faire de la place à ce bouton. Jusqu'à maintenant, lorsque l'utilisateur cliquait sur "Ajouter un [Hotspot\\*](#)", les formes sélectionnables apparaissaient à droite de ce bouton. Désormais, pour libérer de la place, ce bouton est devenu un bouton déroulant. Ainsi, lorsque l'on clique dessus, les formes sélectionnables n'apparaissent pas à droite du bouton, mais au-dessous du bouton. À présent, le bouton "Tout supprimer" a la place nécessaire pour s'ajouter à la liste, comme le montre la Figure 10 ci-après.



**Figure 10 :** Les actions de [hotspots\\*](#) après modification

Quand l'utilisateur clique sur le bouton “Tout supprimer”, une fenêtre de confirmation identique à la [Figure 3](#) apparaît pour lui demander s'il est sûr de son choix. S'il valide, alors tous les [hotspots\\*](#) de son image disparaissent.

#### Remplir automatique le nom à la création d'un objet

Lorsque l'utilisateur veut créer un objet (scène, image, [SoundHotspot\\*](#), ...), les formulaires de création demandent un nom et souvent un fichier sélectionné pour celui-ci. Grâce à cette nouvelle fonctionnalité, dès que l'utilisateur a choisi son fichier et qu'il n'a toujours pas rempli l'entrée du nom, celle-ci se remplit alors automatiquement par le nom du fichier sélectionné.

#### Mise à jour du tutoriel

Il existe un site web de l'application [InterAActionScene\\*](#) [16] sur lequel est disponible, en français et en anglais, sa présentation ainsi que les caractéristiques propres au projet (Contact, Références, Partenaires, Développeurs, etc...). Il met aussi à disposition un mode d'emploi pour soutenir l'utilisateur lors de l'utilisation de l'application. Par ailleurs, ce mode d'emploi, autrement dit ce tutoriel, est le même que celui disponible dans l'application. Suite à toutes les modifications qui ont été apportées sur celle-ci, le tutoriel avait besoin d'une grande mise à jour.

Le site web est déployé grâce au service d'hébergement [Github Pages\\*](#) [17], appartenant à [Github\\*](#). Il permet de prélever des fichiers choisis par les développeurs directement dans un de leurs dépôts [Github\\*](#), afin de les générer comme site web pour ensuite le publier. Concernant la page du tutoriel, elle est développée essentiellement en [Markdown\\*](#), qui est un langage de balisage permettant de formater du texte brut, selon Futura [18].

Pour mettre à jour le tutoriel français et anglais, il a fallu mettre à jour les captures d'écran et les textes aidant l'utilisateur pour qu'ils puissent correspondre à la nouvelle application et expliquer les nouvelles fonctionnalités.

#### IV.3. Tests

##### Supprimer toutes les **scènes\*** et toutes les images d'un coup

Pour tester la suppression de toutes les **scènes\***, il a fallu écrire un test vérifiant simplement que la liste des **scènes\*** soit vide après l'appel de la fonction.

##### Navigation entre les images grâce aux **hotspots\***

Des nouveaux tests ont été écrits pour vérifier que la création de chaque type de **hotspot\*** soit bien fonctionnelle, ainsi que leur modification. Il a aussi fallu tester la fonction **PlayHotspot()** pour s'assurer qu'elle exécute correctement l'action du **hotspot\***, c'est à dire que l'image sélectionnée soit bien modifiée par celle du **hotspot\*** dans le cas d'un **ImageHotspot\***, ou que le son du **hotspot\*** soit bien joué dans le cas d'un **SoundHotspot\***.

##### Ajout et modification du panel de dessin

Après ces ajouts et ces modifications, des tests ont été écrits pour vérifier que la couleur du stylo soit bien modifiée lorsque l'on clique sur les nouvelles couleurs. D'autres ont aussi été écrits pour tester le bon changement de la taille du crayon et de la gomme quand on clique sur les boutons correspondants.

##### Pouvoir renommer les **scènes\*** et les images indépendamment

Des tests ont déjà été écrits pour vérifier que le renommage des images fonctionnait correctement. De petites modifications ont été apportées pour qu'ils puissent vérifier aussi le bon renommage de la **scène\*** sélectionnée.

##### Pouvoir organiser nos **scènes\*** et nos images librement

Concernant le zoom et le dézoom des images, des tests ont été écrits pour vérifier que les dimensions des images soit modifiées au clique des boutons "Zoom" et "Dézoom". Pour ce qui est du glisser-déposer, des tests sont disponibles pour tester qu'il soit bien fonctionnel pour les **scènes\*** et pour les images, et que les listes soient modifiées correctement après l'action.

##### Suppression de tous les hotspots d'une image

Pour vérifier le bon fonctionnement de la suppression, un simple test a permis de vérifier qu'une fois la suppression faite, la liste des **hotspots\*** de l'image soit vide.

##### Remplir automatiquement le nom à la création d'un objet

Tester cette fonctionnalité revient à vérifier que lorsqu'un fichier est sélectionné à la création d'un **hotspot\***, d'une image et d'une **scène\***, le nom de l'objet doit se remplir automatiquement par le nom du fichier, s'il n'est pas encore renseigné. Pour ce faire, il faut simuler un fichier et le passer en paramètre à la fonction concernée. Cependant, le fichier doit être lu correctement pour que la fonction agisse comme il se doit. Or, à ce jour, lors de l'exécution des tests, la lecture est incorrecte et la fonction ne peut pas agir correctement. Après analyse du résultat, cela doit être causé par une mauvaise simulation du fichier.

Voici au final toutes les réalisations que j'ai pu produire durant ce stage. En effet, le planning prévisionnel était plus léger que prévu. Alors, un planning effectif montrant l'organisation de ces réalisations sur la période de stage est disponible en [Annexe](#).

## V. Conclusion

---

Pour conclure, le stage m'a permis de me former au [framework\\* Angular\\*](#) et aux tests [Karma/Jasmine\\*](#). J'ai dû analyser une application et la comprendre pour ensuite continuer son développement, ce qui m'a permis de consolider cette compétence et de revoir ma méthode de travail. Concernant son développement, plusieurs fonctionnalités ont été ajoutées ou améliorées. Des tests ont aussi été retouchés suite à des modifications et d'autres ont été créés pour tester les nouvelles fonctionnalités. L'ensemble de ces missions a fortifié mes compétences en développement puisque j'ai appris à rechercher des nouvelles fonctionnalités cohérentes à l'application, à réfléchir comment les implémenter, à mieux détecter les erreurs et à comment les corriger efficacement. Ce stage fut fort enrichissant, et ses missions étaient très intéressantes à accomplir. Le développement Web est aujourd'hui un développement que j'apprécie particulièrement, et j'aimerais m'orienter professionnellement vers ce domaine là. L'expérience humaine au sein du LIG fut tout aussi intéressante, l'ambiance et les relations avec l'équipe [GETALP\\*](#) étaient fort agréables, et cela me motive à m'insérer dans le monde de l'entreprise.

## Annexes

---

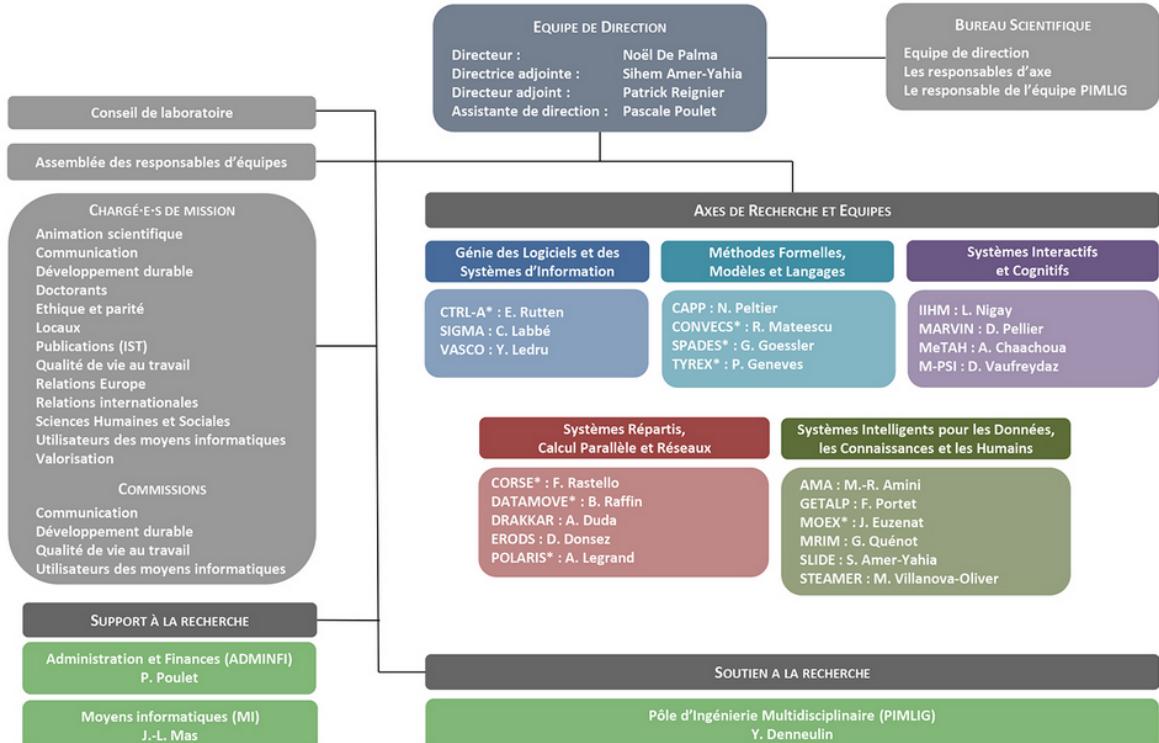
### Annexe 1 - Planning du stage prévu

#### Planning de travail prévu

(détaillez par période de 1 à 2 semaine(s) en précisant les dates)

Période	Travail à réaliser
17/04 au 21/04	Initiation à Angular et aux tests Karma + 1 jour pour la construction du rapport préliminaire
24/04 au 28/04	Correction issues + tests + 1 jour pour la construction du rapport préliminaire
01/05 au 12/05	Nouvelles fonctionnalités + tests dans <u>InteraactionScene</u> + 2 jours pour la construction du rapport
15/05 au 26/05	Nouvelles fonctionnalités + tests dans <u>InteraactionScene</u> + 2 jours pour la construction du rapport
29/05 au 09/06	Nouvelles fonctionnalités + tests dans <u>InteraactionScene</u> + 2 jours pour la construction du rapport
12/06 au 23/06	Nouvelles fonctionnalités + tests dans <u>InteraactionScene</u> + 2 jours de préparation de la soutenance
26/06 au 07/07	Nouvelles fonctionnalités + tests dans <u>InteraactionScene</u>
	Nouvelles fonctionnalités = Navigation entre les images grâce à des "hotspots"
	Ajout et modification du panel de dessin Pouvoir renommer les scènes et les images indépendamment
	Pouvoir ordonner nos scènes et nos images librement Révision de nos Pull Request jusqu'à validation (Merge)
<b>12 juin 2023 au plus tard : envoi du mémoire de stage</b>	
<b>Semaine du 19 juin 2023 : soutenances de stages</b>	

## Annexe 2 - Organigramme du LIG



\* Equipe projet commune INRIA

UMR5217 – www.liglab.fr – janvier 2021

## Annexe 3 - Tour of Heroes : H  ros affich  s dans le mode “Dashboard”

**Tour of Heroes**

Dashboard    Heroes

Top Heroes

Bombasto    Celeritas    Magneta    RubberMan

Hero Search

## Annexe 4 - Tour of Heroes : Informations d'un h  ro

**Tour of Heroes**

Dashboard    Heroes

## DR. NICE Details

id: 12

Hero name: Dr. Nice

save

go back

Annexe 5 - Tour of Heroes : Héros affichés dans le mode “Heroes”

## Tour of Heroes

Dashboard    Heroes

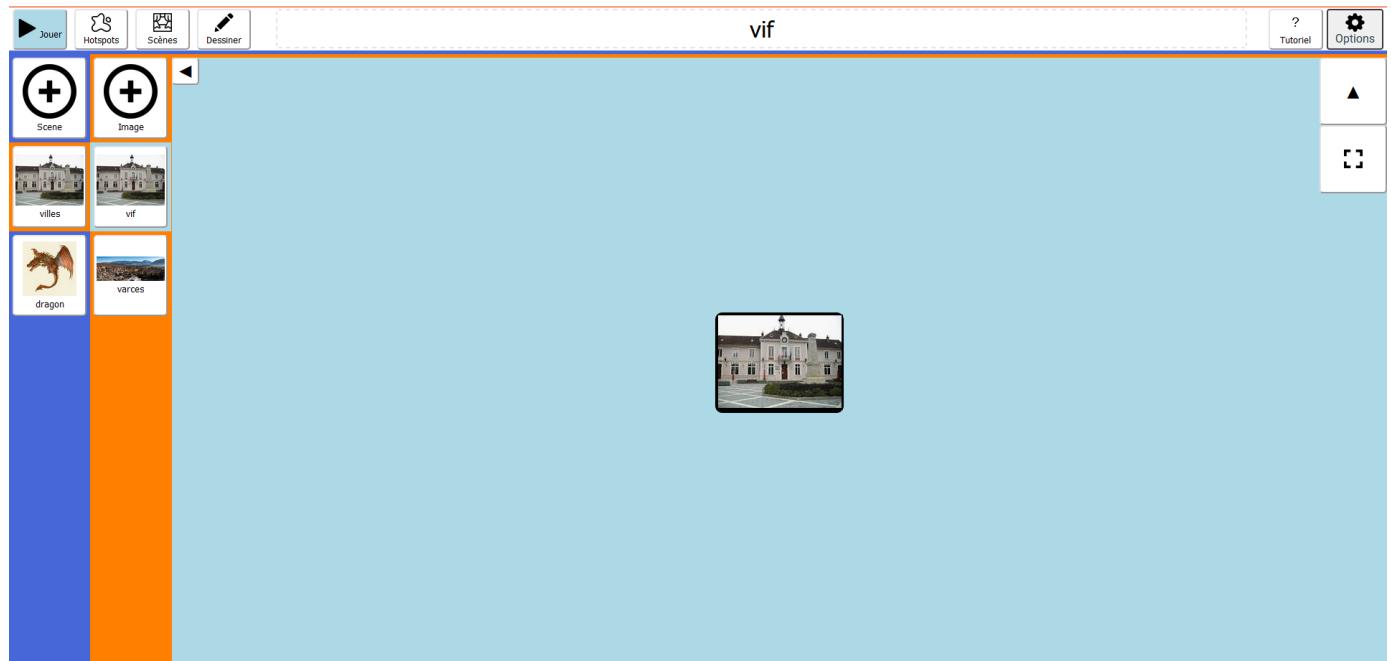
### Heroes List

12	Dr. Nice	X
13	Bombasto	X
14	Celeritas	X
15	Magneta	X
16	RubberMan	X
17	Dynama	X
18	Dr. IQ	X
19	Magma	X
20	Tornado	X

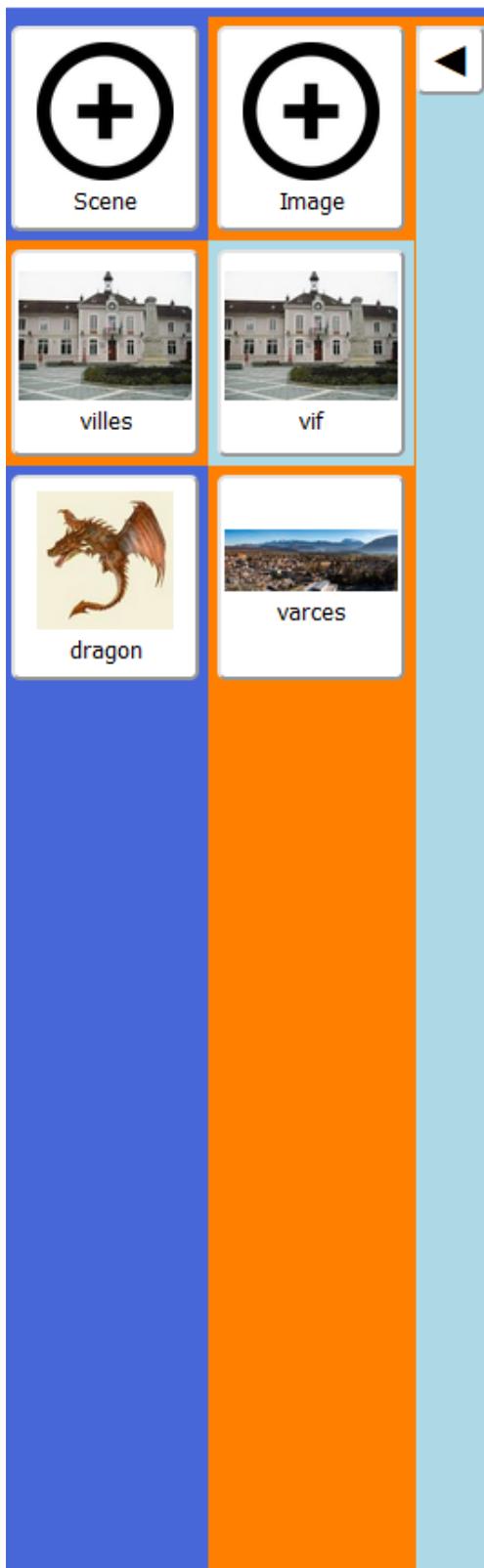
Hero name:

Add hero

Annexe 6 - InterAActionScene : Interface



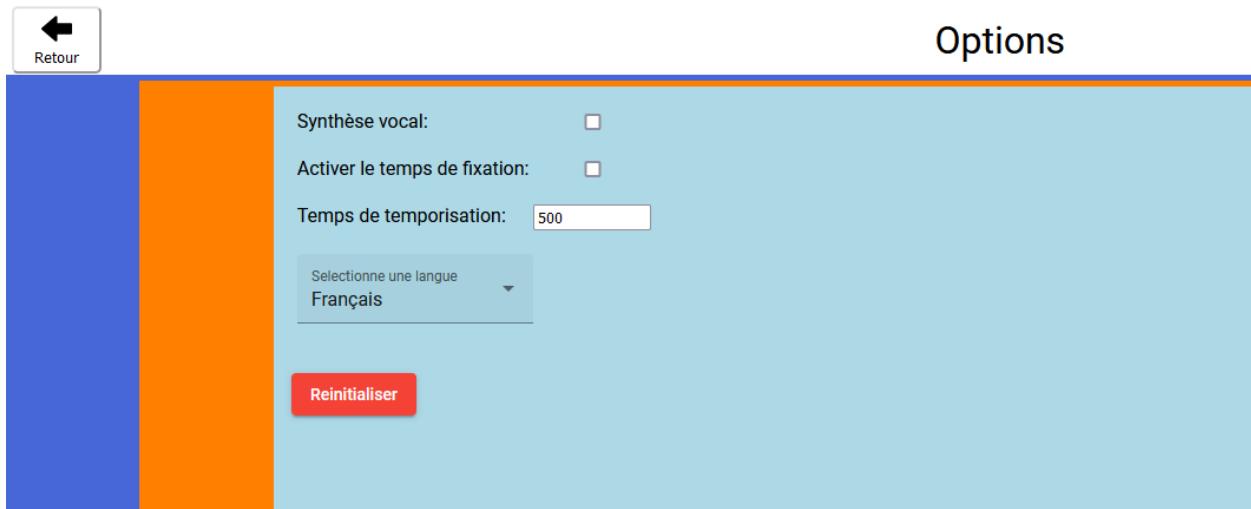
## Annexe 7 - InterAActionScene : Liste des scènes et des images



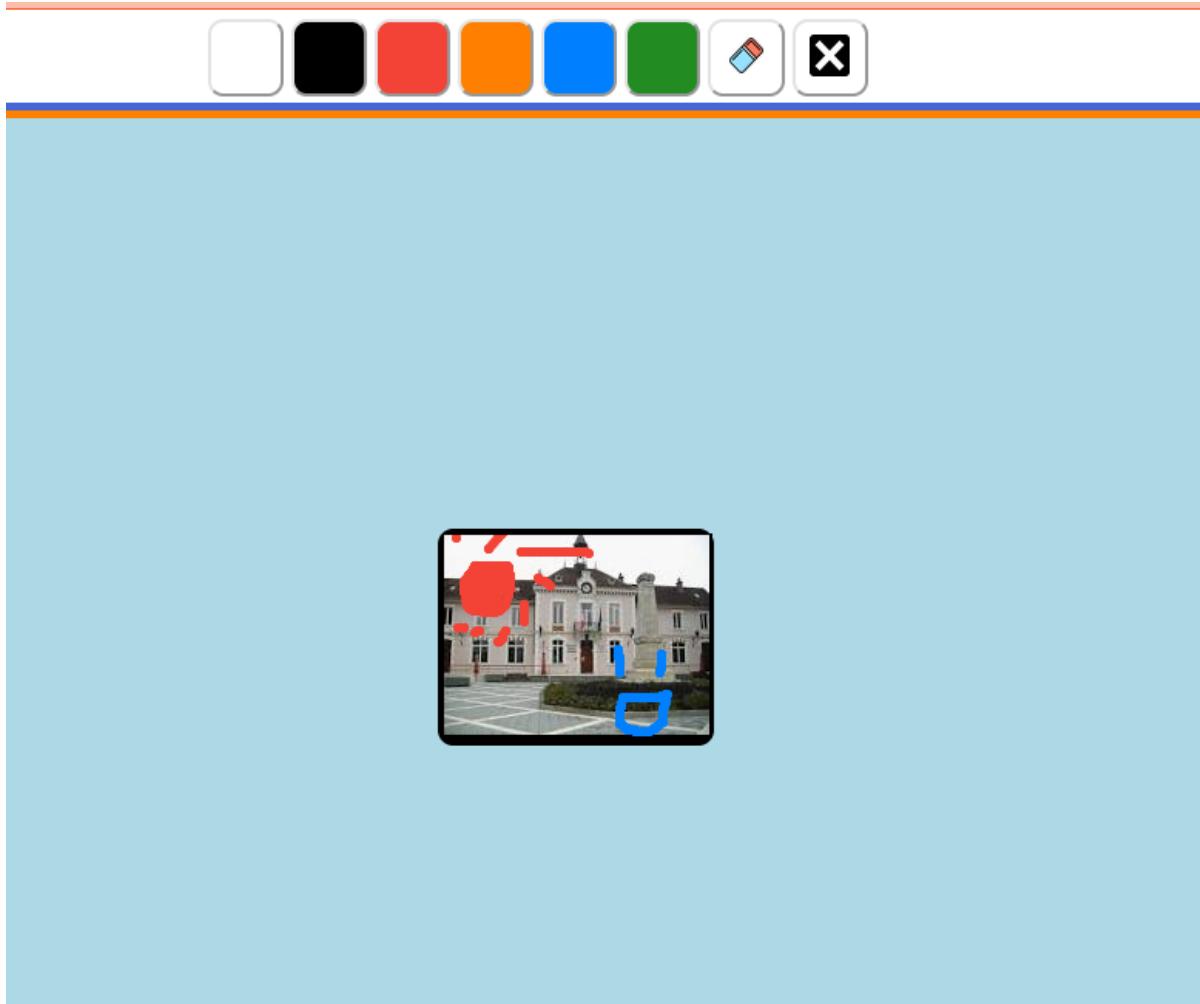
## Annexe 8 - InterAActionScene : Barre de tâches



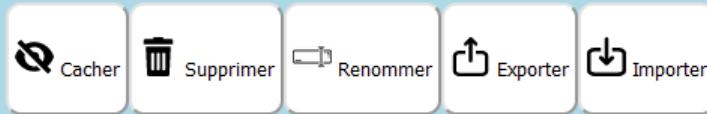
## Annexe 9 - InterAActionScene : Options



## Annexe 10 - InterAActionScene - Panel de dessin et Dessiner sur une image



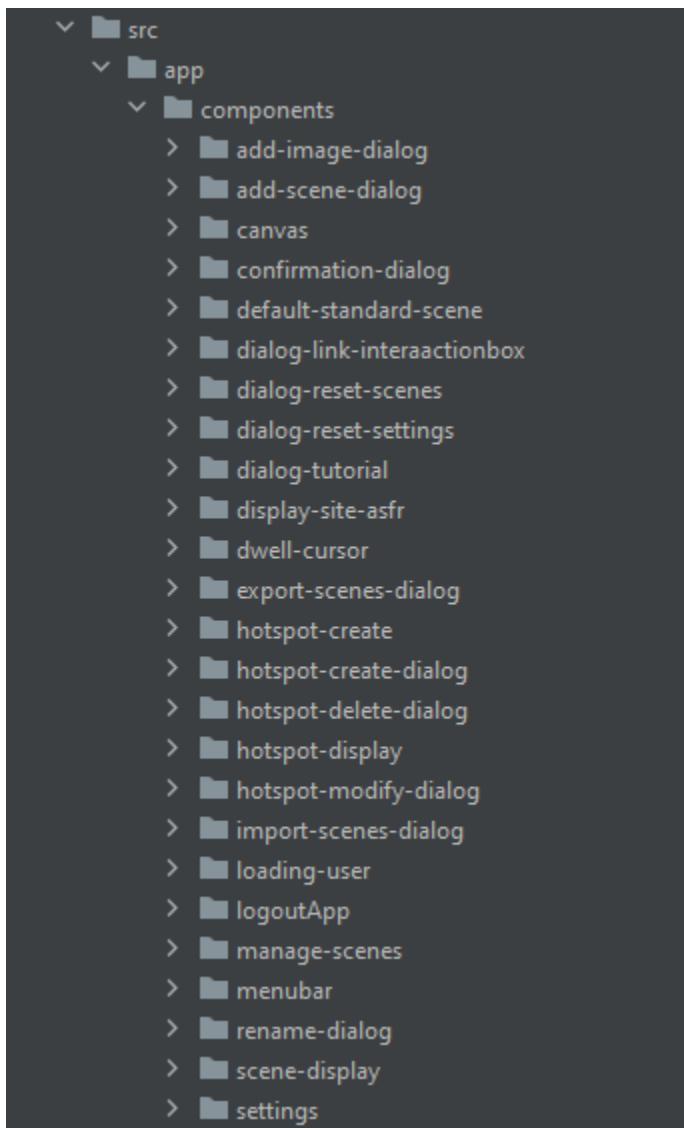
## Annexe 11 - InterAActionScene - La gestion des scènes



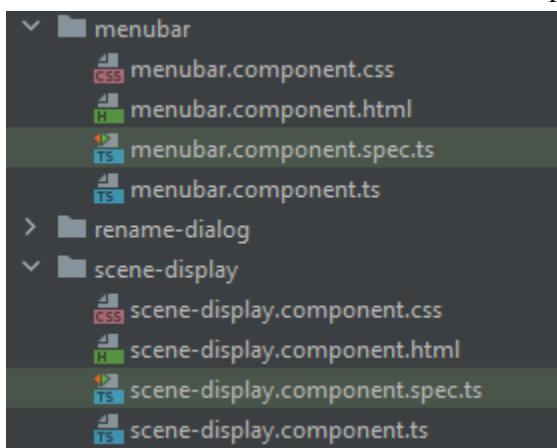
## Annexe 12 - InterAActionScene - Actions sur les Hotspots. Le cercle bleu en est un.



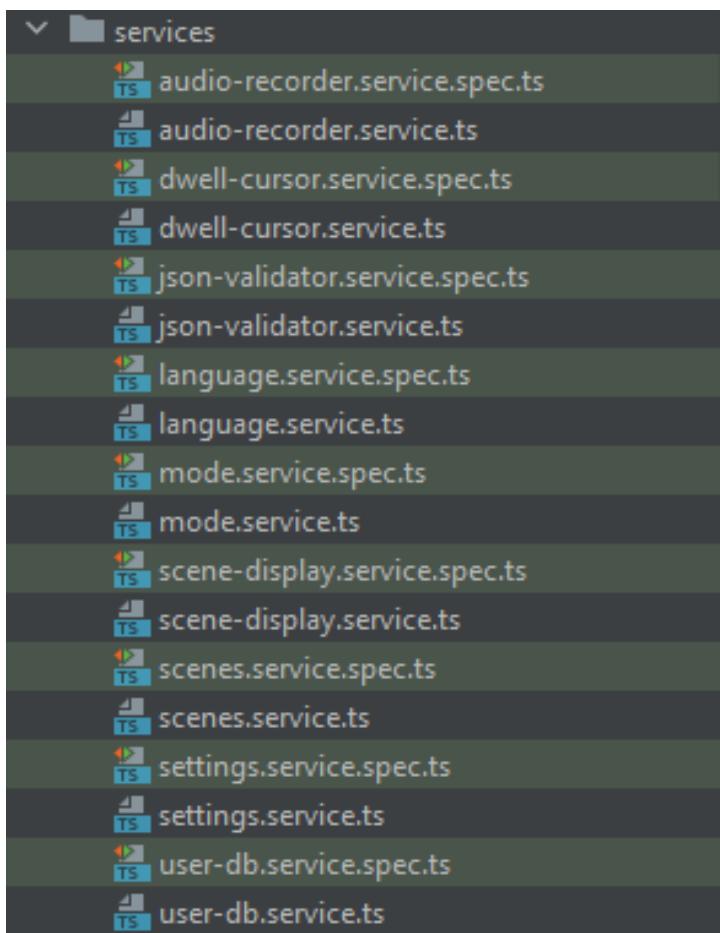
## Annexe 13 - InterAActionScene - Les composants, vu avec IntelliJ



## Annexe 14 - InterAActionScene : Les composants Menubar et Scene-Display



## Annexe 15 - InterAAActionScene : Les services



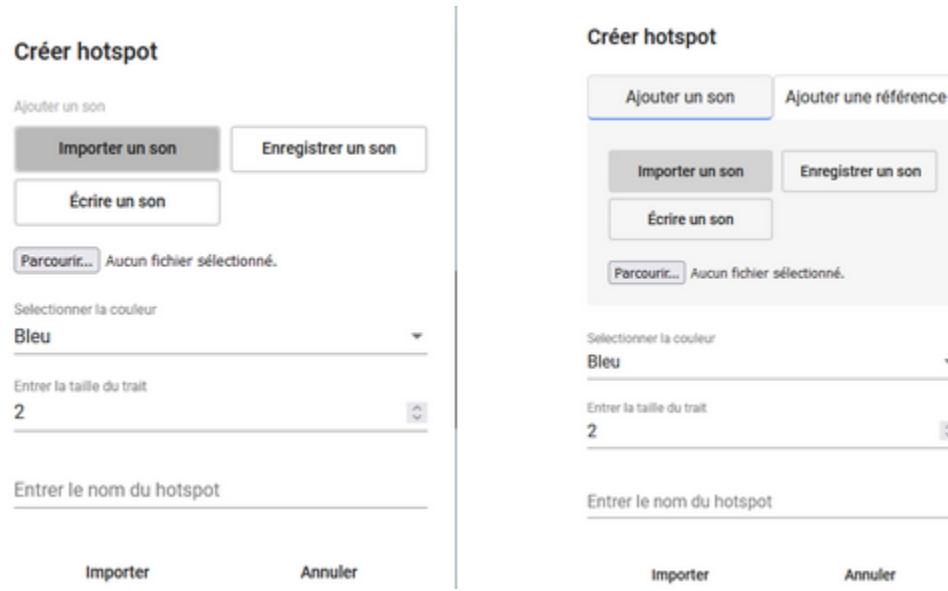
## Annexe 16 - Github : Les issues de InterAAActionScene

The screenshot shows the GitHub Issues page for the repository 'InteraactionGroup / InterAAActionScene'. The page has a light gray background. At the top, there are navigation links: Code, Issues (6), Pull requests (4), Actions, Projects, Security, and Insights. Below these are search and filter fields. The main area displays a list of 6 open issues. Each issue is represented by a card with a blue circular icon, the issue title, and a link to the issue page. The issues are:

- Name of Scene not reset when clean (#146)
- Reset scene and image (#142)
- Upgrade ES and typescript version (#113)
- Set angular for ios or android (#95)
- [PDF User Guide] (#47)
- hotspot and paint on touch screen (#33)

At the bottom of the list, there is a note: "ProTip! Find all open issues with in progress development work with linkedpr." The footer contains standard GitHub links: © 2023 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

### Annexe 17 - InterAACTIONScene : Fenêtre de création de hotspots avant et après



### Annexe 18 - InterAACTIONScene : Fenêtre de modification de hotspots

#### Modifier l'hotspot



## Annexe 19 - L'oculomètre Tobii Eye Tracker 5



## Annexe 20 - Planning effectif du stage

**Planning de travail effectif**

Période	Travail réalisé
17/04 au 21/04	Initiation à Angular et aux tests Karma / Correction Issues et tests + 1 jour pour la construction du rapport
24/04 au 28/04	Nouvelle fonctionnalité : naviguer entre les images grâce à des hotspots + 1 jour pour la construction du rapport
01/05 au 12/05	Tests nouvelle fonctionnalité Modifier panel de dessin + tests Améliorer le renommage d'images, de scènes + tests 2 jours pour la construction du rapport
15/05 au 26/05	Ordonner les scènes et images librement + tests Identification nouvelles fonctionnalités Supprimer tous les hotspots d'une image + tests Mise à jour du tutoriel Remplissage automatique du nom 2 jours pour la construction du rapport
29/05 au 09/06	Tests du remplissage automatique du nom Corrections de bugs Construction du rapport

## Glossaire

---

- Angular CLI : Framework basé en TypeScript, créé en septembre 2016. Il est développé par Google et la communauté Angular. Il suit l'architecture MVC et cherche à fluidifier l'expérience utilisateur en permettant de créer des applications web sans actualisation de page.
- Architecture MVC : Modèle de conception de logiciels composé de trois modules : modèle, vue et contrôleur. Ils ont chacun leurs responsabilités, l'un contient les données de la logique métier, un autre contient les interfaces utilisateur, et le dernier gère les actions des utilisateurs et interagit avec le modèle et les vues.
- Classe : En informatique, une classe représente une catégorie d'objet, ayant ses propres attributs et fonctions, qui définissent son comportement.
- Classe abstraite : Classe ne pouvant être instanciée. Elle est généralement une classe mère, permettant à ses classes filles d'hériter de ses attributs et de ses méthodes.
- Classe fille : Classe héritant des caractéristiques de sa classe mère.
- Classe mère : Classe qui offre ses caractéristiques à ses classes filles.
- Communication Alternative et Augmentée (CAA) : La CAA représente l'ensemble des techniques et outils permettant aux personnes ayant des difficultés pour communiquer de s'exprimer ou de comprendre les autres.
- Composants : En Angular, représentent des éléments propres de l'application. Ils sont chacuns composés de leur template, leur fichier CSS, leur classe et leur fichier de test.
- Constructeur : Méthode d'une classe permettant son instantiation.
- CSS (Cascading Style Sheet) : Langage permettant de mettre en forme et de styliser des pages web, dont des templates.
- Discord: Outil de communication permettant la création de serveurs à thèmes, regroupant plusieurs personnes. Ces serveurs sont composés de canaux textuels et vocaux, permettant à leurs membres de communiquer.
- Environnement de développement intégré (IDE) : application logicielle qui aide les programmeurs à développer efficacement le code logiciel.
- Fichier JSON (JavaScript Object Notation) : Fichier regroupant des données représentées sous un format textuel reprenant la notation des objets du langage JavaScript.
- Framework : Ensemble de composants logiciels, comparable à un "patron". Il établit les fondations d'un logiciel. Les frameworks ont pour but de faciliter le développement d'une application en fournissant une structure pour organiser et gérer le code d'un projet. Cela permet d'accélérer le processus de développement, de créer une application maintenable et de simplifier les tâches courantes des développeurs.
- Git : Célèbre logiciel de gestion de versions, créé en 2005 par Linus Torvalds.
- Github : Application web permettant d'héberger et de gérer des projets informatiques, aussi bien privés que publics, ainsi que leur développement. Pour cela, il utilise le logiciel de gestion de versions Git.
- Github Pages : Service d'hébergement appartenant à Github permettant de prélever des fichiers directement dans un dépôt Github choisis par ses développeurs, afin de les générer et d'en publier le site web.
- Groupe d'Étude en Traduction Automatique et en traitement Automatisé des Langues et de la Parole (GETALP) : Équipe menant des recherches "dont l'objectif est d'aborder

tous les aspects théoriques, méthodologiques et pratiques de la communication et du traitement de l'information multilingue (écrite ou orale).” d'après sa présentation [3].

- Hotspots : Objets de l'application InterAActionScene. Ce sont des zones que l'on peut placer sur les images de l'utilisateur et qui produisent des sons choisis en amont par ce dernier lorsque l'on clique dessus.
- ImageHotspot : classe de InterAActionScene représentant les hotspots qui affichent une autre image lorsque l'on clique dessus.
- Indexed DB : Base de données du navigateur web utilisable par les applications web.
- Instancier : Créer un objet à partir d'une classe.
- InterAActionBox : Projet de l'équipe GETALP visant à favoriser l'apprentissage et la communication des personnes en situation de handicap cognitif, du plus petit au plus grand âge. Plus particulièrement, il intègre un ordinateur ou une tablette permettant des interactions oculaires et tactiles, un système d'exploitation qui gère InterAActionBox et plusieurs applications de communication alternative et augmentée.
- InterAActionScene : Application de CAA du projet InterAActionBox permettant à l'utilisateur de créer des scènes, de leurs attribuer des images et d'y placer des dessins ou des “hotspots”.
- Issues : Sur Github, les issues sont des messages postés sur des projets avertissant un problème aux développeurs, leurs permettant de savoir quelles modifications ou corrections doivent être effectuées pour améliorer leurs projets.
- JavaScript : Célèbre langage de programmation utilisé pour le développement web.
- Karma/Jasmine : Framework de tests utilisable avec Angular depuis sa création. Jasmine permet d'écrire des tests unitaires sous un format facilement lisible et compréhensible pour tout le monde. Karma, lui, est un outil orchestrant l'exécution des tests Jasmine. Il permet d'ouvrir un navigateur pour pouvoir afficher la trace des tests exécutés, mais aussi de les afficher.
- Markdown : Langage de balisage permettant de formater du texte brut.
- Méthode : Fonction d'une classe qui lui est propre.
- Object : type de donnée JavaScript permettant de stocker un ensemble de clé/valeur
- Oculomètre : Outil permettant de mesurer le mouvement des yeux
- Pull Request : Sur Github, représente une nouvelle version du code apportée par un développeur attendant d'être ajoutée ou non au projet initial par l'administrateur de ce projet.
- Scènes : Dans InterAActioScene, représentent des objets comportant plusieurs images.
- Services : En Angular, classe regroupant les données métiers de l'application. Ils permettent ainsi de séparer les responsabilités et de simplifier la communication entre les autres composants.
- SoundHotspot : classe de InterAActionScene représentant les hotspots qui produisent un son lorsque l'on clique dessus.
- Template : En Angular, nom attribué aux interfaces utilisateurs des composants.
- TypeScript : Sur-ensemble du langage JavaScript, existant pour pallier les lacunes de JavaScript, en permettant un typage des variables, la création de classes et d'interfaces et l'import de modules

## Sitographie

---

- [1] Présentation du LIG. Disponible sur <https://www.liglab.fr/fr/presentation> (consultée le 28/04/2023) et sur son livret de 2021  
[https://www.liglab.fr/sites/lig/files/Mediatheque/livret\\_small-2021\\_compressed.pdf](https://www.liglab.fr/sites/lig/files/Mediatheque/livret_small-2021_compressed.pdf) (consultée le 28/04/2023)
- [2] Page Wikipédia du LIG  
[https://fr.wikipedia.org/wiki/Laboratoire\\_d'informatique\\_de\\_Grenoble](https://fr.wikipedia.org/wiki/Laboratoire_d'informatique_de_Grenoble) (consultée le 28/04/2023)
- [3] Présentation de l'équipe GETALP  
<https://www.liglab.fr/fr/recherche/equipes-recherche/getalp> (consultée le 28/04/2023)
- [4] Définition de la communication alternative et augmentée selon EpsilonMelia  
<https://www.epsilonmelia.com/communication-alternative-amelioree/> (consultée le 28/04/2023)
- [5] Organigramme du LIG. Disponible sur <https://www.liglab.fr/fr/presentation/organigramme> (consulté le 05/05/2023)
- [6] Présentation d'Angular - Site officiel de Angular <https://angular.io/> (consulté le 05/05/2023)
- [7] Page Wikipédia d'Angular <https://fr.wikipedia.org/wiki/Angular> (consulté le 05/05/2023)
- [8] Définition d'un framework - Page Wikipédia "Framework"  
<https://fr.wikipedia.org/wiki/Framework> (consulté le 05/05/2023) et Journal du Net  
<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/> (consulté le 05/05/2023)
- [9] Le langage de programmation JavaScript - MDN Web Docs  
<https://developer.mozilla.org/fr/docs/Web/JavaScript> (consulté le 05/05/2023)
- [10] Le framework Karma/Jasmine - CodeCraft  
<https://codecraft.tv/courses/angular/unit-testing/jasmine-and-karma/> (consulté le 12/05/2023)
- [11] Tutoriel Angular Tour of Heroes disponible sur le site officiel d'Angular  
<https://angular.io/tutorial/tour-of-heroes> (consulté le 17/04/2023)
- [12] Tutoriel Karma/Jasmine - LogRocket  
<https://blog.logrocket.com/angular-unit-testing-tutorial/> (consulté le 21/04/2023)
- [13] Qu'est ce qu'un IDE ? - Amazon.com <https://aws.amazon.com/fr/what-is/ide/> (consulté le 12/05/2023)
- [14] Création de diagramme - Lucid <https://lucid.co/fr> (consulté le 26/05/2023)
- [15] Le type Object - MDN Web Docs  
[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects/Object](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Object) (consulté le 01/06/2023)
- [16] Le site d'InterAAActionScene - <https://interactiongroup.github.io/InterAAActionScene/fr> (consulté le 06/06/2023)
- [17] Github - À propos de Github Pages  
<https://docs.github.com/fr/pages/getting-started-with-github-pages/about-github-pages> (consulté le 06/06/2023)

[18] Futura - Markdown : Qu'est-ce que c'est ?

<https://www.futura-sciences.com/tech/definitions/informatique-markdown-17135/> (consulté le 06/06/2023)

**Résumé :**

Ce rapport explique ce que j'ai pu vivre et produire durant ma période de stage. Celle-ci a eu lieu au Laboratoire Informatique de Grenoble, au sein de l'équipe GETALP\*. Une partie de l'équipe GETALP travaille sur leur projet InterAActionBox\*, qui cherche à favoriser l'apprentissage et la communication des personnes en situation de handicap cognitif, du plus petit au plus grand âge. Il intègre plusieurs applications de Communication Alternative et Augmentée (CAA)\*, dont InterAActionScene\*, une application permettant à l'utilisateur d'interagir avec des scènes\* et des images qu'il a créé. Mes missions de stages se sont ciblées sur cette application. Je me suis tout d'abord formé au framework\* Angular\*, avec lequel est développé InterAActionScene\*. Après cela, j'ai pu travailler pleinement sur l'application pour implémenter des nouvelles fonctionnalités, comme améliorer et ajouter des interactions avec les scènes\* et les images de l'utilisateur, améliorer leur gestion, etc... Des modifications et des ajouts ont été ensuite apportés dans les fichiers de tests, suite à ces modifications.

**Mots-clés :**

*LIG, GETALP, InterAActionBox, CAA, handicap cognitif, InterAActionScene, oculomètre, Angular, Karma/Jasmine, initiation, tutoriels, développement, tests, utilisateur, interactions*

**Abstract :****LIG internship report on the InterAActionScene project**

**Yanis Girardin, Grenoble**

This report tells what I experienced and produced during my internship. It took place at the Laboratoire Informatique de Grenoble, within GETALP\* team. A part of GETALP\* work on a project named InterAActionBox\*. This project aims to help people with cognitive disabilities to learn and communicate. It includes several applications of Alternative and Augmented Communication (CAA)\*, such as InterAActionScene\*, an application allows the user to interact with scenes\* and images he created. My internship missions are focused on this application. Firstly, I learned how to use the Angular\* framework\*, which is used to develop InterAActionScene\*. Then, I could begin to work on this application and implement new features, like upgrade and add user interactions with the scenes\* and images, improve their manageability, etc... Finally, changes and add-ons were made on test files following these modifications.

**Keywords :**

*LIG, GETALP, InterAActionBox, CAA, cognitive disabilities, InterAActionScene, eye tracker, Angular, Karma/Jasmine, initiation, tutorials, development, tests, user, interactions*