



Universidade Federal do Rio Grande do Sul  
Porto Alegre, Brésil

# LEPAGE Tim RICM Polytech Grenoble Rapport de stage de 4<sup>ème</sup> année

Interface Homme/Machine appliquée à la réalité augmentée

Tome Principal et Annexes

4<sup>ème</sup> année universitaire  
14.05.2018 - 03.08.2018

Enseignant Polytech : Jean-François Méhaut  
Enseignant UFRGS : Luciana Nedel



# REMERCIEMENTS

Avant de commencer je tiens à remercier toutes les personnes qui m'ont accompagné physiquement, financièrement et moralement pendant ce stage et sans qui cette expérience aurait été impossible.

Tout d'abord je remercie mon professeur Jean-François Méhaut pour m'avoir aidé dans cette recherche de stage et m'avoir mis en relation avec ma tutrice sur place, Mme Luciana Nedel de l'UFRGS. Je le remercie de m'avoir suivi pendant ces douze semaines et d'avoir été à l'écoute en cas de besoin. J'en profite également pour remercier mon colocataire Leo Valette, élève de RICHM pour avoir rendu le séjour plus divertissant.

Je tiens ensuite à remercier ma tutrice de stage Mme Luciana Nedel professeure à l'UFRGS pour m'avoir accueilli dans son laboratoire, accompagné et conseillé tout au long du stage.

J'adresse également mes remerciements à tous les laborantins du laboratoire d'informatique de l'UFRGS pour leur accueil chaleureux qui a facilité mon intégration rapide au sein de leur équipe. Je remercie en particulier Jerônimo G. Grandi pour m'avoir accueilli et pour avoir tenu les paris pendant la Coupe du monde de Football.

Je suis reconnaissant enfin envers la région Auvergne-Rhône-Alpes qui m'a accordé la bourse Explo'ra sans laquelle ce voyage n'aurait pas eu lieu. Je remercie toutes les personnes m'ayant aidé à rédiger ce rapport. Je remercie finalement mes parents pour leur soutien moral et financier durant ces douze semaines.



# RÉSUMÉ

La technologie prend de plus en plus de place dans nos vies, et le monde s'informatise de jour en jour. Il en va de même pour la visualisation de données. Ce qui, dans le temps, se représentait avec un boulier l'est aujourd'hui avec un tableur informatique. Cependant les représentations physiques de données restent très populaires de par leur intuitivité de manipulation. Mais comment mêler le physique et le virtuel pour faciliter encore la lecture de données ?

C'est pour apporter des éléments de réponse à cette problématique que j'ai effectué mon stage de 4<sup>ème</sup> année RICM au sein de l'Institut d'Informatique, de l'Universidade Federal do Rio Grande do Sul (UFRGS) à Porto Alegre, au Brésil.

J'ai travaillé durant douze semaines pour mieux comprendre la data visualisation afin d'offrir une application permettant de faciliter celle-ci en liant le physique et l'informatique. C'est là qu'intervient la réalité augmentée, parfait intermédiaire entre le réel et le virtuel.

Ce rapport fait état du déroulement de ce stage.

# ABSTRACT

Technology is becoming more and more important in our everyday live, and the world is becoming more and more computerized every day. This also applies for the data visualization. What used to be represented with an abacus is today represented with a spreadsheet. However, the physical representations of data remain highly popular because of their intuitive handling. But how to mix the physical and the virtual to further facilitate the reading of data?

In order to provide answers to this problem, I did my 4<sup>th</sup> year RICM internship at the Institute of Computer Science, in the Universidade Federal do Rio Grande do Sul (UFRGS) in Porto Alegre, Brazil.

I worked for twelve weeks to get a better understanding of the data visualization and to offer an application to facilitate it by linking the physical and the computer. This is where the augmented reality plays a huge role, perfect intermediary between the real and the virtual.

This report describes the progress of this internship.



# TABLE DES MATIÈRES

<b>GLOSSAIRE</b>	<b>8</b>
<b>INTRODUCTION</b>	<b>9</b>
<b>ENVIRONNEMENT ET CADRE</b>	<b>10</b>
Le cadre de travail	10
Le cahier des charges	10
<b>TÂCHES ET RÉALISATIONS</b>	<b>11</b>
Vuforia	11
Image Tracking	11
Extended Tracking	13
Ground Plane Tracking	13
OCR.Space	15
OpenCV	16
Concept de l'application	17
Google Calendar API	19
Réalisations	20
L'application	20
Le site web	22
Tests	24
<b>BILAN ET CONCLUSION</b>	<b>26</b>
Perspectives	26
Bilan personnel	26
Conclusion	27
<b>RÉFÉRENCES</b>	<b>28</b>
<b>ANNEXES</b>	<b>29</b>
ANNEXE 1 : Vuforia Ground Plane API Overview	29
ANNEXE 2 : Diagrammes UML du projet	30
ANNEXE 3 : Détail des tâches à effectuer pour le test de rapidité	31
ANNEXE 4 : Diagramme de Gantt du projet	32

# TABLE DES FIGURES

Figure 1: Exemple d'une bonne image pour la reconnaissance avec Vuforia	13
Figure 2: Exemple d'une mauvaise image pour la reconnaissance avec Vuforia	13
Figure 3: Histogrammes en niveau de gris de l'image test avant et après ajustement	14
Figure 4: Aperçu de l'application permettant de mesurer des distances avec Vuforia Ground Plane	15
Figure 5: Exemple de document JSON renvoyé avec OCR	16
Figure 6: Cercle des couleurs	17
Figure 7: Critères HSV utilisés pour filtrer la couleur rouge	17
Figure 8: Rendu conceptuel de l'application sur le smartphone	19
Figure 9: Diagramme de classes de l'API Google Calendar	21
Figure 10: Capture d'écran de l'application avant activation de la synchronisation	22
Figure 11: Capture d'écran de l'application une fois les données synchronisées	23
Figure 12: Capture d'écran du site web listant les événements à venir et proposant de synchroniser le calendrier en ligne en choisissant la date de début	25
Figure 13: Capture d'écran du Google Agenda avec les événements scannés synchronisés	25
Figure 14: Résultats des tests de rapidité	26



# GLOSSAIRE

API	Application Programming Interface, ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels
Augmentation	En réalité augmentée une augmentation est un élément virtuel apparaissant sur l'écran et agrémentant la réalité.
Data Visualisation	L'étude, la science ou l'art de représenter des données de façon visuelle.
HSV	Hue Saturation Value (Teinte Saturation Valeur), un système de gestion des couleurs en informatique.
OS	Operating System (Système d'Exploitation), ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs.
SDK	Software Development Kit, ensemble d'outils logiciels destinés aux développeurs, facilitant le développement d'un logiciel sur une plateforme donnée.

## INTRODUCTION

Du 14 Mai 2018 au 3 Août 2018, j'ai effectué mon stage d'ingénieur Réseau Informatique et Communication Multimédia de quatrième année au sein du laboratoire de l'institut d'informatique de l'Université fédérale du Rio Grande do Sul de Porto Alegre au Brésil. Au cours de ce stage portant sur la réalité augmentée, j'ai pu m'intéresser à l'interface homme/machine pour développer une application permettant de faire de la visualisation de données en réalité augmentée.

D'une manière plus générale, ce stage m'a permis d'appréhender le fonctionnement du travail de chercheur et de la vie en laboratoire de recherche. Bien que j'ai effectué la totalité du stage en travaillant seul, j'ai pu observer le déroulement des projets de recherche de mes collègues laborantins. Ce stage m'a permis d'étendre mes compétences en programmation, en recherche documentaire ainsi qu'en travail en autonomie.

Cette nouvelle expérience qui s'est effectuée à l'étranger, au Brésil, m'a par ailleurs permis de voyager et de quitter l'Europe pour la première fois. Au delà de l'expérience professionnelle que ce stage m'a apportée, j'ai pu donc m'enrichir personnellement.

Mon stage à l'institut d'informatique de l'Université fédérale du Rio Grande do Sul de Porto Alegre, a consisté essentiellement en recherches documentaires sur la visualisation de données et lectures de papiers scientifiques précédant le développement d'une application. La tutrice de mon stage était Mme Luciana Nedel, enseignante-chercheuse à l'UFRGS. Malheureusement sa présence dans mon stage ne fut que très limitée mais nous aborderons ce point plus en détail par la suite.

La rédaction de ce rapport a pour principale source les différents enseignements tirés des tâches effectuées au cours de ces trois mois. Dans la suite de ce rapport, il sera présenté tout d'abord l'environnement et le cadre de ce stage, mentionnant notamment la principale difficulté rencontrée (I). Nous nous intéresserons ensuite aux différentes tâches et activités que j'ai effectuées au laboratoire (II). Enfin seront précisés et analysés les apports que j'ai pu en tirer (III).

## **I. ENVIRONNEMENT ET CADRE**

### **a. Le cadre de travail**

Ce stage de trois mois s'est déroulé dans un laboratoire de l'institut d'informatique de l'UFRGS et plus précisément dans le laboratoire de recherche sur la réalité virtuelle et la réalité augmentée. Ce laboratoire très moderne a été rénové début 2018 et abrite une dizaine de laborantins et doctorants. Le responsable du laboratoire est Jeronimo G. Grandi, un doctorant ayant publié plusieurs papiers sur la réalité augmentée. La tutrice responsable de mon stage était Mme Luciana Nedel, et malheureusement, comme évoqué dans l'introduction sa présence dans mon stage a été très limitée. Pendant les premières semaines, j'ai pu m'entretenir avec elle à plusieurs reprises. Elle a donc pu me transmettre un cahier des charges. Cependant, ce cahier est resté vague quant aux tâches que je devais accomplir. Passé les premières semaines au laboratoire, elle cessa de venir superviser mon avancement et ne répondait que très rarement à mes mails. D'où la principale difficulté rencontrée pendant mon stage : avancer dans la direction qu'elle attendait de moi.

### **b. Le cahier des charges**

L'objectif final de mon stage fut de créer une application de réalité augmentée simple d'utilisation et intuitive permettant de faire de la visualisation de données. Il n'y avait pas de contraintes d'utilisation de software ou de langage en particulier.

## II. TÂCHES ET RÉALISATIONS

Avant de rentrer dans les détails de mes réalisations, il est à noter que tout le code que j'ai produit pendant cette période de trois mois est disponible sur mon dépôt Github : <https://github.com/TimLepage/Stage2018-Augmented-Reality>.

Lors de ma première rencontre avec Madame Luciana Nedel, je n'avais aucune expérience avec la réalité augmentée. C'est pourquoi la première tâche qui me fut confiée consistait à me documenter sur les différents SDK de réalité augmentée et à générer une augmentation simple.

La première difficulté fut la compatibilité entre les SDK de réalité augmentée et, par manque de matériel, mon téléphone personnel (un acer liquid z630) qui tourne sur une version d'android (Android 5.1 API level 22) pour laquelle certains SDK ne sont pas utilisables. Notamment le SDK très poussé de Google : ArCore, supporté uniquement par les appareils de version d'Android supérieure à 7.0. Mon premier choix s'était donc porté sur Vuforia pour son accessibilité et sa compatibilité avec Unity que j'utilisais à ce moment là. La tâche consistait donc à tester les possibilités offertes par Vuforia afin de me rendre compte de si ce SDK conviendrait.

### a. Vuforia

#### Image Tracking

Vuforia permet de reconnaître et de suivre des images en analysant le contraste entre les différents éléments de l'image. Les 3 points clés pour une image cible correcte sont :

- La richesse en détails
- Le contraste élevé
- L'absence de schémas répétitifs

En plus de ces points, pour optimiser la détection, l'image doit être plate et en un matériau non brillant afin d'éviter les reflets à la caméra qui pourraient cacher l'image et empêcher par conséquent sa détection.

Exemple d'une image correcte :

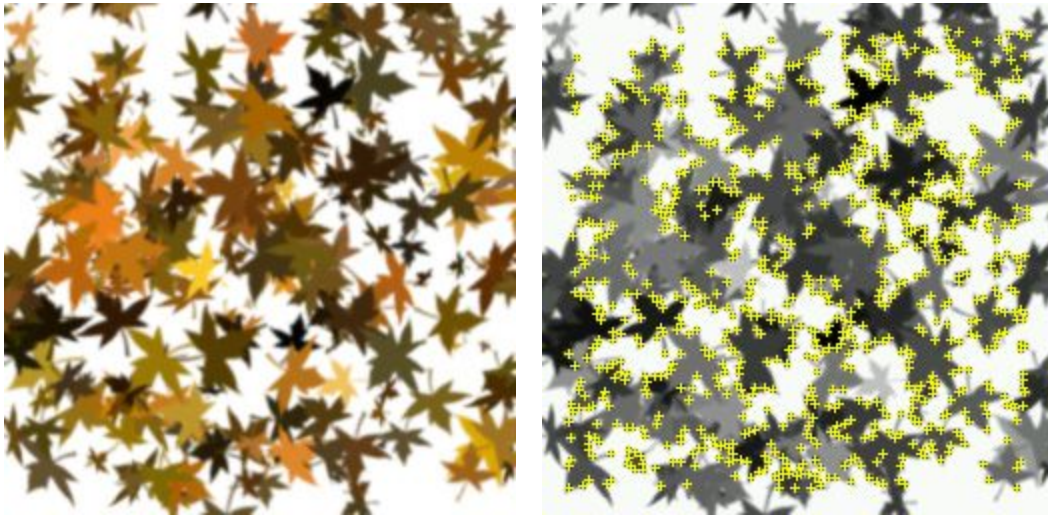


Figure 1 Exemple d'une bonne image pour la reconnaissance avec Vuforia

Exemple d'une mauvaise image :

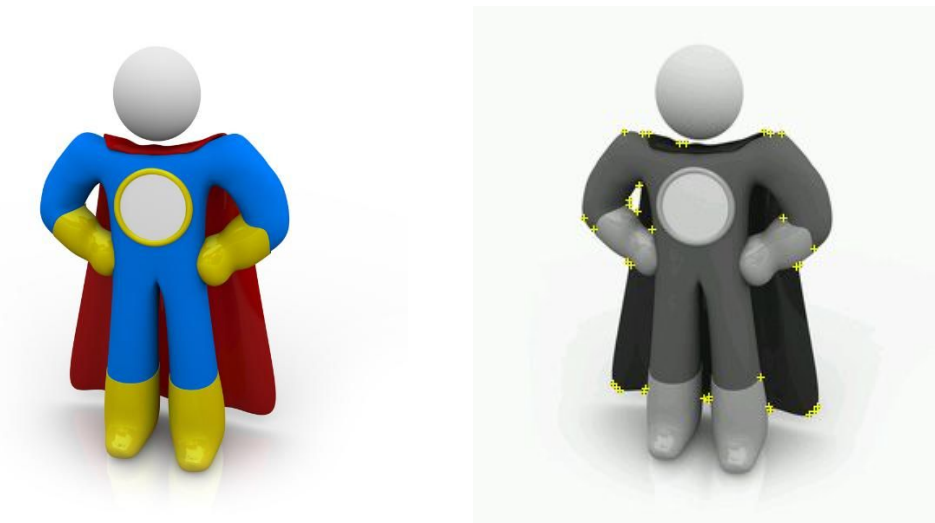


Figure 2 Exemple d'une mauvaise image pour la reconnaissance avec Vuforia

Les figures 1 et 2 montrent des images pouvant être traquées par l'outil de Vuforia, les croix jaunes sont les points de repère qu'utilise l'outil pour garder la trace de l'objet. Pour optimiser les performances, il faut donc une image avec un fort contraste et dont l'histogramme en niveau de gris soit le plus large possible.

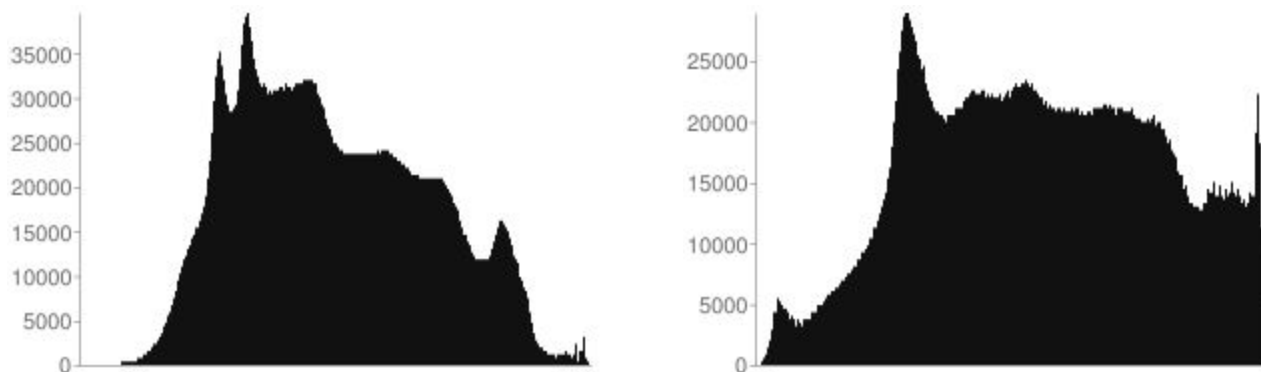


Figure 3 Histogrammes en niveau de gris de l'image test avant et après ajustement

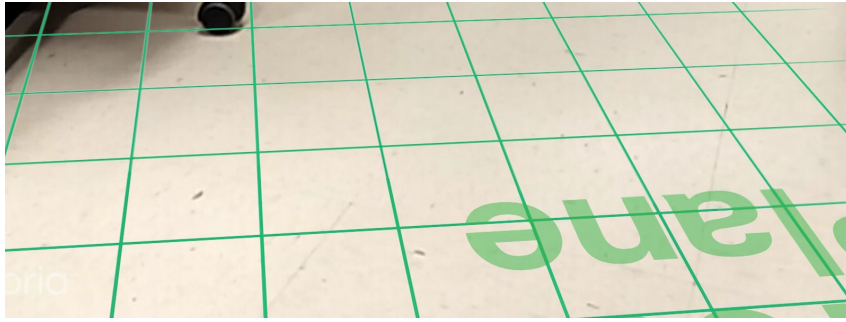
La figure 3 montre un histogramme en niveau de gris de l'image utilisée pour les tests avant (gauche) et après (droite) ajustement pour étendre l'histogramme en augmentant le contraste. Vuforia utilise un système de notation sur 5 pour juger d'une image cible. L'image correspondant à l'histogramme de gauche a obtenu la note de 4/5 et celle de l'histogramme de droite 5/5.

### Extended Tracking

Cette fonctionnalité se sert de l'environnement pour améliorer les performances de suivi et continuer de générer l'augmentation même quand l'image cible n'est plus en vue. Vuforia construit en mémoire une carte de l'environnement autour de l'image cible en supposant que cet environnement ne bougera pas. Cette carte permet ensuite à Vuforia de se repérer lorsque l'utilisateur bouge l'appareil. Cette fonctionnalité est utilisée pour apporter une meilleure stabilité à l'application, en particulier lorsque l'augmentation est grande.

### Ground Plane Tracking

Cette technologie permet d'une part, le placement de contenu sur les surfaces horizontales et, d'autre part, le placement de contenu dans l'air. Cette fonctionnalité utilise l'API Vuforia Smart Terrain. Celui-ci permet de trouver et de suivre les surfaces planes ainsi que d'effectuer des tests de touche sur ces surfaces. Elle utilise parallèlement le Device Tracking permettant de suivre la position et l'orientation de l'appareil dans l'espace (voir [annexe 1](#) pour une vue d'ensemble de l'API Ground Plane Tracking).



L'API détecte une surface plane en projetant des rayons virtuels à une position précise. En fonction de l'intersection ou non avec une surface, l'API indique l'absence ou la présence de celle-ci. Elle en informe ensuite l'indicateur de surface. Une application permettant de mesurer des distances (figure 4) a été réalisée en C# afin de tester la précision de cette fonctionnalité.

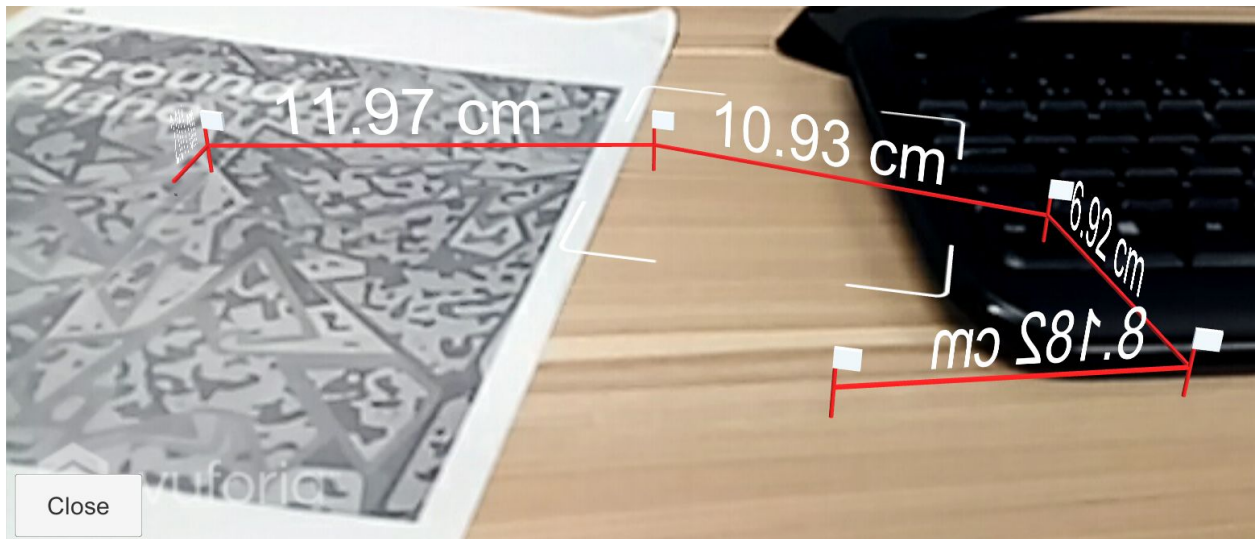


Figure 4 Aperçu de l'application permettant de mesurer des distances avec Vuforia Ground Plane

Cette application m'a permis de bien prendre en main le développement d'application en réalité augmentée. Il était donc temps à ce moment de me pencher plus en détail sur la data visualisation.



## b. OCR.Space

N'ayant aucune expérience avec la data visualisation et n'ayant reçu que peu d'instructions quant à l'application finale je me suis lancé dans la programmation d'une application permettant, grâce à l'OCR (Optical Character Recognition) et Vuforia, de lire et d'analyser des tableaux de nombres. Une des API existantes permettant la reconnaissance optique de caractères est OCR.Space (<https://ocr.space/>). Cette API en ligne permet d'extraire les caractères d'une image et de les transmettre au format JSON. Cependant cette API n'est pas parfaite et est incapable de reconnaître un caractère seul.

L'application que j'ai produite est capable de reconnaître un tableau de chiffres en deux dimensions, de trouver ses points extrêmes et d'en afficher la courbe en augmentation.

```
"Lines": [
  {
    "Words": [
      {
        "WordText": "-40",
        "Left": 94,
        "Top": 32,
        "Height": 27,
        "Width": 51
      }
    ],
    "MaxHeight": 27,
    "MinTop": 32
  },
  {
    "Words": [
      {
        "WordText": "-30",
        "Left": 94,
        "Top": 82,
        "Height": 27,
```

Figure 5 Exemple de document JSON renvoyé avec OCR.Space

Cependant, après discussion avec ma tutrice cette application ne correspondait pas à l'objectif qu'elle avait en tête et nous avons redéfini un cahier des charges plus précis. L'application devait pouvoir utiliser des Legos pour aider à visualiser les données que ceux-ci pouvaient représenter. En effet, comme le montrent plusieurs études[1][2], la manipulation de données est plus simple et intuitive lorsque l'on manipule physiquement les objets. Le but était donc de trouver des données que pouvait représenter l'agencement des Legos et de créer une application permettant de les visualiser (à noter que l'application doit pouvoir reconnaître n'importe quel objet de couleur, par la suite, je ne parlerai que de la reconnaissance de Legos car c'étaient les objets avec lesquels j'ai testé l'application et ceux-ci se prêtaient particulièrement bien à l'exercice).



### c. OpenCV

Dans le but de pouvoir reconnaître et différencier des pièces de Lego, il fallait un outil capable de reconnaître les couleurs, chose impossible avec Vuforia. Ce fut donc la librairie graphique OpenCV qui fut choisie.

OpenCV est une librairie logicielle open source pour la computer vision et le machine learning. Les algorithmes fournis peuvent servir à reconnaître des visages, des objets, créer des modèles 3D d'objets, reconnaître une scène et établir des marqueurs de réalité augmentée, etc. Cette librairie était donc parfaite pour la reconnaissance de pièces de Lego. Voici les étapes de l'algorithme de reconnaissance et de traque des Legos que j'ai développé.

- Récupérer la capture vidéo.
- Convertir la video RGB en HSV.
  - La teinte (*Hue*) est codée suivant l'angle qui lui correspond sur le cercle des couleurs.

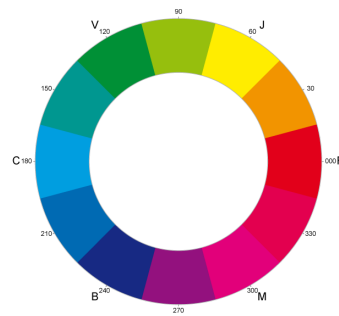


Figure 6 Cercle des couleurs

- La saturation (*Saturation*) est « l'intensité » de la couleur. Plus la saturation d'une couleur est faible, plus l'image sera « grisée » et plus elle apparaîtra fade.
- La valeur (*Value*) est la « brillance » de la couleur. Plus la valeur d'une couleur est faible, plus la couleur est sombre. Une valeur de 0 correspond au noir.
- Filtrer le résultat selon les critères désirés, par exemple pour le rouge dont je disposais, la teinte doit être comprise entre 0 et 4 (sur 256) pour être reconnu.

```
int H_MINR = 0; //RED
int H_MAXR = 4;
int S_MINR = 105;
int S_MAXR = 256;
int V_MINR = 0;
int V_MAXR = 256;
```

Figure 7 Critères HSV utilisés pour filtrer la couleur rouge

La valeur d'intensité est ici fixée à 105 pour optimiser la détection de la couleur malgré l'ambiance lumineuse changeante de mon poste de travail.

- Réduire le bruit et lisser les contours (On ne garde que les zones de plus de 100px par 100px). Cette étape permet un tracking optimal et empêche les petites imperfections de détection d'être perçues comme des éléments individuels en les incorporant dans l'objet traqué auquel elles appartiennent.

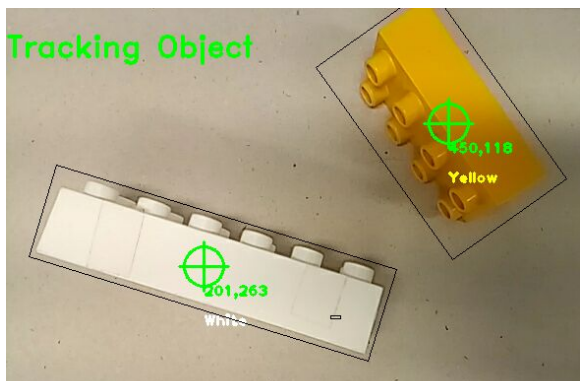
Avant réduction du bruit



Après réduction du bruit



- Tracker l'objet



Grâce à OpenCV il m'était possible de détecter les pièces de Lego et de les suivre en temps réel tout en affichant des informations liées à ces pièces.

Le résultat étant conforme aux attentes de ma tutrice, je devais désormais trouver une idée d'application utilisant ce système pour faire de la data visualisation.

#### d. Concept de l'application

Après recherches et en m'inspirant sur diverses études et réalisations[3][4][5], il me vint l'idée d'une application utilisant les pièces de Lego dans la gestion de projet. Des études montrent en effet qu'il est plus facile et

intuitif de manipuler des données quand elles sont représentées physiquement plutôt que sous format numérique. Alors pourquoi ne pas se servir de Legos pour représenter un calendrier où chaque pièces correspondrait à une tâche et à une durée ? Cela permettrait donc de pouvoir organiser un ou plusieurs calendriers en les manipulant physiquement, de voir qui fait quoi et quand. J'ai moi même souvent été confronté à ce problème lors de projets d'étude : répartir le travail et suivre l'avancée de chacun se révèle souvent être compliqué. Pourtant, comme vu en cours de Génie Logiciel<sup>1</sup> lors de ma formation, ce point est capital pour mener à bien une réalisation collective. L'objectif de cette application est donc de réduire les erreurs de gestion de temps et de mieux organiser la répartition des tâches au sein d'un projet de groupe.

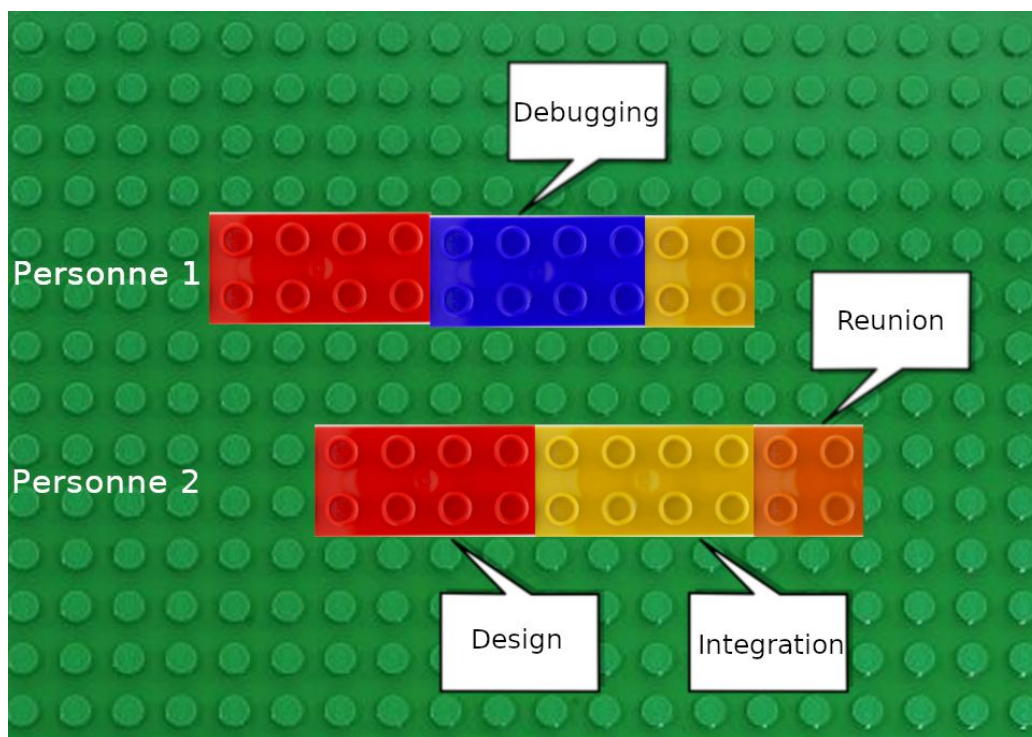


Figure 8 Rendu conceptuel de l'application sur le smartphone

En synchronisant la détection des Legos avec un calendrier en ligne, il est donc possible de partager le calendrier avec toutes les personnes impliquées dans le projet. Ainsi une seule personne peut s'occuper de tenir à jour un calendrier et par l'intermédiaire d'une API, gérer les différentes tâches de tout le personnel.

---

<sup>1</sup> Enseignement Génie Logiciel en RICM4 Polytech Grenoble [link](#)

### e. Google Calendar API

Mon choix d'API pour la gestion d'un calendrier s'est porté sur Google Calendar API car celui-ci offre plusieurs avantages :

- la possibilité d'intégrer Google Calendar (alias Google Agenda) à nos applications grâce à leur API dédiée.
- Google Calendar est accessible via une application sur tous les smartphones Android et iOS qui ne représentent pas moins de 99,6% des smartphones en utilisation dans le monde en 2017[6]. Ainsi l'application peut profiter à la majorité des utilisateurs potentiels.
- Google Calendar est également accessible en ligne depuis les navigateurs Google Chrome, Internet Explorer, Microsoft Edge, Firefox et Safari, ce qui rend accessible l'application à la quasi-totalité des utilisateurs.

La figure 9 montre le diagramme de classes de l'API de Google Calendar.

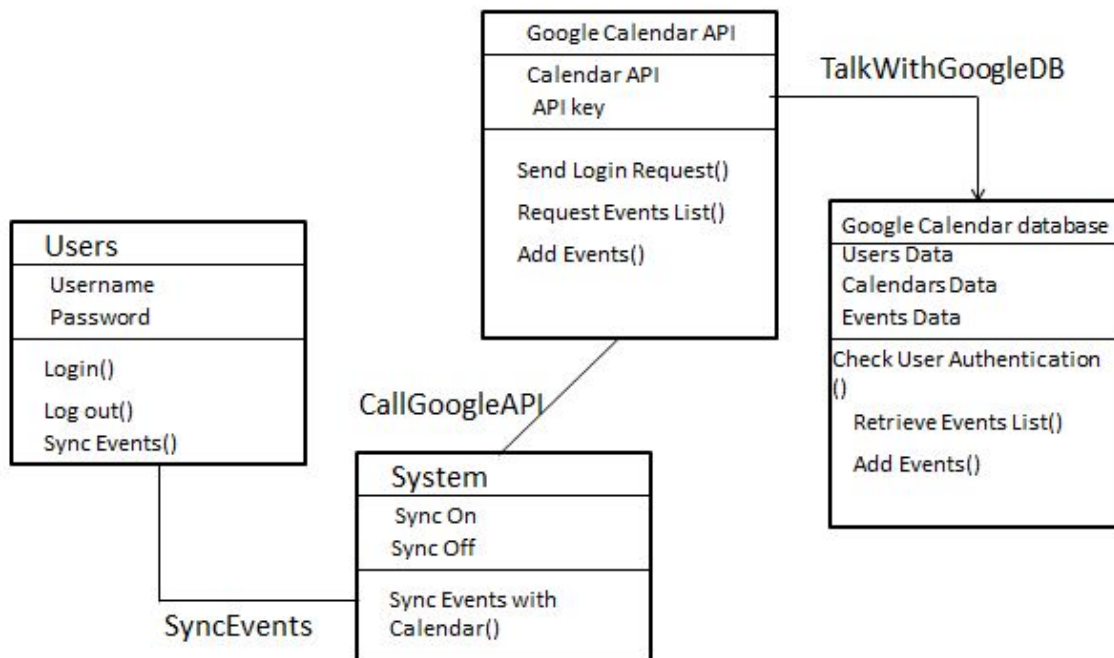


Figure 9 Diagramme de classes de l'API Google Calendar

Suite à ces recherches, je me suis lancé dans mes propres réalisations que je vais détailler ci-dessous.

## f. Réalisations

Dans cette partie je ne m'attarderai que sur mes réalisations personnelles que j'ai retenues. C'est la raison pour laquelle je ne parlerai plus des diverses productions réalisées avec Vuforia notamment. En revanche, je détaillerai l'application utilisant OpenCV que j'ai codée durant mon stage et un site web fonctionnant de pair (voir [annexe 2](#) pour les diagrammes UML du projet).

### L'application

L'application réalisée est codée en C++ en utilisant la librairie OpenCV. Après cadrage des pièces de Legos organisées de telle sorte à former un planning, l'application permet d'afficher à l'écran les informations relatives aux couleurs et pièces de Lego (activité, durée, date de début et de fin, etc.). D'une pression du doigt on peut envoyer ces informations vers le site web afin de les synchroniser avec Google Calendar. L'application n'est composée que d'une page et ne peut s'utiliser seule pour modifier le planning en raison de l'incompatibilité entre le langage C++ et l'API de Google Calendar. Cette page sert donc essentiellement à la visualisation du planning et, surtout, à son élaboration.

Lors du lancement de l'application, l'algorithme détecte automatiquement les pièces de Lego et affiche un rectangle noir autour de chacune d'elles (figure 9). Afin de créer le calendrier associé à cet ordre, il suffit d'appuyer dans l'un de ces rectangles.

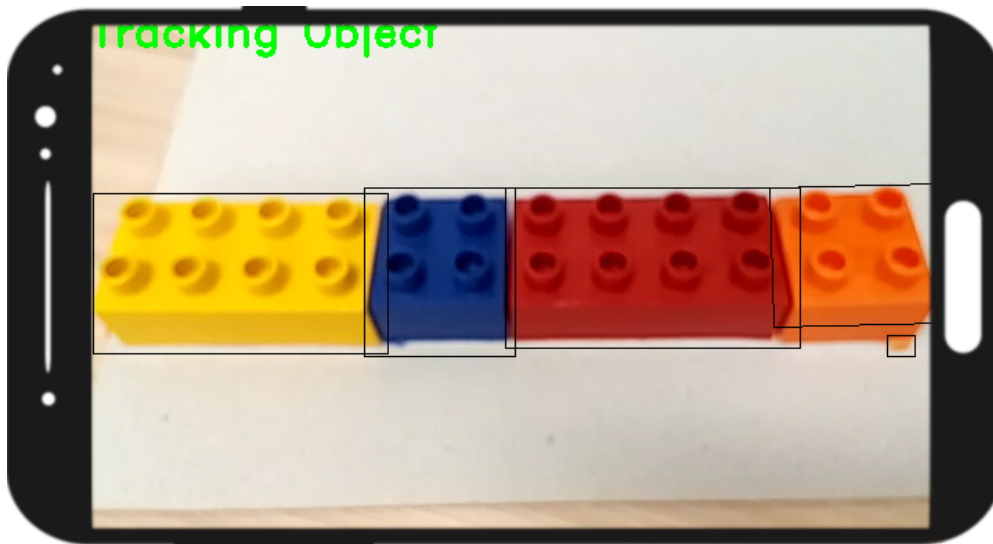


Figure 10 Capture d'écran de l'application avant activation de la synchronisation

Presser une pièce de Lego déclenche l'affichage des informations associées

(figure 11) ainsi que l'envoi des informations jusqu'au site internet du projet. Ici, les données affichées sont la position horizontale des Legos, la couleur détectée, l'aire du rectangle associé ainsi que sa taille en unité de Lego.

Afin de pouvoir mesurer les Legos quelle que soit la distance entre l'utilisateur et les pièces de Lego, il faut utiliser une pièce de Lego qui servira d'étalon. L'application que j'ai créée a été codée de sorte à ce que ce soit la pièce de couleur jaune et de taille 4 qui serve d'étalon. Ce choix était dû au fait que cette couleur se détectait particulièrement bien dans mon environnement de travail et que plus la taille de la pièce étalon est grande plus la mesure des pièces est précise. Ce paramètre est cependant facilement modifiable dans le code C++ de l'application. Afin de calibrer l'application il suffit de cadrer une pièce de Lego jaune de taille 4 avant de cadrer les pièces du planning (le bloc étalon n'est pas nécessairement dans le planning par la suite. Le simple cadrage suffit).

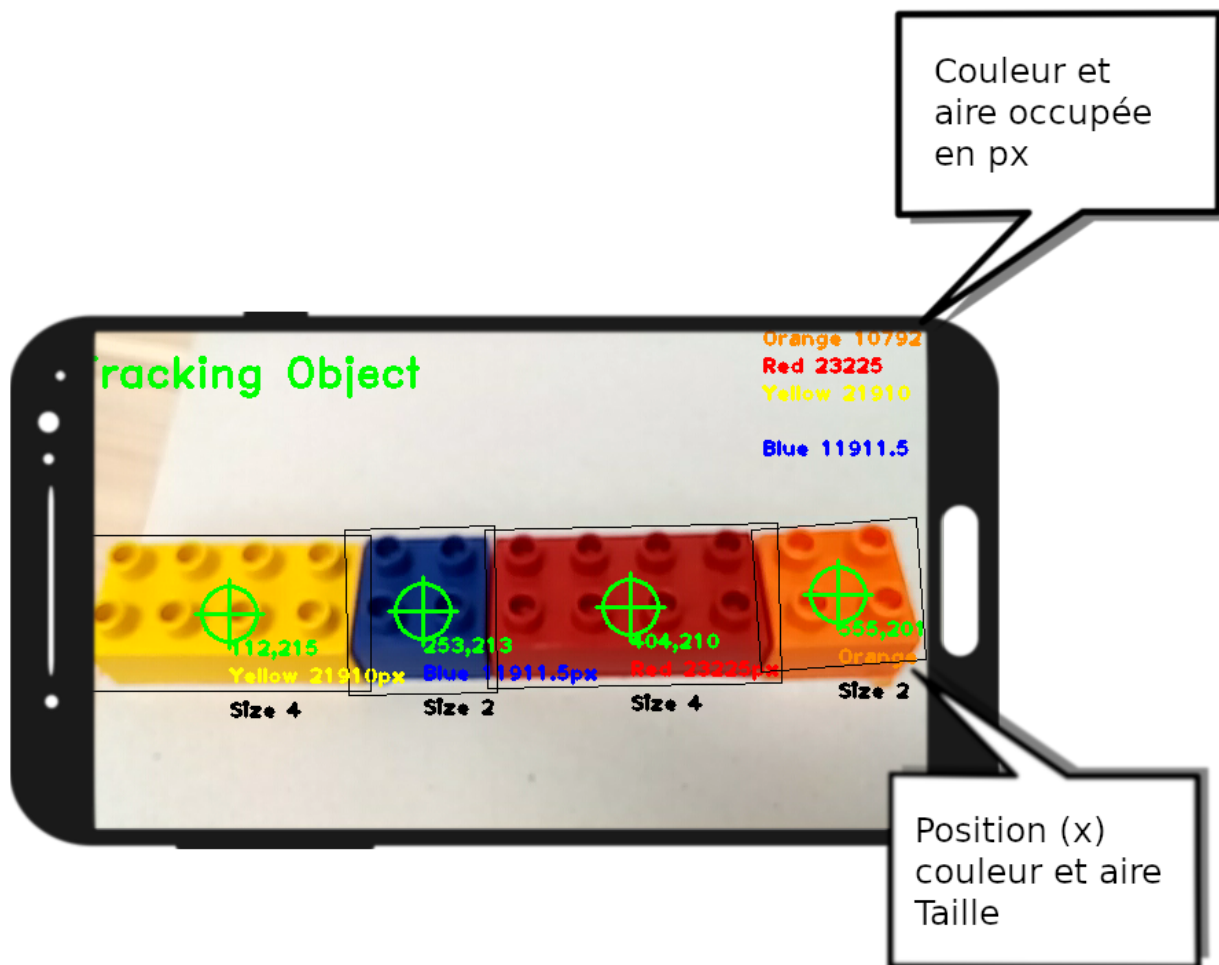


Figure 11 Capture d'écran de l'application une fois les données synchronisées



Par la suite et sur proposition de ma tutrice, j'ai développé la possibilité de détecter et d'enregistrer plusieurs lignes de Legos superposées. Cette fonctionnalité est destinée à être utilisée pour scanner une longue liste de blocs de façon compacte. En effet, une distance de 30 unités de Lego (censée représenter un mois ici) correspond à 50 centimètres de long. En répartissant ces 30 unités sur plusieurs lignes (par exemple 7 unités par ligne pour une semaine) on peut représenter ces 30 jours sur une surface de 20 centimètres carrés, surface bien plus pratique à manipuler, visualiser et scanner.

### Le site web

Afin de pouvoir utiliser l'API Google Calendar il faut pouvoir effectuer des requêtes. Pour faire appel à cette API, la liste des langages supportés est limitée. Mon choix s'est donc porté sur le JavaScript. La principale raison de mon choix réside dans le fait que je connais ce langage. J'ai déjà eu l'occasion d'utiliser au cours de mon projet du second semestre de 4<sup>ème</sup> année[7]. Je me suis donc lancé dans la création d'un site web avec HTML5 et JavaScript développé avec un python webserver en localhost avec la commande *python -m http.server 8000*. Ce site permet à l'utilisateur de synchroniser la ou les ligne(s) de Lego avec son agenda en ligne Google Agenda en choisissant la date à laquelle il souhaite insérer le planning ainsi que les événements à associer aux différentes couleurs scannées. De plus, ce site offre la possibilité de voir les événements à venir. L'utilisateur a également la possibilité de visualiser les couleurs et tailles des blocs scannés, ce qui lui permet de savoir précisément quels événements il va pouvoir ajouter. Le choix d'implémenter un site web pour la synchronisation s'inscrit dans la volonté de rendre le projet utilisable par un maximum d'utilisateurs sans avoir à coder deux versions de l'application pour les deux OS dominants.

## Synchronise your calendar

### Upcoming events :

Select the number of events to display

Event	Start Date	End Date
Yellow	2018-07-16	2018-07-20
Blue	2018-07-20	2018-07-22
Red	2018-07-22	2018-07-26
Orange	2018-07-26	2018-07-28

### Add the scanned events

#### Scanned events :

Yellow size : 4  
Blue size : 2  
Red size : 4  
Orange size : 2

#### Corresponding Event:

Orange	<input type="text" value="Orange"/>
Red	<input type="text" value="Red"/>
Blue	<input type="text" value="Blue"/>
Yellow	<input type="text" value="Yellow"/>

Start Date :

Figure 12 Capture d'écran du site web listant les événements à venir et proposant de synchroniser le calendrier en ligne en choisissant la date de début.

16	17	18	19	20	21	22
Yellow				Blue		Red
23	24	25	26	27	28	29
Red			Orange			

Figure 13 Capture d'écran du Google Agenda avec les événements scannés synchronisés

Il est également possible via le site web de mettre à jour le planning existant en choisissant une date à laquelle un événement est déjà présent. Une boîte de dialogue s'ouvre alors et demande la confirmation de la mise à jour qui écrasera l'ancien événement.

localhost:8000 says

Do you really want to overwrite the event Yellow on 2018-07-10



L'accès de ce site se fait par authentification avec un compte Google. Ce compte est le compte auquel le calendrier en ligne est associé.

## Tests

Bien que l'application ne soit pas entièrement terminée, j'ai voulu effectuer quelques tests d'ergonomie. Le but de l'application étant de faciliter la gestion d'un calendrier, j'ai créé un test permettant d'estimer les performances du projet. Celui-ci est un test de rapidité qui consiste en deux étapes : la première étant de créer une suite de plusieurs événements différents sur une période de 19 jours, et la deuxième consiste à mettre à jour la liste d'événements précédemment créée. La période choisie pour les tests ainsi que le nombre d'événements sont volontairement importants car l'application est justement conçue pour cette utilisation. Il est bien entendu plus rapide de passer directement par Google Agenda pour créer un unique événement plutôt que de scanner une ligne de Lego d'une même couleur. La figure 14 ci-après montre les résultats de ces tests.

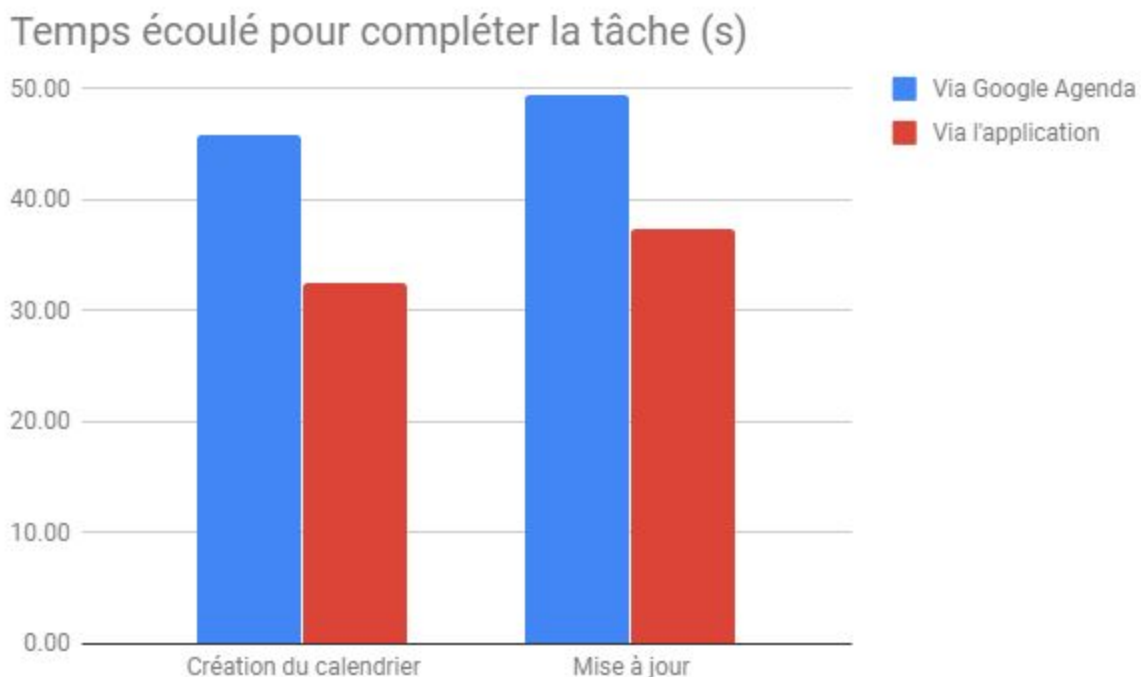


Figure 14 Résultats des tests de rapidité

Le détail des tâches à effectuer pour mener à bien ces deux étapes se trouve en [annexe 3](#). On observe sur ces graphiques un taux d'économie de temps de 29% en moyenne pour la création du calendrier et 24.4% pour sa mise à jour.

Ces tests ont été conçus et réalisés en toute honnêteté par moi-même, je suis donc bien conscient qu'une personne essayant l'application pour la première fois aurait des résultats sans doute moins bons. Cependant, cela montre qu'un utilisateur aguerri finira par obtenir de meilleurs résultats en terme de vitesse d'exécution avec mon projet plutôt qu'avec l'interface Google Agenda. Il est néanmoins nécessaire de garder en tête que la vitesse d'exécution n'est pas le seul point important de ce projet. La facilité d'utilisation et l'intuitivité du produit sont également à prendre en compte. L'application ayant été terminée pendant les périodes de vacances au Brésil, je n'ai malheureusement pas pu la faire tester par mes collègues laborantins absents à ce moment là.

### III. BILAN ET CONCLUSION

Dans ce bilan je vais aborder les différents points qu'il reste à considérer et à corriger pour achever complètement ce projet. Je discuterai ensuite des résultats de ce stage d'un point de vue personnel et je conclurai enfin ce rapport.

#### a. Perspectives

Étant initialement parti dans la mauvaise direction (voir [annexe 4](#) pour le diagramme de Gantt du stage), j'ai perdu un temps précieux et n'ai donc pas pu finir le projet comme je l'aurais voulu. Le plus gros problème au moment où j'écris ce rapport a trait à l'optimisation. Comme l'application mobile effectue de très nombreuses opérations chaque frame (tracking, affichage, écriture et récupération des données scannées, etc.), on constate un délai de presque une seconde entre le mouvement dans la réalité et l'image affichée. Bien que ce soucis n'empêche pas l'utilisation de l'application, cela rend son utilisation quelque peu inconfortable, du moins pour moi. Je pense que ce problème pourrait être facilement corrigé en optimisant mes fonctions, chose que je n'ai pas eu le temps de faire. En dehors de ce délai, il reste quelques problèmes de détection des couleurs, notamment lorsqu'on essaie de détecter plusieurs cubes de la même couleur. Parfois, l'un des Lego n'est simplement pas détecté, d'autre fois, les deux blocs sont détectés comme un unique grand bloc. Mis à part ces deux soucis, et à condition de déployer le site web sur un serveur, le projet est entièrement fonctionnel. Bien sûr, des modifications purement esthétiques devraient également être apportées.

#### b. Bilan personnel

Ces douze semaines de stage à l'Instituto de Informática da UFRGS ont été pour moi une expérience enrichissante au niveau personnel, professionnel mais aussi culturel. J'ai pu découvrir une autre culture, observer ses différences avec la culture française mais également des points communs. Ce stage m'aura permis de voyager dans un pays où je ne serais probablement jamais allé autrement. J'ai donc pu développer mes compétences linguistiques en anglais tout en acquérant des rudiments de portugais. Moi qui n'avais jamais quitté l'Europe, ce voyage fut donc ma première expérience aussi loin de mes racines. J'en tire un bilan positif. Même si je ne suis pas très friand de voyage, ce fut une expérience à vivre.

J'ai également pu renforcer mes compétences informatiques, notamment

en programmation. J'ai découvert un nouveau langage (C#) ce qui ne peut que m'être bénéfique pour ma vie professionnelle. J'ai aussi approfondi mes connaissances dans les langages HTML, JavaScript et surtout C++. En revanche, l'impossibilité de travailler en équipe durant ce projet ne m'a pas permis de mettre pleinement en application les méthodes agiles étudiées en cours. En discutant avec des camarades ayant effectué leur stage en entreprise, cet aspect semble avoir été d'une importance majeure dans leur expérience. En contrepartie, cela m'a permis de travailler sur mon autonomie et d'accroître ma faculté à résoudre des problèmes par moi-même.

### **c. Conclusion**

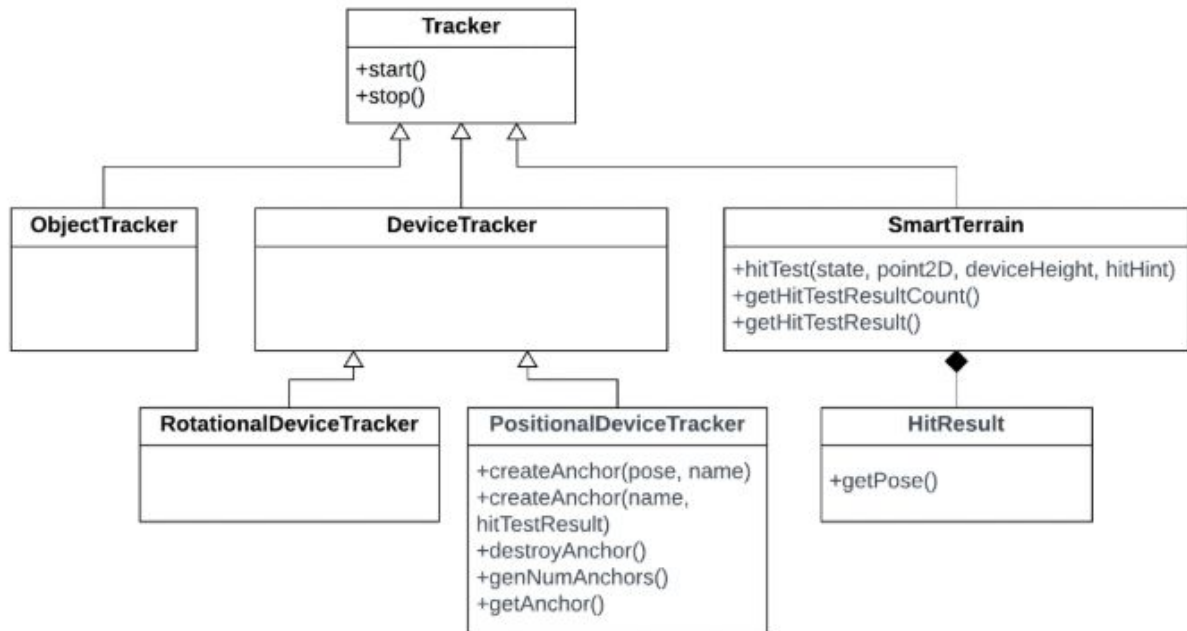
En conclusion je suis plutôt satisfait du travail fourni et ma tutrice Mme Luciana Nedel le semble également. Cependant je pense que le projet n'est pas encore assez abouti pour être réellement utilisé à grande échelle. Comme évoqué ci-dessus, il reste certains dysfonctionnements et soucis d'optimisation que je n'ai malheureusement pas pu résoudre. Malgré tout, je pense avoir acquis des compétences utiles au cours de ce stage que je compte mettre à profit à mon retour à Polytech mais également et surtout, plus tard, dans ma vie professionnelle.

# RÉFÉRENCES

- [1] Yvonne Jansen, Pierre Dragicevic, Jean-Daniel Fekete (2013) Evaluating the Efficiency of Physical Visualizations <http://www.aviz.fr/phys> [28/05/2018]
- [2] Samuel Huron, Yvonne Jansen, Sheelagh Carpendale (2014) Constructing Visual Representations : Investigating the Use of Tangible Tokens <https://hal.inria.fr/hal-01024053> [02/06/2018]
- [3] 2012 – Bit Planner : LEGO calendar <http://www.bit-planner.com/> [18/06/2018]
- [4] Michael Hunger (2008) On LEGO Powered Time-Tracking. Blog post. <http://jexp.de/blog/2008/08/on-lego-powered-time-tracking-my-daily-column/> [18/06/2018]
- [5] Mark Wilson (2012) How GM Is Saving Cash Using Legos As A Data Viz Tool <https://www.fastcodesign.com/1669468/how-gm-is-saving-cash-using-legos-as-a-data-viz-tool> [26/06/2018]
- [6] Worldwide Smartphone Sales to End Users by Operating System in 4Q16 <https://www.gartner.com/newsroom/id/3609817> [19/07/2018]
- [7] Dimitri Sergeant, Tim Lepage (2018) Serious game multi-joueurs pour tables tactiles en réseau [https://air.imag.fr/index.php/Projets\\_2017-2018](https://air.imag.fr/index.php/Projets_2017-2018) [24/07/2018]

# ANNEXES

## ANNEXE 1 : Vuforia Ground Plane API Overview

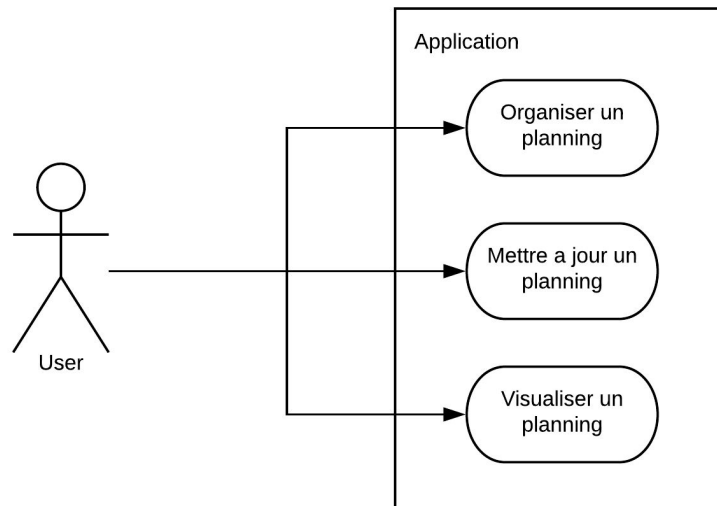


### Étapes de contrôle d'une application utilisant Ground Plane :

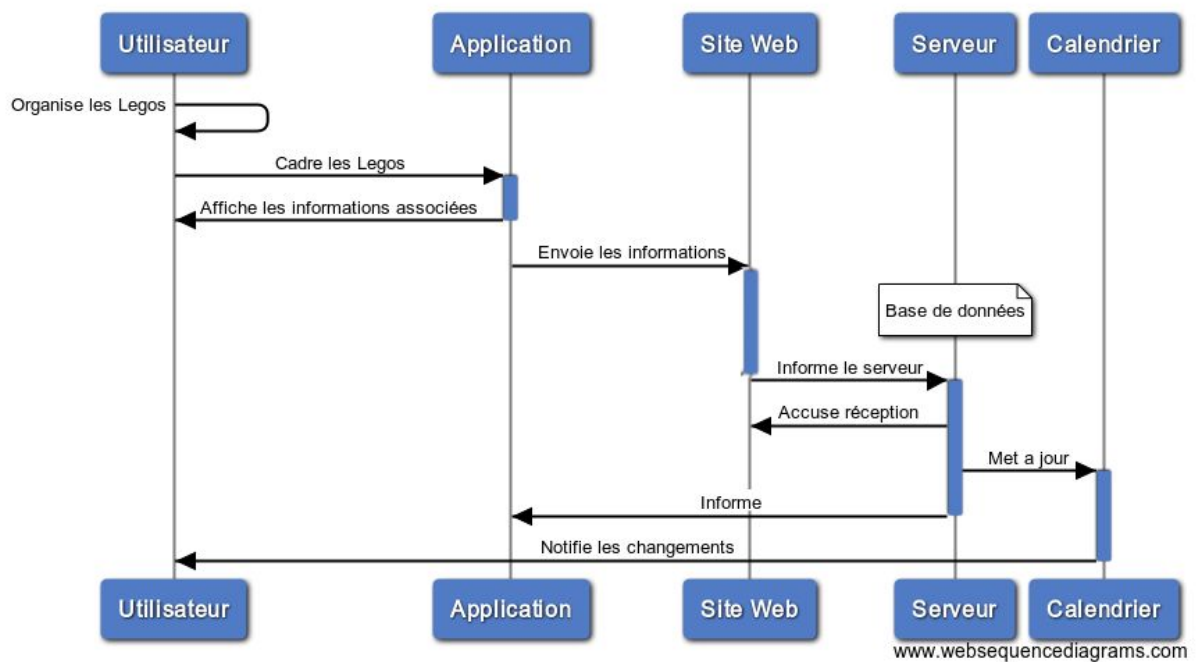
- Initialize Vuforia AR
- Start the AR Session
- Initialize and Configure the Trackers required for Ground Plane (PositionalDeviceTracker and SmartTerrain)
- Create an Anchor in real-world space (either based on hit test or relative to camera)
- Place your content in the anchor
- For Every Frame:
  - Adjust the rendering camera position based on the DeviceTrackable Pose
  - Adjust the transform of the content based on an updated Anchor position
- Deinitialize the Trackers (PositionalDeviceTracker and SmartTerrain)
- Deinitialize Vuforia AR

## ANNEXE 2 : Diagrammes UML du projet

### Diagramme d'utilisation du projet



### Diagramme de Séquence du projet



### ANNEXE 3 : Détail des tâches à effectuer pour le test de rapidité

#### Création du calendrier

Pour Google Agenda :

- Clic gauche pour ajouter un événement
- Nommer l'événement
- Choisir sa date de début et de fin
- Valider
- Recolorer l'événement
- Répéter pour chaque événement à insérer

Pour l'application

- Mettre en ordre les Legos
- Scanner les lignes
- Choisir la date d'insertion sur le site internet
- Valider

#### Mise à jour du calendrier

Pour Google Agenda :

- Supprimer un à un les événements précédents
- Insérer le nouveau calendrier en répétant les étapes de création de calendrier ci-dessus

Pour l'application :

- Mettre en ordre les Legos
- Scanner les lignes
- Choisir la date d'insertion sur le site internet
- Valider
- Confirmer l'écrasement des événements précédents



## ANNEXE 4 : Diagramme de Gantt du projet

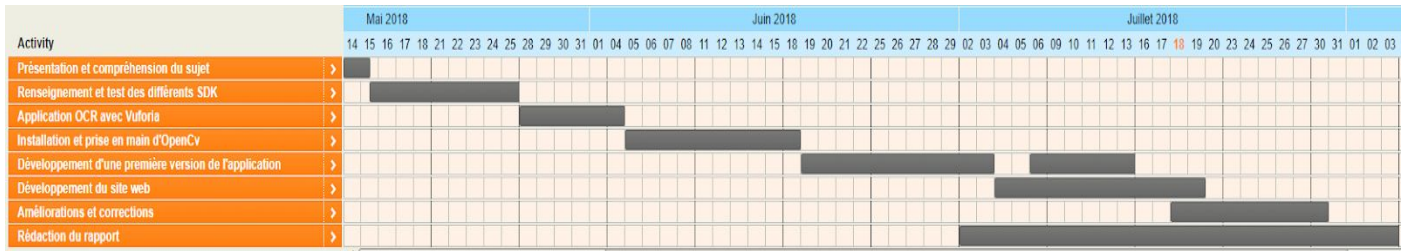


Diagramme Gantt du projet réalisé avec Tomsplanner.fr: <https://www.tomsplanner.fr/>

### Activités de haut en bas :

- Présentation et compréhension du sujet
- Renseignement sur les différents SDK
- Application OCR avec Vuuforia
- Installation et prise en main d'OpenCv
- Développement d'une première version de l'application
- Développement du site web
- Améliorations et correctifs
- Rédaction du rapport

*Etudiant : Lepage Tim*

*Année d'étude : 4<sup>ème</sup> année*

*Entreprise : Universidade Federal do Rio Grande do Sul*

*Adresse complète : Av. Paulo Gama, 110 - Farroupilha,*

*Porto Alegre - RS, 90040-060, Brésil*

*Téléphone : +55 51 3308-6000*

*Tuteur et responsable administratif : Luciana Nedel*

*Téléphone : +55 51 3308-7037*

*Courriel: nedel@inf.ufrgs.br*

*Enseignant-référent : Jean-François Méhaut*

*Téléphone : +33 4 76 61 20 20*

*Courriel : jean-francois.mehaut@univ-grenoble-alpes.fr*

*Titre : Data visualisation avec la réalité augmentée ou comment faciliter la visualisation de données grâce à une application de réalité augmentée.*

*Résumé : La technologie prend de plus en plus de place dans nos vies, et le monde s'informatise de jour en jour. Il en va de même pour la visualisation de données. Ce qui, dans le temps, se représentait avec un boulier l'est aujourd'hui avec un tableur informatique. Cependant les représentations physiques de données restent très populaires de par leur intuitivité de manipulation. Mais comment mêler le physique et le virtuel pour faciliter encore la lecture de données ?*

*C'est pour apporter des éléments de réponse à cette problématique que j'ai effectué mon stage de 4<sup>ème</sup> année RICM au sein de l'Institut d'Informatique, dans l'Universidade Federal do Rio Grande do Sul (UFRGS) à Porto Alegre, au Brésil.*

*J'ai travaillé durant douze semaines pour mieux comprendre la data visualisation afin d'offrir une application permettant de faciliter celle-ci en liant le physique et l'informatique. C'est là qu'intervient la réalité augmentée, parfait intermédiaire entre le réel et le virtuel.*

*Ce rapport fait état du déroulement de ce stage.*