



RÉPUBLIQUE DU BÉNIN  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITÉ D'ABOMEY-CALAVI**

**INSTITUT DE FORMATION ET DE  
RECHERCHE EN INFORMATIQUE**

BP 526 Cotonou Tel : +229 21 14 19 88

<http://www.ifri-uac.net> Courriel : [contact@ifri.uac.bj](mailto:contact@ifri.uac.bj)



# MÉMOIRE

pour l'obtention du

**Diplôme de Licence en Génie Logiciel**

**Présenté par :**

Marie-Parisius Dorian HOUESSO

## **Le jeu Awalé: Interface de Programmation Applicative et Interface en Réalité Augmentée**

**Sous la supervision de :**

Dr Ing Vinansétan Ratheil HOUNDJI

**Maitre de Stage :**

Business Information Security Officer Roger FASSINO

Année Académique : 2016-2017

# Dédicace

A

Mon père **HOUESSOU Anastase Fructueux**

Ma mère **DEGUENON Afiavi Rita**

Mes frères et sœurs

# Remerciements

Je tiens à remercier :

- Le Directeur de l'Institut de Formation et de Recherche en Informatique (IFRI) Monsieur Eugène EZIN pour ses nombreux conseils et pour le temps qu'il n'a jamais manqué de me consacrer pour répondre à mes préoccupations ;
- Le Directeur Adjoint de l'IFRI Monsieur Gaston EDAH pour son accompagnement et assistance tout a long de notre formation ;
- L'administration de l'IFRI ;
- Monsieur Ratheil HOUNDJI qui a cru en moi en me confiant ce thème et m'a aidé a comprendre les algorithmes de base de l'Intelligence Artificielle ;
- Mes professeurs de l'IFRI ;
- Monsieur Roger FASSINO pour son assistance et son soutien ;
- Monsieur Gael AGLIN pour son assistance ;
- Le Staff de ALL IN ONE-BENIN ;
- Le Studio Nobium ;
- A mes camarades et amis.

# Table des matières

Dédicace	i
Remerciements	ii
Glossaire	v
Sigles et Abréviations	viii
Résumé/Abstract	1
Introduction	2
<b>1 Revue de littérature</b>	<b>4</b>
1.1 API : Application Programming Interface ou Interface de Programmation Appli- cative . . . . .	4
1.2 Le jeu de Awalé . . . . .	6
1.3 API et programme existant pour le jeu de Awalé . . . . .	8
1.4 Concept de la Réalité Augmentée (RA) . . . . .	9
<b>2 Matériel et méthodes</b>	<b>12</b>
2.1 Analyse des besoins . . . . .	12
2.2 Matériel . . . . .	12
2.2.1 Matériel intervenant dans la réalisation de l'API . . . . .	13
2.2.2 Matériel intervenant dans la réalisation du jeu en RA . . . . .	15
2.3 Méthodes . . . . .	19
2.3.1 Modélisation . . . . .	20
2.3.2 Développement . . . . .	24
<b>3 Résultats et Discussion</b>	<b>27</b>
3.1 Applications réalisées . . . . .	27
3.2 Création d'agents pour le jeu de Awalé . . . . .	29
3.2.1 Importation de l'API dans un IDE . . . . .	29
3.2.2 Appel d'un agent . . . . .	30

3.3 Discussion . . . . .	31
<b>Conclusion et Perspectives</b>	<b>33</b>
<b>Bibliographie</b>	<b>34</b>
<b>Webographie</b>	<b>35</b>
<b>A Code source Minimax</b>	<b>37</b>
<b>B Expérience</b>	<b>41</b>

## Table des figures

1.1 Type d'APIs Source [10] . . . . .	6
2.1 Logo UML . . . . .	13
2.2 Logo Java . . . . .	14
2.3 Logo IntelliJ Idea . . . . .	14
2.4 Logo JUnit . . . . .	15
2.5 Logo Unity3D . . . . .	16
2.6 Logo CSharp . . . . .	16
2.7 Logo Visual Studio . . . . .	17
2.8 Logo Vuforia . . . . .	17
2.9 Schéma de la Méthode Scrum-Agile Source [38] . . . . .	19
2.10 Diagramme de classe . . . . .	21
2.11 Diagramme de cas d'utilisation . . . . .	22
2.12 Diagramme de séquences : Mode de jeu . . . . .	23
2.13 Diagramme de séquence : Jeu Humain vs IA . . . . .	24
3.1 Interface de Jeu . . . . .	28
3.2 Interface de Jeu . . . . .	29

# Liste des tableaux

3.1	Temps d'exécution de Minimax par rapport à la profondeur de l'arbre . . . . .	31
-----	---	----

# Glossaire

**APK :**

Android Package, Collection de fichiers compressés pour le système d'exploitation Android.

**Cible :**

Image créée pour permettre au traqueur du SDK de Vuforia de reconnaître la scène de jeu

**Framework :**

Cadre de travail en français ; Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel.

**Gameplay :**

Ensemble des éléments d'une « expérience vidéoludique », c'est-à-dire le ressenti du joueur quand il s'adonne au jeu vidéo

**Intelligence Artificielle :**

Capacité qu'a une machine de résoudre les problèmes en faisant recours à l'Intelligence comme le font les humains.

**Intergiciel :**

classe de logiciels systèmes qui permet d'implanter une approche répartie.

**Jeu combinatoire :**

Jeu 1. opposant généralement deux joueurs ou deux équipes ; 2. dans lequel les joueurs ou équipes jouent à tour de rôle ; 3. dont tous les éléments sont connus (jeu à information complète) ; 4. où le hasard n'intervient pas pendant le déroulement du jeu.

**Jeu de société :**

Jeu qui se pratique à plusieurs personnes.

**Jeux à sommes nulles :**

Jeux où la somme des gains des joueurs est nulle

**Microsoft .NET Framework :**

Framework développé par Microsoft incluant une large librairie de classes et permettant une interopérabilité entre plusieurs langages de programmation.

**Mono :**

Plateforme logicielle de développement conçu pour permettre aux développeurs de créer facilement des applications fonctionnant sur n'importe quel plateforme.

**Moteur de jeu :**

Ensemble de composants logiciels effectuant des calculs de géométrie et de physique utilisés dans les jeux vidéo

**Prefabs :**

Type de ressources crée avec Unity pour être des objets de jeu réutilisables autant de fois que possible.

**Slack :**

Outil de collaboration destiné à faciliter la communication et l'échange de données entre les professionnels d'une même entreprise

**Standardisation :**

Procédure qui consiste à développer, à promouvoir et éventuellement à imposer des technologies et des processus fondées sur des normes dans un secteur d'activité donné.



# Sigles et Abréviations

**API** : Application Programming Interface.

**APK** : Android PacKage.

**AR** : Augmented Reality.

**CORBA** : Common Object Request Broker Architecture.

**DFS** : Depth First Search.

**DLL** : Dynamic Link Library.

**IA** : Intelligence Artificielle.

**JDK** : Java Development Kit.

**JSON** : JavaScript Object Notation.

**PCI** : Peripheral Component Interconnect.

**QR Code** : Quick Response Code.

**RA** : Réalité Augmentée.

**REST** : Representational State Transfer.

**RPC** : Remote Procedure Call.

**SDK** : Software Development Kit.

**SOAP** : Simple Object Access Protocol.

**UML** : Unified Modeling Language.

**XML** : eXtensible Markup Language.

# Résumé

Le jeu de Awalé est un jeu de plateau d'origine africaine. Il est aujourd'hui retrouvé dans le monde entier sous différentes formes qui à l'instar des jeux d'échecs, de Go ou d'Othello ont connu plusieurs implémentations sur les terminaux électroniques grâce à l'avènement des Technologies de l'Information et de la Communication. Nous proposons une Interface de programmation applicative (API) à l'intention des développeurs des différentes variantes du jeu de Awalé, afin de rendre facile leur implémentation, ainsi qu'une interface graphique en Réalité Augmentée pour les amateurs du jeu de Awale afin de leur permettre de jouer au jeu directement sur un smartphone. L'implémentation de l'API a été possible grâce à l'utilisation du langage de modélisation UML et du langage de programmation Java. Le jeu en Réalité augmentée quant à lui a été réalisé avec le langage de programmation C#, le moteur de jeu Unity 3D et le SDK de Réalité Augmentée Vuforia. A la fin de la phase de développement, les différents tests effectués montrent qu'aussi bien l'API et le jeu en Réalité Augmentée sont totalement fonctionnels.

**Mots clés :** *Awale , API, JAVA, Intelligence Artificielle, Réalité Augmentée*

# Abstract

The game of Awale is a board game from Africa. It is now played worldwide in various forms. Like the games of chess, Go or Othello it had several implementations on electronic terminals thanks to the advent of the Information and Communication Technology. We propose an Application Programming Interface for developers and programmers of the various variants of Awale in order to make their implementation in general and the one of Artificial Intelligence associated with each one. We also propose a graphical interface in Augmented Reality for lovers of the game of Awale in order to allow them to play the game directly on a smartphone. Implementing the API has been possible thanks to the use of the UML modeling language and the Java programming language. The game in Augmented Reality Was made possible with the C# programming language, Unity 3D game engine and the Vuforia Augmented Reality SDK. At the end of the development phase, the various tests carried out show that both the API and the augmented reality game are totally functional.

**Keywords:** *Awale ,API, JAVA, Artificial Intelligence, Augmented Reality*

# Introduction

Les jeux ont dès le début de l'humanité fait partie intégrante de la vie des hommes. Grâce à l'avènement des sciences informatiques, ils ont connu un nouveau jour et cela a débuté par l'implémentation sur les ordinateurs des jeux de société, jeux de plateaux et des jeux de stratégies quelque soit leur origine.

Au nombre des jeux de plateau et de société, figure le jeu de Awale originaire d'Afrique qui lui aussi a connu plusieurs versions jouables sur les terminaux électroniques et est aussi l'un des jeux africains les plus connus. Le jeu de Awale lui-même existe sous plusieurs formes (Mancala, Oware, Ayo, Wari) selon que l'on passe d'une ethnie d'Afrique à une autre. L'existence de ses multiples variantes rend l'implémentation de ce jeu complexe et occasionne une variation de l'implémentation d'une Intelligence Artificielle pour chacune d'elles. A partir de notre API il serait possible de créer des jeux de Awale qui utiliseront les dernières technologies telles que la Réalité Augmentée pour apporter une nouveauté dans les implémentations du jeu habituellement rencontrées.

Le thème de notre mémoire « Le jeu Awalé : Interface de Programmation Applicative et Interface en Réalité Augmentée » vient donc résoudre ce problème à travers la mise en place d'une API (Application Programming Interface) pour le jeu de Awalé et la réalisation d'un jeu en Réalité Augmentée.

## Objectifs

Notre projet a pour objectifs principaux de :

- Mettre à la disposition de la communauté scientifique, un ensemble de fonctionnalités leur permettant d'implémenter simplement des agents logiciels pour le jeu de Awale ;
- Mettre à la disposition des amateurs du jeu Awalé une interface graphique leur permettant de jouer en utilisant la technologie de la Réalité Augmentée.

L'API à implémenter devra être le plus générique possible. Ce qui permettra de facilement modifier les valeurs suivantes : *la taille du plateau de jeu, le nombre de pions par case au démarrage d'une partie, le temps de jeu par joueur, le sens du jeu.*

---

## Présentation de ALL IN ONE -BENIN

Nous avons effectué notre stage académique à All in One. Nous décrivons dans cette section les principales activités de cette structure.

ALL IN ONE-BENIN est un centre de formation professionnelle en anglais et en informatique. Il est situé au quartier les Cocotiers, Immeuble Jananto rue à droite après Brussels Airlines sur la route de l'aéroport. Il offre des services tels que :

- Des formations en anglais constituées de trois (03) classes (Listening & Speaking, Reading, Writing) ;
- Des préparations aux tests de la langue anglaise tels que le TOEFL, l'IELTS, le GRE et le GMAT ;
- Des formations en anglais pour les enfants de 04 à 17 ans : Kid Program ;
- Des formations en informatique générale (Windows et Suite Office) et en création de site web et d'applications mobile ;
- Le développement d'applications web et mobile ;
- L'audit sécuritaire des systèmes d'informations.

Le choix de cette structure pour le stage a été motivé par le programme Kid Program qui a pour but de permettre aux enfants de 4 à 17 ans d'apprendre la langue anglaise à travers des activités comme les jeux.

En effet le meilleur moyen de vérifier le bon fonctionnement d'une API est de concevoir une application fonctionnelle à partir de celle ci. Dans la phase de déploiement de l'API et pour vérifier son bon fonctionnement, le "Kid Program" a été identifié pour les tests à travers la réalisation d'une interface graphique du jeu de Awalé en Réalité Augmentée.

## Organisation du document

Ce mémoire est constitué de trois principaux chapitres. Le premier chapitre est une revue de littérature où il est présenté ce qu'est une API, le jeu de Awalé et le concept de la Réalité Augmentée. Dans le second chapitre, nous avons présenté le matériel utilisé pour réaliser le travail ainsi que la méthodologie suivie. Enfin le troisième chapitre présente les résultats obtenus et une discussion de ces résultats.

# Revue de littérature

## 1.1 API : Application Programming Interface ou Interface de Programmation Applicative

---

### Définition

On appelle API (Application Programming Interface) un ensemble de routines, de protocoles, et d'outils utilisés pour concevoir des applications. Une API spécifie comment les composants logiciels interagissent. Les API peuvent être aussi utilisées pour créer des composants d'interface graphique. Le but d'une API est de rendre facile la création d'un programme informatique en donnant aux développeurs l'accès à des blocs de code pré-faits et modifiables à souhait. Les APIs proposent aux développeurs la simplification des tâches et la standardisation<sup>1</sup>.

### Types d'API

Il existe aujourd'hui plusieurs types d'API. Cependant elles peuvent être classées en sept (07) catégories [10] :

- Les Services Web : Ce sont des parties d'un logiciel ou d'un système dont l'accès à leurs services se fait via une adresse sur le web. Cette adresse est soit une URL (Uniform Resource Locator) ou une URI (Unified Resource Identifier). Les web services offrent leurs informations dans un format que les autres applications peuvent comprendre et utilisent le protocole HTTP pour échanger les informations avec les applications clientes.  
Les web services les plus connus sont : l'API Facebook , l'API Google Maps [11].
- Les Websockets API : C'est une technologie qui rend possible l'ouverture d'une session de communication interactive entre le navigateur d'un utilisateur et un serveur. Elle permet d'envoyer des messages à un serveur et de recevoir des réponses sans avoir à demander au serveur de répondre. Comme exemple nous avons : Socket.IO et Caddy.

---

<sup>1</sup>**Standardisation** : Procédure qui consiste à développer, à promouvoir et éventuellement à imposer des technologies et des processus fondées sur des normes dans un secteur d'activité donné.

- Les APIs orientées Librairies : Ce sont des APIs dont l'utilisation requiert une référence ou une importation d'un code ou de fichiers binaires. Les fonctions et routines de cette librairie sont utilisées pour créer de nouvelles actions ou échanger des informations.  
Les API JavaScript comme *Google Maps JavaScript API*, *l'API Twilio* en sont de très bons exemples.
- Les APIs orientées Classes : C'est un type spécial des APIs orientées librairie. Les données et fonctionnalités fournies par ce type d'API sont organisées autour des classes comme dans les langages de programmation orientés objets. Chaque classe offre un ensemble discret d'information pouvant être facilement compris.  
Comme exemple l'API JAVA est un ensemble de classes associées à l'environnement de développement (JDK). Le langage Java apporte la syntaxe de programmation basique et les types primitifs (int, float, etc.). Les autres composants comme le type string, les tableaux sont tous apportés par l'API Java. D'autres exemples sont *l'API Android* et *l'API Google Maps Android*.
- Les Fonctions et routines dans un système d'exploitation : Les systèmes d'exploitation comme Windows et Linux offrent plusieurs fonctionnalités qu'on utilise chaque jour sans y penser. Ces systèmes offrent aussi une API qui permettent aux programmes d'interagir avec eux.  
Des exemples sont l'accès aux fichiers système, l'impression de document ou l'affichage des notifications d'erreur.
- Les APIs d'objets distants : Elles sont en général fournies au sein d'un intergiciel<sup>2</sup> (Middleware) pour permettre la communication et les interactions pour des applications réparties. Elles utilisent un protocole de communication par exemple CORBA-Common Object Request Broker Architecture. Ces APIs fonctionnent en implémentant en local une représentation de l'objet distant puis interagissent avec eux. Les interactions réalisées sont alors dupliquées sur l'objet distant via le protocole.
- Les APIs matériels : Elles sont utilisées dans l'adressage des pièces matérielles sur un appareil. Il s'agit par exemple des disques durs, des bus PCI.

Nous avons opté dans le cadre de ce travail pour une API orientée Classes car répondant plus aux exigences du cahier des charges du thème de ce mémoire.

La figure 1.1 ci-dessous présente une classification des types d'APIs.

---

<sup>2</sup>**Intergiciel** : classe de logiciels systèmes qui permet d'implanter une approche répartie.

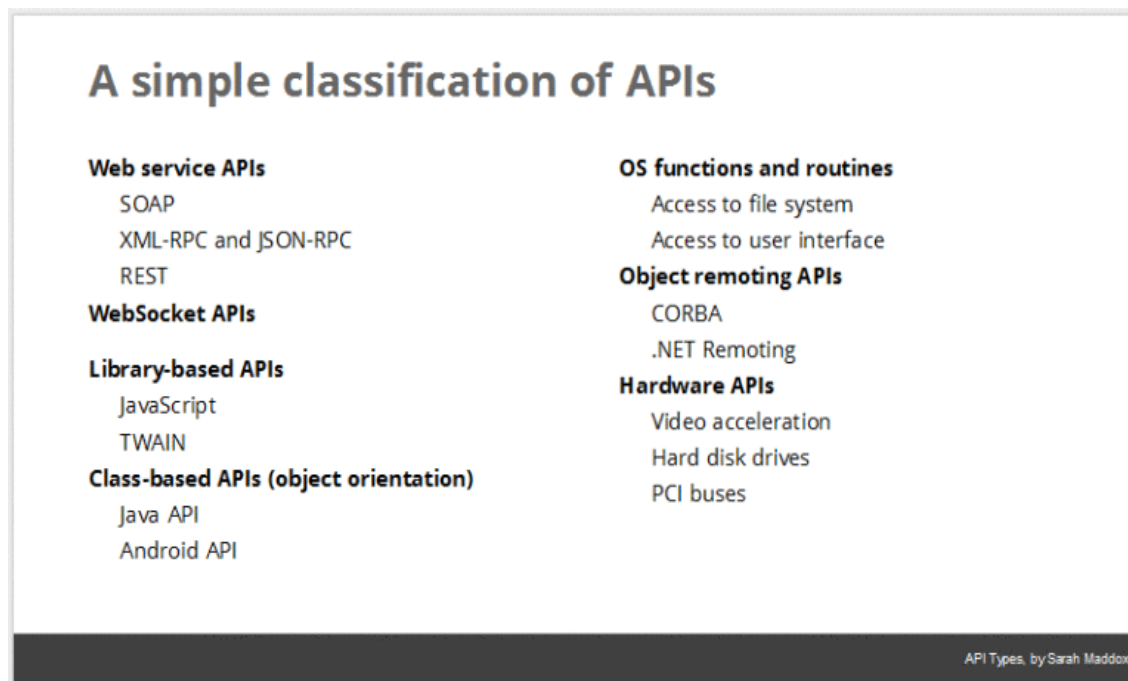


FIGURE 1.1 – Type d’APIs  
Source [10]

## 1.2 Le jeu de Awalé

### Généralité

Le jeu de Awalé [12] est un jeu de société<sup>3</sup> originaire d’Afrique. Son origine exacte semble être l’Ethiopie à partir d’où il s’est répandu en Afrique et a ainsi porté plusieurs noms : *Adji, Awele, Owale, Mancala, Coro Bawo, Ayo, Wari, etc* [12]. C’est un jeu de type “compter et capturer” dans lesquels on distribue des graines dans des trous. Une partie de Awale peut durer environ 20 minutes et requiert des capacités telles que la réflexion, la décision et la stratégie qui le classe au même rang que le jeu d’échecs [12]. Ce jeu se joue à l’aide d’un plateau constitué de deux rangées de six trous. Dans certaines variantes du jeu on peut retrouver deux plus gros trous sur les bords. Les graines jouées dans ce jeu proviennent généralement de l’arbre “*Caesalpinia bonduc*” et sont appelées en langue fon “**Adjikouin**”. Elle peuvent cependant être remplacées par des billes ou des cailloux et doivent être au nombre de quarante-huit (48) pour un Awalé classique. Les jeux se jouent selon des règles bien définies.

<sup>3</sup>Jeu de société : Jeu qui se pratique à plusieurs personnes.

## Règles du jeu

Les principales règles de ce jeu sont :

1. **But du jeu** : Le but est de capturer plus de graines que son adversaire. Le gagnant est celui qui en fin de partie a le plus grand nombre de graines ;
2. **Nombre de joueurs** : Une partie de Awalé est toujours discuté entre deux joueurs situés de part et d'autre du plateau avec une rangée de trous devant chaque joueur ;
3. **Répartition des graines** : Au début d'une partie les quarante-huit graines sont réparties dans les douzes trous à raison de quatre graines par trou ;
4. **Tour de jeu** : Chaque joueur joue de façon alternative. Le premier à démarrer est choisi de commun accord par les joueurs et on choisit un sens de rotation qui vaudra pour toute la partie ;
5. **Jeu** : Le premier joueur prend toute les graines d'un trou de son camp puis il les égrène à raison d'un graine par trou dans toutes les cases qui suivent sur sa rangée puis sur la rangée de son adversaire en suivant le sens de rotation.  
Il est interdit de distribuer des graines dans le trou choisi pour être répartie. En cas d'un tour complet c'est à dire quand on choisit un trou contenant au moins douze graines on saute le trou de départ. Le trou ayant permis de jouer un tel coup est appelé **Kroo** ;

### 6. Capture et capture multiple :

- Capture : Au cours de la distribution si la dernière graine distribuée tombe dans un trou adverse contenant déjà une ou deux autres graines, le joueur ramasse les deux ou trois graines présentes dans ce trou ;
  - Capture multiple : En cas de capture, toujours dans la rangée adverse si le trou précédent, le trou capturé contient aussi deux ou trois graines ces graines sont capturées et ainsi de suite.
7. **Nourrir l'adversaire** : Un joueur ne doit pas affamer son adversaire c'est-à-dire qu'un joueur ne peut jouer un trou qui l'emmenera a capturé toutes les graines présentes sur l'aire de jeu adverse. Si l'adversaire n'a plus de graines il faudra absolument jouer un coup qui lui permettra de rejouer ensuite ;
  8. **Fin de partie** : Une partie de Awalé s'arrête dans les cas suivants :
    - Un joueur a son camp affamé et ne peut être nourri. Dans ce cas le joueur pouvant encore jouer ramasse toutes les graines ;
    - Le jeu est rentré dans une boucle où la configuration du plateau ne change pas. Dans ce cas les graines restantes ne sont pas ramassées par les joueurs ;
    - Un joueur abandonne la partie. Le deuxième joueur ramasse donc toutes les graines restantes ;
    - Un joueur a capturé plus de 24 graines ;



Dans chacun des cas, le joueur ayant capturé le plus de graines a remporté la partie.

Le jeu de Awalé grâce à sa renommée a connu plusieurs applications dans les sciences informatiques et la résolution d'une de ses variantes [13] par 144 processeurs a exploré 889 063 398 406 coups possibles pendant 51h.

## 1.3 API et programme existant pour le jeu de Awalé

---

### Programme existant

Le jeu de Awalé a beaucoup suscité l'intérêt de la communauté scientifique et des développeurs. Cet intérêt est mis en avant notamment dans des domaines tels que l'Intelligence Artificielle. Ainsi le jeu de Awalé a connu l'implémentation de nombreuses de ses multiples variantes. Les éditions du jeu les plus célèbres sont :

- Awalé [14] : Cette version du jeu a été développée par **Myriad Software** et implémente toutes les règles du jeu classique. On peut jouer contre l'ordinateur ou avec un autre joueur humain via une connexion internet. C'est une version de bureau du jeu avec une interface graphique qui propose un gameplay <sup>4</sup> intéressant. Le jeu est à sa quatrième version et peut être installé sur les systèmes Windows. Il est proposé en Français et en Anglais.
- Bot Awalé [15] : C'est un bot développé pour l'application Slack <sup>5</sup> qui permet de jouer au jeu de Awalé directement à l'intérieur de l'application grâce aux commandes. Il a été développé en php avec le framework Symphony et le langage Go pour le côté serveur. Il propose une belle interface tout en étant minimaliste.
- Pyawale [16] : C'est un jeu de awalé développé en Python. Il est jouable dans un terminal grâce aux lignes de commande et propose aussi une interface graphique très simple. Il implémente les règles du jeu de Awalé classique. Sa particularité se retrouve surtout dans son Intelligence Artificielle qui utilise des algorithmes de recherche tels que : le Breath First Search, Minimax, Maximise, NegaMax et AlphaBeta.
- Kissoro Tribal Game [17] : C'est une version très intéressante du jeu développée pour les smartphones tournant sous Android OS. Il propose l'une des variantes du jeu la plus compliquée : à travers un mode histoire et un mode aventure où le joueur affrontera au cours de son avancé une IA de plus en plus forte pour aider le héros qu'il incarne à sauver son royaume. Il propose aussi un mode classique à un ou deux joueurs sur un même téléphone, mais aussi d'affronter d'autres

---

<sup>4</sup>**Gameplay** : Ensemble des éléments d'une « expérience vidéoludique », c'est-à-dire le ressenti du joueur quand il s'adonne au jeu vidéo.

<sup>5</sup>**Slack** : Outil de collaboration destiné à faciliter la communication et l'échange de données entre les professionnels d'une même entreprise.

adversaires en multijoueur ainsi qu'un mode challenge qui oppose les joueurs à une IA pouvant apprendre et ainsi devenir de plus en plus forte. Il a été développé avec le moteur de jeu Unity. Il est disponible sur Google Play Store en téléchargement gratuit.

### API existantes pour le jeu de Awale

A notre connaissance aucune API ou librairie libre d'accès n'est connue à ce jour pour le jeu de Awalé pourtant très développé. Ce qui renforce la raison d'être du présent projet.

## 1.4 Concept de la Réalité Augmentée (RA)

---

### Généralités

La Réalité Augmentée (RA) est une technologie permettant l'ajout de contenu virtuel au monde réel. Elle est souvent associée à l'addition d'un contenu 3D à un environnement vu depuis une caméra. La plus simple des formes de la RA connue est celle des caméras photo. Ces dernières années cette technologie a connu une grande démocratisation grâce aux smartphones composés de plusieurs capteurs et technologies permettant la conception d'application de Réalité Augmentée proposant des expériences riches et variées avec un rendu graphique très élevé.

Cette technologie n'est pas très nouvelle car les premiers articles sur le sujet datent de la fin du vingtième siècle. Elle peut affecter les cinq sens de l'être humain. Les principales utilités de la RA aujourd'hui sont par exemple le scanner des QR Code ou des Codes Barre et le développement de jeu vidéo comme Pokémon Go qui a connu un succès planétaire à sa sortie.

La Réalité Augmentée a aussi de nombreuses applications dans l'architecture, la médecine, le commerce, la presse écrite, etc.

La Réalité Augmentée n'est pas à confondre avec la réalité virtuelle qui elle permet une immersion dans un monde totalement virtuel.

### Formes de la RA

La Réalité Augmentée peut prendre plusieurs formes. Elle dépend toujours d'une forme à l'autre de la technique utilisée pour calculer l'espace 3D relatif à la réalité qui nous entoure. Cependant une de ses formes reste la plus utilisée : le repérage par la méthode de la Vision par l'Ordinateur. Cette forme de la RA peut être divisée en deux sections : le suivi des marqueurs et le suivi sans marqueur.

Dans le cas du suivi des marqueurs, il existe une entité physique appelé *cible* que la vision par ordinateur est formée pour suivre. Cette dernière positionne la perspective de la caméra par rapport à celle-ci. La cible est habituellement traitée en interne pour être l'origine et le centre du monde par rapport auquel la vision par ordinateur peut s'orienter. Parfois, l'alimentation en direct de la caméra est celle qui est considérée comme le centre du monde et la cible ou les cibles sont des objets qui orbitent dans l'espace.

Les techniques de suivi sans marquage sont essentiellement similaires au repérage des marqueurs en

ce sens qu'elles essaient de trouver un point d'origine par rapport auquel on pourra augmenter la réalité. Elles diffèrent dans la manière dont elles trouvent le point d'origine qui, contrairement au repérage des marqueurs, se fait sans utiliser une cible. Dans le suivi sans marquage, la vision par ordinateur est principalement programmée pour suivre certaines des couleurs et des formes avec un certain degré de liberté.

Par exemple, l'ordinateur peut être formé pour suivre les objets vert d'une certaine nuance et les couvrir complètement avec le bleu. Dans ce cas, il suit simplement la couleur. La vision par ordinateur peut même être formée pour reconnaître les visages, tels que toutes les applications de caméra célèbres qui ajoutent des effets animés autour de la tête ou du visage de l'utilisateur. Le suivi sans marquage est certainement plus polyvalent, mais il offre moins de fiabilité que le repérage des marqueurs. De plus, le suivi sans marquage est naturellement plus compliqué à développer, contribuant à la popularité de la technique de suivi du marqueur pour la création d'application en Réalité Augmentée.

Pour ce travail la forme de la Réalité Augmentée utilisée est le suivi du marqueur car parfaitement adapté aux jeux de plateau en RA.

## Applications de la RA

La Réalité Augmentée est aujourd'hui utilisée dans plusieurs domaines de la vie. Dans le domaine de l'art par exemple, l'on retrouve des applications telles que le navigateur Junaio AR [18] qui permet de voir grâce à la caméra d'un smartphone les informations attachées aux différentes oeuvres présentées au Musée Britannique.

C'est aussi le cas pour le monde de la mode où de nombreux catalogues proposent des aperçus de leurs produits en Réalité Augmentée.

Une autre application de la RA est la décoration de l'intérieur. Il est souvent commun de faire l'acquisition d'un meuble ou d'un objet décoratif et qu'au moment de l'installer dans la maison on remarque que cet objet n'est pas du tout assorti avec le design déjà en place. Ainsi des sociétés spécialisées dans la vente de meubles et d'objets de décoration de l'intérieur comme IKEA [19] propose des applications mobiles permettant d'avoir un aperçu de l'aspect qu'aurait la décoration actuelle d'un salon par exemple avec leurs produits installés dans celui-ci.

Les domaines de l'architecture et de l'éducation ne sont pas en reste de cette révolution autour de la RA. Les architectes peuvent aujourd'hui grâce à des services tels que : proposer à leurs clients de faire un tour dans leur futur immeuble à partir d'une maquette réalisée en 3D.

Le domaine qui a récemment connu une vraie expansion de la Réalité augmentée a été celui du jeu sur mobile à travers le jeu Pokémon Go [20] qui utilisait aussi la géolocalisation. Ce jeu a connu un grand succès durant la période de l'été 2016. Le but de ce jeu disponible pour les terminaux mobiles est de collectionner des monstres virtuels appelés pokémon disséminés sur la surface du globe pour les entraîner et les faire combattre tout en se déplaçant. Les déplacements du joueur étant suivi et retranscrits en temps réel dans le jeu grâce à la géolocalisation.

## Outils de la RA

Pour concevoir une application de Réalité Augmentée, on a besoin d'un élément primordial : Une caméra ou une webcam qui servira d'interface de rencontre entre le monde réel et les éléments virtuels.

Les éléments obtenus à travers la caméra sont ensuite analysés, interprétés par un ensemble de logiciels et de capteurs qui y ajoutent des informations. En ce sens les smartphones sont l'outil matériel le plus adapté à l'utilisation d'application de Réalité Augmentée. La logique derrière le fonctionnement de ce type d'application est gérée par des bibliothèques, API ou SDK Spécialement conçu pour aider les développeurs dans le processus de création. On retrouve notamment

Des logiciels open-source tels que :

- **ARToolKit** [21] : Bibliothèque C qui a ensuite été portée pour différentes plateformes et langages comme Android, Flash, Silverlight. Elle est très utilisée dans les projets relatifs à la RA ;
- **DroidAR** [22] : Framework pour la création d'application de RA pour Android basé à la fois sur l'utilisation de la localisation et des marqueurs.

Et des outils propriétaires comme :

- **Le SDK de Réalité Augmentée Vuforia** [35] : Précédemment connu sous l'appellation QCAR, c'est un ensemble d'outils pour la création d'applications de RA pour les terminaux mobiles à l'aide du système de la vision par l'ordinateur pour le suivi de cibles pouvant être unique ou multiples, la reconnaissance d'image et la possibilité d'utiliser une base de donnée dans le cloud. Il supporte les langages C#, C++, et Java ;
- **Wikitude SDK** [23] : Il s'agit d'un ensemble d'outils comme Vuforia mais beaucoup plus basés sur la création d'applications utilisant la géolocalisation pour les plateformes Android, IOS.

# Matériel et méthodes

## 2.1 Analyse des besoins

---

La réalisation de ce projet a nécessité l'utilisation d'un ensemble d'outils et de services qui ont été déterminés après la réalisation du cahier de charges du thème et l'analyse des besoins.

A partir des objectifs visés par le projet et des pré-requis définis dans le cahier de charges, les besoins suivants ont été identifiés :

- L'utilisation de la modélisation orientée objet ;
- L'utilisation d'un langage de programmation orienté objet ;
- La réalisation des tests tels que les tests unitaires sur l'API réalisée ;
- L'utilisation d'une bibliothèque ou d'une API de RA.

## 2.2 Matériel

---

Le matériel à utiliser pour la réalisation du projet se décline en deux ensembles : le matériel pour la réalisation de l'API et celui pour réaliser le jeu en RA.

### 2.2.1 Matériel intervenant dans la réalisation de l'API

Pour réaliser l'interface de programmation nous avons besoin d' :

Un langage de modélisation orientée objet ;

Un langage de programmation orientée objet ;

Un environnement de développement intégré pour l'implémentation de l'API ;

Une librairie pour effectuer les tests unitaires.



FIGURE 2.1 – Logo UML

#### Langage de modélisation

UML (Unified Modeling Language) est [24] un langage de modélisation utilisé en génie logiciel. Il a pour but de proposer un moyen standard de visualiser un système informatique à implémenter avant sa réalisation proprement dite, en mettant en relief les informations qu'utilisera ce système ainsi que les différentes interactions qui interviendront entre ces informations et les différentes parties du système [24]. Dans la même famille d'outils permettant de modéliser un système d'information nous avons la méthode MERISE. Définie comme : Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprises. D'origine française, son but est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. Cependant elle semble plus adaptée à la conception d'un système impliquant l'utilisation d'une base de données. Ce qui n'est pas le cas pour ce projet. L'UML a été choisie pour modéliser cet API car il permet de représenter le système informatique suivant le paradigme orienté objet ce qui donne plus de facilité lors de la phase de développement à l'aide d'un langage orienté objet. Le langage UML permet de schématiser le système d'information et ses composants à l'aide de diagrammes pouvant être classés en trois groupes : les diagrammes de structure ou diagrammes statiques, les diagrammes de comportements et les diagrammes d'interactions ou diagrammes dynamiques. Le diagramme de classe, le diagramme de cas d'utilisation et les diagrammes de séquence présentés dans la partie Modélisation de ce mémoire appartiennent chacun respectivement à l'un des groupes pré-cités et ont été réalisés avec l'environnement de développement intégré MyEclipse en version d'essai. Une fois

ces diagrammes réalisés il fallait maintenant passer à la phase d'implémentation à l'aide d'un langage de programmation orientée objet.



FIGURE 2.2 – Logo Java

### Langage de programmation

Java [25] un langage de programmation orientée objet, basé sur les classes et conçu pour permettre aux développeurs de concevoir des applications ayant un seul code source mais pouvant fonctionner sur n'importe quel ordinateur indépendamment du système d'exploitation qui s'y exécute et de son architecture. C'est un langage interprété et compilé développé par Sun Microsystems mais appartenant actuellement à Oracle Corporation. Il embarque plusieurs bibliothèques sous licence GNU GPL <sup>1</sup> La version de java utilisée pour la réalisation de ce projet est la version 8. Le langage java est multi-paradigme : orienté Objet, structuré, impératif, générique, réflexif et concurrent. Il a été choisi à cause du paradigme orienté objet et de la portabilité qu'il permet mais aussi parce que c'est un langage très utilisé pour la réalisation des jeux en général et du jeu Awalé en particulier. L'écriture du code de l'API a nécessité aussi un environnement de développement intégré (IDE).



FIGURE 2.3 – Logo IntelliJ Idea

### IntelliJ Idea

IntelliJ Idea est l'environnement de développement intégré utilisé pour développer l'API dont ce mémoire est l'objet. La version utilisée pour ce projet est *IntelliJ IDEA 2016.3 EAP Community Edition* permet de l'utiliser sans avoir à payer aucun frais contrairement à la version *Ultimate* payante.[26] C'est l'un des meilleurs IDE utilisé pour l'implémentation des programmes écrits en Java à l'instar de Eclipse et Netbeans. Il a été créé par l'entreprise JetBrains aussi connue sous le nom d' IntelliJ. Son choix a été

<sup>1</sup> GNU General Public License. [27] est une licence utilisée pour les logiciels libres.

déterminé par affinité pour ses capacités d'assistance pendant le codage, l'écosystème des plugins qui y sont associés, son ergonomie et son interface très agréable et facile à utiliser.



FIGURE 2.4 – Logo JUnit

## JUnit

JUnit est un framework<sup>2</sup> pour le langage de programmation Java [28]. Il est utilisé pour effectuer des tests unitaires pendant le développement d'un programmes en Java. Il fait partie de la famille de framework pour les tests unitaires connu sous le nom de **xUnit**. La version utilisée ici est la version 5 sous la licence *Eclipse Public License*. C'est un framework de tests unitaires très utilisé pour les projets Java.

### 2.2.2 Matériel intervenant dans la réalisation du jeu en RA

La réalisation du jeu de Awale en RA a nécessité les outils suivant :

- Un moteur de jeu ;
- Un langage de programmation orientée objet ;
- Un environnement de développement intégré ;
- Un convertisseur de code ;
- Un SDK<sup>3</sup> pour la création des applications de RA.

---

<sup>2</sup>**Framework** : Cadre de travail en français ; Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel. [29]

<sup>3</sup>**SDK** : Outils de développement logiciel permettant la création d'applications pour un système, une plateforme matériel ou une console de jeux vidéo donnée.





FIGURE 2.5 – Logo Unity3D

## Unity

Unity est un moteur de jeu [30]<sup>4</sup> utilisé pour développer des jeux vidéo et des simulations en 2D et en 3D pour les ordinateurs, les consoles de jeu et les terminaux mobiles et ceci sur **27 plateformes** différentes dont : Windows, Linux, IOS, Android, XBOX, PlayStation, Nintendo, Google Cardboard et Oculus Rift et sur les architectures 32 bits, 64 bits et ARM. Il permet de créer des jeux hautement optimisés et graphiquement impressionnants tout en accélérant le processus de développement. Il est proposé sous 4 licences dont trois payantes : les versions *Plus*, *Pro*, *Entreprises* et la licence *Personal* totalement gratuite et qui est celle utilisée pour la réalisation de ce projet et existe sous 6 version ; la version 5.5 étant celle utilisé ici. Il a été lancé le 8 Juin 2005 par **Unity Technologies**. C'est l'un des meilleurs moteurs de jeux au côté de **Unreal Engine**, **Frostbite**, **OpenGL**. Il supporte la création de jeux vidéo en RA et en Réalité Virtuelle d'où son choix. Le moteur de jeu Unity propose trois langages de développement : C#, JavaScript, Boo.



FIGURE 2.6 – Logo CSharp

## C# Langage de programmation

Le choix de ce langage de programmation a été motivé par le fait qu'il est l'un des langages utilisé pour développer des jeux avec Unity et il est aussi un langage de programmation Orienté-Objet. Aussi il y a sur le web une bonne documentation et assez de tutoriels permettant d'apprendre comment créer un jeu avec C# et Unity. Le C# (lire *See Sharp*) est un langage de programmation multi-paradigme comme le langage Java, développé par Microsoft influencé par les langages tels que C++, Java, Visual Basic etc. [32] Dans ce projet il a été utilisé dans sa version 7.0 avec l'environnement de développement intégré Visual Studio Community 2015.

---

<sup>4</sup>**Moteur de jeu [31]** : Ensemble de composants logiciels effectuant des calculs de géométrie et de physique utilisés dans les jeux vidéo.



FIGURE 2.7 – Logo Visual Studio

### Visual Studio Community

Visual Studio Community est la version gratuite de l'IDE par défaut pour les langages de la famille .NET [33] dont fait partie le langage C# précédemment mentionné. Si Unity se charge du côté graphique et des aspects physiques dans la réalisation des jeux vidéos, la mise en place des différentes méthodes et processus intervenant dans ces jeux est réalisée grâce à cet IDE. Tous les scripts y sont écrits. Comme mentionné plus haut, le jeu de Awalé à réaliser devait se baser sur l'API Java principal objectif de ce projet. Ainsi il fallait trouver un moyen de faire passer du code Java au code C#.

### IKVM.NET

Il s'agit d'une implémentation [34] du langage Java pour Mono<sup>5</sup> et du framework Microsoft.NET<sup>6</sup>. Il inclut les composants suivants :

- Une machine virtuel Java implémenté en .NET ;
- Les librairies Java implémenté en C# ;
- Des outils qui permettent l'interopérabilité entre Java et .NET.

IKVM.NET permet d'utiliser un code compilé Java directement avec le Framework Microsoft.NET dont fait partie le langage C#. Il a été développé par Jeroen Frijters et publié en Décembre 2012.



FIGURE 2.8 – Logo Vuforia

### Vuforia SDK

Le SDK de RA Vuforia [35] est un ensemble d'outils qui permettent de créer des applications de RA pour les terminaux mobiles. Il utilise la technologie dénommée *Computer Vision* pour reconnaître et traquer des cibles en temps réel pouvant être des images planes ou des objets en trois dimensions. Il propose

<sup>5</sup>**Mono** : est une plateforme logicielle de développement conçu pour permettre aux développeurs de créer facilement des applications fonctionnant sur n'importe quel plateforme. [36]

<sup>6</sup>**Microsoft.NET Framework** est un framework développé par Microsoft incluant une large librairie de classes et permettant une interopérabilité entre plusieurs langages de programmation.[37]

des APIs pour les langages Java , C++, Objective C++ et les langages de la famille .NET tels que C#, F# , Visual Basic etc. Il supporte la création d'application de RA pour tout les terminaux mobiles et les casques de RA et de réalité virtuelle. Ce SDK peut être utilisé avec Unity, Android Studio, Xcode.

## 2.3 Méthodes

Ce projet a été réalisé en suivant la méthodologie Scrum-Agile. C'est un ensemble de principes de développement logiciel ayant pour but de bien gérer la conception, l'implémentation et la mise en place d'un produit ou d'une solution informatique. Cette méthodologie décrit une stratégie flexible de développement où les développeurs travaillent sur différents modules permettant d'obtenir le produit final. Pour la réalisation d'un projet informatique plusieurs méthodes sont utilisées comme la méthode en cascade ou le cycle en V. La méthode en cascade par exemple consiste à dérouler le projet dans toutes ses phases une fois le besoin du client connu et le cahier des charges établi. Ce qui ne permet pas de faire face efficacement aux imprévus qui peuvent souvent remettre en question tout le projet. L'avantage de la méthode Scrum est qu'elle met au coeur de tout le processus de réalisation l'utilisateur final de la solution en cours de développement et permet de s'adapter plus facilement aux modifications ou changements pouvant survenir pendant de l'exécution du projet. Cette méthode de travail à été adoptée pour ce projet principalement pour ses avantages et aussi car étant celle utilisée par l'équipe de développement à All In One - Benin. Elle s'inscrit dans le schéma organisationnel du centre qui voulait que chaque vendredi au cours de la réunion hebdomadaire chacun de ses membres présente le travail effectué aux niveaux des tâches qui lui avaient été assignées en début de semaine. Cette méthode de travail est expliquée à travers la figure 2.9 :

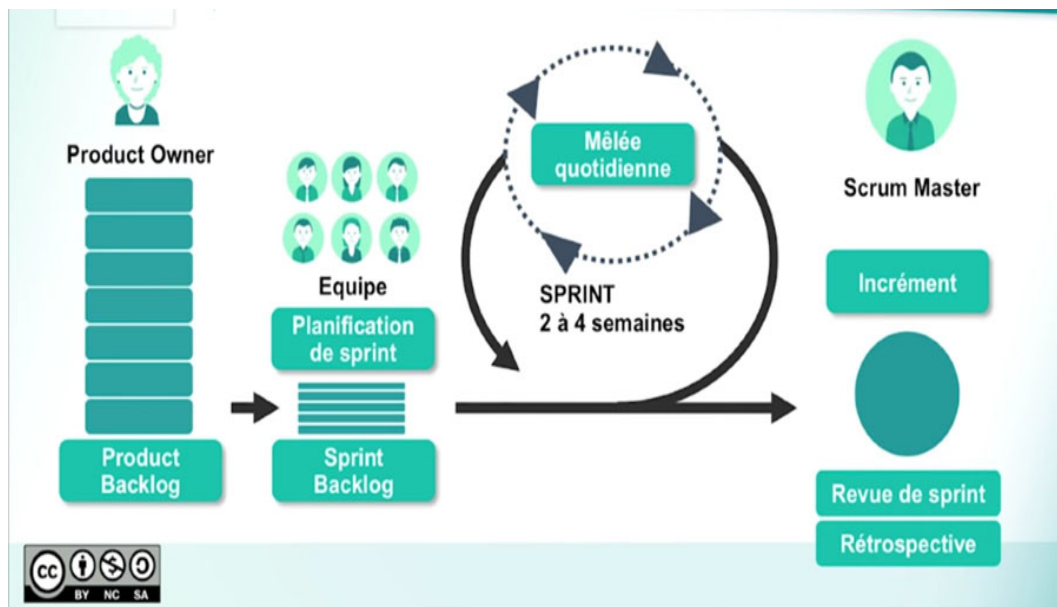


FIGURE 2.9 – Schéma de la Méthode Scrum-Agile  
Source [38]

On retiendra qu'au début de tout processus de développement basé sur la stratégie Scrum-Agile le Product Owner réalise une analyse des besoins basés sur les attentes du client représentant ici le centre où a été effectué le stage académique. A ces attentes on ajoutera aussi dans le cas du présent projet les spécifications prévues par le thème du projet. Une fois ces besoins identifiés vient la détermination du matériel nécessaire à l'exécution du projet préalablement divisé en modules représentant les fonctionnalités du produit final. Un calendrier est donc établi avec une date limite pour la livraison de chaque module. En fonction du temps qui lui a été accordé chaque module est développé et les tests unitaires y sont réalisés (Sprint). Le module terminé il est présenté au chef de Projet et au client pour validation. Un rapport est aussi réalisé à cette étape. Une fois qu'il est validé ce module est mis à part et on passe au développement d'un autre module et cela jusqu'à ce qu'on ait épuisé tous les modules. Ces modules sont donc intégrés l'un avec l'autre et on réalise les tests d'intégrations pour s'assurer du bon fonctionnement du logiciel développé. Ce logiciel est ensuite mis en recette puis en production. Dans le cadre de la réalisation de l'API du jeu de Awalé et celle du jeu en RA, les modules identifiés sont les suivants :

- Modélisation UML de l'API ;
- Implémentation des classes Joueur, Plateau, Awale ;
- Implémentation de l'Intelligence Artificielle ;
- Compilation du fichier Jar final ;
- Installation de l'environnement Unity ;
- Mise en place de l'interface graphique en RA<sup>7</sup> ;
- Implémentation des mécaniques du jeu ;
- Test du jeu ;
- Création d'un site web de présentation de l'API ;
- Mise à disposition de l'API au public ;
- Installation du jeu sur les terminaux mobile de ALL IN ONE-BENIN.

### 2.3.1 Modélisation

La modélisation du système a permis d'avoir les diagrammes suivants :

---

<sup>7</sup>RA : Réalité Augmentée ; RV : Réalité Virtuelle.

## Diagramme de Classe

Le diagramme suivant présente quatre classes : **Board** (pour représenter un plateau de jeu), **Awale** (classe principale), **Player** (représentant un joueur) ; **Artificial\_Player** (classe décrivant le fonctionnement de l'IA). La classe Board regroupe la majorité des méthodes permettant d'implémenter le jeu de Awale, les classes Artificial\_Player, Human\_Player et Player quant à elles représentent les possibles joueurs du jeu qu'ils soient humain ou virtuel. Notons que les classes Artificial\_Player et Human\_Player héritent de la classe Player.

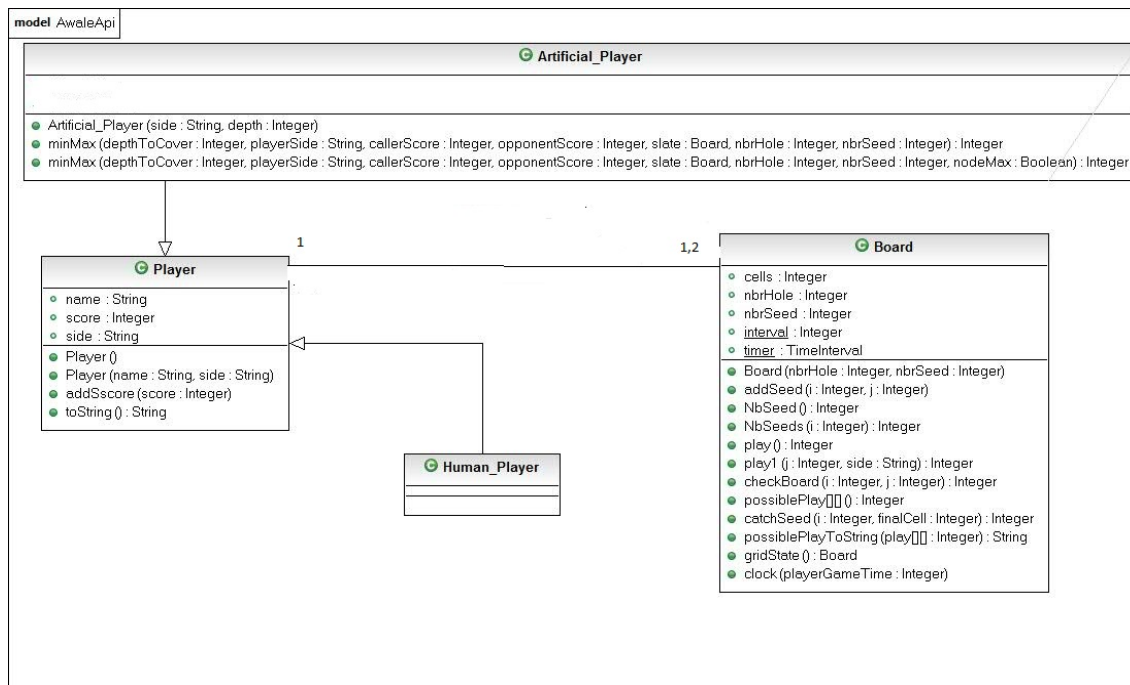


FIGURE 2.10 – Diagramme de classe

Une attention particulière sera portée sur certaines méthodes présentes sur le diagramme de la figure 2.10. Dans la classe **Artificial\_Player**, la méthode **minMax** qui permet de reconnaître l'algorithme de recherche utilisé actuellement dans l'API. Il s'agit de l'algorithme MiniMax qui sera expliqué plus bas. La méthode **Artificial\_Player.minMax()** prend en paramètres la profondeur à parcourir, la position de l'agent sur le plateau de jeu, son score et celui de l'adversaire. Viennent ensuite les méthodes **Board.addSeed()** ; **Board.catchSeed()** ; **Board.play()** ; **Board.play1()** et **Board.possiblePlay()**. La méthode :

- **Board.addSeed()** : permet d'ajouter 1 graine dans un trou lors d'un coup en fonction du trou joué ;
- **Board.catchSeed()** : permet de ramasser les graines gagnées par un joueur lors d'un coup ;
- **Board.play()** : permet de jouer un coup lorsque le sens de jeu choisi est celui d'un jeu de Awale classique (sens inverse des aiguilles d'une montre.) ;
- **Board.play1()** : permet de jouer un coup lorsque le sens de jeu choisi est le sens inverse d'un jeu de Awale classique (sens des aiguilles d'une montre.) ;
- **Board.possiblePlay()** : retourne les trous pouvant être joués par les deux joueurs.

## Diagramme de Cas d'utilisation

Le diagramme de cas d'utilisation décrit ce que les utilisateurs (ici des développeurs ou programmeurs) peuvent faire grâce à l'API Awalé.

Ces derniers pourront utiliser l'API pour créer des variantes du jeu de Awalé ce qui inclura la création d'un plateau de jeu. Ils pourront aussi, s'ils le désirent, créer un ou des agents intelligents pouvant jouer au jeu de Awalé.

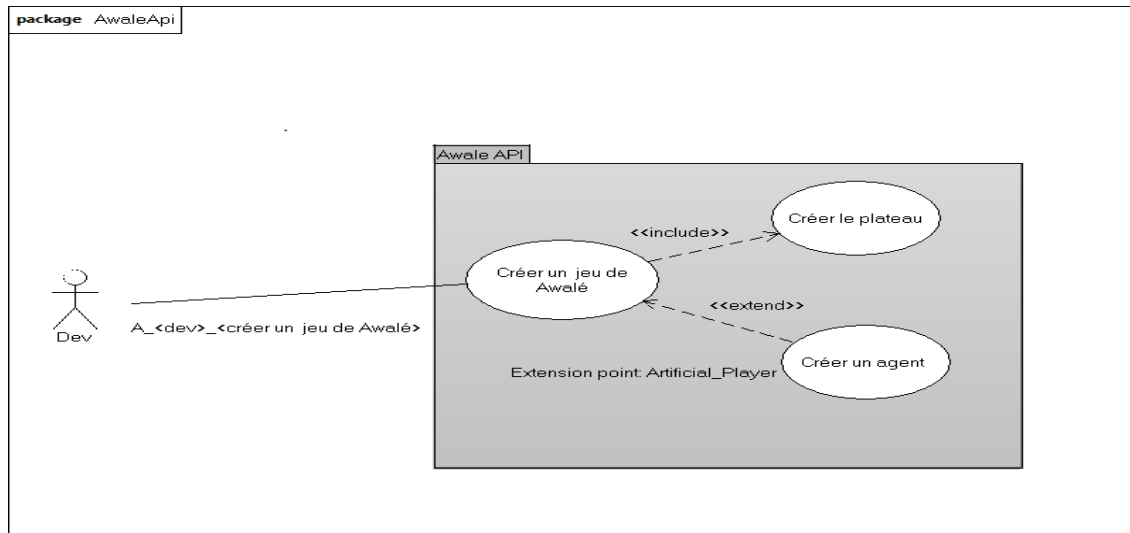


FIGURE 2.11 – Diagramme de cas d'utilisation

## Diagrammes de séquences

Les figures 2.12 et 2.13 présentent 2 diagrammes de séquence illustrant respectivement les différents modes de jeu pris en charge actuellement par l'API et le jeu entre un humain et une IA.

1. Diagramme de séquences illustrant les deux modes de Jeu : **Humain vs Ordinateur, Ordinateur vs Ordinateur**

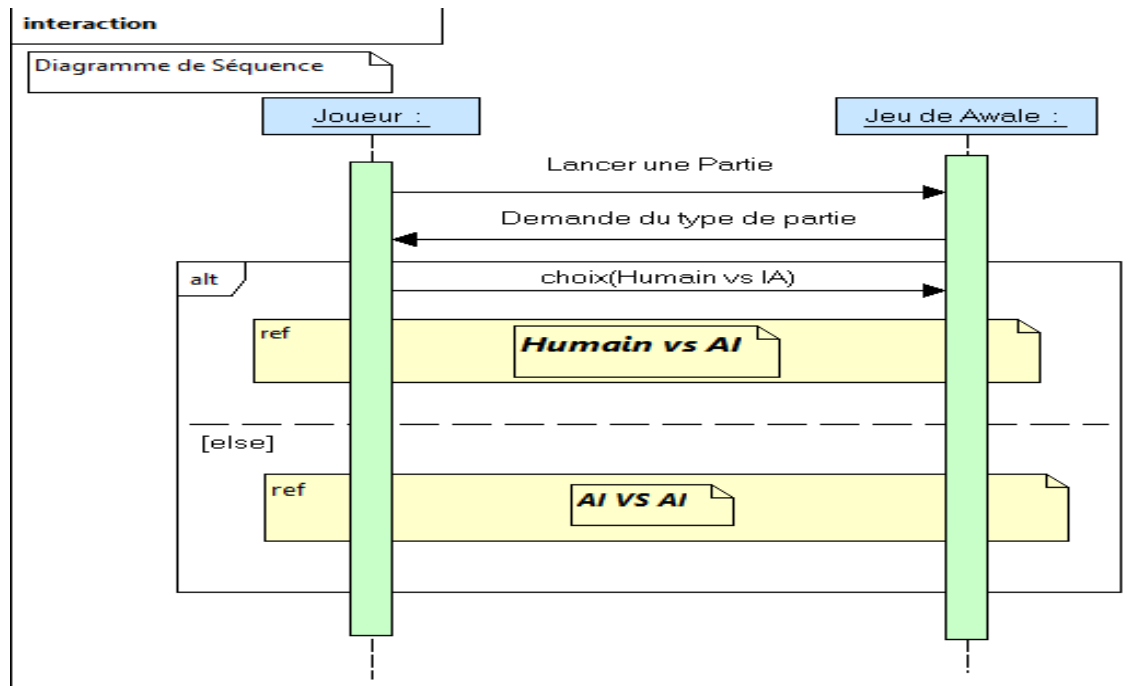


FIGURE 2.12 – Diagramme de séquences : Mode de jeu



## 2. Diagramme de séquences illustrant le Jeu entre un Humain et une IA

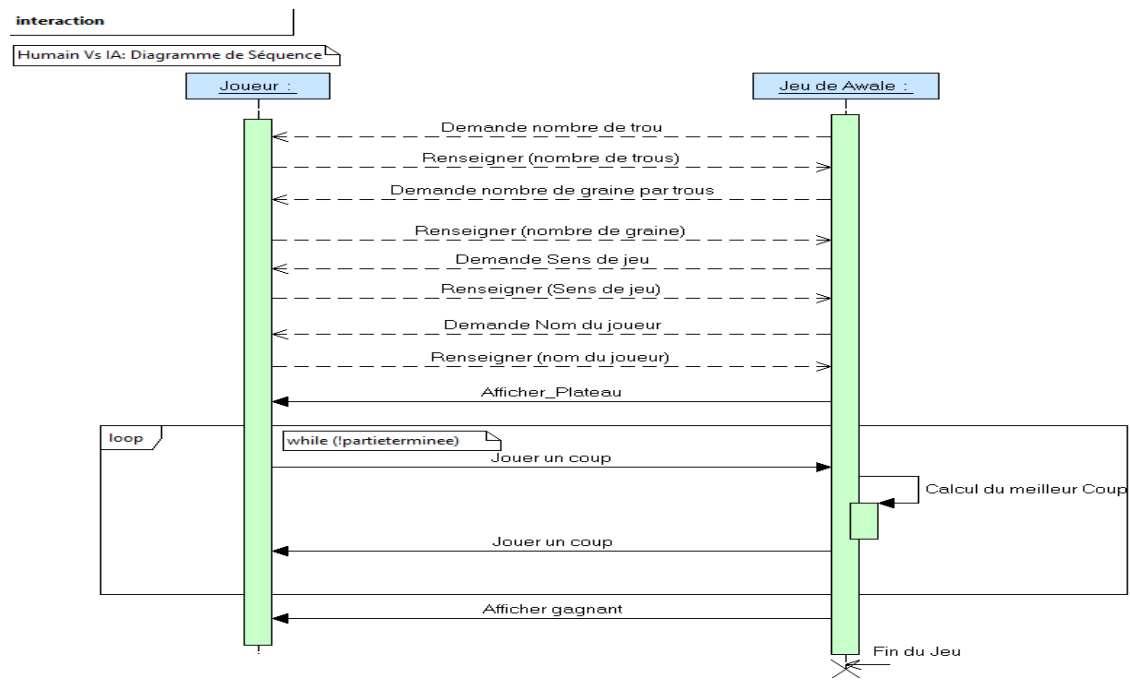


FIGURE 2.13 – Diagramme de séquence : Jeu Humain vs IA

### 2.3.2 Développement

La phase de développement s'est déroulée en deux parties :

1. Développement de l'API;
2. Développement du jeu en RA.

#### 2.3.2.1 Développement de l'API

A l'aide de MyEclips, il a été générée à partir du diagramme de classe les squelettes des différentes classes Java que nous avons ensuite importée dans IntelliJ IDEA. L'API a été nommée : **AwaleApi**. Ces squelettes ont été complétés pour obtenir le code source de l'API qui a été déposé dans un répertoire **Github** disponible sur le lien : <https://github.com/Parisius/API-Awale>. Le développement de cet API a fortement puisé dans les aspects très connus de la programmation en Java tels que : la manipulation des chaînes de caractères, la création et l'utilisation des tableaux, la création des arbres.

L'implémentation de l'algorithme d'Intelligence Artificielle a été la partie la plus importante dans cette partie du projet. L'algorithme implémenté a été l'algorithme **MiniMax**[2]. Cet algorithme utilisé pour la

réalisation d'IA pour les jeux à somme nulle<sup>8</sup> tels que Awale, Tic Tac Toe, les Échecs. Pour implémenter l'algorithme Minimax il faut considérer le problème de jeu comme un problème de recherche dans un arbre allant de la racine (noeud ou état initial) aux feuilles (états finaux). Les niveaux de l'arbre sont répartis entre deux joueurs Min et Max et représentent les gains possibles de ces joueurs en fonction des coups qui peuvent être joués sachant que Max joue en premier. On parcourt de manière récursive l'arbre de jeu en profondeur (DFS<sup>9</sup>) des feuilles à la racine en déterminant grâce à l'état des noeuds feuilles obtenues à partir de l'exécution d'une fonction d'utilité, l'état des noeuds parents. Ensuite à chaque niveau de l'arbre en fonction du joueur qu'il représente on détermine le meilleur coup à jouer sachant que le joueur Max veut maximiser son gain tandis que le joueur Min cherche à minimiser le gain de son adversaire.

Une partie de jeu est donc interprétée comme la distribution d'une récompense au joueur Max.

L'algorithme Minimax :

- retourne une solution si l'arbre est fini ;
- retourne toujours la meilleure solution contre un adversaire qui joue de façon optimale ;
- a une complexité en temps de  $O(b^m)$  avec **b** le nombre maximum de coups à chaque étape du jeu et **m** la profondeur de l'arbre ;
- a une complexité en espace mémoire de  $O(bm)$ .

De ces caractéristiques on remarque que le plus grand inconvénient de MiniMax est le fait qu'il doit parcourir l'arbre de jeu de façon complète. Ce qui peut s'avérer contraignant dans un jeu tel que celui de Awale car l'arbre de jeu dans ce cas est très grand et le parcourir totalement prend de temps. C'est la raison pour laquelle pendant son implémentation nous avons pensé à limiter la profondeur de l'arbre à parcourir en fonction de la difficulté du jeu. Ainsi pour le niveau de difficulté facile, la profondeur est de 3 tandis que pour le niveau difficile la profondeur est de 10. Le code de notre algorithme MiniMax est en Annexe A (page 37).

### 2.3.2.2 Développement du jeu en RA

#### Conversion du fichier Jar en fichier dll

Les classes java ont été compilées pour créer l'API du jeu Awalé. Pour passer au développement du jeu en RA le package java a été converti avec le logiciel IKVM.NET afin d'obtenir un package non plus d'extension *.jar* mais d'extension *.dll*<sup>10</sup>. Pour la conversion, il a fallu exécuter la commande *ikvmc -target :library awaleapi.jar* pour obtenir le fichier *awaleapi.dll* depuis un terminal en se positionnant dans le répertoire où se trouvait *awaleapi.jar*.

<sup>8</sup>**Jeux à somme nulles** : Jeux où la somme des gains des joueurs est nulle

<sup>9</sup>**DFS** :Depth First Search

<sup>10</sup>**Dll** :Dynamic Link library

## Installation de Unity

L'installation de Unity a été très simple. L'installateur de Unity 5.5 a été téléchargé sur le site officiel [unity.com](http://unity.com). Il a été ensuite exécuté et grâce à la connexion internet il a téléchargé tous les composants nécessaires au fonctionnement de Unity ainsi que les projets de démonstration.

## Mise en place de l'interface graphique

Le moteur de jeu Unity propose en son sein plusieurs outils qui permettent la mise en place rapide du rendu graphique d'un jeu. Parmi ces outils figurent les *prefabs*<sup>11</sup>, les *objets construits en 2D ou en 3D* et d'autres éléments comme les *boutons*.

La première action effectuée dans la réalisation du jeu fut la création de notre cible.<sup>12</sup> Cette dernière fut sauvegardé sur un compte développeur sur le site de Vuforia afin de l'importer en tant que paquet utilisable Unity.

Suite à cette action il fut ensuite téléchargé le SDK de Vuforia qui contient plusieurs prefabs comme : la *Caméra RA*, les *boutons virtuels*. Avec ces éléments nous avons conçu l'interface d'un jeu standard de Awalé comportant deux côtés : Nord et Sud avec sur chaque côté 6 cases contenant quatre graines représentées par une valeur numérique. Le plateau en 3D utilisé dans le jeu a été importé du projet open source Mancala Lite réalisé par l'éditeur Nobium Studio. Ce projet est disponible à l'adresse : <http://niobiumstudios.com/project/mancala/>

## Implémentation des mécaniques du jeu

Suite à la mise en place de l'interface graphique, il a été importé dans Unity notre API préalablement converti en fichier dll<sup>13</sup>. Grâce au langage C# les routines chargées du fonctionnement du jeu ont été rapidement implémentées. Le programme a été ensuite compilé depuis Unity au format APK<sup>14</sup> permettant de l'installer sur un smartphone tournant sous Android OS. Enfin il a été effectué plusieurs tests qui ont permis de s'assurer du bon fonctionnement du jeu.

---

<sup>11</sup>**Prefabs** :Type de ressources créer avec Unity pour être des objets de jeu réutilisables autant de fois que possible.

<sup>12</sup>**Cible** :Image crée pour permettre au traqueur de du SDK de vuforia de reconnaître la scène de jeu.

<sup>13</sup>**DLL** :Dynamic-link library

<sup>14</sup>**APK** :Android Package, Collection de fichiers compressés pour le système d'exploitation Android

## Résultats et Discussion

### 3.1 Applications réalisées

---

Au terme de la phase d'implémentation en général, il a été produit une API Java (bibliothèque) du jeu de Awale et un jeu de Awale en RA appelé AR-Awale jouable sur les smartphones Android grâce à un casque de Réalité Virtuelle.

#### **API Awale**

L'API développée à la fin de ce projet est un ensemble de classes java compilées et compressées en fichier d'extension `.jar`. Cette API peut être téléchargée et ajoutée à un projet Java comme n'importe quelle bibliothèque depuis le répertoire github <https://github.com/Parisius/API-Awale>. Un site web de présentation de l'API a été conçu pour permettre aux développeurs d'avoir une documentation sur l'API. Il est à l'adresse [awaleapi.heliohost.org](http://awaleapi.heliohost.org).

### AR-Awale

Les figures 3.1 et 3.2 présentent l'interface du jeu de Awale en Réalité Augmentée. Sur la figure 3.1 nous avons un aperçu de la scène de démarrage du jeu. On peut y voir deux boutons **Play** et **Rules** permettant respectivement de lancer une partie de Awale et de lire les règles du jeu. Au niveau de la figure 3.2 nous remarquons les zones indiquant les graines de chaque joueur, leurs scores respectifs et les boutons virtuels que l'humain utilisera pour interagir avec le jeu. Notons qu'à l'étape actuel, **AR-Awale** ne propose que le mode de jeu Humain vs IA.



FIGURE 3.1 – Interface de Jeu

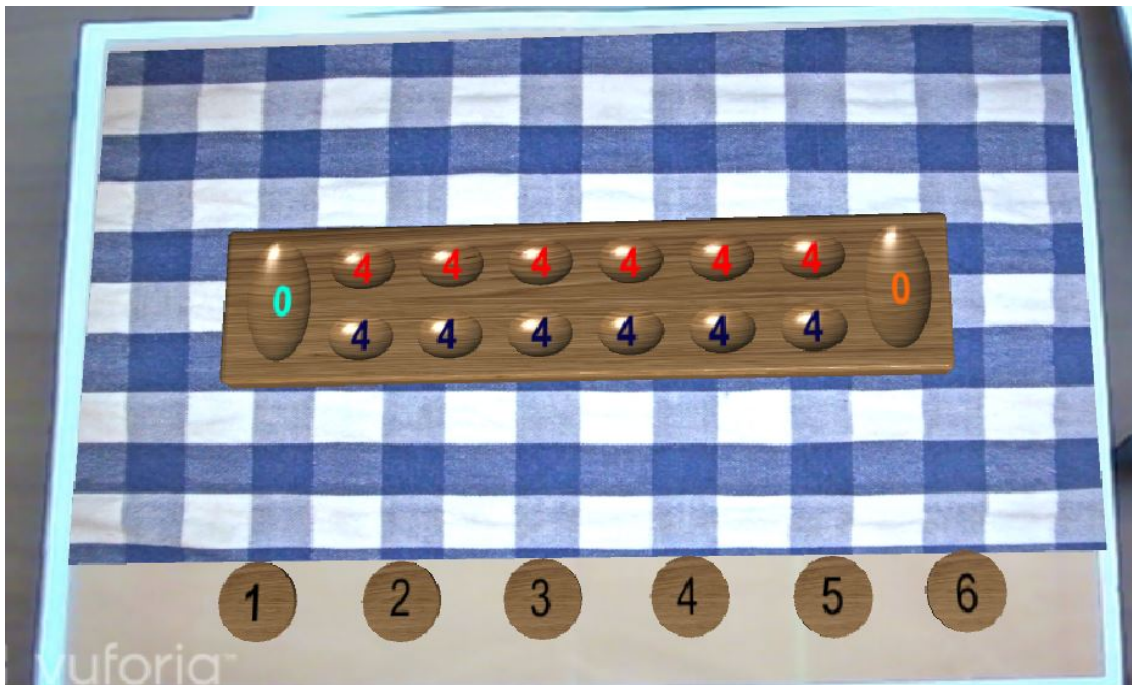


FIGURE 3.2 – Interface de Jeu

## 3.2 Création d'agents pour le jeu de Awalé

---

L'API créée a pour vocation d'être utilisée pour l'implémentation d'agent intelligent pour le jeu de Awalé. Étant développé en Java elle pourra être utilisée directement avec les environnements de développement commun aux langages Java tels que Netbeans, IntelliJ Idea et Eclipse. Avant toute chose il faudra télécharger l'API sur le site officiel [awaleapi.heliohost.org](http://awaleapi.heliohost.org) ou depuis le répertoire Github mentionné ci-dessus.

### 3.2.1 Importation de l'API dans un IDE

#### Cas de Netbeans

Nous présentons ici les étapes à suivre pour ajouter l'API **AwaleAPI** à un projet avec Netbeans.

1. Créer un projet Java ;
2. Faire un clic droit sur le nom du projet et sélectionner l'option "Properties" ;

3. Choisir l'option "*Librairies*" puis cliquer sur le bouton "*Add JAR Folder*";
4. Spécifier le chemin menant à **AwaleAPI.jar** sur votre poste et cliquer sur le bouton "*OK*";
5. Vérifier dans le dossier "*Librairies*" si **AwaleAPI.jar** est affichée. Si oui la bibliothèque a bien été ajoutée à votre projet . Si non reprendre le processus.

### Cas de IntelliJ Idea

Nous présentons ici les étapes à suivre pour ajouter l'API **AwaleAPI** à un projet avec IntelliJ Idea.

1. Créer un projet Java ;
2. Cliquer sur l'option "*File*" depuis le menu puis sélectionner l'option "*Project Structure*";
3. Sélectionner maintenant l'option "*File*" puis cliquer sur l'onglet "*Dependencies*";
4. Sur cette page a votre droite cliquer sur le bouton + en couleur verte puis sur le bouton "*Choose JARs or Directories*";
5. Spécifier le chemin menant à **AwaleAPI.jar** sur votre poste et cliquer sur le bouton "*OK*";
6. Vérifier dans le dossier "*External Librairies*" si **AwaleAPI.jar** est affichée. Si oui la bibliothèque a bien été ajoutée à votre projet . Si non reprendre le processus.

### Cas de Eclipse

Nous présentons ici les étapes à suivre pour ajouter l'API **AwaleAPI** à un projet avec Eclipse.

1. Créer un projet Java ;
2. Faire un clic droit sur le nom du projet et sélectionner l'option "*Properties*";
3. Sélectionner ensuite l'option "*Java Build Path*" puis cliquer sur l'onglet "*Librairies*";
4. Cliquer sur le bouton "*Add External JARs...*";
5. Spécifier le chemin menant à **AwaleAPI.jar** sur votre poste et cliquez sur le bouton "*Open*" puis "*OK*";
6. Vérifier dans le dossier "*Referenced Librairies*" si **AwaleAPI.jar** est affichée. Si oui la bibliothèque a bien été ajoutée à votre projet . Si non reprendre le processus.

## 3.2.2 Appel d'un agent

Un agent intelligent peut être créé à l'aide de l'API en suivant les instructions suivantes :

- Ajouter dans la classe de votre projet la ligne suivante :

```
1 import awaleapi.* ;
```

- Puis comme l'appel de n'importe quelle méthode dans un projet Java entrez :

```
1 Artificial_Player Ordinateur(<<Position sur le Plateau ("P1" et "P2")>>,
2 <<Profondeur en entier>> );
```

L'appel de la classe Awale à travers sa méthode main() est aussi importante car un agent intelligent n'a sa valeur que si il peut jouer une partie de Awale.

### 3.3 Discussion

Au vu des résultats obtenus, nous pouvons dire que les objectifs visés par ce projet ont tous été atteints. En effet nous avons pu créer une API (section 3.1) générique qui donne la possibilité à tout développeur ou programmeur de créer facilement des agents intelligents (section 3.2.2) pour le jeu de Awale ainsi que des plateaux de jeu qui diffèrent les uns des autres par leurs tailles, le nombre de graines à jouer et la durée de jeu par joueur. Aussi grâce au jeu "AR-Awale" donnons nous la possibilité aux joueurs d'avoir une interface graphique pour jouer à Awale. Celle-ci diffère des interfaces habituelles du jeu de Awale car ce jeu est en Réalité Augmentée.

L'API créée à la fin de ce projet est orientée classes et peut être facilement incluse dans tout projet Java qui a pour but de simuler un jeu de Awale. Elle n'impose aucun standard de programmation particulier mais propose une grande modularité dans le processus de développement. Au niveau de l'algorithme utilisé pour l'implémentation d'un agent intelligent notre API connaît cependant un retard par rapport aux programmes existant. Notre API n'implémente que l'algorithme "**Minimax**" tandis que les programmes cités dans la revue de littérature de ce mémoire fonctionne presque tous avec l'algorithme "**Alpha-Bêta**". Pour donc améliorer les résultats de Minimax dans notre cas nous avons pensé à diminuer la profondeur de l'arbre de recherche en la fixant en fonction du niveau de difficulté choisi par le joueur. Le tableau suivant montre le temps d'exécution de notre algorithme par rapport à la profondeur choisie sur un ordinateur possédant un processeur fonctionnant à 2GHz avec 4Go de mémoire RAM.

Profondeur	5	7	10	15
Temps d'exécution	0.01s	0.3s	60s	+10h

TABLE 3.1 – Temps d'exécution de Minimax par rapport à la profondeur de l'arbre

Avec de tels résultats on peut conclure donc qu'au delà de la profondeur 10 l'algorithme "Minimax" implémenté dans notre API prendra un long temps d'exécution ce qui pourrait avoir un impact négatif sur l'expérience utilisateurs des futurs jeux que l'API Awale permettrait de concevoir. L'idéal serait donc de remplacer l'algorithme "Minimax" par un algorithme moins gourmand en temps d'exécution comme Alpha-Bêta avec des heuristiques intelligentes.



En ce qui concerne le matériel utilisé pour réaliser l'API et le jeu en Réalité Augmentée, on constate une longue liste d'outils et une redondance dans la nature de certains d'entre eux . Par exemple nous avons utilisé deux IDE : IntelliJ Idea et Visual Studio Community 2015 et deux langages de programmation C# et Java et dans chacun de ces cas on aurait pu réduire les outils. Bien que les raisons de leur utilisation aient été expliqués plus haut (section 2.2.1), nous aimerions mentionner qu'ils n'étaient pas les seuls adaptés à la réalisation du projet. Nous aurions pu travailler seulement avec le langage Java pour l'API et le jeu de Réalité Augmentée en utilisant cette fois un seul IDE : IntelliJ Idea et à la place du moteur de jeu Unity nous aurions choisi le moteur "**Unreal Engine**" qui lui utilise pour langage de programmation le Java. Cela aurait cependant nécessité un temps d'apprentissage supplémentaire.

L'exécution de ce projet à suivi la méthode Scrum-Agile. Cette méthode s'est révélée très bénéfique pour nous car elle nous a permis d'évoluer de manière graduelle dans la réalisation des différentes tâches composants ce projet.

Enfin, les solutions présentées ici ne se présentent pas comme les solutions parfaites pour la résolution de la problématique de ce projet. Des améliorations de l'API et du jeu en RA sont encore possibles.

# Conclusion et Perspectives

Le but de notre travail était de proposer une interface de programmation applicative pour faciliter le développement du jeu Awale ainsi qu'une interface graphique en Réalité Augmentée pour permettre aux amateurs de ce jeu d'expérimenter une toute nouvelle manière d'y jouer. Ces deux objectifs ont été atteints à travers la création d'une API orienté classes en Java et la création d'un jeu de Awalé Classique en Réalité Augmentée jouable sur les smartphones à l'aide d'un casque de Réalité Virtuelle.

Cependant le travail réalisé pourrait être amélioré. Il pourrait être encore beaucoup plus approfondi à travers l'implémentation d'un agent intelligent dont les temps de réponses pourraient être meilleurs ceux proposés ici grâce à l'algorithme Alpha-Bêta et la généralisation du fonctionnement de l'agent actuel aux autres jeux de type awale en lui permettant de prendre en compte les règles et particularités propres à chacun de ces jeux. D'autre perspectives seraient la création d'agents plus performants au jeu de Awale en utilisant les techniques d'apprentissage du Machine learning et l'ouverture de cet API aux langages de programmation les plus utilisés pour la conception d'agents intelligents tel que *Python* ou *Scala*. L'ajout d'un mode multijoueur en réseau au jeu en Réalité Augmentée et la réalisation du jeu grâce au moteur de jeu Unreal Engine sont également des perspectives intéressantes.

# Bibliographie

- [1] Dominic Cushnan, Hassan El Habbak, Septembre 2013 : *Developing AR Games for IOS and Android*, Published by Packt Publishing, 117 p.
- [2] Crina Grosan and Ajith Abraham, 2011 : *Intelligent Systems A Modern Approach*, Published by Springer, 465 p.
- [3] Stuart Russell, Peter Norvig, 2011 : *Artificial Intelligence A Modern Approach*, Published by PRENTICE HALL, 1112 p.
- [4] Brian Mulloy : *Web API Design Crafting Interfaces that Developers Love*, Published by Apigee, 38 p.
- [5] James Rumbaugh, Ivar Jacobson, Grady Booch, Juillet 2004 : *The Unified Modeling Language Reference Manual Second Edition*, Published by Addison-Wesley, 742 p.
- [6] Herbert Schildt, Juillet 2007 : *Java The Complete Reference, Seventh Edition*, Published by McGraw-Hill Companies, 1057 p.
- [7] Svetlin Nakov & Co. , 2013 : *Fundamentals of Computer Programming with C#*, 1122 p.
- [8] Edd Wilder-James, Niel M. Bornstein, Juillet 2004 : *Mono : A Developer's Notebook*, Published by O'Reilly Media, 304 p.
- [9] India Community, Decembre 2004 : *.NET Tutorial for Beginners* , Published by O'Reilly Media, 224 p.

# Webographie

- [10] API Types, Sarah Maddox *Simple Classification of APIs*, <https://www.slideshare.net/sarahmaddox/api-types> Consulté le 23 Juin.
- [11] <https://webdam.com/blog/top-10-web-apis-bridging-todays-technology> Consulté le 21 Août.
- [12] Awalé *Généralité et règles*, <http://www.myriad-online.com/resources/docs/awale/english/background.html> Consulté le 04 Juin.
- [13] Article *L'informatique au lycée, Chapitre 10 : Intelligence Artificielle et jeux*, <http://ow.ly/5iIy8> Consulté le 04 Juin.
- [14] Jeu Awalé : <http://awale.fr.downloadastro.com/> Consulté le 10 Avril.
- [15] Awalé pour Slack : <https://marmelab.com/blog/2017/03/15/awale-slack.html> Consulté le 10 Avril.
- [16] PyAwale : <http://freecode.com/projects/pyawale> Consulté le 10 Avril.
- [17] Kissoro Tribal Game : <http://masseka-game-studio.com/\https://fr.ulule.com/kissoro-tribal-game/> Consulté le 14 Juillet.
- [18] Augmented Reality - Using Junaio in the British Museum : <http://elearndev.blogspot.rs/2011/08/augmented-reality-using-junaio-in.html> Consulté le 12 Septembre.
- [19] Ikea launches augmented reality app lets you preview digital furniture in your physical house : <http://www.digitalmeetsculture.net/article/ikea-launches-augmented-reality-app-lets-you-preview-digital-furniture-in-your-physical-house/> Consulté le 12 Septembre.
- [20] Site officiel Pokemon Go : <http://www.pokemongo.com/> Consulté le 12 Septembre.
- [21] Site officiel de ARToolKit : <https://www.artoolkit.org/> Consulté le 4 Juillet 2017.
- [22] Site officiel de DroidAR : <https://bitstars.github.io/droidar/> Consulté le 4 Juillet 2017.
- [23] Site officiel de Wikitude : <http://www.wikitude.com/> Consulté le 4 Juillet 2017.
- [24] Unified Modeling Language Site Web officiel : 14 Juillet <http://http://www.uml.org/what-is-uml.html> Consulté le 04 Juin.
- [25] Java SoftwareProgramming Language Site Web officiel : Oracle <https://www.oracle.com/java/index.html> Consulté le 04 Juin.
- [26] IntelliJ IDEA Site Web officiel jetbrains s.r.o : <https://www.jetbrains.com/idea/> Consulté le 04 Juin.

- 
- [27] GNU General Public License *Site Web officiel* : <https://www.gnu.org/licenses/gpl-3.0.en.html> 17 Juillet 2017 Consulté le 04 Juin.
- [28] JUnit Framework by JUnit *Site Web officiel* : <http://junit.org/junit4/> Consulté le 04 Juin.
- [29] Framework Auteur David biologieek *Définition* : <https://larlet.fr/david/biologieek/archives/20070117-definition-et-avantages-d-un-framework-web/> Consulté le 10 Juin.
- [30] Unity by Unity *Site Web officiel* : [unity3d.com](http://unity3d.com) Consulté le 04 Juin.
- [31] Game Engine *Définition* : by michael enger [www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/](http://www.giantbomb.com/profile/michaelenger/blog/game-engines-how-do-they-work/101529/) Consulté le 04 Juin.
- [32] C# langage de programmation *Site Web* : Auteur Microsoft [www.visualstudio.com](http://www.visualstudio.com) Consulté le 04 Juin.
- [33] Visual Studio Auteur Microsoft *Site Web* : <https://www.visualstudio.com/fr/vs/community/> Consulté le 04 Juin.
- [34] IKVM Jeroen Frijters *Site Web* : <http://www.ikvm.net/> Consulté le 04 Juin.
- [35] Vuforia Augmented Reality SDK Vuforia *Site Web* : <https://www.vuforia.com/> <https://developer.vuforia.com/> Consulté le 4 Juillet 2017.
- [36] Mono *Site Web* : Mono Project auteur <http://www.mono-project.com/> Consulté le 04 Juin.
- [37] Microsoft .NET Framework *Site Web* : <https://www.microsoft.com/net> Consulté le 04 Juin.
- [38] Moog : Gestion de projet // Gestion de projet agile avec Scrum Scrum Picture // Cours et MOOC de gestion de projet *Site Web* : <https://gestiondeprojet.pm/category/les-cours/scrums> Consulté le 04 Juin.

## Code source Minimax

```
1 public class ArtificialPlayer extends Player
2 {
3     /*Variable
4     * difficulty: set the difficulty. Can be Easy, Beginner, Normal, Hard;
5     */
6     private int difficulty;
7     private int nbrHole;
8     private int nbrSeed;
9     private int allSeed ;
10    private int miHole ;
11    private int miSeed ;
12
13    /*Constructor*/
14    public ArtificialPlayer() {
15        super();
16        // TODO Auto-generated constructor stub
17    }
18
19    public ArtificialPlayer(int difficulty,int nbrHole,int nbrSeed)
20    {
21        super("Computer");
22        this.difficulty = difficulty;
23        this.nbrHole = nbrHole;
24        this.nbrSeed = nbrSeed;
25        miHole = nbrHole/2;
26        miSeed = allSeed/2;
27        allSeed = nbrSeed*nbrHole;
28        // TODO Auto-generated constructor stub
29    }
30
31    public int getDifficulty() {
32        return difficulty;
33    }
34
35    public void setDifficulty(int difficulty) {
36        this.difficulty = difficulty;
37    }
38
39    public int getNbrHole()
40    {
41        return nbrHole;
```

```

42     }
43
44     public void setNbrHole(int nbrHole)
45     {
46         this.nbrHole = nbrHole;
47     }
48
49     public int getNbrSeed() {
50         return nbrSeed;
51     }
52
53     public void setNbrSeed(int nbrSeed) {
54         this.nbrSeed = nbrSeed;
55     }
56
57
58
59     /*-----Methods-----*/
60
61     /*
62     * minMax() function
63     * return: a play judge to be interesting for the caller, but during
64     the recursive call this method
65     * return the values of leafs and nodes the algorithm choose.
66     */
67     public int minMax(int depthToBeCovered, int playerNum,
68     int callerScore, int opponentScore, Grid grid)
69     {
70         return minMax(depthToBeCovered, playerNum, callerScore,
71         opponentScore, grid, true);
72     }
73
74     private int minMax(int depthToBeCovered, int playerNum, int callerScore,
75     int opponentScore, Grid grid, boolean nodeMax)
76     {
77         int bestNodeValue, choice = -1, seedNumberBeforePlay, seedNumberAfterPlay,
78         nodeValue;
79         Grid gridCopy;
80
81
82         //-----Evaluation Functions-----
83         if (depthToBeCovered == 0) // if max depth is reached
84             return callerScore - opponentScore; //return the leaf value
85         if (callerScore > miSeed) //if caller score is > half number of seed
86             return 1000; // return a maximum value
87         if (opponentScore > miSeed) //if opponent score is > half number of seed
88             return -1000; //return a minimal value
89         //-----
90
91         if (nodeMax) { //nodeMAX: simulate the play the caller can produce
92
93             bestNodeValue = -1000; // assign the best value to the minimum
94
95             // browse the hole possible for the caller i.e nbrHole/2
96             // playerNum varied between 0 et 1, 0 for player 1 and 1 for player 2.
97             for (int i = 0 + (miHole * playerNum); i < miHole + (miHole * playerNum); i++) {
98
99                 //if the hole can be played (the hole contains at least one seed) then
100                 if (grid.testValid(i)) {

```

```

101 //make a copy of actual grid
102 gridCopy = grid.gridState();
103 //control the number of seed before this play
104 seedNumberBeforePlay = gridCopy.amountOfSeedInGrid();
105
106 //choose to play the actual hole
107 gridCopy.pickedUp(i, 1 + playerNum);
108 //control seed's number after the play
109 seedNumberAfterPlay = gridCopy.amountOfSeedInGrid();
110
111 //calculation of caller's new score
112 callerScore += seedNumberBeforePlay - seedNumberAfterPlay;
113
114 /* recursive call of minMax function
115  * will allow us to know the actual node's value
116  * then we send the new parameters:
117  * depth to browse reduction
118  * change for node MAX to node MIN
119  * new caller's score
120  * opponentScore( don't vary during this call)
121  * state of new Grid after this play
122  */
123 nodeValue = minMax(depthToBeCovered - 1, playerNum, callerScore,
124 opponentScore, gridCopy, false);
125
126 //if this value is superior than the value of the best node then
127 if (nodeValue > bestNodeValue){
128 //replace the best value of the node by actual node's value
129 bestNodeValue = nodeValue;
130
131 //the choice become the one of the played hole
132
133 choice = i;
134 }
135 }
136 }
137 }
138 /*
139  * if all the recursive call were played
140  * then return the final choice
141  */
142 if (depthToBeCovered == this.difficulty)
143 return choice;
144 }
145
146 else{ //nodeMIN: simulate the play opponent can produce
147
148 bestNodeValue = 1000;
149 // browse the hole possible for the opponent i.e nbrHole/2
150 for(int i = miHole - (miHole * playerNum); i < nbrHole - (miHole * playerNum);
151 i++){
152 //if the hole can be play then
153 if(grid.testValid(i)){
154
155 //make a copy of actual grid
156 gridCopy = grid.gridState();
157 //control the number of seed before this play
158 seedNumberBeforePlay = gridCopy.amountOfSeedInGrid();
159

```



---

```

160 //choose to play the actual hole
161 gridCopy.pickedUp(i, 2 - playerNum);
162 //control seed's number after the play
163 seedNumberAfterPlay = gridCopy.amountOfSeedInGrid();
164
165 //calculation of opponent's new score
166 opponentScore += seedNumberBeforePlay - seedNumberAfterPlay;
167
168 /* recursive call of minMax function
169  * will allow us to know the actual node's value
170  * then we send the new parameters:
171  * depth to browse reduction
172  * change for node MIN to node MAX
173  * new opponent's score
174  * state of new Grid after this play
175 */
176 nodeValue = minMax(depthToBeCovered - 1, playerNum, callerScore,
177 opponentScore, gridCopy, true);
178
179 //if the value < best node value then
180 if (nodeValue < bestNodeValue){
181     //replace the best value of the node by actual node's value
182     bestNodeValue = nodeValue;
183 }
184 }
185 }
186
187 }
188 return bestNodeValue; // Return the best node value
189 }
190
191 }

```

## Expérience

Le projet faisant l'objet du présent mémoire à été riche tant en apprentissage qu'en approfondissement des notions acquises durant les trois années de formations à l'IFRI. Il a permis de revisiter les notions de base de l'algorithmique et de la programmation Orienté-Objet à travers l'utilisation des chaînes de caractères, les tableaux, les structures de données tels que les arbres et la complexité algorithmique, l'utilisation des classes et la modélisation de différents diagrammes UML. Ce fut aussi l'occasion de la pratique de la recherche personnelle car certaines notions telles que l'Intelligence Artificielle, les concepts de la Réalité Virtuelle et la Réalité Augmentée ne faisant pas partie de la formation reçue ont été découvert et exploité. L'utilisation de la méthode Agile-Scrum a aussi été l'un des grands acquis de l'exécution de ce projet.

---