Awale Game: Application Programming Interface and Augmented Reality Interface

Marie-Parisius Dorian Houessou[1,2], Vinasetan Ratheil Houndji[1,2(✉)],
Eugene C. Ezin[1], Manhougbé Probus A. F. Kiki[2,3], Harold Silvere Kiossou[2,3],
Jean-Baptiste Maureen Sossou[2,3], and Faizath Jedida Zoumarou Walis[2,3]

[1] Institut de Formation et de Recherche en Informatique (IFRI), UAC,
Abomey-Calavi, Benin
`parisius.houessou@gmail.com`, `ratheil.houndji@uac.bj`
[2] Machine Intelligence For You SARL, Abomey-Calavi, Benin
[3] Ecole Polytechnique d'Abomey-Calavi (EPAC), UAC, Abomey-Calavi, Benin

Abstract. Awale game is one of the famous board games from Africa with many variants and is now played worldwide in various forms. In this paper, we propose an open-source Application Programming Interface (`API`) for developers to allow an easy implementation of the various variants of Awale as well as artificial intelligence based players. The `API` is available online at https://github.com/Machine-Intelligence-For-You/Awale. Based on this `API`, we propose a PC Awale game, a mobile Awale game, and an Augmented Reality Game. The Awale `API`, PC game, and mobile game are implemented in the programming language `Java` while the game in Augmented Reality is realized with the `C#` programming language, Unity 3D game engine and the `Vuforia` Augmented Reality `SDK`. The various tests carried out show that the `API` and the different games are totally functional. This `API` was also used for the first edition of MAIC, an Artificial Intelligence contest https://mify-ai.com/maic2017/.

Keywords: Awale · `API` · Augmented Reality · Artificial Intelligence Board game

1 Introduction

It is well known that games are an integral part of the human life. Among the board and society games, Awale is one of the most famous African games. It has several playable versions on electronic terminals. Awale itself exists in several forms (Mancala, Oware, Ayo, Wari, etc.) according to specific ethnic groups in which it is found. The existence of its different variants makes hard the implementation of this game and causes a variation of the implementation of an Artificial Intelligence (AI) for each of them. However, to the best of our knowledge, there is no generic `API` Awale Game to ease the implementation of

the different variants of the game. Here we develop an `API` to ease this game implementation such that it would be possible to create various kinds of Awale games, which can use the latest technologies such as Augmented Reality. This paper proposes an open-source Application Programming Interface (`API`) for developers to allow an easy implementation of the various variants of Awale as well as artificial intelligence based players. Our goal is to make available to the scientific community a set of functionalities to easily implement software agents for the Awale game. The `API` is very generic and allows the modification of several values such as the game board's size, the number of holes, the number of seeds per hole when starting the game, the time of play per player, and the game's direction. To ensure that the `API` is functional, we develop (based on it):

– a graphical PC Awale game;
– a mobile Awale game;
– an Augmented Reality (`AR`) based on the mobile Awale game.

The paper is organized as follows: Sect. 2 gives a background of Awale game, `API` and `AR` technology; Sect. 3 describes the different tools used; Sect. 4 presents the class diagram; Sect. 5 shows some graphical interfaces obtained; and Sect. 6 concludes.

2   Background

Awale game is a board game from Africa. It seems to come exactly from Ethiopia and spread to the whole African continent [7]. It's a "count and capture" type game in which you distribute seeds in holes. Awale game requires abilities such as reflection, decision and strategy that ranks it at the same level as chess. A classical Awale game is played using a board containing two rows of six holes. In some variants of the game, one can find two bigger holes on the edges. The seeds played in this game usually come from the tree "Caesalpinia bonduc" and are called in the Fongbe language "Adjikouin". They can however, be replaced by balls or pebbles and must be forty-eight for a classic Awale. Games are played according to well-defined rules. Many video games exist today on several terminals in the Awale Game especially on smartphones and are available on the different mobile apps downloading platforms like PlayStore or AppStore. Below we define `API` and `AR` which are the two main contributions about Awale game in this paper.

`API` stands for Application Programming Interface and is a set of routines, protocols, and tools used to design applications [2]. An `API` specifies how the software components interact. `APIs` can also be used to create Graphical User Interface components. The goal of an `API` is to make easy the development of a computer program by giving developers access to pre-made and modifiable blocks of code. `APIs` offer to developers the simplification of tasks and standardization. There are many types of `API` which can be classified into seven categories [6]: Web Services, WebSockets `API`, Library based `API`, Class based `API`, Operating System functions and routines, Object Remoting `APIs`, and Hardware `APIs`. In

the world of games, several `API` exists to help developers design games. For example, we have a Chess Game's `API`[1] that allows developers to embed a chess game on their website or develop a Chess Game for iPhone. We also have many similar examples for designing `RPG`[2] games such as League of Legends, Witcher or Kingdom Hearts II. One of them is Readgame available at https://pastebin. com/jgNes9j8.

Augmented Reality (`AR`) is a technology for adding virtual content to the real world [1]. Note that `AR` should not be confused with virtual reality that allows immersion in a totally virtual world. It is often associated with the addition of 3D content to an environment seen from a camera. `AR` is not very new because the first papers on the subject are from the end of the twentieth century. However, in recent years, this technology experienced a great democratization thanks to smartphones composed of several sensors. It has many applications in architecture, medicine, commerce, print media, etc. The main uses of `AR` today are, among others, QR code or bar code scanner, and video game development like Pokemon Go [8] that has a worldwide success. Augmented Reality comes in many forms. Many tools exist to help developers to create `AR` applications. Note that we have many games today using `AR` technology. For example:

– `AR` defender [9] is a commercial casual game that was released on the iPhone. It makes use of its proprietary marker pattern printed on a card so that the software can make use of the phones camera to detect the position and orientation to place a virtual tower. The goal of the game is to defend the tower by moving the camera and shooting various weapons at the enemy units that try to take down the tower. It is claimed to be the first complete and fun `AR` game on the iPhone [10];
– Ingress [11] is a location-based, `AR` mobile game developed by Niantic. This game has a science fiction back story with a continuous open narrative. Ingress is also considered to be a location-based exergame;
– Augmented Reality Chess [12] is a marker based `AR` Chess game on Android developed by Contra Labs Official. By using pattern printed on card and the phone camera to play chess game like the classical one.

3   Materials and Methods

The set of material used was divided in two main parts:

– the tools needed to implement the `API` and the different graphical interfaces;
– the material to design the game in Augmented Reality.

Figure 1 gives a general view of the different tools used.   To implement the Awale `API` we used the following tools:

– Unified Modeling Language (UML) to model the system by mainly designing a class diagram and a sequence diagram [3];

---

[1] https://chess.gallery/api/gameref.
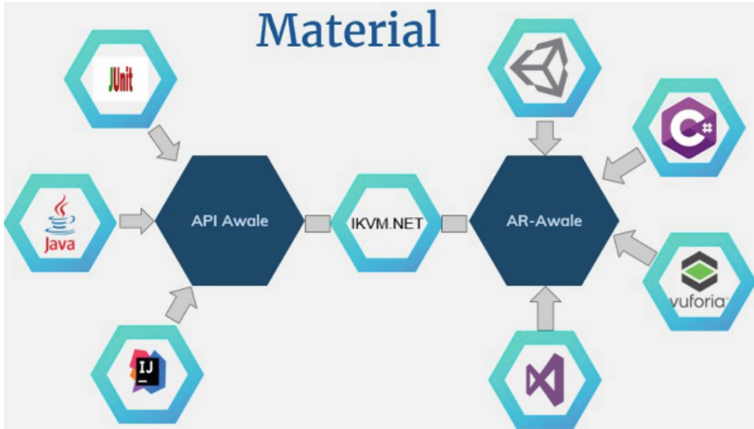[2] Role-Playing Game.

Fig. 1. Different tools used

– `Java` programming language to implement the different classes and the interaction between them and also the basic Artificial Intelligence algorithms [4];
– Junit as test framework for unit testing [13].

  For the Awale `AR` Game we used:

– The game engine Unity for the game designing [14];
– `C#` as an object-oriented programming language to implement the game mechanics [5];
– `IKVM.NET` to convert the Awale `API` source code from `Java` to `C#` [15];
– the most popular `SDK` for Augmented Reality, `Vuforia` [16]. It is a set of tools to create `AR` applications for mobile terminals. `Vuforia` uses a computer vision technology to identify the targets (2D or 3D dimension objects) in real-time.

This work was lead using the scrum sgile methodology. It had two main parts: the modeling part and the development part. The aim of the modeling part is to give a specific view of the system using UML diagrams such as class diagram, use-case diagram and sequence diagram. After the modeling we started the development phase. This latter has two parts: the `API` development and the `AR` game development. For the implementation of the Awale `API` we use the results from the modeling, especially the class diagram to implement in `Java` all the classes, the attributes, the methods, and the relations between the different classes. This `API` allows an easy implementation of an AI based player. Note that, thanks to this ease, the Awale `API` was used for an Artificial Intelligence contest, MAIC 2017 (https://mify-ai.com/maic2017/).

To develop the `AR` Awale Game, first we convert the `API Java` code using `IKVM.NET` [15]. Then we obtain a `dll` package from the Jar package. Thus the `API` can be used in Unity with the `C#` language. After importing the model and 3D graphic object of the game, we add the different features from `Vuforia` to enable `AR` mode of the game and then implement the game mechanics.

4   Modeling

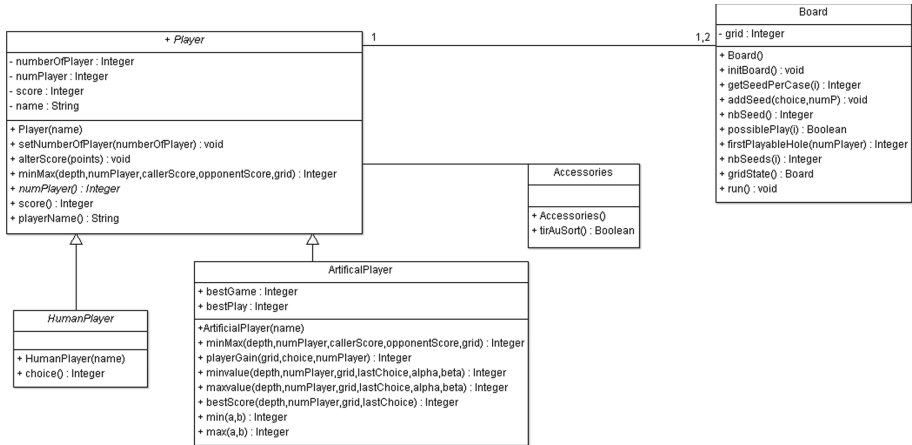Figure 2 shows the class diagram used to implement the Awale `API`.

Fig. 2. UML class diagram

This `API` had two important classes: `Player` and `Board`.

First let us focus on the class `Board`. This class has the attribute `grid`, an array of integers, which contains the state of the board with the number of seed by hole. The class `Board` has some interesting methods such as:

– `initBoard()` initialize the board game with a given number of seeds by hole;
– `addSeed(choice, numP)` adds one seed in each hole during the playing move based on the hole *choice* chosen by the player *numP*;
– `firstPlayableHole(numPlayer)` gives the first playable hole (from 0) on the board with respect to the given player *numPlayer*;
– `possiblePlay(i)` is true the given hole *i* is playable and false otherwise;
– `gridState()` returns an object `Board` that represents the current state of the board game.

The class `Player` has currently four attributes:

– `numberOfPlayer`, a class variable that contains the number of players;
– `numPlayer`, the identifier of the player in the game;
– `score`, the score of the player;
– `name`, the name of the player.

We now describe some methods of the class `Player`.

– `alterScore(points)` sets the score of the player to *points*;
– `score()` returns the current score of the player;

– `playerName()` gives the name of the player.

Actually the class `Player` is an abstract class. In the game, one should instantiate one of its sub-classes `HumanPlayer` and `ArtificialPlayer`. Note that the class `ArtificialPlayer` has some methods such as `minMax`, `maxValue`, `minValue`, `bestScore` to ease an implementation of AI based player by using the classical `Alpha-beta` algorithm with customized heuristics.

Also to ensure that our API is totally functional we implemented and run some unit tests.

5   Results

After the implementation and the different tests, we obtained a functional `Java` `API` for Awale game. This `API` is open-source and is available at https://github. com/Machine-Intelligence-For-You/Awale. By importing the Awale `API` in any `Java` project, it is easy to develop any variant of Awale Game and associate it to an intelligent agent.

Figure 3 shows a Graphical User Interface of the Awale game developed for PC. It is developed in `Java` and was used for the first MIFY Artificial Intelligence Contest (https://mify-ai.com/maic2017/). The source-code as well as the best algorithms of the contest are available at https://github.com/Machine-Intelligence-For-You/MAIC2017.
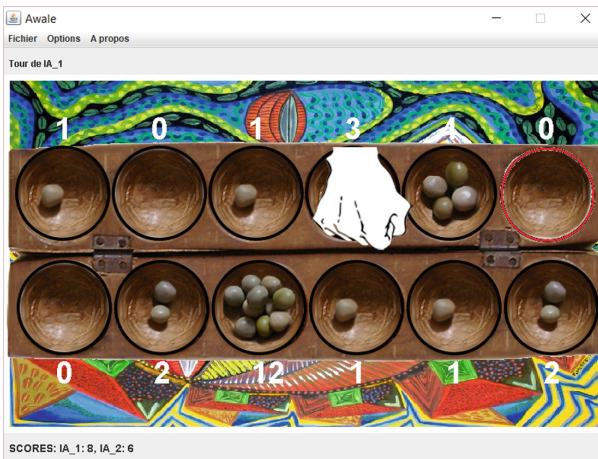
Fig. 3. PC game

We also realized an `AR` Awale game playable on iPhone and Android smartphone. Figure 4 presents the interface of the `AR` Awale's Game. We use the features of `Vuforia` knows as virtual buttons to interact with the board. Only the
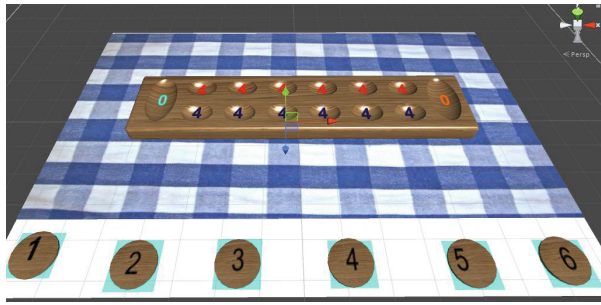
Fig. 4. `AR` game

mode Player vs AI is currently available. This `AR` Awale game is playable by using a classical virtual reality headset.

A mobile Awale game (see Fig. 5) is also developed for smartphone under Android OS. In its current version, the game supports Alpha-Beta algorithm and has an implementation of AI algorithms developed by the first three winners of the MAIC 2017.
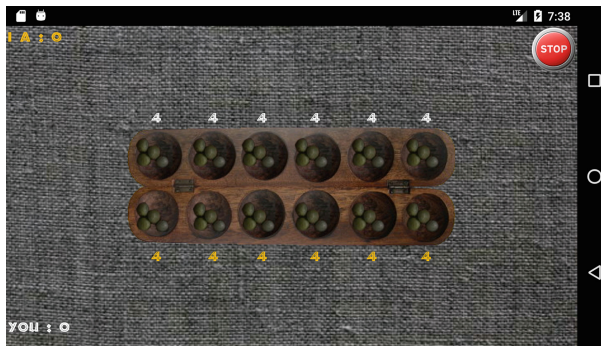

Fig. 5. Mobile game

6   Conclusion

In this paper, we have introduced an open-source Awale Game Application Programming Interface (`API`) for developers. The aim is to allow an easy implementation of the various variants of Awale. This `API` makes available to the scientific community a set of functionalities to allow an easy implementation of software agents for the Awale game. We also presented a PC Awale game, a mobile Awale game, and an augmented reality Awale game, developed based on this `API`. We do hope that the `github` repository of the Awale `API` (https://github.com/Machine-Intelligence-For-You/Awale) will have more contributors to improve its performance and functionalities.

References

1. Cushnan, D., El Habbak, H.: Developing AR Games for IOS and Android, p. 117. Packt Publishing, Birmingham (2013)
2. Mulloy, B.: Web API Design Crafting Interfaces that Developers Love, 38 p. Apigee (2012)
3. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modeling Language Reference Manual, 2nd edn, p. 742. Addison-Wesley, Boston (2004)
4. Schildt, H.: Java The Complete Reference, 7th edn, p. 1057. New York City, McGraw-Hill Companies (2007)
5. Nakov, S., & Co.: Fundamentals of Computer Programming with C#, 1122 p. (2013)
6. API Types, Sarah Maddox Simple Classification of APIs. https://www.slideshare.net/sarahmaddox/api-types. Accessed 23 June 2017
7. Awale Game Background and Rules. http://www.myriad-online.com/resources/docs/awale/english/background.html. Accessed 04 June 2017
8. Pokemon Go: Official web site. http://www.pokemongo.com/. Accessed 12 Sept 2017
9. AR defender. http://www.ardefender.com/. Accessed 19 Apr 2018
10. ORI Inbar: The RST fun augmented reality game on the iphone app store was just submitted 2010. http://gamesalfresco.com/2010/09/23/the-first-fun-augmented-reality-game-on|-the-iphone-app-store-was-just-submitted. Accessed 19 Apr 2018
11. Ingress. https://www.ingress.com/. Accessed 19 Apr 2018
12. Augmented Reality Chess. https://play.google.com/store/apps/details?id=com.contralabs.game.archess. Accessed 19 Apr 2018
13. JUnit Framework by JUnit. http://junit.org/junit4/. Accessed 04 June 2017
14. Unity. https://unity3d.com. Accessed 04 June 2017
15. IKVM Jeroen Frijters. http://www.ikvm.net/. Accessed 04 June 2017
16. Vuforia Augmented Reality SDK Vuforia. https://www.vuforia.com/content/vuforia/en.html. Accessed 04 June 2017