Jonatas Chagas

Learning Foreign Languages through Mobile Game

Helsinki Metropolia University of Applied Sciences

Master’s Degree

Information Technology

Master’s Thesis

26 March 2016

Author(s) Title Number of Pages Date	Jonatas da Silva Chagas Learning Foreign Languages through Mobile Game 65 pages 26 March 2016
Degree	Master’s Degree
Degree Programme	Information Technology
Instructor(s)	Kari Järvi, Principal Lecturer

Games are considered not only as means of entertainment but also as a tool that can be part of the learning process.

Mobile devices are largely used across the world and have become an essential utility in many people’s life. With the evolution of the hardware and the software platform of the mobile devices, developers have become empowered to build high quality games for the platform.

Graphic Adventure Games are known for immersing the players into the story and in the game world. This game genre is characterized by dialogs between the player and the characters.

This master’s thesis describes the process of building a prototype of a mobile Graphic Adventure Game targeted to help students of foreign languages to practice and learn the studied language. The project consists of a mobile game that provides an environment and a context where the student will be immersed, by participating in a story in a place where the language of study is spoken. The dialogs in the story are used to teach new words, expressions and to put the student in a context where he can improve his vocabulary. At the conclusion of the project, a prototype of the game was developed utilizing the proposed technologies.

For the creation of the game, the engine Unity 3D was utilized. The content of the game was created and managed through a content management system running on the Amazon Web Services cloud platform, based on a Serverless architecture.

Keywords	Games, Education, Language Learning, Unity 3D, Amazon Web Services

Contents

Glossary

API	Application Programming Interface, is a set of functions and procedures that allow the creation of applications which access features or data of an operating system, application, or other service.
NPC	Non-player character, is any character from the game that is not controlled by the player.
Unity 3D	Popular Game Engine that allows development of games for multiple platforms and is provided by Unity Technologies.
HTTP	Hypertext Transfer Protocol is an application protocol for hypermedia information systems. Commonly utilized in the internet.
HTML	Markup language for describing web documents.
RPG	Role-playing Game is a genre of game in which the player assumes a character role in a fictional setting.
Graphical Adventure Game	Adventure games that use graphics to convey the environment to the player.
Game Client	In games that use a client-server architecture, Game Client is the part of the game that is executed in the client device. In this case is the mobile device.
Amazon Web-Services (AWS)	Cloud computing services that make up the on-demand computing platform offered by Amazon.com.
AWS DynamoDB	NoSQL database service provided by AWS.
AWS Lambda	Cloud Platform Computing service provided by AWS that manages and runs code in the cloud.
AWS Identity Management System (IAM)	AWS system to manage users and permissions.
AWS Cli	AWS command line tool utilized to access Amazon Services.

AWS Console	AWS web portal that allows the management of the AWS resources and services.
AWS Cloud-Watch	AWS service that performs centralized monitoring and logging from several AWS services and resources.
AWS Cloud-Front	AWS Content Delivery Network system.
AWS Simple Storage Service (S3)	AWS service that provides unlimited storage and other features such as web hosting of static HTML pages.
AWS Javascript SDK	AWS software development kit for the Javascript computer language which allows the access from the AWS services from the Javascript context.
Amazon Cognito	AWS service that provides mechanisms of authentication and user management.
Content Delivery Network (CDN)	Globally distributed network of servers deployed in multiple data centers that is used to provide in high performance content to users via the nearest geographical server.
Caching	Mechanism used to store data and serve faster to subsequent requests.
Amazon API Gateway	Service offered by AWS to provide and manage HTTP Rest APIs.
Web Application	Application that runs on the web platform. Usually composed of a client and a server component.
Game Engine	Software framework utilized for the development of games.
REST	Stands for representational state transfer, which is a software architecture used in the web environment. Mostly is used with the HTTP protocol.
Denial Of Service	A type of attack to computer servers, in order to disrupt their functioning in the network.
NoSQL	Type of database that is not based on tabular relations but rather in collections of Key-Values.

Cloud Computing	The practice of using a network of remote servers on the Internet for the purposes of serving applications, data processing or other means.
Cloud Services	Services made available on demand to users over the internet provided by a Cloud Computing provider.
Provisioning	Provisioning of services or IT resources, such as servers, for customers over the network.
Scalability	Ability of a system, network, or process to handle a growing amount of work.
High Availability	Characteristic of a system to be always available and to be able to handle failure without downtime.
Resiliency	Ability to provide and maintain an acceptable level of service in face of failures and problems.
Redundancy	Is a system design practice in which the data is duplicated in order to support failure of components and hence avoid the data loss.
UI	User Interface
Shaders	Computer program utilized to treat graphics. Usually it is executed in the video card hardware.
Serverless Framework	Framework developed to ease the development of applications that utilize AWS Lambda and Amazon API Gateway services.
Serverless Architecture	Architecture of servers that is completely reliant on managed services by Cloud Providers, without a single managed server.
BigData	Term utilized for data sets that are larger and more complex than traditional data.
JSON	Javascript Object Notation is a format for exchange of data.
CSV	Comma Separated Values is a file format utilized to exchange data.
DOM	Document Object Model is a cross-platform language independent convention for representing and interacting with objects in HTML, XHTML and XML.

JQuery	Popular Javascript framework.
CSS	Cascading Style Sheets are a language for defining styles of pages written in HTML.
Ajax	Group of technologies that allow the creation of rich interfaces in the web client.
NodeJS	Open-source cross-platform runtime environment for software developed in the Javascript language.
Javascript	Programming language popular in the Web platform.
C#	Programming language that was initially developed by Microsoft and later approved by ECMA and ISO as a standard.

1 Introduction

This master's thesis describes the process of building a mobile Graphic Adventure game targeted to help students of foreign languages to practice and learn their language of study.

1.1 Background

Nowadays, due to the globalization, learning foreign languages has become a regular topic for students in general. The study of foreign languages usually requires commitment and effort from the students since it is a complex subject to master and it demands much practice and memorization. Once the student has assimilated the vocabulary and the grammatical structures he will be able to recollect that information and use it when communicating.

One of the keys aspects that involve the study of a foreign language is the context. When students are contextualized about what they are listening or reading, it is easier to make sense of the words and to know how to group them together in order to respond a coherent answer. During the foreign language classes, much of this contextualization is achieved through teacher-oriented discussions, videos, listening audio records, exercises and through the reading of texts.

The learning process of foreign languages is more effective when the students constantly interact with the language. These interactions normally challenge the student to communicate and make use of what was learned in the classroom. Once the foreign language becomes useful for the student he is rewarded by his efforts on learning the language and he is also motivated to practice and learn more, until he reaches a fluent level.

Recently video games have evolved from being considered only as sources of entertainment to become tools for educational purposes. One of the main advantages presented by games when applied to education is the capability to engage and motivate students to practice by playing and as a result learn indirectly. The game mechanics are designed mainly to challenge and entertain the players and as such, the exercises and content of studied subjects can be converted to game-like mechanics. This process of conversion is called “gamification”, which consists of applying design principles, processes and systems, utilized in games, into productive activities in order to positively influence, engage and motivate individuals (1).

There are several types and genres of games, where the mechanics vary and influence the game-play. One particular form of game is called Graphic Adventure Games, which is characterized by placing the player in an adventure story where he can interact with the environment and other characters (2). When interacting with characters, the player is presented with dialogs where he can respond by selecting options. The dialog will then develop based on the selected option. These dialogs not only extend the story, but also add to the development of the characters and determine how the storyline will evolve.

One of the main attractive points of the Graphic Adventure Games is that the development of the story tends to engage and involve the player in a similar fashion as watching a movie or reading a storybook. The main difference is that the player is actively participating on the decision-making of the story and hence affect its development.

The focus of this thesis is to develop a mobile Graphic Adventure Game that will help the students of foreign languages to learn the language of study by involving them in an adventure story that makes extensive use of the foreign language. The game will provide a context and elements that will help the student to build vocabulary, improve his grammar and most importantly to facilitate his understanding of the foreign language and its aspects. The goal is to engage the student and provide means for him to explore the virtual environment, practice the language and learn.

1.2 Technology Problem

The Mobile ecosystem has experienced a huge growth in the past years. Mobile devices became increasingly popular and have found their space in almost everyone’s lives by providing features that facilitate in the daily routine. Part of this growth is due to the improvements and enhancements of both the hardware and software in the mobile platforms (3). With these enhancements the main manufacturers of mobile devices provided APIs and toolkits to developers, allowing them to extend their services. As a result, companies are able to develop mobile applications and to reach the customers wherever they are.

The evolution of the mobile platform was also beneficial for game developers. The modern mobile devices have a powerful hardware that allows the displaying of advanced graphics and audio. With the enhancement of these hardware capabilities, the mobile devices became a target platform for the game developers (4). Thanks to the portability aspect of mobile devices, users can be entertained by high quality games wherever they are and at any time.

Since the mobile Internet speed has become much faster than ever before (5), the users can download new content, applications and more specifically games at any time. This constant availability of Internet also allows users to play games online with other players utilizing their mobile device.

Considering this modern landscape, the challenge is to build a mobile game that will make use of the graphical and audio capabilities offered by the hardware of modern mobile devices. The game was developed in a client-server architecture. The server side stores and provides the content of the game and stores information about the players. The communication between the client and the server side takes place via the HTTP protocol.

1.3 Research Objective and Outcome

The objective of the research is to explain in detail the process of creating a mobile game that facilitates the immersion of students of foreign languages in a context where the language is extensively used.

The research was conducted through the several steps. In each of these steps the procedures are detailed and explained. The reasoning and theory behind the design decisions both related to the content and structure of the game are elaborated throughout the Sections 4, 5 and 6. The technical requirements are addressed by exploring the current available technologies in both the mobile as in the server-side landscape. The motives and reasons behind the choice of the determined technologies are explained in detail.

The outcome of the research is a functional prototype that demonstrates the main functionalities of the game and the relationship of each component in the final solution. The challenges and difficulties encountered during the design and development phases are also described.

2 Method and Material

This section describes the steps regarding how the research was conducted, in terms of methodology, strategy and planning.

2.1 Context

The process of building an application consists firstly of investigating the domain of the application and secondly choosing the appropriate technical implementation.

The domain of the application in this study is the process of learning foreign languages through the use of an educational graphical adventure game. Specific areas regarding games in education, in particular in language learning, are selected for the purposes of this study and are overviewed in Section 3. Those aspects are subsequently applied into use in the proposed mobile game in Section 4. Thus, to build an application that is coherent to its purpose, the domain is first investigated and the area for it is specified.

Since the game is targeted to the mobile platform, a game engine that supports this environment was selected and is presented in Section 5.1.1. A description of the selected game engine is then presented and the reasoning that supports the selection of the engine for this prototype is also presented on that section. The server side technologies utilized for the construction of the prototype are presented in Section 5.1.2 along with the reasons why those technologies were chosen for this project.

The domain and the technical specification for the project and the execution and implementation of the proposed application are presented in Section 6.

2.2 Research Process and Design

Given that the problem to be solved by the application is how to enable students of foreign languages to learn through the use of a mobile game, the study of the domain is divided in the following activities:

- An overview of the techniques for learning foreign languages through digital means, more specifically the Microlearning technique.

- An overview of the process of Gamification.
- An overview about games in education.
- Design considerations and overall structure of the game.
- Presentation about the design and the elements of the Game.
- Development and presentation of the proposed application.

The technical activities that need to be executed for the solution of this research problem are:

- Analysis and presentation of the selected technologies that are used for the development of the prototype.
- Creation of an architectural blueprint about the components that are utilized in the application.

Once the domain and the technical activities are executed and finished, the following tasks can be executed:

- Build a high-level prototype of the application.
- Validate if the combination of the chosen technologies were appropriate to solve this technical challenge.
- Validate if the application is helpful to students of foreign languages

The focus of the thesis is the development of the technical solution. As such, the description and analysis of the implementation was elaborated in detail. The benefits and reasons that support the selection of each technology and framework utilized in the project are discussed in depth, along with a brief overview of their inner workings. Once the technologies are presented, an architectural blueprint was developed in order to provide a clear view of how each component of the solution works in cooperation with one another. The implementation of each of the parts is described in practice and the final prototype is presented.

2.3 Research Strategy

The research strategy that applied in this Master's thesis is the qualitative strategy.

Given that the core of this research is about solving the problem of building a mobile application that can effectively help foreign language students to learn, the qualitative

approach helps to find the solution through the use of individual interviews through means of questionnaires and personal enquiries. The result of these interviews provided feedback that contributed to the understanding of the role of the mobile application in the process of the study activities of the foreign language learners.
Pursuing this further, the first group of interviewees for this master’s thesis were foreign language teachers. Their expertise provided means to find the right techniques and the key content that helps students to learn. Therefore, their experience in the field and their knowledge was vital and was a key factor in the success of this prototype. Furthermore, their ideas and concepts were collected and materialized into a functional prototype of the application.

Another valuable input came from the students of foreign languages. They helped to validate the ideas and the concepts that were established and designed by the former group. Once the prototype was presented to different groups of students and individuals who study foreign languages, the game, its content and its UI components were tested and evaluated. The result of the evaluation provided a list of improvements, changes, perceptions and opinions about the application. Having these improvements and changes implemented, the application is more valuable to the target audience.

The experience of IT professionals is the final, third strong factor to take into account when benchmarking and selecting a technology for the implementation of a project. Therefore, in order to make sure that the right technology was chosen for the implementation of the application, experienced software engineers were interviewed by personal communication. Their insight and experience provided a valuable input and had a direct influence on the success of the project.

Based on the insight from these three groups - teachers, students and IT professionals - the proposed prototype was developed.

2.4 Data Collection and Data Analysis

This project utilized a number of data sources: personal interviews, analysis of technologies and development discussions with IT professionals.

To collect the data, questionnaires were elaborated for the students. The questions were not the mandatory script for each interview, but rather were the basis from where the discussions started. In order to validate the prototype and gather feedback about the application, a number of selected students tested the application. The testing sessions were extremely helpful to extract insights, opinions, perceptions and attitudes from the individuals that were presented to the prototype. These reactions shaped the final version of the application.

3 Theory and Existing Knowledge

This section describes the theory and fundamentals that serve as a base for the research and for the development of the prototype.

3.1 Microlearning in Mobile Devices

Microlearning is a pedagogical technique that aims to solve the problem of learning on demand in a short span of time (6). The technique is mostly based on the idea of presenting the content in small chunks at distinct times of the day to the learner, allowing him to study whenever he has time and wherever he is. Microlearning is better illustrated as a set of models of learning. These models are divided in micro dimensions, which can elaborate better the concept (7). The micro dimensions are:

- Time: short effort, small iterations
- Content: small, narrow and simple units
- Curriculum: part of curricular setting, parts of modules, informal learning
- Form: fragments, facets, exercises
- Learning type: repetitive, activist, reflective

Considering the micro dimensions mentioned above as the pillars of the technique, one could easily notice the uniqueness of this technique and how it can be a perfect fit to mobile devices and foreign language learning.

Most of the learning life is often informal and in control of the learner. Since the learning process requires attention and focus, the context, the experience and the people are strong factors in the informal learning (8). Due to the expansion of the technological field in our quotidian, the technology has also influenced deeply in the informal learning. As the word informal suggests, this method of learning is not necessarily structured in pre-defined sessions but rather at ease and necessity of the learner.

3.2 Motivation

The learning process of foreign languages by adults is characterized by several factors that contribute and influence the students to achieve success or to failure in the study of the desired foreign language. These factors may not be the root cause and serve as the only and true evidence behind the status of progress of a student, but can be highly correlated to common reasons why students succeed or not in learning a foreign language.

The key factors for foreign language learning have been summarized as motivation, attitude, socialization, self-image and ego (9). These factors highly influence the process of learning languages by adults. Motivation is brefily described as the choice of a particular action, the persistence and the effort that is applied (10), and is highlighted as one of the most important factors, since it is highly related to the willingness and the effort of the student whether to study or not. By being so important and decisive point in the learning procress, motivation is one of the biggest challenges for educators to address.

3.3 Gamification Process of Education

Gamification can be described as the design of a process that is used to engage and motivate individuals, or groups, to achieve certain tasks. This design, in its nature, is similar to the mechanics that are utilized in Games (1).

The education of students can be much benefited through the use of the Gamification in exercises. Games are by their nature engaging and fun. These aspects, if applied coherently to educational tasks, can promote interest and the engagement of the students towards the subject that they are learning. The main driver behind the learning through games, from the perspective of the student, is the fun and other positive stimulus that the student experience while doing a task that sometimes is considered monotonous and not interesting.

The process of Gamification consists of transforming a regular task into a game. This transformation is made through certain designs that are known to drive behaviors that stimulate the users positively towards an end goal (1). Educational exercises are a good match for games since they present challenges, which are a key factor for any game. In most of the games there is the concept of progress, where the player starts at an initial level and progresses towards harder and more laborious levels, which is analogous to the process of the student in his field of study. This progress can be easily measured in a similar fashion as in game-levels.

Another aspect of games that can be extracted into the gamification of education is the social part. Students often interact with other students and with their instructors during the learning process, which results partly into a social experience. Certain kinds of games also have social aspects and mechanics, which in turn, present to the players the possibility of competing against other players or to collaborate in order to achieve a certain goal. Players also tend to be more engaged to games where they are presented to the possibility of belonging to a group or to meet friends.

More specifically, the exercises that belong to the domain of languages can be easily transformed into games due to their nature. The learning process of languages requires understanding of basic grammar and a substantial amount of practice, mostly through exercises. Normally the grammar is learned effectively if constantly repeated until the information is familiar to the student and the rules are applied naturally when the student needs to speak or to write. This repetition can be exhaustive and not attractive to the students, making the process of learning a language hard and frustrating.

By presenting the students with exercises, designed in a game like manner, the student will be more motivated to study and to learn the language. The exercises need to be designed so that the student will be challenged enough to remain interested in the game. If the challenges are too hard for the student’s level, he will tend to be frustrated and will give up the exercises. Therefore, the level of difficulty of the game needs to be set in accordance to the level of the student in the given language.

Part of the instructor’s task is to give hints to the students when they ask questions or when they have doubts about a certain exercise or topic. A hint mechanism can be introduced to the game by detecting the amount of time the student is taking to answer the exercise, the number of mistakes the student has done or simply by adding an element to the interface, such as a button, where the game will not give the answer but will give a hint to help the student into answering the exercise.

Another task from the instructor that can be done by the game is the guidance through which topic to study next. This can be offered through an analysis of the development of the student by quantifying the number of successfully answered questions and the number of errors. By analysis of the scores of a student the game can suggest which topics the student must address in order to advance to the next levels.

Nowadays most of the games are making use of the evolution of the Internet speed by providing multiplayer mechanics. These mechanics can be complex such as two or more players playing concurrently as a team against other group, or as individuals, or can be simple such as sharing rewards, trading game goods, participating of leader-boards or by comparing your progress against your friends. These known mechanics provide a variety of ways to introduce social features into educational games.

To conclude, exercises about learning languages could be a good match for the process of gamification, since several of the mechanisms discussed can be applied to this category of exercises and can potentially leverage the learning of the students of foreign languages.

4 Prototype

This section describes the design considerations that were taken into account in the development of the Prototype.

4.1 Design Considerations

The game needs to be developed in such a way that it will be extensible for several languages and must support translation of the content. When discussing the design of the game with teachers and students, the following requirements were pointed out:

- The students need to be able to easily find the translation of the words and tenses found in the storyline and in the dialogs.
- When the Non Player Characters (NPCs) engage in a dialog, there should be not only the response in text, but also in audio, if possible, so the student can play the same part of the dialog several times, read and listen carefully to the pronunciation of words and their meaning.
- Key words that are important for the development of the story should be underlined or displayed in bold font.
- The student should be able to explore every possibility presented in the dialogs to see every possible outcome for the given options. He should be able to replay the same dialogs as many times as necessary.
- The dialogs must have colloquial language, so the students can learn how the language is spoken by the locals.

Given the requirements presented in this project, one of the biggest challenges was to produce and provide all the content for the game. The content is composed of the story, dialogs, messages and possibly sound effects.

Besides the quality of the content, the story needs to be interesting in order to engage the student. Even though real-life experiences would be the most helpful scenarios that the game would provide, in graphical adventure games it is really important to have a plot that challenges the player and keeps him interested and curious about how the story unfolds.

Another design aspect that needs to be taken into consideration is the mobile aspect of the game. Even though smartphones nowadays present a higher resolution, the game is limited to a small screen, which is particularly a design challenge since the game has a considerable amount of text in the dialogs and in the story.

Most of the mobile games are casual and that means that players can stop and resume playing the game at anytime. Furthermore, casual games are characterized by having short sessions of game play and by the ability of being resumed at any given point of time while keeping the progress of the player. This casual aspect needs to be carefully considered in the design of the game and when building the dialogs and the story line. Differently from a console or a pc game, the player will rarely spend hours staring at the mobile screen, but rather will play the game during short sessions. That means that the text needs to be concise and the story cannot be so complex, since the player will be switching on and off from the context of the story quite regularly.

Even though nowadays the mobile devices have a considerably large amount of storage capacity in their hardware, there is a limit of the size for the games to be placed in the application store. That means that the game cannot be bundled with all the dialogs and audio files. Players also would not be pleased to have their phone filled with assets of the game when they want to have space for videos, pictures and other applications. Another limitation is the speed of the Internet. Even though the Internet nowadays is faster and allows the users to watch high quality videos in the mobile screen, the game cannot use the whole Internet connection from the device only to download assets from the game. Considering the amount of content that the game will require and the issues stated above, the game needs to be designed in such a way that the text and audio files are relatively small and are provided on demand via internet.

Lastly, in order to identify whether the content of the game and the story are being helpful and are engaging to the students, the game needs to have a mechanism to track the actions and the behavior of the players throughout the gameplay session. By analyzing the player’s behavior, we can identify parts of the game that need to be changed in order to provide a better experience to the student. This analysis mainly consists of the amount of time spent in certain steps of the game, interactions with the user interface, errors messages from the game, length of sessions and the general activity of the players when playing the game. The analysis of the game is vital to understand whether the game is helping the students and is keeping them engaged throughout the learning experience.

Social features could be introduced later in the game. The players could write letters and drop them around, in the scenario of the game, in order to give hints to other players about what they need to do to achieve their goal. This would encourage the students to practice their writing skills in the language as well. Another possibility would be to allow native speakers to help in producing the content of the game, by allowing them to name and add other objects to the game and possibly create NPCs that are not related to the main story but would also add value to the immersion experience of the player in the game environment.

Even though the game is intended to be simple, since the topic is so wide, there are several features and other possibilities that could be introduced in the game and hence enrich the learning process of the student during the game play.

5 Technologies and Architecture

This section describes the technologies that were utilized to build the prototype. It also gives a detailed explanation of how the prototype was developed.

5.1 Tools, Technologies and Frameworks

Several open-source and proprietary tools were utilized during the development of the application. These tools are strongly backed up by a community of developers who contributes with tutorials, articles, code samples, patches, extensions and helps to maintain working Forums where users and contributors can find answers to their questions and get feedback and suggestions about projects. Most of these projects are actively maintained and have bugs fixed, new features and new releases being published quite often.

5.1.1 Client Side Technologies

The mobile platform has become increasingly popular for developers, in particular for game developers (4). Every new generation of phones comes equipped with powerful hardware enabling devices to process advanced graphics, enabling game developers to build visually attractive games for the mobile platform. Another feature of the platform that has sparkled the interest in the developers is the capability of selling the game in the application market and to utilize micro-transactions, so called “in-app purchases”, in the games.

Due to the attractiveness of this new platform for game developers, several game engines for mobile devices have been developed. Most of these game engines are capable of porting the game across platforms such as Android and iOS and provide tools to handle graphics, sounds, controls and input and output. Some engines are specialized into 3D games or 2D games and a small subset of these engines provide the capability of developing both 2D and 3D games. When it comes to the pricing, there are engines that require a usage fee, some are free for development but have royalties in the profit made by the game and some are completely free for usage.

For the development of this project the Unity 3D engine was utilized. This engine has become very popular over the years due to its simplicity, ease to learn and to its features set that continues to grow.

5.1.1.1 Unity3D Engine

Unity is a game engine developed by Unity Technologies. This engine is widely used by both renowned game studios and by a large community of game indie developers (11). Unity had its user base to grow exponentially over the years due to its rich set of features and to the fact that it is relatively easy to learn. One of the main features from Unity that is highly appreciated among its user base is the capability that it provides to export the games and apps to multiple platforms, including mobile, game consoles and computer platforms.

In Unity, the game is composed of Game Objects. These objects have properties and components that determine their behavior in the game. Unity allows the users to define the behavior of the Game Objects by editing properties of the components that compose the objects, adding and removing new components and by defining custom components via scripting languages such as C# and Javascript. The APIs from Unity are well documented and can easily be accessed in the internet. In order to allow its users to make games that are more realistic, Unity also offers a physics engine called NVIDIA PhysX 3 for 3D games and another physics engine for 2D games called Box2D. This engine also allows the users to use their own shaders in order to create custom effects in the game (12).

Another feature that speeds up the development of games in Unity is the graphics pipeline that makes the life easier of the game developers, by allowing them to easily import assets with proprietary file types that are generated by technologies such as 3D Max, Maya and Blender. Unity has a mechanism of animation called the Animator Component, that allows the game developers to define, in a state machine like model, the transitions between the animations from the game object models. These transitions are triggered when custom variables from the animator component have their values changed, usually via scripting. As a result, the animations are decoupled from the scripting and hence make the code to look clear and concise.

From a programmer point of view, Unity is very well structured since all the scripts represent behavior that can be added to the Game Objects by working and being structured in a very similar fashion as the Composite Design Pattern (13). Hence, in order to build a game, the programmer must compose the game objects with the desired behavior. Concerns such as rendering, path finding, physics and layout building are addressed by engine and the programmer is left to implement the core gameplay rules by modeling behavior in the Game Objects. Once the scripts are created they can be reutilized by being applied as a component to other game objects. The Unity editor looks a bit complex and overwhelming at first since it has a designer oriented UI. However, after studying tutorials and the documentation, one learns the concepts of the engine and the UI becomes self-explanatory.

Unity is an excellent engine to prototype and build games quickly and also has advanced features that allow the experienced game developers to take the most out of the hardware of the intended target platform. The growth of the engine in the market place also drives Unity Technologies to develop more features and to address issues that are found by game developers. Since one of the main points of the engine is to export to multiple platforms, Unity also handles all the work required to change and to adapt to the new platforms that are changing more often nowadays, allowing developers to focus on the game, rather than on the technologies.

5.1.2 Server Side Technologies

The server side components of the game are responsible for storing the current state of the players, that is, their progress in the game, their settings and any other information that is relevant to their game play. The server will also be responsible for storing the content of the game, which is composed by the dialogs, quests and characters of the game.

Due to the advance of the Cloud Computing technologies, providers such as Amazon Web Services offer not only infrastructure in the cloud, but also services, on a pay-as-you-go model (14), that allow developers to focus on the core business of their applications and all the other requirements such as high availability, resiliency, redundancy, scalability and auto recovery are simple offered as features to the application developer. This model has been growing in such a scale that more companies are adopting and utilizing these technologies in order to increase their time to market, by focusing on their product, rather than on the efforts of setting up and maintaining infrastructure for their internet based services (15). Not only small startups but also giant companies have acknowledged both the financial and sustainable benefits of running their applications in the cloud.

With the evolution of the Cloud based services, technologies became completely abstracted to APIs and their internal workings can be analogue to black boxes that are completely unknown to the application developer. The underlying hardware and deployment is hidden behind clear and concise APIs. These services are also designed to be resilient and scalable. The resiliency comes with the premise that every component of the service is replaceable. This is achieved through an architecture designed to be redundant, where the data is distributed and copied systematically. Thus, if one component fails, another component that has the same data takes the role to replace the problematic component. The scalability comes with the fact that most of the Cloud services function based on virtualization and are managed by a software layer that can allocate and de-allocate more resources according to the demand of the end user (16).

Even though both resiliency and scalability are requirements that are difficult to achieve in a standard environment, these aspects greatly influence the end user's experience and since some Cloud services offer these features included in most of their products, companies turned to utilize these services for a competitive edge and in some cases to save costs of infrastructure (17).

In the following sub-sections each of the server side technologies is presented and described in detail.

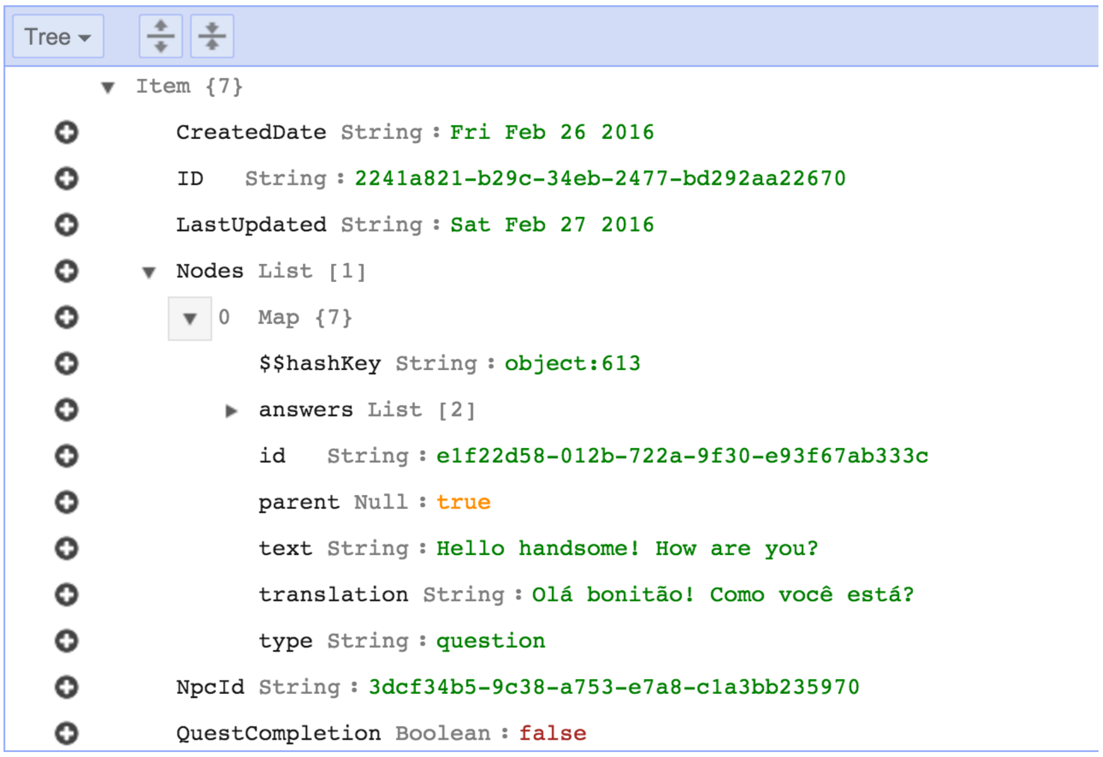
5.1.2.1 AWS DynamoDB

Amazon DynamoDB is a NoSQL database service offered by Amazon where the hosting, maintenance, scalability, redundancy, backups and hardware provisioning are handled by Amazon. This service in essence consists of a solid and well defined API that allows developers to save, search, fetch, delete and update records in DynamoDB tables. Scalability is addressed by DynamoDB through the means of traffic throughput configuration. Redundancy, high availability and fail-safe mechanisms are also built-in into this technology (18).

Databases are part of the core components of several digital businesses. In order to maintain a database, in terms of performance optimization, disaster recovery, scalability and security requires often times a full time experienced database administrator. Companies that fail to address these requirements when maintaining databases, often face issues where the data is lost, the end user experiences slowness when using the software, or the application is unable to take on new users. These issues often times cause the companies to lose customers since the user experience is affected and as a result they do not trust the service as they did before. Even though these duties to keep a database running are very important, small and even bigger companies do not have enough personnel to tend to these issues and these administrative tasks. This is where Amazon DynamoDB service comes into the picture.

Amazon DynamoDB spreads the data automatically in between enough servers in different datacenters so as to guarantee that if one server goes down the data is safely stored in other servers. This concept is called redundancy. The distribution of the data between several servers also enables the handling of increased traffic, thus resolving scalability and performance issues. The automatically provision and configuration of the database servers lifts from the developer the responsibility to perform these time consuming and arduous tasks (19).

DynamoDB is simply composed by Tables. Each table has Items and each Item has a set of attributes. The attributes of the items can have up to 32 levels of nested items (18). The attributes are basically a set of key-value pairs. The keys are naturally Strings and the values can be of the types: String, Binary, Number, Boolean or Collections. The Collections comprehend of: StringSet, NumberSet, BinarySet, Map and ultimately List. Each item from a given DynamoDB table does not need to have necessarily the same number of attributes as the other records, making this database extremely flexible to save different data structures. An example that illustrates the structure of an Item from a DynamoDB table is displayed in Figure 1.



```
▼ Item {7}
    CreatedDate String : Fri Feb 26 2016
    ID   String : 2241a821-b29c-34eb-2477-bd292aa22670
    LastUpdated String : Sat Feb 27 2016
  ▼ Nodes List [1]
    ▼ 0  Map {7}
          $$hashKey String : object:613
        ▶ answers List [2]
          id   String : e1f22d58-012b-722a-9f30-e93f67ab333c
          parent Null : true
          text String : Hello handsome! How are you?
          translation String : Olá bonitão! Como você está?
          type String : question
    NpcId String : 3dcf34b5-9c38-a753-e7a8-c1a3bb235970
    QuestCompletion Boolean : false
```

Figure 1. Example of DynamoDB Item in the AWS Console.

For the definition of unique Items each DynamoDB table has a primary key that can have two types: Partition Key or Partition Key and Sort Key. The Partition Key is basically an attribute that defines the uniqueness of each record. DynamoDB utilizes the hash of the Partition Key to determine where the Item will be stored and where it can be found. The second type of primary key, which is the Partition Key and Sort Key, is basically a composition between two attributes of the Item. The first attribute is the Partition Key that defines the location of the item or items in the database based on a hashing of its value. When retrieving the items that are saved under the same Partition Key, DynamoDB will utilizes the Sort Key to sort the items from that partition.

At this point, up to 5 Global Secondary Indexes and 5 Local Secondary indexes can be created per DynamoDB table (18). The Global Secondary Indexes allow the developer to query the DynamoDB Items in a table using another Key than the Primary Key that was defined. Similarly, to the Primary Key, the Global Secondary Indexes (GSI) have a Partition Key and a Sort Key. The Local Secondary Index is defined for items that have the same Partition Key but a different Sort Key is defined. The possibility of defining Primary key and Global Secondary Indexes provide much flexibility for the developers when creating tables in DynamoDB.

The configuration necessary to handle the amount of traffic required is addressed through the Provisioning throughput levels. These levels can be adjusted via the AWS Console, AWS SDK or via the AWS Cli tool. The Provisioning levels define how many writes and reads are expected per second on a given Index. When creating a DynamoDB table the developer must estimate the necessary provisioning level for the Primary Key of the table and for any other additional Index (Global or Local) added to the table (18). As the technology evolves, the limits per write and read operations change. At the moment, one read capacity unit allows one consistent read operation per second for items up to 4 KB in size. As to the write operations, at the moment 1 write per second allows one consistent write operation for items up to 1 KB of size. The total number of operations for both read and writes will depend on the size of the items. Each Index (Primary Key, Global Secondary Index or Local Secondary Index) have its own provisioning levels. Once the provisioning level is reached and is exceeded, DynamoDB will throttle the upcoming requests. When this is the case, the DynamoDB's API will return failed requests with the code HTTP 400 (Bad request) and a ProvisioningThroughputExceededException. The partitioning from DynamoDB and the data distribution occur transparently to the developer and are based on the Provisioning Throughput Levels.

The AWS DynamoDB pricing is defined by a combination of factors such as the: Provisioning Throughput Levels, the size of the Indexed data storage, data transfer, among others (18). This service is extremely powerful and is the corner stone for a Serverless Architecture.

5.1.2.2 AWS Lambda

AWS Lambda is a service offered by Amazon Web Services (AWS) that allows developers to run their applications utilizing AWS infrastructure. Once the developers upload their application code to the platform, the service is responsible for allocating the infrastructure required to execute the application and to manage resources such as the actual servers required to run the application. The applications are structured in the format of functions which are only executed when events are created and assigned to these functions. The immediate advantage of this service is that the pricing model is charged for each time that the function is executed as opposed to traditional pricing model where servers are charged for the amount of time that they are running (20).

The events that trigger the execution of the functions have several types, including changes in the data in Amazon S3 or in Amazon DynamoDB, HTTP requests through the AWS API Gateway, scheduled events and via the AWS SDK. This service currently supports functions to be developed in the computer languages of NodeJS, Java and Python (20). In Figure 2 is displayed a simplistic diagram that illustrates how the AWS Lambda functions work.

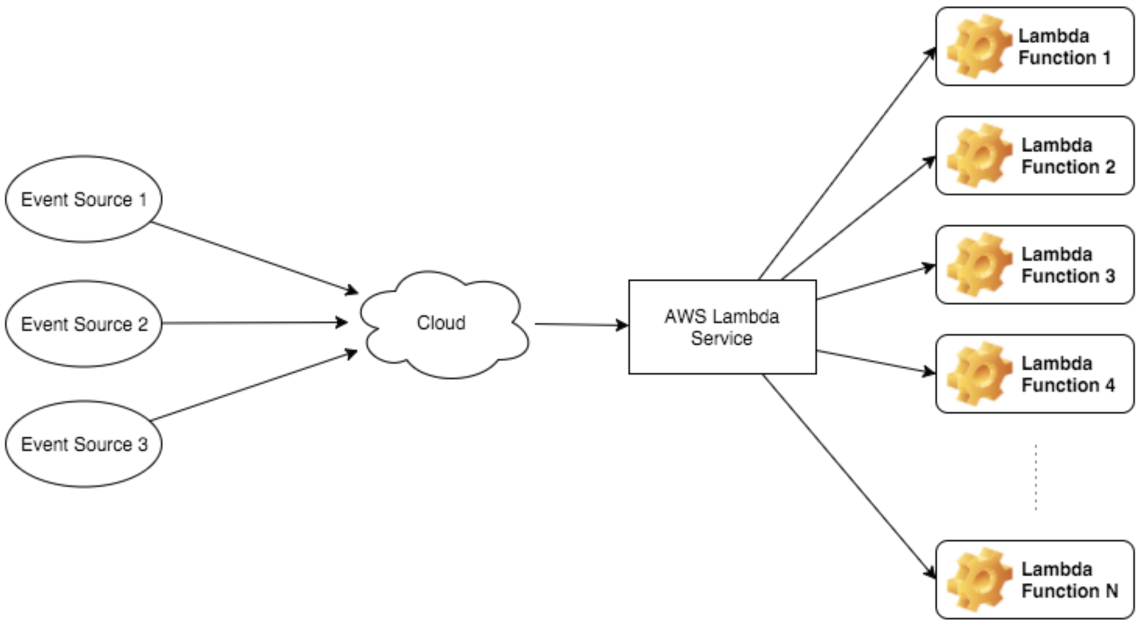


Figure 2. Diagram that illustrates how the AWS Lambda Functions work.

The advantages of utilizing such a service is that Amazon manages the servers, configures the environment of execution, handles the maintenance, provides monitoring, stores the logs and scales automatically upon the the amount of traffic, leaving the developer to focus on defining the business logic from the server side function. Since the environment that handles the execution of the functions is isolated, the application is secured and will not affect the other events that are being processed.

A standard architecture pattern that is utilized in conjunction with this technology is the Microservices architecture, which is in simple terms, the definition of multiple small independent services where each one has a very specific responsibility and together, these services, compose the whole application (21). Usually the services are deployed separately, by automated processes, and run in an independent environment and have no tight connections to each other in terms of runtime. Since there are no tight connections between the services, if one fails the others will not be affected. These services are very narrow in terms of responsibility and knowledge about the other components of the application. Each service is a small self-contained business unit. This architecture solves several issues that were constant in the monolithic architectures, which are normally complex and act as a single unit. Since the Microservices architecture is more flexible, it can scale better and it is easier to maintain since it is less complex and is not tightly coupled to other services. This architecture is certainly not the solution for every software problem, but has been proven to be very efficient and reliable in several scenarios and business requirements. The Microservices architecture coupled to AWS Lambda is an excellent way of building applications as isolated and scalable services.

AWS Lambda runs in isolation to other lambda function calls. This aspect guarantees that if one function execution has memory leaks, for example, or gets into a problematic state, it will not affect the other functions that are being executed concurrently. This aspect is very important since it guarantees that the issues are isolated and will not affect other parts of the application.

Since the AWS lambda is scalable by design and the pricing model is based on the function executions and not based on the servers running the application, it is up to Amazon Web Services to handle the provisioning, balancing and management of resources for the given function (20). In terms of fast application development, this feature is essential for companies that need to release their product fast and do not have enough human resources to guarantee that the servers will be up and running all the time and will be able to handle scalability without human supervision.

Besides the computer language that the function is written on, the developer also needs to select the amount of memory that each function will allocate for each execution. Notice that the higher the memory allocated per execution the higher the price per execution (20).

The security part of this technology is handled via the IAM system from Amazon, where the developer will define an IAM role for each Lambda function. Based on that role the privileges on the other AWS resources that the function will utilize must be defined. For example, if the developer wants the function to insert records into a DynamoDB table, a grant insert record privilege must be added to the IAM policy of the AWS Lambda function’s role.

The deployment of the AWS Lambda functions can be done via the AWS Console, which is uploading the packaged application in a zip file format via the web based dashboard from AWS. Another way of uploading AWS lambda functions and changing configurations is via the AWS Cli tool that can be utilized via the terminal. Some open-source projects such as Serverless developed tooling that allows developers to easily deploy AWS Lambda functions through the use of the command line terminal. These tools also provide facilities to run the lambda functions locally, hence facilitating the development process. The logging from the Lambda functions is handled via the AWS Cloud Watch technology.

Coupled with other technologies such as AWS DynamoDB and AWS Simple Storage Service, AWS Lambda is one of the main parts of an architecture completely Serverless, in a sense that, the developer only needs to worry about developing the code, not worrying about configuring, provisioning, monitoring and maintaining servers.

5.1.2.3 Amazon API Gateway

Amazon API Gateway is a service provided by Amazon to manage, monitor and secure APIs. This service is designed to be the forefront layer of applications that are exposed via APIs. API Gateway functions by receiving HTTP requests and forwarding those to AWS Lambda or any other Web Application. By design the API Gateway can handle a high load of traffic (22). One of the biggest benefits from the API Gateway is the fact that it is designed to resist and protect the backend of the APIs from Denial of Service (DOS) attacks and to support high loads of traffic. Other features such as security, access control, monitoring and API versioning also make a compelling case to utilize this technology.

With regards to performance, API Gateway utilizes AWS CloudFront technology in order to distribute its endpoints. By working on top of the AWS CloudFront, API Gateway has the advantage of having its APIs distributed across the world through the content delivery network (CDN) and hence provides low-latency access to its clients, since the endpoints are geographically close to the clients. Automatically configured caching is also a great feature from this technology. With a smart caching mechanics provided by the API Gateway, the API's backend is guarded and protected from attacks or malicious activity. API Gateway uses the AWS CloudWatch technology to provide logging of the API calls, monitoring of the latency of the API responses and other relevant metrics and the definition of Alarms when certain thresholds are reached.

One challenge often found by API providers is the ability to host and support multiple API versions. Once an API is released, client applications of the API are built, and so the functions exposed via the API cannot have simply their signature changed without breaking the client applications that depend on those functions, yet new functionalities need to be added and sometimes the APIs need to be redefined. For those cases, API Gateway provides a built-in support to API versioning. In terms of security, the API Gateway can easily connect with other Amazon security services such as Amazon Cognito and Amazon Identity Management System (IAM). Mechanisms of throttling can also be defined in the APIs exposed via API Gateway, in order to protect the implementation from the APIs from unpredictable spikes in traffic and malicious attacks.

One of the biggest features from Amazon API Gateway is its ability to work in conjunction with AWS Lambda. APIs can be tied to Lambda functions. By connecting the two technologies, developers can build REST APIs that are Serverless and have built in caching mechanism, are fault tolerant, are geographically distributed, high available, scalable and secure.

5.1.2.4 Serverless Framework

Serverless is an application framework that provides tooling for creating applications based on AWS Lambda and API Gateway technologies. This application framework is utilized via a command line based application that helps to create and manage Lambda functions and APIs in the API Gateway. Serverless also provides an easy to use way of managing AWS Resources utilized in the application, such as IAM rules (23).

The tooling provided by AWS for the creation of Lambda Functions and APIs in the API Gateway can be somewhat hard to use, specially for beginners in the AWS platform. Serverless provides a very easy to use toolset for creating applications based on these technologies. It also adds a module and component based structure that enables the developer to create well defined and organized applications and an environment-based configuration, allowing the developer to switch from development to production environments at ease.

5.1.2.5 AWS Cognito

AWS Cognito is a service provided by Amazon WebServices that provides features related to authentication and user's data management. It has integrations for authenticating with Google, Facebook, Amazon and also support custom authentication systems. Since user management is often times a mandatory feature for every system, AWS offers this service in order to allow developers to focus on the core business of their applications by handling the user management for them (24).

In AWS Cognito, in order to create authentication for a given application, the developer needs to create an Identity Pool. In this Identity Pool it is defined a IAM role that will be utilized by Authenticated users and another that will be utilized by Unauthenticated users. Once the Identity pool is created, the roles for both the authenticated and unauthenticated users need to be configured in order to restrict or grant access to other AWS resources, such as AWS Lambda Functions or APIs defined in API Gateway. For example, a certain lambda function can be executed by every user, both authenticated and unauthenticated users. In this case both roles will be granted execution access to the Lambda function. However, if the lambda function can be only executed by authenticated users, both roles need to be configured accordingly in order to satisfy this requirement.

When working with other Identity Providers, such as Facebook, for example, the workflow between the application and AWS Cognito work as described in Figure 3.

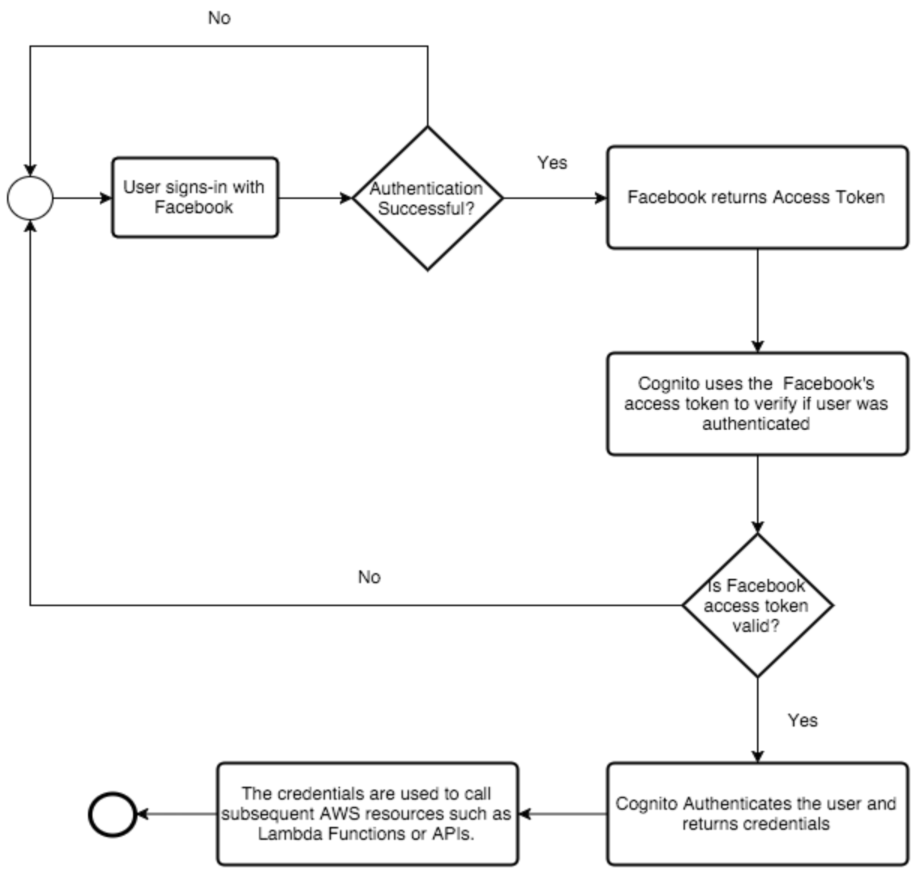

No
User signs-in with Facebook
Authentication Successful?
Yes
Facebook returns Access Token
Cognito uses the Facebook's access token to verify if user was authenticated
No
Is Facebook access token valid?
Yes
Cognito Authenticates the user and returns credentials
The credentials are used to call subsequent AWS resources such as Lambda Functions or APIs.

Figure 3. AWS Cognito authentication flow to Facebook.

AWS Cognito also has features to store and to synchronize the authenticated user's data between different devices. This is particularly useful when developing applications for mobile devices, since users nowadays tend to use applications in multiple devices such as phones and tablets.

This service is another piece of the puzzle in the AWS based architecture. Since it works based on the IAM system, it can be easily utilized to secure and restrict access to other AWS resources.

5.1.2.6 AWS Simple Storage Service

Amazon Simple Storage Service, also known as S3, is a powerful cloud storage service provided by Amazon Web Services that allow developers not only to store data, but also to distribute content and host websites with static HTML content. This simple object storage, as other AWS services, consists of a concise API interface that enables the developers to upload, fetch and delete files. S3 is also designed to be redundant and high available and has no limits as to the amount of data that it can store. As to the security, Amazon S3 is integrated to the IAM system, where rules regarding the access to the data can be defined in a granular manner to users or groups of users. The access to the files can also be logged in S3, giving to the developer extremely visibility and control about the access to the data (25).

S3 consists of buckets that are named globally and hence have to have unique names. Each bucket can have keys and these keys can be defined as a tree folder based structure, similar to the file structures found in file systems. The security permissions can be granted to the whole bucket, or individually to the files and folders.

Up to this day, S3 is one of the most popular and powerful services offered by Amazon (26). S3 has been used as primary storage for BigData, where companies can dump logs, CSVs, JSON and other types of files that can be analyzed by Data Scientists through the use of BigData technologies. S3 has been also used for keeping backups, versions of packages and any other data that needs to remain durable and safe. Nowadays, the generation of content is enormous, specially due to the ability possessed by the users through their phones to record high quality videos and pictures. The fact that internet is broadly available and is faster than ever, companies need to have a way to store data generated by their users safely on the cloud and this is one of the biggest use cases where Amazon S3 is perfect for the job. The web server feature from S3 has been widely used not only for the distribution of images, videos, CSS files and other static content but also to serve HTML and Javascript based web pages and web applications.

The old-fashioned web applications were built in such a way that the server side technologies were generating the HTML pages dynamically. The modern web applications utilize only HTML and Javascript. The content of the pages is fetched from the servers via REST APIs. Since the technology has evolved not to need dynamically generated HTML pages, S3's web server hosting feature became extremely useful. Since S3 offers high availability, durability and is scalable, meaning that it can handle a very heavy traffic, the web server feature became the prime platform for deployment of the static HTML web applications.

A new feature that is also very attractive from S3 is the ability to integrate with other Amazon services, such as AWS Lambda, AWS Cloud Front, AWS Cloud Watch, etc. The pricing from S3 is also extremely affordable and is based on the amount of data that is stored and on the traffic to access the data (25).

S3 is another Serverless service that enables developers to focus on the core business of their application.

5.1.2.7 AngularJS

With the evolution of the web environment, Javascript became the de-facto language for the web platform, mainly due to the fact that Javascript is natively supported by the web browsers. The combination of the evolution of the hardware, of the Javascript language and the web technology in general, provided tools for developers to build rich web applications with similar features that were only available for Desktop applications. Among the great innovations provided by the new powerful web technologies, one of the biggest products of these new technologies are the Single Page Web applications (27).

The Single Page Web applications are characterized by providing an interactive user experience similar to the experience provided by the Desktop based applications. Users are presented with a web page that is loaded only once and its contents, or components are interactively changed, without the full page refresh. The assets and the content are dynamically loaded via interactions between the frontend and the backend end server in a way that is transparent to the end user. This is possible due to the Ajax technology and also due to the ability that Javascript has to change the DOM dynamically (28).

Technologies such as JQuery allowed the rapid development of interactive web pages. Through a well defined API, developers were allowed to easily manipulate the DOM elements, animate components of the web page, communicate to the server side via Ajax and to change CSS attributes from the components of the page dynamically. Once JQuery was more and more consolidated in the environment, the open source community developed JQuery based libraries dedicated to specific components, such as image carousels, date pickers, drag and drop components enabling web developers to quickly build very interactive and good looking web pages. Even though JQuery is very successful to address certain problems and to facilitate the use of the standard Javascript APIs, it does not offer a framework for a more standard structure for building web application pages. The structural part was later addressed by technologies such as AngularJS.

AngularJS is an open source application framework written in Javascript that offers an elegant and structured solution for the creation of Single Page Applications. AngularJS follows the Model View Controller architecture (29) and inherently aids developers to build structured applications where the views and the controllers are well defined and separated entities. This separation of concerns makes AngularJS applications to be prepared for testing and also encourages the developers to break down the application into individual self-contained components.

This technology relies heavily on the scope definition. Every Controller or component has its own scope. In the scope, functions and variables are defined. The variables and functions defined in the scope of a controller are referenced in the view. When a scope variable that is bound to the view has its values updated, the view is automatically updated with the values from the scope due to AngularJS digest loop. The scope variables and functions are passed down to the child components of a given controller through inheritance.

AngularJS has a dependency injection mechanism that is used to inject the dependencies into Controllers, Directives and Services. In AngularJS, views are associated to controllers. Controllers are components that respond to UI interactions such as clicks, form submissions or events. The Controllers often depend on services to perform its actions. The services are automatically injected into the controllers. Services usually have an oriented and well-defined responsibility, such as communicating to a server side REST API. Directives are reusable components that can be included into views through the HTML markup. The core directives provided by AngularJS make the connection between the Controller layer and the view.

A simplistic diagram from AngularJS that illustrates the main building blocks and their responsibilities is found in Figure 4.

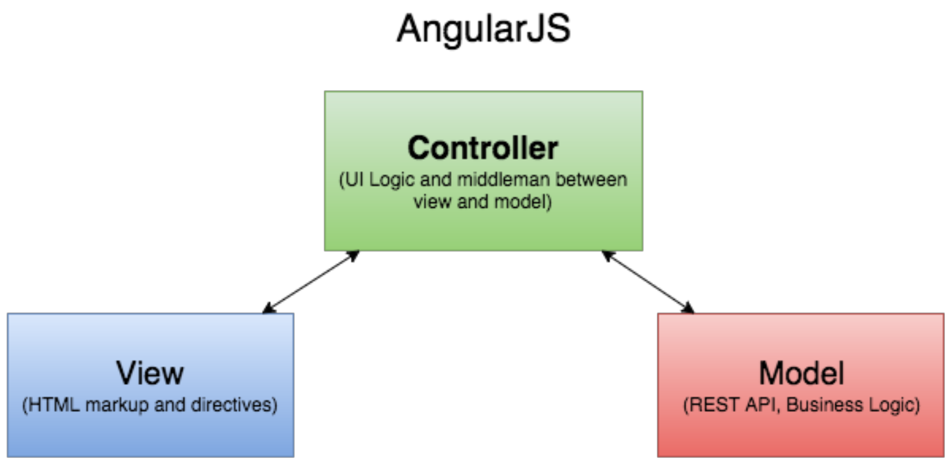

AngularJS
Controller
(UI Logic and middleman between view and model)
View
(HTML markup and directives)
Model
(REST API, Business Logic)

Figure 4. Diagram that shows the main building blocks from AngularJS.

One of the main benefits of AngularJS is the structure that is provided to the developers when creating client web applications. This structure helps to organize the application into small independent components. Also this separation of concerns helps into the creation of maintainable and re-usable code. AngularJS has become extremely popular in the web environment and has been proven to be robust and reliable.

5.2 Architecture Blueprint

The architecture from this application consists of server-side components and a game client component. The server-side components were designed to be Serverless, that is, to rely on Cloud based services where the maintenance, provisioning and scalability of the servers is managed by a cloud provider. Since the chosen cloud provider of this project is AWS Amazon, the technologies selected for this project are AWS proprietary services.

The application consists of a CMS component that is utilized by the game designers and the dialog writers to input the content of the game. The server-side APIs utilized to manage the content were also re-utilized to provide the content for the game client component. A high-level diagram from the Architecture is displayed in Figure 5.

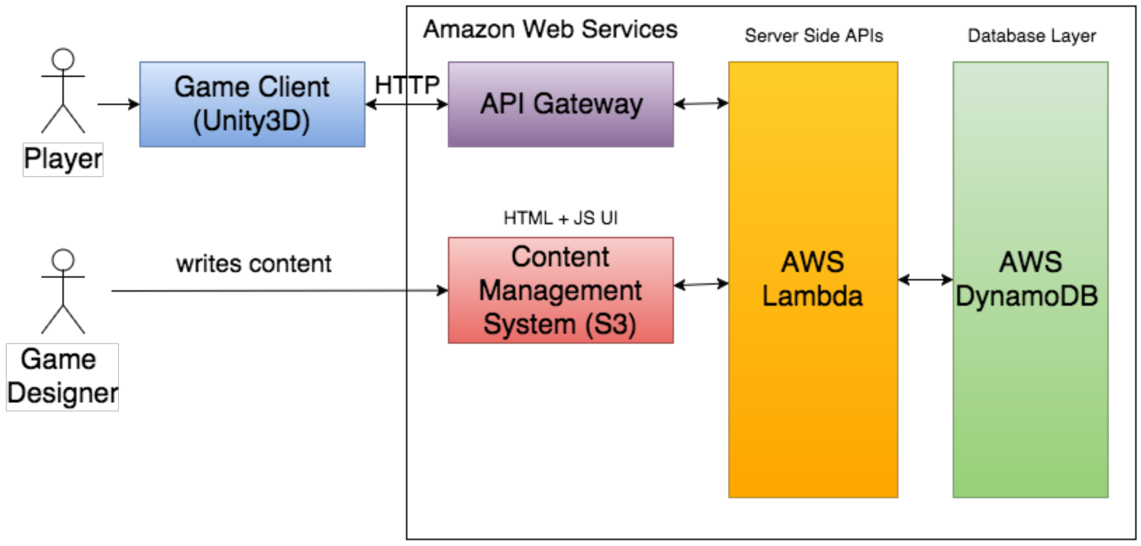


Figure 5. Game Architecture Blueprint.

The implementation of each of the components of the Architectural blueprint displayed in Figure 5 is discussed in detail in Section 6.

6 Development of the Prototype

This section describes in detail the implementation of the main parts of the prototype considering the server side and the client side.

6.1 Building Prototype with Unity3D

The development of the game was done utilizing the engine Unity 3D. In order to simplify the development of the game, every feature was built separately and then assembled together at the end.

6.1.1 Point and Click Navigation System

In an graphical adventure game based on point to click exploration, the gameplay mechanics involve around pathfinding algorithms. In this game, when the player clicks on a certain location on the game world, the main character must walk towards the selected location. A pathfind solution is then utilized to find the route between the player’s character and the selected destination. By following this route the main character will avoid the obstacles and will arrive to the required location, via the fatest path. Pathfind algorithms can become complex, specially in 3D environments. Even though there are several pathfinding algorithms that can be utilized in Unity 3D, the engine provides a built-in path finding system that works well and is very accurate. Unity’s built-in path finding system is called Navigation system.

The Navigation System from Unity is utilized in order to create the point to click control mechanics in the game. The Navigation System is composed of the Navigation Mesh, the NavMesh Agent and Off-Mesh Link (not utilized in this game).

The Navigation System uses the meshes from the game objects in the scene that are marked as “Navigation Static” in order to determine the walkable places of the game scene (30). Hence, the Navigation Mesh is defined from the geometry of the level defined in the scene. In order to create a Navigation Mesh, the user needs to execute the required steps in Unity editor. The process of computing the geometry of the meshes in the game scene is called “baking”. The baking process produces the

Navigation Mesh data structure, which is utilized by the engine to define which places of the scene are walkable and where the obstacles are. Once the Navigation Mesh is baked, the walkable area is displayed in blue, as illustrated in Figure 6.

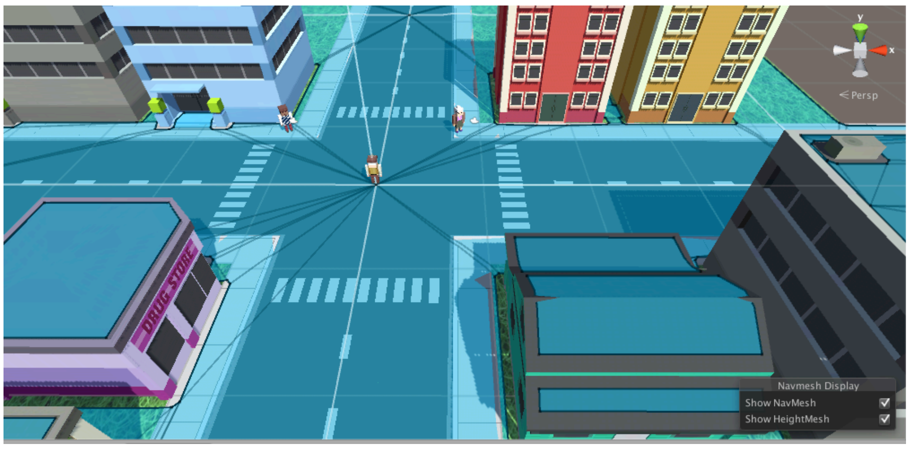


Figure 6. Navigation Mesh is displayed as the “blue” area.

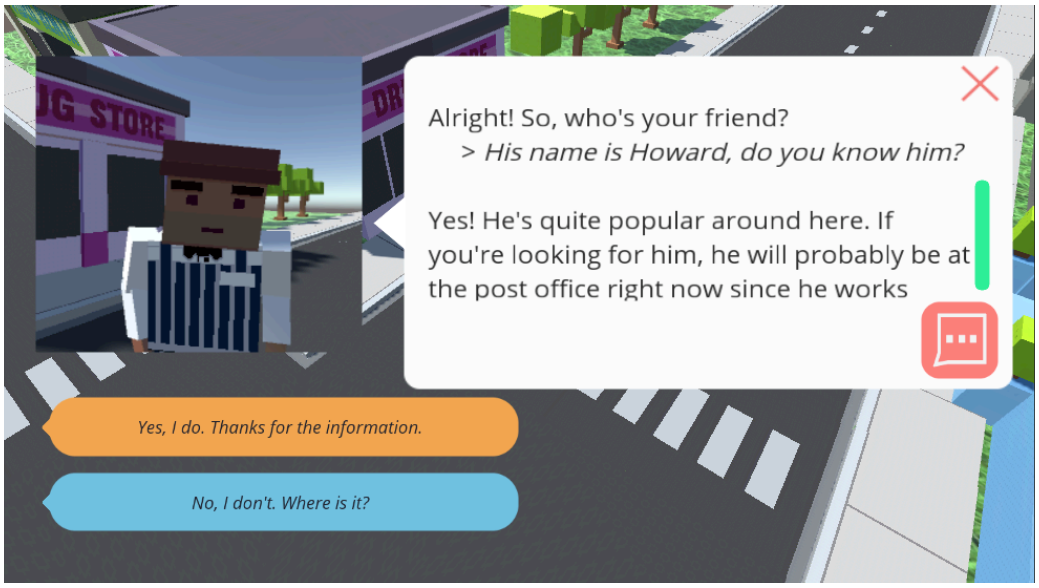
In order to use the Navigation Mesh the game objects must have assigned a component called NavMesh Agent which is responsible for the path finding and for moving the game object towards the destination. In this component the user can define the speed that the game object will move among other relevant parameters. The movement of the character in the map is set into motion when the player selects a location on the screen. After the location is selected a custom script gets a reference to the NavMesh Agent component from the player character’s game object and sets the destination location. When the destination is set, the path is generated and the NavMesh Agent component starts to move the player character to the target direction. The same mechanics apply to the NPCs that are ramdomically walking in the game scene.

6.1.2 Dialog Tree UI

In the latest versions from Unity 3D, the team responsible for maintaining the engine has re-build the GUI system, enabling the creation of rich user interface components, such as scrollable text areas, animated buttons and other UI elements that allow the game developers to create appealing interfaces for their games.

Since this game is mostly about Dialogs between the in-game characters and the player, the UI must adapt dynamically according to the length of the content. Buttons that will alow the player to take decisions during the dialog must also be provided.

The main dialog balloon where the NPC speech is displayed, illustrated below in Figure 7, is composed by a Canvas element along with a ScrollRect component and a Text view. The text expands in accordance to progress of the dialog. When the translation button is pressed, the text speech bubble then displays the translated text. The player can then switch back and forth between the target and the translated language by toggling the translation button. Once the text length is bigger than the area in the NPC’s speech bubble, a scrollbar is displayed and the content is automatically rolled upwards, in order to show the end of the dialog. Once the dialog text is completely displayed the player may use the scrollbar to read parts of the text that are hidden upwards.


Alright! So, who's your friend?
> *His name is Howard, do you know him?*

Yes! He's quite popular around here. If you're looking for him, he will probably be at the post office right now since he works

JG STORE
DR
Yes, I do. Thanks for the information.
No, I don't. Where is it?

Figure 7. Dialog UI. NPC speech bubble and the possible options for the Player to respond.

Buttons with different colors are displayed on the bottom of the Dialog UI. These buttons are only visible when the NPC has concluded its speech. The Buttons have their content dynamically set according to the available answers defined in the Dialog tree. Once the player selects an answer the next node of the dialog is then displayed in the NPCs speech bubble. The Dialog UI is displayed on Figure 7.

When the player finishes the dialog with the NPC the Dialog UI is set invisible and the player is again allowed to navigate in the game scene.

6.1.3 Functionality

Every game has a core loop that defines how the game is played, how the player progresses and when the player reaches the end of the game. Since this game resolves around the dialogs between the player and the NPCs, the progress of the game is based on the dialogs.

In order to identify a player and keep track of his progress, the game utilizes the device identifier in order to identify the player. When the game starts, the game client fetches the device identifier from the phone and sends to the server for authentication. The server then verifies if that device identifier is already registered in the database or not. If the devide identifier is found in the database, the server retrieves the progress of the player and returns that information to the game. If the device identifier is not found, the new player is then registered in the database and the server instructs the game to start the story from the beginning.

Since the game is based on a sequence of Quests, after the player is authenticated, the server returns the current Quest that the player needs to complete in order to advance in the game. After the Quest is loaded, a introduction message from the Quest is displayed and the player can then start to explore the game world and to interact with the NPCs. If one Dialog with a NPC is marked as “Completes the Quest”, once the player finishes the dialog, the Quest completion message is displayed and then the game client asks the server side for the next Quest. The core loop of the game Quest system is displayed in Figure 8.

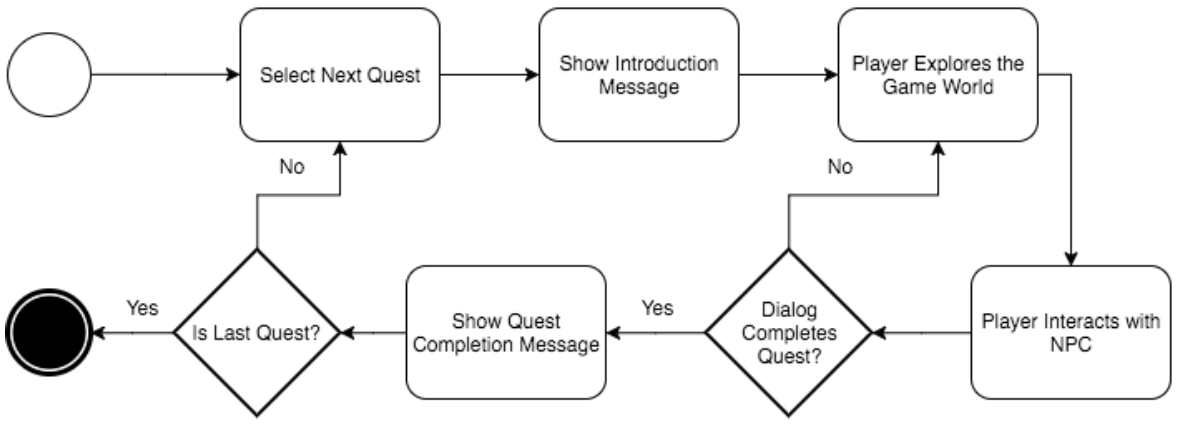

Select Next Quest
Show Introduction Message
Player Explores the Game World
No
No
Yes
Is Last Quest?
Show Quest Completion Message
Yes
Dialog Completes Quest?
Player Interacts with NPC

Figure 8. Core Loop of the Quest System.

Even though the Quest system is simple, it can be used to build complex stories since the number of NPCs involved is unlimited and the number of Quests can be also unlimited.

Once the player starts to interact with the NPC, by tapping on it, the game client retrieves the NPC identifier and the current Quest Identifier and asks the server side for a Dialog based on the given identifiers. If the server fails to find a Dialog for that given NPC in the given Quest, it returns the default Dialog, if any default Dialog is registered, or it simply returns nothing. If a Dialog is found and is returned from the server, the game client reads the Dialog and enables the Dialog UI and starts the Dialog engine.

The Dialog engine starts by selecting the next Question or Statement that will be spoken by the NPC. Next, the Dialog engine starts to display the NPC speech in the NPC speech bubble. In this game, the animation pattern encountered in RPG games was followed when displaying the text. This pattern consists of an animation where the characters that compose the words in the Dialog text are displayed one by one onto the UI. Once all the text from the Dialog is displayed, the engine then fetches the available answers from the Node of the Dialog tree that is associated to the question or statement that was posed by the NPC. The structure of the Dialog tree is described in detail in Section 6.2.1.

Once the Dialog Node is completely displayed on the NPC speech bubble, the options to answer the NPC are then displayed in the bubble buttons with different colors at the bottom of the screen as displayed in Figure 7. Based on the selected option by the player, the engine proceeds to select the next Node from the Dialog tree. The cycle then starts again until there are no more options to be displayed to the player and the Dialog ends. The Dialog Engine loop is illustrated on the diagram in Figure 9.

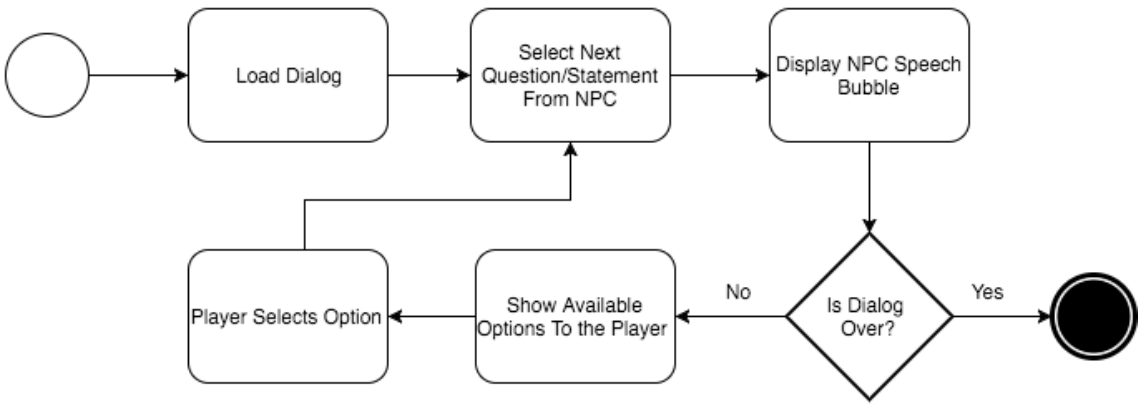

Load Dialog
Select Next Question/Statement From NPC
Display NPC Speech Bubble
Is Dialog Over?
Yes
No
Show Available Options To the Player
Player Selects Option

Figure 9. Core Loop of the Dialog System.

After the core Dialog mechanics were stabilished, final touches to the exploration mechanics were made. The Camera functioning was defined by a custom C# script that takes the current location of the main player's character and applies to the Camera's position, adding a pre-determined offset distance. As a result, the Camera "follows" the main character as displayed in Figure 10.

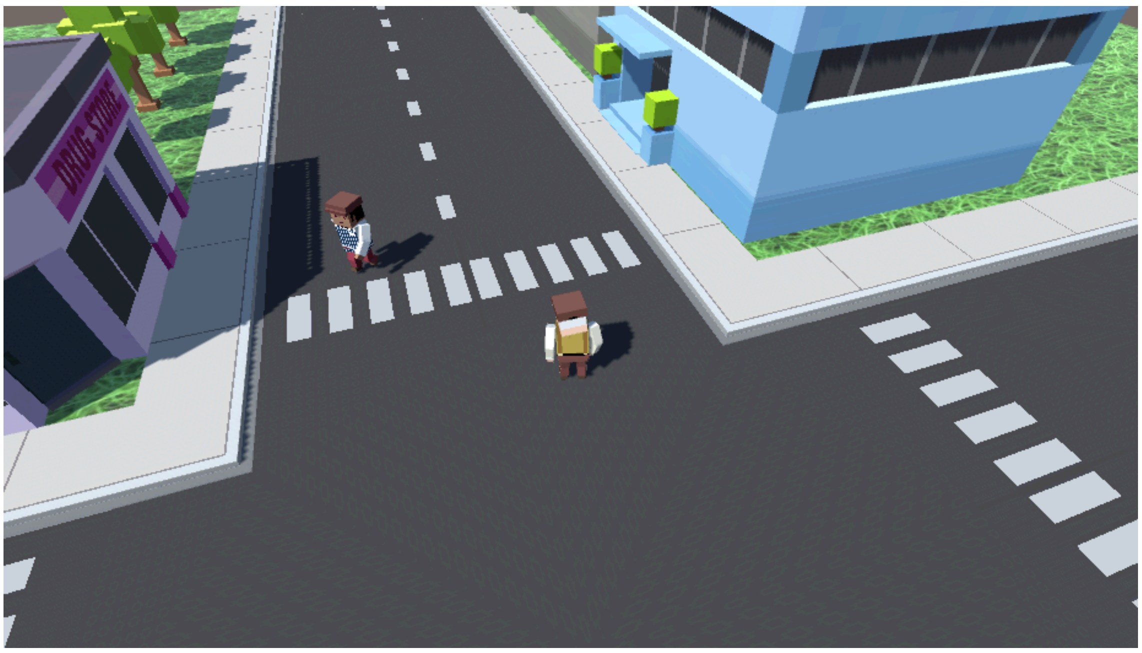


Figure 10. Screenshot of the exploration mode in the final Game Client prototype.

In order to make the NPCs to walk around in the game world, the same path finding feature from Unity 3D that was utilized for the Point to Click control, presented in Section 6.1.1, was applied to the NPCs.

Given that the prototype and the core mechanics are simple, and taking into consideration the features that Unity 3D engine provides, the code base from the game client remained small, with approximatelly only 20 C# classes that translate into components that are added to the Game Objects.

6.2 Building Server Side

The server side components of the game consist of the Database and the APIs that are utilized by both the CMS and the Game Client. The technologies utilized to implement the server side are discribed in Section 5.1.2. Each component of the server side is described in detail in the following sub-sections.

6.2.1 Database Layer

The Database layer of the server side is supported by the technology AWS DynamoDB. In order to simplify and re-utilize the base functions that are utilized to fetch, write and delete data from the DynamoDB tables, the key “ID” was defined by convention as the Primary Key index. Therefore, every table in the schema has the “ID” field as the Primary Key. Figure 11 displays the database entities.

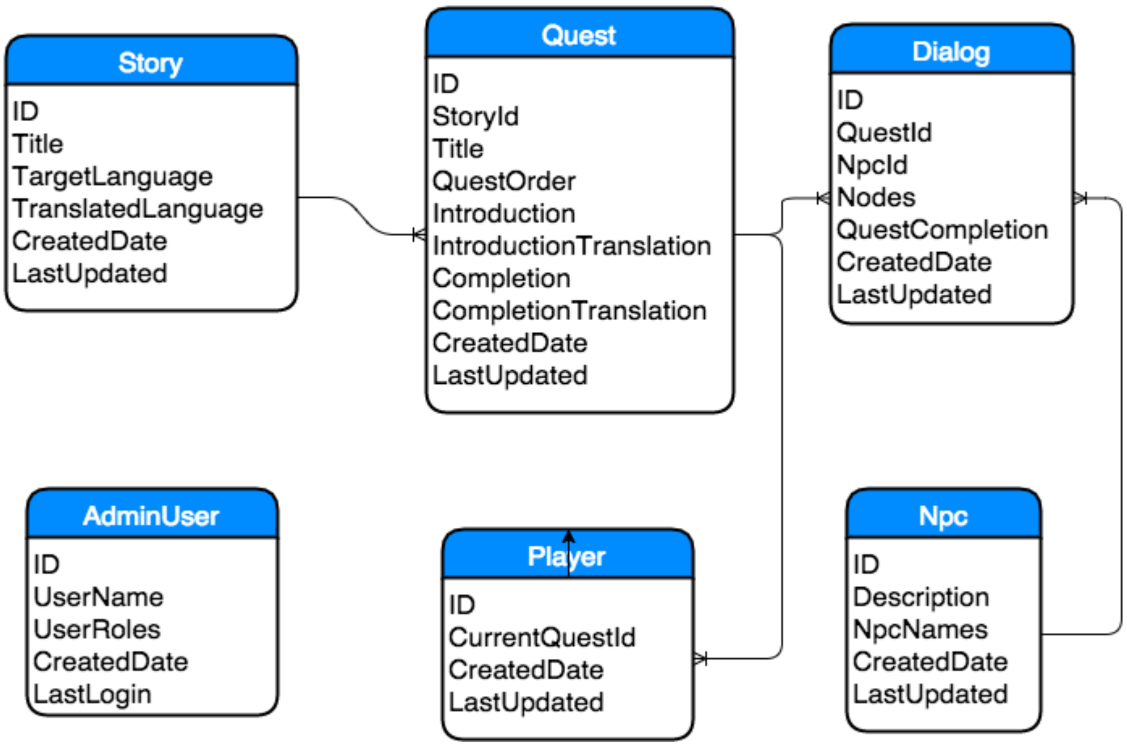


Figure 11. Game Database tables (entities) in AWS DynamoDB.

The database schema of the game is fairly simple. Most of the entities have a one-to-many relationship to one another. The entities and their contents are described as follows:

- **NPC**: Every character that interacts with the player is registered in this entity. The Description field is used to describe where the NPC is in the world of the game and what the NPC is doing. The “NpcNames” field is utilized to provide different names for the NPC, depending on the language that the Game is played, since traditional names are also part of learning a language.
- **Story**: Each game story is stored here. A Story has a target language, which is the language that the player is aiming to learn, and a translated language, which is the language that the player will utilize to resolve his doubts about what was said in the target language. The “Title” field is self-explanatory.
- **Quest**: During the story, the player will solve Quests in order to progress in the game. Each quest is a task that the player needs to complete. Once a Quest is completed, the player advances to the next Quest. When the player completes all the Quests, the game is finished. Each Quest is

connected to a parent Story through the StoryId field. When the player starts a Quest, the “Introduction” text is displayed, showing the instructions of the Quest. When the player finishes the Quest, the “Completion” text is displayed. Both the “Introduction” and “Completion” fields have their respective “Translation” texts that will be used to help the player. The “QuestOrder” field holds the order of the Quest in the given Story.

- **Dialog**: The Dialog entity is one of the most important entities in the game since it holds the conversations that will take place in the game between the player and the NPCs. Each Dialog is connected to a Quest through the “QuestId” field. A Dialog is also assigned to a NPC via the “NpcId” field. The “QuestionCompletion” field is a Boolean value that tells whether that Dialog completes the Quest or not. The Nodes are a tree of questions and answers that compose the conversation. The Nodes structure will be described in more detail in this sub-section.
- **Player**: The Player entity has the information about each player of the game. The “CurrentQuestId” field denotes the current quest that the player is assigned to. The ID field is the Device Id.
- **AdminUser**: This entity is utilized by the CMS. Each user has a set of roles. The roles dictate which actions the user can perform in the CMS system. The other fields from this entity are self-explanatory.

The Nodes utilized to compose the conversation in the Dialog entity have their structure illustrated in Figure 11.

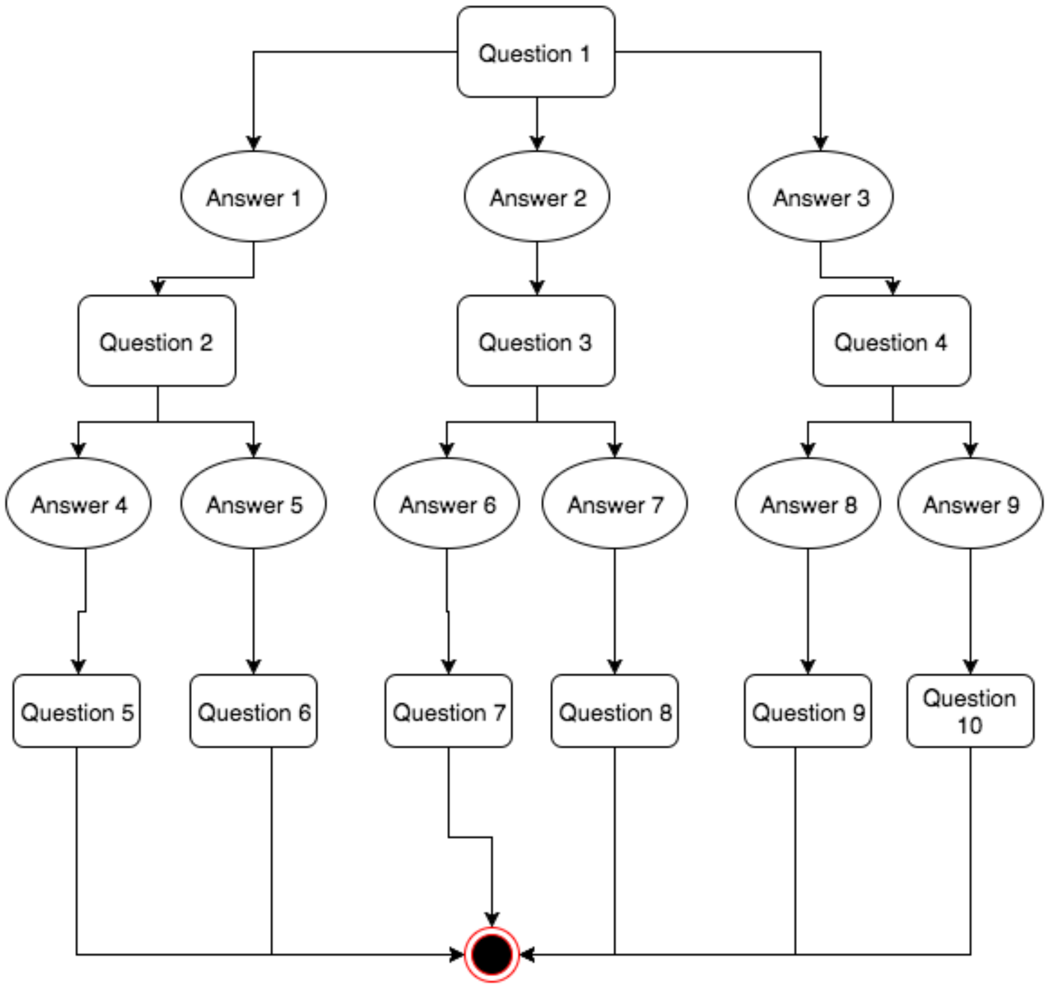


Figure 11. Dialog tree structure is composed by Question and Answer Nodes.

In every Dialog tree there is the root Node, which is the first question or statement that is spoken by the NPC. Usually it is a greeting or a question such as “Hello there!” or “How can I help you?”. The player then is presented with answers. Each answer will lead to a next statement or question, which in turn will have different possibilities for answers. The Dialog tree continues indefinitely until the last question or statement Node does not have any answer or option for the player to select.

This simple structure provides means for the writers and game designers to create very long dialogs in between the player and the NPC. Each Question and Answer Node have translations that can be used to help the Player. The data structures utilized for each Node of the tree are illustrated in Figure 12.

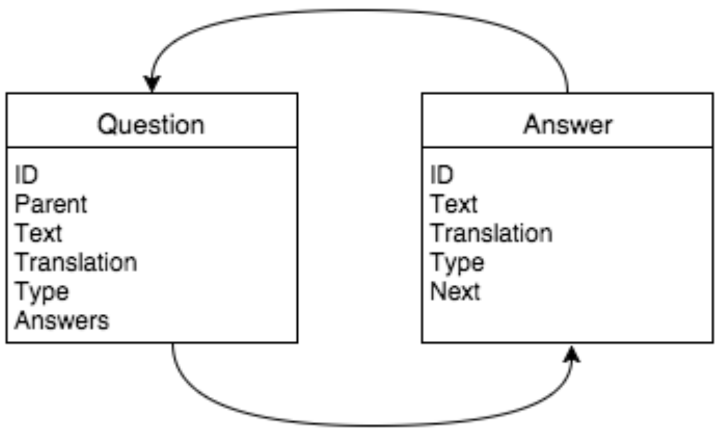


Figure 12. Data structures utilized to build the Dialog trees.

The Question entity has the “Parent” field that is connected to the parent Answer Object. If the “Parent” object is “null” (empty), that Question is the root Node of the Dialog tree. Every Question has the “Text” and the “Translation”. The Answers field has the child Answer Nodes of that Question.

The Answer entity has the “Text” and “Translation” fields to store the content. The “Next” field points to the Question that will follow that answer.

This structure for the Dialogs is fairly simple and yet is very powerful since it enables the content creators to create complex and deep Dialog trees.

6.2.2 Server Side APIs

The Server Side APIs were implemented via the AWS Lambda technology. All the APIs resolve around providing or updating the data from AWS DynamoDB database. As displayed on Figure 13, some of the APIs are exposed to the Game Client through HTTP API definitions in API Gateway, which are public by default and so, they can be accessed by anyone.

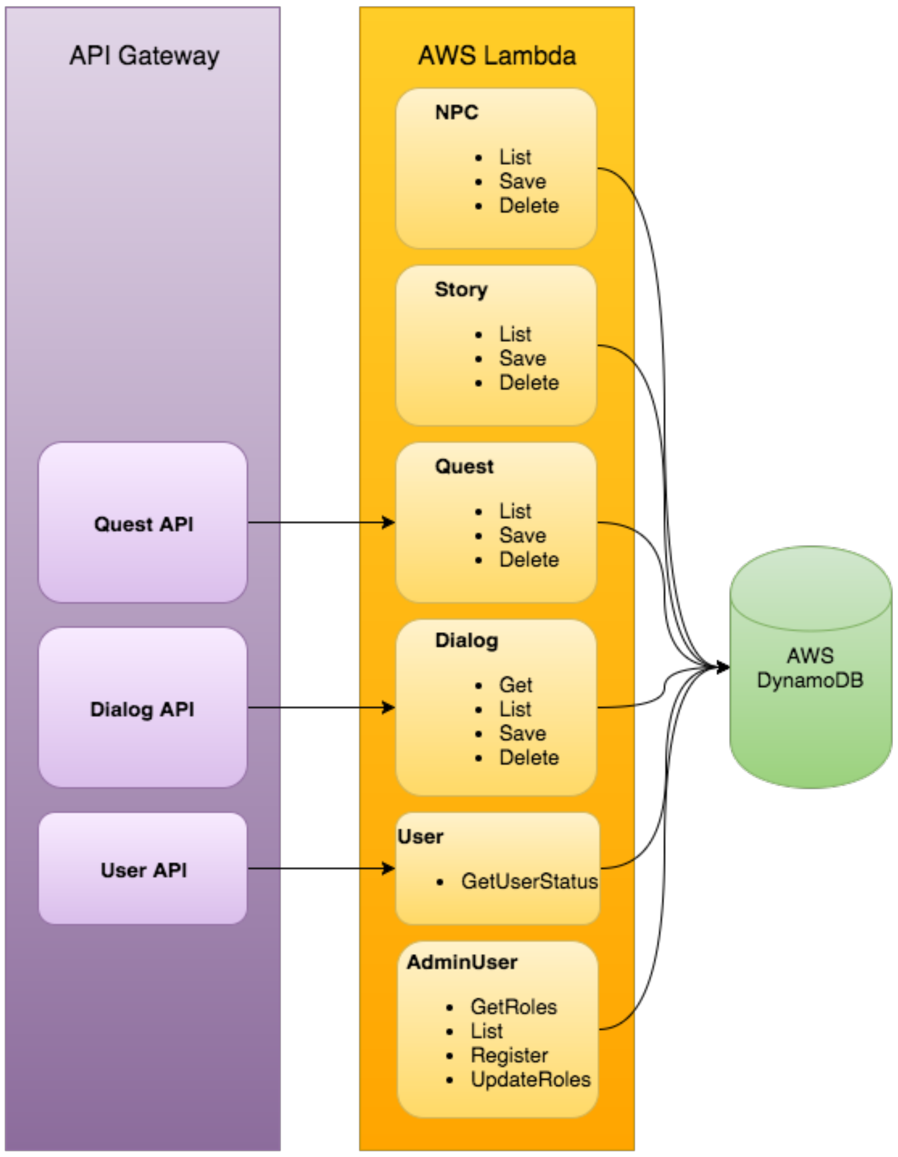


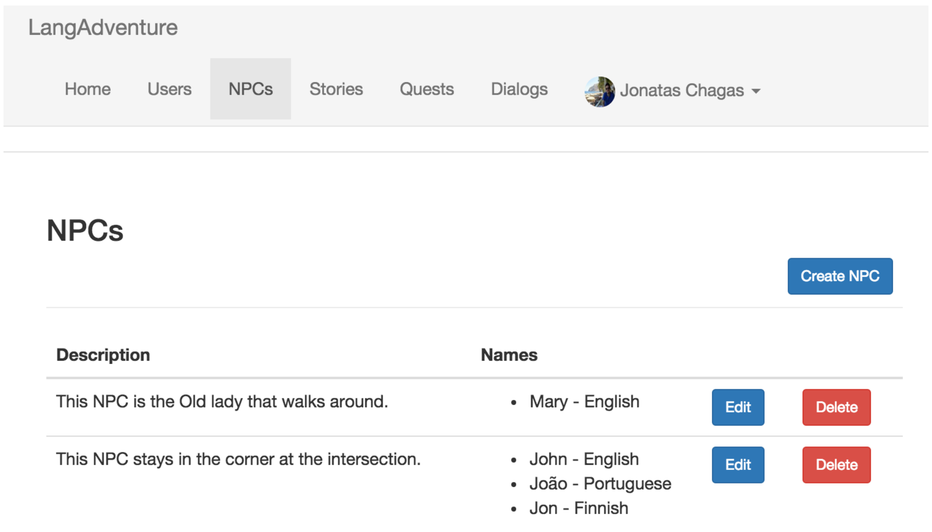
Figure 13. Blueprint of the Server Side APIs.

All the other AWS Lambda functions, that are not behind the HTTP layer defined in API Gateway, can only be accessed and executed if the caller is authenticated. The authentication is done via AWS Cognito.

6.2.3 CMS

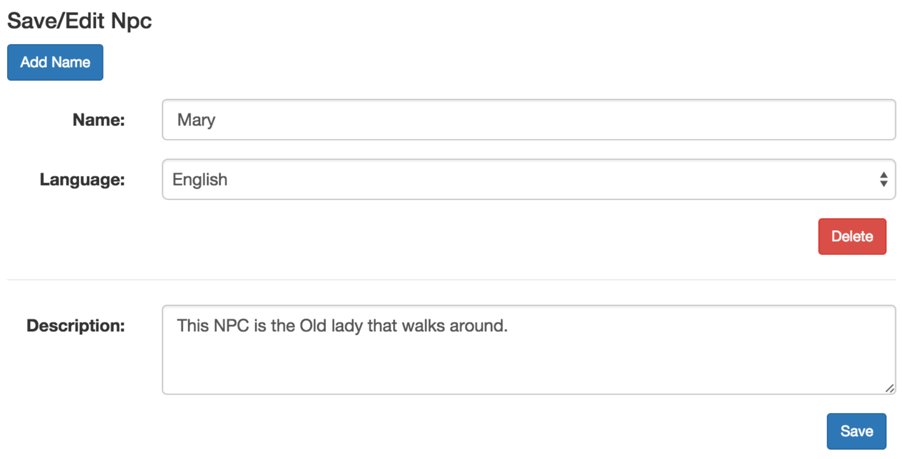
The CMS system is a HTML + Javascript (AngularJS) application that is deployed and destributed via the AWS Simple Storage Service (S3) web server feature. This tool is utilized to create the content of the game, such as NPCs, Quests and Dialogs. The CMS also has a very simple access control system that allows to set permissions to the users by adding and removing roles to their profile. The CMS interface offers functions to manage NPCs, Stories, Quests and Dialogs. The CMS executes the AWS Lambda functions by utilizing the AWS Javascript SDK.

The management of the NPCs consists of the creation, updates, deletion and listing of the available NPCs in the game, as displayed in Figure 14 and Figure 15.



Description	Names
This NPC is the Old lady that walks around.	• Mary - English
This NPC stays in the corner at the intersection.	• John - English • João - Portuguese • Jon - Finnish

Figure 14: CMS screen that shows the registered NPCs in game.

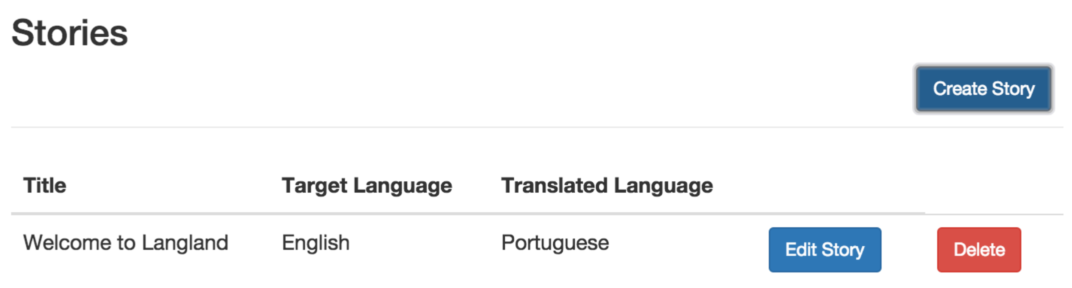


Name: Mary
Language: English
Description: This NPC is the Old lady that walks around.

Figure 15: CMS UI to create new NPCs in the game.

As displayed in Figure 15, the NPC creation UI allows to define multiple names for the NPC depending on the selected language.

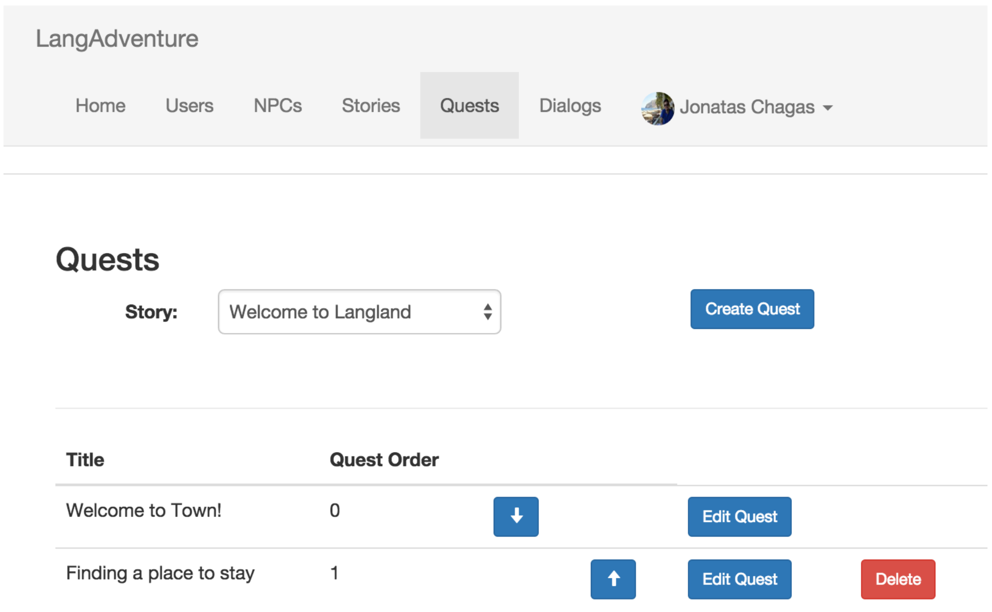
The Stories are very simple entities that are used to aggregate the Quests and the Dialogs as a unit. When registering a Story in the CMS, the user can select the target language and the translated language, as displayed in Figure 16.



Title	Target Language	Translated Language		
Welcome to Langland	English	Portuguese	Edit Story	Delete

Figure 16: CMS UI to manage Stories.

The Quests management UI allows the creation, changes, deletion and the definition of the order of the Quests within the Story as displayed in Figure 17. To create a new quest, the game designer needs to first select a Story.

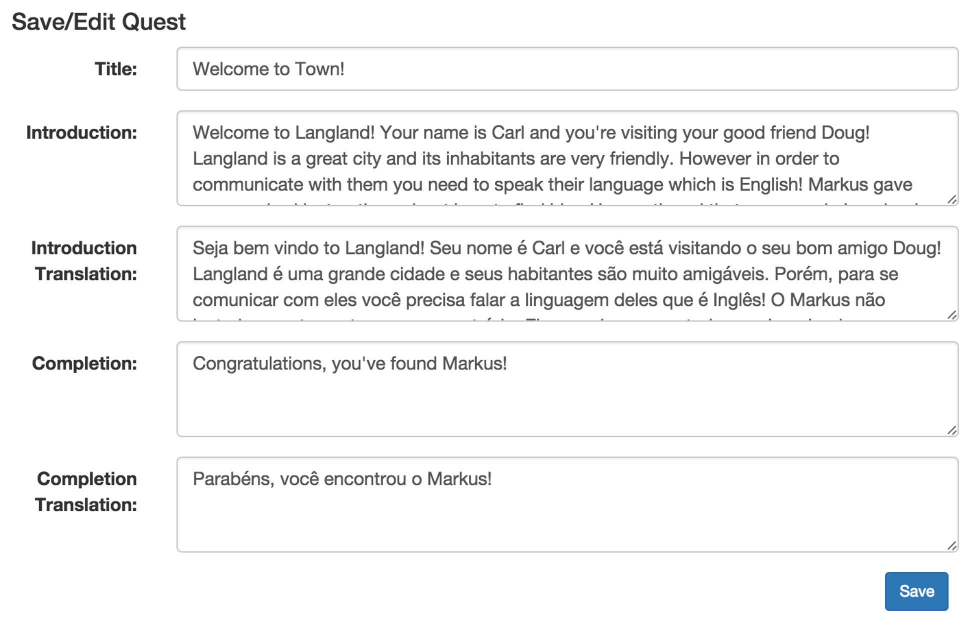


Story: Welcome to Langland

Title	Quest Order			
Welcome to Town!	0	⬇	Edit Quest	
Finding a place to stay	1	⬆	Edit Quest	Delete

Figure 17: CMS UI to manage Quests.

When creating Quests in the management UI, the user must input an Introductory message to the Quest and a Completion message, as displayed in Figure 18.

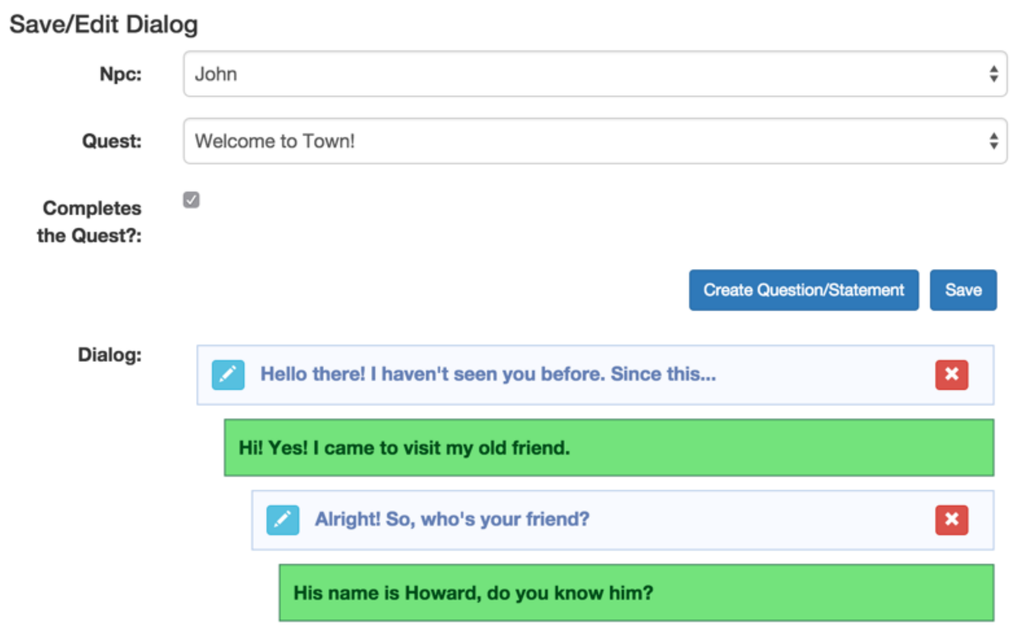


Save/Edit Quest

Title:	Welcome to Town!
Introduction:	Welcome to Langland! Your name is Carl and you're visiting your good friend Doug! Langland is a great city and its inhabitants are very friendly. However in order to communicate with them you need to speak their language which is English! Markus gave
Introduction Translation:	Seja bem vindo to Langland! Seu nome é Carl e você está visitando o seu bom amigo Doug! Langland é uma grande cidade e seus habitantes são muito amigáveis. Porém, para se comunicar com eles você precisa falar a linguagem deles que é Inglês! O Markus não
Completion:	Congratulations, you've found Markus!
Completion Translation:	Parabéns, você encontrou o Markus!

Figure 18: CMS screen that allows the creation of a Quest.

The CMS also provides an UI for the creation of the Dialog trees as displayed in Figure 19.



Save/Edit Dialog

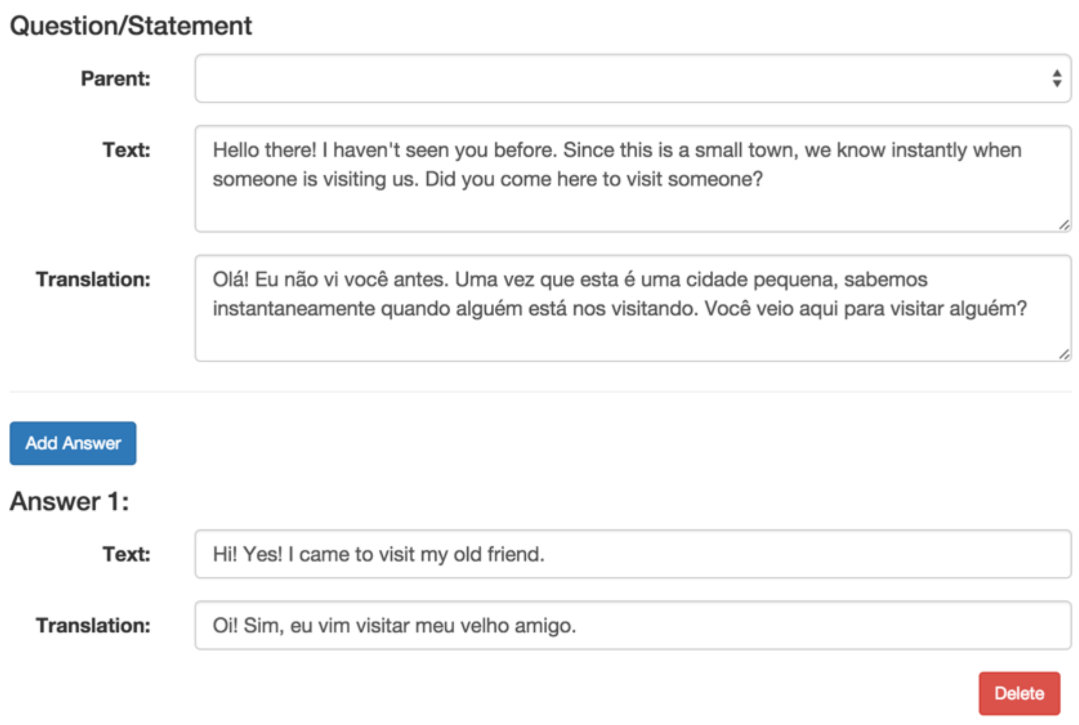
Npc:	John
Quest:	Welcome to Town!
Completes the Quest?:	✓

Dialog:

- Hello there! I haven't seen you before. Since this...
  - Hi! Yes! I came to visit my old friend.
    - Alright! So, who's your friend?
      - His name is Howard, do you know him?

Figure 19: CMS screen that displays the structure of a Dialog Tree.

The blue Nodes of the Dialog tree UI are the Questions/Statements and the green Nodes are the possible Answers that can be selected by the player during the Dialog. Every Question/Statement and its Answers are created individually and then added to the Dialog tree as displayed in Figure 20. The Interface also allows the dialog tree nodes to be edited and changed accordingly.



Question/Statement

Parent:	
Text:	Hello there! I haven't seen you before. Since this is a small town, we know instantly when someone is visiting us. Did you come here to visit someone?
Translation:	Olá! Eu não vi você antes. Uma vez que esta é uma cidade pequena, sabemos instantaneamente quando alguém está nos visitando. Você veio aqui para visitar alguém?

Answer 1:

Text:	Hi! Yes! I came to visit my old friend.
Translation:	Oi! Sim, eu vim visitar meu velho amigo.

Figure 20: Question/Statement creation/edition UI. The Answers are appended to the Question/Statement.

When creating the Question or Statement, the game designer must select the Parent Node in the dialog tree. If no Parent is selected, the root node of the Dialog tree is replaced. The Question/Statement CMS UI allows the definition of multiple Answers for each question or statement in the dialog tree.

7 Future of Project

This section describes the new features that can be implemented in the future in order to make the game more appealing to the players and to enhance their learning experience.

7.1 Puzzle Mechanics and Textbook Exercises

Even though the core mechanics of the game are into place, there is still room for improvement. Since the primary goal of the game is to help the players in the learning process of a language, different game mechanics can be introduced in order to address other aspects that were not covered by the Dialog mechanics.

When studying foreign languages, exercises are very important since they provide a means for the students to put the grammar into practice. Traditionally, in text books, exercises are composed and presented in different ways in order to be attractive and entertaining to the students, ultimately aiming to ease the learning process. These exercises include: fill in the blanks, scrabble, connect the matching items, select the right picture for the given word and sentence building. These traditional exercises can be transformed into game mechanics and included as part of the game as puzzles.

In certain RPG or graphical text adventure games, puzzle mechanics are included in order to present challenges to the player and to bring the game play experience beyond dialogs. These puzzles challenge the player’s intelect and when solved, bring satisfaction, a feeling of accomplishment and also give a sense of progression in the game. Puzzles usually consist of re-arranging UI elements to match a certain pattern or to solve the given problem.

The puzzle mechanics and the textbook exercises can be easily merged and can be used as part of the game to both challenge the player and also contribute to the learning experience. Puzzle mechanics can be also used as a battle mechanism to challenge the progress of the player in the adventure.

This feature is the first on the list of the improvements that need to be addressed in order to make the game more interesting. Since the puzzle mechanics can be complex to implement and require more planning, they did not fit in the scope of this initial prototype.

7.2 Voice Over Text

One of the most important abilities that must be acquired for foreign languages students is the ability to listen and comprehend people speaking on the foreign language. To hear native speakers pronouncing the words also helps in the learning of the proper pronunciation of the words and helps the students to assimilate the different sounds of the language. Given the importance and the benefits that are brought to the student by the voice over text, this feature must be implemented along side the dialog trees in order to improve the learning experience.

When considering implementing this feature, the CMS system must be changed so it will accept audio files along with the text input. Once the audio files are included in the dialog, they should be uploaded to AWS Simple Storage Service (S3). After uploaded to S3, the path of the audio file would be stored in the database and would be provided to the game client via the server API. Once the game client has access to the path of the audio file in S3, it can use its public URL to stream the audio to the game client. This approach would require more internet bandwidth available when playing the game but would not require the audio files to be permanently stored in the game client. Once the infrastructure to suport voice over is ready, the next challenge is to produce the content.

Even though it is relatively simple to play audio files in Unity 3D, this feature did not fit in the scope of this prototype and hence has to be postponed to the future.

7.3 Interaction with Objects and Improved UI for Translation

When learning languages, it is important to expand the vocabulary and to learn how objects are called. Since the main objective of the game is to immerse the foreign language learner into an environment where the language is used and spoken, the inspection of ordinary objects plays a big part in such a environment, especially in terms of vocabulary building. These objects would have to be presented as images and their given names would have to be displayed in the language of study. Also, the objects could be part of the game story or they could be just ordinary objects.

In order to implement such a feature, a catalog of objects must be implemented in the CMS. Once the objects are registered in the CMS, they must be placed in the game world. In the game client, the interaction mechanics and also the UI part to display the objects also must be implemented. This feature is not as challenging as the voice over text or the puzzle mechanics, in terms of implementation, however it requires a large amount of content and hence did not fit in the scope of the prototype.

The UI controls in the Dialog engine could be improved and developed further. Features such as word highlighting, Interactive dictionary and glossary of words could be included in order to enhance the experience of the player. Even though Unity3D UI components are more advanced than before, controls such as word highlingting are not implemented in the engine yet, therefore some work must be done in order to implement such a feature. Also, to build an interactive dictionary and glossary of words would require changes both in the game and in the CMS. Given the complexity of these features, they were posponed for the future versions of the prototype.

7.4 Multiplayer

Nowadays, some of the most successful mobile games attribute their success to the fact that the content of the game is generated by the players of the game via an online collaborative multiplayer environment. Mobile multiplayer games engage the players due to community values, such as friendship, competitivity and commitment. Since the content is created by the players themselves, the game becomes a platform where players interact and contribute for the growth of the community in-game.

In a game about learning foreign languages, students that study the same foreign language can interact and practice the language among themselves by chatting or challenging themselves in multiplayer puzzles. Such an environment also allows native speakers of the language to join and help the students in the learning process by chatting with them and also adding content to the game.

In order to make a multiplayer game, multiple technical challenges need to be addressed and the way the game works must be re-designed. Even though this feature would considerably improve the player experience and would influence positively his learning process, it requires a considerable amount of time to be developed. However, despite the required efforts, this feature should be developed in the future.

8 Conclusion

The learning process of foreign languages requires intensive study through memorization and practice. Even though various sources of content for practice of the language are available in the internet, a complete immersion in an environment where the language is spoken is far more effective then self-study. Since it is not always possible for students of foreign languages to travel and live in the countries where the language of study is spoken, different methods and approaches can be attempted in order to emulate the immersion of the student in such ennviroment.

Certain genres of games entertain the players by providing a virtual world where the player is part of a story and participates into dialogs with characters in the game. One main representative of this genre is the Graphical Adventure Game, which provides a full immersion of the player in the story of the game through dialog mechanics and exploration of a virtual environment.

Games are slowly becoming part of the education system. Educators have spotted the benefits of gamification in the learning process. Among the main benefits of including games in education is the fact that the students effectively study and learn while being entertained. One of the factors that drove the adoption of games into the educational system was the fact that mobile technology became more affordable and accessible to students. Due to the advanced technical capabilities of the mobile devices, mobile games became increasingly popular since the players can enjoy highquality games anywhere at anytime.

An aspect of learning that was enhanced through the advance of the mobile technology is MicroLearning. The concept of MicroLearning basically consists of short sessions of study spread throughout the day. As an outcome of MicroLearning, the students can optmize their study time by studying while doing mundane tasks such as commutes, waiting on lines, etc.

Combining the study of foreign languages to an immersive game such as a Graphical Adventure utilizing the mobile platform and taking advantage of concepts such as MicroLearning, create an effective tool for the study of foreign languages.

In order to build a prototype of such a game, the Unity3D engine was utilized to build the game client. During the development of the game client it was discussed how to utilize the engine to build the Point To Click navigation and the Dialog system.

In order to create and store the content of the game a CMS was built utilizing a Serverless architecture, where all the components of the application are managed and mantained by Amazon Web Services. By having a Serverless architecture, time from configuring an infrastructure and maintaining servers is reduced and the developers can focus on building the application.

For the database, AWS DynamoDB was utilized. DynamoDB is a NoSQL key-value store fully managed by AWS. The application server layer was developed in NodeJS, based on the AWS Lambda platform, that provides fully management of the infrastructure used to execute the application server code. The AWS Lambda service allows the developer to pay only for the utilization of the functions, as opposed to the traditional model where the whole period of utilization of the server, even if idle, is charged from the developers.

The CMS web client application was developed in Javascript through the utilization of a popular framework called AngularJS. The CMS web client application is hosted into AWS Simple Storage Service (S3) through the utilization of the web server feature.

By utilizing these technologies the whole server side is Serverless, meaning that there is not a single specific server maintained by the developer, and all the deployment of the server side is completely managed by Amazon WebServices.

The goal of the thesis was to build a prototype of a Graphical Adventure Game designed for foreign language students by utilizing Unity 3D as the game engine and to utilize the new Serverless architecture.

During the beginning of the project a fairly amount of time was spent to understand the inner workings of the Serverless components and to have an idea of how to unite those components in order make them work together. Through research, an understanding of the nature of the components of the server side was formed. The Serverless opensource project, that provides tooling for developing and deploying AWS Lambda functions, was utilized in the development of the APIs.

Once the overall structure of the server side APIs was ready, the next challenge involved harmozing the technologies that compose the CMS, which are mainly AngularJS, AWS Javascript SDK and the AWS Cognito technology. The main idea was to authenticate the users using AWS Cognito and once successfully authenticated, call the AWS Lambda functions, running on the server layer, straight from the AngularJS service layer. In order to achieve that, the documentation from the AWS Javascript SDK had to be thoroughly studied. However, once the first blocks started to connect, the rest of the assembly process of the basic structure of the application was easily carried out.

After setting up both the server side and the client side base structure, the development of the CMS and of the server side APIs was fairly straightforward. As the last part of the implementation, once the server side APIs were in place, the development of the game client started. The communication in between Unity 3D and the server side APIs was done via HTTP through the use of the API Gateway, which is simply a HTTP interface in front of the AWS Lambda functions.

Once the game client and the server APIs were glued together, the architecture proved to be effective in terms of maintainability and also agile, since the whole solution was developed in a month. Overall, the Serveless architecture works really well for rapid development of applications and saves maintenance and management time. Unity 3D proved to be powerful and also very easy to utilize and to learn. Combining both technologies builds a strong architectural case that can be utilized for other solutions.

The initial versions of the prototype were demoed to students of foreign languages and their feedback was overall positive. However, the students have expressed the need for the development of additional features that were listed in Section 7. Therefore, the development of this project is to be continued.

References

1. Huang WHY, Soman D. Behavioural Economics in Action @ Rotman – A Canadian Research Hub. [Online].; 2013 [cited 2016 03 26. Available from: http://inside.rotman.utoronto.ca/behaviouraleconomicsinaction/files/2013/09/GuideGamificationEducationDec2013.pdf.
2. Fernández-Vara C, Osterweil S. The Key to Adventure Game Design: Insight and Sense-making. Article. Cambridge: Massachusetts Institute of Technology, Department of Comparative Media Studies; 2010 Oct. Report No.: ISBN/ISSN.
3. Bonnington C. Wired. [Online].; 2015 [cited 2016 03 26. Available from: http://www.wired.com/2015/02/smartphone-only-computer/.
4. Peterson S. gamesindustry.biz. [Online].; 2013 [cited 2016 03 26. Available from: http://www.gamesindustry.biz/articles/2013-08-09-mobile-game-market-growth-opportunity.
5. Pereira V, Sousa T, Mendes P, Monteiro E. Evaluation of Mobile Communications: From Voice Calls to Ubiquitous Multimedia Group Communications. Article. Coimbra: University of Coimbra, Department of Informatics Engineering; 2004.
6. Gabrielli S, Kimani S, Catarci T. The Design of MicroLearning Experiences:A Research Agenda(On Microlearning). Article. Rome: Università di Roma ”La Sapienza”, Dipartimento di Informatica e Sistemistica; 2006.
7. Hug T. Micro Learning and Narration: Exploring the possibilities of utilization of narrations and storytelling for designing of “micro units“ and didactical micro-learning arrangements. Article. Innsbruck: University of Innsbruck, Institute of Educational Sciences; 2005.
8. Marchionini G, Maurer H. The roles of digital libraries in teaching and learning. Communications of the ACM. 1995: p. 67-75.
9. Gass SM, Schachter J. Linguistic Perspectives on Second Language Acquisition Cambridge: Cambridge University Press; 1989.
10. Nakata Y. Motivation and Experience in Foreign Language Learning. 1st ed.: Peter Lang International Academic Publishers; 2006.
11. Sherr I. CNET. [Online].; 2014 [cited 2016 03 27. Available from: http://www.cnet.com/news/unity-one-gaming-development-platform-to-unite-them-

all-up-for-sale/.

12. Technologies U. Unity3D. [Online].; 2016 [cited 2016 03 27. Available from: https://unity3d.com/unity/engine-features.

13. Chase J. http://justinmchase.com/. [Online].; 2014 [cited 2016 03 27. Available from: http://justinmchase.com/2014/12/09/the-composition-design-pattern/.

14. Amazon Web Services. aws.amazon.com. [Online].; 2016 [cited 2016 March 27. Available from: https://d0.awsstatic.com/whitepapers/aws_pricing_overview.pdf.

15. T-Systems Enterprise Services GmbH. White Paper Cloud Computing. [Online].; 2016 [cited 2016 03 27. Available from: http://www.t-systemsus.com/umn/uti/508260_1/blobBinary/White+Paper+Cloud+Computing+%257B%257BPDF%252C+351+KB%257D%257D.pdf.

16. SunGard Financial Systems. SunGard Availability Services. [Online].; 2012 [cited 2016 03 27. Available from: http://www.sungardas.co.uk/Documents/Cloud%20Computing%20-%20Resilience%20is%20the%20key%20to%20success%20White%20Paper.pdf.

17. Harvard Business Review. Harvard Business Review. [Online].; 2011 [cited 2016 03 27. Available from: https://hbr.org/resources/pdfs/tools/16700_HBR_Microsoft%20Report_LONG_webview.pdf.

18. Amazon WebServices. Amazon WebServices. [Online].; 2016 [cited 2016 03 27. Available from: http://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html.

19. DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, et al. All Things Distributed. [Online].; 2007 [cited 2016 03 27. Available from: http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf.

20. Barr J. AWS Blog. [Online].; 2014 [cited 2016 03 26. Available from: https://aws.amazon.com/blogs/aws/run-code-cloud/.

21. Namiot D, Sneps-Sneppe M. On Micro-services Architecture. International Journal of Open Information Technologies. 2014 Sep; 2(9): p. 24-26.

22. Amazon WebServices. Amazon WebServices. [Online].; 2016 [cited 2016 03 27. Available from: http://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html.

23. Serverless Framework. Serverless. [Online].; 2016 [cited 2016 03 27. Available

. from: https://github.com/serverless/serverless.

24 Amazon WebServices. Amazon WebServices. [Online].; 2016 [cited 2016 03 27.
. Available from: http://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html.

25 Amazon WebServices. Amazon WebServices. [Online].; 2016 [cited 2016 03 27.
. Available from: http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html.

26 Magalhaes R, Magalhaes M. The Three Most Widely Used Amazon Web Services.
. [Online].; 2014 [cited 2016 03 27. Available from: http://www.insideaws.com/articles-tutorials/general/three-most-widely-used-amazon-web-services.html.

27 Mikowski M, Powel J. Manning Publications. [Online].; 2012 [cited 2016 03 28.
. Available from: http://deals.manningpublications.com/spa.pdf.

28 Mesbah A, Deursen Av. Migrating Multi-page Web Applications to Single-page Ajax
. Interfaces. In Software Maintenance and Reengineering, 2007. CSMR '07. 11th European Conference on; 2007; Amsterdam: IEEE. p. 181-190.

29 Seshadri S, Green B. AngularJS: Up and Running: Enhanced Productivity with
. Structured Web Apps. 1st ed. O'Reilly , editor.: O'Reilly Media; 2014.

30 Unity Technologies. Unity Documentation. [Online].; 2016 [cited 2016 03 27.
. Available from: http://docs.unity3d.com/Manual/nav-InnerWorkings.html.