

Outdoor Localization and Machine Learning-based Educational Treasure Hunts

Edoardo Cogotti, Marsha Gomez Gomez, Francesco Pesciatini, Gianmarco Petrelli

e.cogotti@studenti.unipi.it, m.gomezgomez@studenti.unipi.it, f.pesciatini@studenti.unipi.it, g.petrelli@studenti.unipi.it

ABSTRACT

Nowadays the mobile technology offers new opportunities to extend the learning activity. The students could use these devices to learn and play at the same time, especially for little children. We present an effort to provide teachers with a dedicated application to design location-based educational games via smartphones. The application allows teachers to define customized treasure hunts on which insert educational questions for their little students. The system is designed to support outdoor learning activities such as object recognition (using Machine Learning kit) and orienting themselves and learning information about the environment (using Google Maps API and Google Geocoding API). To complete the game the children have to explore their environment and solve all the ordered questions and they can learn notions and discover the environment such as an urban area, a park or an archeological site.

1 Introduction

The application could be used also for entertainment purposes, but the scenario on which we focused is educational. Teacher often have the problem to teach in an effective way and this is especially important when the student is a child. Gamification is a powerful concept and could be used also to teach in a new way a certain type of notions to children. The chosen approach is to allow to make treasure hunts, old pedagogical approach already used without smartphones or other technologic devices. In each game students are challenged to solve all the steps. Each step has a question that the child has to solve and it can be a place step or a recognition step. In the first case the child must orient himself/herself and found the correct place. Participants are geo-localized in real time, saving the current position. In the second one the child has to understand which object is the correct solution and take a picture of it. A trained machine learning model will classify the object in the picture. Obviously, the sensors that are used for each game are the GPS (so it is an outdoor location-based application) and the camera of the smartphone. This is the major strength of the technological approach: it can exploit geo-localization and machine learning tools and present information in a multimedia format. In particular our application has a lightweight approach: each student can download a game and then it is tracked and he/she can completely play locally, reducing the backend load and in general keeping the online communication at the minimum. Teachers are involved in the design process and they can build own treasure hunts and let play them to their students. Teachers and especially children are

normal users, so the application must be strongly user-friendly. In the game creation the teacher can simply click a button and his/her coordinates will be evaluated and he/she just has to insert the question for the Place questions and clicking another button for the Recognition questions a predefined list of possible solutions (that the machine can classify) is provided. In the recap the teacher can reorder the steps by drag and drop and delete some step. Each not allowed action is alerted to the teacher to inform him/her about the issue. Instead, when a child plays, he/she can see his/her position to the Google Map. He/she can check if he/she is in the correct position clicking on a button, list the current question and the previous ones and can have a limited number of hints per game. In a place question the hint is a circle displayed in the Google Maps within the correct solution is contained and, in a recognition question, the hint is shown in a hangman game.

2 Architecture

The architecture is formed by a database that relates to the Spring backend. The mobile frontend sends REST requests to the backend and the Google APIs.

2.1 Database

The database is created on MySql and it's simply formed by an unique table TreasureHunt. Each record of this table represents a new step of a given treasure hunt game. The table is composed by the following fields:

- gameId
- Step
- GameName
- Question
- Answer
- StepType (boolean to distinguish between recognition and place questions)

The primary key is formed by gameId and step.

2.2 Backend

The backend is realized with a Maven project with Spring Boot parent for the web server creation. The REST API calls are sent to the backend by the mobile application part. The other main dependencies are with mysql-connect, json-simple and hibernate. Spring Boot allows to realize the backend application and define the REST calls using annotations.

Mysql-connect (and the persistence.xml file in META-INF folder) allows to connect backend and database.

Json-Simple allows to create Json objects to use in the response of some REST calls and to implement the toJson() method in the model classes.

Hibernate allows the Object-Relational Mapping service handling the data persistence in the MySQL database using annotations and to maintain a relational database formed by Java classes. In addition to this it allows the definition of an EntityFactory that can be used to get EntityManagers that make persistent the entities and execute CRUD operations on them. The model classes are TreasureHuntStep (a table record), TreasureHunt (wrapper class of TreasureHuntStep to allows the serialization of it in the POST /game request) and CustomObject for the response of POST /game.

- POST /game: the backend gets (if not yet got) the highest gameId in the table and saves in the database the steps sent by the application with that id plus one. In the response is contained the gameId used in the database for the game sent by the application.
- DELETE /game/gameId=x: delete all the record of the treasure hunt with gameId x. Used in debugging.
- GET /games?initName=x: return gameId, gameName and number of steps for every treasure hunt that starts with x
- DELETE /step?gameId=x&step=y: delete a record of a given treasure hunt and deal with the inconsistency decrementing the step field of the next record in that treasure hunt, if there are. Used in debugging.

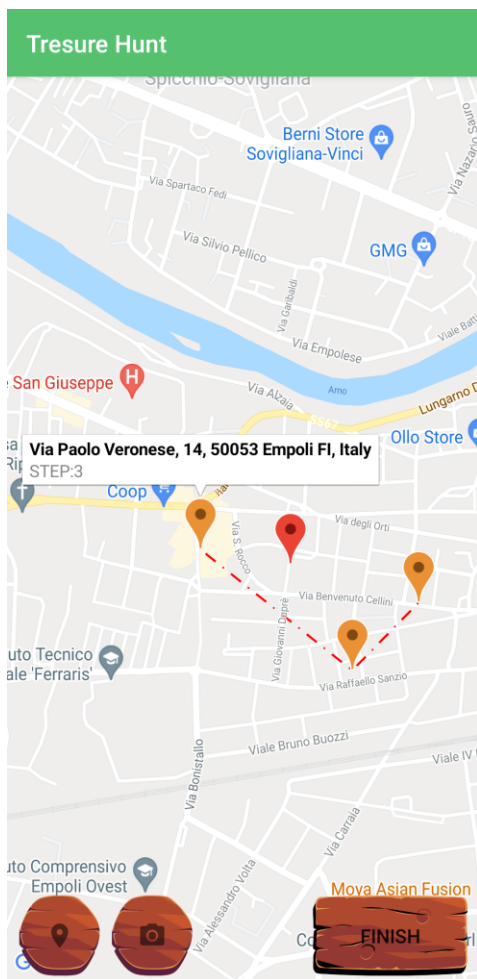


Figure 1: Screenshot of the application (Make part)

2.3 REST API calls

- GET /game?gameId=x: returns every step of the treasure hunt with gameId x

2.4 Frontend

Android Studio Oreo (8.1) is used for the frontend. For instance, the frontend as shown in the Figure 1 is very user friendly. The red marker is the current position. The figure is related to the maker side: the teacher can insert Place questions with the first button and Recognition questions with the second button. Each new Place step is inserted in the map with an orange marker and a path between consecutives step is formed. If the treasure hunt is composed by at least 2 steps he/she can click on the finish button to be redirected to the recap activity.

3 Experimental results

The application has some crucial point in terms of performance. The first one is the communication with the backend to interact with the database. The others are the Google Maps and the Machine Learning Kit APIs and in particular we want to understand their accuracy and their computational cost.

3.1 Backend stress test

We stress the backend by sending many requests at the same time using Jmeter. Running the backend on a simple laptop the Backend can serve without problems 500 GET requests per second and can serve 750 GET requests per second with an average delay of 2 seconds. Instead, 1000 GET requests are an issue and the backend crashes in that setup. It is important to highlight the application context. As we mentioned in the introduction the number of requests is limited: the main computation is on the mobile application. In fact, the user sends just a POST in the entire process of the creation of a game and chosen a game it is necessary just a GET request. The heaviest communication is the search bar: the user can insert a text and clicked a button he/she use a GET request to get the list of treasure hunt in the database that start with that text. In this context the total number of requests of a single user is very low. Considering this context and the limited computer capabilities during the Jmeter test we can consider the backend stable enough for its goal.

3.2 Machine Learning and Google Maps

ML kit and Google Maps API are provided by API and they are the state of art in their field. We can trust about them in term of reliability, accuracy and computation complexity. In particular, the Google Maps API don't have real alternatives. However, tests in real environments for both APIs reacted in the correct way about the different behaviors that a normal user could have.

4 Conclusion (e improvement)

In conclusion the application is a good tool for teachers to integrate with the traditional children teaching. It can be used to create and play pedagogical treasure hunts in many possible ways. For instance: urban areas, parks, cultural areas (such as archeological sites or zoo) or woods for boy scouts. Many improvements are still possible: a drag and drop in the recap activity could be more intuitive; we could insert other multimedia elements to enrich the child experience with other content about the notions; the machine learning model could be customized in order to be more specific for educational purposes; we could improve Google Maps style (more user-friendly zoom and path between markers that respect the real path) and finally we could allow to search a particular position in a search bar in the maker part instead of constraint to physically move the user towards the real position to create the treasure hunt.. However, this version is a good application, stable and with an user-friendly UI designed to simplify the interaction for the children.

REFERENCES

- [1] Marc Schickler, Manfred Reichert, Philip Geiger, Jens Winkler, Thomas Funk, MMicha Weibach, 2020. Flexible development of location-based mobile augmented reality applications with AREA *Journal of Ambient Intelligence and Humanized Computing* (2020). DOI: <https://doi.org/10.1007/s12652-020-02094-9>
- [2] Tarun Kulshrestha, Divya Saxena, Vaskar Raychoudhury. 2017 SmartITS: Smartphone-based Identification and Tracking using Seamless Indoor-Outdoor Localization.. DOI: 10.1016/j.jnca.2017.09.003