A Gamification Platform for Inspiring Young Students to Take an Interest in Coding

Nearchos Paspallis

University of Central Lancashire, Cyprus
Pyla, Cyprus

Abstract

Coding skills are widely considered an invaluable asset in today’s and tomorrow’s business landscape. However, universities in North America and Europe are expected to produce computer scientists at a rate that is well below the projected demand. This has called for initiatives like the Computer Science Education Week in the United States and the Europe Code Week in the European Union. This paper proposes a novel, engaging activity that aims at inspiring students to take an interest in Coding via a mobile phone-based treasure hunt. The main contributions of the paper are the description of the platform used to realize the treasure hunt, and the experience report of the event that evaluated its effectiveness.

Keywords: Education, Coding

1. Introduction

Coding skills are widely considered an invaluable asset in today’s and tomorrow’s business landscape [1]. By learning how to code, one not only improves their employability but also their potential of becoming a successful entrepreneur – even when their activity is not directly related to the software sector. At the same time, universities in North America and Europe are expected to produce computer scientists at a rate that is well below the projected demand [1]. This has called for initiatives like the Computer Science Education Week in the United States [2] and the Europe Code Week in the European Union [3].

This paper proposes a novel engaging activity that aims at inspiring students to take an interest in Computer Science in general, and Coding in particular. This is based on a mobile phone-based treasure hunt, implemented as an Android app communicating with an AppEngine server. This approach, combined with a one-hour motivation speech and training, was tested and evaluated in an event involving 65 high school students, aged 12 to 18. The two main contributions of this paper are the description of the open-source platform used to realize the treasure hunt activity, and the experience report of the event which evaluated its effectiveness.

The rest of this paper is organized as follows: Section 2 discusses gamification and its use in the education sector. Then, Section 3 introduces the Gamification Platform, and Section 4 presents the experience report. The paper closes with conclusions in Section 5.

2. Gamification and Education

While gamification has historically been used even before computer use became mainstream, its potential has seen wide acceptance after the success of location-based service Foursquare [12], which was followed by more, general-purpose gamification apps like Funifi [4]. A popular
definition of gamification is “the use of game design elements in non-game contexts” [8]. In its more recent incarnation, gamification is also viewed as a “software service layer of reward and reputation systems with points, badges, levels and leader boards” [8].

In education, technology has been blamed for burdening learning by distracting students [16]. At the same time, technology is used to engage students and enhance learning, largely based on game-like approaches. Particularly in teaching programming, gamification has been used with great success: While technically not a game, Scratch [15] has succeeded in engaging programming novices with its jigsaw puzzle-like interface to creating fun programming projects. Similar to Scratch, Tynker provides a jigsaw puzzle-like programming language, which is used to solve small puzzles and progress to the next one [5]. Other, more traditional games, such as the Lightbot educational game [9], aim at inspiring computational thinking. Beyond teaching programming, gamification has also become popular in general instructional lectures, in the form of digital Game Based Learning [17]. Extensive work has also been conducted in “determining how to match instructional content to specific game design features” [11].

In terms of inspiring students to take an interest in computer science, various attempts have been documented. For example, the most recent Computer Science Education Week (code named Hour of Code) utilized self-paced puzzle-like games. These were used to introduce students to new concepts first (like loops and conditionals) via recorded videos featuring popular figures, and then to guide them into applying the newly acquired knowledge in solving new stages of a puzzle game. To further entice students, popular themes such as Angry Birds [20] – and more recently Flappy bird [18] – were used as the background story. The Hour of Code event was able to attract millions of students, and was arguably successful in giving students “a taste of programming and motivate some of them to continue onward” [10].

Finally, besides inspiring young students to take an interest in coding, gamification has also been used to engage existing computing students. For example, Mitchell et al. [14, 13] describe a particular case, where Freshman students go through an intensive, four week challenge during which they build a mobile application in teams. The main goal of it is “to lead students through a challenging (yet highly scaffolded) project to show them where their course could take them.”. In this case, the target mobile application is a game, but the course itself is also organized and run as a game itself.

3. Gamification Platform

This paper describes an approach for inspiring high school students to take an interest in computer science in general, and coding in particular. The proposed approach is centered around events which aim at engaging high school students. The events include the standard motivational talks and programming tutorials, but also – importantly – a fun, game-like treasure hunt activity, designed to interest participants.

3.1. Computerized Treasure Hunt

The success of a motivational event is dependent on its inclusion of a fun, memorable activity. In this section, we describe the Treasure Hunt Challenge, an activity aiming to inspire and motivate students to take an interest in computing in general, and coding in particular.

The Treasure Hunt Challenge (THC) is a traditional treasure hunt in the sense that it provides a sequence of clues, which can lead the participants to the treasure. Finding the answer to the current clue opens up the next one. The process repeats until the final clue is uncovered, which leads to the treasure. In our implementation, users can also skip a step if they choose, which however results to a penalty.

In the THC, the clues are questions that are either directly related to computing-related facts or questions that require puzzle-solving and coding skills (see Figure 1). To further engage
participants, the challenge can include location-aware activities (such as requiring that you go to a specific place before answering a question) and task-oriented challenges (such as scanning a QR-code with the camera to reveal the answer).

Figure 1. Screenshots from the Treasure Hunt app.

Unlike traditional treasure hunts, the THC is fully computerized and requires the use of a network-connected smartphone, which must be equipped with a location service (typically based on a GPS sensor) and a camera. Currently, only Android devices are supported, even though there are no technical limitations for extending support to other mobile platforms.

From a technical standpoint, the THC requires the continuous availability of a server-based system which provides the clues, and verifies the submitted answers. The same system is also responsible for keeping track of the score, which is a combination of how many questions were answered correctly and how many not, as well as of when was the treasure hunt completed (the time is used to break ties, in case two teams collect the exact same number of points).

Figure 2. The live scoreboard.

Adding to the participant’s experience, and to the overall engagement achieved, is a live scoreboard. The scoreboard displays both the top teams with their corresponding score, as well as location of the teams on a map, both in real-time. As the live scoreboard is web-based, it can be easily projected to supported, network connected monitors. This is particularly useful when the facilities allow for a large display in the area where the treasure hunt takes place, as it engages the teams to compete with each other (see Figures 2 and 3).
3.2. System Architecture

The THC platform follows the client-server architecture. The core logic of the treasure hunt is implemented in the server side, and is made available to clients via a REST-ful [7] Application Programming Interface (API). The developed API is largely based on the one used in the Four Week Challenge [14] for realizing custom mobile apps using AppInventor [19]. It was designed to be simple, while enabling interested students to develop their own version of the client.

Figure 3. Students participating to the Treasure Hunt.

The API allows the client to query the available categories (i.e. competitions), and start a new session for a selected category. While in an active session, the client can query the server for the current question and the score. It can also use a similar query to submit an answer or skip the current question. Finally, while in a session, the client can also update its current location – which is required for location-aware questions – and also request the full scoreboard.

Passing arguments to the service is always done via HTTP GET requests with parameters. For instance, assuming that a session has been established (providing the client with a valid Session UUID), the current question can be requested as follows:

http://uclan-thc.appspot.com/api/csv/currentQuestion?session=agtzf0VjbfLMXRoY31UxI2Vzc2vbhiAglCAgOe8CQw

Two variants of the API are supported, allowing for encoding the data in standard Comma Separated Values (CSV) or in JavaScript Object Notation (JSON) format. For instance, the above request could produce the following output (guiding the player to the clue from Figure 4):

OK,
This question is exactly at this location: (35.007584, 33.696870). Use a map to find it!
false

At the same time, using the JSON variant (i.e. by replacing CSV with JSON in the request URL), the output would be as follows:

{ "status": "OK",
  "question": "This question is exactly at this location: (35.007584, 33.696870). Use a map to find it!",
  "isLocationRelevant": false
}

A detailed description of the API is available at: https://uclan-thc.appspot.com/guide.
3.3. Implementation

The implementation of the Computerized Treasure Hunt system includes two main apps, the mobile app, and the server.

A standard mobile app was developed and made available in Google’s Play store. While the API allows for custom implementation, even using the App Inventor framework, our choice was to develop a native app that allows finer customization of the User Interface and the functionality (screenshots of the mobile app can be seen in Figure 1).

For the server side, we chose Google App Engine, which is a popular Platform as a Service (PaaS) framework. This provides a convenient platform as it is highly scalable: as minimal traffic is required in normal times, with high peaks during the competitions, serving a medium size treasure hunt can generally stay within the free quota limits, while providing extremely high reliability and performance when needed. Additionally, this platform provides many ready-made services, like user authentication, which minimize the required development effort.

The implementation of the server required the development of both a front-end – i.e. the services realizing the API calls – and a back-end – i.e. the system that can be used to create and edit a treasure hunt. The front-end was realized as Java Servlets, implementing all the API calls, such as those described in Subsection 3.2. The back-end was realized also via Java Servlets, but with the employment of Java Servlet Pages (JSP) to implement the graphical UI of the system.

The source code of both the mobile client and the server is available under the Lesser General Public License (LGPL) at: https://github.com/nearchos/uclan-tch.

4. Evaluation

Our approach was evaluated in a Treasure Hunt event where high school students were invited to attend the event and compete in the treasure hunt. There were no prior coding skills required, or any other requirements, besides the limit age of: students should be in high school and be age 12-18. This event received funding from the Cyprus Fulbright Commission, which allowed us to advertise it online, and also use some popular prizes to make it more appealing.

The event was organized in sessions, which included traditional elements like an inspiration talk, and a guided programming tutorial. It also included less traditional sessions such as the smartphone-based treasure hunt and a movie. However, this kind of events have been used widely already, and we have learned from them [10]. In our case, we wanted to go a step further, and see how a game could add to effectiveness of the event, and in particular in its engagement and memorability. In this paper we focus on the treasure hunt, and the feedback it received from the participants.

The participants were not aware of the nature of the treasure hunt, or the Android app itself, before the event. They were only asked to bring along a compatible Android device (phone or tablet). The venue provided WiFi access to all the participants, at which point they were informed of the nature of the competition and were asked to install the app. During the launch of the app, the students were asked to enter their names and email, and were allowed to form single-person or two-person teams.

The treasure hunt questions were prepared in advance, and the competition was automatically set to start at a predefined time – with a countdown clock displayed on a large screen at the starting area (see Figure 3). At that point, the students were able to get the first question and start finding clues and answering the questions. The competition was programmed to last for one hour, after which the system automatically terminated the treasure hunt.

This particular competition was designed with 12 questions that included puzzles, observation tasks, and some programming tasks. For instance, we asked questions related to Scratch [15] programming language’s loop and conditional mechanisms, which were presented in the talks before the Treasure Hunt (see Figure 4).
Figure 4. Sample clue which includes a Scratch-based puzzle.

4.1. Design Elements

The uttermost aim of the event was to engage high school students to take an interest in coding. In doing so, we focused on enjoyment and playfulness, as these have been identified as promoting factors [6]. In the following, we discuss how various Game Elements [8] have been dealt with in the design of the mobile app:

**Immersive environment:** We aimed at blending the real world with the digital. As such, the clues are partially in the app, but partially also in the real world. For example, once you follow the clue provided by the app, you might be asked to deal with a real-world challenge (see Figure 4). This proved to have various advantages: the players seemed to have more fun, while they would see how competing teams were faring.

**Feedback:** Answering a question immediately provides feedback, notifying the player if they were successful or not. If the question is location-sensitive, then the feedback will include a hint if the player is not at the expected location.

**Time pressure:** Time is commonly a main aspect of many games. In the treasure hunt, time pressure was used to enhance the gamefulness of the competition, but also to make sure that everything stayed on schedule.

**Scoreboards:** Besides the typical, in-app scoreboards that are available in many games (including this), a public, big-screen scoreboard was also used (as shown in Figure 2). Besides score and ranking, the location of the teams was also displayed in realtime, visualizing the dynamics of the game, and further enhancing the game experience.

4.2. Experience Report

The evaluation event attracted 65 high school students, with the treasure hunt being one of the most successful features. After the completion of the event, we emailed the students and asked them to complete an online survey for feedback. Out of them, 17 students (i.e. 26%) submitted a feedback report, rating the event positively (65% rating it as good or excellent) and saying that they would recommend it to peers (54% answering that this is likely or very likely).

In addition to this, the students were asked to identify the feature that they liked the best in the event, and the majority of them selected the Treasure Hunt – 11 out of 17 (i.e. 65%).

Based on comments we received, but also on what we observed, we have identified some areas of improvement:

**Practice stage:** As students were not familiar with the use of the treasure hunt mobile app, they asked for a practice stage where they could familiarize themselves with the app. Having said that, our experience was that the vast majority of them was very quick at picking up how
the competition – and the mobile app – worked.

More mobile platforms: Because of time and resource limitations, we only developed an Android mobile app. However, many students did not own an Android phone or tablet. Even though most of them were able to get one from friends and family, it would be useful to develop at least a web app client so more platforms could be used. Even students with Android devices faced problems, particularly when the device was running an older version of Android (the mobile app requires Android version 4.0 and up).

Location accuracy: As some questions required that the users were at specific locations to submit an answer, having an accurate location service was important. We depend on the built-in location service of Android, but our experience showed that different devices responded very differently to location sensing, both in terms of accuracy and speed. Also, as most of the treasure hunt took place indoors, getting a precise location reading was even more challenging.

Traceability of answers: The only information available to participants was the scoreboard (see Figure 2). Some students suggested that more information should be made available. For instance, after the termination of the competition, an automatic email could be sent to all the participants with a report on how they did (i.e. when each question was attempted, if the answer was correct, the gained marks – or penalty, etc.)

Treasure hunt length: As mentioned, we used 12 questions in the treasure hunt and allowed the teams one hour. It turns out that today's teenagers are indeed technology natives, and most of them were able to finish the treasure hunt in less than half hour, implying that more and harder questions should be used.

5. Conclusions

This paper describes an approach used for engaging and motivating high school students to take an interest in coding. At the core of this approach is a gamified Treasure Hunt, which is supported by a mobile client and a server platform. Various game elements were used in the design of the mobile app, making the treasure hunt more fun and game-like. The approach was evaluated both qualitatively and quantitatively. For the latter, a Treasure Hunt event was organized and students who participated to that rated the event positively, and said that they would recommend it to others as well.

For future work, we would like to further enhance this platform, possibly allowing teams to interact with each other via the application. Also, we would like to enable more mobile platforms via an HTML5 version of the app, as well as native apps for other popular platforms like iOS and Windows Phone devices. Lastly, we will aim to provide better location support via a custom-made indoors location system.

Acknowledgements

The author would like to acknowledge the financial support provided by UCLan Cyprus and the Cyprus Fulbright Commission. The input from associates from both UCLan Cyprus and UCLan UK campuses is also duly acknowledged.

References