Towards a Catalog of Gestures for M-learning Applications for the Teaching of Programming

Anderson S. Marcolino, Alef Santos, Marco Schaefer, Ellen F. Barbosa
Institute of Mathematics and Computer Science (ICMC-USP)
Avenida do Trabalhador São–Carlense, 400, São Paulo, Brazil.
Email: andersonmarcolino@gmail.com, alef.segura.santos@usp.br, marco.schaefer@usp.br, francine@icmc.usp.br

***Abstract*—Teaching of programming disciplines is mandatory in several CS courses worldwide. However, this domain still faces limitations and challenges. While these limitations continue to be a reality in classrooms, mobile devices have earned more and more users and, as a consequence, the development of mobile learning applications is rapidly growing as well. Based on the limitations in such domain, on the increasing popularity of mobile devices and on their capability to support the teaching and learning processes through their hardware specificities, in this paper we propose a catalog of gestures for mobile learning applications to be adopted in the teaching of programming. The proposed catalog also provides a way to investigate if mobile devices may motivate students to learning to programming. Four steps were conducted in this work. Firstly, primary studies that propose surface gestures interfaces were identified. Secondly, a list of main commands and programming codes were analyzed and selected to be represented by graphical gestures. Thirdly, the returned primary studies were used to propose a catalog of gestures for those code structures. With two or three possible gestures for each code structure, an evaluation to identify which of those gestures were preferred among undergraduate students was conducted. Based on the results of 69 participants, the final catalog was defined. Finally, in a fourth moment, a gesture recognition application tool was used to perform a usability evaluation. Preliminary evidences suggested the catalog can adequately support m-learning applications for the teaching of programming. Furthermore, the catalog pointed out a positive feedback that can bring a more attractive way for coding, when considered the adoption of mobile devices' small keyboards.**

I. Introduction

The teaching of programming disciplines is mandatory in several undergraduate CS courses curricula worldwide [1], [2]. Despite the widespread of such disciplines, the teaching of programming foundations still faces limitations and challenges [3].

Souza et al. [4] performed a systematic mapping for identifying and classifying programming problems; as a result, six categories of problems were defined: (1) Learning of Programming; (2) Application of Programming Concepts; (3) Understanding of Programs; (4) Refactoring and Factoring Programs; (5) Learners Motivation; and (6) Teacher Difficulties. Such categories were also investigated in a survey performed by Marcolino and Barbosa [3]. The results showed a consensus between the Brazilian problems and the categories of programming problems established based on international primary studies.

Based on the collected evidences, ways to reduce and solve the reported programming problems (mainly those related with learners' motivation) become to be analyzed. Among the possibilities, it is highlight the adoption of mobile learning applications [5], [3].

Mobile learning or m-learning, is one of the learning modalities resulting from the adoption of information and communication technologies (ICTs). In short, m-learning allows to learn anytime and anywhere through the mobility of devices and wireless network [6].

In this perspective, several applications (apps) which supports the teaching of programming can be found over the Internet. However, the majority of such apps supports informal learning, providing limited feedback. Additionally, ways to facilitate the interaction with mobile devices in the context of coding and programming is still an issue to be addressed, since their small native keyboards may limited coding tasks [5].

Motivated by this scenario, this research aims at the creation and evaluation of a catalog of gestures for: (i) identifying preliminary evidences if gestures may be a more attractive interaction way in mobile devices for the teaching of programming and coding tasks; (ii) identifying if the catalog of gestures may motivate learners in learning to programming through mobile platforms; and (iii) investigating how satisfactory is the users' experience to substitute the small native keyboards of mobile devices by gestures as text input medium (coding).

Additionally, secondary aims were defined: (a) identification of studies which propose techniques and methods of gestures conceptions; (b) identification of nuances about users' preference in the selection of gestures; and (c) availability of the catalog of gestures, developed as a service, to allow the conduction of new researches to identify if touchable gestures interfaces adopted in m-learning applications can result in a more engaging learner experience and, consequently, improve the processes of teaching and learning in the programming setting.

The gestures and the evaluations performed so far showed preliminary evidences that the proposed catalog can be successfully adopted in the teaching of programming foundations through mobile platform.

This paper is organized as follows. Section II summarizes the related works, discussing the background concepts addressed throughout our research. Section III discusses the process of conception of the catalog and the evaluation regarding the selection of the gestures. Section IV presents a practical usability evaluation of the proposed gestures in mobile devices. Section V summarizes our contributions, lessons learned and perspectives for future work.

II. Background and Related Work

The increasing popularity of mobile devices and their capability to support the teaching of programming, mainly in informal learning, creates a great opportunity to investigate if the teaching and learning programming problems, in formal learning, can be mitigate somehow by the adoption of such devices [7], [8], [5].

The extensive adoption of such devices is a direct consequence of their hardware specificities (e.g., accelerometer, gyroscope, and touchscreen), reduced costs and support to several daily activities. This adoption worldwide also facilitates their use for formal learning. However, depending on the subject of study, such devices may hindered their adoption and may lead to learners' demotivation or an incomplete or fail learning process. [7], [5].

Informal learning corresponds to the process of learning where users do not have the responsibility of making their learning activities[9]. Students are free to decide when they want to study and, in some apps, what they want to learn; they do not have an enrollment in an educational institution/course. On the other hand, formal learning has a teacher, tutor or a mediator who is responsible to check and ask students, regularly enrolled, about the learning progress, since the activities and the course in which the students are enrolled have a well-defined schedule and conclusion date [6]. Based on the formal learning, students usually seen mobile games and other apps as attractive ways to interact. Such attraction should also be part for the learning applications in introductory programming disciplines, being a requirement to guarantee that both experience and motivation be maintained. Furthermore, when considering the coding activities, mobile platforms presented some specific limitations, mainly when the main way to input code is through the native small keyboards. But, how to make this task more comfortable and, at the same time, an attractive factor to learn through apps?

An interesting approach to start answering this question can be the adoption of surface gestures interfaces (or just gestures interfaces) as a medium to motivate and solve some of the limitations of mobile devices, besides to investigate a different and potentially innovative way to teach programming concepts. There is also an opportunity to address teaching and learning programming issues related to motivation, since mobile devices is a common and nowadays, a medium to low cost technology which many younger have access [3]. Some of the main contributions found in the literature on gestures interface area are summarized next. They supported us in the creation of our catalog of gestures.

Zhai et al. [10] presented a compilation of studies about gestures conceptions. Based on such studies, the authors defined a set of design principles for creating gesture interfaces. Among them, we highlight the conception of gestures analogous to physical human movements or cultural conventions, keeping gestures simple and distinct. Their set of design principles were considered as a checklist to guarantee that our gestures covers the same principles.

Anthony et al. [11] selected gestures interfaces regarding usability elements and performed an empirical study. The aim was to verify if the feedback after the insertion of gestures would improve the process of interaction with them and make users, of several ages, more comfortable and confident to use such gestures. The authors proposal, results of visual feedback and the design recommendations for new surface gestures interfaces were also considered in the process of creation of our gestures.

Lü and Li [12] developed a tool entitled Gesture Coder, for programming multi-touch gestures by demonstration. Basically, their contribution focuses in the creation of gestures to be used in software and apps. In this same perspective, Dean [13] and Wobbrock [14] proposed algorithms for supporting gestures recognition in software and applications. Those researches are the nearest of the programming area and bring good contributions for the development process; however, they are not related with teaching and learning of programming.

In the context of teaching gestures for users, an important contribution was proposed by Westerman [15]. The author defined a set of steps to teach how to use gestures and, also as in Anthony et al. [11], adopted feedback to improve such process in multi-touch sensitive devices.

Another studies also discussed techniques for the creation of gestures [16], [17], [18], [19]. The aim is the creation of gestures not in an arbitrary way, but considering human experience and natural interaction process. Such type of gestures are known as naturals interfaces' gesture. Its use a software layer developed based on human behavior to propose interactions with the computer in a more natural way [20]. Despite the importance in the exploration of this technology with mobile devices, the number of studies which propose natural interfaces for learning apps is still incipient.

Studies as conducted by Shamir et al. [21], Vatavu et al. [22] and others have the concern on investigate how natural interfaces and gestures recognized by sensor (e.g., accelerometer, gyroscope), web cams, touchscreen or touchable surfaces and on virtual reality systems, may improve the teaching and learning process. However, few studies focus in mobile devices, and none of them in the teaching of programming [5].

In general, the works presented herein address important issues on the adoption of gestures to improve the users' interactions: (i) the gestures definition; (ii) the gestures development process; and (iii) how teaching users to interact with applications by these gestures. One of the main concerns involved refers to the design, presentation and implementation of interaction interfaces for mobile platforms. Historically, such area has trouble in keeping standards and good practices [23]. In the context of creation of gestures for learning apps, few studies were found, the main contribution was developed by Reis [24].

Reis [24] proposed a set of gestures to allow the draw and study of geometry in Android mobile devices. They were motivated by the limitations of traditional methods in math area [25]. The study obtained positive results in the adoption of gestures as a way to improve interaction method in the target domain.

Motivated by the lack of primary studies for proposing natural gestures interfaces for the teaching of programming in mobile learning applications, in this paper we propose and discuss a gesture catalog for m-learning applications for the teaching of programming. The catalog is presented next.

III. The Gestures Interface Catalog

Our gesture catalog comprises 12 gestures; each gesture corresponds of an excerpt of code, representing main programming structures.

A. Gestures Catalog Conception

The planning and elaboration of the gesture catalog followed three main tasks: (i) mapping the programming concepts and selection of languages to be represented through gestures; (ii) mapping the code structures from the chosen languages and selecting a main set of such structures to be taught in the apps; and (iii) proposition of interaction gestures for the respective code structures based on the primary studies described on Section II. In the first task, several curricula of programming disciplines taught in different CS courses of University of São Paulo were analyzed; as a result, a structure composed of the main programming concepts and topics was modeled. One model covers programming procedural aspects[1] and other model covers object oriented aspects[2]. Such models were adopted to allow the selection of two programming languages and, in consequence, the definition of code excerpts or structures to be represented by the gestures in the next task. Regarding the programming languages, C and Python were selected; the first representing procedural aspects and the other addressing object oriented issues. Both were selected taking into account the CS curricula previously analyzed.

In the second task, the code structures (functions and/or methods) were mapped. Such code structures were chosen through analysis of educational materials and courseware of C and Python languages. Table I shows the 12 code structures covered in our catalog. In this first moment only basic structures in these two languages and their both programming aspects, procedural and objected oriented, were considered, leading the omission of structures from C++ and other languages.

TABLE I
Code Structures in C and Python.

#Id	Structure	C Code	Python Code
1	Conditional If	if(){ }	if :
2	Conditional If then Else	if(){ } else{ }	if : else:
3	Loop While	while(){ }	while :
4	Do While	do(condition){} while();	-

5	Condicional Switch Case	switch(){
		default: break;
		}
			-

6	Ternary Operator	? : ;	-
7	Loop For	for(;;){ }	for in :
8	Block Commentary (I)	/* * */	''' '''
9	Line Commentary (II)	//	#
10	Function	return type name(){}	def () :
11	Exception	-	try : except () []:
12	Class	-	class :

In the third task, each code structure was analyzed and a set of gestures for each of them were planned and created. The related works (Section II) and their methods were adopted to support such gestures creation. To reduce the number of proposals for each code structure, the gestures' usability and the establishment of an intuitive relation between them and the code (e.g., be a letter, a word, or even a reference to the way that the software behaves during the execution of the code), were considered as well. To illustrates this task, imagine the cycle of execution of a loop, such as `for()` or a `while()` in C language or the bifurcation of process in conditional structures as in `if()`, for the same language. For the loop case, circular figures gives a good intuitive understanding of the structure and, for the if case, a flow represented by two lines can give the idea of a conditional structure for the normal and alternative path, as depicted in Figure 1. The complete questionnaire and graphical representation of each proposed gestures is available at https://goo.gl/Kusyvr.

Based on the issues pointed out in the related works (Section II), the hypothesis that specialists can elaborate a set of adequate gestures for a given application, which will be fully accepted by the end users can be mistaken.

[1]Procedural paradigm model: https://goo.gl/HcERoa
[2]Object Oriented paradigm model: https://goo.gl/ywLdmU

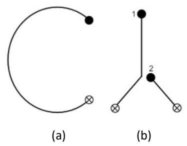

Fig. 1. Gestures Examples: (a) for-loop and (b) if-else.

At a practical context, the user does not evaluate gestures as faster or costly, but they prefer simple and intuitive gestures. There is still the user previous exposure level to different interactions mechanism, which can vary from person to person [17]. In this perspective, a first evaluation of the gesture catalog proposed is discussed next.

B. Gestures Selection Evaluation

As our gestures catalog received two or three variations of possible gestures, a selection of the best was needed. To do so, a first evaluation was conducted. The aim of such evaluation was the selection of a gesture that better fit with the code structure or to find out a new gesture, since the participants could draw a new gesture. Additionally, it allows the identification of first impressions and possible improvements from the participants, which were a total of 69 undergraduate students.

The process of the first evaluation was composed by six phases, divided in planning, execution and analysis: (i) planning and creation of a questionnaire for the validation of the gestures vocabulary; (ii) pilot application of the questionnaire with researchers at Technical University of Darmstadt; (iii) enriching the questionnaire based on the pilot application results; (iv) reviewing the proposed questionnaire with an expert; (v) applying the questionnaire; and (vi) summarizing the results and data analysis.

Planning

In the planning, a questionnaire with 12 questions was prepared. Each question corresponds to a code structure and two to three gestures to be selected by the participants. A blank space was included allowing the participants made their our gesture suggestion, when they totally disagreed with the proposed ones. Additionally, they needed to justify their proposition.

The questionnaire was validated through a pilot application with two data mining and learning environment researchers from the Technical University of Darmstadt (Germany). Some of the gestures in the questionnaire received suggestions, which were considered for preparing a new version of the questionnaire.

Finally, the questionnaire was reviewed by a teacher from CS, who suggested a few changes. At the end, a final version of the questionnaire was ready to be applied with undergraduate students.

Execution

In the execution, the questionnaire was applied face to face after a class with students of a CS course at University of São Paulo, in a single application section. All of them from the same year of course. The only requirement to the student participate was to have some contact (basic or advanced) with programming languages. They were volunteers, being free to participate or not.

In the execution process, each participant received a copy printed of the questionnaire. The applier read the instructions, which basically instructed the participants to select one of the prefered gesture for each of the 12 source-code structures. If the participant did not agree with the available gestures to be selected, he or she could drew a new gesture, justifying such proposed. After they selected or drew a new gestures for each of the twelve structures in the questionnaire, the participants could leave the classroom where the execution occurred.

C. Result Analysis

In the last phase, the evaluation data was summarized and the results were analyzed. The time to fill the questionnaire was, in average, of 7 minutes; the minimum time was 4 minutes and the maximum, 21 minutes.

Table II summarize the results in percentage. The "Dominance of gesture most selected" varies from 45% to 78% of acceptance, as shown in Table II. Neither the percentage of gestures proposed by the participants (column "Rejection") or the sum of the percentage of selection for the other gestures also available to be selected (column "Other Gestures") exceeded the percent of selection of a gestures systematically developed in the conception tasks of the catalog. To illustrate such affirmation, the gesture #1, for example, had three possible options of gestures to be selected (gestures a, b, c - d was the blank space to receive a participants' propose). While the most selected corresponds to 45% (gesture c) from the total of selections from the participants, the sum of the selections for the other gestures (gestures a, b), corresponding only to a percentage of 39%. Thus, such sum of percentage for the other gestures did not exceed the 45% obtained by gesture c. Finally, the percentage of rejection (gesture corresponding to the letter d), based on the number of proposes made by participants, corresponding only of 16%, which also did not exceed the percentage of the gesture most selected.

The rejection rate of the proposed gestures, counting by number of drew gestures, vary from 4% to 16% (10% in average). The total of suggestions in the 12 questions was 78. Finally, an average of 3% of blank answers for each question was also registered.

The selection of one gesture for each of the 12 code structures in the questionnaire was made taken into account when the rejection rate and the number of participant's gesture proposal did not exceed its dominance rate. So, the number of selections for the dominance gesture must be greater than the number of rejections and the sum of

TABLE II
First Evaluation Gestures Selected Results.

#Id	Code Sstructure Name	Gesture a	b	c	d	no answer	Dominance of gesture most selected	Other Gestures	Rejection (drew their own getures)	No answer
1	Conditional If	18	9	31	11	0	45%	39%	16%	0%
2	Conditional If Then Else	17	9	33	10	0	48%	38%	14%	0%
3	Loop While	40	20	9		0	58%	29%	13%	0%
4	Do While	39	21	9		0	57%	30%	13%	0%
5	Conditional Switch Case	34	29	5		1	49%	42%	7%	1%
6	Ternary Operator	44	18	3		4	64%	26%	4%	6%
7	Loop For	46	16	7		0	67%	23%	10%	0%
8	Block Commentary (I)	44	18	7		0	64%	26%	10%	0%
9	Line Commentary (II)	54	11	4		0	78%	16%	6%	0%
10	Function	32	28	7		2	46%	41%	10%	3%
11	Exception	23	34	3		9	49%	33%	4%	13%
12	Class	39	19	3		8	57%	28%	4%	12%

selections for the other gestures. The gesture dominance rate had 57% average, with an amplitude from 45% to 78%. That is, at least one of the suggested gestures showed itself intuitive and learnable, being satisfactory in relation to the computing function it represents and, therefore being concise and accepted to be included in the final catalog and in a app.

The gesture with the highest rejection rate was the conditional structure `if()`. It not only had the highest rejection rate but also the lowest dominance rate (45%). Even so, it satisfied the conditions to be selected. Nevertheless, in case of need of adaptation or changing some gesture to allow a better usability and user acceptance, `if()` is the main candidate. Furthermore, we got 78 suggestions, one for each rejected gesture, giving us the possibility of investigating new standards more acquainted to undergraduates students.

The gestures of `switch-case`, `ternary-operator`, `functions`, `exception` and `class` had null answers; the last three, selected from Python language, had less answers since the participants were less experienced in the Object Oriented paradigm. This same justification can be applied for the first and second null answers, corresponding to gestures in C language.

Finally, another relevant point to be discussed regards the graphical representations selected. Based on participants' selected options, all of them corresponding to the most simple gesture available. The workload process regarding the interaction in a touchscreen or any similar interface must be intuitive; such statement was reaffirmed, based on the selected gestures identified [10], [11].

D. Validity Evaluation

Threats to Conclusion Validity: the major concern is the sample size, which must be increased in prospective studies.

Threats to Construct Validity: the qualitative analysis considered the students' answers. As some of them can still not understand the importance of their contributions for research, make their participation obligatory could led to collect imprecise results. In this context, students were volunteers, being free to perform or not the evaluation.

Threats to Internal Validity:

- **Differences among participants:** as we considered undergraduate students that could be enrolled in different disciplines, they knowledge may lead to null answers in the collected questionnaires. This could be mitigated with a possible training in the code structures included in the questionnaire. Thus, training sessions will be considered in future evaluations.
- **Fatigue effects:** as the average to answer the questionnaire was 7 minutes, fatigue effects was not considered relevant; and
- **Influence among subjects:** participants took the evaluation session under supervision of a human observer. We believe that this issue did not affect the internal validity.

Threats to External Validity:

- **Instrumentation:** to guarantee that the questionnaire may not present some limitation that would difficult to get a concise result in the selection of the gestures, a pilot session was conducted with expert's researchers in the Technical University of Darmstadt.
- **Participants:** undergraduate students were selected. However, students from secondary schools should also be considered in future evaluations, allowing to generalizing the study results.

IV. Gestures Catalog Usability Evaluation

Based on the set of gestures defined, a usability evaluation was also conducted. Such evaluation aimed at the investigation of how satisfactory was learners' experience during the use of the proposed gestures in mobile devices. Three main aspects were verified: (1) the catalog acceptance; (2) the users' experience using the proposed gestures; and (3) if the catalog can motivate learners in the adoption of mobile devices in programming courses and coding tasks.

The process of the second evaluation was composed by five phases divided in planning, execution and analysis: (i) investigation and selection of an app to allow a practical evaluation of the gestures; (ii) creation of a questionnaire to evaluate the usability of the gestures; (iii) pilot application of instrumentation; (iv) execution of the evaluation; and (v) summarization of results and data analysis.

Planning

The initial idea in the planning phase was the development of a prototype to allow the usability evaluation. However, due to time constraints and the complexity in the development and adoption of algoritms to support the gestures recognition, an app that allows the gestures registration, their insertion by users and recognition through feedback was adopted. Three mobile apps from Google Play Store[3] were selected as candidates to be used. As we already had available devices with Android platform, only Google Play Store was considered. After performing tests with the gestures in each app, the Gesture Tool[4] was selected. It is important to notice that some gestures suffered small adaptations to be used in such tool.

After choosing the app, a questionnaire covering the 12 graphical representation of the gestures was prepared. For each gesture, three fields were available, each corresponding to an attempt in the insertion of the respective gesture. If the user had make some mistake or the gesture had not been recognized by the app, a new attempt was made. If all of the attempts had been used and the participant did not have the gesture correctly recognized, the supervisor indicated to the participant to move on to the next gesture. At the end of the questionnaire, three questions were answered.

1) Would you adopt an app with the gestures that you've just used in this evaluation as a substitute in programming tasks?

2) How much satisfactory was your experience with the gestures used?

3) In your opinion, could the presented gestures motivate students to learn Programming using mobile devices?

To pre-evaluate the instrumentation, a pilot session was performed with two graduate students. At the end of the session, they reported to be totally satisfied with the questionnaire, with the gesture app and with the interaction. So, no improvement was necessary in the instrumentation and we proceed to the execution phase.

[3]https://play.google.com/store

[4]https://play.google.com/store/apps/details?id=com.davemac327.gesture.tool

Execution

In the execution phase, the usability evaluation was applied. For this evaluation, we adopted a different approach. In total, two tablets and two smartphones with Android Platform were available and, as each participant should use a mobile device and answer the questions related to such use, the execution was performed individually, in face to face sections. The participants were chosen aleatory in CS Department at University of São Paulo by an applier. All of them were volunteers and from CS area. Furthermore, the applier was responsible for each collect of answer. He/she also supported each participant in using the Gesture Tool app, for checking and controlling the number of attempts, ensuring the same number of insertions for each gesture. The applier was warned to do not make any intervention while the participant was inserting the gestures and to take notes of any unexpected behavior during the execution (e.g., a participant's mistake during the insertion of the gesture). Figure 2 depicted a screen of the gestures inserted on the Gesture Tool and a participant inserting a gesture on it.

All 12 gestures were previously inserted in the tool. To test the gesture, the "test" option was selected in the app and a black screen was shown. In this screen, user inserted the draw of the gesture and, if the gesture was correctly recognized, the name assigned to the gesture during its registration on the tool was displayed as feedback. The need of registering the gestures in the Gesture Tool before the collection of the answers led us to the individual application, since the registration could take too much time to be given as a task for the participants. Besides that, as the application was available only for the Android platform, the participants who did not have an Android smartphone could not participate.

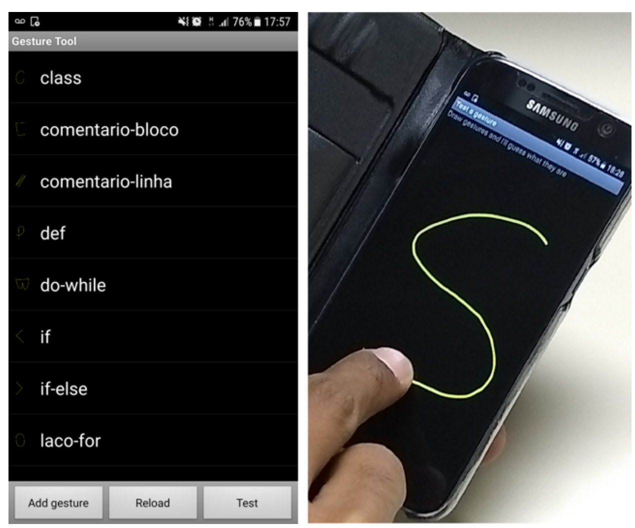

Fig. 2. Gesture Tool and the Gestures for Usability Evaluation.

The individual application also brought a different limitation: a reduced number of participants, since each execution required one of the four available devices we had. In this perspective, a total of 40 participants evaluated the catalog: 16 graduate students from master (5) and doctoral (11) CS programs; and 24 undergraduate students from CS courses. As the time was not a relevant factor in this evaluation, it was not considered.

A. Results Analysis

In the last stage, the results were summarized and analyzed.

One of the aspects registered during the execution was the number of attempts for each gesture (Figure 3). This corresponds to the insertion and a possible mistake performed by the participant or by the app in recognizing the gesture inserted. Despite the highest numbers for `line comment` and for `ternary operator` structures/gestures, we noticed that none of the mistakes were consequence of an incorrect recognition of gestures by the tool. The number of attempts was due to the lack of attention from participants in the insertion of more complexes gestures.

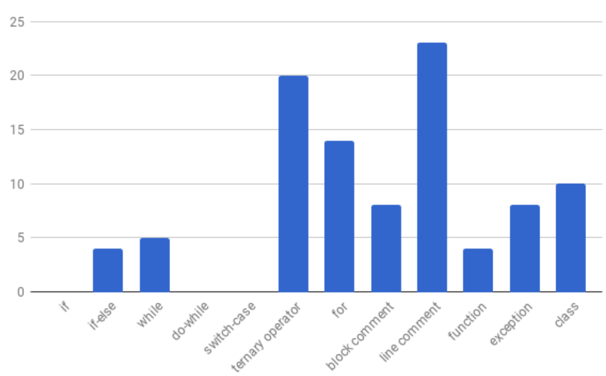

Fig. 3. Number of Attempts per Code Structure

Even so, we have also performed a deep analysis of `line comment` and `ternary operator` gestures and their structures. For such analysis, the studies used in the conception of the gestures were considered again. They supported us in the verification of some possible mistakes in the definition of such gestures. However, no evidences of mistakes were found. Thus, we intend to include the second most selected gesture from the first evaluation in future interventions to allow a further investigation of user's preference and how they feel using two options of gestures for the same structure.

For the first question related to this usability evaluation (*Would you adopt an app with the gestures that you've just used in this evaluation as a substitute in programming tasks?*), 34 participants answered "yes" (they would adopt apps that use such gestures) and 6 answered "no"(they would not use an app with the proposed gestures). This result provides preliminary evidences that the catalog propose a gesture feature capable of being adopted to replace tasks of text insertion, according to the majority of the participants.

For the second question (*How much satisfactory was your experience with the gestures used?*), the participants were asked to sign one option in a five-range scale: "Totally Satisfied", "Very Satisfied", "Satisfied", "Few Satisfied", "Unsatisfied". Figure 4 summarizes the results.

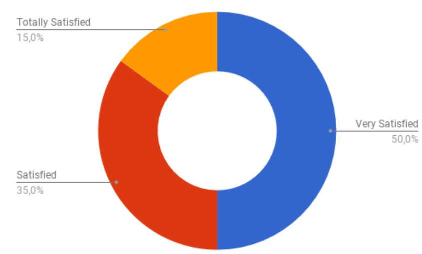

Fig. 4. Participants' Satisfaction in the Use of the Gestures

Only three of the five ranges were signed. Half of the participants (50%) answered they were "very satisfied"; 35% said they were "satisfied"; and 15% said they were "Totally satisfied". These results give us a positive feedback that the gestures satisfied the participants and the users, mainly due the greatest part of the participants were undergraduate students, being considered our main target users of apps that will be supported by our gestures catalog. In this first moment, only undergraduate students, from 18 and up are considered our target users, mainly by the availability of such users. However, we aim to develop apps which will adopt the gestures to support students from primary and secondary schools in near future.

In the third question (*In your opinion, could the presented gestures motivate students to learn Programming using mobile devices?*), 34 participants answered "yes", that is, they believe the set of gestures can motivate students; and only four participants answered "no". Once again, these results suggest that the proposed catalog can be an interesting mechanism in helping students to learn how to program through mobile devices. Of course, we still need more concrete evidences to state that our gestures can really motivate students; however, such feedback encourages us to implement our gesture catalog for further evaluation in real learning environments.

B. Validity Evaluation

Threats to Conclusion Validity: similarly to the first evaluation, the major concern here is the sample size (40 participants), which should be increased in future evaluations.

Threats to Construct Validity: here we also faced the same threat of the first evaluation. However, as the usability evaluation was performed by a "supervisor" and individually applied, we believe this issue was mitigated.

Threats to Internal Validity: we dealt with the following issue:

- **Fatigue effects:** although the time had not been relevant in the second evaluation, it was considered in our pilot in order to identify a possible threat. As the pilot was executed in 10 minutes, at most, it was not considered as relevant.

Threats to External Validity:

- **Instrumentation:** to guarantee that the questionnaire may not present limitations on the collection of results in the selection of the gestures, a pilot session was conducted with two graduate students. However, one important issue was the need to adapt some gestures from the first evaluation to be used in the Gesture Tool. Thus, in short term, we intend to conduct a new evaluation with our own implemented app.
- **Participants:** undergraduate and graduate students were selected to take part of the usability evaluation. So as in the first evaluation, a more heterogeneous students profiles must be considered in future studies in order to better generalize the results.

V. Conclusions and Future Work

Despite the increasing importance of programming disciplines and their adoption in CS curricula worldwide, the teaching of programming still presents several limitations and challenges [4]. While such problems have being faced, more and more mobile devices have being adopted and used in different learning contexts [6], [7]. Based on this scenario, in this paper we propose a gesture catalog to deal with the problems in teaching programming and, at the same time, to motivate learners in the adoption of mobile learning apps, mainly in the context of codes insertion. The catalog was developed following methods presented in the literature. Two evaluations were performed. The first evaluation allowed the creation of a primary version of the gesture catalog. The second one evaluated the users' usability and satisfaction perspective. In both evaluations, preliminary but positive evidences regarding the applicability of the gestures catalog were collected.

The collected evidences also led us to elicitate several lessons and benefits related to our secondary aims:

To define gestures, user experience must be considered. As this process requires a deep understanding of human behaviors and experiences, well-founded studies in this area are needed to give support to the proposition of the gestures. In this context, we have adopted two main studies to ensure such understanding (Section II).

The selection of the gestures also considered the users' level of difficult and facility in the draw/insertion of them. As discussed in the first evaluation (Section III), users tend to select the easiest gesture, mainly by their workload and tendency to make tasks in a more comfortable way. This is also influenced by their experience, mentioned in the previous item. Thus, considering the results from the first evaluation, simple tasks must be considered to drives the creation of interactions, either them of simple gestures for mobile devices, or based on the use of sensors, as pointed out in some of the related works (Section II).

Lack of attention may lead to mistakes. This issue is definitely true but, considering learning tasks, it must be specially addressed; the attempts in second evaluation (Section IV) showed that. Based on the supervisors' notes, it was clear the higher number of attempts was a result of the lack of attention. This is also a known factor when we look at several students' answers in tests or learning activities. In this perspective, this fact bring us reflections that ways to address this issue are necessary. Anthony et al. [11] showed how important is the feedback in the context of gestures insertion, and this should be one of the possible strategies to reduce the attention problems. Other possibility is taking into account sensors, speakers and other hardware specificities of mobile devices to work together with screen messages, warning learners and conduct them to pay more attention in whatever task they are doing in a learning app. Penalties through gamification concepts [26] can also be an interesting alternative.

The catalog developed may motivate such strategy to integrate the teaching and learning processes through mobile devices for other disciplines. As the research of Reis [24], our proposal may allow the development of other gestures catalogs. Based on the widespread of mobile devices and their cost reduction, they clearly becomes a medium to make the teaching and learning processes occurred at anyplace and anywhere. But, how it was discussed, mobile devices limitations (e.g., small keyboards), may hinder or create some constraints in their adoption as a educational tool. The catalog proposed is one of several possible ways to reduce these barriers resulting in better apps solutions to support teachers and learners.

As future works we highlight the need of:

- Developing our own m-learning app to be evaluated with more students, based on related works [13], [14];
- Studying and integrating in the app more feedback features besides textual messages, as an attempt to improving students' attention;
- Planning and conducting new evaluations with our app, allowing improvements in the set of gestures; and
- Integrating the developed app as a service in a software product line for the conception of m-learning applications for the teaching of programming[27].

Finally, making our gesture catalog available through mobile applications and also as a service, researchers may adopt it and, consequently, new studies can be conducted to propose better or new features for allowing a more intuitive and natural experience in m-learning applications.

Acknowledgment

The authors acknowledge Brazilian funding agencies – FAPESP (Process 2014/03389-9), CAPES (Procad 071/2013) and CNPq – for the financial support provided to this research.

References

[1] SBC, "Sociedade Brasileira de Computação. Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática (in portuguese)," Availble in http://www.sbc.org.br/index Acessed on January, 04th, 2017, 2003.

[2] C. S. P. IEEE and P. ACM, "Joint Task Force for Computing Curricula." Available in http://www1.acm.org/education/curricula.html Acessed on May, 4th, 2016., 2005.

[3] A. da Silva Marcolino and E. F. Barbosa, "A Survey on Problems related to the Teaching of Programming in Brazilian Educational Institutions." in *47th Annual Frontiers in Education Conference (FIE 2017)*, Indianapolis, Indiana, EUA, oct 2017.

[4] D. M. Souza, M. H. S. Batista, and E. F. Barbosa, "Problemas e Dificuldades no Ensino de Programação: Um Mapeamento Sistemático (in portuguese)," in *Revista Brasileira de Informática na Educação.*, vol. 24, 2016, pp. 1–14.

[5] A. S. Marcolino and E. F. Barbosa, "Softwares Educacionais para o Ensino de Programação: Um Mapeamento Sistemático (in portuguese)," *Simpósio Brasileiro de Informática na Educação*, vol. 1, pp. 190–199, 2015.

[6] M. Sarrab, "M-learning in Education: Omani Undergraduate Students Perspective," *Procedia - Social and Behavioral Sciences*, vol. 176, pp. 834 – 839, 2015, int. Educational Technology Conference IETC Chicago, IL, USA.

[7] A. Jalil, M. Beer, and P. Crowther, "Pedagogical requirements for mobile learning: A review on mobilearn task model," *Journal of Interactive Media in Education*, vol. 2015, no. 1, 2015.

[8] N. F. Duarte Filho and E. F. Barbosa, "A Requirements Catalog for Mobile Learning Environments," in *Proceedings of the Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2013, pp. 1266–1271.

[9] E. T. Straub, "Understanding technology adoption: Theory and future directions for informal learning," *Review of educational research*, vol. 79, no. 2, pp. 625–649, 2009.

[10] S. Zhai, P. O. Kristensson, C. Appert, T. H. Anderson, X. Cao *et al.*, "Foundational issues in touch-surface stroke gesture design: an integrative review," *Foundations and Trends® in Human–Computer Interaction*, vol. 5, no. 2, pp. 97–205, 2012.

[11] L. Anthony, Q. Brown, J. Nias, and B. Tate, "Examining the need for visual feedback during gesture interaction on mobile touchscreen devices for kids," in *Proceedings of the 12th International Conference on Interaction Design and Children*, ser. IDC '13. New York, NY, USA: ACM, 2013, pp. 157–164. [Online]. Available: http://doi.acm.org/10.1145/2485760.2485775

[12] H. Lü and Y. Li, "Gesture coder: a tool for programming multi-touch gestures by demonstration," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2012, pp. 2875–2884.

[13] D. Rubine, "Specifying gestures by example," *SIGGRAPH Comput. Graph.*, vol. 25, no. 4, pp. 329–337, Jul. 1991.

[14] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '07, New York, NY, USA, 2007, pp. 159–168.

[15] W. C. Westerman, "Gesture learning," Apr. 12 2016, uS Patent 9,311,528.

[16] P. H. C. Braga, A. Cardoso, and E. A. L. Jr., "Desenvolvimento de gestos personalizados para criação e navegação em ambientes de realidade virtual e aumentada associada a dispositivos móveis (in portuguese)," Availble in https://www.researchgate.net/publication/303920129_Desenvolvimento_de_Gestos_Personalizados_para_Criacao_e_Navegacao_em_Ambientes_de_Realidade_Virtual_e_Aumentada_associada_a_Dispositivos_Moveis Acessed on April, 19th, 2018, 01 2012.

[17] M. Gutierrez, "Análise de métodos para criação de vocabulário de gestos para aplicações em superfície multi-toque (in portuguese)," Master's thesis, Escola Politécnica da Universidade de São Paulo, 2015.

[18] A. C. S. Medeiros, "Processo de desenvolvimento de gestos para interface de usuário (in portuguese)," mathesis, May 2015. [Online]. Available: http://tede.biblioteca.ufpb.br:8080/handle/tede/7863

[19] A. Kamal, "Enabling techniques to support reliable smartphone-based motion gesture interaction," Master's thesis, 2015. [Online]. Available: http://hdl.handle.net/10012/9030

[20] C. P. Santosa, R. Segatto, Y. P. Teixeira, and T. F. Willers, "Interface natural na educação: Um estudo de caso no ensino de geografia (in portuguese)," Availble in http://br-ie.org/pub/index.php/wie/article/view/3106 Acessed on April, 19th, 2018, 2014.

[21] S. Alavi, D. Arsenault, and A. Whitehead, "Quaternion-based gesture recognition using wireless wearable motion capture sensors," *Sensors*, vol. 16, no. 5, p. 605, 2016.

[22] R.-D. Vatavu, G. Cramariuc, and D. M. Schipor, "Touch interaction for children aged 3 to 6 years: Experimental findings and relationship to motor skills," *International Journal of Human-Computer Studies*, vol. 74, pp. 54–76, 2015.

[23] G. Stockwell, "Investigating learner preparedness for and usage patterns of mobile learning," *ReCALL*, vol. 20, no. 3, pp. 253–270, 2008.

[24] H. M. Reis, "Concepção de um software de geometria interativa utilizando interfaces gestuais para dispositivos móveis (in portuguese)," mathesis, Universidadede São Paulo, Oct. 2014.

[25] *Interação em Interfaces de Softwares de Geometria Interativa: Um Mapeamento Sistemático (In Portuguese)*, 2013.

[26] J. Simões, R. D. Redondo, and A. F. Vilas, "A social gamification framework for a k-6 learning platform," *Computers in Human Behavior*, vol. 29, no. 2, pp. 345–353, 2013.

[27] A. S. Marcolino and E. F. Barbosa, "Towards a software product line architecture to build m-learning applications for the teaching of programming." *In: 50th Hawaii International Conference on System Sciences (HICSS-50)*, pp. 6264–6273, 2017.