Interactive Geometry Goes Mobile with GeoTouch

University of Sao Paulo
Brazil
e-mail: sisotani@icmc.usp.br

Abstract—Interactive Geometry (IG) is a computational approach to teach geometry using interactive geometric objects. It is based on the learning-by-doing method where the representation and manipulation of geometric spaces can support students to construct their knowledge through experience. Several research findings indicate various benefits of using IG in classroom. Nevertheless, IG software developed to date has been based on the desktop model where the interaction between a user and the IG software occurs through the conventional inputs and outputs methods: keyboard, mouse, and a monitor. Currently, this model is being challenged by the widespread adoption and use of mobile devices. In this scenario before building an IG software that can be effectively used in mobile devices such as smartphones and tablets, where the input is based on touches and movements of fingers and the output is a small screen, there is a need to rethink the interface and what interaction methods should be employed. In response to this need, our paper examines the design and development of a GI application for mobile devices. In this study, we have proposed and implemented an interface and interaction model that is suitable for developing a GI software for mobile devices.

Keywords—Interactive geometry, m-learning; Math education

I. INTRODUCTION

Education has undergone considerable changes in recent years. Innovative computational resources, new and efficient communications technologies and an increasing amount of research in the area of education, have led to a restructuring of the classical and traditional way of teaching and learning. Another factor that has brought about this change is the aptitude the modern generation has for these new technologies. All this is supported by the efforts made to devise and develop new and more productive teaching tools in schools [10]. These changes have acted as a driving-force for research studies in this area, in a way that has made the teaching-learning process increasingly productive both inside and outside of the classroom.

The technology involved in mobile devices, such as cell phones, smartphones and tablets has evolved at a much faster rate than, for example, personal computers. Every year, new ways of using these appliances are being developed which go beyond the original communications purposes for which they were created. The widespread dissemination of these devices, which is made possible by the low costs and advances features involved, justifies conducting investigative research into their potential use in educational environments, particularly when they concern nations in development [12][14]. The research field that investigates how these devices can collaborate to the educational environments is known as Mobile Learning, or m-learning, [9].

Although the benefits of m-learning has been highlighted in several research findings [3][6][7], in geometry learning where interaction with geometric objects and concepts are fundamental to support the learning process, few advances have been reported to date. One of the reasons, is that currently that are few interactive geometry software (IGS) available to use in mobile devices [4]. IGS has been used successfully to help students learn geometry with support of desktop computers. It provide several functionalities that help students to interact and visualize geometric concepts which increase participation, motivation and learning gains [5][13][11].

Thus, the main objective of this work is to propose and develop an IGS that can be used in any kind of mobile device. To achieve our goals, first we analyzed how to reduce standard IGS interfaces to fit smaller screens available in current smartphones and tablets. Then, we propose an interaction pattern that maps the mouse-keyboard interaction to multi-touch interaction. Finally, we implemented the tool and carried out a usability test with five User Interface (UI) specialists.

This article is structured in the following way: in Section 2 there is an outline of the theoretical principles underlying the interactive teaching of Geometry. In Section 3, there is an examination of the issues concerning the development of the interface and the data framework. In Section 4 a usability test is presented. And Finally, Section 5 provides a summary of the final considerations.

II. INTERACTIVE GEOMETRY

Interactive Geometry (IG), also known as Dynamic Geometry, was first presented by Nick Jackiw and Steve Rasmussen. The term “interactive” can be understood as being opposed to the “static” way in which the traditional geometry constructions were taught to students [5]. This type of software allows a dynamic and interactive environment to be created, where the students construct their learning by manipulating and altering geometric objects on the computer screen. A simple example for a better understanding about Interactive Geometry: say that want to build a bisector...
between two points A and B. To construct the bisector between these points, it is necessary to find two distinct points which equidistant A and B, and draw a straight line named ‘r’. This line is the perpendicular bisector between A and B (Figure 1).

An application that implements an Interactive Geometry would allow the user to move two points A and B and its bisector will automatically update. This way the user would know which are and how the properties of the bisector works through the manipulation of geometric space, because no matter how the points are moved, the bisector will continue between two points.

According to Piaget, the mind forms its structures of knowledge by taking data from the external world and then interpreting, transforming and reorganizing it. It can be concluded that to “learn” it is necessary to “do”. Unfortunately, the area of Geometry is overlooked and this makes it necessary to employ new forms and methods for teaching it. According to Brandão et al. [2] and Usikin [13], little attention has been paid to the teaching of Geometry either in primary and secondary or in higher education. In addition, currently Geometry is taught in a traditional and static way without any concern to teach an understanding of basic principles. For this reason, it should be stressed that Interactive Geometry is a suitable method for assisting the teaching of geometry with the aid of technologies to explore all the available computational resources. The use of Interactive Geometry software allows the student to have training in problem-solving and by doing this, discover a solution.

The main difference between the application shown here and other types of software that implement Interactive Geometry, is the kind of interaction that exists between the student and the software. In software based on the desktop computer, such as iGeom [2][5], the interaction occurs by means of the mouse and keyboard. However, in touch sensitive panel applications, the interactivity that is provided by using the fingers to manipulate the geometric objects, means that the activities and use of the tools can become much more natural for the students. In the next section, there will be an outline of the application development process.

III. DEVELOPMENT OF AN INTERACTIVE AND COLLABORATIVE APPLICATION FOR MOBILE DEVICES

The application, called GeoTouch, is being developed in the Android platform and was planned to perform the basic functionalities of an Interactive Geometry software. Among the functionalities already developed are the following: the creation, manipulation and removal of points, straight lines and circumferences; the drawing of points belonging to a line or a circumference; the drawing of intersections between lines and circumferences. Among the functionalities being developed, are those that give support to the collaborative learning environment. It is intended that this environment will allow two or more users to manipulate geometric objects in the same working space although using different devices.

4. Interface Development

The initial idea for the development of interactive applications was that all the activities could be carried out by means of intuitive gestures. A straight line would appear if two fingers pressed the screen at each point for three seconds; and a circumference would be designed if one of the fingers could be held on the screen, while the other made the outline in the same way as with a compass. Unluckily the original idea proved to be unsuccessful in the technology and usability testing, because a number of problems arose such as the fact that the devices did not support multi-touch sensing and it was impossible to carry out some of the activities easily. At this point, it was decided to establish the interaction by means of menus and buttons to ensure that the software would be compatible with most of the available appliances. Thus at this stage of the project, there were no multi-touch operations.

The interfaces of various design applications for touch-sensitive devices were observed and analyzed. On the basis of this knowledge and the technical use of Human-Computer Interaction (HCI), a solution was reached that only required a minimum area of the device screen (Figure 2). The option to create a menu with buttons could affect what is of paramount importance to a design application in a mobile device – the size of the screen. The way of solving this problem was to create a button in the lower left-hand corner, that only activates a tools menu. This button keeps the icon of its chosen tool in its main icon to allow the user to identify which tool should be selected at any given moment.

A button to remove objects from the screen was also created. Since the removal of objects can only occur when an object is selected, this button only appears when a point (or other object) is selected. In this way, the removal option will only appear when there is a need for it and the space on the screen...
will be optimized. The removal button is illustrated in the right-hand corner of Figure 1 on the right. The selected object is the green point in the diagram with the longest radius. When this button is clicked, the selected object will be removed, as all the others depend on it.

B. Development of a data interaction framework

The first stage in establishing the internal framework in an application for Interactive Geometry was to create the object database. Within the object database used in this study, there is one that is the base for everything – this includes the point, which was meticulously studied to obtain the best possible manipulation. In view of the fact that all the mobile devices that would be used would have to be Touch Screen, or rather, touch-sensitive, the finger would be the manipulator of the objects. The design of the point on the screen was carried out with precision in a suitable size and in a way that prevented the user from having difficulties in handling the objects. The design to represent the point on the screen is a circular form which the user is able to manipulate by touch and by dragging the finger across the screen.

The application base comprises two classes: 

*GeoTouchActivity* – which handles all the touch events and contains all the application objects and *Point Class* – this extends the Android View class to enable its objects to be added as the offspring of the *GeoTouchActivity* class, and are thus displayed on the screen. After the application base was ready, the next stage was to add the functionalities for the creation of straight lines and circumferences. At least two points are needed to design a straight line or a circumference.

Two other classes were created by using the *create points* functionality: the *Straight line Class* and the *Circumference Class*. Both have the same structure and receive two points in their constructor so that they can be designed. In the case of the line there is a problem: no standard Android class has a function that allows an “infinite” line to be drawn. Two calculations had to be made to solve this problem.

For a line R, formed from points P1(x1,y1) and P2(x2,y2), it is possible to obtain the equation of R: y=ax+b, Where:

\[ a = \frac{(y_2 - y_1)}{(x_2 - x_1)} \]  

\[ b = y_1 - \frac{(y_2 - y_1)}{(x_2 - x_1)} x_1 \]

By means of these equations, it was possible to employ two methods for the *Straight line class* called *FX(float y)* and *FY(float x)*. The *FX* yields the x value of a point, given its y value; and the *FY* method yields the value y of a point, given its x value. These methods use the equation of the straight line to obtain these values. It is necessary to find the coordinates of the points of Line R, which are not visible to the user, so that it is possible to design a section of the line that is longer than the screen. To do this, the coordinates of P1 and P2 are compared to discover the slope of the line and hence determine the coordinates of the points by employing the *FX* and *FY* methods. Figure 2 illustrates the reasoning that is used.

With the values of points P3 and P4, it is possible to design a line that will be long and enough to cross the screen and give the sensation that the created line is infinite (Figure 4 – to the left). For designing the circumference, the only calculation needed is to draw the radius which is the distance between P1 and P2 (Figure 4 – to the right).

![Figure 3. Calculation of the points on the line](image)

![Figure 4. An example of lines and circumferences in the application](image)

The creation of lines and circumferences depends on the formation of points. Thus the removal of points must be dealt with in a special way. If a line is created and afterwards, one of its base points is removed, there is no way that this line can continue to exist since a line cannot be drawn on the basis of only one point. Hence, it was necessary to create a structural relationship between the objects of the application to ensure that the objects that are dependent on other objects will be removed in a consistent way. This relationship can be analyzed as a graph.

In Figure 4 on the left, an R line can be observed that is dependent on Points A and B. If A or B are removed, line R will also be removed. The complexity increases when new lines or circumferences are created by using new points or the points that already exist. This line of reasoning applies to more than one functionality of the application: the points belonging to the lines or circumferences. In their turn, these points will be dependent on the objects which they were created for. In the example on the right of Figure 5, a C point is formed above the R line (or rather, C belongs to R). In this case, if Point A or B are removed, line R will be removed and hence Point C which is dependent on line R will also be
removed. If there are more dependencies in the hierarchy below C, these will also be removed.

![Figure 5. Structural Relationship](image)

In practice, the drawing of a point on the touch-sensitive screen has proved to be a more complicated procedure than was originally imagined. Owing to the reduced size of the screen, it was more difficult to draw a new point on the screen that already existed with precision. The solution found was to draw a new point in an area that was not occupied by the screen and its later handling above the line where the new point had to be included. In this way, instead of drawing the point directly on the line, the user manipulates the point above the required line and then makes it “belong” to this line (or rather, it remains above the line). Before this solution could be implemented, it was necessary to calculate the collision between the point being moved and the other geometric objects (lines and circumferences for example) of the drawing space. This was carried out by employing the `isTouched` method in the Line and Circumference classes to determine whether or not, at a given point \( P(x,y) \), there is a collision with the line or circumference. Figure 5 shows how the collision between the point and line is detected.

![Figure 6. Collision between the point and line](image)

There is a margin of error \( E \) to ensure that there is a close proximity with the line since, with the use of the touch system, the interaction is not very precise and thus it can be difficult to position a point exactly above a line. The same principle was used for the circumference (Figure 7) although the mathematical formula is different.

![Figure 7. Collision between the point and circumference](image)

When the solution is implemented, it is possible to draw points and change them into points that are dependent on a line or circumference. A different color was chosen for the point that is dependent on a line or circumference to make it easier to use this functionality. In Figure 8, the point positioned in the center of the line and the point in the upper part of the circumference are dependent points.

![Figure 8. Point dependent on the line and circumference](image)

### IV. PROOF OF CONCEPTS

We carried out a usability test with five specialists based on the ten usability heuristics for user interface proposed by Nielsen [8]. According to Ardito et al. [1] a heuristic evaluation can be performed by a small number of usability experts and end-users to find the main problems with interfaces of educational software. Artido et al. also consider it the cheapest and most efficient way to find errors in interfaces. Some of the heuristic criteria are:

1. **Visibility of system status**: Ensures that the user knows what is happening in the application.
2. **Match between system and the real world**: Ensures that the user can easily understand the application.
3. **User control and freedom**: Ensures that the user has control over the application.
4. **Consistency and standards**: Ensures that the application is consistent and easy to use.
5. **Error prevention**: Ensures that the application prevents errors.
6. **Recognition rather than recall**: Ensures that the user can easily recognize information.
7. **Flexibility and efficiency of use**: Ensures that the application is flexible and efficient.
8. **Aesthetic and minimalist design**: Ensures that the application is visually appealing.
9. **Help users recognize, diagnose, and recover from errors**: Ensures that the application provides helpful feedback.
10. **Help and documentation**: Ensures that the application provides helpful information.

To check the severity of problems in each heuristic we propose a series of tasks (e.g. create a point, move a circle, etc) and questionnaires. Then, we asked the specialists to use them to evaluate the difficulty to complete the tasks and describe the problems identified during this process. The specialists used a tablet with 10-inch screen to evaluate four different IGS available to date for mobile devices, namely, GeoGebra 1, GeometryPad 2, Sketchmetry 3 and our tool Geotouch. At the end, we compile the overall score (from 0, no problem, to 6, severe problems) for each heuristic presented by each specialist. The final result is shown in Figure 8.

The results shown that the interface of Geotouch have outperformed the other three applications in all analyzed heuristics. Even for H05, H06 and H07 where the other applications have shown severe problems.

---

According to the obtained results Geotouch has successfully passed (i.e. no problems identified) in six of the heuristic criteria: H02, H03, H04, H08, H09 and H10. Nevertheless, specialists have identified not severe problems that could be easily fixed in future versions of the tool. The list of the problems were: (i) lack of correspondence between the system and the real world (e.g. no link between our system and conventional tools such as ruler and compass), (ii) no error prevention, (iii) buttons that supports recognition rather than recollection and (iv) no help button for users. Although problems were identified, all specialists indicated that they had no problems to execute the proposed tasks after a few mistakes. Furthermore, they highlighted that the capabilities to move geometric objects using fingers and share the screen with other devices can become an important tool to support collaborative knowledge construction in class.

V. FINAL CONSIDERATIONS

This study has outlined the main challenges and solutions for developing an application for teaching geometry with mobile devices in the Android platform. Producing educational software for mobile devices is a research field that is still expanding alongside the evolution of technologies that increasingly employ more resources and offer greater flexibility in the teaching-learning process. Producing educational software for mobile devices is not a simple task because it requires discovering methods of interaction and computational solutions with the aid of technologies that are constantly evolving and being updated.

The functionalities that have been developed until now, implement basic functions of Interactive Geometry such as the following: the creation and removal of points, lines and circumferences; the dynamic manipulation of created objects; and the chance of creating simple intersections like the points above objects. A number of research and technical challenges had to be overcome. Among these, the following can be highlighted: the creation of an intuitive interface for the construction of geometric objects, the establishment of a data structure to meet the needs of interactive geometry and the resolution of problems of interaction related to the drawing of geometric objects by means of touch-sensitive screens. In future studies, we intend to implement more application functionalities in a way that can foster collaborative learning among users by means of multiple devices. Furthermore, experiments with teachers and students will also be conducted to validate Geotouch and its benefits to support geometry learning.

REFERENCES