PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE

ESCUELA DE INGENIERIA

INTERACTIVE LEARNING: A COMPARISON OF INDIVIDUAL AND INTERPERSONAL COMPUTER TECHNOLOGIES

ARTURO J. TAGLE CORREA

Thesis submitted to the Office of Research and Graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Engineering

Advisor:

MIGUEL NUSSBAUM

Santiago, Chile, January 2012

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

INTERACTIVE LEARNING: A COMPARISON OF INDIVIDUAL AND INTERPERSONAL COMPUTER TECHNOLOGIES

ARTURO J. TAGLE CORREA

Members of the Committee:

MIGUEL NUSSBAUM

ALVARO SALINAS

ROSA ALARCON

JUAN E. COEYMANS

Thesis submitted to the Office of Research and Graduate Studies in partial fulfillment of the requirements for the Degree of Master of Science in Engineering

Santiago, Chile, January 2012

To my family and friends.

ACKNOWLEDGMENTS

I would like to thank Cristián Alcoholado and my advisor, Miguel Nussbaum, for their enthusiasm and for always believe in this project during this four years working together.

I also would like to thank to all the people who was in the development team and everyone who made this research possible.

TABLE OF CONTENTS

DEDICATORY ii

ACKNOWLEDGMENTS iii

LIST OF TABLES vi

LIST OF FIGURES vi

RESUMEN x

ABSTRACT xi

1. INTRODUCTION 1
   - 1.1 Objectives 1
   - 1.2 Scope 1
   - 1.3 Content 1

2. BACKGROUND 3
   - 2.1 Motivation 3
     - 2.1.1 One Mouse per Child history 4
     - 2.1.2 Historical development of the application 5
   - 2.2 Hardware 12
     - 2.2.1 Class distribution 13
   - 2.3 Software 19
     - 2.3.1 Software objectives 19
     - 2.3.2 Software architecture 19
       - 2.3.2.1 Shared components 20
       - 2.3.2.2 One mouse per child components 25
       - 2.3.2.3 Sumas Single 53

3. INTERACTIVE LEARNING: A COMPARISON OF INDIVIDUAL AND INTERPERSONAL COMPUTER TECHNOLOGIES 61
   - 3.1 Abstract 61
   - 3.2 Introduction 62
   - 3.3 Interactive arithmetic 64

3.4 Experimental design 69
3.5 Results 73
3.5.1 Quantitative results 73
3.5.1.1 Length of sessions 76
3.5.1.2 Exercises completed 76
3.5.2 Qualitative results 79

4. CONCLUSIONS AND FUTURE WORK 85
4.1 Conclusions 85
4.2 Future work 86

BIBLIOGRAPHY 87

APPENDIXES 90

APPENDIX A: LETTER OF ACKNOWLEDGMENT OF SUBMISSION 91

APPENDIX B: ARITHMETIC PEDAGOGICAL RULES 94

LIST OF TABLES

Table 2-1: MMMCommon DLL class description 21

Table 2-2: SumasCommon DLL class description 24

Table 2-3: MultiPointControl DLL class description 29

Table 2-4: Framework class description 36

Table 2-5: SumasPlugin class description 47

Table 2-6: Sumas Single class description..59

Table 3-1: Comparison of systems..70

Table 3-2: Pre-tests and post-tests results 75

Table 3-3: Number of observations by attribute 81

Table 3-4: Correlations among boredom, collaboration and interruptions attributes......81

Table B-1: Pedagogic rules for addition 94

Table B-2: Pedagogic rules for subtraction...96

Table B-3: Pedagogic rules for multiplication..98

Table B-4: Pedagogic rules for division..99

LIST OF FIGURES

Figure 2-1: First approach of the software screenshot...8

Figure 2-2: First version of the software with vertical exercises.......................................8

Figure 2-3: Second version of the software sketch 9

Figure 2-4: Second version of the software screenshot 9

Figure 2-5: Crushed application during a trial 10
Figure 2-6: First successful test with over 30 mice 10
Figure 2-7: India trials..11
Figure 2-8: Luis Gregorio Ossa School trials 11
Figure 2-9: U class distribution...15
Figure 2-10: Double U class distribution...15
Figure 2-11: Rows distribution 16
Figure 2-12: Traditional class distribution..16
Figure 2-13: Group distribution 17
Figure 2-14: Enclosed hub 17
Figure 2-15: Briefcase, computer and projector 18
Figure 2-16: MMMCommon DLL class hierarchy...21
Figure 2-17: SumasCommon DLL class hierarchy...23
Figure 2-18: MultipointControl DLL class hierarchy..28
Figure 2-19: IMMPlugin interface and MMMPluginInfo class diagram 30
Figure 2-20: BasicMouse class diagram 31
Figure 2-21: Framework flow chart 33
Figure 2-22: Framework dependencies 34
Figure 2-23: Framework class hierarchy...35
Figure 2-24: Kidbox screenshot...40
Figure 2-25: Kidbox displaying wrong feedback...41
Figure 2-26: Kidbox displaying correct feedback..41

Figure 2-27: Kidbox displaying next level feedback 41

Figure 2-28: Kidbox displaying sleep feedback..42

Figure 2-29: Kidbox displaying deep sleep feedback 42

Figure 2-30: SumasPlugin screenshot 43

Figure 2-31: Points bars detailed screenshot..44

Figure 2-33: Summary window screenshot..45

Figure 2-32: Teachmode screenshot 45

Figure 2-34: SumasPlugin class hierarchy...46

Figure 2-35: SumasPlugin basic flow chart 49

Figure 2-36: Kidbox flow chart..50

Figure 2-37: Teach mode flow chart..51

Figure 2-38: SumasPlugin dependencies 52

Figure 2-39: Classmate PC (CMPC)...55

Figure 2-40: Sumas Single screenshot 56

Figure 2-41: Sumas Single flow chart..57

Figure 2-42: Sumas Single class hierarchy 58

Figure 2-43: Sumas Single dependencies 60

Figure 3-1: ITS interface display states 66

Figure 3-2: Interpersonal computer information display 66

Figure 3-3: Netbook information display...68

Figure 3-4: Number of completed exercises per student by session and technology group 78

Figure 3-5: Level reached by end of experiment 78

Figure 3-6: Observations of collaboration by observed session 82

Figure 3-7: Observations of exercise help request by observed session 82

Figure 3-8: Boredom by time interval over a single session.. 84

RESUMEN

La introducción de tecnología en la sala de clases cada día cobra más importancia. Entre las distintas tecnologías que han surgido últimamente, la iniciativa *"One Laptop per Child"* ha sido la que ha tenido un mayor impacto debido a su bajo costo de cerca de US$200 por dispositivo. Sin embargo este precio aun es muy caro para países en vías de desarrollo. *"One Mouse per Child"* surge como una alternativa de menor costo que *"One Laptop per Child"*. Esta consiste en el uso de un mouse por alumno con un computador interpersonal, compartido por toda la clase.

En la siguiente tesis se presenta una descripción del software utilizado en los experimentos y luego se presenta un estudio comparativo entre el uso de tecnologías de computadores personales e interpersonales para una aplicación de enseñanza de aritmética en un curso de tercero básico. Los grupos que trabajaron con tecnología fueron contrastados con un grupo que trabajo en ejercicios similares en papel y lápiz. En este estudio, los resultados experimentales mostraron que hubo diferencias significativas entre el grupo que trabajó con un computador interpersonal y el grupo que trabajó con papel y lápiz, a pesar de que los últimos realizaron una cantidad mayor de ejercicios. Este resultado sugiere que las diferencias en el aprendizaje se deben a la presencia de retroalimentación inmediata, sea esta mostrada de manera pública o privada.

Esta tesis cuenta con el apoyo del proyecto Fondecyt 1100309.

Palabras Claves: Un *mouse* por niño, computador interpersonal, tecnología, educación.

ABSTRACT

Every day, introduction of technology in the classroom becomes more important. Among several technologies, “One Laptop per Child” initiative has had the greatest impact for its low cost of around US$200 per device. But even at that price, it’s still very expensive for developing countries. “One Mouse per Child” arises as a lower cost alternative than “One Laptop per Child”. It consists in the use of one mouse per student and an interpersonal computer shared by the whole classroom.

In this thesis, a description of the software used in the experiment is presented and then a comparative study between the use of personal and interpersonal computer technologies for an application for arithmetic teaching in a third grade class. The groups that worked with technology were compared with a group that did similar exercises using pen and paper. In this study, results showed statistically significant differences between the group that used an interpersonal computer and the group that used pen and paper, although the latter did more exercises. This result suggests that differences in learning outcome are because of the presence of immediate feedback, either shown in a shared display or privately.

This thesis was partially supported by project Fondecyt 1100309.

Keywords: One mouse per child, interpersonal computer, technology, education.

1. INTRODUCTION

1.1 Objectives

The objective of this study is to determine differences in children's learning outcome and class behavior when they interact with personal and interpersonal computer technologies in a multi-session experiment conducted in a Chilean primary school.

The specific objectives of this study are:

1) To determine, in terms of children's learning, which technology is better: personal computer or an interpersonal computer using One Mouse per Child.
2) To determine differences of providing public or private instant feedback.

1.2 Scope

This study includes three major phases:

1) A software development for both the personal and interpersonal computer.
2) Software implementation in a public school.
3) Result analysis and conclusions

It was not included in the scope of this study the research of the teacher's role in the classroom and how a proper orchestration and the integration of digital and non-digital resources may affect the children's outcome.

1.3 Content

The next chapter includes the motivation of this study and a brief history of One Mouse per Child technology. Then, it is described the hardware and software used in the experiments.

In chapter three of this document, it’s presented the experimental design and results. Conclusions and future work are presented in the last chapter.

2. BACKGROUND

2.1 Motivation

Over forty years have passed since the first computer-related technologies have been introduced in school classrooms (Feurzeig, Papert, Bloom, Grant, & Solomon, 1970), but still some authors say the impact of introducing technologies in classrooms and the cost of doing it is not very clear, especially when it comes to the total cost of ownership (Oppenheimer, 2004; Trucano, 2005).

In the past few years, One Laptop per Child has led cheap classroom technologies at a cost of around US$200 per device (OLPC, 2011). However, still at this price, these solutions are too expensive to be implemented in most of developing countries (Trucano, 2010; Toyama, 2011) and it would be cheaper to continue teaching the same way it has been since beginnings of 20th century: A teacher with a whiteboard and a chalk speaking to his class. But, as Foreman (2003) says, there is a gap between the perceived cost of a traditional classroom and a classroom with technologies:

> “Cheap products always conceal their total cost. In the case of the large lecture and its student consumers, the cost (could it be quantified) is the lost opportunity for more meaningful and more enduring learning. The fact is that too many students in large lecture courses are uninterested pragmatists who cram for tests, commit the material to short-term memory, and quickly forget it thereafter.”

In this context, trying to introduce technology in the classroom with very low cost, One Mouse per Child is conceived. In this model, over 40 students can interact at one time by using a single computer, a projector, hubs and one mouse per user.

The idea of using an interpersonal computer dates back to 1991, where Bier and Freeman designed a user interface and software architecture to be used by groups sharing a single workstation and screen (Bier & Freeman, 1991).

2.1.1 One Mouse per Child history

The idea of using multiple mice in one computer was first developed by Inkpen *et al* (1995), who experienced with two mice in the same computer. Eight years later, Stanton and Neale (2003) compared the performance of two students sharing one mouse versus each with a mouse, showing great collaboration and division of tasks in the groups that each student had a mouse.

In 2005, the concept of using more than one input device in one computer was again taken up by UC Berkeley's Ph.D. student Joyojeet Pal, while he was doing an internship in Microsoft Research India (MSR India). During his stay, he researched about how computers were used in India's schools. He discovered that computers were always used by 2, 3 and even 5 students at one time. From this research, the idea of giving each student one mouse arise. The objective of giving one mouse to each student is for all to work at the same time in one computer either in a personal or collaborative activity (Microsoft Developer Network, 2008). The same year Udai Pawar started working in MSR India, who developed the first multiple mice prototypes and experiments. In Pawar's experiments, only 4 mice where attached to one computer (Pawar, Pal, & Toyama, 2006).

Multiple mice technology began to expand because Pawar, while he was working in MSR India also developed a Software Development Kit (SDK) for multiple mice applications, called Multipoint Mouse SDK. This SDK was published for free use for people who wanted to develop Multimouse applications on January of 2007. Today, this SDK is in its 1.5 version and still is maintained by Microsoft.

Since this SDK was released, some applications have come out. The most notorious has been Mouse Mischief developed by Microsoft, which acts as Microsoft PowerPoint plug-in and allows teachers to create interactive lessons which students are allowed to interact using their own mouse (Microsoft, 2010).

2.1.2 Historical development of the application

The development of the One Mouse per Child arithmetic application started in the beginning of the year 2008, and a lot has passed since then. In this section different versions of the software will be reviewed until the final version was implemented.

At the end of the year 2008, the first version was created. It consisted in random exercises that need to be solved by the students, who had a number that matched an exercise solution. In this version, all the mice were allowed to move around the screen as can be seen in Figure 2-1 and Figure 2-2. This turned out to be a total chaos. After a trial, it was possible to notice that students were unable to recognize their own mouse because of what we called the “flies effect”, where all the mice flew around the screen and no one could identify their own pointer. There were also some flaws in the design of the pedagogic activity.

This activity was redesigned with the help of teachers and the main difference with the previous version was that the students were restricted to their own work space instead of moving around the whole screen to eliminate the "flies effect", as it can be seen in the sketch of Figure 2-3. It was also added an overall score and an identification symbol for the students so they can recognize themselves easier in the screen that also worked as a confirmation button. Finally, it was added an individual feedback in the figure of happy or sad face when the students enter their answer, or a sleepy face when the student was not working (Figure 2-4). The objective of the feedbacks is to reinforce the acquired information and it helps learners benefit from practice (Bangert-Drowns, Kulik, Kulik, & Morgan, 1991; Butler & Winne, 1995; Kluger & deNisi, 1996).

Like the first time, the trial was a complete disaster, but the cause this time were software limitations, not the pedagogical model, which is maintained until today for the arithmetic application. Due to the Multipoint SDK limitations, it was impossible to work with more than 10 students because the software got lagged with too many mice connected in the beginning, and crushing at the end (Figure 2-5). This hypothesis was proven the day after, were a successful test was done with 8 students working simultaneously showing the viability of the pedagogical model.

After that trial, some modifications were done to the Multipoint SDK to support more mice and in May of the next year it was possible to conduct a successful test with over 30 students (Figure 2-6). After that, in July of 2009 tests in India were made (Figure 2-7) and a comparison study was done that showed that even in different environmental conditions, it is possible to implement this technology with minimal equipment at a very low cost (Alcoholado, et al., 2011).

After the India experience, 66 pedagogical rules (appendix B) based on criteria defined by the Ministry of Education of Chile (MINEDUC, 2011) were added to match the curriculum requirements of schools and a design phase was done to improve the software graphics. With the pedagogical rules, a multi-session test was done at Luis Gregorio Ossa School, were 40 students divided in two groups of 20 increased their performances in 17.86% after 7 sessions carried out twice a week (Alcoholado, et al., 2011).

After the Luis Gregorio Ossa School trials (Figure 2-8), in 2010 a 7 month study was conducted. This study is described in chapter two of this thesis. Today, a second 7 month study is being done in 5 different schools in Santiago.

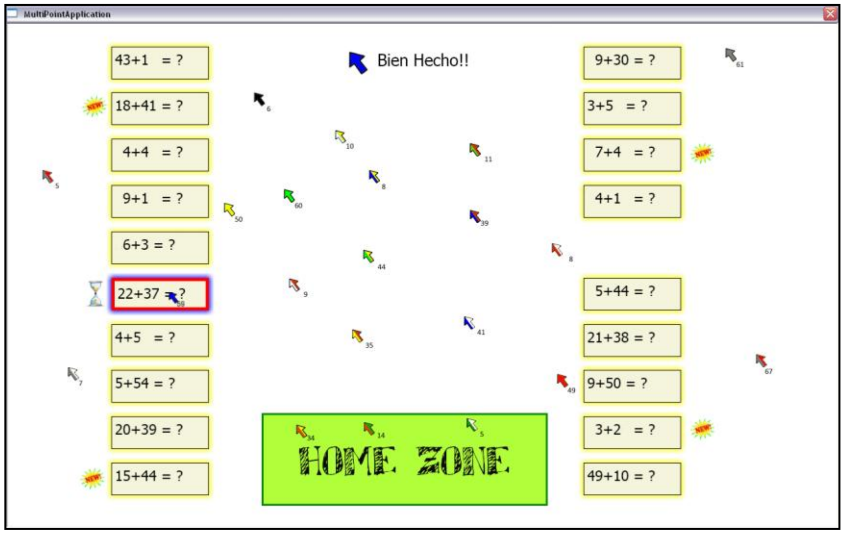

Figure 2-1: First approach of the software screenshot

Figure 2-2: First version of the software with vertical exercises

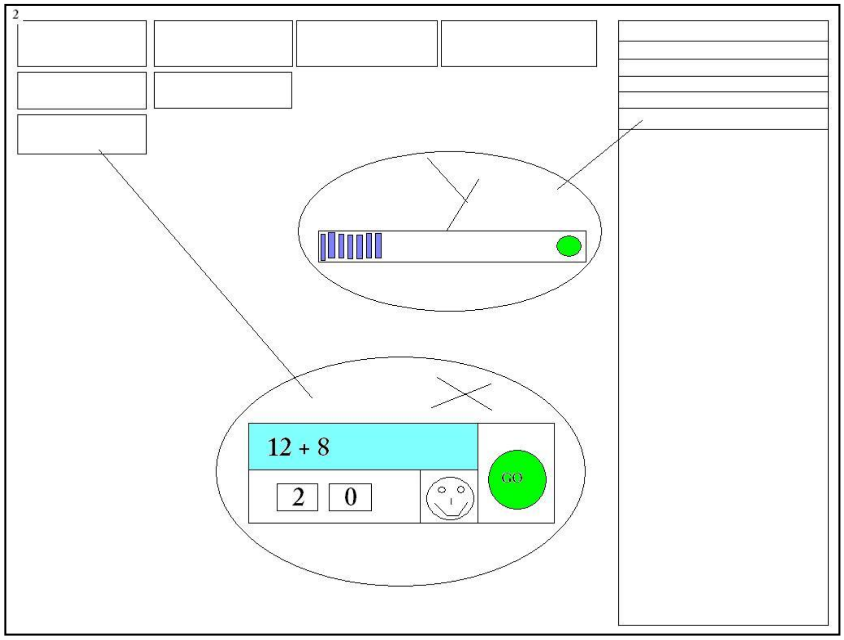

Figure 2-3: Second version of the software sketch

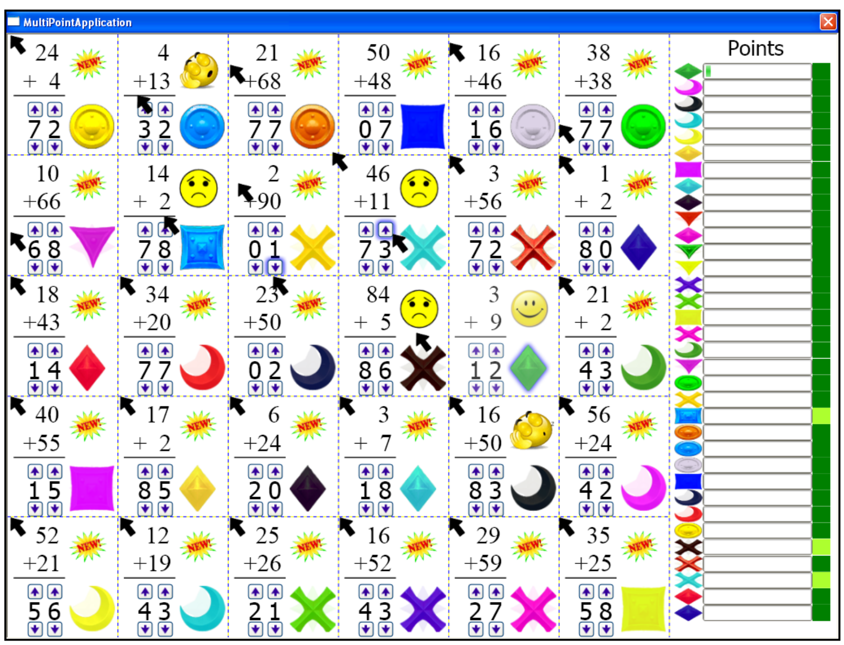

Figure 2-4: Second version of the software screenshot

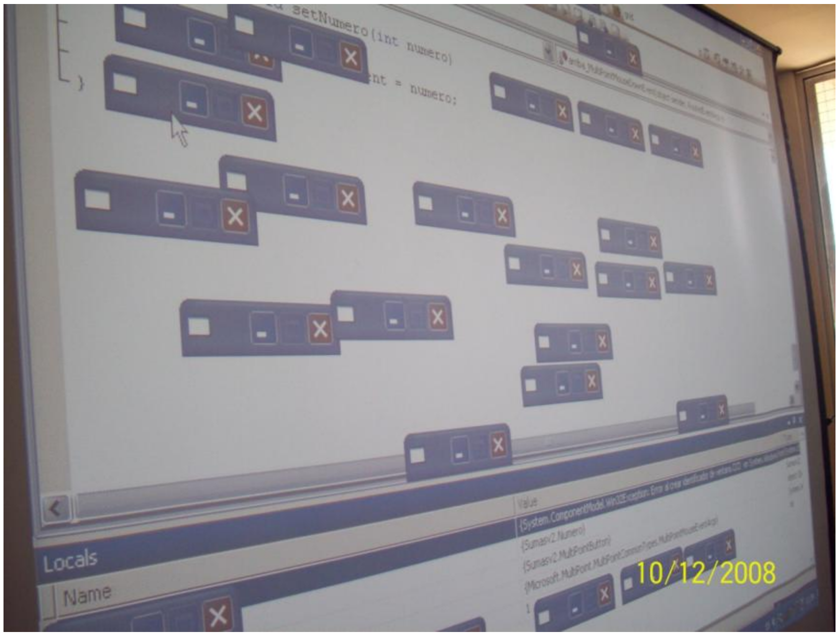

Figure 2-5: Crushed application during a trial

Figure 2-6: First successful test with over 30 mice

Figure 2-7: India trials

Figure 2-8: Luis Gregorio Ossa School trials

2.2 Hardware

In order to implement a One Mouse per Child solution, it is required a computer, a projector, some USB hubs and mice. In this section the requirements that this elements must match will be reviewed.

1) Computer

The computer, to be compatible with the implemented solution must be a 32-bit Windows computer, with the XP version or further. Also, the computer must have at least a 1.8 GHz dual-core processor and a memory of 2 Gb. Work is been done to make the software compatible with 64-bit computers.

2) Projector

To visualize everything correctly in the screen, it’s recommended to use a projector with a native resolution of 1024x768 pixels. Projectors with native resolution of 800x600 pixels had been used, but in large classrooms some images can get blurry.

3) USB hubs

This are required to increase the available USB ports in a computer. The only technical requirement is that they must have connection to the electric network. Some hubs don’t have and it’s highly possible that mice behave erratically due to the lack of electric power. The amount of hubs will depend on the class distribution and the total of people working. D-Link 7-port hub model DUB-H7 have shown a great performance in different trials working with 30 to 40 mice.

4) Mice

The only requirement is that mice must have a USB connection. PS/2 mice are not supported.

2.2.1 Class distribution

Another challenge has been to deploy the equipment described before in a classroom. Normally the space is limited and there are a lot of wires around, because of the mice and USB hubs.

After several trials in classrooms, we have proved that maybe the best distribution is the “U distribution”, as shown in Figure 2-9. In this distribution, tables and chairs are arranged in the classroom in the form of a U, with all the wires facing inside the U and the computer and projector placed in the middle of the U, letting the students move around the classroom outside the U without risk of tripping with wires.

But sometimes the space available in a classroom is not enough for making a U distribution or in some cases there are too many students. In those cases, we have tested alternative distributions like the double U distribution, the row distribution or the traditional class distribution.

In the double U distribution, a U distribution is formed and then some students are placed inside the U, like in Figure 2-10. The main disadvantage of this distribution is the risk of students inside the U of tripping with the wires.

The row distribution (Figure 2-11) has been used when the space in the classroom is very limited. When there is no possibility to move the desk, we have tested the traditional class distribution (Figure 2-12). Most recently, a group distribution shown in Figure 2-13 has been tested in some Chilean schools, where students gather in groups around one table.

This distribution has been tested using a mice-briefcase where all the cables are pre-plugged and the hubs are enclosed to prevent disconnections as shown in Figure 2-14. The advantage of this system is that it is really easy and faster to deploy in a classroom than the others, but it’s more expensive than just plug-in manually all the equipment. The briefcase, as shown in Figure 2-15, also has a power supply to plug the computer and projector.

Figure 2-9: U class distribution

Figure 2-10: Double U class distribution

Figure 2-11: Rows distribution

Figure 2-12: Traditional class distribution

Figure 2-13: Group distribution

Figure 2-14: Enclosed hub

Figure 2-15: Briefcase, computer and projector

2.3 Software

For the study presented in section 3, two versions of the arithmetic software were developed: one for personal computers and another for an interpersonal computer, using one mouse per child as input. Both programs were developed using C# and .NET 3.5. Additionally, the one mouse per child version uses Microsoft Multipoint 1.0 Software Development Kit (SDK) that allows recognizing several mice plugged in to the computer, and adds some basic functionality.

2.3.1 Software objectives

The main objective of the software is teaching arithmetic. This is achieved by giving the student an exercise that needs to be solved. The student enters an answer and feedback is given so the student knows if his answer was correct or incorrect. The software adapts to each student controlling the level of the exercise given. There are 66 pedagogic rules implemented based on Chilean Ministry of Education math curriculum (MINEDUC, 2011). These rules can be found in Appendix B.

2.3.2 Software architecture

Two versions of the software were implemented: the one mouse per child version, with n students working in an interpersonal computer or 1:n, and the single version, for personal computers or 1:1. Some parts of the software are shared between the one mouse per child and the single version, and other components are specific for each version. In this section the different components of both softwares will be described.

2.3.2.1 Shared components

The shared component consists in two DLL's (dynamic link library) that provide functionality for common tasks. The first one of this DLL's is MMMCommon and the second is SumasCommon which are described below.

2.3.2.1.1 MMMCommon DLL

This library provides course, students and session management functions. It also provides functionality for reading and writing this data in the correct XML format. Figure 2-16 shows the class hierarchy for this DLL and Table 2-1 shows a brief description of each class.

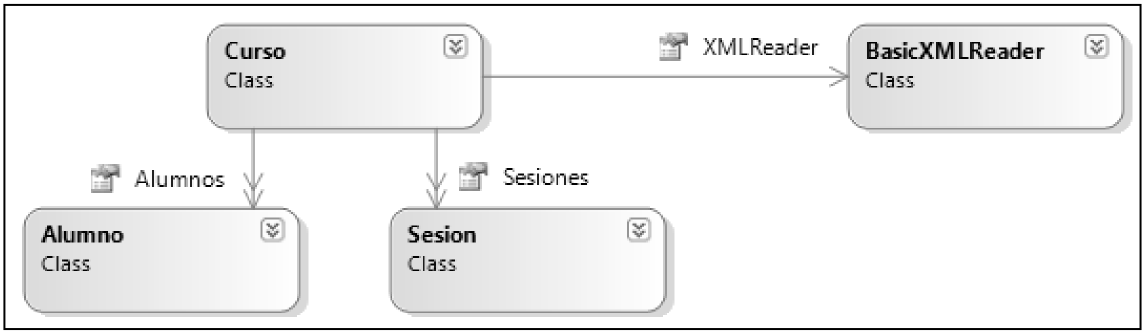

Figure 2-16: MMMCommon DLL class hierarchy

Table 2-1: MMMCommon DLL class description

Class	Description
Alumno	Contains basic information for each student such as id, name, last name and identification symbol.
BasicXMLReader	Provides functionality for reading and writing into the Xml file that stores the course information.
Curso	Contains a list of students and the sessions they have participated. It also has an XMLReader object for writing and reading this data stored in a Xml file.
Sesion	Contains basic information about sessions that students have played such as date and duration.

2.3.2.1.2 SumasCommon DLL

This library provides specific functions for the generation of exercises according to the rules defined by the Ministry of Education (MINEDUC, 2011). Specifically, the *Reglas* class contains the logic to generate an exercise of any of the 66 levels implemented in the software.

It also extends the *Curso*, *Alumno* and *Sesion* classes to provide specific functionality regarding the exercises and the arithmetic application. Figure 2-17 shows the DLL's class hierarchy and Table 2-2 briefly describes the main characteristics of these classes.

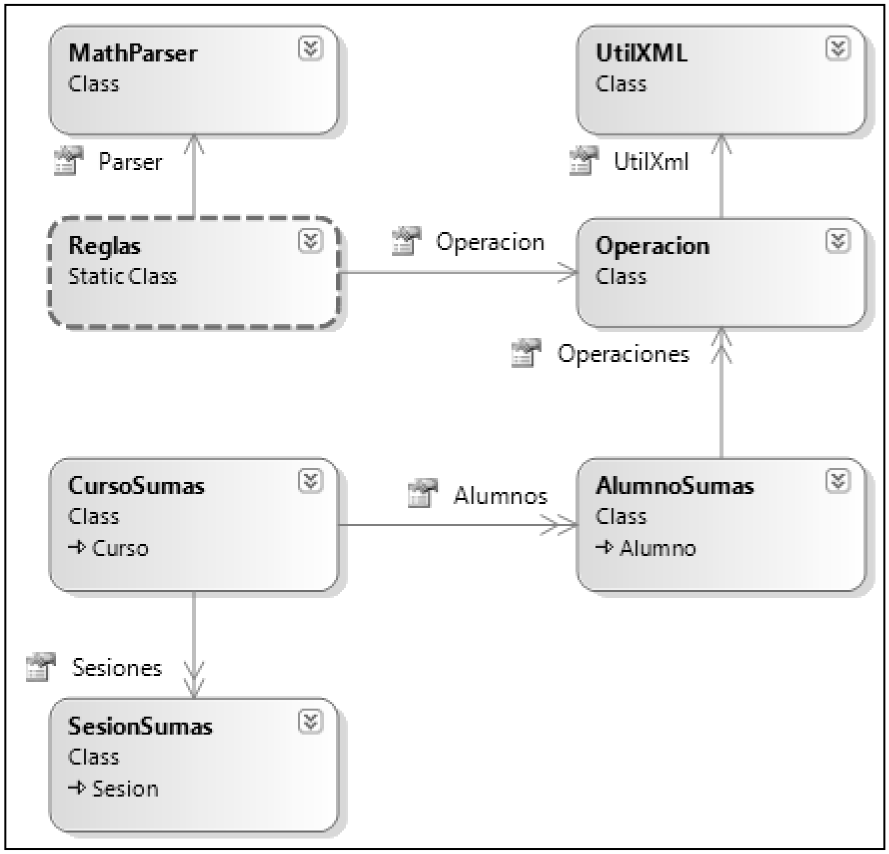

Figure 2-17: SumasCommon DLL class hierarchy

Table 2-2: SumasCommon DLL class description

Class	Description
AlumnoSumas	Extension of *Alumno* from MMMCommon that adds the ability to store a list of operations
CursoSumas	Extension of *Curso* from MMMCommon to provide specific functionality for the arithmetic plug-in
MathParser	This class provides functions that supports the exercise generation.
Operacion	Holds information about the operations generated, such as the exercise, the correct answer and the points obtained.
Reglas	Static class that generates operations for each of the 66 levels of addition, subtraction, multiplication and division developed based on the curriculum proposed by the Ministry of Education.
SesionSumas	Extension of *Sesion* from MMMCommon that adds the ability to support arithmetic sessions.
UtilXML	Reads and write the operations into the Xml file that stores the course data.

2.3.2.2 One mouse per child components

This software supports up to 43 mice connected at the same time. To achieve this, Multipoint SDK 1.0 developed by Microsoft in 2007 was used. This SDK originally supported up to 8-12 mice depending on the computer's characteristics due to performance problems, but some modification and optimizations were done to reach 43 mice working simultaneously, transforming a personal computer into an interpersonal computer. Because of limitations of Windows on the recognition of USB devices, it has not been possible to increase this number. However, this is a significant improvement over the actual version of the SDK (1.6) that supports only up to 25 mice simultaneously (Microsoft, 2011). It hasn't been possible to replicate the optimizations done in version 1.0 of the SDK due to restrictions in the source code of version 1.6, but everything indicates this might be possible having access to the source code.

Multipoint 1.0 provides basic functionality to develop multimouse applications. This library handles the device recognition and it draws them in the screen. In addition, it handles mouse events such as movement and click events and other events more sophisticated like connection and disconnection of devices.

The software developed is divided in two main components: the framework and the plug-ins or applications. The framework pretends to be a common base for all one mouse per child applications. It solves several one mouse per child applications common problems such as mouse recognition and provides basic functionality for course management, XML data reading and writing and teacher functions.

The applications are developed as plug-ins for the framework described before. This plug-ins needs to implement IMMMPlugin interface that comes in MultipointControl DLL. The interface not only enables the plug-in to implement an application to be used with the framework, but it also gives the ability to extend some of the functionalities provided by it. For the experimental activity, an arithmetic application was developed. This application is described in section 2.3.2.2.3.

2.3.2.2.1 MultipointControl DLL

This library has some common controls such as buttons or labels ready to use re-implemented to work with multipoint events. This re-implementation is very important because common WPF controls don't recognize multipoint events, so they are useless in multipoint environments. In order to make this re-implementation easier, MultiPoint SDK declares an interface that must be implemented in a control in order to make it capable of recognizing multipoint events. Controls in this library implement this interface and some provides further functionality. It's possible to find the DLL's class hierarchy in Figure 2-18 and a brief class description in Table 2-3.

In this DLL it is also the declaration of IMMMPlugin interface. This interface provides the capability to generate the communication between the framework and the plug-ins.

MMMPluginInfo class defines some attributes that plug-ins need to declare in order to get basic information about the plug-in during the opening of it. Figure 2-19 shows a UML class diagram for these classes.

Finally, it is the BasicMouse class. An UML class diagram can be found in Figure 2-20. This class work as a wrapper for DeviceInfo class from Multipoint SDK in order to make an easier way to interact with this class and to provide new functionalities and properties such as define a teacher mouse, show, hide, enable and disable a mouse, among others.

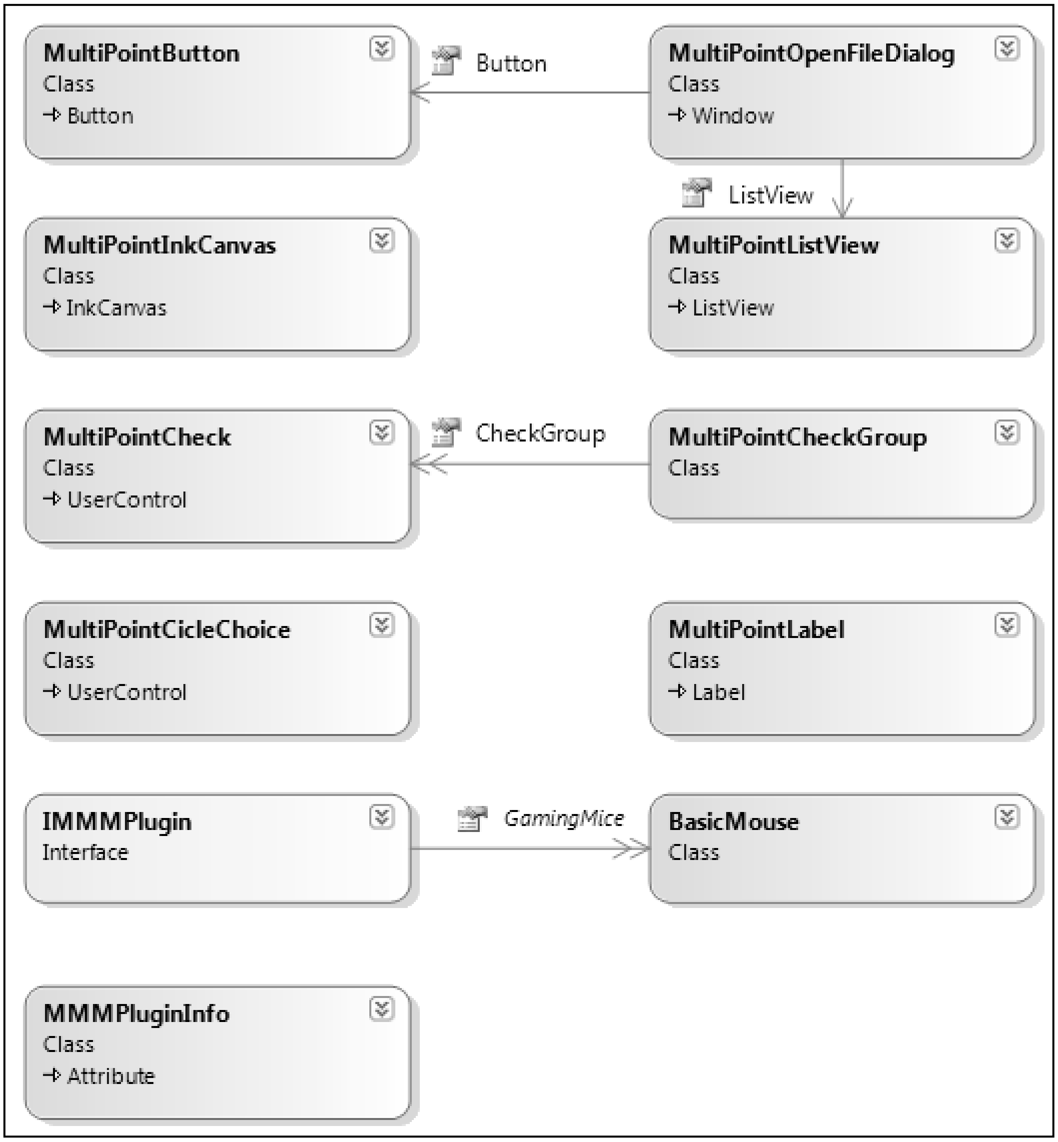

Figure 2-18: MultipointControl DLL class hierarchy

Table 2-3: MultiPointControl DLL class description

Class	Description
BasicMouse	Wrapper for Multipoint's DeviceInfo class to facilitate its use and provide new functionality.
IMMMPlugin	Interface that defines the methods and properties that every application needs to define in order to become a plug-in for the framework.
MMMPluginInfo	Inherited from Attribute, defines attributes that are read using reflection when a plug-in is opened.
MultiPointButton	Inherited from Button, this class extends this functionality to work with Multipoint events.
MultiPointCheck	Implements a checkbox that supports multipoint.
MultiPointCheckGroup	Changes the behavior of a group of MultiPointChecks to behave like radiobuttons: only one MultiPointCheck checked at a time.
MultiPointCicleChoice	Shows an element from a list and two controls, up and down, to change it.
MultiPointInkCanvas	Inherited from InkCanvas, this class extends this functionality to work with Multipoint events. It is set to paint with a red line when left button is down and with yellow highlighter when right button is down.
MultiPointLabel	Inherited from Label, this class extends this functionality to work with Multipoint events.
MultiPointListView	Inherited from ListView, this control provides a way to show a list of items, with vertical and horizontal scroll bar.
MultiPointOpenFileDialog	Implements an open file dialog that supports multipoint.

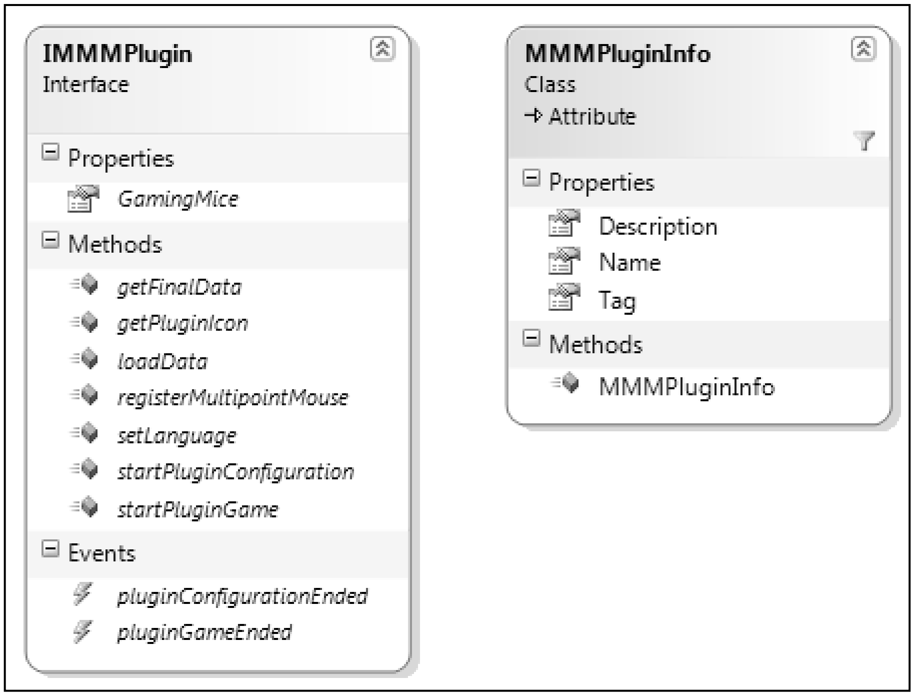

Figure 2-19: IMMPlugin interface and MMMPluginInfo class diagram

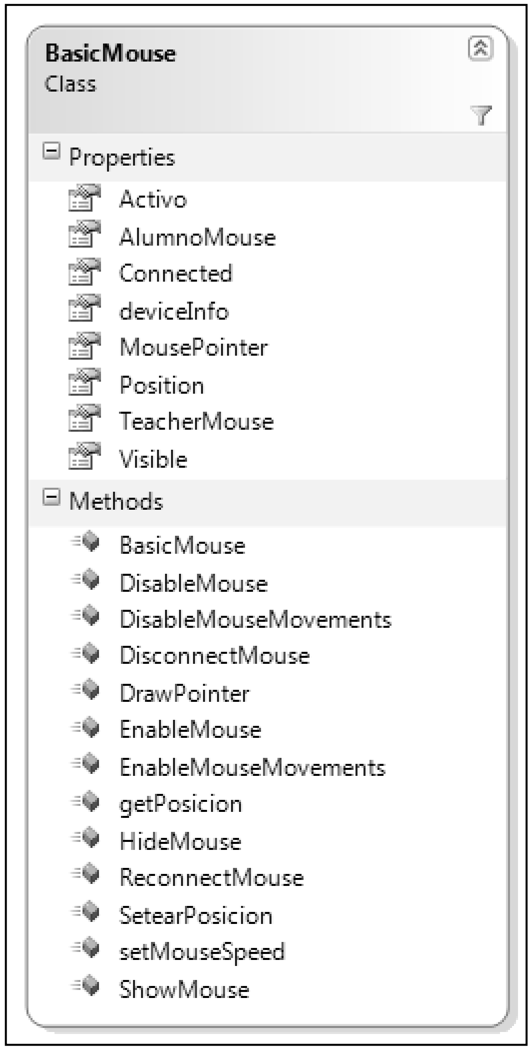

Figure 2-20: BasicMouse class diagram

2.3.2.2.2 Framework

This framework was motivated by the idea of developing one mouse per child applications in faster and easier way, in order to make a skeleton for applications that can be customized by developers (Johnson, 1997). Like all frameworks, it has non-modifiable code but it can be extended. Also, the controller of the software it's the framework and not the extension (inversion of control) and finally, it has a default behavior is defined by the framework shown in

Figure 2-21. The application may interact with the framework in three points of the program: In the application configuration process, the main application activity and the application summary. The first one, in case it's defined by the application, it's an opportunity to ask the user for settings before the main activity is initialized. The second one is the main activity of the application, defined by the application developer, and finally, there is a chance to show a summary of the activity, in case the latter requires it.

In order to develop new applications for this framework, it's necessary to make a plug-in. To create one, the developer needs to implement IMMMPlugin interface from MultiPointControl DLL described in section 2.3.2.2.1.

As shown in Figure 2-22, the framework depends on DLLs MultiPointControl and MMMCommon. Figure 2-23 shows the class hierarchy diagram. As it can be seen, it's a very simple class hierarchy, which handles the flow described in

Figure 2-21.

Finally, Table 2-4 has a description of the classes described in the class hierarchy diagram.

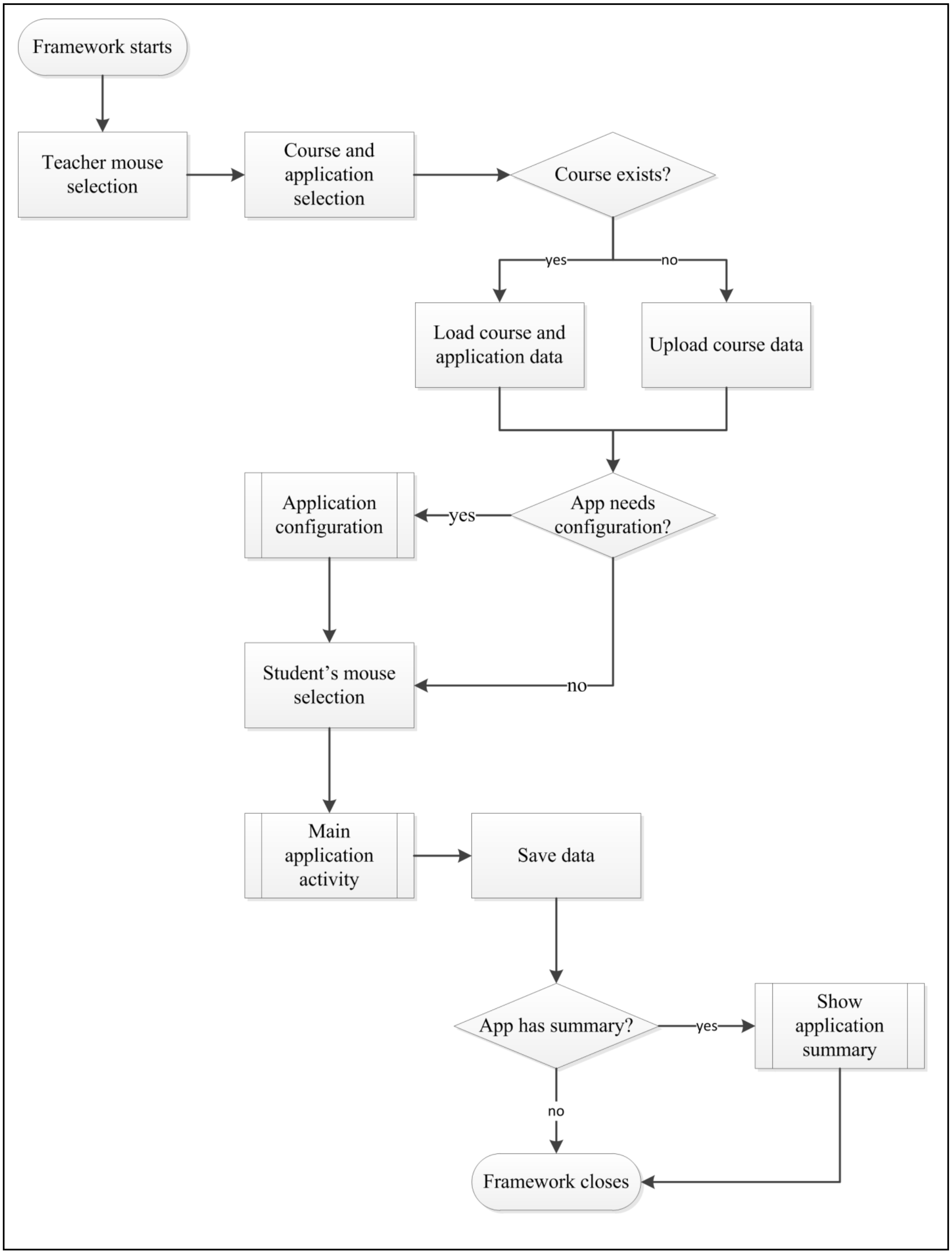

Figure 2-21: Framework flow chart

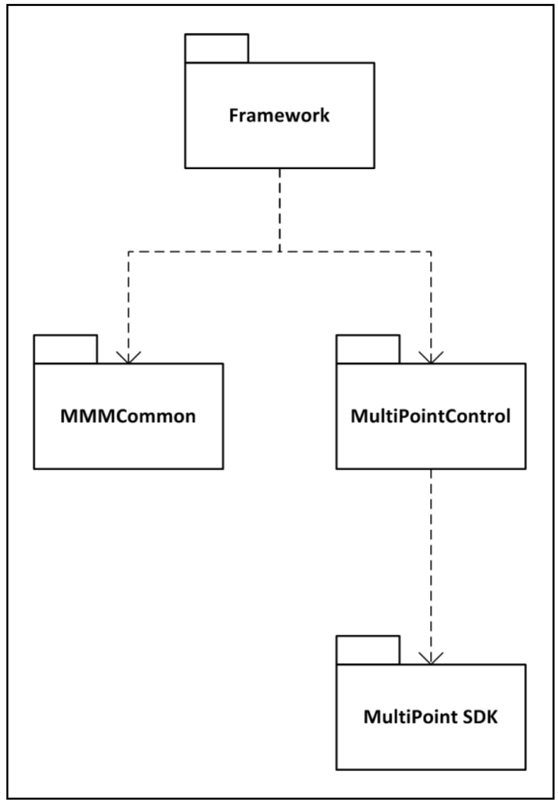

Figure 2-22: Framework dependencies

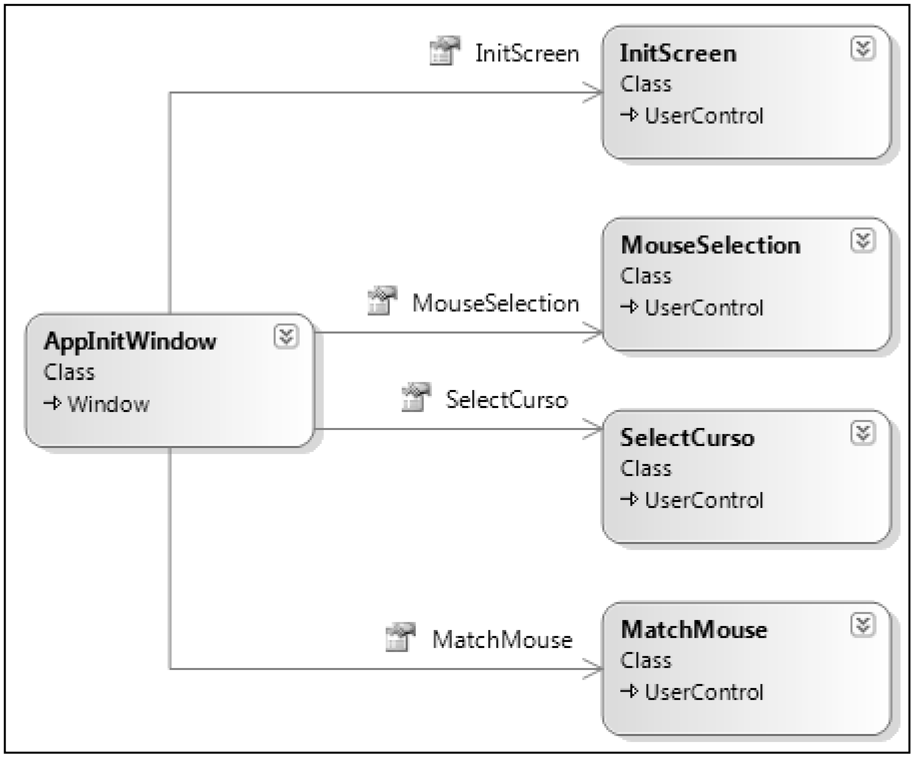

Figure 2-23: Framework class hierarchy

Table 2-4: Framework class description

Class	Description
AppInitWindow	Main window of the program. Controls the flow of the framework.
InitScreen	Shows the application logo and version number.
MatchMouse	This control allows the students to select their name from the list of the class in order to do the match between mice and students.
MouseSelection	Allows selecting a teacher mouse, by pressing the M key and right clicking. It also gives the chance to reset the selection in case someone else gains control of the teacher mouse by pressing the R key. After the teacher mouse is selected, it turns red to differentiate it from the other mice.
SelectCurso	Control that allows the teacher to select the course and the application that the course will work. After the course and application are selected, the data is obtained from the XML file.

2.3.2.2.3 SumasPlugin

The application was developed in order to use the arithmetic software in a one mouse per child environment using it as a plug-in for the framework described in the previous section.

When this plug-in starts its main activity, the screen is divided in one section for every mouse connected to the computer and one kidbox is drawn for each section. The kidboxs are the space where students can work.

Figure 2-24 shows a kidbox. The red box is the current exercise display space. The green box is the space where the student has to enter the answer by clicking the arrows of the digits. Finally, in the blue box it's the identification symbol, which in this case it also works as a confirmation button that the student has to click when he student thinks it's ready.

When the student clicks the symbol, a feedback is displayed depending on whether the answer was correct or incorrect. If the answer is wrong, feedback is shown (Figure 2-25) and the student must try again. If the answer is correct, feedback is shown (Figure 2-26) but there is also a possibility that the student may have passed to the next level and another kind of feedback is shown to let the student know he is in the next level (Figure 2-27). The conditions to get to the next level are to answer the first 10 exercises correct or to answer the last 8 exercises out of the last 15 correct. By doing this a student is enforced to learn how to solve an exercise and not doing it by trial and error. Additionally there are two more feedbacks that a student may receive: when he is not working or "sleeping" (Figure 2-28) or when he is not working at all or in a "deep sleep" state (Figure 2-29).

Figure 2-30 shows a screenshot of the main activity of plug-in working with 36 mice at one time. The screen, in addition to the kidbox, shows in the right side the points section, where one point bar is displayed for every mouse that is participating on the activity.

The points bars, detailed in Figure 2-31, have 3 main elements: the symbol in the left to indicate which student represents, the number in the right that shows the level of the students, and the points won. Each exercise is represented by a rectangle that may be filled with green if the exercise was answered right by the student in the first attempt, yellow if it was right in the second attempt or red if it was right after three or more attempts.

A basic flow of the plug-in is shown in Figure 2-35. Once the Kidboxs are drawn and sorted, in the "kidbox flow" process, the math activity start, following the flow described in Figure 2-36. This flow continues until the application is closed or the teacher enters the Teachmode. When the latter occurs, the application follows the flow described in Figure 2-37. Again, this flow continues until the teacher interrupts it.

The Teachmode, as shown in Figure 2-32, consist in a special module where the teacher can show how to solve a specific exercise to the whole class. When the program goes to this mode, all mice are hided and only the teacher mouse can interact. The teacher besides the possibility to solve an exercise using the same kidbox that students have, it also has a highlighter tool (blue box in Figure 2-32) and a pen tool to write in the screen (red box).

When the Teachmode is closed, the flow continues with the main activity and all the students return to their exercises and continue where they left them.

When the application is closed, the kidbox flow is interrupted and the data of the students is sent back to the framework for saving the file. After the file is saved, the framework shows the plug-in's summary window (Figure 2-33).

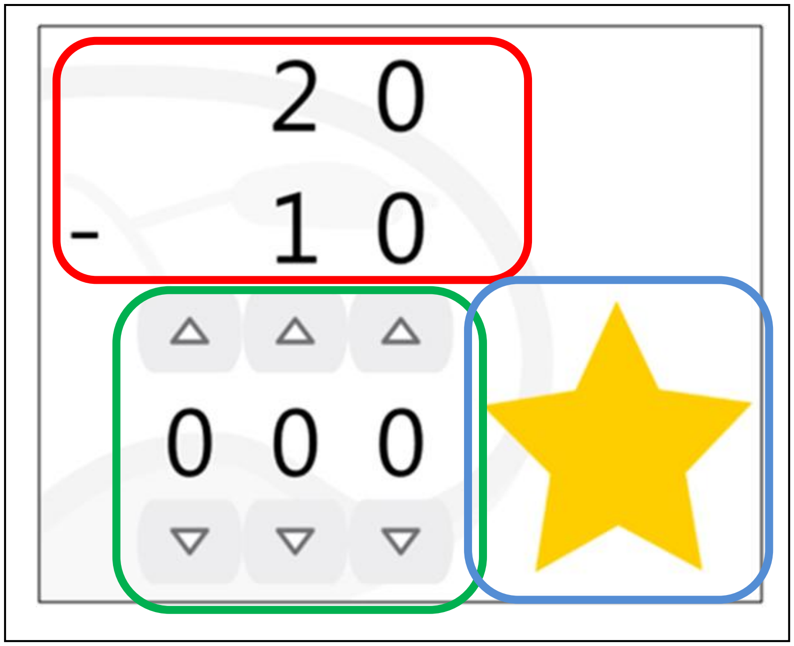

Figure 2-24: Kidbox screenshot

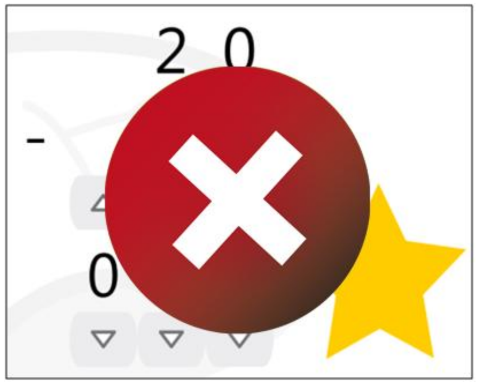

Figure 2-25: Kidbox displaying wrong feedback

Figure 2-26: Kidbox displaying correct feedback

Figure 2-27: Kidbox displaying next level feedback

Figure 2-28: Kidbox displaying sleep feedback

Figure 2-29: Kidbox displaying deep sleep feedback

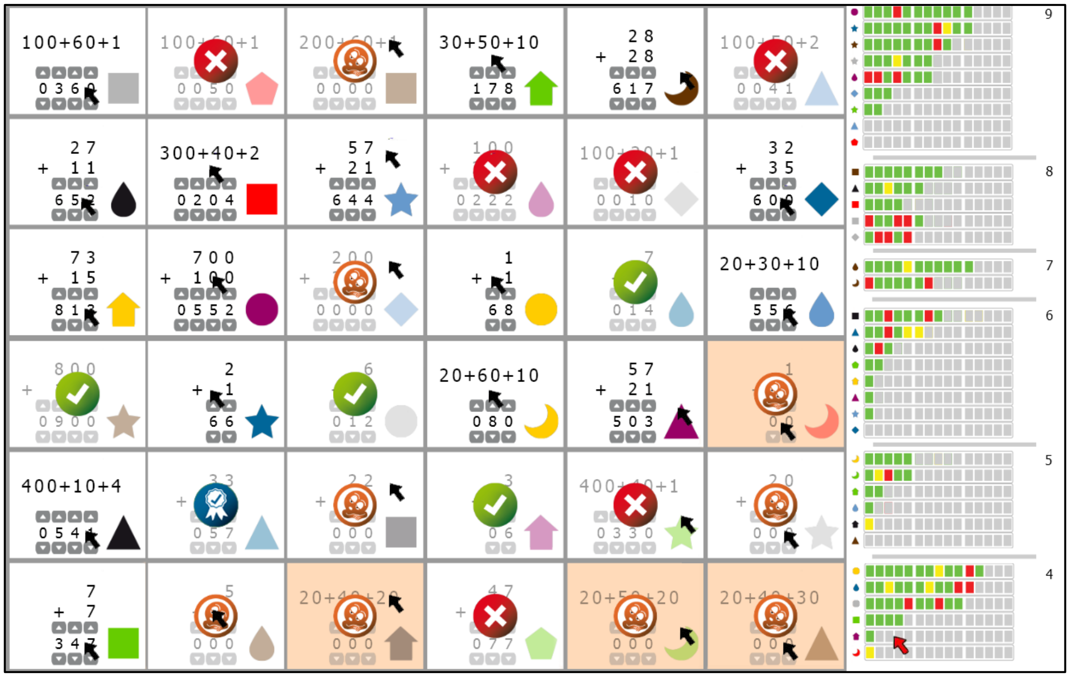

Figure 2-30: SumasPlugin screenshot

Figure 2-31: Points bars detailed screenshot

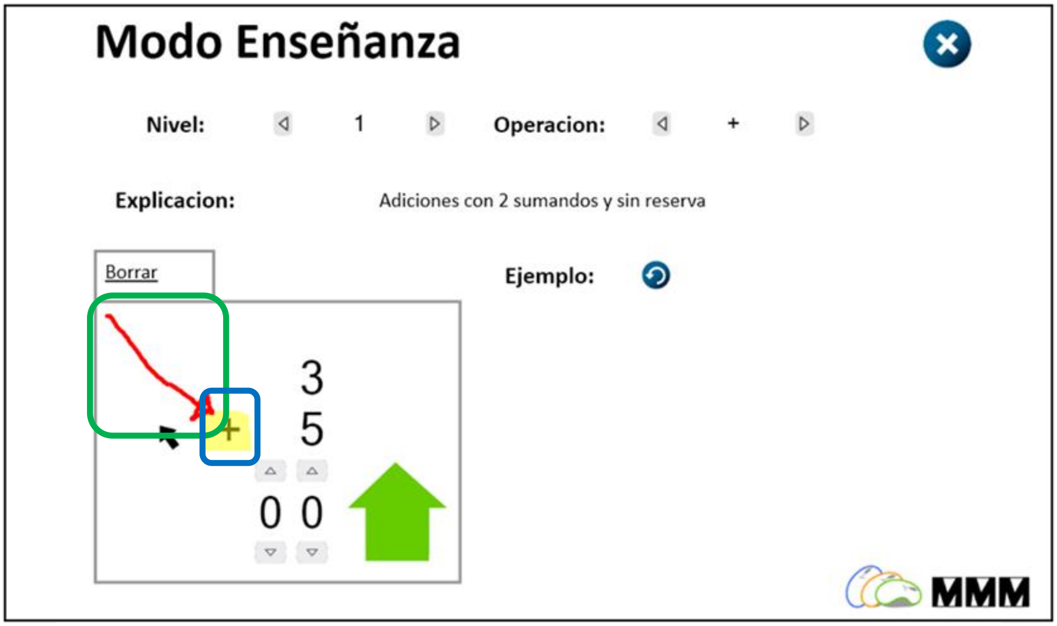

Figure 2-32: Teachmode screenshot

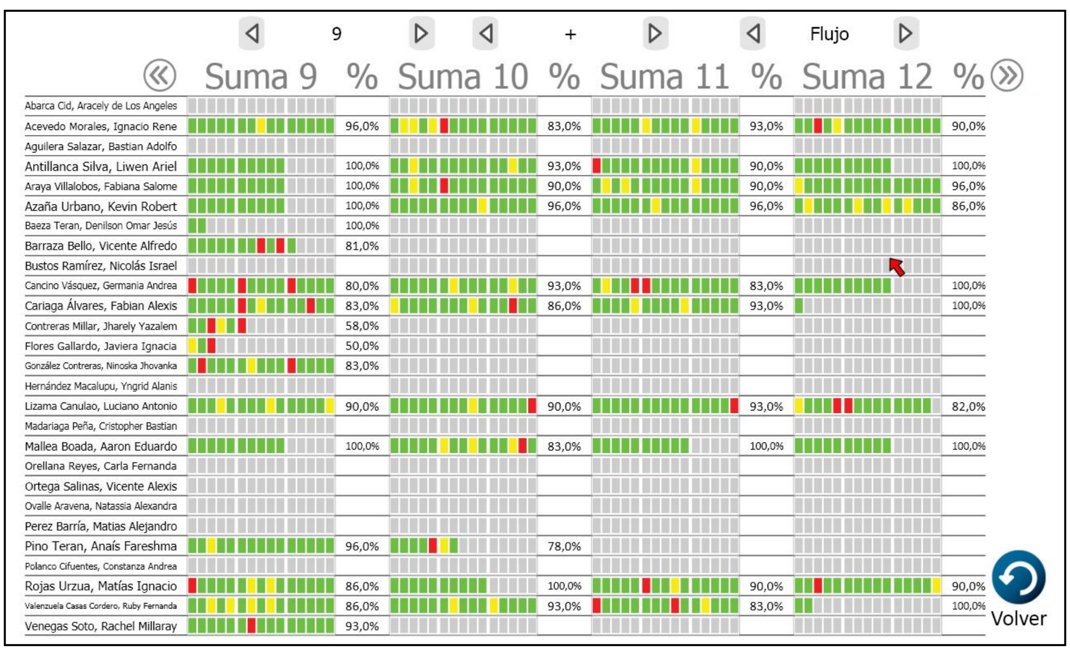

Figure 2-33: Summary window screenshot

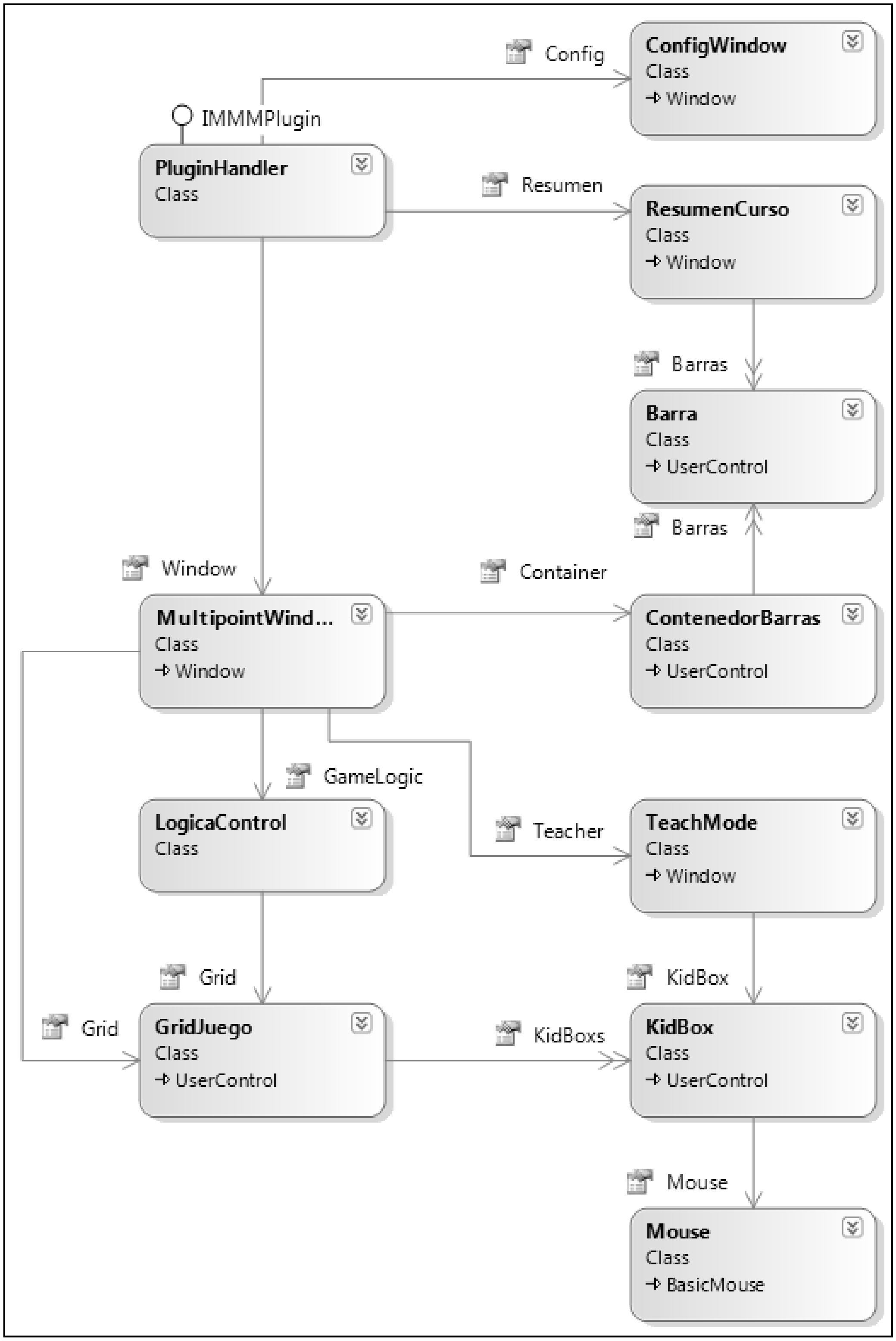

Figure 2-34: SumasPlugin class hierarchy

Table 2-5: SumasPlugin class description

Class	Description
Barra	Represents a single point bar like the ones shown in Figure 2-31. When an exercise is answered correct, it updates itself to show the point with the corresponding color (green, yellow or red).
ConfigWindow	Window that shows the settings before the main activity begins. The options that can be set are if the students start an activity all over again from a defined level of if they continue the last session.
ContenedorBarras	Contains a group of point bars. It also handles the sorting by points and grouping by level.
GridJuego	Main grid of the game. Contains a group of kidbox and set their position depending on how many mice are connected at the beginning of the activity.
Kidbox	Space where a student can work, as shown in Figure 2-24. It has an exercise (red box), the digits (instances of MultiPointCicleChoice from MultiPointControl DLL) to enter the answer (green box) and the identification symbol (blue box).
LogicaControl	Handles all the exercises. This class determines when an answer is enter if it it's correct or not. In case it's correct, this class asks the Reglas class from SumasCommon DLL for the next exercise.
Mouse	Inherited from BasicMouse from MultiPointControl DLL, this class extends the functionality provided by this class in order to incorporate some specific elements from this plug-in.

MultiPointWindow	Main window of the activity. It's launched when the framework calls for the main activity of the plug-in. It also coordinates the interactions between LogicaJuego, the kidboxs and ContenedorBarra.
PluginHandler	This class implements IMMMPlugin interface from MultiPointControl DLL in order to become a plug-in for the framework described in section 2.3.2.2.2. It mediates the interactions between the framework and the application.
ResumenCurso	Shows a summary of the course work after a session is finalized.
TeachMode	When it's called, stop the class and shows a window with the Teachmode, where a teacher can show how to solve exercises from specified levels.

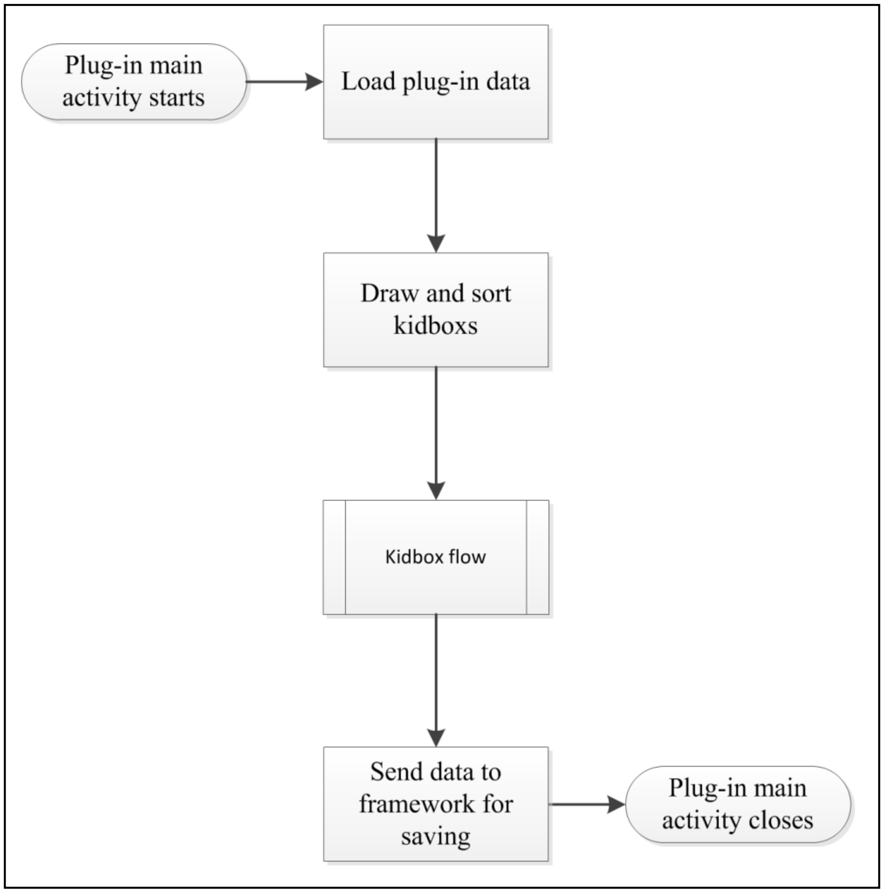

Figure 2-35: SumasPlugin basic flow chart

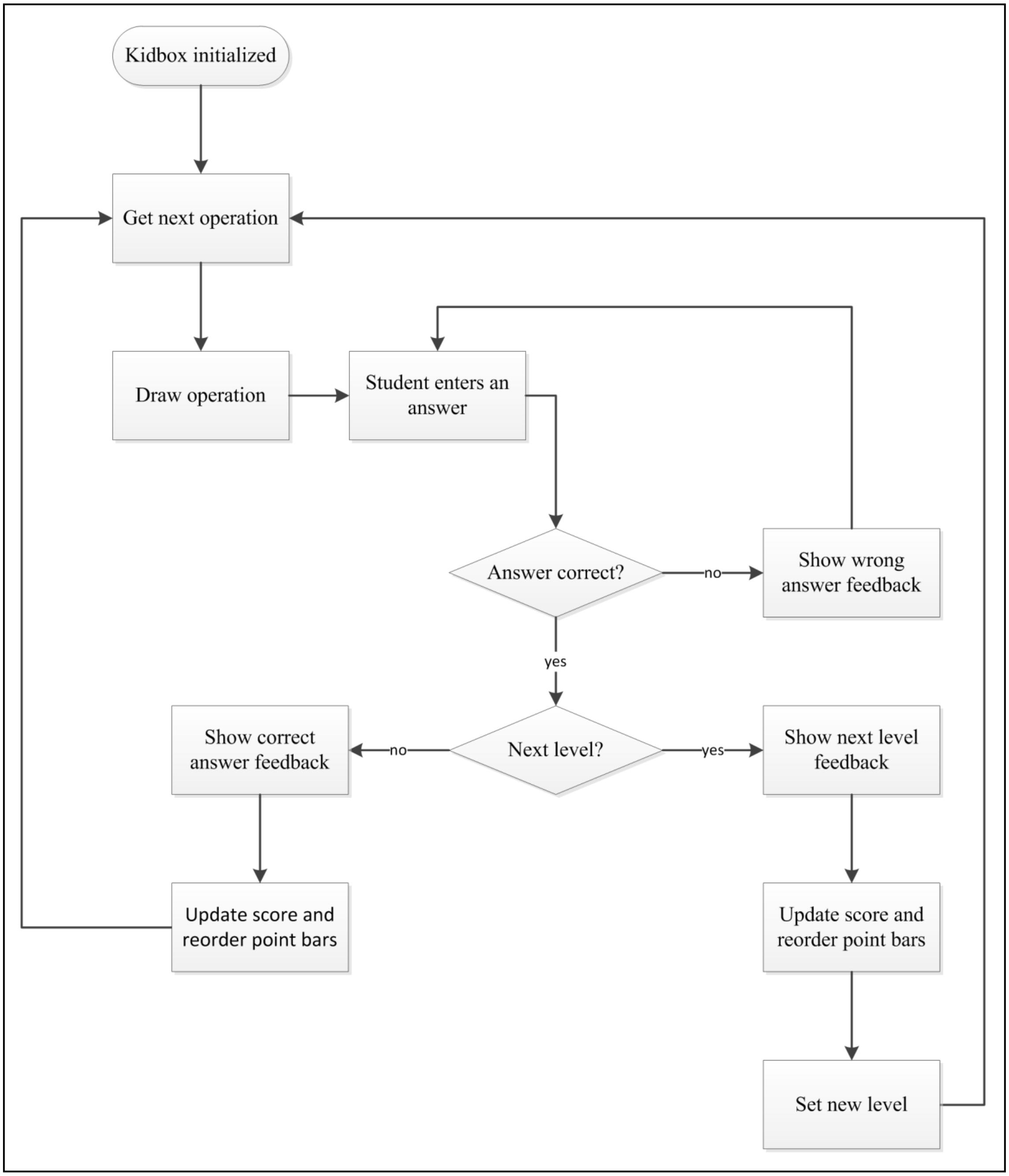

Figure 2-36: Kidbox flow chart

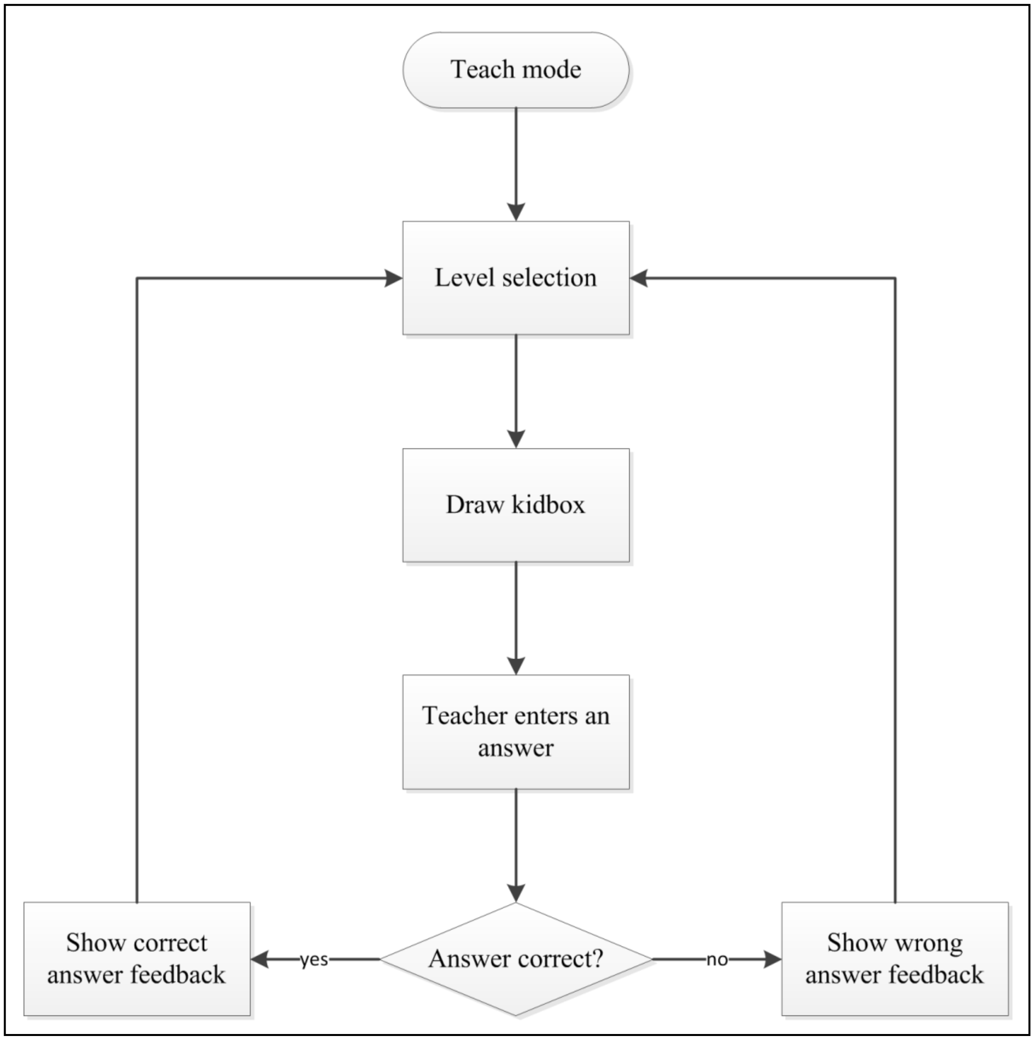

Figure 2-37: Teach mode flow chart

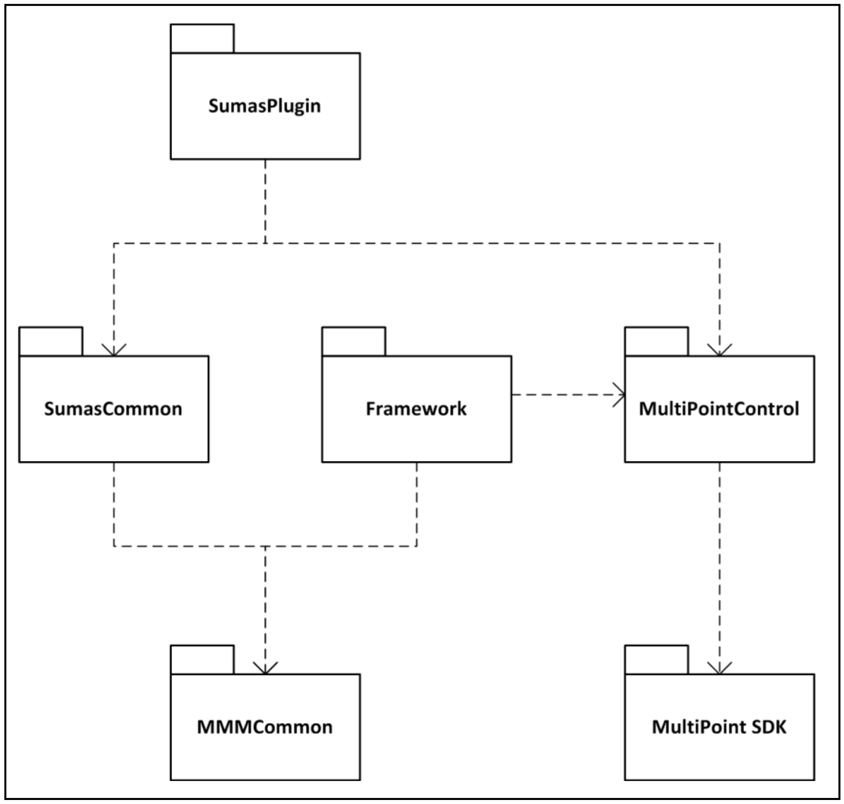

Figure 2-38: SumasPlugin dependencies

2.3.2.3 Sumas Single

This version of the arithmetic software was made to run in a personal computer, using a 1:1 approach, in other words, one computer for each student. This software was developed using the one mouse per child version as a base. It is designed to be used in a netbook or a Classmate PC (CMPC), like the one in Figure 2-39, but it runs on Windows in any PC.

Figure 2-40 shows a screenshot of the software. Just like the one mouse per child version, in the red box is the current exercise space. In the green box is the space where the student has to enter the answer by clicking the arrows of the digits. In the blue box is the symbol that the student has to click when he think he has his answer ready. When he does it, a feedback appears: incorrect feedback (Figure 2-25), correct feedback (Figure 2-26) and next level feedback (Figure 2-27). Finally, in the black box it's the point bar space. It shows the current level of the student and his current score in the activity.

The flow of this software, described in Figure 2-41, is very similar to the SumasPlugin flow, described in Figure 2-35. The main difference it's that in this case the same software handles the course and student selection, meanwhile in SumasPlugin is handled by the framework. The kidbox flow is exactly the same as the one for SumasPlugin shown in Figure 2-36. Other differences are that in this case the Teachmode doesn't exist as well the final summary that is shown at the end of the session. Finally, the main difference between the two systems is that in this one the feedback provided is personal, and in the one mouse per child version is public.

Figure 2-42 shows a hierarchy class diagram and Table 2-6 shows a description of each class. Just like SumasPlugin, Sumas Single also depends on SumasCommon DLL and MMMCommon DLL, however, because of this version doesn't support Multipoint, all references to the Multipoint SDK and MultiPointControl DLL where eliminated.

Figure 2-39: Classmate PC (CMPC)

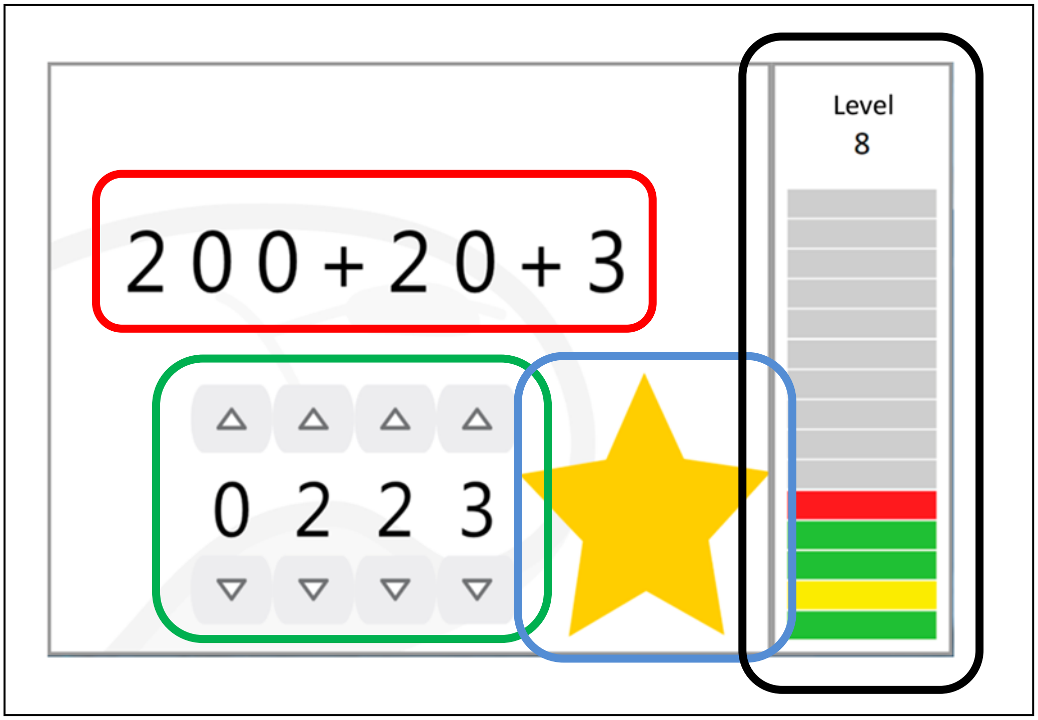

Figure 2-40: Sumas Single screenshot

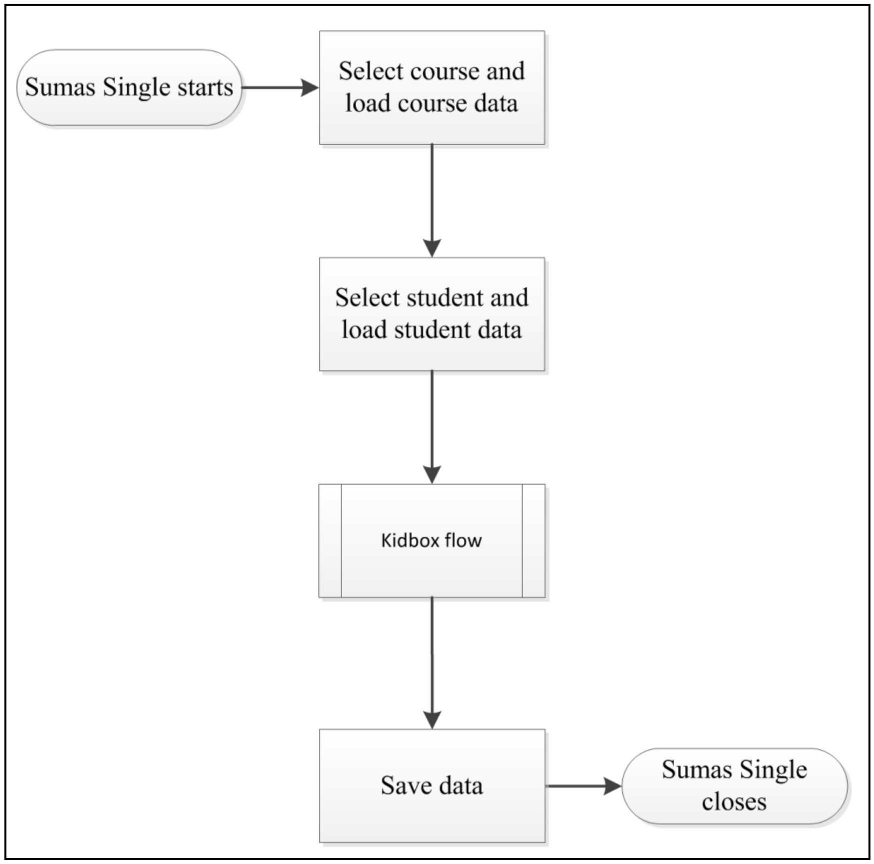

Figure 2-41: Sumas Single flow chart

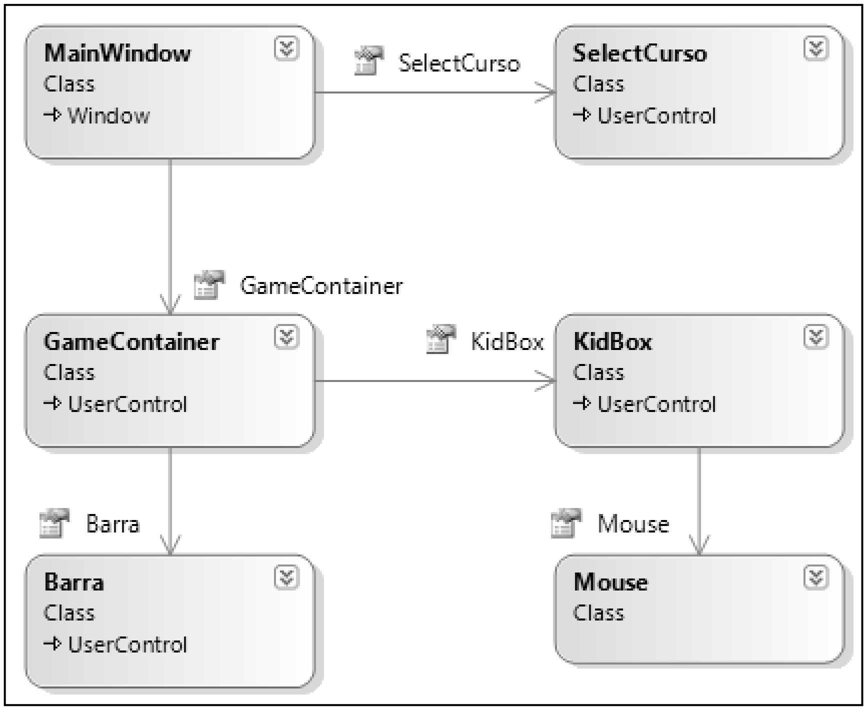

Figure 2-42: Sumas Single class hierarchy

Table 2-6: Sumas Single class description

Class	Description
MainWindow	Main window of the software. It also controls the main flow of it. It has a SelectCurso and a GameContainer control.
SelectCurso	Control that allows the user to select a course and the name of the student that is working. After, it loads the data necessary for the activity.
GameContainer	Control that contains the KidBox and the point bar.
Barra	Point bar that shows the actual score and level. It works just like the one mouse per child version: Green for answers correct at the first attempt, yellow for answers correct at the second attempt and red for answers correct in the third or higher attempt.
KidBox	Draw the current exercise and it also controls the flow of the set of exercises during a session.
Mouse	Legacy of the one mouse per child version. Stores the data of the session to save it later.

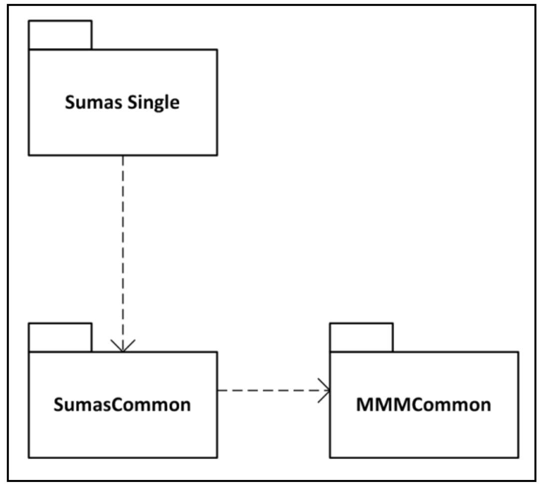

Figure 2-43: Sumas Single dependencies

3. INTERACTIVE LEARNING: A COMPARISON OF INDIVIDUAL AND INTERPERSONAL COMPUTER TECHNOLOGIES

3.1 Abstract

The introduction of technology in the classroom has centered around a variety of different systems. Among the education technology projects that have been adopted by developing country governments, the one which has had the greatest impact is the One Laptop per Child initiative, which aims to provide 1:1 educational computing to students in emerging economies. But a more affordable option has proven to be the interpersonal computer, which can achieve interactive learning with a group of students using just a single computer, a projector and a mouse for each child. The aim of this study is to determine the differences in children's learning outcomes and classroom behavior when they interact with an interpersonal computer as opposed to a personal device. A multi-session experiment conducted at a Chilean primary school to compare the two technologies and conventional pen-and-paper work methods found that even though the children using pen and paper completed more exercises, those working with the technologies actually learned more (though the difference was only statistically significant for those using the interpersonal computer). The lack of a significant difference in learning between the two technologies suggests that the difference observed between them and pen-and-paper methods is the ability of the former to provide instant feedback, whether on an individual or a shared monitor.

3.2 Introduction

The introduction of technology in the classroom has centered around a variety of different systems, including mainframe computer terminals (Feurzeig, Papert, Bloom, Grant, & Solomon, 1970), interactive whiteboards (Higgins, Beauchamp, & Miller, 2007), mobile devices (Zurita & Nussbaum, 2004), interpersonal computers (Stewart, Bederson, & Druid, 1999; Moraveji, Inkpen, Cutrell, & Balakrishnan, 2009) and others. These technologies have enabled interactive learning, which in turn allows students to deepen their participation and engage in more reflective action (Beauchamp & Kennewell, 2010). Interactivity is used in intelligent tutoring systems (Orey & Nelson, 1993) to promote a construction of knowledge in which the learner attempts to build a coherent representation of tutorial content based on previous knowledge (Self, 1998).

Among the education technology projects that have been adopted by developing country governments, the one which has had the greatest impact is the One Laptop per Child initiative, which aims to provide 1:1 educational computing to students in emerging economies (Kraemer, Dedrick, & Sharma, 2009). But even at just US$100 per device, this solution is still much too expensive to be implemented in most developing communities around the world (Trucano, 2010). A more affordable option has proven to be the interpersonal computer (Moraveji, Kim, Ge, Pawar, & Mulcahy, 2008), which can achieve interactive learning with a group of students using just a single computer, a projector and a mouse for each student at a cost of one dollar per child per year (Alcoholado, et al., 2011). This alternative is the motivation for the research question addressed in the present study: Are there differences in children's learning outcomes and classroom behavior when they interact with an interpersonal computer as opposed to a personal device?

The use of technology in mathematics teaching can capture children's attention, motivate them and help them construct mathematics concepts in meaningful ways (Smith, Gentry, & Blake, 2011). This can be reinforced by practice drills, which develop fluency and understanding in mathematics (Bobis, 2007; Wong & Evans, 2007). But technology will not necessarily improve learning quality unless it is integrated into the curricular process (Keengwe, Onchwari, & Wachira, 2008). In (Alcoholado, et al., 2011) we showed how an interpersonal computer can be used to teach arithmetic; here, we attempt to answer our research question by comparing it with a personal computer using the same underlying intelligent tutoring system as its software.

The objective of this paper is thus to study how two different technologies, a personal computer (netbook) and an interpersonal computer, influence the learning achieved through arithmetic drills. We examine not only their impacts on learning but also how they affect student behavior. As will be explained, a multi-session experiment was conducted in which two groups of schoolchildren performed arithmetic exercises on the two technology platforms (one group on each) while a third did the same exercises with pen and paper. The three groups advanced through the same levels of difficulty as the learning process progressed under the supervision and mediation of a teacher, who provided the necessary support when students either requested help or were seen to need it. The system used with both platforms is described in Section 2.3. The design of the experiment is presented in Section 2.4 and the results are set out in Section 2.5. Finally, our conclusions are discussed in Section 2.6.

3.3 Interactive arithmetic

The development of the intelligent tutoring system (ITS) for arithmetic practice drills used on the technology platforms is based on the arithmetic curriculum for the first four school years set by the Chilean Ministry of Education (MINEDUC, 2011). The system is built around 65 rules, of which 18 relate to addition, 18 to subtraction, 13 to multiplication and 16 to division (Alcoholado, et al., 2011). The rules follow a sequence defined by the curriculum framework, and the number of exercises assigned for each rule depends on the student's proficiency. Students advance from one rule to the next once they successfully solve either 10 exercises with no mistakes or at least 8 of the last 15.

Both the netbook and the interpersonal computer use the same intelligent tutoring system and user interface. The interface has four display states as shown in panels A through D of Figure 3-1. The state depicted in 1A is the display of an exercise generated by the intelligent tutoring system depending on the student's progress. The states in 1B and 1C show the system's respective responses to an incorrect and a correct answer. To solve an exercise, the student must construct his/her answer by incrementing or decrementing each digit in the answer row (the three 0's in 1A). This method ensures the answer cannot be just guessed. Once the answer has been entered, the student clicks on the symbol, which in this case is a star. On the interpersonal computer this symbol also serves as the individual student's identifier. The fourth and final display state (1D) appears when a student has answered all the exercises for a rule, thus completing a curricular objective.

On the interpersonal computer, each student has a box containing their current activity and the identifier symbol they have chosen. The various boxes are all displayed simultaneously in a grid as shown in Figure 3-2. A column on the far right of the figure indicates each student's current state by means of a string of squares that represents the 15 exercises for the rule they are working on. The strings are identified by the identifier symbols, which display the corresponding student's name once the symbol is clicked by the teacher. The squares are color-coded to show the outcome of the students' efforts on each exercise: green for a correct answer on the first attempt, yellow for a correct answer on the second attempt and red for a correct answer on the third or further attempt. These indicator strings are grouped in the column by the curricular level of the students' current rule (4 to 9 in Figure 3-2). The teacher thus monitors the students' progress, providing assistance and reinforcement to those that need it.

(a) (b) (c) (d)

Figure 3-1: ITS interface display states

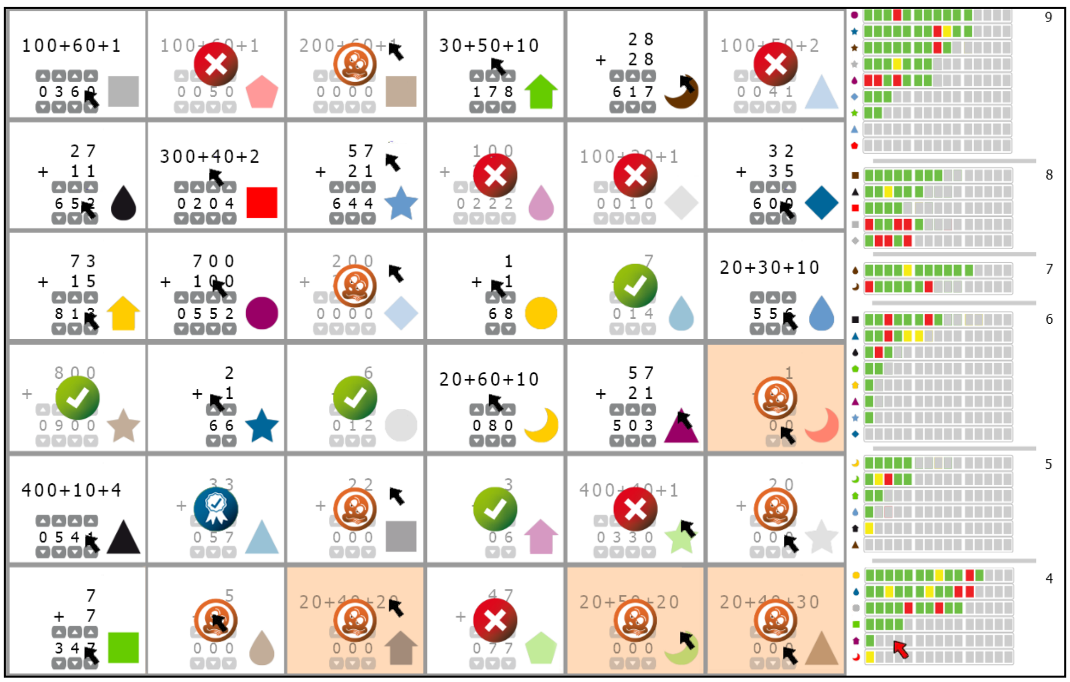

Figure 3-2: Interpersonal computer information display

On the personal computer (netbook) system, instead of showing the whole classroom information, each student's individual screen (Figure 3-3) displays the information in Figure 3-1A plus the same state information displayed on the interpersonal computer, that is, the current curricular level the student is working on and the outcomes of their attempts to solve the 15 exercises.

The students' activities are essentially the same on the two technological platforms since on either one, the children work at their own pace. The only major difference is that those with netbooks begin working on their exercises immediately while those using the interpersonal computer must first complete the identification step in which they recognize their identifier symbols using their respective mice. This initial task took the students about 8 minutes in the early sessions but once they were familiar with it this dropped to about 3 minutes.

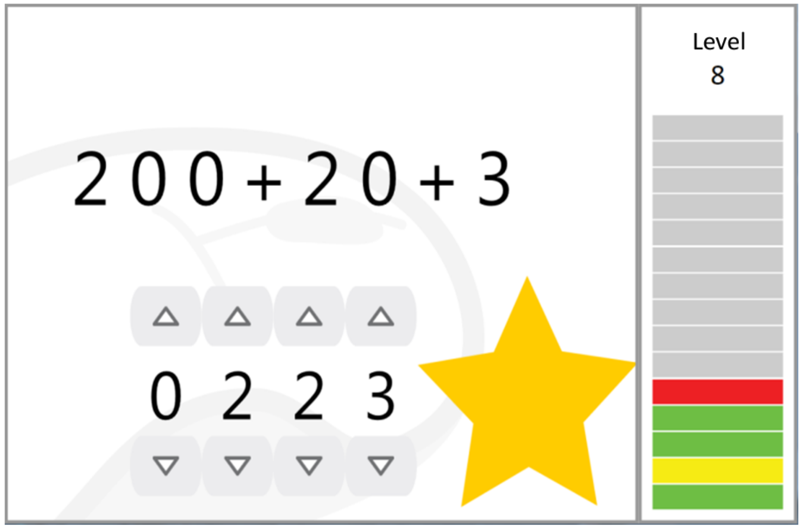

Figure 3-3: Netbook information display

3.4 Experimental design

As noted in the introduction, the design of our experiment is built around three groups of students. The two computer technologies were each assigned to one of the groups while the third group used no technology at all, performing the same exercises used in the intelligent tutoring system but with pen and paper. Two booklets were distributed to the third group students, one a workbook with the exercises and the other containing the answers. The main difference between the third group and the other two was in the immediacy of the feedback. Whereas the groups using the technologies received immediate feedback and could not advance to the following exercise until they had answered the current one correctly, the group working with pen and paper had to complete all the exercises at a given curricular level before finding out which ones were answered correctly, and did not necessarily have to redo the wrong answers since the teacher did not always supervise them directly.

A comparison of the three systems is set out in Table 3-1. Note that for students using pen and paper the number of exercises is fixed at 15 whereas for those using either of the two computer technologies, the number is variable and will depend on the individual student's performance.

Table 3-1: Comparison of systems

	Interpersonal computer	Netbook	Pen and Paper
Type of feedback	Immediate		Upon completion of all 15 exercises in a curricular level
Display of information	Public	Private	Private
Display of score	Public (ranked by progress)	Private	(no display)
Exercises to be completed before advancing to next level	10 exercises, all answered correctly, or 15 or more exercises, of which at least 8 must be answered correctly.		15 exercises

The multi-session experiment was conducted at a government-subsidized private school in Santiago, Chile attended by children from low-income families in which 60% of the parents did not finish grade school. The 81 participants were all in third grade and included 44 boys and 37 girls between the ages of 8 and 10. They were divided randomly into three groups of 27, one group for each system, which were maintained throughout the study. The final analysis was based on the results of 54 students who performed both the pre-test and the post-test, 19 of which were from the interpersonal computer group, 17 from the netbook group and 18 from the pen-and-paper group.

The sessions began in May and ended in November, shortly before the end of the Chilean school year. It was originally intended that a 40-minute activity would be performed each week simultaneously by all three groups. However, there were a number of weeks in which no sessions could be conducted due to extra-curricular events and legal holidays. In the end, the pen-and-paper and netbook groups each held a total of 14 sessions while the interpersonal computer group held only 12, the other 2 being cancelled due to technical problems with the software. Two additional sessions were needed with all three groups to administer the pre-test and post-test. These activities were above the regular 8 hours per week of 40-minute mathematics classes.

Each group held all of its sessions in the same room and was assigned a mathematics teacher from the school. To minimize the possible effects of differences in the quality of support given to the students, the teachers were rotated twice a year so that every group had the same three teachers for equal periods during the experiment. The two groups using computers also had support staff to supervise the activities and handle any technology-related problems.

The pre-test and post-test consisted of a pen-and-paper exam with 45 questions based on the various curricular levels worked on during the sessions. The questions were chosen by teachers at the school from among a set generated at random according to the 65 ITS rules discussed in Section 2.3 and were based on the knowledge they expected students to have by the end of third grade. Before applying this instrument it was validated on 76 fourth grade pupils at the same school, the results displaying a Cronbach's alpha of 0.8901. Thirty minutes were allowed for answering the exam questions, the students' scores being simply the number of correct answers.

To analyze the children's behavior, qualitative data were recorded during the entire period of the experiment. To standardize the results, a single observer was rotated from week to week between the three groups. Since 11 observations (or more precisely, sets of observations) were taken, the first and last were eliminated, thus leaving an equal number (3) for each group. These data were categorized according to the following attributes:

1) Fatigue

(1) Boredom: comments suggesting the children did not want to continue the activity or were not enjoying it; (2) Tiredness: physical or verbal indications; (3) Interruptions: any act by a student whose objective or actual result was to interrupt the flow of the activity.

2) Interaction

(1) Collaboration: helpful comments or actual assistance between two pupils; (2) Competition: comments between two or more children.

3) System

Teacher's explanations of the use of the system to a particular student.

4) Pedagogy

Teacher's explanations of a particular exercise to a particular student.

These behavioral data were entered in a Tablet PC which automatically registered the time the observation was made so that the evolution in these attribute variables could be tracked as the experiment progressed.

3.5 Results

3.5.1 Quantitative results

The progress made by the three groups, as demonstrated by the difference between their pre-test and post-test results, was in all cases significant and with a large effect size. As can be seen in Table 3-2, the students using the two computer technologies displayed similar advances even though the interpersonal computer (IPC) group had two fewer sessions. Indeed, the difference between the netbook and the IPC group results was shown by a *t* test (normality assumption proven with Kolmogorov-Smirnof's test) not to be significant ($p < .3685$). The difference between the netbook and pen-and-paper (P&P) groups was also not significant (*p*

$< .1020$), but the IPC's superior performance compared to the P&P group was significant ($p < .0260$).

Table 3-2: Pre-tests and post-tests results

	No. of students	Pre-test		Post-test		Δ%	Significance	Cohen's d
		$\bar{X}$	S	$\bar{X}$	s			
IPC	19	18.95	8.09	29.26	9.94	54.44%	*p<.0001*	1.17
Netbook	17	19.06	6.56	28.71	6.88	50.62%	*p<.0001*	1.48
P&P	18	25.50	5.97	32.44	6.59	27.23%	*p<.0001*	1.14

3.5.1.1 Length of sessions

The students' actual working time on the interpersonal computer increased from 11 minutes in the first 40- minute session to almost 38 by the last. Students assigned netbooks showed little change in their working time, which was 31 minutes in the first session and averaged 33 over the entire period of the experimental.

This working time difference between the two group's trends was due mainly to the different adaptation times for the two technologies. The interpersonal computer in particular included an identification phase in which the students had to recognize their names with their mice for data traceability (Alcoholado, et al., 2011). Also, much of the first session with the interpersonal computer was devoted to explaining how it worked (29 minutes) while for the netbook only 9 minutes of the first session were needed to explain its functioning.

As for the pen-and-paper group, the students began working on the exercises almost immediately, the only delay being the time taken to hand out the paper notebooks.

3.5.1.2 Exercises completed

The students completed an average of 26.71 exercises per session on the interpersonal computer, 32.8 on the netbook and 52.5 using pen and paper. As shown in Figure 3-4, the interpersonal computer group steadily increased its total over the first three sessions as the children mastered the software, a process the netbook group evidently did not require. From that point forward, however, all three groups exhibited a decline as the exercises' level of difficulty increased. In the final sessions, exercises completed by the interpersonal group overtook that of the netbook group.

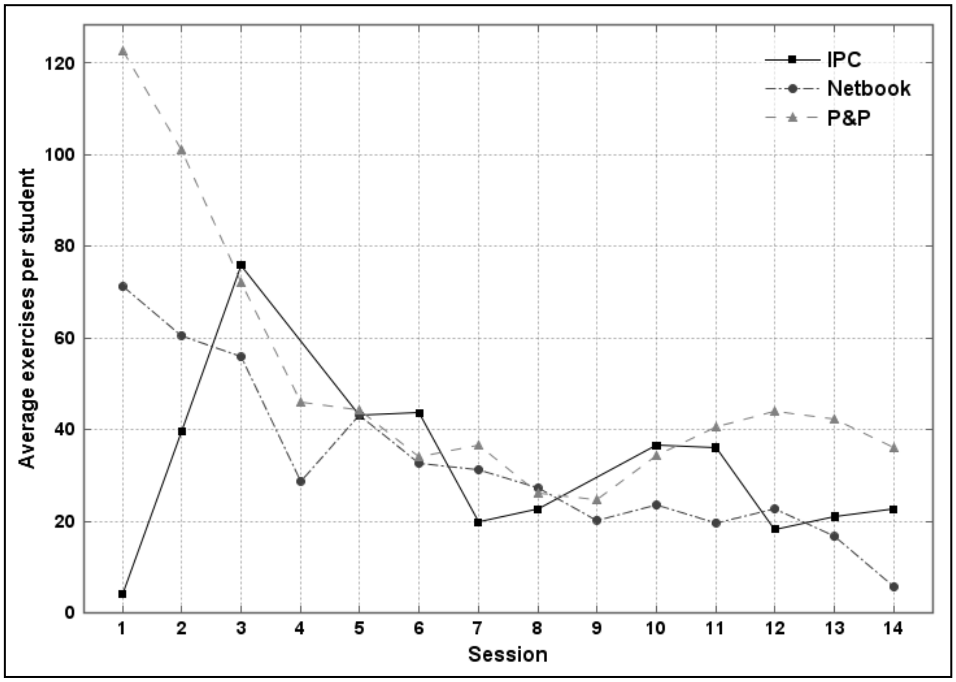

Figure 3-4: Number of completed exercises per student by session and technology group

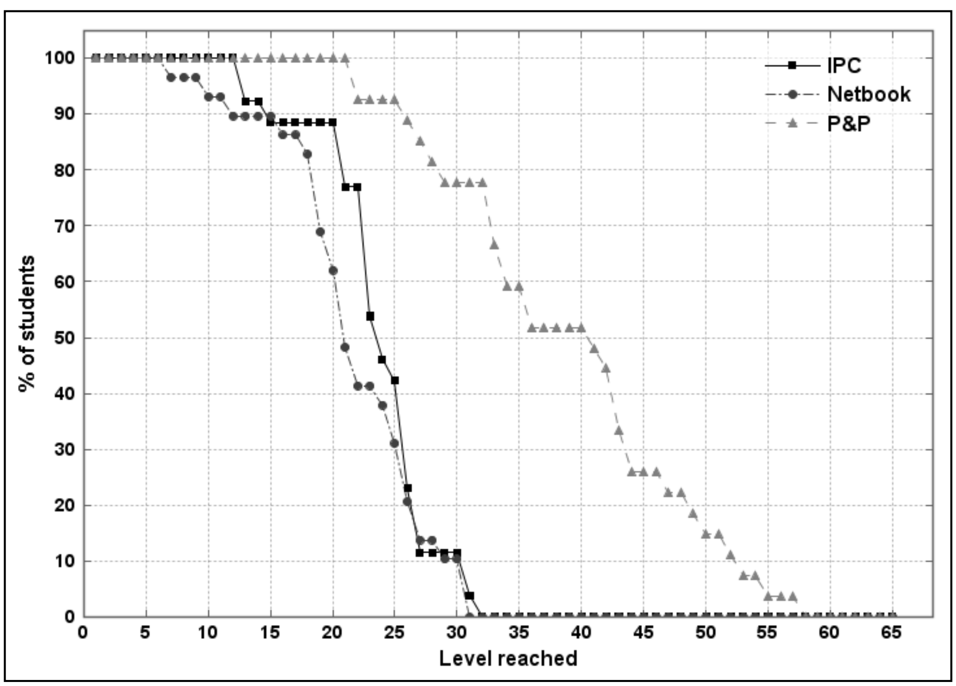

Figure 3-5: Level reached by end of experiment

3.5.2 Qualitative results

The qualitative behavioral observations for the three groups are summarized in Table 3-3. They show that the two technology groups generated a more collaborative environment than did the pen-and-paper group even though such was not the intention of the design. As for competitive behavior, even though they could see each other's ranking at every moment the children working on the interpersonal computer exhibited no real difference in this attribute from the other two groups. According to the teachers, the children tended to be very focused on their own individual boxes, leaving only the teacher to keep an eye on their ranking from moment to moment.

Also revealed by Table 3-3 is that the netbook group had more difficulties with their system than did the interpersonal computer group. Whereas in the netbook case the problems generally involved charging batteries and dealing with system popups, the problems arising with the interpersonal computer system had mainly to do with setting up the equipment at the beginning and end of each session.

Signs of tiredness were relatively few, the interpersonal computer group accounting for more than half of the cases observed. This was because children in the group with poor vision struggled to read the display projected on the classroom wall. The problem was minimized through appropriate changes in the children's seating arrangements.

The collaboration observed in each group is illustrated in Figure 3-6 for each group's three observed sessions. The netbook and pen-and-paper groups both showed a downward trend over the course of the sessions, the latter group by the last session showing no signs of collaborative behavior at all. By contrast, requests for help with the exercises increased steadily in the interpersonal group between the first and the last sessions, as demonstrated in Figure 3-7.

The correlations between the boredom, collaboration and work interruption observations by group are summarized in Table 3-4. As can be appreciated, the behavior of the pen-and-paper group as regards interruptions was completely the opposite of that exhibited by the two technology groups. In all three groups, the correlation between collaboration and interruptions and between boredom and collaboration was high, suggesting that when the students begin to tire of their work they turn to their classmates, whether to collaborate with them or just to interrupt them.

Table 3-3: Number of observations by attribute

		Netbook	IPC	P&P
Fatigue	**Student shows signs of boredom**	34	54	54
	Student shows signs of tiredness	4	13	6
	Student interrupts flow of activity	85	91	64
Interaction	**Student shows signs of collaborative behavior**	63	48	11
	Student shows signs of competitive behavior	6	8	10
System	**Student requests help with the system**	17	3	-
Pedagogy	**Student requests help with the exercises**	55	47	27

Table 3-4: Correlations among boredom, collaboration and interruptions attributes

	Boredom vs. Collaboration	Boredom vs. Interruptions	Collaboration vs. Interruptions
Netbook	0.98	0.76	0.87
IPC	0.80	0.67	0.98
P&P	1.00	-0,93	-1.00

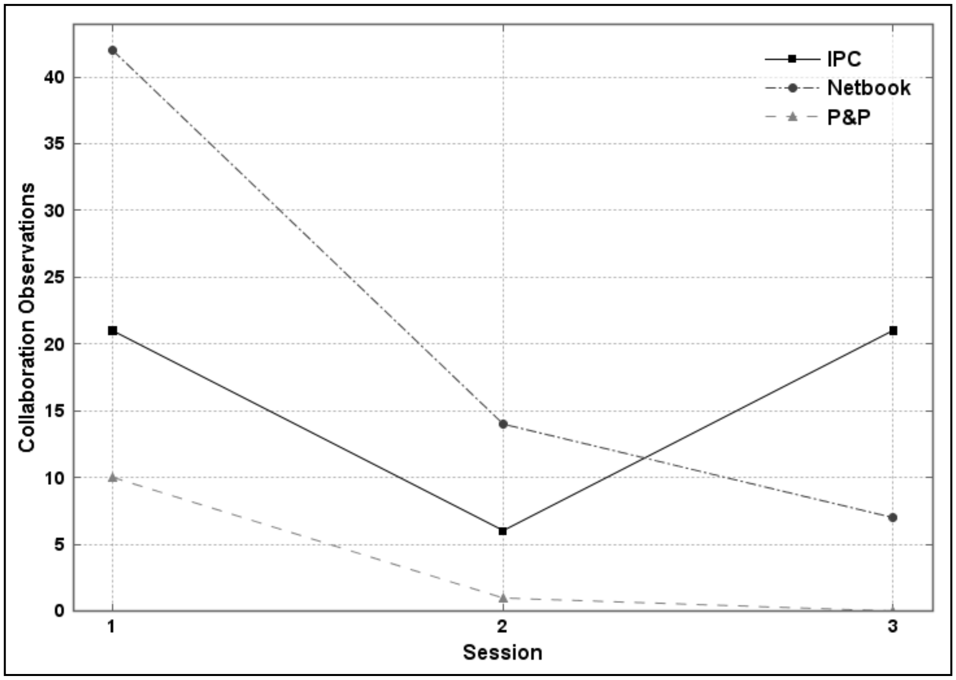

Figure 3-6: Observations of collaboration by observed session

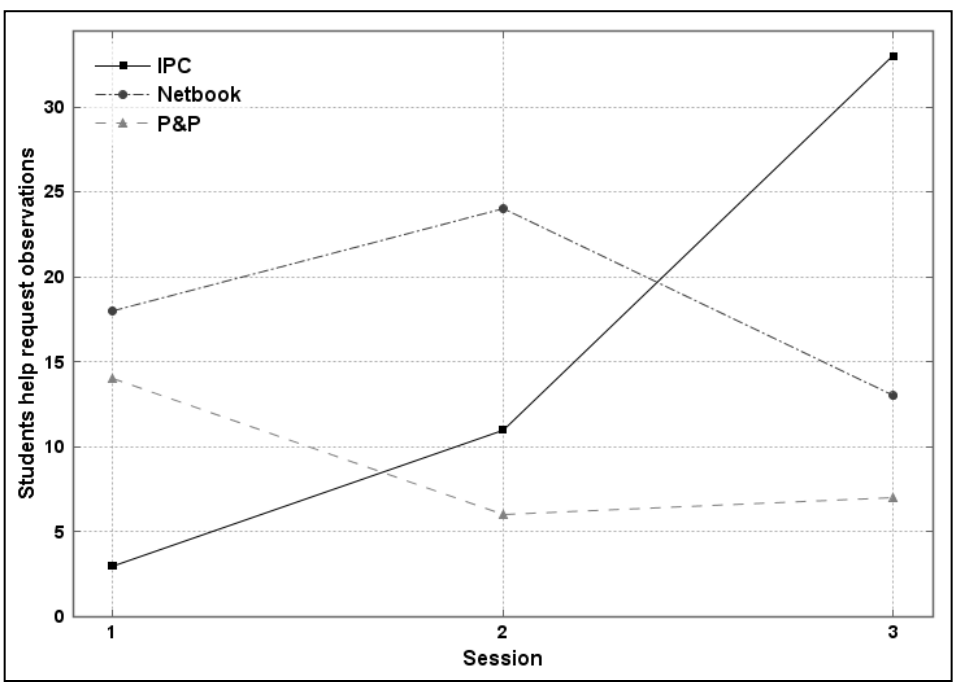

Figure 3-7: Observations of exercise help request by observed session

To further investigate the phenomenon of boredom, a single session was divided into three equal time intervals and the observations recorded and graphed. The results, shown in Figure 3-8, indicate that the pen-and-paper group experienced greater boredom than the other two groups. This was likely because of a positive predisposition towards using the technologies, a novelty for many of the children. As the session progressed, however, the boredom level among the technology groups generally grew while that of the pen-and-paper group stayed relatively constant.

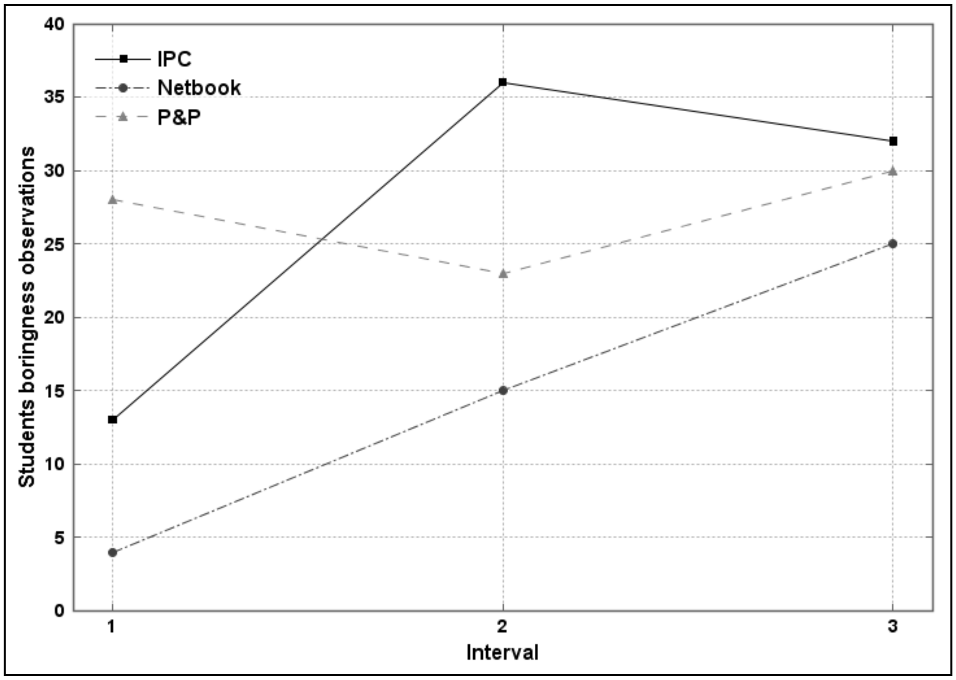

Figure 3-8: Boredom by time interval over a single session

4. CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

Two technology-based methods for conducting arithmetic drills were compared to the traditional pen-and-paper approach as regards both learning impacts and differences in student behavior. Students participating in a multi-session experiment in which they worked through arithmetic exercises at different curricular levels were divided into three groups, one using personal computers (netbooks), another an interpersonal computer and the third pen-and-paper.

The results of the experiment showed that although the pen-and-paper group completed 33% more exercises per session and reached a curricular level on average 60% higher than the two technology groups, their learning progress was in fact smaller (although the difference was only significant statistically with the interpersonal computer group). The lack of a significant difference between the interpersonal computer and netbook groups suggests that the difference between the two technology groups on the one hand and the pen-and-paper group on the other lies principally in the ability of the technologies to give instant individualized feedback, whether on an individual or a shared monitor. We thus conclude that instant feedback can be provided "publicly" (on an interpersonal computer) and obtain similar results to "private" provision (netbook).

If the learning impacts of the two technologies were similar, their acquisition costs obviously are not. The clear advantage of the interpersonal computer scheme on this fundamental criterion is illustrated by the fact that equipping a classroom with

50 children would involve an expenditure about 90% less than that for netbooks. The use of the interpersonal computers is thus a highly attractive alternative for personalized learning in developing economies where 1:1 equipment or computer labs are not generally accessible.

4.2 Future work

The authors plan to extend the research reported here by studying how personal and interpersonal computers differ in their learning and behavioral impacts for school subjects other than arithmetic, such as native language skills and foreign language acquisition, complementing the mice with earphones and keyboards. Another area we hope to explore is how the type of technology affects the results for whole-class collaborative work (Szewkis, Nussbaum, Rosen, Caballero, Tagle, & Alcoholado, 2011). A third important area of investigation is the optimal integration of digital and non-digital resources in the classroom (Nussbaum, Dillenbourg, Fischer, Looi, & Roschelle, 2011).

BIBLIOGRAPHY

Alcoholado, C., Nussbaum, M., Tagle, A., Gómez, F., Denardin, F., Susaeta, H., et al. (2011). One Mouse per Child: interpersonal computer for individual arithmetic practice. *Journal of Computer Assisted Learning* .

Bangert-Drowns, R., Kulik, C.-L., Kulik, J., & Morgan, M. T. (1991). The instructional effect of feedback in test-like events. *Review of Educational Research* , 213-238.

Beauchamp, G., & Kennewell, S. (2010). Interactivity in the classroom and its impact on learning. *Computers and Education* , 759-766.

Bier, E., & Freeman, S. (1991). MMM: A User Interface Architecture for Shared Editors on a Single Screen. *Proceedings of User Interface and Software* (pp. 79-86). ACM Press.

Bobis, J. (2007). From Here to There: The Path to Computational Fluency with Multi-Digit Multiplication. *Australian Primary Mathematics Classroom , 12* (4), 22-27.

Butler, D., & Winne, P. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research* , 245-281.

Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1970). *Programming Languages as a Conceptual Framework for Teaching Mathematics.* Cambridge: Beranek and Newman, Inc.

Foreman, J. (2003, July/August). Next generation: Educational technology versus the lecture. *Educause Review* , 12-22.

Higgins, S., Beauchamp, G., & Miller, D. (2007). Reviewing the literature on interactive whiteboards. *Learning Media and Technology , 32* (3), 213-225.

Inkpen, K., Booth, K. S., Gribble, S. D., & Klawe, M. (1995). Give and Take: Children Collaborating on One Computer. *Proceedings of CHI '95* (pp. 255-259). Conference Companion.

Johnson, R. E. (1997, October). Frameworks = (components + patterns). *Communications of the ACM* , 40-42.

Keengwe, J., Onchwari, G., & Wachira, P. (2008). The use of computer tools to support meaningful learning. *AACE Journal , 1* (16), 77-92.

Kluger, A., & deNisi, A. (1996). The effects of feedback interventions on performance: A historical. *Psychological Bulletin* , 254-284.

Kraemer, K. L., Dedrick, J., & Sharma, P. (2009). One Laptop Per Child: vision vs. reality. *Communications of the ACM 8* , 66-76.

Microsoft Developer Network. (2008, November 6). *The birth of Multipoint*. Retrieved November 2, 2011, from http://blogs.msdn.com/b/multipoint/archive/2008/11/06/the-birth-of-multipoint.aspx

Microsoft. (2010). *Microsoft Mouse Mischief*. Retrieved November 20, 2011, from http://www.microsoft.com/multipoint/mouse-mischief/

Microsoft. (2011). *Windows MultiPoint Mouse SDK*. Retrieved November 19, 2011, from http://www.microsoft.com/multipoint/mouse-sdk/

MINEDUC. (2011). *Programa de Estudio, Primer año Básico, Educacion Matemática.* Retrieved September 27, 2011, from http://curriculumenlinea.mineduc.cl/descargar.php?id_doc=201102080047580

Moraveji, N., Inkpen, K., Cutrell, E., & Balakrishnan, R. (2009). A Mischief of Mice: Examining Children's Performance in Single Display Groupware Systems with 1 to 32 Mice. *Proceedings of the 27th international conference on Human factors in computing systems* (pp. 2157-2166). Chicago: ACM, New York.

Moraveji, N., Kim, T., Ge, J., Pawar, U., & Mulcahy, K. (2008). Mischief: supporting remote teaching in developing regions. *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems* (pp. 353-362). Chicago: ACM.

Nussbaum, M., Dillenbourg, P., Fischer, F., Looi, C., & Roschelle, J. (2011). How to Integrate CSCL in Classroom Life. *Orchestration Workshop at CSCL*, (p. 1199). Hong Kong.

OLPC. (2011). *One Laptop per Child*. Retrieved November 2011, from http://laptop.org

Oppenheimer, T. (2004). *The Flickering Mind: Saving Education from the False Promise of Technology.* Random House.

Orey, M., & Nelson, W. (1993). Development Principles for Intelligent Tutoring Systems: Integrating Cognitive Theory into the Development of Computer Based Instruction. *Educational Technology Research and Development , 41* (1), 59-72.

Pawar, U., Pal, J., & Toyama, K. (2006). Multiple mice for computers in education in developing countries. *International Conference on Information and Communication Technologies and Development, ICTD*, (pp. 64-71).

Self, J. (1998). The defining characteristics of intelligent tutoring systems. *International Journal of Artificial Intelligence in Education , 10*, 350-364.

Smith, A., Gentry, A., & Blake, S. (2011). Mathematics Learning through. In D. W. Sally Blake, *Technology and Young Children: Bridging the Communication-Generation Gap* (pp. 199-221). IGI Global.

Stanton, D., & Neale, H. (2003). The effects of multiple mice on children's talk and interaction. *Journal of Computer Asisted Learning* , 229-238.

Stewart, J., Bederson, B., & Druid, A. (1999). Single display groupware: a model for co-present collaboration. *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit* (pp. 286-293). New York: ACM.

Szewkis, E., Nussbaum, M. D., Rosen, T., Caballero, D., Tagle, A., & Alcoholado, C. (2011). Collaboration between large groups in classroom. *Computer-Supported Collaborative Learning , 10*, 561-575.

Toyama, K. (2011, January). *There are no technology shortcuts to good education*. Retrieved November 2, 2011, from http://edutechdebate.org/ict-in-schools/there-are-no-technology-shortcuts-to-good-education/

Trucano, M. (2005). *Knowledge Maps: ICTs in Education.* Washington, D.C.: infoDev/WorldBank.

Trucano, M. (2010). *One Mouse Per Child, EduTech: A World Bank Blog on ICT use in Education*. Retrieved September 28, 2011, from http://blogs.worldbank.org/edutech/one-mouse-per-child

Wong, M., & Evans, D. (2007). Improving basic multiplicaton fact recall for primary school students. *Mathematics Education Research Journal , 19* (1), 89-106.

Zurita, G., & Nussbaum, M. (2004). Computer Supported Collaborative Learning Using Wirelessly Interconnected Handheld. *Computers & Education , 3* (42), 289-314.

APPENDIXES

APPENDIX A: LETTER OF ACKNOWLEDGMENT OF SUBMISSION

From: <bjeteditor@wiley.com>
Date: Tue, Dec 20, 2011 at 12:52 PM
Subject: BJET: Manuscript ID BJET-0542-Dec-2011-OMS
To: cfalcoho@puc.cl, mn@ing.puc.cl
20-Dec-2011

Dear Mr. Alcoholado,

Thank you for sending a contribution entitled "Interactive learning: a comparison of individual and interpersonal computer technologies" for possible publication in the British Journal of Educational Technology. Your manuscript ID is BJET-0542-Dec-2011-OMS. Please mention the above manuscript ID in all future correspondence or when calling the office for questions.

Please note that it is a condition for submission that your contribution is original and has neither been published previously nor is currently being considered for publication elsewhere. If this is not the case then please withdraw your submission promptly.

BJET reviewers read papers very carefully and sometimes make quite detailed suggestions. I can't predict how quickly I will be able to write again with their recommendations and comments, but please feel free to contact me again for an update. You will be notified by our web-based system as your submission progresses. You can also view the status of your manuscript at any time by checking your Author Centre after logging in to http://mc.manuscriptcentral.com/bjet .

If your contribution is in the form of colloquium piece then I will read it as quckly as possible and give you a decision.

If the recommendation is in favour of publication, there is a further stage and authors vary in how quickly and how positively they respond to reviewers' comments. We cannot commence copy editing until we have the final manuscript, so if you are likely to be unavailable or at a different address for any significant time in the near future, please let me know. If there are any changes in your street address or e-mail address, please log in to ScholarOne Manuscripts at http://mc.manuscriptcentral.com/bjet and edit your user information as appropriate.

BJET is now published six times each year and articles occasionally have to be carried over at the last minute because of space constraints, so it is normally not possible to say in advance in which issue a specific contribution will appear, even after it has been accepted for publication in a revised form.

Meanwhile, thank you for your patience and for sending your material to BJET.

Please note that this is an automated email. Feel free to contact me if you want any futher information.

Nick Rushby
Editor, British Journal of Educational Technology
209 Junction Road, Burgess Hill,
West Sussex RH15 0NX UK
Tel/fax: 44 1444 243092
Mobile 44 (0)7968 721 958
Email: bjeteditor@wiley.com

British Journal of Educational Technology submitted article

Interactive learning: a comparison of individual and interpersonal computer technologies

Journal:	*British Journal of Educational Technology*
Manuscript ID:	BJET-0542-Dec-2011-OMS
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	20-Dec-2011
Complete List of Authors:	Tagle, Arturo; Pontifical Catholic University of Chile, Computer Science Alcoholado, Cristián; Pontifical Catholic University of Chile, Computer Science nussbaum, miguel; puc, computer science Infante, Cristián; San Luis Beltrán School,
Keywords:	Interpersonal computer, Individual feedback, 1:1 education, 1:n education, Intelligent tutoring systems, Arithmetic teaching, Comparative study, Computer-assisted learning

APPENDIX B: ARITHMETIC PEDAGOGICAL RULES

Table B-1: Pedagogic rules for addition

Level	Description	Examples
1	Additions with 2 addends, without carrying	3+4
2	Additions with 3 or more addends, without carrying	2+3+2
3	Additions with 2 addends, without carrying, up to the tens	20+7
4	Additions with 3 or more addends, with tens in each one, without carrying	30+40+20
5	Additions with 2 addends, each one with two digits, without carrying	25+33
6	Additions with 2 identical addends, one digit each, with or without carrying	4+4 6+6
7	Additions with 2 addends, without carrying	3+4, 30+40 300+400
8	Additions with 3 addends, without carrying	200+50+10
9	Additions with 2 identical addends, one and two digits, with or without carrying in the ones	32+32
10	Additions with 3 identical addends, one and two digits, with or without carrying in the ones	450+30 354 + 23
11	Additions with 2 addends, and carrying in the ones	14+18 135+325
12	Additions with 2 addends, multiples of 10, and carrying in the tens	80+30 140+27
13	Additions with 2 addends and carrying in the tens and ones	38+73 156+266
14	Additions with 2 addends, without carrying	3.200+54

		3.271+2716
15	Additions with 2 addends, carrying only once, in one position (tens or ones)	28.146+37 26.734 + 139
16	Additions with 2 addends, carrying only once, in one position, except in the tens of thousands	28.146+1.337 37.235 + 51.337
17	Combined addition and subtraction exercises, with parentheses	(36+24)-15 (364+24)-15
18	Combined addition and subtraction exercises, without parentheses with numbers	36+24-15 364+24-1

Table B-2: Pedagogic rules for subtraction

Level	Description	Examples
1	Additions with 2 addends, where an addend is missing, without carrying	6+_=9 63+__=96
2	Simple subtraction, without carrying	6-3 60-30
3	Intermediate subtraction, without carrying	63-20 63-23
4	Successive subtractions with 3 terms, with only one digit	9-2-1
5	Advanced subtractions, without carrying	7-3, 70-30 700-300
6	Subtractions with carrying in the units, and one-digit subtrahend	50-2 150-2
7	Subtractions with carrying in the units, and one-digit results	45-36 345-338
8	Subtractions with carrying in the units, and two-digit results	45-18
9	Subtractions with carrying in the tens	451-61 451-161
10	Subtractions with carrying in the units and the tens, and one-digit subtrahend	500-2 700-9
11	Open numeric subtraction phrases that involve adding or subtracting, without carrying, to be solved	__-5=43 ___-215=143
12	Subtractions with carrying in the units and the tens, and two-digit subtrahend	451-62 374-96
13	Subtractions with carrying in the units in the tens, and three-digit subtrahend	451-162 374-196

14	Subtractions with 5-digit minuend, without carrying	13.427-426 13.437-13.426
15	Subtractions with carrying in only one position	28.146-147 24.257-9.023
16	Subtractions that require carrying twice, in any position	28.146-17.247 2.678-1.849
17	Open numeric subtraction and addition phrases that involve adding or subtracting, to be solved. Operations may or may not require carrying	_____-145=1.893 5.806-____=522
18	Combined addition and subtraction exercises, without parentheses with numbers	(36+24)-15 364+24-15

Table B-3: Pedagogic rules for multiplication

Level	Description	Examples
1	Successive addition with up to 5 identical addends lower than 6	3+3+3 4+4+4+4
2	Basic multiplication combinations of 2,4, and 8, with a product equal to or smaller than 80	2*2 4*5
3	Basic multiplication combinations of 5 and 10, with products equal to or smaller than 100	5*7 10*4
4	Basic multiplication combinations of 3, 6, and 9, with products equal to or smaller than 90	3*6 6*8
5	Basic multiplication combinations of 1 through 10, with products equal to or smaller than 100	7*9 3*4
6	Multiplication of 2 or 3 factors smaller than 10	3*2*5
7	Multiplication of 2 factors: one greater than 10 and smaller than 100, and the other a one-digit number	34*5 40*8
8	Multiplication of 2 factors: one smaller than 100, and the other is 10, 100, or 1,000	85*10 85*1.000
9	Open numeric phrases where one factor is missing, and the product is smaller than 100	8*_=72 _*7=56
10	Multiplication of 2 factors: one is a 3 or 4-digit number, and the other is 10, 100, or 1,000	6.296*100 271*1.000
11	Combined multiplication and addition exercises, first without parentheses, and then with, so as to verify the distributive property	First: 80*3+80*5 Then: 80*(3+5)
12	Combined exercises with parenthesis priority that include: addition, subtraction and multiplication	5*(17-3) (23*9)-(14*7)
13	Combined exercises (except division) with operation priority and no parentheses	13*8-27 23+70*9

Table B-4: Pedagogic rules for division

Level	Description	Examples
1	Open numeric multiplication phrases with one-digit factors, or a factor that is a multiple of 10	5*__=56 8*__=80
2	Exact divisions, where: (1) Dividend smaller than 100 (2) Divisor smaller than 10 (3) The dividend is a multiple of the divisor	56:8 36:6
3	Exact divisions, where: (1) Dividend smaller than 1.000 (2) Divisor smaller than 10 (3) The dividend is a multiple of the divisor	390:6 565:5
4	Open numeric division phrases where: (1) Divisions are exact (2) If the divisor is missing, the dividend is a two-digit number and the quotient has one digit (3) If the dividend is missing, the divisor and the quotient have one digit	21:_=3 __:5=3
5	Exact division families, with the same quotient, where: (1) Dividend smaller than 1.000 (2) Divisor is smaller than 100 and a multiple of 10 (3) The dividend is a multiple of the divisor	42:6 63:7
6	Exact divisions where: (1) Dividend smaller than 1.000 and a multiple of 10 (2) Divisor is a multiple of 10 (3) The dividend is a multiple of the divisor	360:60 180:3
7	Exact divisions where: (1) Dividend smaller than 10.000 (2) One-digit divisor (3) The dividend is a multiple of the divisor	1060:10 2765:5
8	Divisions with a remainder, where: (1) Dividend smaller than 1.000 (2) Divisor is a one-digit number	67:2 821:4
9	Exact divisions, and divisions with a remainder	

	where: (1) Dividend smaller than 10.000 (2) Divisor is a one-digit number	434:9 5061:2
10	Open numeric phrases of division with a remainder (R) where: (1) If the divisor is missing, the dividend has 3 digits, and the quotient has 1 (2) If the dividend is missing, the divisor has 1 digit and the quotient has 2 and the remainder is specified	547:__=9 R=7 ___:7=85 R=2
11	Open numeric phrases with a missing term, of multiplication and exact division, where: (1) In multiplications, the product is smaller than 10.000 and one factor is a one-digit number (2) In divisions, when the dividend is missing, that the divisor has one digit and the quotient is less than 10.000, be of one digit (3) In divisions, when the divisor is missing, the dividend is smaller than 10.000 and the quotient is a one-digit number	8*____=8.000 ___*3=1.254
12	Combined exercises with multiplications and divisions that come from the basic combinations of the Pythagorean table, with parentheses priority and up to 2 independent parentheses	5*(64:8) (7:7)*(24:6)
13	Combined exercises with multiplications and divisions that come from basic combinations of the Pythagorean table, with one parenthesis inside the other	100:[(5*4):2]
14	Combined multiplication and division exercises, without parentheses: (1) With multiplications and divisions that come from basic combinations of the Pythagorean table (2) Up to 3 operations	54:9*54:6 2*50:10*10

15	Combined exercises with parentheses priority: (1) Addition, subtraction, multiplication and division (2) All operations up to 2 digits (3) Exact divisions (4) Up to 2 independent parentheses	(135:3):(4+5) (3+6)*(7-4)
16	Combined exercises with operation priority: (1) Addition, subtraction, multiplication and division (2) All operations up to 2 digits (3) Exact divisions (4) Up to 4 operations	80+8-8*10 120:5-3*7