A Multi-user Application using Microsoft MultiPoint and Adobe Flash

Giancarlo Nascimento
Universidade Federal Fluminense

Bruno Moreira
Universidade Federal Fluminense

André Brandão
Universidade Federal Fluminense

Christian Ruff
Universidade Federal Fluminense

Esteban Clua
Universidade Federal Fluminense

Figure 1: *Multiple mouses in one single computer*

Abstract

The Microsoft MultiPoint technology is commonly used to develop software with focus on e-learning since its main feature is the possibility to use several mouse devices on the same screen, allowing many students to interact with the same computer at the same time. This work aims to use this interesting technology in a different way. The idea is to create a multi-user game that will help on the research of the behavior of candidates in a group dynamics, providing an innovative and useful tool for psychologists and human resources professionals to evaluate the candidates.

Keywords:: Group Dynamics, Microsoft MultiPoint, Human Behavior

Author's Contact:

{giancarlotaveira,brucostam,frantzruff,brandaoihc}@gmail.com
esteban@ic.uff.br

1 Introduction

Microsoft researchers created a technology known as MultiPoint to help students from countries in development to take the most out of computers, allowing them to work together using several mouses, instead of just one mouse as an input device, in a single computer. MultiPoint also allows equal opportunity of engagement for multiple children sitting on a single computer [Gupta et al. 2010].

The brothers Jimmy, Mark e Luke Dickinson, which live in the city of Tigard in the state of Oregon in the United States of America, are big fans of the MultiPoint. That makes perfect sense since they grew up in a family of thirteen brothers [TechFlash 2009]. The brothers Dickinson used the MultiPoint as the technology to develop a collection of educational games [MyMouseGames 2009] for the technology competition between students promoted by Microsoft: the Imagine Cup. Their project won the highest prize in the Imagine Cup of 2009, held in the U.S., in the Software Design category.

The theme of that year's Imagine Cup was to use technology in order to solve some of the world's hardest problems, as described in United Nations' document that mention the Millennium Development Goals [UnitedNations 2009].

These goals include Global Education, which is where the MultiPoint fits. This technology was born in the Microsoft's research laboratory in India, when researchers were looking for ways of using the computer more effectively in schools that have a limited number of machines but with a large number of students.

Observing the work done by the Dickinson brothers, one can see the potential within the applications developed with the Microsoft MultiPoint Software Development Kit (SDK). The motivation of this work is to develop an application using this innovative technology but also little known.

The following paper aims to demonstrate development details of an application for the investigative use of the behavior of candidates in a special type group dynamics directly connected to the responsibilities of the selected job, which will be called Ecological Group Dynamics.

This paper presents the existing motivations for the development of this work and its ultimate goal. We also present several themes, such as the reason why the MultiPoint technology was developed, how it is used and what is its purpose. It will be explained what is the technology, followed by a brief history about its creation. Besides that, it will also be introduced some libraries and methods used to make the integration between the MultiPoint technology and Flash, which is a development tool that provides a more friendly environment with good graphics interface and is also multiplatform.

This work also presents what was done in terms of studies about the technology and about the psychology area, requirements gathering, code implementation and project organization. Furthermore, we show a preparation of what is considered the ideal environment for using the software with the presented purposes.

Later, we list the difficulties encountered during the implementation phase of the software. Next, we make a critical analysis of the tool and the technologies involved, listing their advantages and disadvantages, and then we present suggestions for future work as well as improvements that could be made.

One of the objectives of this project is to develop a software for the usage in group dynamics that makes possible the appliance of the technique of ecological group dynamics. The ecological group dynamics is that one which places the candidate very close to the scenario he will find in the company's work environment. It allows the psychologists to make a more unique evaluation on the candidate's profile, once the daily tasks in his work environment are being taken into account.

The software aims to expose the candidate to more complex scenarios, elaborated and realistic than those possible with only pen

and paper, such as the use of psychological tests, which have been applied in organizational settings since the mid-70 [Alencar and Gomes 2005]. Psychological tests, although important, are restricted to a specific psychological or cognitive assessment. In addition, computer use in group dynamics caters for a more comfortable environment for the candidate, since its inhibition decreases, thereby allowing him to act more naturally. This is another factor that contributes to a differentiated assessment of the candidate.

For the development of this application we used the Microsoft MultiPoint technology which provides all the necessary libraries to the interaction of various input devices (mouse) at the same terminal. This technology stimulated the creation of several applications, for the most part, focused on e-Learning. This paper aims to apply this technology in a different way, since it is no longer applied solely in school settings and is now applied within the organizations.

With that in mind, another goal of this project is to expand the applicability of this multi-user technology. Thus, the computer would help in evaluating the human behavior, enabling a collaborative environment which would facilitate further investigation of the behavioral aspects.

Following these two aspects, the main focus of this work is to show that the Microsoft MultiPoint technology can be applied with a different purpose from which it was created, and show that the need to use of the same application of ecological group dynamics is highly relevant.

The next section presents the MultiPoint SDK and discuss some of its main functionalities. The third section discuss on the psychology point of view about the application. Section 4 describes the functionalities requirements. Next, section 5 details the design and implementation phases of the proposed application, which uses the MultiPoint technology. In the section 6 we present the final results and we report the difficulties encountered the development phase. After that, we make a critical analysis on the SDK, based on our experience developing with it. Lastly, we present our conclusion and future work suggestions.

2 Related Work

Several applications have been developed using multiple devices within a collaborative context. The work of [Villanueva et al. 2010] explains the development of a system to improve the collaboration possibilities among participants in face-to-face meetings and working groups. Users can collaborate through the shared zone using their own mobile devices, such as mobile phones, Personal Digital Assistant (PDA) devices, laptops and so on. To use the shared zone, users have their own cursors, so they can identify themselves. In another work, by [Gohlke et al. 2010], the authors have built a set of wireless, mouse-like devices enhanced with rotary controllers aiming at integrating the flexibility and transparency of mouse interaction with the benefits of physical controllers. Their goal is to use these controllers for editing audio and video, but with better flexibility. To emulate multiple cursors on the same screen, the authors used the Microsoft MultiPoint technology. The use of multiple mouses to share a single computer in a collaborative environment has been evaluated by [Pawar et al. 2007].

In [Infante et al. 2009], the authors present an application called Exchange, which is a Collaborative Learning application originally developed for wirelessly interconnected Pocket PCs, devoted to engage students and their teacher in a face to face Computer Supported Collaborative Learning (CSCL) activity, in a Single Display Groupware (SDG) mode. That is, three users interact on the same display using three different mouse devices being mediated by the technological network, while preserving the original collaborative interaction. When children work collaboratively together, in small groups, they show increased participation in group discussions, demonstrate a more sophisticated level of discourse, engage in fewer interruptions when others speak, and provide more intellectually valuable contributions to those discussions [Shachar and Sharan 1994].

In [Gupta et al. 2010], an evaluation of immediate learning and retention among primary school children aged 10-12 across separate instruction mediums: multiple-user-multiple-mouses PC, multiple-user-single-mouse PC (SingleMouse: Multiple children sharing a mouse and playing the game), video (a group of children watching a video on a big screen) and No-Visual Narration (A group of children was narrated the learning material). The aim was to see how the interactive learning games affected a student's learning. For SingleMouse, DISHA itself was used, albeit with only a single connected mouse. For the Video, a film of the storylines in DISHA was created. Since the games simply reinforced the content of the storylines, the video content was equivalent to that of DISHA. For the No-Visual Classroom, a narration of the same content was given. The main conclusion of the experiments was that student retention of concepts is highest in the multiple-user-multiple-mouses scenario, notably with single gender groups of three.

The work of [Pal et al. 2006] uses field observations of primary school children in rural India using computer-aided learning modules. Their findings had a 'position paper' feel in proposing a variety of possible scenarios for future thought. In the short run, that research has already been instructive in the design of a multiple-mouse system for children's classroom games. The authors applied two study cases: Case 1: Observation of four 5th grade users with math module and Case 2: Observation of four 6th grade users with language module. Their study suggests that the lack of equal access to the content during concurrent sessions may mean that children without control of the input devices may gain progressively less from the learning experience. Another conclusion of the authors was that all the children wanted to use the mouse or keyboard, but it was left either up to one, or to some rotational combination of all. Finally, the authors concluded that research on the feasibility of multiple users having their own input devices is very important, and this could either mean the use of multiple mouse environments. It was an initial exploratory paper, based purely on what they found in the field as the challenges being faced by the pioneering teachers working with limited resources and very high expectations.

In [Pawar et al. 2007], a study evaluated single-mouse and multiple-mouses configurations for computer-aided learning in schools where access to computers is limited due to resource constraints. The experiment's results suggests that the value of a PC can be inexpensively multiplied by employing a multi-input shared-use design. In a previous work [Pawar et al. 2006], the same authors investigated: (1) whether children could understand the Multi-mouse paradigm, (2) how children interact with and share multiple mouses, and (3) how engagement with the PC changes with multiple mouses. They found that not only children were able to absorb the Multimouse paradigm with five mouses easily and rapidly, but that excitement and individual engagement with the PC dramatically increased. These results suggested positive value in using Multimouse. The evaluation with users, made in [Pawar et al. 2007] provides statistically significant findings to support the hypotheses of collaboration leading to better learning, in terms of quantifiable and tangible educational value attributable to the use of multiple mouses suggesting that PC value for certain kinds of education can be inexpensively multiplied through shared use.

In a more recent work, [Shekhar et al. 2011] report their experiences from an exercise conducted in India to help twelve 9th and 10th grade teachers design games that promote collaborative learning. Specifically, they describe a strategy for designing collaborative learning games through active teacher participation. They present four key elements that are essential for teachers to successfully design games for learning: (1) the affordances of the technology for which they are designing; (2) a vision of how the games will be incorporated in the teacher's daily lessons; (3) the concept of games and interactivity; and (4) a nuanced understanding of the pedagogical goal.

It becomes clear the relevance of the MultiPoint technology. We are able to see that this technology has been vastly applied within an educational context. Even though the mentioned authors have obtained positive results when using MultiPoint and games as a new approach, this innovative technology is not being explored to its maximum potential. This work aims at using the MultiPoint technology within a different context, other than purely educational.

3 The MultiPoint SDK

The MultiPoint SDK is primarily written to be programmed in Visual C#, an object-oriented programming language developed as part of the .NET platform. For that reason, most applications are written in that language. However, using the Visual Studio tool, which is a software package from Microsoft for software development that has an extension which supports the Windows Presentation Foundation (WPF) it is possible to develop applications with MultiPoint in other languages, like C++ with Visual C++ or Basic, the Visual Basic. Besides the development tool, it is necessary to install the .NET Framework version 3.0 or newer. Microsoft MultiPoint technology works with Microsoft Windows XP and Microsoft Windows Vista.

The application presented in this paper was developed using .NET 3.5 with Visual C# running on a PC with Microsoft Windows XP.

For many schools, providing one computer per student is simply not an affordable option [Microsoft 2009]. As part of Microsoft's commitment to expanding the benefits of technology to underserved communities, Microsoft Research developed the Microsoft MultiPoint platform.

Microsoft MultiPoint enables one computer to serve many students simultaneously. Using multiple computer mouses that drive uniquely-designed on-screen cursors, Microsoft MultiPoint allows up to 30 students to simultaneously use and learn from educational software on one computer. This not only offers a more affordable solution, it also creates an active, collaborative learning experience that engages each and every student [Agarwal 2006].

This technology can be used in a computer lab with a small group working on a student controlled application. It can also be applied in a classroom with a large number of students participating while a teacher controls the application.

MultiPoint places multiple cursors on the PC screen when users plug in multiple mouses – assigning one cursor to each mouse. Normally people using several mouses on a single machine would be struggling for control of a single cursor. MultiPoint is meant to open up the experience to more collaboration and competition. That way, the technology helps shift the student from passive to active learning, creating a collaborative environment for students to interact with.

According to Microsoft, in a classroom of 40 children with only four PCs among them, 10 students crowd around each machine, while one student takes center position and controls the mouse. Other students point, gesture and compete for the control of the mouse, but they ultimately have no direct control of the PC and often lose interest and shift their attention elsewhere.

The company says that boosting the PC-to-student ratio by buying more PCs is not a viable solution to many schools in developing countries, and even with more machines, traditional PC set-ups do not allow for collaborative learning and teamwork.

Microsoft MultiPoint tries to solve this problem, by helping students to use PCs to learn together versus having an isolated computer experience where they are each on their own PC. In addition, MultiPoint offers an affordable way to decrease student-to-PC ratios, and provides a platform for Windows education software developers to create collaborative learning applications.

The Microsoft MultiPoint SDK makes it easy to capture clicks and other mouse actions. In addition, it also provides means to save these actions, allowing the programmer to focus on developing solutions for his application.

MultiPoint should not be confused with an application that allows multiple users to control multiple mouses to perform basic tasks on the operational system or any other type of application. In these situations, the system can not identify which mouse performed a particular action and, in general, there are no options to control the permissions of each mouse. MultiPoint is a framework that enables developers to create applications specifically designed to take advantage of using multiple devices, including the ability to recognize and manage each mouse clicks, treat each user differently and, moreover, assign different permissions for each mouse. For instance, the teacher's mouse in a classroom needs to have a permission level different from that of the students.

As noted before, the softwares developed so far are purely educational and showed to be fairly simple applications that do not require much of the MultiPoint technology, such as more advanced operations or new methods. This demonstrates to be a negative aspect of having a technology with this large potential being used in only one context.

4 Psychology Analysis

To apply the ecological group dynamics it is necessary to create a specific context and, for that to happen, the proposed application was based on a sales company. Therefore, it is essential to develop a simulation of a work environment within this type of industry. This simulation is composed by a company that needs to be managed by the users involved in the task.

At first, the candidate must choose a character to represent himself. This step is quite interesting because it has several purposes and, among them, is the ability to relieve tension in the air or, speaking more colloquially, to "break the ice." In addition, the psychologist, which is the responsible for the dynamics management, can ask the candidate at this point to present himself and ask him what was the reason that led him to choose a certain character, if that was his first choice or, if not, what was the reason for that.

Besides the questions that can be asked about the character choice, the chosen figure will also be used in the next activity as the mouse cursor. This helps for easy identification throughout the activities. The activity screens will be presented and detailed in Section 7.

After each candidate has chosen the cursor to represent himself, the manager, psychologist or another member of the Human Resources department, who is controlling the computer, will initiate the activity itself. The activity must be composed by two different groups: Group 1 and Group 2. The manager must decide in advance which candidates will be part of each group.

Each group will be responsible for sectors of a company such as Marketing, Sales, Human Resources and Production. The candidates will have to, for example, buy production supplies, hire workers and make decisions about the sale price, among other tasks, that fit the company that is selecting the candidate.

Each candidate will receive a script, assembled by the psychologist himself and printed on a sheet of paper containing information about the company. This information is about the selling product, employees available to hire as well as their curriculum and all other necessary data about the company. With these in hand, candidates should make their decisions.

This script can be mutable, depending on which features the company wants to investigate about the candidate and depending on the quality of the work he will able to achieve, if selected.

After some time, defined by the manager, the activity will end. After the end, the data analysis screen appears. It displays all the actions performed by each mouse during the activity.

Considering that the proposal presented is about a multidisciplinary project, we started by gathering information on the field of psychology from discussions with professionals. We conclude that one software with the idea described above can bring many benefits to the technique of ecological group dynamics. This can bring to the evaluators a great number of aggregated information on the candidate. Several times, the amount of information gathered is so vast that it exceeds the expectations of the evaluators and can greatly help in selecting a professional to fill the position available.

Such informations are important for the technical assessment of the candidates, since real difficulties of the job are presented [Andrade 1999]. To [Ivancevich et al. 1995], selection is the process by which an organization chooses, from a list of candidates, the person who best achieves the selection criteria for the position available,

considering the current market conditions. In short, as conceptualizes [Chiavenato 1983], the selection of Human Resources could be defined as "choosing the right man for the right job", so to maintain or increase the efficiency and productivity. For this to be done, the author also suggests that the selection needs to be based on the characteristics of the job position to be occupied in order to have greater accuracy and objectivity regarding the selection. Therefore, this study emphasizes this tool that can be developed to better suit the characteristics of the job to be filled.

The software provides better behavioral research on the candidates, not only by the data collected about the actions but also by the benefits offered. One of these benefits is that the software allows the candidate to be placed in a virtual environment, and therefore provides a higher concentration and immersion in the task at hand, reducing the tension caused by the presence of professional of human resources, many times seen as evaluators of human behavior.

With the use of multiple mouses, the investigation becomes even more interesting because when it is given an elaborate and complex task to a team, different behaviors can be noticed. For instance, candidates may subdivided themselves to meet the task at hand in different ways (hierarchy or division of tasks), and the evaluators may observe the candidates' initiative. That is, to think about how to structure a team knowing that everyone has the power for decision making.

Another very important advantage is due to the fact that during the execution of the dynamics and at the end of it, a psychologist may choose from many types of approaches to extract more information about the candidates. The activity creates a context for all the psychologists to do their research by asking questions and generating discussions. As an example, we can mention that at the end of the activity, psychologists ask each group to evaluate the reason for making a specific decision and what they would change on the decisions of the competing group.

5 System Requirements

When talking with professionals in psychology, with experience in behavioral analysis and feature extraction of human beings, a requirements assessment was administered to know the system functionalities that the Human Resources professional wants the software to have.

It is important that there are distinct roles within the application: the manager and the candidate. Each of these have different actions and have different permissions during the execution of the activity.

The manager is the one responsible to start and end the application. He is an special actor because he is the one who has control over the keyboard.

The candidate is the participant who follows the instructions of the manager. He will be asked to solve the proposed activities and will be observed and evaluated all times. He can decide how he will make his decisions, if he is going to click and modify the values of the activities or if he will discuss with his group about the best approaches to accomplish the proposed activity.

At any time, the manager may pause the activity, freezing the candidates' mouses, making them unable to perform clicks. This feature can be useful if the manager desires to stop the activity and make interesting observations, or if the time is up and he needs to end the activity.

From these considerations and the initial description of the application, we elaborated an use cases diagram, shown in Figure 2. It has the main purpose of documenting the system's functionalities and their relationships with the external environment, composed by the users. The diagram clearly illustrates how each actor, represented in this scenario by the candidate and the manager, interacts with the system. Each actor has a finite set of actions he can perform and each of these actions must be implemented within the system. It is important to have a list of all possible user interactions with the system so that any scenario is left out.

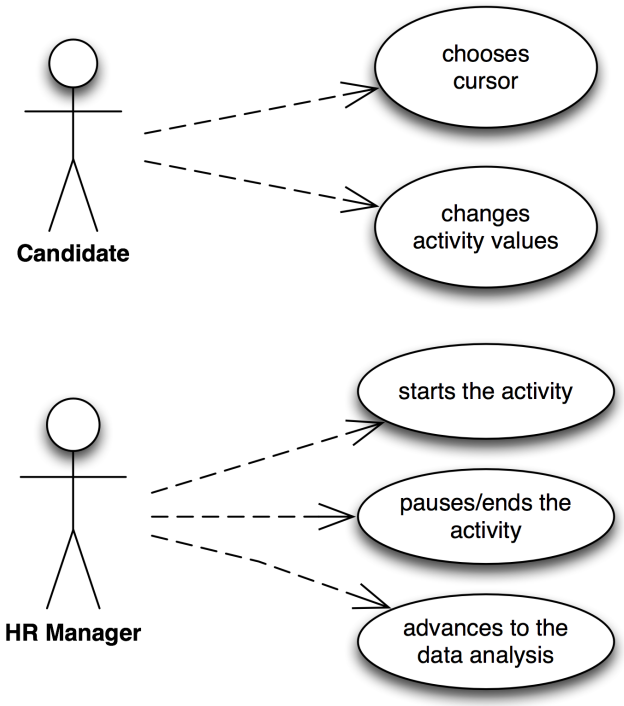

Figure 2: *Use Cases diagram*

As one can observe, the use cases diagram of the proposed application is a very simple diagram since every interaction between the user and the computer is through the several mouse devices, by simply changing the values of the activities. These interactions aim to place the candidate in a favorable environment to questions, discussions and other activities outside the virtual environment. These outside of the virtual environment activities must be proposed by the human resources manager or psychologist involved.

Upon completion of the activity, the next screen displays the statistics of each mouse's clicks. With this data, it is possible to evaluate the actions of each candidate and question them individually of the group about the reasons for each decision.

From this analysis we created a primary list of the system's requirements, in order to facilitate the design of the project:

Functional requirements:

- [FR001] – The manager must be able to start, pause and end the application.
- [FR002] – Each candidate must be able to modify the values of the activities.
- [FR003] – The system must record the clicks quantity of all mouse devices.
- [FR004] – The manager must have control over changing screens, from the activity to the data analysis screens.
- [FR005] – The system must provide different images for candidate to choose from, so it uniquely represents him.

Non-functional requirements:

- [NFR001] – The system must provide eight images of caricatured characters from different ethnic groups.
- [NFR002] – The system must provide the interaction of up to eight mouses on the same screen.

Inverse requirements:

- [IR001] – The system must not allow the candidate to pause or end the activity.

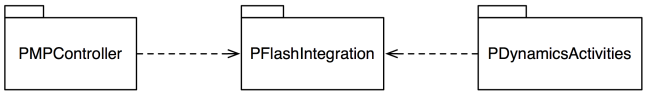

Figure 3: *Packages diagram*

6 Design and Implementation

From the analysis of the system requirements, it is necessary to detail all the steps to be done during implementation.

It is known that not only the creation of the activity is necessary but also the integration with the MultiPoint technology. The first encountered difficulty is the integration of the Visual C# module with the Flash platform. It is necessary then some sort of diagram that gives a clear notion of this integration. The Figure 3 represents this integration in form of a packages diagram.

With this diagram it becomes clear the integration of the technology with the activity itself. The diagram also exposes the modularization of this technology. One sees how we separated the classes of the MultiPoint SDK controller, the classes responsible for the functions that make the integration with Flash and dynamics activity itself.

The package defined as *PMPController* represents the classes implemented in C# and the MultiPoint SDK's Dynamic Link Library (DLL) files, responsible for managing event messages and rendering the multiple mouses on screen. The package named *PDynamicsActivities* contains the classes that belong to the Flash part of the application. These classes are responsible for the dynamics itself, for recording the values of the decisions and for displaying statistics of each mouse's clicks. The classes of this package will be detailed later.

The package *PFlashIntegration* is responsible for enabling the MultiPoint SDK integration with Flash communication through Visual C# code. This package contains DLL's for communication of events that will be received, sent and handled by the C# and ActionScript code.

To begin the activity, it is necessary take into account the roles of each actor within the system. It is known that the manager should be the only one able to control the activity's flow, that is, to move from one screen to another, to freeze the candidate's mouses and end the application. Given these requirements, we decided that all of the manager's actions will be done through the keyboard. The keyboard commands are listed below:

- ESC - ends the application.
- SPACE - freezes/unfreezes the movement and enables/disables mouse clicks.
- ENTER - proceeds to the next screen.

The codes that handle the keyboard events are defined in C#, as the following example:

```
private void KeyDown_Event(object sender,
                           KeyEventArgs e) {
   if (e.Key == Key.Escape)
   {
      App.Current.Shutdown();
   }
}
```

Following this example it is possible to handle any other keys, handling the event as the programmer sees fit. The application also features an interaction phase between the MultiPoint SDK and the activity, implemented in Flash, for the selection of the character image by each candidate. As mentioned before, each mouse cursor will be represented by an unique image chosen by each candidate. This phase is not as generic as most applications, which simply assign colors or pictures for each cursor randomly. The proposed activity must assign each candidate's cursor according to their choice. Seen as one more functionality by the psychology, the choice of the character is already considered an important phase of the group dynamics in terms of evaluating the candidate.

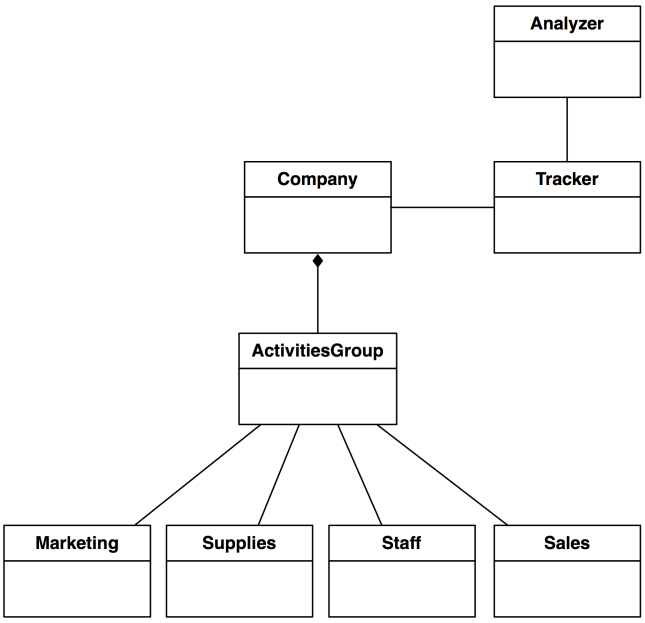

Figure 4: *Class diagram*

As the cursors are rendered by the Visual C# and the graphical interface of the activity is made in Flash, it is necessary a communication between the two parts to achieve the expected result. Taking this fact into consideration, the following implementations were made:

- On the ActionScript part of the application, the characters available for choice are shown. The candidates may select them by clicking the mouse. This sends a click event to the C# code and informs it which character was selected for which device.
- In the Visual C# part, the click event is handled so that the character figure chosen by the candidate replace the pointer image that represents him inside the application. Accordingly, the pointer becomes the user's face or a striking feature of the chosen character.

The second phase, after the cursors selection, is the activity itself. The class diagram, shown in Figure 4, portrays the general organization of the code that implements the main activity.

From the diagram presented in Figure 4, we are now able to better understand the classes and functionalities of the activity, and we present them in detail below:

- *Company*: representation of the fictional company that needs to be administered during the group dynamics. It is responsible for the activity as a whole. This class has an instance of each group of activities.
- *ActivitiesGroup*: this class holds all the different instances of activities that must be performed within a group dynamics.
- *Marketing*: Responsible for the company's marketing investments.
- *Supplies*: stores different types of production supplies that candidates can choose to use, as well as the quantity to be purchased of each type.
- *Staff*: responsible for the hiring of professionals to work in the virtual company. Salary values are stored in this class as well as the number of employees to be hired for each company's department.
- *Sales*: represents the company's sales department. It holds the decisions related to the price of the company's product.

Figure 5: *First step - cursor selection phase*

- *Tracker*: stores the mouse clicks data collected during the activity so that they can be later used for analysis.
- *Analyzer*: responsible for displaying on screen the data collected and stored in the Tracker class.

The system requires a screen at the end of the dynamics that displays the history of each mouse's clicks. The recording of this data is extremely important as a system functionality, since it provides a wider range of questions that can be made to the candidate about his behavior.

Given this need we created a class called Tracker that acts as its name suggests. In other words, it records in its data structure the ID information and the amount of clicks from each mouse, logging the candidate's actions throughout the group dynamics. This class provides methods that return detailed data in accordance to a specific input. This allows for better encapsulation of data, so that changing the internal structure of the class Tracker does not incur in side effects from to other classes that use these data.

In addition to recording the history of clicks from each mouse, another concern that arises is the arrangement of such information on the screen. To manage this in an organized way of displaying the data on screen, a class called Analyzer was created. It uses Flash components to display the ID information and the number of mouse clicks.

The classes mentioned above were implemented in ActionScript 3.0 [Adobe 2009], inside the Flash part of the application. It is important to note that the Flash tool is not used solely for the graphical interface. The Flash classes package also includes scripts to record information. With activity defined as above and the integration with the MultiPoint SDK already running, the software is now ready to be used for testing or demonstration to psychologists, or other professionals interested in this technology.

7 Results

The following sections show some screenshots, list the software requirements to run the application and suggest one possible disposition of the equipments, for a better environment preparation.

7.1 Finished product

The finished product presents a division into three screens. The first screen, and the first contact of the dynamics is the choice of characters to represent the cursors, as displayed in Figure 5. The options are displayed in a very friendly way and the candidates can make their own decisions on which character he wants to represent him within the activity.

The second screen, as seen in Figure 6, is the main stage where all of the interactions occur and where the candidates make decisions of several different types within the fictitious company. At this point, the screens with the decisions of both groups are displayed simultaneously.

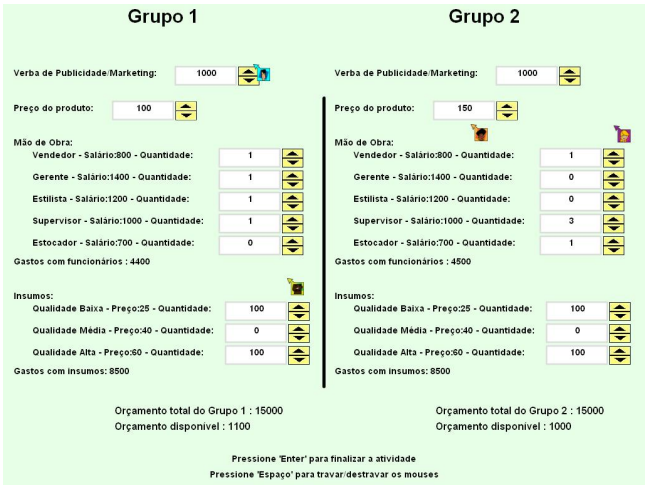

Figure 6: *During the activity*

The third screen, as shown in Figure 7, displays the data collected during the whole activity. This screen is very important and very useful to the psychologist or manager responsible for applying the dynamics technique in candidates, since that is where he can observe the behavior of the candidates, for example, in making decisions in one group that is not his, or candidates who simply did not take any decision, leaving the rest of the group to perform the task.

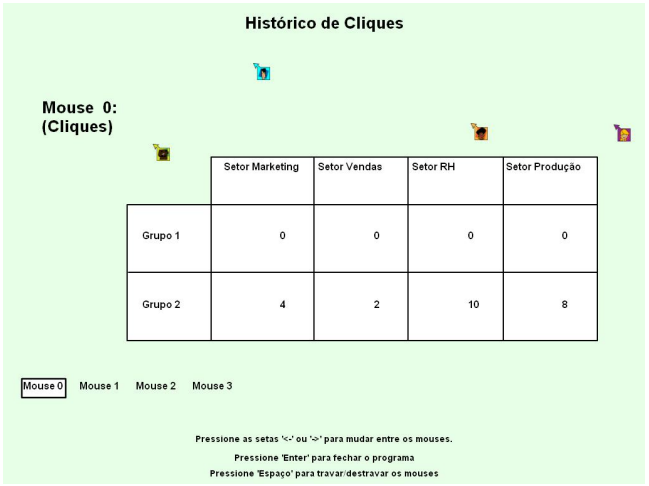

Figure 7: *Mouse clicks statistics*

After the implementation phase of the proposed application, it was possible to deliver the software to professionals in the field of psychology so that they could report their opinion on it.

The main observation made about the software is that it provides the extraction of a wide variety of information about the candidates in more details, concerning the actual task he performs. Through these observations, the evaluators can draw numerous conclusions of the behavior of the candidates. Thus, the first impression of these professionals regarding the application showed that it was well accepted.

It is still necessary to create one specific scenario of a company that can be applied in group dynamics to a team of candidates. Even so, the different questions that can be asked to the candidate have proven to be quite satisfactory for a good behavior assessment.

In summary, the developed application demonstrated to meet the expectations of psychology and human resource professionals, fulfilling the purpose for which it was created.

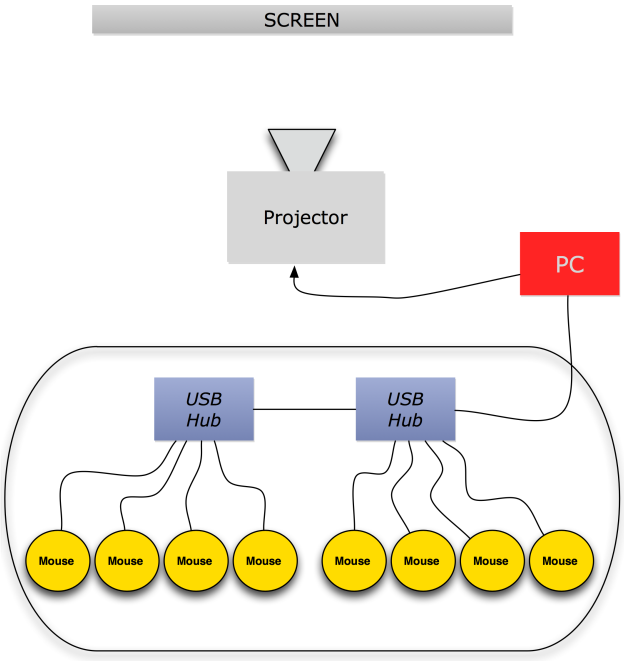

Figure 8: *Recommended equipments layout*

7.2 Software requirements to run the application

To run the application it is necessary to have installed the framework Microsoft .Net 3.0, or higher, and Flash Player 9, or later.

7.3 Environment preparation

The disposition of multiple mouse devices can be arranged in two ways: in a position similar to that of a laboratory or using a projector. In a laboratory setting, an instance of the application runs on each machine that can used by up to 4 or 6 candidates. In the configuration which uses a projector, the main computer is connected to a projector that will display a large screen that is visible to all users. All mouse devices used are connected to the computer via USB hubs. There must be a person responsible for the main computer that controls it and launches the application.

In environments where there is no projector, it is recommended a maximum of six people per computer, since those furthest away from the monitor will not take full advantage of the application, since they have difficulty viewing the monitor. When the environment is used with the configuration of a projector, it is possible to have up to thirty people interacting simultaneously on the same computer. Although the system theoretically supports up to 250 devices, notice that a very large number of cursors on the screen may cause confusion, and thereby make the application lose its original purpose.

The most suitable environment for implementing the project of a dynamic ecological group is shown in Figure 8. This configuration allows candidates to sit at opposite sides, forming two groups and candidates will be physically close enough to be able to discuss the decisions to be taken to perform the activity.

In addition, candidates from one group may also hear the discussions and decisions from the other group so they can later review the decisions of each one. Meanwhile, all the actions made by each candidate will be constantly displayed on the screen and the evaluators, which are the psychologists in this case, can monitor the candidate's behavior and draw conclusions at all times.

8 Difficulties

Given the idea to use this new technology with a distinct purpose from which it is commonly used, it is natural the risen of unexpected effects and problems during the implementation of the application. Thus, it is relevant to report the main difficulties encountered while working with technology. They are described below:

• There were problems when dealing with methods in the ActionScript language that rely on mouse events from the operating system, for example, the click events in Adobe Flash standard buttons. The native events are not recognized since the mouse cursor at the code from C# is not considered a standard operating system cursor, but only an emulated one. To solve this problem we used basic forms of interaction and handled them one by one, analyzing these events from the basic internal C# and causing them to be sent to the Flash file. An example is to imagine a simple button, that instead of having its click events handled automatically by the native Windows mouse, they are handled through messages sent by the C# code to Flash, which then tests the click on the button and performs the operations that button triggers.

• The SDK restricts each user to have only one mouse but not a keyboard. For some of the activities in the dynamics it is necessary to insert values and the lack of keyboard makes it impossible for the user to choose any arbitrary value. To try to minimize this problem, we adopted the following solution: the activities have limited options of values. Users will be given, depending on the type of activity, around 10 to 30 options to choose from. In this work, this is still a viable solution for the development of the application.

• The MultiPoint integration with Flash depends on the code of the ExternalInterface class used in Flash. A limitation that vastly complicates the programming of this interaction is that when you have classes instantiated inside each other and these have methods that respond to a single function call to external code in Flash, only the method of the last class instantiated is executed and other methods of the previous instance are ignored. To solve this, it was necessary to handle the external call in only one class and use the method that will be executed to pass the call again with all parameters to the internal functions of other classes instantiated. This is compared to a cascade effect, in order to call the methods in all required classes.

9 A Critical Analysis of the SDK

The MultiPoint SDK was able to meet the needs required by the presented work. The application successfully recognized multiple devices and it also managed to track and collect data on these multiple devices. With that in mind, it is possible to say that the SDK actually does what it claims to do, leaving the programmer to worry only about developing his application, focusing on more important funcionalities.

One major advantage presented by the SDK is its simplicity and modularity. The inclusion of only some DLL's and a simple standard initialization are needed to create a basic application that uses multiple mouse devices.

Although the SDK presented satisfactory results for the developed software, we noticed that it has some problems and restrictions that may become a stumbling block depending on the type of application developers may desire to implement. Since, in general, these problems only become clear when the programmer begins to work with the technology, some of the problems and constraints we encountered are listed below followed by brief explanations:

• Full screen only: the SDK was designed to work in full screen only. This is essential so that the operating system's native mouse cursor is not visible while the multiple mouse devices created by the MultiPoint SDK are also being rendered, thereby generating confusion for the users. Moreover, this avoids problems with the relative position of mouses' clicks, ensuring that click events are handled correctly. This limitation brings a great loss though, for example, for applications not intended to be executed in full screen or wishing to use the multiple pointers only in internal windows.

• Movement limitation of the cursors: this feature is not offered by the SDK. This would be of great importance to applications which require that specific devices are enclosed in certain regions of the screen. This would ensure that certain regions of the screen will not be clicked by unauthorized devices.

• Problems recognizing devices: we can split this problem into two parts:

- Devices incorrectly added: the SDK sometimes recognized devices that were not actually connected or were not mouse devices. From what we could observe, the SDK detects the devices according to the drivers in use by the USB ports. Later, solely those devices that are mouse devices are filtered out and then they are truly added. The problem is that this filter does not work correctly. Some devices that are not mouse devices, when connected to USB, were interpreted by the MultiPoint SDK as mouse devices and then additional cursors were wrongly rendered inside the application. In another example, the SDK also detects drivers of tablet devices just because they are installed on the machine, even if the tablet is not connected to USB.

- Devices not added: since the devices recognition is done through their USB drivers, other devices that use inputs different from USB, such as some PS/2 mouses, are not detected.

• Absence of insertion criteria: we were not able to detect an established criteria for the insertion order of the devices inside the application. It is not possible to predict, until the application is launched, what will be each device's ID. This makes it almost mandatory that identification process is done during the execution of the application, and not previously. For example: it might be interesting that only the main mouse may be able to click a specific button. To achieve that, it is necessary to somehow find the ID of that particular mouse, so that the ID information is passed to the application and only after that the corresponding mouse will receive the right privileges. It is not possible to guarantee that one certain mouse is going to have a predefined ID number.

• Low performance: depending on the type of the application developed with the SDK, the processor can become overloaded and the devices created by the MultiPoint SDK become unresponsive, creating "ghosts" due to delayed actions. These "ghosts" are caused by a delay in the rendering of the cursors.

• Poor documentation: for unknown reasons, Microsoft does not provide a complete documentation of all classes of the SDK and how they work in detail. We could only find documents with step-by-step instructions on how to start a small application and a couple of video tutorials. This is a negative aspect that makes the SDK resemble a black box.

We observed that the technology is very useful but still has many aspects that can be improved. This would broaden the range of usage possibilities even more. According to Microsoft, updates will come and maybe soon enough some of the problems highlighted above will be solved and new features will be available to the developers.

10 Conclusion And Future Work

It was possible to see how the Microsoft MultiPoint technology can be used to create applications that support multiple mouse devices on the same screen. Using this technology it was possible to develop an application specially design for ecological group dynamics.

For the development of this application, MultiPoint technology met all the needs but proved to be quite limited and often confusing to work with. The poor documentation and the several problems encountered during the development of the proposed application contributed to a negative experience with the Microsoft MultiPoint technology. It is believed that the mentioned problems will be solved and a better documentation will be written in future versions of the MultiPoint SDK. That said, we believe that developing applications with this SDK will be a more positive experience.

To gather the requirements of the system it was necessary to work together with other areas, such as Psychology. Since it is a software aimed to be used in behavioral analysis, by Human Resources professionals and psychologists, it was necessary to work together to correctly develop and test the application's functionalities.

Since the application requires that a company's specific scenario is created to be possible to apply the ecological group dynamics, we were not able to carry out tests on real candidates. In addition, performing a group dynamics incurs in financial costs to the company. However, psychologists consulted to test and comment on the application concluded that the amount of information that can be extracted with the use of our application is well above the average. This positive feedback has shown that the application has the potential to even become a commercially viable product.

Based on this and what has been seen throughout the project, we believe that the initial goal was achieved and that the proposed application meets the purpose for which it was created.

For future work, there are two main issues regarding the presented application. The first one is about the difficulties concerning the MultiPoint SDK and how to overcome these difficulties in the best way to get a more user-friendly and more useful interface. The second one is on the inclusion of new features in the tool, in order to bring more benefits and a much greater range of possibilities for anyone who is using it.

So, concerning the difficulties of working with the technology, we present the two following suggestions:

• Create a class that acts as an interpreter between the events generated by emulated mouses and the events generated by the standard mouse functions of the native operating system. That would be extremely useful for those who are developing an application using the MultiPoint with Flash integration, making it automatic to recognize any actions performed by mouse events on Flash components.

• To try to solve the problem of lack of keypad for entering any values, it is suggested to create a class that handles in a generic manner these options and their values. That may be possible through an XML data file, so that the modification looks simple to the psychologist who will control the activity, removing the need to make any changes directly to the application code. Another option would be to create a virtual keyboard placed next to the text box to be edited with a variety of characters needed, such as a numeric or an alphanumeric keypad, for example.

In more general terms, the suggested improvements concerning new features are to rescue the data from the activity and export them to an external file, such as an XML file, for instance. Thus, it would be possible to provide the means for the psychologists and dynamics evaluators to analyze the candidate's actions in a different visualization tool.

In the future, it should be done a study on the arrangement of the activity informations on the screen, in order to make the activity more organized. This allows a higher density of information and provides the insertion of scripts, which are currently distributed using sheets of paper, inside the activity without damaging the visualization of the elements on the screen and without interfering with the performance of the activity by the candidates.

In addition to these suggestions related to the software development, mentioned above, we believed that the application still needs an usability analysis and experiments simulations. If possible, it is desirable to perform experiments with real candidates, so that the psychologists using the software can report their satisfaction, as well as assess the software's efficiency.

Beyond the thorough analysis and validation of the proposed application, we also suggest the creation of new products using the technology adopted, Microsoft MultiPoint, to highlight and confirm that when applied to different contexts it can bring a large number of benefits to different types of users, thus exploring the technology's full potential.

Acknowledgements

The authors would like to thank the psychologist Vanessa Karam de Lima Ferreira, who greatly contributed to this work. The authors would also like to thank Pedro Thiago de Souza Catunda Mourão, who provided the beautiful icons used in this application.

References

ADOBE, 2009. Actionscript 3.0 reference. Help Resource Center.

AGARWAL, A., 2006. Attach multiple mice to one computer. [Online] Digital Inspiration, December.

ALENCAR, E., AND GOMES, A. A., 2005. Recursos humanos e a aplicação de testes psicológicos para seleção de pessoal.

ANDRADE, S. 1999. *Teoria e prática de dinâmica de grupo: jogos e exercícios*. Casa do Psicólogo, São Paulo.

CHIAVENATO, I. 1983. *Recursos Humanos*. Atlas, São Paulo.

GOHLKE, K., HLATKY, M., HEISE, S., AND LOVISCACH, J. 2010. Flexiknobs: bridging the gap between mouse interaction and hardware controllers. In *Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, ACM, New York, NY, USA, TEI '10, 241–244.

GUPTA, A., SHEKHAR, P., SAMDARIA, N., JAIN, M., AND PAL, J. 2010. Disha: Multiple mice in narrative content-based computer aided learning for children. *Proceedings of IDID*, 1.

INFANTE, C., HIDALGO, P., NUSSBAUM, M., ALARCÓN, R., AND GOTTLIEB, A. 2009. Multiple mice based collaborative one-to-one learning. *Computers & Education 53*, 2, 393–401.

IVANCEVICH, J. M., GIBSON, J., CUCCIO, S., AND AQUINO, C. 1995. *Organizações: comportamento, estrutura e processos*. Atlas, São Paulo.

MICROSOFT, 2009. Unlimited potential website. [Online] Available at `http://www.microsoft.com/multipoint/mouse-sdk`.

MYMOUSEGAMES, 2009. Customizable educational games for the classroom. Available from: `http://mymousegames.com` [Accessed 03 November 2009].

PAL, J., PAWAR, U., BREWER, E., AND TOYAMA, K. 2006. The case for multi-user design for computer aided learning in developing regions. In *Proceedings of the 15th international conference on World Wide Web*, ACM, 781–789.

PAWAR, U., PAL, J., AND TOYAMA, K. 2006. Multiple mice for computers in education in developing countries. In *International Conference on Information Technologies and Development*, Citeseer.

PAWAR, U., PAL, J., GUPTA, R., AND TOYAMA, K. 2007. Multiple mice for retention tasks in disadvantaged schools. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 1581–1590.

SHACHAR, H., AND SHARAN, S. 1994. Talking, relating, and achieving: Effects of cooperative learning and whole-class instruction. *Cognition and Instruction 12*, 4, 313–353.

SHEKHAR, N., GUPTA, A., AND HUTCHFUL, D. 2011. Teachers as game designers: Lessons from a collaborative learning exercise.

TECHFLASH, 2009. Multipoint games put brothers in running for big microsoft prize. [Online] Seattle Techonology New Source.

UNITEDNATIONS, 2009. United nations millennium development goals. [Online] `http://www.un.org/millenniumgoals`.

VILLANUEVA, P. G., TESORIERO, R., AND GALLUD, J. A. 2010. Multi-pointer and collaborative system for mobile devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, ACM, New York, NY, USA, MobileHCI '10, 435–438.