

# Towards IDEs for Physically Challenged Programmers

Amir Zeid, Hani Ghalayini, Mohammad Al-Sulaiman, Ghanim Al-Ghanim, Abdulla Al-Jassim,  
Abdulrahman Al-Failakawi

American University of Kuwait  
Division of Sciences and Engineering

## ABSTRACT

Programming is one of the activities that need talent and persistence. Physically challenged people avoid programming jobs and related academic degrees due to lack of tools that accommodate their disabilities. Most people think to program for the physically challenged. Our approach is different, we provide tools to make the physically challenged write computer programs in an easy and affordable way.

In this paper, we introduce Proviac (Programming Via Voice) which is a software that helps physically challenged programmers to write their programs in a friendly way. Proviac currently supports C# but it could be extended to include other programming languages. Proviac can be considered a step towards providing IDEs tailored for physically challenged programmers.

## General Terms

Software engineering, Programming languages.

## Keywords

Programming environments, Physically challenged programmers.

## 1. INTRODUCTION

Many people think that physically challenged people lack the abilities and skills needed in our world, and that they can't perform in some fields and aspects like normal people would do. Others think that physically challenged are not mentally sound, however this is not true. When people lose an ability, they either develop an existing ability to become a substitute for the missing one, or they gain new abilities that make them special and help in overcoming the daily challenges of life. A real life example for this is Ben Underwood, a 17 years old boy who lost both of his eyes because of cancer when he was 2 years old. Ben developed an ability that helps him in moving around and navigating his surroundings using his voice and its echo; Ben says that "there is nothing that I can't do," as he skates in the streets and surfs like any normal healthy boy [1].

But not all physically challenged people are able to overcome their challenges, and here comes the role of scientists to develop new technologies and tools that will help challenged people. Proviac (Programming Via Voice) is a software created to help the physically challenged realize their potential in programming. A prototype was developed and tested by some physically

challenged computer science students at the American University of Kuwait.

The rest of this paper is organized as follows: In section 2, the background about the problem and the solution are introduced. In section 3, the architecture of Proviac is introduced. In section 4, we show how to use Proviac. In sections 5 related work and assessment are discussed. Finally, section 6 concludes the paper and provides some pointers for future enhancements.

## 2. BACKGROUND

### 2.1 Statistics

According to the US Census Bureau, there are around 43 million Americans with special needs, and according to the US bureau of labor the employment population ratio for people with special needs was 19.2 in 2009 [3]. The percentage of disability for people working in computer and mathematical jobs is 1.6%, and the percentage of disability in the whole information technology industry is 2.1% [4].

Currently, we could not find published statistics for the Middle East region about physically challenged programmers. Yet, as far as we know the number of such programmers is very minor.

### 2.2 Microsoft Speech Recognition Tool

Windows Speech Recognition allows the user to control the computer by giving specific voice commands. The program can also be used for the dictation of text so that the user can control their Vista or Windows 7 computer.

Applications that don't present obvious "commands" can still be controlled by asking the system to overlay numbers on top of interface elements; the number can subsequently be spoken to activate that function.

Windows Speech Recognition has a fairly high recognition accuracy. Training could also be completed to improve the accuracy of speech recognition. Currently, the application supports several languages, including English (U.S. and British), Spanish, German, French, Japanese and Chinese (traditional and simplified). Support for additional languages is planned for future releases [5]. We used Microsoft Speech Recognition tool to capture users' commands.

### 2.3 Camera Mouse Software

Camera Mouse is a freeware to control the mouse pointer on the screen using head movements[2]. The program was developed to help people with disabilities use the computer. The main audience for this program is people who do not have reliable control of a hand but who can move their head. People with Cerebral Palsy, Spinal Muscular Atrophy, ALS, Multiple Sclerosis, Traumatic Brain Injury, various neurological disorders use this program and its predecessors to run all types of computer software. It will be used to help physically challenged users to edit the code generated by Proviac.

### 2.4 Problem in more details

Physically challenged people with hand related limitations can be smart, talented and creative in many fields including computers and technology. However because of their disability they can't fully interact and use computers, especially when it comes to computer programming. Some people argue that physically challenged people with hand related limitations can program using the current voice recognition technology. As we know this technology is still relatively new and doesn't offer any support for the programming commands because it's not specifically designed for it. It is not sufficient by itself to support programmers.

### 2.5 Proposed Solution

Proviac (Programming via voice) will help physically challenged people to program without using their hands. Proviac will interact between Microsoft Speech recognition tool and Microsoft Visual Studio 2008 IDE. Users will communicate with Proviac via voice commands. Using the concept of voice commands will change the way programming is done. Proviac will utilize a variety of templates which will make it easier for the users to develop computer programs. Consequently, syntax errors are minimized.

## 3. PROVIAC ARCHITECTURE AND USE CASES

### 3.1 Use Cases

Proviac can support three different user scenarios:

- Physically challenged users (with hand related limitations) can use Proviac to write computer programs using voice commands without needing to use hands in the process. They can also edit the code using the camera mouse software which simulates all mouse-like interactions.
- Normal users
  1. Expert Programmers: can use Proviac to write and edit computer programs using any standard keyboard and mouse in addition to the feature of using voice commands.
  2. Beginner Programmers: can use Proviac to learn how to program in C# programming language.

Figure 1 shows the normal user and the administrator's use case. The administrator can add programming languages support.

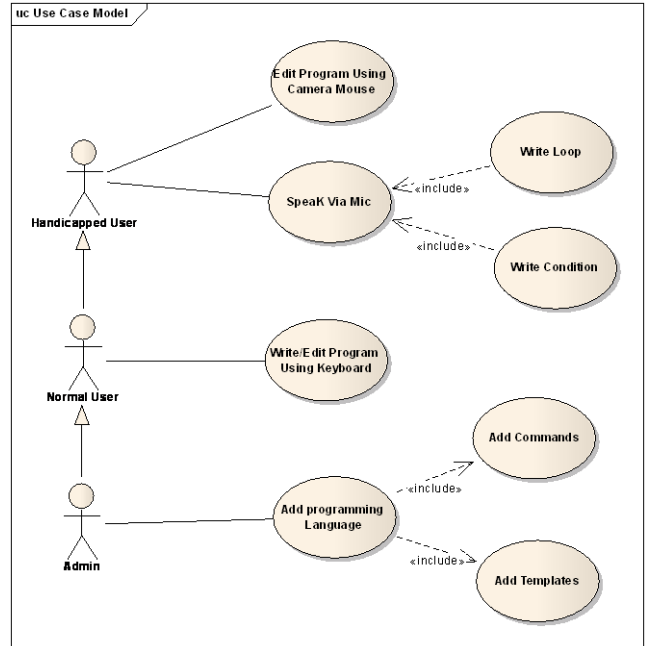


Figure 1 Use case diagram of Proviac.

The handicapped user can perform two types of commands. The first one is the menu commands, which are: open, run, compile, and save a file, or exit. The second is the write code commands, which are: write text directly, choose variable type, choose loop, choose condition, and delete.

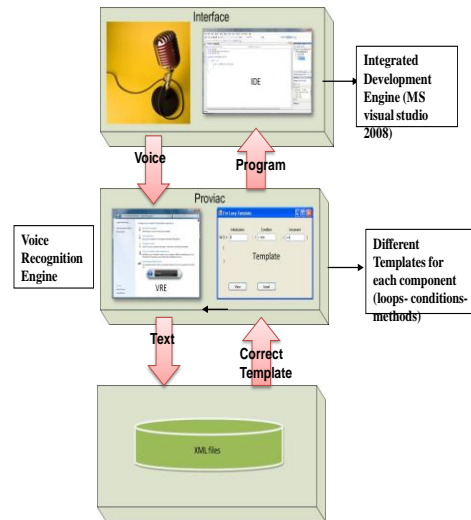


Figure 2 The Architecture of Proviac.

### 3.2 Architecture

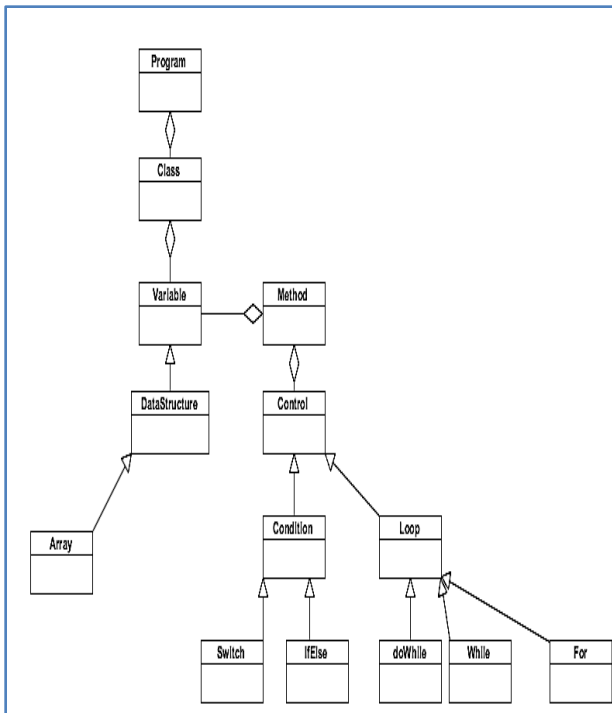
Proviac has 3 layers as illustrated in Figure 2:

1. User Interface Layer: this layer includes a microphone and the integrated development environment. Currently we are using Microsoft IDE to test the prototype. This layer mainly captures the input of the user through the Microphone. In addition, it is used to display the generated code resulting from user's choices.
2. Proviac Layer: this layer includes the voice recognition engine and different templates for each component (loops-conditions-methods-classes). The voice recognition engine mainly uses Microsoft Speech Recognition tools to interface with Proviac and to generate the required templates according to the chosen components.
3. XML Grammar Layer: this layers includes the grammar of the C# programming language.

In addition, there is a Website where users register. The site includes a quick link to download the latest available version of Proviac. The site also includes XML files for different programming languages.

### 3.3 Class Diagram

The class diagram of Proviac is shown in Figure 3. Proviac has two main types of classes: the voice recognition engine and the components. The components are a variety of classes that represents programming component. Some of them stand alone, like class and method, and others like loop and condition have different classes inherited or aggregated from them.



**Figure 3 Main classes of Proviac.**

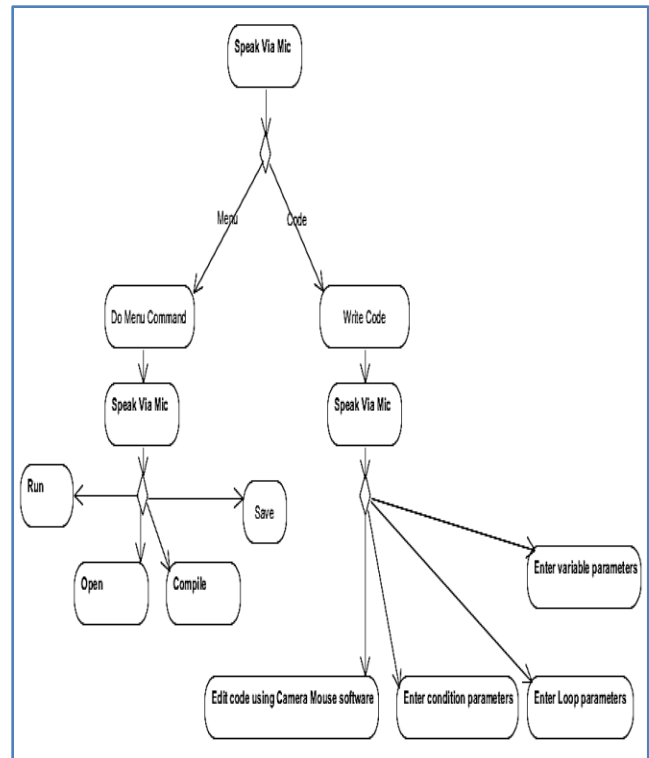
### 3.4 Used technologies

Since Proviac was one of the competing projects in Microsoft Gulf imagine cup, the initial prototype was developed using the following Microsoft tools:

- Microsoft .Net framework v3.5 used for manipulating the voice recognition libraries with the IDE.
- Windows 7 Operating System which has more enhanced voice recognition .
- Microsoft Visual Studio 2008 Professional Edition as the IDE used for developing the software as well as the end user IDE used to program.
- Camera Mouse software: a freeware to control the mouse pointer on the screen using head movements[2].

### 4. USING PROVIAC

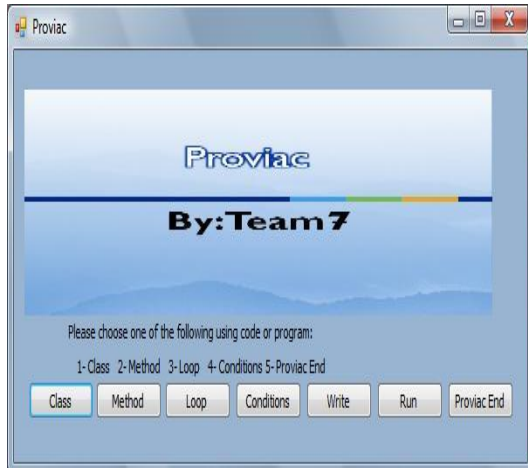
Figure 4 shows the activity diagram for Proviac. The programmer will use a microphone as an input device. After that, the voice will be sent to Proviac's voice recognition engine (VRE) where it will be converted to text. The text will be compared with the XML (Extensible Markup Language) files to check for similar commands. Proviac will get the output and write it using the corresponding template. Finally, Proviac will write the corresponding code or command into the integrated development environment (IDE) in use by the programmer.



**Figure 4 Activity Diagram for Proviac.**

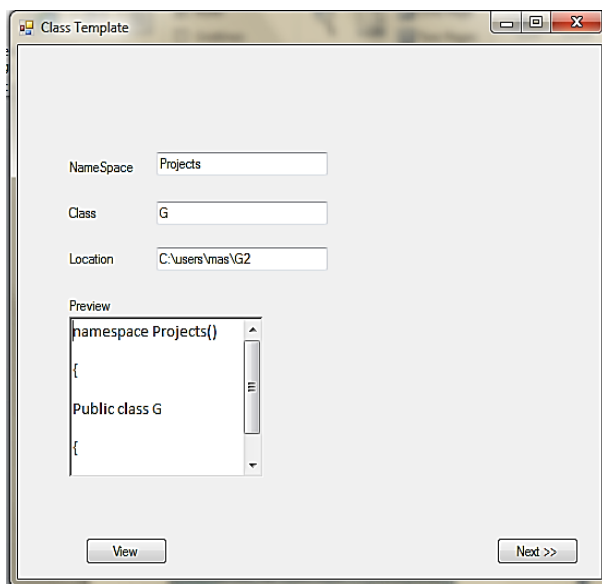
The following example is a simple program written with the help of Proviac. The program prints the output of even numbers between 0 and 10. The program is written in the main screen of Microsoft Visual Studio IDE.

Figure 5 shows a snapshot of the main interface of Proviac. The user can select from various options (class, method, conditions...) either by using voice commands or by normal selection. The following steps will show how a user produces a simple program using Proviac.



**Figure 5 Main interface of Proviac.**

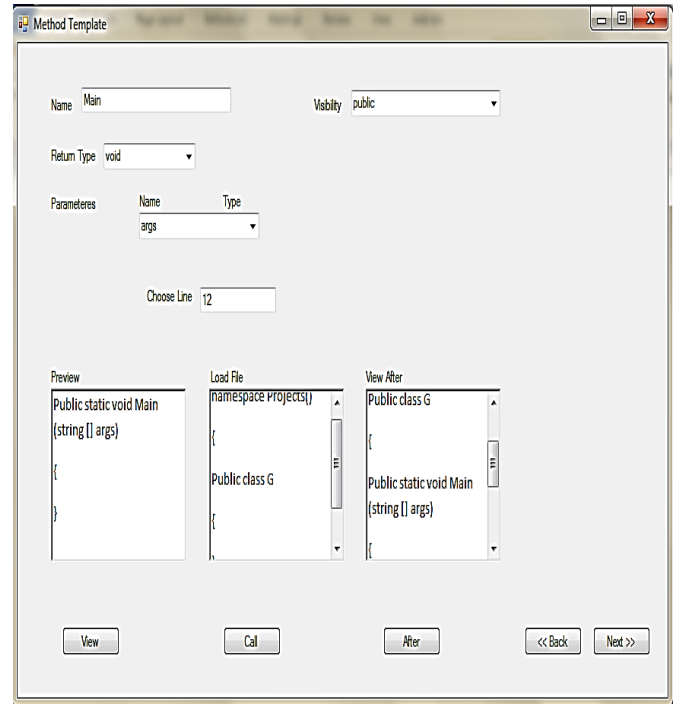
Figure 6 shows how a user starts by creating a class using Proviac. The user has to specify the class name and then it will be inserted properly in the code.



**Figure 6 Creating a class.**

In Figure 7 the user accesses the method menu where he/she can specify method name, choose the method type from a list of options, view and call the method. The user then inserts the

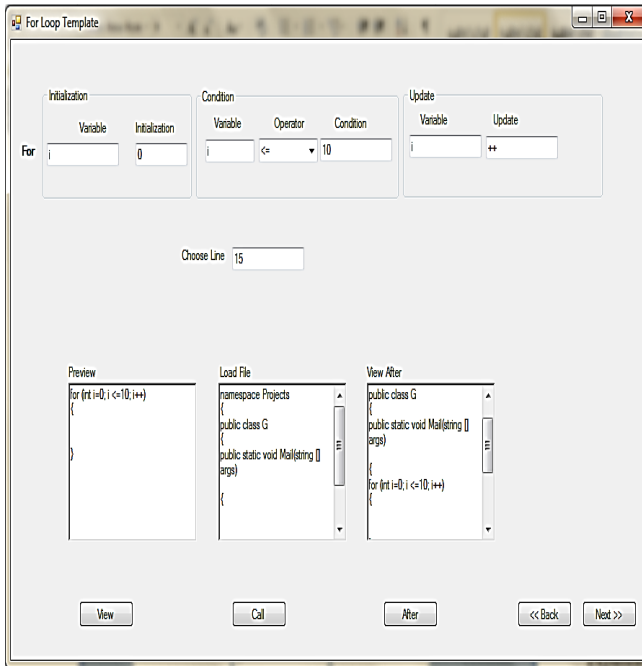
method code at the required position (in this particular example it is after the current position).



**Figure 7 Method template**

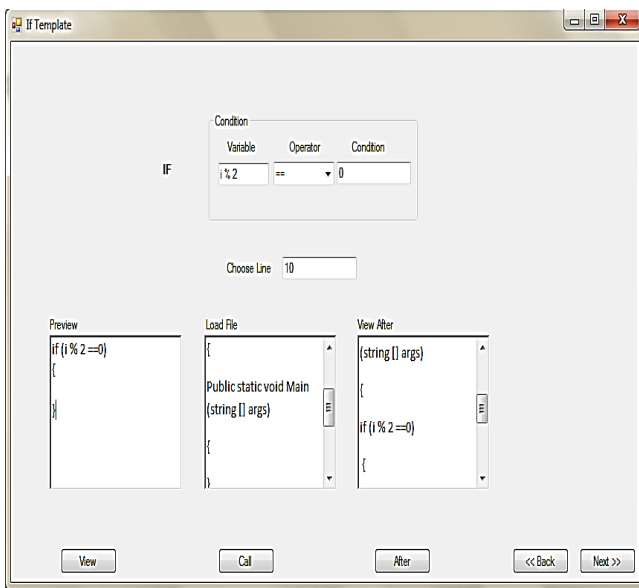
Figure 8 shows the template of the “for” loop, the user will be able to select the option loop from the main menu and then will be presented with different loop components to choose from. In the loop part, the user will be able to specify the variables, initialize them, select the operator from a list of different operators as well as specify the conditions. In this particular example, the user inserts a loop that goes from 0 to 10.

Figure 9 shows the template of an “if” statement. Using the same theme, the user will input the components of the condition to be checked, then he/she will be able to insert the code at the right position using the choose line box. He can view his code in the view after pane. In this particular example, the user sets the condition to  $(i\%2 == 0)$  to check for the even numbers.



**Figure 8 Loop template.**

Finally the user inserts an output command by choosing the ConsoleWrite template then inserting the command at the right position.



**Figure 9 “If” template.**

After the user is done editing his program, he can move to publish his program. The outcome of this stage will display the program in an IDE.

## 5. Similar work and assessment

### 5.1 Similar Work

Most people think to program for the physically challenged. Our approach is different, we program to make the physically challenged program. Raman and Lee [6] came up with a speech enablement for Eclipse IDE using AspectJ.

There are several products that target the same problem. An example for that is the seeing tongue technology. This technology uses a camera connected to a sensor that converts the images captured by the camera to signals. The user puts the sensor on his tongue; the signals will be received by the user’s tongue and sent to the brain simulating the eyesight [7].

Netbeans for Java [8] provides code templates. Code templates are predefined pieces of code that have abbreviations associated to them. The users type the abbreviation and the code will be expanded accordingly.

Scratch [9] was developed at MIT to help young kids learn programming. We used some of the ideas of Scratch like the templates for the programming constructs. It teaches kids to do programming using the same way they assemble jigsaw puzzles. However, Scratch is a programming language by itself. It cannot be used with other existing languages.

### 5.2 Assessment

We carried out experiments at the American University of Kuwait to assess Proviac. We conducted several tests to assess the software, and general observations on the software included the following comments.

#### 5.2.1 Instructors’ comments:

- My first initial reaction was that I never heard of similar tools before; however, it is very challenging to have a complete product to support physically challenged.
- Using Proviac for educational purposes could help all types of students. The difference is instead of typing, you would be speaking it, so you could be driving and actually programming at the same time. That’s really the wave of the future. There are many venues that we’ll be seeing in the future but I think voice recognition is one of them and it’s still in its infancy so far. For handicapped people or people with amputated hands who won’t be able to use the keyboard Proviac will reach out and bring out more of their capabilities. I think it’s a very unique and useful thing now and in the future.
- Yes absolutely I can use it. It is simple for anyone to understand and use. It will be great that this will be introduced to society so parents with children with physically challenged conditions can feel comfortable and see an improvement in their kids, being able to do something without supervision.

#### 5.2.2 Students’ assessments

- Proviac was found user friendly because of the use of various forms and drop-down lists that show the different commands for the user to choose from.

- The use of grammar files made the voice acquisition process for inputting the commands faster even though the user needs to train the engine to get better recognition performance.
- The users found the option to review the code before it could be inserted into the Integrated Development Environment (IDE) useful.

## 6. CONCLUSION AND FUTURE WORK

We are currently planning to provide the following enhancements:

- Support other programming languages.
- Support more Integrated Development Environments.
- Proviac will be implemented as an add-on for Microsoft Visual Studio IDE.
- Future versions of Proviac will include command lists, tool tips and help frames.

In this paper, we presented Proviac which is a software to help physically challenged to write computer programs using the C# programming language. The applicability of the system is open. It could be used with different programming languages for different educational and professional purposes.

Proviac will be used to experiment with at the American University of Kuwait. Then we will conduct a detailed survey to analyze the opinions of physically challenged programmers.

## 7. REFERENCES

- [1] “Ben Underwood a Celebration of Life” Accessed February, 15<sup>th</sup> 2011, <http://www.benunderwood.com/aboutme.html>
- [2] Margrit Betke, James Gips, “The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities” IEEE Transactions on Neural Systems and Rehabilitation Engineering, VOL 10, NO 1, MARCH 2002
- [3] Matthew Brault, “Disability Status and the Characteristics of People in Group Quarters: A Brief Analysis of Disability Prevalence Among the Civilian Noninstitutionalized and Total Populations in the American Community Survey” US Census Bureau, 2008. <http://www.census.gov/hhes/www/disability/GQdisability.pdf>
- [4] “U.S. Employment Statistics for Persons with a Disability” U.S. Bureau of Labor Statistics, 2010. Accessed February 10<sup>th</sup>, 2011, <http://www.disabled-world.com/disability/statistics/disability-employment-statistics.php>
- [5] “How to Use Speech Recognition with Windows 7” Windows 7 News and Tips, Accessed February 9<sup>th</sup>, 2011, <http://www.windows7news.com/2009/09/18/how-to-use-speech-recognition-with-windows-7/>
- [6] TV Raman, Aliso Lee, “Speech-Enablement of Eclipse IDE and Eclipse Rich Client Applications Using AspectJ” Web Collaboration. Accessed February 10<sup>th</sup>, 2011. “<http://www.webcollab.com/alee/papers/avios05.pdf>”
- [7] Jonathan Beale, “New glasses allow blind soldier to 'see' with his tongue” BBC News, Accessed February 10<sup>th</sup>, 2011. <http://news.bbc.co.uk/2/hi/8568547.stm>
- [8] “Code Assistance in the NetBeans IDE Java Editor: A Reference Guide” NetBeans Documentation. Accessed February 11<sup>th</sup>, 2011. <http://netbeans.org/kb/docs/java/editor-codereference.html#codetemplates>
- [9] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. (2010). The Scratch Programming Language and Environment. ACM Transactions on Computing Education, November 2010.