

ProBot: Juego para el aprendizaje de lógica de programación

Julian Moreno

Universidad Nacional de Colombia
Colombia
jmoreno1@unal.edu.co

Edgar Alberto Montaña

Universidad Nacional de Colombia
Colombia
eamontan@unal.edu.co

ABSTRACT

In this paper ProBot is described, an educational game in the field of programming logic in which students may reinforce and improve their abilities with regard to the concepts of sequencing, defined iteration and nesting. The game was implemented in Flash, it has an intuitive interface and it is aesthetically appealing. It owns several difficulty levels that are challenging for students and encourage them to analyze their solutions looking for the ones that are more efficient.

RESUMEN

En este artículo se describe ProBot, un juego educativo en el campo de la lógica de programación de computadores en el que lo estudiantes pueden afianzar y mejorar sus habilidades respecto a los conceptos de secuenciación, iteración definida y anidación. El juego fue implementado en flash, cuenta con una interfaz intuitiva y es estéticamente llamativo. Posee varios niveles de dificultad que son desafiantes para los estudiantes y que los incentiva a analizar sus soluciones en búsqueda de aquellas que sean más eficientes.

KEYWORDS

Juego educativo, Lógica de programación, Flash, Compilador.

INTRODUCCIÓN

El juego como mecanismo de aprendizaje y mejoramiento de habilidades es innato no sólo para los humanos si no para muchas otras especies.

En el contexto académico, tal como se menciona en [15] los juegos pueden ser una herramienta auxiliar del alumno en la construcción de su conocimiento sistematizado. Esta sistematización, por medio del computador, posibilita un mejor acompañamiento del alumno verificando sus errores más frecuentes y presentándole recursos multimedia de manera diferente a como se hace en el modo convencional usado en el aula de clase.

Una amplia variedad de estudios señalan la importancia de la utilización de juegos, y particularmente videojuegos, para alcanzar objetivos de aprendizaje como el desarrollo verbal, matemático, lógico, visual, motor-sensorial, así como habilidades de resolución de problemas [13]. Otros estudios muestran que los videojuegos pueden aumentar diálogos significativos entre los estudiantes y que tienen efectos positivos en las habilidades sociales [11]. Este mismo autor

argumenta que utilizar videojuegos para el aprendizaje abre enormes oportunidades para llevar experiencias de la vida cotidiana a la educación de los alumnos, aumentando la motivación y el compromiso con el aprendizaje.

En otros trabajos a lo largo de las dos últimas décadas [4-8, 12, 16, 17] se presenta una visión general sobre el uso educacional de los videojuegos, describiendo su utilidad en diversas áreas del conocimiento

En el caso particular de la enseñanza de las ciencias de la computación, tal como lo menciona [3], el uso de software educativo se encuentra rezagado respecto a otras disciplinas, aunque el panorama ha ido cambiando en los últimos años. Algunos ejemplos del uso de juegos para diferentes dominios de conocimiento dentro de las ciencias de la computación se presentan en los trabajos [2, 9, 10, 14, 18, 19].

El trabajo presentado en este documento se enfoca en el campo específico de la lógica de programación sin hacer énfasis en ningún lenguaje específico si no abordando el problema general del diseño de algoritmos considerando la programación estructurada [1]. Al menos en Colombia su enseñanza en los niveles de básica secundaria no es obligatoria aunque si lo es para la mayoría de programas de pregrado en ingeniería y obviamente en los programas técnicos en el área de informática.

Volviendo al tema, como pueden constatar los docentes familiarizados con cursos relativos a este tema, la dificultad en su enseñanza radica en que programar es parte ciencia y parte arte, en el sentido que al estudiante se le pueden enseñar las instrucciones básicas pero depende de él su utilización de una u otra manera para la solución de un problema específico. Basta por ejemplo proponer un ejercicio simple en un aula de clase como: “diseñar un algoritmo para leer tres valores numéricos y determinar cuál es el mayor de los 3” para verificar que en un grupo de digamos 100 alumnos pueden encontrarse al menos 10 tipos de soluciones diferentes. Tales soluciones se diferenciarían no sólo en cuestiones estéticas (el uso de más o menos instrucciones), si no en otras más importantes desde el punto de vista de la computación como la eficiencia respecto al uso de recursos (procesamiento y memoria).

Teniendo presente esta característica y considerando que en muchas ocasiones la dificultad en el aprendizaje de programación radica en que algunos conceptos pueden ser muy abstractos para muchos estudiantes, se propone en este trabajo un juego educativo llamado ProBot con el que se pretende que el estudiante afiance sus conocimientos respecto a la secuenciación y la iteración definida (dos de los pilares de la programación de computadores). Este se

Moreno, J., Montaña, E. (2009). ProBot: Juego para el aprendizaje de lógica de programación. En J. Sánchez (Ed.): Nuevas Ideas en Informática Educativa, Volumen 5, pp. 1-7, Santiago de Chile.

fundamenta en un juego de Boxeo en el que el estudiante debe competir contra una serie de oponentes que emplean cierta serie de movimientos. Para esto el estudiante debe “programar” al ProBot definiéndole a manera de algoritmo cuales movimientos debe efectuar para contrarrestar los del oponente y ganarle.

DESCRIPCIÓN DEL JUEGO

Interfaz

El juego fue desarrollado completamente en flash con el fin que los estudiantes puedan acceder a él desde un navegador Web luego de haberlo cargado a un servidor (local o de Internet), por lo que puede ser utilizado tanto en cursos

presenciales como virtuales. La interfaz es completamente visual y se divide básicamente en 4 partes tal como se muestra en la figura 1.

En la parte superior izquierda se muestra el nivel en el que se está jugando, mientras que en la parte superior izquierda se muestran las opciones para habilitar y deshabilitar los sonidos del juego y de la música de fondo. En la parte superior central se muestran las instrucciones que se encuentran disponibles para el nivel correspondiente y que constituyen el conjunto de estructuras (hablando en el contexto de los lenguajes de programación estructurados) válidas para los algoritmos que el estudiante puede definir.



Figura 1. Interfaz del juego

En la parte izquierda se muestra el espacio de trabajo donde el estudiante puede definir su algoritmo de manera secuencial. Este espacio puede ser la analogía de el archivo fuente de algún lenguaje de programación, como por ejemplo el que se muestra en la figura 2, con la diferencia que el número de líneas de “código” que pueden escribirse está limitado a una cantidad determinada de casillas, donde cada casilla puede contener una única instrucción.

En la parte inferior se encuentran dos opciones: a la derecha un botón con forma de guantes con el que se puede observar “el entrenamiento del oponente”, es decir, los movimientos que este hará durante la pelea; mientras que a la izquierda un botón con forma de campana con el que puede iniciarse la interacción entre el ProBot (acorde con las instrucciones que se le hayan programado) y el oponente. En la parte inferior también se encuentran las variables que denotan el estado

tanto del oponente como del ProBot durante la pelea. Para el primero se muestra el nivel de resistencia (como es común en la mayoría de video juegos de lucha), mientras que para el segundo además de este nivel se muestra el de energía. Este último fue agregado al juego con el fin que los estudiantes tengan en cuenta que algunos problemas pueden tener restricciones de recursos.

Finalmente en la parte central de la interfaz es donde se lleva a cabo el desarrollo del juego como tal y, aunque es relativamente simple, fue diseñado pensando en que fuera estéticamente agradable y llamativo.

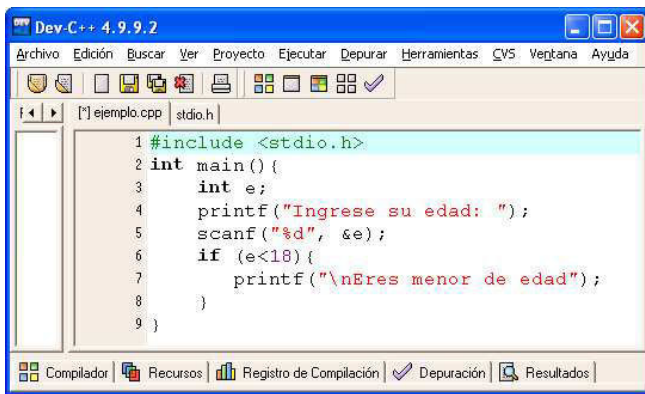


Figura 2. Ejemplo de archivo fuente en C++ empleando el IDE Dev-C++

Compilador

En el contexto de la informática, un compilador puede definirse como un programa que permite traducir un código fuente definido en un lenguaje de alto nivel, a otro lenguaje de nivel inferior (típicamente lenguaje máquina). Precisamente, esta es la función principal del motor de ejecución del ProBot que se encarga de traducir las instrucciones que el jugador define en una secuencia de acciones llevadas a cabo por el personaje.

Un compilador generalmente se compone de dos elementos fundamentales: un analizador lexicográfico (lexer) y un analizador sintáctico (parser). El primero se encarga de validar la sintaxis de las instrucciones y el segundo de identificar la semántica del código. En el caso del ProBot no se hace necesario un lexer pues las instrucciones no se escriben libremente si no que se escogen del menú que aparece en la parte superior de la interfaz.

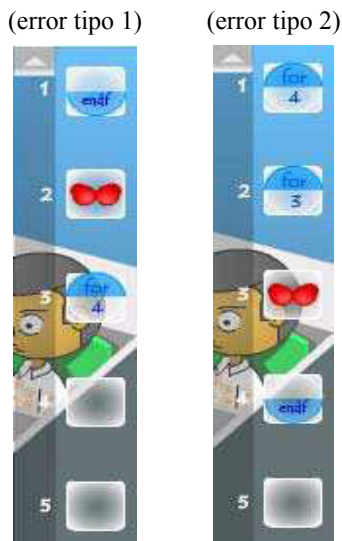


Figura 3. Errores semánticos que pueden presentarse en ProBot

Las funciones del parser para el ProBot se enfocan básicamente en dos aspectos. El primero es la traducción de las instrucciones, siempre y cuando estén bien escritas, en código que el ProBot pueda entender. Precisamente respecto a ese concepto de “bien escrito”, el segundo aspecto es la identificación de los errores en el código en cuanto al uso de la iteración definida. Estos errores pueden referirse a que el orden de las instrucciones esté equivocado (error tipo 1) o a que no coincidan el número de inicio de las instrucciones de iteración con el número de cierres (error tipo 2). La figura 3 muestra dos ejemplos de estos errores, mientras que la figura 4 muestra la analogía del segundo error con el que un estudiante podría cometer en un código real en C++.

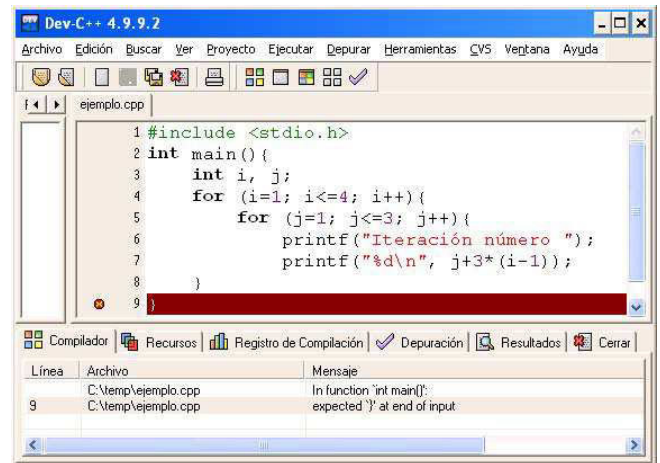


Figura 4. Error en C++

Niveles

El juego cuenta en total con siete niveles de dificultad los cuales van aumentando gradualmente a medida que los jugadores van progresando. A parte de que esta mecánica es obvia en los videojuegos de diferente índole para mantener el interés de los jugadores, también tiene una finalidad pedagógica en el sentido que presenta un orden natural para abordar los conceptos que buscan impartirse en este juego en particular. Es decir, los primeros niveles tienen que ver con el concepto de secuenciación, los siguientes adicionan el concepto de iteración definida y los últimos el de anidación de instrucciones.

Otra característica fundamental de cada nivel, tal como se muestra en la figura 5, es la incorporación de nuevas instrucciones disponibles así como de casillas en el área de trabajo que le permiten al estudiante definir algoritmos más complejos a medida que avanza. Esta característica asemeja lo que sucede en los juegos tipo RPG (sigla en inglés para Role-Playing Game, que significa literalmente: “juego de interpretación de papeles” y que es un tipo de juego en el que, uno o más jugadores desempeñan un determinado rol y donde generalmente sus habilidades se incrementan a medida que el juego avanza).



Figura 5. Interfaz del juego

Personajes

Como puede observarse en las figuras 1 y 3 todos los personajes a excepción del protagonista tienen figuras humanas caricaturizadas. Cada nivel cuenta con un oponente diferente, los cuales se muestran en la figura 4, y que poseen su propio nivel inicial de resistencia y fuerza en los golpes.

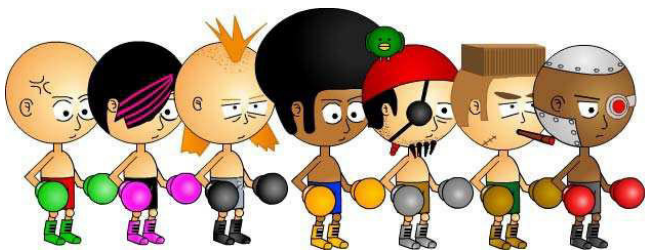


Figura 6. Oponentes

Cada uno de estos oponentes tiene sus propias características de sonidos, gestos y mensajes, las dos primeras tienen el fin de entretener al jugador, mientras que la última busca alentararlo avivando su deseo de ganarle. Como se muestra en las figuras 1 y 3, el público también va cambiando incorporándole los oponentes vencidos así como nuevos espectadores.

Puntuación

Como también es común en muchos videojuegos se incorporó el elemento de “claves de nivel” con el objetivo de que cuando un estudiante supere un nivel pueda guardar esta clave para ingresar al juego en otro momento y

retomararlo desde el punto donde lo había dejado y no tener que empezar siempre desde el inicio.

Esta característica esconde otra funcionalidad adicional y es la de identificar el puntaje acumulado que el jugador tiene hasta el momento, estos se hace de una manera encriptada como se muestra en la figura 5 para evitar manipulaciones en dichos valores.

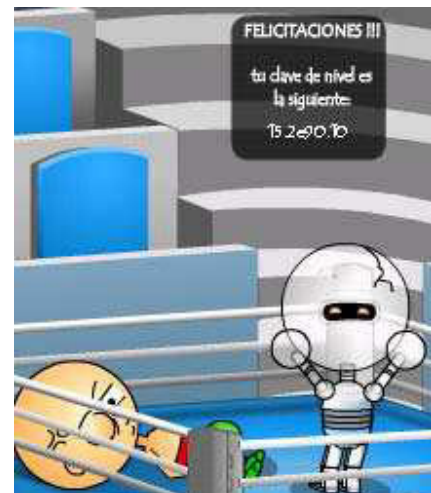


Figura 7. Presentación de clave de nivel

Para utilizar esta clave se emplea la última opción de la pantalla inicial del juego, como se muestra en la figura 6.



Figura 8. Sección de la pantalla inicial

Para este juego la puntuación de un nivel se definió como la suma entre el nivel de resistencia y energía restantes luego de ganarle al oponente, y por tanto la puntuación acumulada en el nivel n corresponde a la sumatoria de la puntuación obtenida en los niveles $n, n-1, n-2, \dots$, hasta el 1 .

Como ya el lector podrá intuir, esta puntuación da cuenta de la eficiencia de los algoritmos definidos por el jugador pues al igual que en una situación de clase o real, para un determinado problema pueden haber muchas maneras de resolverlo, pero no todas ser igual de eficientes. Con esta característica lo que se pretendió fue promover por medio de la sana competencia buenas prácticas en el ámbito de la programación, pues en su afán de obtener un puntaje alto se fuerza al estudiante a pensar en el rendimiento de las soluciones que proponen en vez de simplemente cumplir con el problema que se le pone.

Este fenómeno fue validado durante las pruebas con estudiantes reales que se describen en la siguiente sección, donde se observó que muchos estudiantes al notar que su solución aunque satisfacía el problema (permitía ganarle al oponente), podía ser mejor, preferían volver a jugar el nivel en vez de avanzar, permitiéndoles mejorar sus habilidades.

Para que los estudiantes tuvieran conocimiento de dichas puntuaciones se implementó una aplicación Web donde se podían ingresar los nombres y las claves de nivel con el fin de validarlas y mostrar un “ranking” de manera dinámica. La tabla 1 muestra el tipo de información que se les presentaba, donde es claro que ellos mismos pueden medir la eficiencia de sus soluciones respecto a la de los demás compañeros.

PRUEBAS

Dado que al momento de la presentación de este artículo el desarrollo del juego descrito apenas está terminando, se decidió hacer una prueba con estudiantes de la Universidad Nacional de Colombia – Sede Medellín, no del curso de “Fundamentos de programación” impartido en los primeros semestres que sería el público objetivo, si no con estudiantes de semestres más avanzados que ya lo cursaron. La idea de esta prueba fue usar esta población a manera de “Beta testers” (en el contexto de la informática se consideran usuarios de software pendiente de terminar su fase de desarrollo, o alcanzar un alto nivel de funcionamiento, pero que aún no son completamente estables. Los beta testers

usan sus conocimientos informáticos y su tiempo para detectar errores en el software y así poder informar de éstos para que los desarrolladores los corrijan). En total fueron 58 estudiantes de los semestres cuarto al décimo quienes probaron el juego y a quienes luego de interactuar con él se les solicitó llenar la encuesta que se presenta a continuación.

Nombre	Nivel	Puntuación
Yosel Del Valle	6	770
Daniel Correa Botero	6	635
Diego Luis Garavito	6	575
Sebastian Gomez	6	525
Johnny Montoya Franco	6	435
Juan Pablo Palacio Isaza	5	335
Tatiana Chamat Cujia	5	305
Carlos Andres	4	240
Willington Vega	4	225
Edward Munoz	3	220
Andres Felipe Sossa	3	215
Sebastian Munera Alvarez	4	185
Juan Camilo Osorio	4	120
Carolina Rincon Lopez	3	120
Juan Jose Perez	3	105
Bryan Zapata Ceballos	3	100
Henry Alberto Pineros	2	90
Cristian David Mazo	2	70
David Saldana	2	60
Katerine Villamizar Suaza	2	45
Elizabeth Toro	1	45
Nathalia Meneses Piedrahita	2	35
Sonia Maria Berrio Valencia	1	35
Andres Felipe Gonzalez	2	30
Andres De Leon	2	25

Tabla 1. Listado de puntuaciones

Pregunta	Calificación
1. De manera general, ¿Cómo calificaría el juego?	1-10
2. Desde el punto de vista académico, ¿Qué calificación le daría al juego como instrumento para fortalecer los conceptos de secuenciación e iteración definida?	1-10
3. Desde el punto de vista de gráficos, animación, sonidos, efectos, etc., ¿Cómo calificaría el juego?	1-10
4. Sobre a la interfaz del juego, ¿Qué calificación le daría respecto a la facilidad de su manejo, su claridad, etc.?	1-10
5. Le pareció adecuado el grado de dificultad de los niveles?	Si/No
6. Si tiene algún comentario sobre el modo de juego, la interfaz, los niveles, los personajes, el diseño o cualquier otro aspecto del juego, por favor escríbalo a continuación.	Libre

Tabla 2. Encuesta

La figura 9 muestra una fotografía de algunos de los estudiantes mientras probaban el juego, mientras que la tabla 2 presenta el resumen de la encuesta.



Figura 9. Estudiantes durante la prueba

Pregunta	Promedio	Desviación estándar
1	8.12	1.41
2	8.56	1.57
3	8.67	1.33
4	7.67	2.25

Tabla 3. Resultados de la encuesta

En la pregunta 5 la totalidad de los encuestados respondieron afirmativamente, mientras que el comentario más común en la pregunta 6 tuvo que ver con la finalización de cada nivel puesto que el juego no muestra el puntaje acumulado si no solamente la clave para almacenarlo junto con el nivel alcanzado. Aunque la gran mayoría de los estudiantes pudieron jugar intuitivamente sin mayores explicaciones previas, el segundo comentario más común se refirió a la falta de instrucciones al inicio del juego (esta funcionalidad no estaba lista al momento de la prueba). Ambos aspectos fueron considerados en una nueva versión del juego.

CONCLUSIONES

El uso de videojuegos educativos ha ido tomando fuerza en las últimas décadas e incluso hay numerosos trabajos que validan su utilización en aulas de clase. En este contexto, el trabajo presentado en este artículo es un aporte importante en cuanto apunta a un dominio de conocimiento específico como es la lógica de programación en el cual no existen muchos desarrollos y menos aún con las características didácticas y emotivas con las que cuenta ProBot.

En particular, el juego explora los conceptos de secuenciación, iteración definida y anidación, haciéndolo de una manera amena y desafiante.

La prueba realizada con estudiantes que ya habían tomado el curso de fundamentos de programación en la Universidad Nacional de Colombia dio cuenta del interés que despierta el juego no sólo por su atractivo visual y auditivo si no también por su filosofía de niveles con incremento de dificultad. Igualmente validó su utilidad para afianzar conocimientos en el dominio de conocimiento para el que fue creado y para incentivar el análisis de la eficiencia de las soluciones propuestas. Un descubrimiento adicional con esta prueba fue el juego incentiva el desarrollo de otras habilidades cognitivas generales aparte de la lógica y el razonamiento abstracto como es la memoria. Esto salió a la luz al observar que muchos estudiantes preferían memorizar los movimientos de los oponentes en vez de escribirlos para posteriormente traducirlos en las instrucciones requeridas correspondientes.

Como trabajo futuro inmediato es necesario volver a probar el juego, pero esta vez con estudiantes que estén en un curso de programación, esto con el fin de validar de una manera realista el impacto que su proceso de aprendizaje tendría. En particular, sería de interés analizar como ProBot puede ayudar a estudiantes de rendimiento bajo para afianzar los conceptos explorados.

REFERENCIAS

- [1] Antonakos, J; Mansfield, K. (1998) Programacion Estructurada En C. Pearson Educación.
- [2] Baker, A., Navarro, E., & van der Hoek, A. (2005). An experimental card game for teaching software engineering processes. *The Journal of Systems and Software*, 75, 3–16.
- [3] Becker, K. (2001). Teaching with Games: The Minesweeper and Asteroids Experience. *Journal of Computing in Small Colleges*, 17(2), 22–32
- [4] Bergman, P. (2003). Digital Games and Learning: A Research Overview. Unpublished manuscript.
- [5] Cavallari, J., Hedberg, J., & Harper, B. (1992). Adventure games in education: A review. *Australian Journal of Educational. Technology*, 8(2), 172–184.
- [6] De Freitas, S. (2005). Learning through Play. Using educational games and simulations to support post-16

- learners. London: London Learning and Skills Research Centre.
- [7] Dempsey, J. V., Rasmussen, K., & Lucassen, B. (1996). *The Instructional Gaming Literature: Implications and 99 Sources*. University of South Alabama.
- [8] Egenfeldt-Nielsen, S. (2006). Overview of research on the educational use of video games. *Digital Kompetanse*, Volume: 1 Issue: 3 Pages: 184-213.
- [9] Gander, S. (2002). Does Learning Occur through gaming. *Electronic Journal of Instructional Science and Technology*, 3(2).
- [10] Gorriz, C. M. and Medina, C. "Engaging Girls with Computers through Software Games", *Communications of the ACM*, Vol.43 #1 Jan. 2000 p42.
- [11] James, J., Beaton, B., Csete, J., Vogel, D. (2003). Mobile educational games. In: Lassner, D., McNaught, C. (eds.) *Proceedings of ED-MEDIA 2003*, pp. 801–802.
- [12] Kirriemuir, J., & McFarlane, A. (2002). *The use of computer games in the classroom*. Coventry: Becta.
- [13] Klopfer, E., & Yoon, S. (2005). Developing Games and Simulations for Today and Tomorrow's Tech Savvy Youth *TechTrends*. *Linking Research & Practice to Improve Learning*, 49(3), pp. 33-41.
- [14] Lawrence, R. (2006). Teaching data structures using competitive games. *IEEE Transactions on Education*, 49(1), 459–466.
- [15] Lewandowski, F., Soares, A. (2008). Desenvolvimento de Jogos Educacionais Apoiados por um Agente Tutor Pedagógico. En J. Sánchez (Ed.): *Nuevas Ideas en Informática Educativa*, Volumen 4, pp. 7-14, Santiago de Chile.
- [16] McGrenere, J. L. (1996). *Design of Educational Electronic Multi-player Games: A literature Review*. Vancouver: Department of Computer Science.
- [17] Mitchell, A., & Savill-Smith, C. (2004). *The use of computer and video games for learning: A review of the literature*. London: Ultralab: Learning and Skills Development Agency.
- [18] Moser, R. (1997) A Fantasy Adventure Game as a Learning Environment: Why Learning to Program is so Difficult and What can be Done about it. *Proceedings of the Conference on Integrating Technology in Computer Science Education*, June 1–5, Uppsala, Sweden p114–116.
- [19] Zapata, C. M. (2009). Teaching software development by means of a classroom game: The software development game. *Developments in Business Simulation and Experiential Learning*, Volume 36.