

Diseño e Implementación de un Taller de Programación de Juegos Digitales con Scratch como Apoyo a Fundamentos de Programación

Roberto Muñoz^{1,3}, Thiago Barcelos^{2,4}, Rodolfo Villarroel³, Ismar Silveira⁴

¹ Universidad de Valparaíso
General Cruz, 222, Valparaíso, Chile – CP 2362905

² Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP
Av. Salgado Filho, 3501, Guarulhos, SP, Brasil – CEP 07115-000

³ Pontificia Universidad Católica de Valparaíso
Brasil, 2241, Valparaíso, Chile – CP 2362807

⁴ Universidade Presbiteriana Mackenzie
Rua da Consolação, 930 – São Paulo, SP, Brasil – CEP 01302-907

Abstract. *Skills related to the fundamentals of programming should be developed by all students in the first semesters of undergraduate courses related to Computing. However, high attrition rates are also identified in this period. Didactic activities in which students develop their own digital artifacts, such as digital games, may increase the students' motivation to learn and apply programming skills in a more tangible way. In this article we describe the development and application of a game design workshop to support a Programming Fundamentals course aimed at students of the Informatics Engineering career in a Chilean university. The results indicate a considerable increase in the approval rates of students that were enrolled in the course for the first time.*

Resumen. *Habilidades relacionadas con los fundamentos de la programación deben ser desarrollados por todos los estudiantes de los primeros semestres de cursos relacionados con la informática. Sin embargo, las altas tasas de deserción también se identifican en este periodo. Actividades didácticas en las que los estudiantes desarrollen sus propios artefactos digitales, tales como los juegos digitales, pueden aumentar la motivación de los estudiantes para aprender y aplicar conocimientos de programación de una manera más tangible. En este artículo se describe el desarrollo y aplicación de un taller de diseño de juego para apoyar un curso de Fundamentos de programación dirigida a los estudiantes de la carrera de Ingeniería Informática en una universidad chilena. Los resultados indican un aumento considerable en las tasas de aprobación de los estudiantes que se inscribieron en el curso por primera vez.*

1. Introducción

A nivel mundial, en los últimos años, se ha visualizado un déficit de profesionales en el área de las Tecnologías y Comunicaciones (TIC) [“Expectativas del profesional del futuro” 2014, “Occupations with the most job growth, 2012 and projected 2022” [S.d.]]. Por otra parte, estudios demuestran el interés por estudiar carreras asociadas a las TIC ha ido creciendo [Muratet et al. 2009], [Crenshaw et al. 2008]. Si bien en Chile, existe una alta demanda por carreras asociadas a las TIC, logrando llenar sus vacantes de manera temprana, cuando se analizan sus estadísticas, en lo que respecta a retención y titulación oportuna, se puede inferir alguna de las causales por la escasez de profesionales.

De acuerdo las estadísticas nacionales de Chile, en cuanto a lo que respecta a la tasa de retención de Ingeniería Civil Informática, para primer año es de un 67,7%, y para el segundo año es de un 49,9 %. En el caso de la Escuela de Ingeniería Civil Informática de la Universidad de Valparaíso, la tasa de retención para primer año, cohorte 2013, fue de un 57,3%, mientras que la tasa de titulación oportuna es tan sólo de un 5,9%.

Esta situación ha motivado acciones por parte del cuerpo académico para disminuir las cifras de deserción, como por ejemplo, la continua actualización del programa de estudios de la carrera, y la reorganización y redistribución de los contenidos entre distintas asignaturas. En esta línea, una iniciativa importante es la incorporación de nuevas herramientas de aprendizaje, especialmente enfocadas en los estudiantes de primer año [Muñoz et al. 2012].

El currículum estándar para cursos de pregrado de computación, de acuerdo a la ACM y la IEEE, define que el dominio de los fundamentos de programación de computadores es una habilidad esencial a ser desarrollada por los estudiantes [ACM-IEEE Software Engineering 2008 2008]. Los estudiantes de Ingeniería Civil Informática en la Universidad de Valparaíso reciben su formación en programación a través de tres cursos: inicialmente, uno de Fundamentos de Programación, que aborda tipos de datos, estructuras de control, y la definición y uso de algoritmos para la resolución de problemas. Posteriormente, y previa aprobación de Fundamentos de Programación (FP), los estudiantes cursan Programación I (P1), en el que se estudia el paradigma imperativo de programación, usando el lenguaje C. A Programación II, donde se estudia el paradigma de orientación a objetos, llega un número considerablemente menor de estudiantes que los que inician Fundamentos de Programación, esto debido a la alta tasa de deserción.

Con base a lo antes expuesto, este trabajo tuvo como objetivo diseñar e implementar un taller de programación de juegos digitales con Scratch como apoyo a la asignatura de fundamentos de programación, el cual se basa en principios del constructivismo y en aprendizaje basado en problemas (ABP). Los resultados muestran un aumento significativo en lo que respecta a las calificaciones que aquellos que asisten con una frecuencia alta al taller en su totalidad aprueban la asignatura de Fundamentos de Programación.

2. Trabajos Relacionados

La construcción de artefactos computacionales por los estudiantes es una estrategia didáctica para el desarrollo de habilidades informáticas que viene siendo frecuentemente

utilizada. Además de relacionarse con “hacer computación”, propuesto por [Hu 2011], esta estrategia revisita la clásica propuesta de constructivismo de Papert [1980]. Actualmente, las posibilidades de actividades que involucren construcción de artefactos computacionales se amplían debido a una mayor disponibilidad de ambientes de desarrollo orientados al usuario principiante, tales como Scratch (MIT MEDIA LAB, 2012), Alice [Carnegie Mellon University 2013], Greenfoot [University of Kent 2013], GameMaker [YoYo Games, Ltd. 2014] y Kodu Game Lab [Microsoft Research 2014]. Tales ambientes permiten el desarrollo de juego, animaciones e historias interactivas con poca o ni una necesidad de sintaxis de lenguajes de programación.

El desarrollo de juegos digitales viene siendo evidenciado como una alternativa promisoría en ese contexto. Los juegos digitales están insertos en el contexto cultural de los nativos digitales. De tal forma Peppler e Kafai [2009] proponen que los estudiantes de nuevas generaciones deberían desarrollar una fluidez en los juegos, es decir una mayor comprensión de sus características como medio digital, por ejemplo, su mecánica y los procedimientos necesarios para su construcción. Algunos relatos de la evaluación de actividades de creación de juegos digitales por estudiantes [Denner et al. 2012; Maloney et al. 2008] evidencian que la necesidad de incorporar aspectos de interacción en los juegos puede haber contribuido para un uso más intenso de algunas estructuras de programación, o el mismo aprendizaje de algunos conceptos por parte de los estudiantes de manera autónoma, es decir, sin supervisión de instructores.

Para finalizar, es importante mencionar que en [Rizvi et al. 2011], se utiliza Scratch en un curso de Ciencias de la Computación introductorio con el fin de mejorar la retención de estudiantes de alto riesgo, resultando en un aumento sustancial en la matrícula con respecto a años anteriores.

3. Estructura del Taller

La estructura constó con 12 sesiones, en la que cada encuentro duró una hora y media. En cada encuentro el docente propuso la construcción de uno o más mecanismos de interacción relacionados con la construcción de un juego digital. Para esto, los estudiantes fueron invitados a explorar conceptos relacionados con desarrollo de juegos (animación de sprites, colisión, controles por teclado y mouse) y de Fundamentos de Programación (variables, mensajes, estructuras condicionales, estructuras repetitivas). A partir de los tópicos a ser cubiertos en la asignatura de FP, se definieron cuatro directrices:

1. La construcción de juegos debe motivar el desarrollo de todas las actividades del taller. Esta directriz se basa en el principio de fluidez de juegos puesto por Peppler y Kafai [2009], que argumentan que el proceso de diseño y construcción de artefactos en el contexto cultural de los juegos digitales puede promover un proceso de reflexión y aprendizaje para los estudiantes.
2. Las actividades deben progresivamente llevar a la construcción de la mecánica de un juego completo. En trabajos anteriores que discuten el desarrollo y conducción de actividades prácticas con el objetivo de fomentar el desarrollo del Pensamiento Computacional. Lee et al. [2011] proponen el esquema Usar – Modificar – Crear, en el cual este principio se basa. Inicialmente a los estudiantes se les presentan programas listos para interactuar y comprender su funcionamiento, a partir de la comprensión obtenida en esa primera etapa, los

estudiantes son invitados a introducir modificaciones en su funcionamiento y su apariencia. Finalmente, a medida que los estudiantes adquieren mayor confianza, comienzan a crear sus propios juegos, aplicando los conocimientos adquiridos en etapas anteriores. Cabe destacar que esta estrategia no es aplicada de forma lineal. Un estudiante puede actuar como creador de una etapa del proceso de aprendizaje y en una etapa siguiente, volver a actuar como “usuario” para comprender un nuevo concepto. Sin embargo, la estructura de las actividades y los conocimientos y habilidades a ser movilizados inducen al estudiante a actuar cada vez como “creador” a lo largo del taller. La estrategia todavía tiene impacto en la mantención de un nivel adecuado de desafío.

3. Las actividades deben progresivamente demandar que nuevos conceptos sean explorados por los estudiantes, al mismo tiempo, solicitar que el estudiante utilice nuevamente conceptos explorados anteriormente. La introducción de nuevos conceptos de forma paulatina en las actividades (a través del esquema Usar - Modificar - Crear) tiene como objetivo introducir continuamente nuevos desafíos que induzcan al estudiante a buscar nuevos conocimientos. El proponer nuevos desafíos al mismo tiempo en que conceptos ya explorados son requeridos nuevamente, tiene como objetivo que los estudiantes estén en un estado de flujo en lo que se refiere a su trabajo autónomo en la construcción de juegos [Nakamura and Csikszentmihalyi 2009]. De la misma forma, se espera que el profesor pueda actuar como facilitador en los momentos en que los estudiantes presenten dificultades puntuales relacionadas, por ejemplo, a un concepto específico. En estos casos, se espera que la secuencia planeada de las actividades mantenga a los estudiantes en la zona de desarrollo proximal [Vygotsky 1978], permitiendo así su avance en las actividades.
4. La mecánica de los juegos, a pesar de ser simples, debe traer referencia al universo de los juegos “reales” para que sean significativas para los estudiantes. Alineados con una propuesta constructivista, en que la construcción de artefactos digitales por los estudiantes demanda postular de una forma razonablemente autónoma, las actividades del taller siguen el abordaje de aprendizaje basado en problemas (ABP) [2002]. Según Merrill [2002], el aprendizaje basado en problemas es una estrategia centrada en el estudiante, que trabaja de manera colaborativa en la solución de algún problema, en la cual la figura del profesor sirve como apoyo y la construcción del conocimiento es gradual y empírica. De esta forma, en cada actividad del taller, los estudiantes reciben instrucciones sobre los objetivos propuestos para el juego; además de eso, es presentado un ejemplo del juego propuesto siendo ejecutado y a partir de ahí inician el trabajo. El profesor actúa como un facilitador observando el trabajo e interviniendo a medida que los estudiantes solicitan mayores apoyos.

En las primeras semanas del taller, las actividades son relacionadas con la utilización del entorno Scratch y para reforzar los conceptos vistos en la cátedra de FP. Los juegos propuestos son: Piedra Papel y Tijera, Simulación de Guerra y prototipos de los famosos juegos Breakout y Pacman. Para finalizar los estudiantes debían presentar su versión del juego Aero Fighters.

En la Tabla 1, se presenta la programación de actividades del taller de programación de juegos con Scratch.

Tabla 1. Actividades taller de programación

Sesión	Actividades / contenido
1	Familiarización con el ambiente <i>Scratch</i> (conceptos <i>sprite</i> y colisión entre <i>sprites</i>)
2	Variables y estructuras repetitivas
3	Estructuras repetitivas y estructuras condicionales
4	Crear juego Piedra-Papel-Tijera
5-6	Crear el juego Simulación de Guerra
7-8	Crear el juego Breakout
9	Pacman – Crear la mecánica básica de los movimientos de los personajes
10-11	Pacman – Implementar las demás características del juego final
12	Presentación del proyecto Final (Libre)

4. Perfil de los Participantes

Se realizó un cuestionario disponible en [“Encuesta de Percepción de Juegos Digitales Survey” [S.d.]] con el fin de determinar el perfil de los participantes y su relación con juegos digitales.

Los participantes casi en su totalidad (78%) correspondieron a estudiantes entre 18 y 19 años. Sólo el 22% de ellos tenían más de 19 años (min: 20 años, máx: 25 años, desviación standard de 1.64). En lo que respecta al sexo, sólo el 12,2% correspondió a femenino y el 87,8% restante a masculino. El 26,8% de los estudiantes provenía de establecimientos municipales (público), el 68,3% de establecimientos particulares subvencionados (financiamiento mixto), y tan sólo el 4,9% de los participantes provenía de colegios particulares (pagados). En lo que respecta a su procedencia, el 63,4% provenía de la región en que la Universidad de Valparaíso se encuentra establecida y el 36,6% de otra región del país.

En lo que respecta a su experiencia en juegos digitales, el 87,8% de ellos declara tener experiencia en juegos digitales, mientras que el 12,20% declara que no. Un elemento a considerar es que de ese 12,2% que declara no tener experiencia en juegos digitales el 7,3% corresponden a mujeres. Este es un elemento que se deberá considerar en una próxima iteración del taller de programación de juegos digitales.

Los estudiantes tenían experiencia en juegos digitales, declaran el 16,67% jugar menos de una hora diaria, el 36,1% entre una y tres horas diarias, el 22,2% entre 3 y 5 horas diarias, el 19,4% entre 5 y 8 horas diarias, el 2,8% entre 8 a 12 horas diarias y 2,8% jugar más de 15 horas diarias.

Cabe mencionar que el 91,7% afirma jugar hace 3 años o más, concentrando un 58,3% a aquellos que juegan hace más de 9 años.

En lo que respecta al tipo de juego preferido las preferencias de los estudiantes se distribuyeron principalmente de la siguiente forma. El 33,3% de preferencias fue a *Role-Playing Games* (RPG), el 16,7% a Acción y Aventura, el 13,9% a *Multiplayer Online Battle Arena* (MOBA) y Shooter en primera persona (FPS) con Deportes, ambos

con un 11,1%. Cabe mencionar que los estudiantes tenían 10 opciones de categorías de juegos en las cuales podían marcar sus preferencias.

5. Resultados Preliminares

El taller se ofreció a los estudiantes durante el primer semestre del año 2015. A todos los estudiantes de primer año matriculados en Fundamentos de programación se les permitió participar en el taller, sin embargo, su participación no fue obligatoria. El taller fue considerado como una actividad extra en el programa del curso. Sin embargo, se les solicitó la elaboración de un video juego al término de taller.

El videojuego debía contar con una introducción que era una animación creada por los estudiantes, un menú inicial el cual debía contar con 2 botones, uno para iniciar la partida y uno para modificar la dificultad del juego (en base a la dificultad seleccionada se debía provocar cambios en el juego ej: disminuir o aumentar la cantidad de enemigos en base al nivel), y 5 niveles jugables. Además, el usuario debía contar con 3 vidas para avanzar durante el juego, tras llegar a 0 vidas se debía producir el fin del juego y debía comenzar todos los niveles nuevamente. Para finalizar, el fondo del juego debía tener un scroll a medida que avanzara con el personaje controlado por el jugador (emulando una resolución de pantalla de 960px. o más). Ejemplos de los trabajos realizados por los estudiantes, son presentados en las figuras 1 y 2.



Figura 1. Ejemplo Juego Realizado Alumno 1

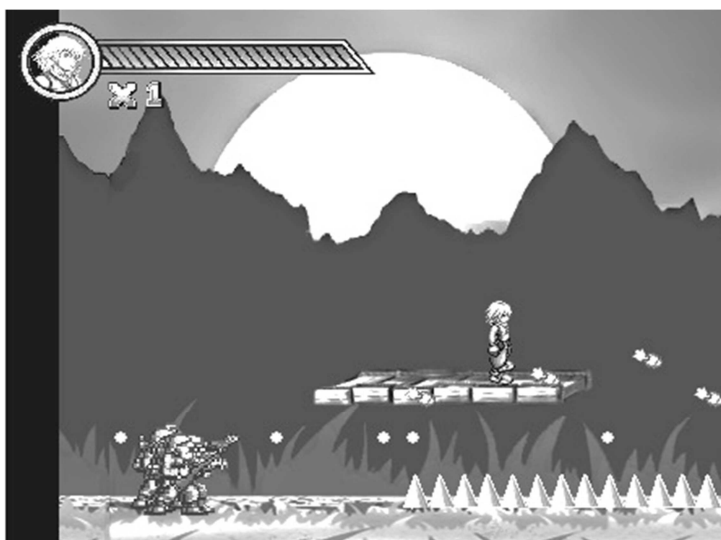


Figura 2. Ejemplo Juego Realizado Alumno 2

Para finalizar, en la figura 3, se puede visualizar la relación entre la asistencia al taller en contraste con el estado final en la asignatura.

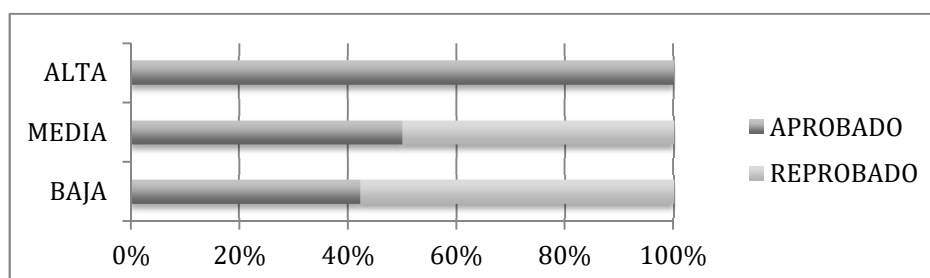


Figura 3. Relación asistencia taller vs estado

En una primera instancia, se puede visualizar que aquellos que poseen una asistencia alta (sobre el 66%) al taller en su totalidad aprobaron el curso de Fundamentos de Programación. Sin embargo el resultado no es claro para aquellos que poseen una asistencia baja (menor al 33%) y media (entre 33% y 66%). Es por esto es que se decidió determinar si existe o no una correlación entre el promedio obtenido en el taller y su asistencia. En la Figura 4 se puede visualizar un gráfico de cajas con el promedio, la desviación estándar en cada uno de los grupos de estudiantes mencionados anteriormente.

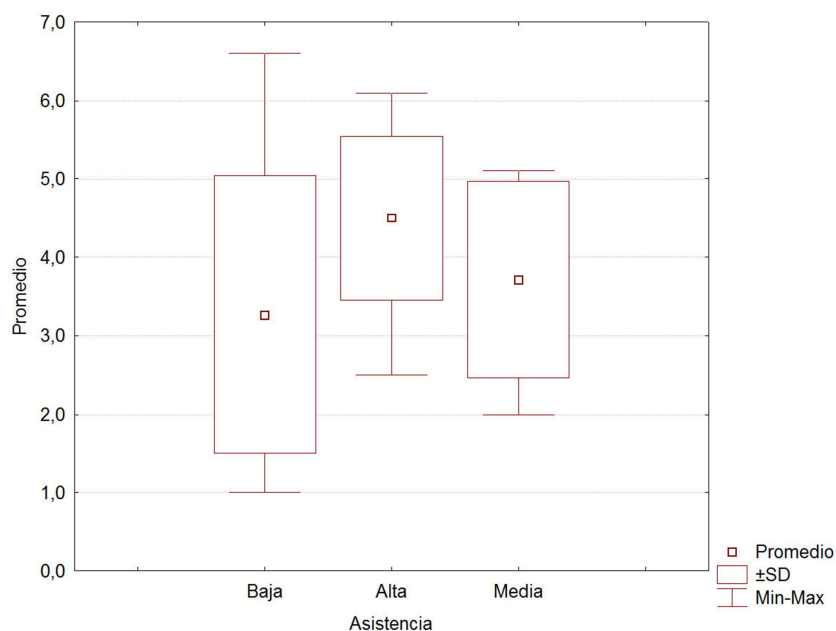


Figura 4. Distribución de grados por nivel de asistencia a el taller.

Es posible visualizar una correlación entre la nota promedio de los estudiantes que asisten al curso de Fundamentos de Programación y el nivel de asistencia al Taller, así como una variación menor de las notas a medida que aumenta el nivel de asistencia al taller. A pesar de que el desempeño de los estudiantes podría estar relacionado con un mayor interés y compromiso global por el curso, es que se requiere continuar con la investigación. Sin embargo la correlación identificada anteriormente puede indicar una influencia positiva de las actividades de construcción de juegos en el desempeño académico de los estudiantes.

Por otra parte, si los resultados se comparan con el porcentaje de aprobación con el de otros años, los resultados si son alentadores. Esto debido a que en la segunda aplicación del taller, el aumento en las tasas de aprobación continúa en aumento. Llegando en esta ocasión a un 51% en contraste con el 34% y 40% obtenidos en el en los años 2012 y 2014 respectivamente.

6. Conclusiones

Adquirir los conceptos fundamentales de la programación de computadores es una habilidad crítica para los estudiantes de cursos relacionados con la informática. Sin embargo, las asignaturas relacionadas con este tema suelen tener altas tasas de deserción. Estrategias didácticas orientadas a la aplicación de los conceptos de programación abstracta como la construcción de juegos, pueden ser útiles en este contexto. Partiendo de esta premisa, los autores desarrollaron un taller de programación de juegos digitales con Scratch para apoyar y complementar la asignatura inicial de Fundamentos de Programación. Si bien la estructura del taller debe refinarse, los resultados son bastante alentadores. En lo que respecta al trabajo a corto plazo, se espera realizar un análisis profundo a los artefactos creados por los estudiantes, y así obtener métricas tales como número de funcionalidades adicionales, aspectos de usabilidad incorporados, diferencias entre género, entre otros. Como trabajo futuro se espera incorporar la programación de dispositivos móviles debido al alto interés y penetración

que éstos tienen en los adolescentes. También se espera poder replicar este taller en otras carreras de la facultad que incorporen Fundamentos de Programación en sus mallas curriculares.

7. Agradecimientos

Roberto Muñoz es beneficiario de la Beca de Doctorado INF-PUCV 2015.

8. Referencias

ACM-IEEE Software Engineering 2008 (2008). Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. IEEE Computer Society and Association for Computing Machinery.

Carnegie Mellon University (2013). Alice - An educational software that teaches students computer programming in a 3D environment. <http://www.alice.org>, [accessed on Aug 15].

Crenshaw, T. L., Chambers, E. W., Metcalf, H. and Thakkar, U. (2008). A case study of retention practices at the University of Illinois at Urbana-Champaign. In *Proc. SIGCSE 2008*. . ACM.

Denner, J., Werner, L. and Ortiz, E. (jan 2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, v. 58, n. 1, p. 240–249.

Encuesta de Percepción de Juegos Digitales Survey ([S.d.]). <https://es.surveymonkey.com/r/VXSWY7S>, [accessed on May 24].

Expectativas del profesional del futuro (2014). . Asociación Chilena de Empresas de Tecnologías de Información - AG. http://www.acti.cl/files/AIEP_ACTI_ESTUDIO_PROFESIONALES_TIC.pdf, [accessed on May 17].

Hu, C. (2011). Computational thinking: what it might mean and what we might do about it. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*. . ACM. <http://doi.acm.org/10.1145/1999747.1999811>.

Lee, I., Martin, F., Denner, J., et al. (feb 2011). Computational thinking for youth in practice. *ACM Inroads*, v. 2, n. 1, p. 32–37.

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M. and Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. , SIGCSE '08. ACM. <http://doi.acm.org/10.1145/1352135.1352260>.

Merril, D. (2002). A Pebble-in-the-Pond Model For Instructional Design. *Performance Improvement*, 7. v. 41, p. 41–46.

Microsoft Research (2014). Kodu Game Lab Community. <http://www.kodugamelab.com/>, [accessed on Jan 16].

MIT Media Lab, Lifelong Kindergarten Group (2012). Scratch. <http://scratch.mit.edu>, [accessed on Apr 27].

Muñoz, R., Noël, R., Barría, M. and Pérez, F. (2012). Un Taller de Robótica para el Apoyo de la Enseñanza de Programación de Computadores Basado en Estilos de Aprendizaje. In *XVII Congreso Internacional de Informática Educativa 2012*.

Muratet, M., Torguet, P., Jessel, J.-P. and Viallet, F. (2009). Towards a Serious Game to Help Students Learn Computer Programming. *International Journal of Computer Games Technology*, v. 2009, p. 1–12.

Nakamura, J. and Csikszentmihalyi, M. (2009). Flow theory and research. In: Snyder, C. R.; Lopez, S. J.[Eds.]. *Oxford Handbook of Positive Psychology*. 2. ed. Oxford: Oxford University Press. p. 195–206.

Occupations with the most job growth, 2012 and projected 2022 ([S.d.]). <http://www.bls.gov/news.release/ecopro.t05.htm>, [accessed on May 17].

Papert, S. (1980). *Mindstorms: children, computers and powerful ideas*. New York: Basic Books.

Peppler, K. and Kafai, Y. (1 nov 2009). Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio. *International Journal of Learning and Media*, v. 1, n. 4, p. 45–58.

Rizvi, M., Humphries, T., Major, D., Jones, M. and Lauzun, H. (jan 2011). A CS0 course using Scratch. *J. Comput. Sci. Coll.*, v. 26, n. 3, p. 19–27.

University of Kent (2013). Greenfoot. <http://www.greenfoot.org>, [accessed on Aug 15].

Vygotsky, L. S. (1978). Zone of Proximal Development. In: Cole, M.; John-Steiner, V.; Scribner, S.; Souberman, E.[Eds.]. . *Mind in society: The development of higher psychological processes*. Oxford: Harvard University Press. p. 52–91.

YoYo Games, Ltd. (2014). GameMaker: Studio. <https://www.yoyogames.com/studio>, [accessed on Jan 16].