

Videojuego educativo para enseñar algoritmos relacionados a grafos

Alonso Utreras

Profesor Guía: Iván Sipirán

Departamento de Ciencias de la Computación

Universidad de Chile

Santiago, Chile

`alonso.utreras@ug.uchile.cl`

`autreras@dcc.uchile.cl`

1 Introducción

Una forma común de enseñar y aprender es acompañar la información con dibujos o gráficos. La presente investigación busca llevar las instrucciones llevadas a cabo por los algoritmos relacionados a grafos a una representación visual e interactiva para que los estudiantes de computación puedan aprender acompañándose con esta herramienta.

El trabajo consiste en presentarle un videojuego donde se muestren grafos, en que el jugador debe ejecutar las instrucciones de estos algoritmos, apoyándose en elementos visuales. El objetivo de esta investigación de tesis es determinar si se ven diferencias en los niveles de motivación percibidos por el estudiantado y en el entendimiento de los algoritmos, medido a través de tareas de programación. El público objetivo son estudiantes que estén en su primer año de ciencias de la computación (CS por sus siglas en inglés).

2 Trabajo relacionado

Un videojuego es una forma de aprendizaje activo, pues el proceso de enseñanza no se desarrolla partiendo por un profesor exponiendo frente a un estudiante. En este caso, quien aprende debe ejecutar pasos y participar en alguna actividad a través de la cual se construye el conocimiento. En la reseña realizada por Hartikainen et al. [13] se enumeran justificaciones para el aprendizaje activo: mejores resultados, recomendaciones políticas y las nuevas demandas de la vida laboral actual, como capacidades de comunicación o descubrimiento por cuenta propia.

Los videojuegos serios (Serious Gaming) son una forma de aprendizaje activo. Un trabajo de Bell y Gibson [11] indican que juegos educativos son más efectivos que las clases, lecturas, videos y tareas. Se resume que el uso de juegos resulta en 9% mejor retención, 11% mejor conocimiento fáctico, 14% mejor conocimiento basado en habilidades y 20% mayor autoeficacia. Sin embargo, recomiendan que los juegos deben ser acompañados de otras formas de enseñanza, así como hacer actividades post juego donde se le pregunta al estudiantado cómo se relacionan los juegos a la materia.

Bell y Gibson [11] identificaron y clasificaron videosjuegos de Ciencias de la Computación (CS), considerando un total de 41 videojuegos. Uno de ellos, Map Coloring, se trata sobre grafos, tomando el tema de coloreo de grafos. Se realizó una búsqueda del juego a la fecha (2022), pero no se encontró ningún material al respecto.

Kiili y su equipo [14] analizaron el uso de videojuegos en enseñanza y evaluación en matemáticas a través de los títulos “Semideus” y “Wuzzit Trouble”. A través de estos, llegaron a la conclusión de que es posible utilizar videojuegos para enseñar y evaluar al mismo tiempo. Además, caracterizaron estadísticamente las diferencias producidas por el uso de videojuegos entre resultados de un pre test y un post test.

En el trabajo realizado por Zhao y Shute [23] se enumeran ejemplos de videojuegos pensados para enseñar programación, tales como Wu’s Castle [6], CodeCombat [1], CodeSpell [7], MiniColon [3], tales ejemplos utilizan programación con texto. Sin embargo, también hay numerosos ejemplos que utilizan programación por bloques, como LightBot [12], Scratch [15], [2] y RoboBuilder [20], los cuales abstraen el trabajo de aprender una sintaxis relacionada a los lenguajes de programación. Sin embargo, el impacto de estos videojuegos no ha sido evaluado en muchos casos. En los casos documentados, se cuenta con muestras pequeñas, además de que se trata principalmente de evaluaciones puramente cualitativas [23], [10].

Kiili y su equipo, Bell et al., Giani y otros [18] están de acuerdo en que no hay sistematización en la evaluación de videojuegos educativos. En efecto, hay juegos serios que ni siquiera se denominan o consideran como tales [11]. Por otra parte, no existe una forma estándar de evaluarlos, razón por la cual Petri, Giani y otros [18] crearon el modelo MEEGA+ (Model for the Evaluation of Educational Games and EGameFlow scale).

Entre las faltas mencionadas al momento de crear videojuegos, se mencionan las faltas de 1) Definición de un objetivo de evaluación; 2) Diseño de investigación; 3) Programa de medición; 4) Instrumentos de recolección de datos y 5) Métodos de análisis de datos. Un ejemplo de falta de sistematización: una práctica común al analizar estas herramientas son los comentarios informales por parte del estudiantado [18].

2.1 Propiedades que debería tener el juego

Según Gibson y Bell [11], el juego debería ser divertido, acotado en tiempo y espacio (separado), incierto (Su desarrollo no está predeterminado), gobernado por reglas y ficticio. Además, debería generar inmersión y enganche en un jugador. Por otra parte, se recomiendan ciertos lineamientos como innovación y escalamiento en la dificultad, lo cual permite generar inmersión (en inglés y en la literatura más conocido como flow), esto le permite a un jugador generar confianza en un ambiente libre de riesgos para experimentar y poner a prueba sus conocimientos.

Un problema del aprendizaje activo es que, pese a lograr mejores resultados, incomoda a los estudiantes, pues les genera la percepción de que su aprendizaje fue menor con respecto a actividades de enseñanza pasiva [5]. Los autores de este trabajo atribuyen tales resultados al sentimiento de desconocimiento que sienten los alumnos mientras realizan estas actividades, ahunado a que, al no estar familiarizados con aprendizaje activo intenso en clases, no comprenden que el esfuerzo cognitivo implicado es un signo de aprendizaje efectivo.

Sin embargo, un videojuego que entregue retroalimentación constante al jugador de que sus acciones son correctas o incorrectas, sumado a que cada estudiante estaría realizando las actividades de forma individual, dejaría de exponer sus errores y quitaría esta incomodidad para los estudiantes, sobretodo si el juego logra la inmersión del jugador, haciendo que este olvide que está aprendiendo.

El juego se acotará a los cuatro algoritmos: BFS, DFS, Kruskal y Prim relacionados a grafos a través de distintos niveles. El jugador podrá elegir partir por BFS o DFS. Una vez los haya ejecutado correctamente, podrá acceder a los niveles para resolver distancias en grafos con Kruskal y Prim.

3 Problema

Autores como Zehetmeier et al. [22], Ghezzi et al. [9] y Wing [21] afirman que el pensamiento abstracto es una de las habilidades fundamentales del pensamiento computacional y para las ciencias de la computación. Se observa a nivel general que hay estudiantes que ven muy difícil el abstraer problemas o la forma de una solución. Ante esto, han surgido intentos como Scratch [2], Snap! o Blockly [8], que buscan funcionar como herramientas de programación visuales para facilitar el aprendizaje y facilitar el proceso de abstracción [19].

McGonigal [16] y Barab et al. [4], indican que los videojuegos son una herramienta potencial para el aprendizaje y motivación a través de la inmersión en el videojuego.

4 Preguntas de Investigación

P1: ¿Cómo se comparan los resultados en un test de programación entre un grupo de control y otro que jugó un videojuego que enseña algoritmos relacionados a grafos?

P2: ¿Cómo se comparan los niveles de interés entre un grupo de control y otro que jugó un videojuego que enseña algoritmos relacionados a grafos?

5 Hipótesis

H1: Un videojuego que enseña algoritmos relacionados a grafos es una herramienta que mejora los resultados en un test de programación de los estudiantes de computación.

H2: Un videojuego que enseña algoritmos relacionados a grafos genera mayor interés en el estudiantado. Se entienden mejores resultados académicos como notas más elevadas en evaluaciones estándares realizadas por los equipos docentes a cargo del ramo.

Se entiende interés como una medida subjetiva de cuánto un usuario desea aprender sobre un tema en particular.

6 Objetivo principal

Diseñar y desarrollar un videojuego que muestre grafos y algoritmos relacionados a estos, que permita al usuario ir ejecutando acciones en la aplicación según lo pide un algoritmo desplegado en la misma pantalla. La aplicación debe indicar al usuario cuando está en lo correcto y cuándo se equivoca.

7 Objetivos específicos

- Diseñar una aplicación interactiva que muestre grafos y permita seguir los pasos relacionados a algoritmos que trabajen con grafos.
- Medir el interés de estudiantes de computación relacionado a grafos antes y después del uso de esta herramienta.
- Medir la comprensión y capacidad de replicación de un algoritmo relacionado a grafos de un grupo de estudiantes de computación antes y después de probar el videojuego presentado.

8 Metodología

Se desarrolla un videojuego considerando el feedback de un profesor guía y otros estudiantes que no sean parte de la población objetivo del estudio, ajustándolo según sea necesario.

Asegurarse que el videojuego cumple con las características seguidas por autores que proponen metodologías para la evaluación de videojuegos educativos: Gibson et al. [11], Petri, Giani y otros, [18] y Kiili et al. [14]. Si no, seguir iterando.

Cuando se haya impartido la materia relacionada a grafos y se hayan visto los algoritmos BFS y DFS, preguntarle a los alumnos si se creen capaces de programar tales algoritmos desde 0 y guardar estos resultados. Presentar el juego en los cursos cuando se esté impartiendo la materia relacionada a grafos. El uso del videojuego será voluntario y sugerido, con el requisito de llenar un formulario posterior al uso del juego. El formulario poseerá un cuestionario siguiendo los lineamientos de MEEGA+ planteados [17]

Se revisarán los datos llenados por los estudiantes que accedieron al formulario, en conjunto con los datos recolectados durante el juego mismo, para ver si estos se condicen. Por ejemplo, el juego medirá el tiempo entre clicks. Se espera que un jugador que esté motivado posea un alto nivel de actividad, y una alta cadencia de clicks.

Finalmente, se realizará una prueba de programación en los cursos que vean la materia de grafos, donde los estudiantes tendrán que responder preguntas relacionadas a los algoritmos vistos en el juego. Se separarán dos grupos, uno de control con la gente que decidió no jugar, y otro experimental que sí probó el videojuego.

El programa ofrecerá a los jugadores ejecutar tareas relacionadas con los algoritmos BFS, DFS, Kruskal y Prim relacionados a grafos. Para completar los desafíos presentados, el usuario deberá realizar las instrucciones dictadas por un computador, invirtiendo los roles de un programador y un computador. En este caso, el juego limita al jugador por medio de sus reglas, a seguir los pasos del algoritmo presentado. Si el jugador ejecuta correctamente las instrucciones, ganará, pudiendo pasar a otro nivel.

El videojuego guardará datos durante su ejecución y los enviará a alguna base de datos donde se almacenarán para su posterior análisis. Los datos que se guardarán serán: clicks, movimientos del mouse, acciones del teclado a lo largo del tiempo. Estos datos se usarán para contrastar con la información final y para determinar el grado de veracidad de las respuestas.

8.1 Desventajas y debilidades del proceso

El tamaño muestral puede no ser suficiente. No se sabe cuánto será el tamaño de los grupos de control y experimental, pero se espera que el total de participantes sea superior a 50.

Existe un sesgo al presentar el uso del videojuego como una opción voluntaria, por lo que los estudiantes con más tiempo e interés tenderán a probar el videojuego, pero son quienes a su vez tienen mejores notas.

9 Resultados esperados

Se espera observar un nivel de interés superior en los estudiantes con respecto a tópicos de algoritmos relacionados a grafos a través del cuestionario entregado. Se esperan mejores resultados en el grupo que pruebe el videojuego, en un intervalo del 8% al 14% de diferencia, lo que debe ser entre 5 y 7 décimas en una escala de evaluación del 1.0 al 7.0 considerando valores decimales intermedios. Además, se busca generar una metodología de prueba para herramientas interactivas cuyo fin sea la enseñanza, de tal

manera que otro memorista o tesista en el futuro pueda hacer pruebas análogas con otros algoritmos o estructuras de datos e incluso aplicadas a otras materias.

Por otra parte, se espera publicar un artículo en alguna revista como ACM o IEEE Education respecto a los resultados obtenidos.

References

- [1] *CodeCombat: Coding games to learn python and JavaScript*. <https://www.codecombat.com/>. Accessed: 2022-May-20.
- [2] *Scratch: Learn Programming with blocks*. <https://scratch.mit.edu/>. Accessed: 2022-May-20.
- [3] Reham Ayman, Nada Sharaf, Ghada Ahmed & Slim Abdennadher (2018): *MiniColon; teaching kids computational thinking using an interactive serious game*. In: *Joint International Conference on Serious Games*, Springer, pp. 79–90.
- [4] Sasha A Barab, Brianna Scott, Sinem Siyahhan, Robert Goldstone, Adam Ingram-Goble, Steven J Zuiker & Scott Warren (2009): *Transformational play as a curricular scaffold: Using videogames to support science education*. *Journal of Science Education and Technology* 18(4), pp. 305–320.
- [5] Louis Deslauriers, Logan S McCarty, Kelly Miller, Kristina Callaghan & Greg Kestin (2019): *Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom*. *Proceedings of the National Academy of Sciences* 116(39), pp. 19251–19257.
- [6] Michael Eagle & Tiffany Barnes (2009): *Experimental evaluation of an educational game for improved learning in introductory computing*. *ACM SIGCSE Bulletin* 41(1), pp. 321–325.
- [7] Sarah Esper, Stephen R Foster, William G Griswold, Carlos Herrera & Wyatt Snyder (2014): *CodeSpells: bridging educational language features with industry-standard languages*. In: *Proceedings of the 14th Koli calling international conference on computing education research*, pp. 05–14.
- [8] N Fraser (2013): *Blockly*. Available at <https://developers.google.com/blockly/>:Google.
- [9] Carlo Ghezzi, Mehdi Jazayeri & Dino Mandrioli (1991): *Fundamentals of software engineering*. Prentice-Hall, Inc.
- [10] Andreas Giannakoulas & Stelios Xinogalos (2018): *A pilot study on the effectiveness and acceptance of an educational game for teaching programming concepts to primary school students*. *Education and Information Technologies* 23(5), pp. 2029–2052.
- [11] Ben Gibson & Tim Bell (2013): *Evaluation of games for teaching computer science*. In: *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, pp. 51–60.
- [12] Lindsey Ann Gouws, Karen Bradshaw & Peter Wentworth (2013): *Computational thinking in educational activities: an evaluation of the educational game light-bot*. In: *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, pp. 10–15.
- [13] Susanna Hartikainen, Heta Rintala, Laura Pylväs & Petri Nokelainen (2019): *The Concept of Active Learning and the Measurement of Learning Outcomes: A Review of Research in Engineering Higher Education*. *Education Sciences* 9(4), doi:10.3390/educsci9040276. Available at <https://www.mdpi.com/2227-7102/9/4/276>.
- [14] Kristian Kiili, Keith Devlin, Arttu Perttula, Pauliina Tuomi & Antero Lindstedt (2015): *Using video games to combine learning and assessment in mathematics education*. *International Journal of Serious Games* 2(4), pp. 37–55.
- [15] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman & Evelyn Eastmond (2010): *The scratch programming language and environment*. *ACM Transactions on Computing Education (TOCE)* 10(4), pp. 1–15.
- [16] Jane McGonigal (2011): *Reality is broken: Why games make us better and how they can change the world*. Penguin.

- [17] Giani Petri, Christiane Gresse von Wangenheim & Adriano Ferreti Borgatto (2018): *MEEGA+: A method for the evaluation of educational games for computing education*. INCoD–Brazilian Institute for Digital Convergence, pp. 1–47.
- [18] Giani Petri et al. (2018): *A method for the evaluation of the quality of games for computing education*.
- [19] Christina Tikva & Efthimios Tambouris (2021): *Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review*. *Computers & Education* 162, p. 104083.
- [20] David Weintrop & Uri Wilensky (2012): *RoboBuilder: A program-to-play constructionist video game*. In: *Proceedings of the constructionism 2012 conference*. Athens, Greece.
- [21] Jeanette Wing (2011): *Research notebook: Computational thinking—What and why*. *The link magazine* 6, pp. 20–23.
- [22] Daniela Zehetmeier, Axel Böttcher, Anne Brüggemann-Klein & Veronika Thurner (2019): *Defining the competence of abstract thinking and evaluating CS-students' level of abstraction*. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*.
- [23] Weinan Zhao & Valerie J Shute (2019): *Can playing a video game foster computational thinking skills?* *Computers & Education* 141, p. 103633.