Chapter 12

Game Development-Based Learning: A New Paradigm for Teaching Computer and Object-Oriented Programming

Alaa Khalaf Al-Makhzoomy  
Yarmouk University, Jordan

Ke Zhang  
Wayne State University, USA

Timothy Spannaus  
Wayne State University, USA

ABSTRACT

This chapter presents the findings from a quasi-experimental study analyzing the effect of Game Development-Based Learning on students’ academic performance in programming courses in Jordan. The study tested an argument proposing a positive significant association between GDBL instruction and students’ performance. The analysis of variance results investigating the effect of enrollment and completion of a concurrent GDBL course to normal courses found that the treatment group outperformed two other groups: the control and the comparison group. The positive gains in the post-assessment scores, were consistent across the two programming courses: C++ and Object-Oriented Programming. This finding confirms the earlier results across countries and contexts documenting the salubrious effect of GDBL on students’ academic performance in Computer Science and Information Technology courses. Findings also support the overarching constructionist approach where the use of scaffolding and technology in instruction and assessment yield better academic outcomes for learners.

DOI: 10.4018/978-1-7998-0249-5.ch012
INTRODUCTION

Computer Science (CS) jobs are expected to grow at the highest rate compared to any other occupation in the next decade according to the Bureau of Labor Statistics of the United States. An average annual increase of 13% between 2016 and 2026 is projected yielding an addition of 557,100 openings to the marketplace, the best choice for any uncertain student looking for job security (Bureau of Labor Statistics, 2019). All such jobs feature the rudimentary or advanced use of programming from coding to data analytics. Therefore, mastering programming knowledge, skills and abilities (KSAs) proves essential in obtaining, retaining and advancing in a CS or information technology (IT) job.

Empirical evidence on passing rates of programming courses is conflicting. Two quantitative studies based on low response rates and available secondary data, mostly surveys-based, indicate that passing rates of programming is about 67%, a figure contradicting the popular claim that most students fail programming (Watson & Li, 2014; Porter, Guzdial, McDowell, & Simon, 2013). Other studies have concluded that students’ dropout rates from computer science are the highest among college majors and such a trend is plausibly due to the difficult learning curve faced in programming courses (Baytak, Land & Smith, 2011; Wu & Wang, 2012; Ismail, Ngah, & Umar, 2010). This is evident in the lower rates of conferred degrees in CS and IT compared to other fields like Engineering or Biological Sciences. No matter what the truth is, whether passing rates are high or low in programming, instructors need to improve their instruction and assessment of programming, providing students with the best available education for them to succeed (Ernst & Clark, 2012; Alkhawaldeh & Menchaca, 2014).

Game Development-Based Learning (GDBL) is an emerging educational strategy where students design and build their games as learning activities aimed at mastering programming content (Charlier, & De Fraine, 2012). GDBL is based on the constructivist learning approach where learners engage in experiential and interactive learning with the instructor and peers, thereby increasing their involvement and motivation for taking part in the learning process (Overmars, 2004). The use of GameMaker® in teaching programming across countries and courses has demonstrated a steady improvement in students’ outcomes and skills (Papastergiou, 2009). Mathematical-Logical Intelligence plays a crucial role in computer programming where learners think conceptually and abstractly dealing with numbers, memory and statements. It also plays a role in GDBL especially when using GameMaker® alongside with the Visual-Spatial Intelligence. The learner will design and develop games using their ability to visualize the objects and their actions.

This chapter outlines the results of an experiment that took place at a public university in the north of Jordan where GameMaker® was introduced in two courses, C++ and Object-Oriented Programming, and was found to positively and significantly improve students’ motivation, performance and retention in CS and IT (Eagle & Barnes, 2009; Yarmouk University, 2017b). The findings recommend CS instructors to incorporate GDBL learning strategies in their classes and advocates that such a practice should be embedded in departmental curricula (Doman, Sleigh & Garrison, 2015).

BACKGROUND

Previous research has noted the wide array of benefits gained through the use of GameMaker® for teaching programming at all educational levels. First, students’ motivation and interest in CS and IT improves with game authoring and design as part of programming courses (Johnson, 2017). GameMaker® can be