# Workshop on Using Games to Teach Software Engineering Concepts

Emily Navarro
*University of California, Irvine*
*emilyo@ics.uci.edu*

Gil Taran
*Carnegie Mellon University*
*gtaran@cs.cmu.edu*

### *Abstract*

*Educational games have become a popular and effective teaching method in almost every domain. Software engineering has followed suit in recent years, as a number of games have been developed and used to teaching students software engineering concepts. This workshop will explore the use of games in software engineering education—in particular, which games are available, how they have been used, what lessons have been learned concerning their use, and how the field of software engineering education as whole can be progressed through games. The workshop will consist of presentations by participants with accepted papers, and discussion of both these presentations and the field as a whole.*

## 1. The Topic, Theme, and Goals

It is well-known that a fun learning experience is a memorable learning experience. When students are having fun while learning, they learn the lessons better and deeper. It is because of this that the field of educational games is now a multi-million dollar industry. Games offer players the chance to develop analytical skills, practice lessons repeatedly, and put into practice the concepts they learn through other means such as lectures and textbooks, all in an entertaining and motivating environment. In the past several years many in the field of software engineering have followed this trend by developing and incorporating games into software engineering courses. Software engineering in particular is a good fit for learning through games, since games can be used to simulate the many real-world aspects of software engineering situations that are infeasible to practice otherwise—large teams of people, multiple stakeholders, substantially-sized projects, and random adverse events, among others.

The goal of this workshop is to expose participants to how games are used or have been used in software engineering education, and provide a forum for discussion and learning from each other about this topic. Submissions can describe the development of an educational software engineering game, the use of a game in a software engineering educational setting, or anything else pertaining to games in software engineering education. Submissions should include a clear statement of lessons learned through the authors' experience—lessons that others can use in their own courses. It is our intent that workshop attendees will learn:

1. The different games available for use in software engineering education
2. How to use games effectively in software engineering education. In other words, the critical considerations that must be made concerning the use of games in this field.
3. How to move the field of software engineering education forward through games.

In addition we hope that the workshop activities will spark ideas leading to the creation of new educational software engineering games, or new uses for existing games.

## 2. Facilitators and Their Affiliations

Emily Navarro is an Assistant Project Scientist at the University of California, Irvine. She received her Ph.D. in Information and Computer Sciences from the University of California, Irvine in 2006. Emily developed SimSE, a widely-used educational software engineering simulation game, and has also contributed to the development of Problems and Programmers, an educational software engineering card game. She has studied the use of learning theories in software engineering educational approaches and has designed and conducted numerous studies involving the use of games in the software engineering classroom.

Gil Taran is an Associate Professor at Carnegie Mellon University. He has experience in planning, development and management of large-scale multi-national projects working mostly with government clients in the financial, operational and defense sectors. He earned his bachelor's degrees in Economics and in English at the university of Tel-Aviv, and his Masters in Information Security at Carnegie Mellon University. He is continuously involved in setting up educational software engineering programs in Eastern Europe, and travels extensively to consult companies around the world through GIMETO LLC, a company he founded to help teach software engineering courses.

## 3. Intended Audience and the Preparation Required of Them

The intended audience is those who teach any software engineering-related courses, either in industry or academia, and want to learn more about the use of games in this field. There is no preparation required of them (unless they have a paper accepted, in which case they will be presenting), as there will be presentations made and discussions centered around the ideas given in these presentations.

## 4. Activities and Format

The workshop will begin with an introductory presentation by the facilitators, giving an overview of the use of games in software engineering education. Following this will be two or three presentations of accepted papers, followed by a discussion period, a break, one more set of two or three presentations, one more discussion period, and one wrap-up discussion period.

The discussion periods will center around the presented papers, taking the form of question and answer sessions in which attendees can ask the presenters questions about their presentations. In the wrap-up discussion period, the facilitators will lead the group to summarize and analyze the lessons learned by posing the following questions to the participants, which will be addressed in a workshop-wide discussion format:

1. Which software engineering concepts are best/least able to be taught through games?
2. Where have we seen the greatest successes/failures in the use of games in software engineering education? What can we learn from these successes/failures about how to proceed from here, both in our own classrooms and in the field as a whole?
3. Where are the holes in the field—which software engineering concepts have the most potential to be taught through games, but are not currently taught through games?
4. What are some of the overarching lessons we have learned about the use of games in software engineering education, that seem to be applicable across the field?
5. How is software engineering as a field well-suited and/or not well-suited to the use of educational games?

## 5. Anticipated Time Requirements

This workshop will be half-day.