4th International Workshop on Games and Software Engineering (GAS 2015)

Judith Bishop
Microsoft Research
Redmond, U.S.A.
jbishop@microsoft.com

Kendra M.L. Cooper
The University of British Columbia
Vancouver, Canada
kendra.m.cooper@gmail.com

Walter Scacchi
University of California, Irvine
Irvine, U.S.A.
wscacchi@ics.uci.edu

Jim Whitehead
University of California, Santa Cruz
Santa Cruz, U.S.A.
ejw@soe.ucsc.edu

***Abstract*—We present a summary of the 4th ICSE Workshop on Games and Software Engineering. The full day workshop is planned to include a keynote speaker, game-jam demonstration session, and paper presentations on game software engineering topics related to software engineering education, frameworks for game development and infrastructure, quality assurance, and model-based game development. The accepted papers are overviewed here.**

***Index Terms*—Game engineering, software engineering.**

I. Introduction

The 4th ICSE Workshop on Games and Software Engineering (GAS) is held on May 18, 2015 in Florence, Italy, located with the 2015 ACM/IEEE International Conference on Software Engineering.

Building upon the first three ICSE GAS Workshops, we are excited to provide a forum to interactively explore leading edge research in game software engineering from a number of perspectives, including (1) the role of games in education for diverse communities including university and middle school students as well reaching broad audiences with crowdsourced games; (2) development and infrastructure frameworks (3) quality assurance via testing and formal verification approaches; and (4) model-based game development.

The full day program is organized to offer an interesting mix of activities including a keynote speaker, game-jam demonstration session, and paper presentations; the intent is to promote discussions and interactions among the participants. The workshop has accepted nine high quality submissions from research groups across Asia, Europe, North America (eight long and one short). In this summary, we present an overview of these papers.

II. Research Paper Presentation Sessions

A. Games in Education

Game-based learning in undergraduate software engineering education continues to receive attention in the research community; a study is reported in "Instructor's Acceptance of Games Utilization in Undergraduate Software Engineering Education, A Pilot Study in Turkey" authored by Ozlem Albayrak, which explores factors that have a significant impact on an instructor's decision to use games in software engineering education. The author proposes a model that embodies factors influencing an instructor's decision. The study reveals factors that impact the decision to use games (e.g., "the number of hours per week the instructor plays game"); interestingly, other factors (e.g., "instructor's previous research experience in education" do not impact the decision.

Game development courses need to include pedagogical aspects from problem-based, cooperative, blended and experiential learning to provide a high quality learning experience. In the article "Experiences from an Experiential Learning Course on Games Development", the co-authors Stephan Krusche, Barbara Reichart, Paul Tolstoi, and Bernd Bruegge present their course design, which draws upon a broad foundation of pedagogical aspects and embodies a diverse collection of software engineering skills: modeling; design patterns; configuration management; and soft-skill development. Their results indicate the course is highly appreciated as a valuable, long-term learning experience by the students.

Focusing on younger students, the article "Serious game development as a creative learning experience: lessons learnt", co-authored by Varvara Garneli, Michail N. Giannakos, Konstantinos Chorianopoulos, and Letizia Jaccheri presents a study exploring alternative project-based learning approaches with middle school students. Two research questions were considered in this work - the effects of teaching strategies (serious game development, simulation) on the students' programming habits and the influence of a top-down and a bottom-up approach when using a visual programming environment. The students in the bottom-up treatment completed their projects successfully; students in the more traditional top-down approach chose to experiment with more complex curriculum in their projects, but were not always successful.

Games for very broad audiences, such as those for crowdsourcing science problems, need appealing, consistent, visual metaphors to represent advanced, underlying concepts. In the article "Visualizing Loops and Data Structures in Xylem: The Code of Plants", co-authored by Heather Logas, Richard Vallejos, Joseph Osborn, Kate Compton, and Jim Whitehead, the authors adopted a plant and flower metaphor in their game, Xylem: The Code of Plants. The game is designed to crowdsource formal software verification via loop invariant specifications. The plant metaphor accommodates multiple

978-1-4799-1934-5/15 $31.00 © 2015 IEEE
DOI 10.1109/ICSE.2015.314

types of data structures, maintains the integrity of the game narrative, and abstracts the complexity of the target source code. Xylem has had over 2,100 downloads to date.

B. Frameworks

3D, gesture-interactive, web-based applications are a growing category of games that require virtual collaboration and motion sensing. In the article "Space Connection: A New 3D Tele-Immersion Platform for Web-Based Gesture-Collaborative Games and Services", co-authored by Chun-Han Lin, Pei-Yu Sun, and Fang Yu, a new infrastructure framework is proposed, based on a client-server architecture. The authors present a cross-platform framework for thin-client, all-web-usable 3D gesture-interactive applications. To ensure strong response time performance, a new socket transmission protocol is defined to provide transparent APIs for browsers and external devices. The platform has been developed and validated with two game applications: an interactive ping pong game and a rehabilitation system.

Game development frameworks provide capabilities to accelerate the iterative design process, often for one genre of game. In the short article "A Game Genre Agnostic Framework For Game-Design A.I.", the co-author Jonathan Tremblay and Clark Verbrugge propose a broadly applicable framework that uses an independent service to provide intelligent, non-player characters that play the game and collect data. The game designer needs to provide the framework with a formal game representation and model of the game rules. The framework has been developed and validated with three games, from distinct genre.

C. Quality Assurance

Quality assurance in game development is a broad topic, including more traditional software quality aspects (capabilities, code quality), in addition to the game quality aspects (art, design, creativity) and the user experience (fun, satisfaction). The article "A Comprehensive Study on Quality Assurance in Game Development", co-authored by Maria Komal, Zaineb Khalil, and Mehreen Sirshar presents a survey spanning 20 approaches that are analyzed using 15 comparison criteria. Based on the analysis, the authors conclude that the primary focus in game development is on the quality attributes such as the player experience, fun factor, and game rule balance, rather than more traditional software quality attributes such as performance, reliability, security, or efficiency.

As an alternative to testing early prototypes of games, the article "Case Studies of Application of Probabilistic and Statistical Model Checking in Game Design", co-authored by Paolo Milazzo, Giovanni Pardini, Dario Sestini, and Pasquale Bove, explored the use of model checking techniques to game design problems. A model of the core mechanics of a game can be defined and precisely evaluated with respect to properties such as the game duration, the probability of victory, the probability of victory using different strategies, and the role of each game component in determining the winner. Three example board games were used to validate the approach, exploring aspects of game randomization, the existence of different winning strategies in a (competitive) strategy game, and collaboration. The established probabilistic/statistical model checker adopted was the PRISM tool.

D. Model-based Game Development

The Model-View-Controller (MVC) is an established architectural style that is used in game designs to effectively modularize gameplay code from user interface code (e.g., rendering), which enables transitions to new technology or different platforms. In the article "Evolution and Evaluation of the Model-View-Controller Architecture in Games", co-authored by Tobias Olsson, Daniel Toll, Anna Wingkvist, and Morgan Ericsson, an investigation is presented on the implementation of the MVC style across a collection of five games from a small development company. The authors define a metrics-based quality model to assess software quality goals such as portability and rendering engine independence. Using this quality model, their analysis reveals three different evolutions of the architecture in the game collection, each with differing quality.

Game engines provide designers with environments to re-use extensive collections of gameplay assets, making the implementation of interesting gameplay less expensive and more robust. The development of game engines remains challenging, involving the definition of architectures and gameplay design facilities that are flexible for game designers and enable the maintenance and evolution of the engine over time. The co-authors Victor Guana, Eleni Stroulia, and Vina Nguyen present an experience report in the article "Building a Game Engine: A Tale of Modern Model-Driven Engineering", on their systematic development of a game engine, PhyDSL-2, for 2D physics-based games. The complexity of the project is managed using model-driven engineering technologies, which support the systematic refinement of more abstract models (e.g., architecture) to the code level via transformations. The authors have developed a tool that allows game designers to visualize models, transformation scripts, and generated architectural components.

III. ACKNOWLEDGMENT

We extend our thanks to all members of the international program committee for their time and support:

Navid Ahmadi, Actimator, U.S.A.
Phaedra Boinodiris, IBM, U.S.A.
Bernd Bruegge, Technische Universitaet Muenchen, Germany
David Callele, Experience First Design, Canada
Kuan-Ta Chen, Academia Sinica, Taiwan
Michael D. Ernst, University of Washington, U.S.A.
Robert Hall, AT&T Labs, U.S.A.
Mario Herger, Austrian Innovation CTR Silicon Valley,U.S.A.
Letizia Jaccheri, NTNU, Norway
Gail Kaiser, Columbia University, U.S.A.
Chris Lewis, Google, U.S.A.
Fabio Petrillo, Univ. Federal do Rio Grande do Sul, Brazil
Ioannis Stamelos, Aristotle University of Thessaloniki, Greece
Nikolai Tillmann, Microsoft Research, U.S.A.
Clark Verbrugge, McGill University, Canada
Tao Xie, University of Illinois at Urbana-Champaign, U.S.A.