

exemplar has proven to be useful as the first contact of students with design patterns. Yet, there are other subjects, both within and outside the gaming context, that can be equally good. Our aim is to contribute to the software engineering teaching community with a ready-to-use, familiar, fun, and validated exemplar.

7.1 Future Work

We have several plans for future work: (i) perform further evaluations of our exemplar, with a greater number of classes in the courses of Systems Modeling, Software Engineering, and OO Programming, to obtain more evidence with respect to usefulness and efficacy. (ii) Identify more applications of the same patterns within Angry Birds and provide examples of pattern combinations (in response to the students' feedback). (iii) Augment examples with auxiliary resources (e.g., source code and sequence diagrams). (iv) Explore a more collaborative scenario, in which students can work together for solving problems, experiencing different games and other scenarios in a creative fashion. (v) Use open source implementations based on Angry Birds ³, allowing students to practice design patterns implementation. (vi) Explore "side effects" of design patterns, such as the drop in application performance [7]. (vii) Use games to teach other software engineering concepts. We are currently working on item *iii* and also tackling replicability.

ACKNOWLEDGMENTS

We thank Evandro Macedo, Rodrigo Bello and Anny Mayara for their feedback on the patterns. We thank CNPq and FAPERJ for the financial support. The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 712949 (TECNIOspring PLUS) and from the Agency for Business Competitiveness of the Government of Catalonia.

REFERENCES

- [1] Ellen Agerbo and Aino Cornils. 1998. How to Preserve the Benefits of Design Patterns. In *13th ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications (OOPSLA)*. 134–143.
- [2] David Paul Ausubel, Joseph Donald Novak, Helen Hanesian, et al. 1968. Educational psychology: A cognitive view. (1968).
- [3] K. Beck, R. Crocker, G. Meszaros, J. Vlissides, J. O. Coplien, L. Dominick, and F. Paulisch. 1996. Industrial experience with design patterns. In *18th International Conference on Software Engineering (ICSE)*. 103–114.
- [4] Phyllis Blumenfeld, Elliot Soloway, Ronald Marx, and Joseph S. Krajcik. 2011. Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. *Educational Psychologist* (2011), 369–398.
- [5] Kajal Claypool and Mark Claypool. 2005. Teaching Software Engineering Through Game Design. In *10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. 123–127.
- [6] Elisabeth Freeman, Eric Freeman, Bert Bates, and Kathy Sierra. 2004. *Head First Design Patterns*. O'Reilly Media, Inc.
- [7] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley, Boston, MA, USA.
- [8] Paul Gestwicki and Fu-Shing Sun. 2008. Teaching Design Patterns through Computer Game Development. *Journal on Educational Resources in Computing* (2008), 1–22.
- [9] A. Ghazarian. 2012. Work in Progress: Transitioning from Novice to Expert Software Engineers through Design Patterns: Is It Really Working?. In *Frontiers in Education Conference (FIE)*. 1–2.
- [10] Raph Koster. 2013. *Theory of Fun for Game Design* (2nd ed.). O'Reilly Media, Inc.
- [11] D. M. Nascimento, K. Cox, T. Almeida, W. Sampaio, R. A. Bittencourt, R. Souza, and C. Chavez. 2013. Using Open Source Projects in software engineering education: A systematic mapping study. In *2013 IEEE Frontiers in Education Conference (FIE)*. 1837–1843. <https://doi.org/10.1109/FIE.2013.6685155>
- [12] Nelishia Pillay. 2010. Teaching design patterns. In *Southern African Computer Lecturers Association Conference (SACLA)*. 1–4.
- [13] A. Radermacher, G. Walia, and D. Knudson. 2014. Investigating the skill gap between graduating students and industry expectations. In *36th International Conference on Software Engineering (ICSE 2014)*. 291–300.
- [14] IEEE Computer Society. 2014. *Guide of the Software Engineering Body of Knowledge - SWEBOOK, version 3.0*. IEEE Press.

³See <https://github.com/estevaofon/angry-birds-python> for an example

- [15] Jonathan Sillito. 2013. Simple collaborative qualitative analysis. <http://www.saturateapp.com>
- [16] Rovio Entertainment Ltd. 2015. Angry Birds. <http://www.rovio.com/en/our-work/games/view/1/angry-birds>
- [17] Dirk Riehle. 2011. Lessons learned from using design patterns in industry projects. In *Transactions on Pattern Languages of Programming II*, James Noble and Ralph Johnson (Eds.). Springer-Verlag, Berlin, Heidelberg, 1–15.
- [18] Marcelo Schots, Claudia Rodrigues, Cláudia Werner, and Leonardo Murta. 2010. A Study on the Application of the PREViA Approach in Modeling Education. In *XXIX International Conference of the Chilean Computer Science Society (SCCC)*. Antofagasta, Chile, 96–101.
- [19] Forrest Shull, Janice Singer, and Dag IK Sjøberg. 2008. *Guide to advanced empirical software engineering*. Vol. 93. Springer.
- [20] Lev Semenovich Vygotsky. 1980. *Mind in society: The development of higher psychological processes*. Harvard University Press.