LESSONS LEARNED FROM GAME PROGRAMMING PROJECT IN A SOFTWARE ENGINEERING COURSE

Girts Strazdins, Department of ICT and Natural Sciences, NTNU, Norwegian University of Science and Technology, gist@ntnu.no

1. MOTIVATION

Software engineering best practices and principles is an important part of every computer engineering university curriculum. However, information in software engineering related courses is typically very theoretical. Practical activities are needed to let students experience the development processes instead of simply memorize the theory. Software development projects can play the role of practical experience. Game programming is one option for project topics. When it comes to introducing such projects, course teachers meet several challenges. How to involve game programming in software engineering courses in an efficient way? How to help the students who have no previous experience with neither game design, nor game engines and other tools? What consequences will it have on the course?

The purpose of this poster and demo is to share experience from an undergraduate software engineering course taught in the Spring 2018, at the Norwegian University of Science and Technology, NTNU. The demonstration includes student-developed games available at study program's website[1].

2. EXPERIMENTAL GAMELAB COURSE

In addition to theoretical lectures in the course ID202712 "System development and modelling" (4th semester bachelors of computer engineering, 10 ECTS) students were working in groups and developed a game as part of their software projects. 33 students were split into 9 groups of 3-4 students. The teams were formed by the teacher based on results of the Jung's personality test that students filled in the beginning of the course and their self-evaluation of skills in programming, design and communication.

Projects had pupils from Aalesund International School Years 3 and 4 as customers. Students worked for 13 weeks and had a final presentation of the games in a form of a workshop with demonstrations and voting. Three awards were given: Audience award, best educational game and best technical solution.

Students did not have any subjects on game programming or design in the previous studies. However, they had introductory programming courses and general user interface design rules have been discussed. To help students familiarize with game programming concepts and game engine principles, a tutorial[2] with code examples and documentation was developed by course teacher and two student assistants.

3. RESULTS

Feedback in the course was collected in two ways. First, a questionnaire was sent to all students. Response rate: 79% (26 of 33 students). In addition, an open discussion with all students was performed after the last lecture of the course. Results shows that:

1. Students are able to develop unexpectedly rich games in the short period of time and with the limited previous experience.

[1] https://www.ntnu.no/studier/004da/studentoppgaver

[2] https://github.com/strazdinsg/gamelab/wiki

2. Despite teacher's suggestion to use Java-based game engines such as LibGDX[3] or jMonkey[4], most students (4 of 9 teams involving 15 of 33 students) chose Unity 3D game engine. Even those who did not, at the end of the course reflected that next time they would choose Unity due to its rich and intuitive graphical user interface. Unity allows building most of game components without writing code.
3. It is difficult to develop a unified tutorial for all students. It is perhaps better to simply let the students explore existing online tutorials.
4. Choice of groups is essential. Forming groups based on student skill in programming and Jung's personality type has been a success (with few exceptions).
5. During the process, students get confident that they can participate in software development projects with best practices and common tools (such as GIT and Jira).
6. Games as projects and external customers is the strongest motivation behind efficient work: 89% of students get motivated. In comparison, mandatory weekly meetings with teacher is a bad motivator: only 50% of students see it as motivating.
7. While students put extra effort in the game projects, they sometimes don't see clearly the connection between the project and theoretical lectures and books. It should be expected that in the *fight* "Game projects versus book reading" the former would win. Involving enough theory in the projects is challenging for course teachers.

4. ACKNOWLEDGEMENTS

The author would like to thank head of NTNU IIR department Anniken Th. Karlsen for supporting the experimental ideas used in the course and Aalesund International School for letting their pupils to play the role of customers in the projects. Thanks to Ole-Martin Steinnes and Thomas Sund Mjåland for great effort in developing the game programming tutorial. The project is partly supported by the Centre for Excellent IT Education, ExcIted[5].

[3] https://libgdx.badlogicgames.com/

[4] http://jmonkeyengine.org/

[5] https://www.ntnu.edu/excited