Pedagogic case and specific course in which designed tasks and units are used

Name of university: UCM

Contact person: Adrián Riesco (ariesco@ucm.es)

Pedagogic case:	• Teaching formal specification in rewriting logic • Teaching formal verification with Linear Temporal Logic via model checking. • The course is aimed to Master's students in Computer Science.
Description (including temporal scheme for design, development and implementation)	• Rewriting logic is a logic of change, where transitions are described with rules. Verification in these systems is done by analyzing the corresponding automata generated by applying rules in a non-deterministic way. • The automata of these systems stands for the state space of many different games, so the idea is to introduce the system with simple games: inquire will be used in programming assignments for students.

Aim of pedagogic case	• To reflect on how to use games to teach mathematical concepts to non-mathematicians.
Mathematical concepts	• Rewriting logic. • Linear Temporal Logic.
Addressed practice	• Master's course on Computer Science. First semester
Place in specific course Course name Place of units	• One semester course. • Auditory and Quality Assurance. • Unrelated with other subjects; the Master's course gives a wide description of several topics but details are explored in the Master's Thesis only.
Learners profile orientation, year, age, prior knowledge, other such as math anxiety, special needs, ..	• Approximately 23 years old. • Degree in Computer Science. • Around 25 students.
Organisation of specific course study credits/hours, location, group size	• 60 hours. 4 hours per week; 15 weeks. • The case focuses on part of the subject, around 20 hours. • Each week the teacher presents the concepts for 2 hours (Wednesday) and the students work under the supervision of the teacher for 2 hours (Friday).
Expected learning outcomes	• Students are able to specify and verify simple but real systems.
Envisioned use of digital technology	• Completely; students use Maude, a specification language, to code their assignments.
Planning of tasks	• Discussion with other teachers involved in specification of systems. • Related work - Maude book. • Design of programming examples and assignments. • Concepts and syntax are taught via games. • First assignments are games as well; final assignments are a more complex version of these games. • 1st assignment - Discussion in large groups (the whole class under the teacher supervision). • 2nd and 3rd assignments - Discussion in small groups. • 4th and 5th assignments - Discussion in groups of at most 2 students. Graded. • Wednesdays are used to introduce new example and discuss those problems that students could not solve on Friday. • Study how well theoretical ideas are coded.

	• Discussion of results.
Names of persons involved	• Adrián Riesco
Course:	Master Course in Computer Science
Learning objectives	On completion of this module, students should be able to: **Knowledge and Understanding** - Specify average transition systems. - Verify these systems. **Subject-specific Skills** - Equational and rewriting logic. - Model checking. **Transferable Skills** - graph theory; - declarative programming.
Learning contents	• Membership equational logic - Maude functional modules. • Rewriting logic - Maude system modules. • Modal logic. • Model checking - Maude MODEL-CHECKER module. • Partial order reduction.
teaching /learning activities	• Programming assignments. • Programming examples via games.
Media	• Maude
Evaluation	• Correctness of ideas. • Correctness of code.
Instructor role	• On Wednesday he/she presents the basic ideas and helps students with those problems they could not solve. • On Friday he/she first leads discussions. in large gropus Then he/she helps in smaller groups.
Student roles	• Engage in assignments. In lectures they propose ideas but the teacher leads.