EAIE

Esports as Accelerator for Social Inclusive Education

**LESSON**

# PHPUNIT ARENA

**AUTHOR**

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

# PHPUNIT ARENA

PHPUnit Arena is a test-design tournament built around intentionally flawed PHP code. Learners create unit tests that expose edge cases, improve implementation confidence and demonstrate that useful coverage depends on test quality rather than percentages alone.

### Subject and level

VET PHP and backend learners with object-oriented programming foundations who are beginning or consolidating automated unit testing with PHPUnit.

### Connection to curriculum

The exercise supports code quality, unit testing, assertions, test organisation, edge-case analysis and refactoring. It links testing to maintainability by requiring students to prove behaviour before and after a code correction.

### Didactic and pedagogic considerations

Friendly competition motivates breadth of testing, but peer review prevents students from treating coverage as the only objective. Hidden cases encourage specification-based reasoning and show the value of testing boundaries and failure conditions.

### Time estimate

2 hours

# PHPUNIT ARENA

**Group size**

15-25 students

**Overall introduction to exercise**

Teams receive small PHP classes containing subtle defects. Their first task is not to repair the code but to design tests that expose incorrect behaviour. A second round adds hidden acceptance cases and peer review.

The arena rewards tests that are readable, independent and meaningful. Students therefore learn to use automated testing as a design and reasoning tool, not merely as a final verification step.

# PHPUNIT ARENA

## Exercise as presented to the students

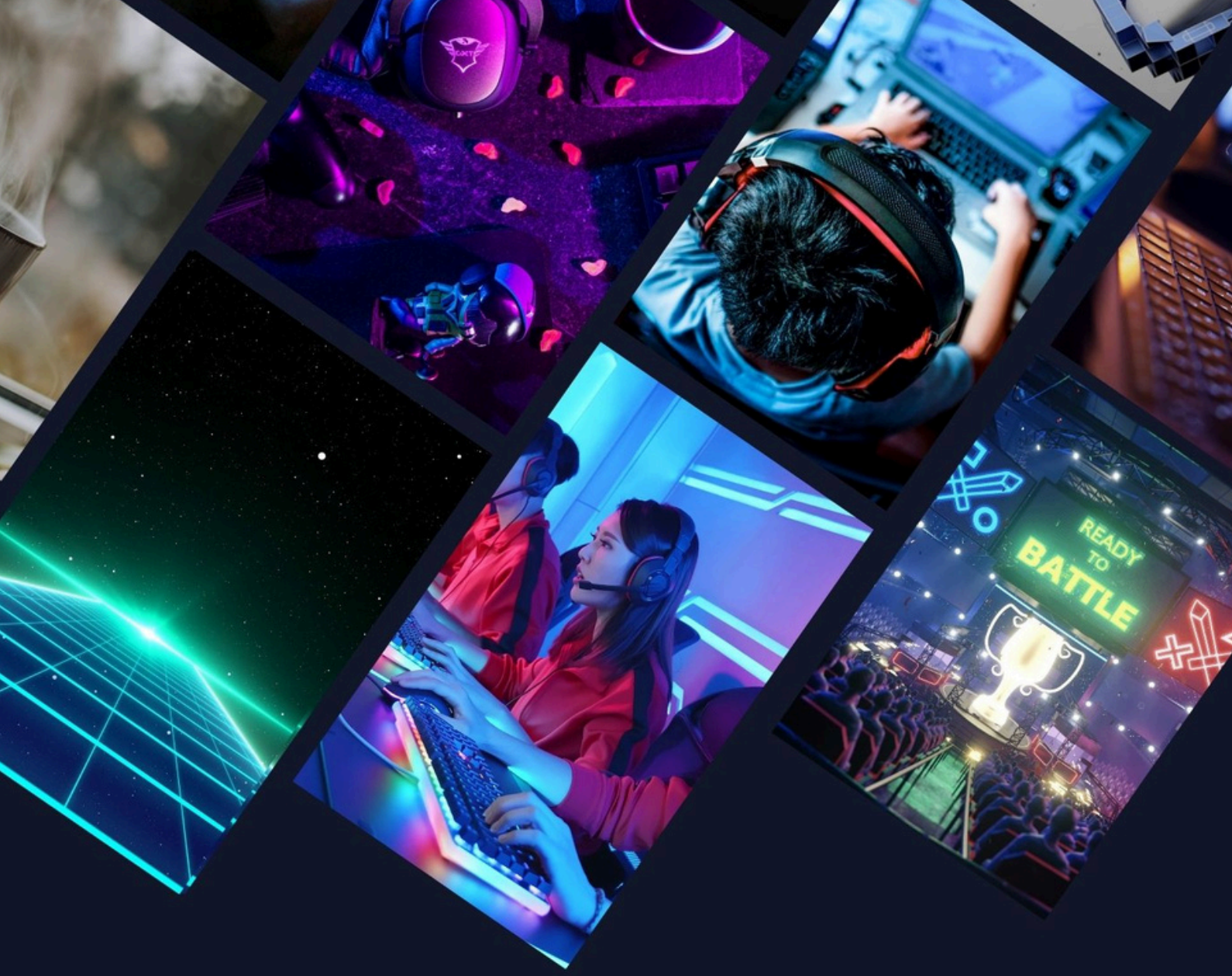
The lesson opened with a brief demonstration of a PHPUnit test containing arrange, act and assert steps. Students then formed pairs and received one of several small PHP classes: a discount calculator, date helper or basket total. Each class contained edge-case defects that were not visible from the simplest example.

Round one gave teams 30 minutes to write tests without changing production code. Points were awarded when a new test failed for a legitimate reason and clearly described the expected behaviour. Students quickly discovered that normal cases were insufficient and began testing empty values, limits, invalid input and boundary conditions.

Once a defect had been exposed, teams fixed the implementation and reran the full suite. The teacher then swapped code between pairs for a short peer review focused on test names, independence and assertion quality. Tests that merely reproduced implementation details were challenged and rewritten to describe behaviour more clearly.

In the hidden-test round, the teacher supplied only a short specification and later ran additional cases unknown to the students. This encouraged them to predict likely boundaries instead of tailoring tests to visible examples. Coverage reports were shown, but teams were asked to explain why a high percentage could still hide weak assertions.

The arena finished with a comparison of the most valuable tests rather than simply the highest score. Students discussed which cases prevented regressions and which were redundant. The exercise built confidence with PHPUnit while developing careful reasoning, specification reading and the professional habit of proving a fix with repeatable tests.




EASIE
Esports as Accelerator for Social Inclusive Education

# E-SPORT AS ACCELERATOR FOR SOCIAL INCLUSIVE EDUCATION

WWW.EASIE.GG

Co-funded by
the European Union

FUNDED BY THE EUROPEAN UNION. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR THE EUROPEAN EDUCATION AND CULTURE EXECUTIVE AGENCY (EACEA). NEITHER THE EUROPEAN UNION NOR EACEA CAN BE HELD RESPONSIBLE FOR THEM.