Co-funded by
the European Union

EAIE
Esports as Accelerator for Social Inclusive Education

LESSON

THE NIGHTWATCH DETECTIVE

AUTHOR

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

THE NIGHTWATCH DETECTIVE

The Nightwatch Detective turns browser automation into a software investigation. Learners write Nightwatch tests against a deliberately faulty training application, gathering repeatable evidence for defects in login, navigation, forms and checkout behaviour.

Subject and level

VET front-end and QA learners who understand basic JavaScript and browser behaviour and are beginning end-to-end automated testing with Nightwatch.

Connection to curriculum

The exercise reinforces test design, DOM selection, assertions, reproducible bug reporting and automated validation of user journeys. It links quality assurance with development by requiring defects to be expressed as executable checks.

Didactic and pedagogic considerations

A detective narrative rewards careful observation rather than random clicking. Hidden defects and a prioritised incident encourage learners to distinguish critical behaviour from cosmetic issues, while pair work supports discussion about selectors, assertions and reliable test design.

Time estimate

2 hours

THE NIGHTWATCH DETECTIVE

Group size

15-20 students

Overall introduction to exercise

Pairs investigate a local e-commerce training application containing intentional defects. Every reported bug must be supported by a Nightwatch test or clearly documented reproduction path, turning quality assurance into evidence-based technical work.

The exercise develops systematic thinking and shows how automation protects behaviour over time. Students experience the difference between discovering a defect once and creating a test that can detect the same regression repeatedly.

THE NIGHTWATCH DETECTIVE

Exercise as presented to the students

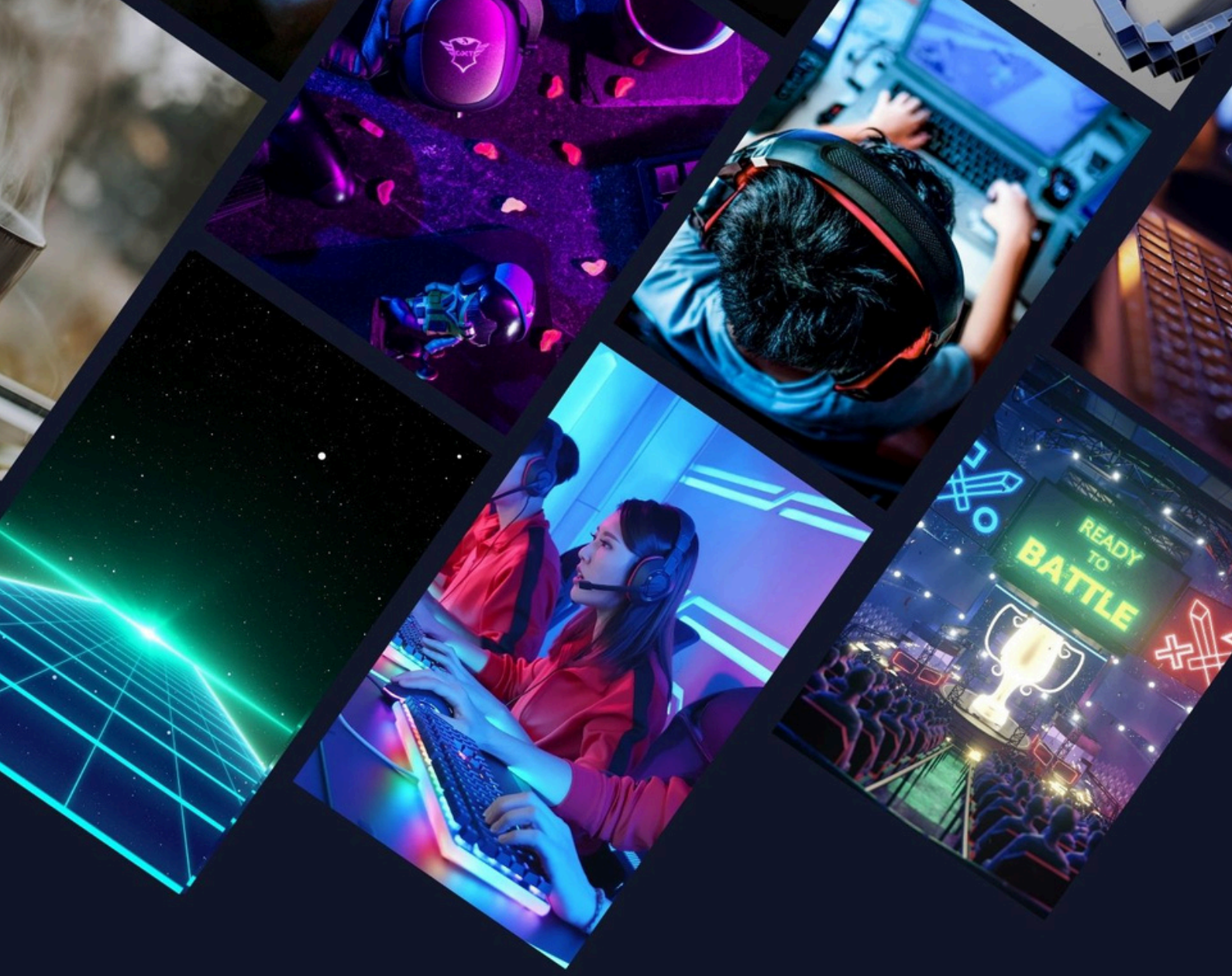
Students were introduced as QA detectives assigned to a fictional online shop shortly before release. Each pair received a Docker-based copy of the application and a Nightwatch project with one example login test. Their case file listed four customer journeys but did not reveal where the defects were hidden.

Pairs first explored the application manually and wrote hypotheses about risky areas. They then automated the login and product-search flows using stable selectors and assertions. A defect counted as evidence only when the team could reproduce it consistently or express the expected behaviour in an automated test.

The teacher had planted bonus defects such as an incorrect validation message and a navigation element that disappeared at a particular step. Midway through the lesson, a critical incident announced that checkout had stopped working, forcing teams to prioritise the business-critical journey before returning to lower-impact findings.

Students alternated between Nightwatch output, browser DevTools and the application itself. When a test failed for the wrong reason, they had to improve the selector or wait condition rather than simply increase delays. Pairs that finished early reviewed another team's test for readability and reliability.

The session closed with a short detective report: defects found, automated coverage created and one example of a false or unstable test. Teachers emphasised that quality assurance requires precision and communication as well as code. Learners finished with a reusable end-to-end test suite and a clearer understanding of regression testing.



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.