EA IE

Esports as Accelerator for Social Inclusive Education

LESSON

ANGULAR COMPONENTS CHALLENGE

AUTHOR

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

ANGULAR COMPONENTS CHALLENGE

Angular Components Challenge turns component architecture into a collaborative construction task. Learners build a dashboard from reusable pieces and progressively connect data, events, routing and services until the interface behaves as one coherent application.

Subject and level

Intermediate and advanced VET front-end learners with JavaScript or TypeScript foundations who are learning component-based application development with Angular.

Connection to curriculum

The design supports outcomes in user-interface development, modular architecture, event handling, data binding and service-based data access. It also reinforces maintainability and reuse as criteria for professional front-end code.

Didactic and pedagogic considerations

The challenge alternates short technical goals with peer review. Students can see and test each component immediately, making abstract ideas such as inputs, outputs and services visible through behaviour. Team construction also creates opportunities for explanation and code review.

Time estimate

2 hours

ANGULAR COMPONENTS CHALLENGE

Group size

20-30 students

Overall introduction to exercise

Small teams receive a base Angular project and a dashboard specification. They must create independent components, connect parent and child data flows and use a service to provide shared information. Progress is demonstrated visually as the dashboard becomes functional.

The activity rewards reusability, clarity and successful collaboration rather than decorative complexity alone. Students experience why good component boundaries simplify later changes and how an interface can be assembled efficiently when responsibilities are separated.

ANGULAR COMPONENTS CHALLENGE

Exercise as presented to the students

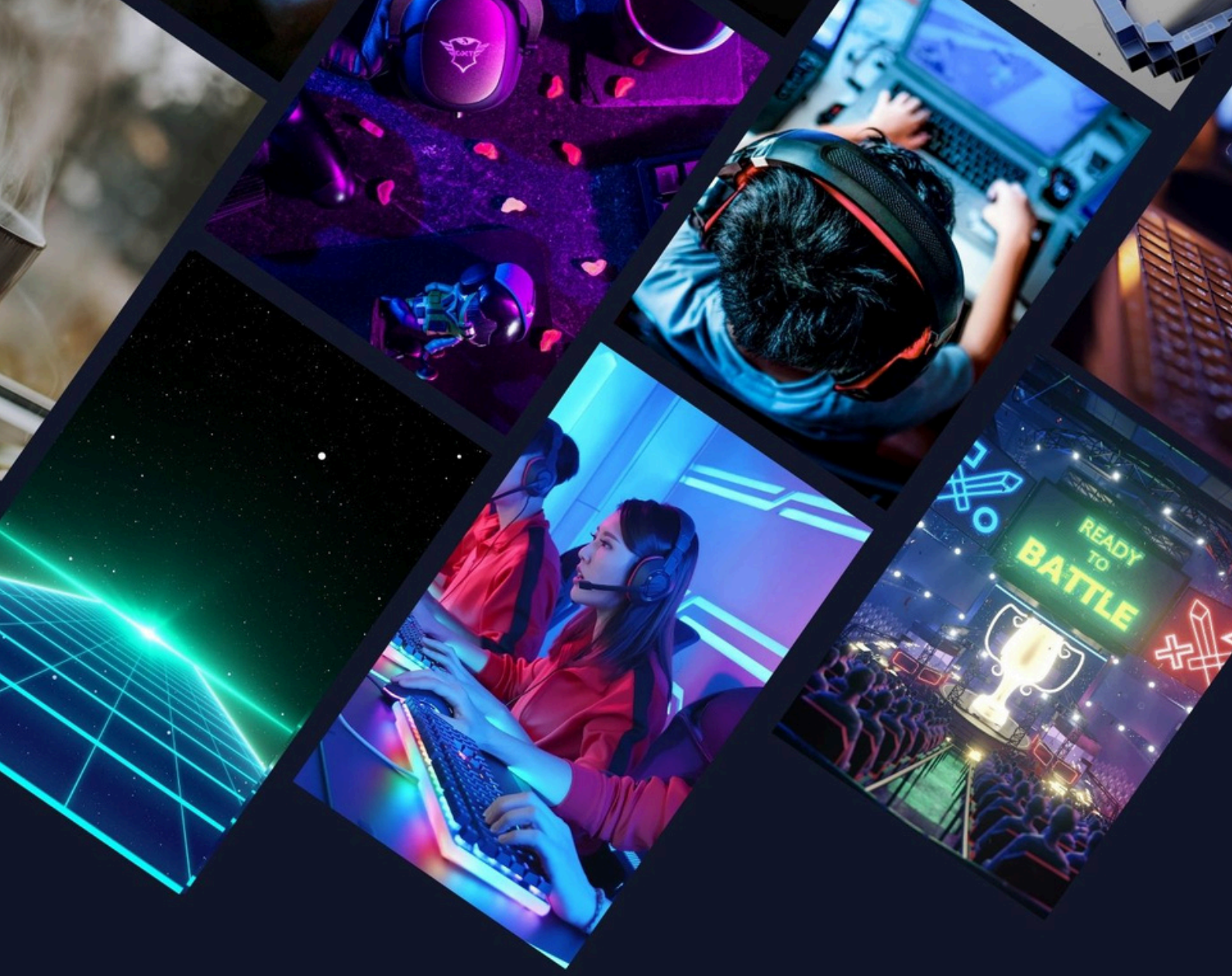
The session opened with a five-minute interactive review of property binding, event binding, inputs and outputs. Students then formed teams of three and received the same Angular starter application containing only a shell layout and a small local dataset. Their goal was to turn it into a functional student-services dashboard.

The first challenge required a reusable navigation component and two routed views. The second introduced a card component that received data through @Input and emitted a user action through @Output. Teams had to demonstrate both behaviours before moving to the service challenge.

Next, learners created an Angular service to expose shared data to more than one component. A surprise requirement then asked them to replace one synchronous value with an asynchronous observable or delayed data call. Students used browser tools and Angular error messages to debug imports, templates and lifecycle issues as they appeared.

When core functionality was complete, teams swapped laptops for a short peer review. They checked whether components were reusable, whether responsibilities were clearly separated and whether another developer could understand the naming. Suggestions had to be implemented or explicitly rejected with a technical reason.

The final demonstrations showed how different teams had solved the same dashboard brief. The teacher linked the solutions back to maintainability and scalable front-end architecture. Students finished with a working Angular application and direct experience of data flow, component contracts, services, routing and constructive code review.



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.