# Part 3: Modify the Pig Dice game to use advanced features

Part 3 of this case study has you modify the Pig Dice game from part 2 so that it uses modules. In addition, it has you use Node.js to provide an API that returns the dice roll value, the Fetch API to get the dice roll value, and the Canvas API to display the dice roll value.

If you haven’t completed part 2 yet, you’ll need to do that first, unless it’s given to you as a starting point.

Prerequisites: Chapters 1 to 16

## User interface



## Specifications

* Update the Pig Dice game to use ES modules.
* Modify the module that exports the game object so the Die object is private.
* Add a server.mjs file to host the die roll API. Add a rollDie() function to the server that returns a random number between 1 and 6.
* Modify the roll() method of the Die class to use the Fetch API to get a value from the server.mjs die roll API.
* Use a command prompt to start the servers needed to run the app.
* Add a <canvas> element to the HTML to display the die as shown above.
* In the Die class, add the following methods that accept a drawing context object:
* A draw() method that draws the die as shown above.
* A clear() method that clears the die.
* Modify the takeTurn() method of the game object to accept a drawing context object and draw the die.
* Modify the reset() and changePlayer() methods of the game object to accept a drawing context object and clear the die. For the changePlayer() method, make this optional.