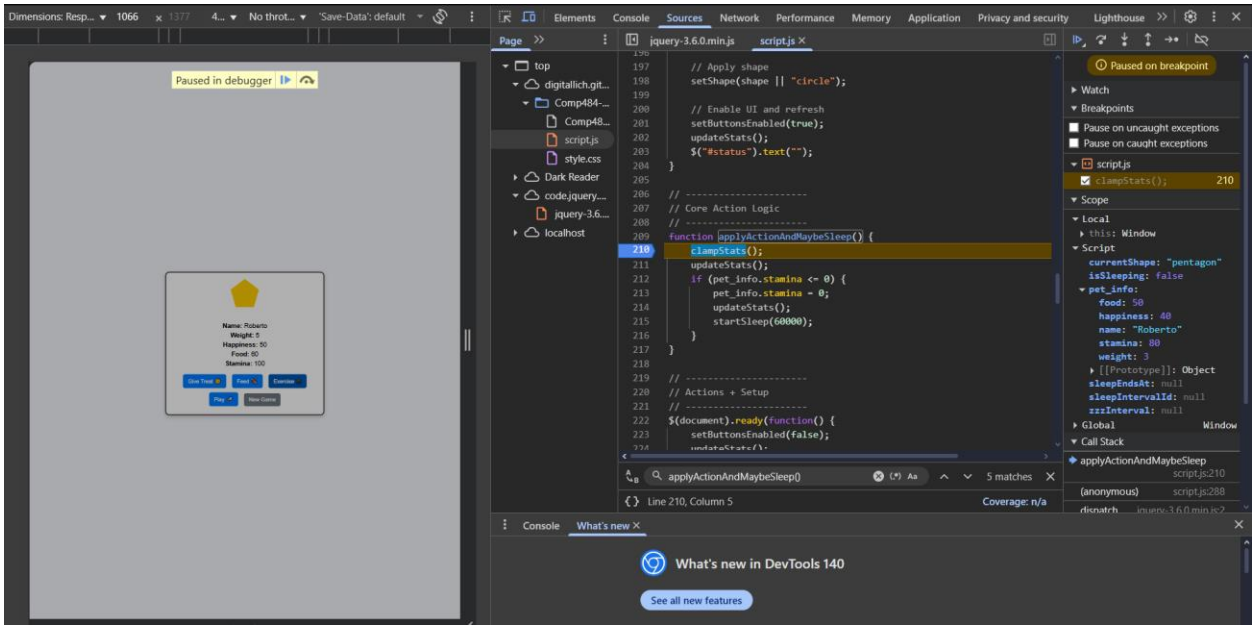
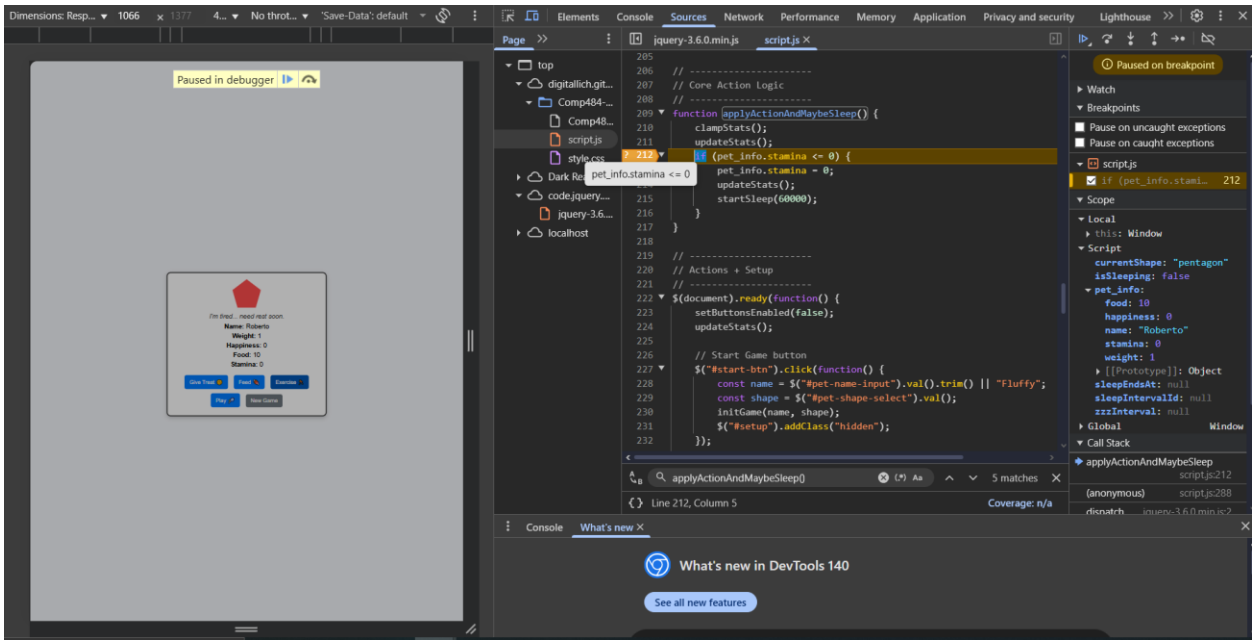
Homework 10

Devtools

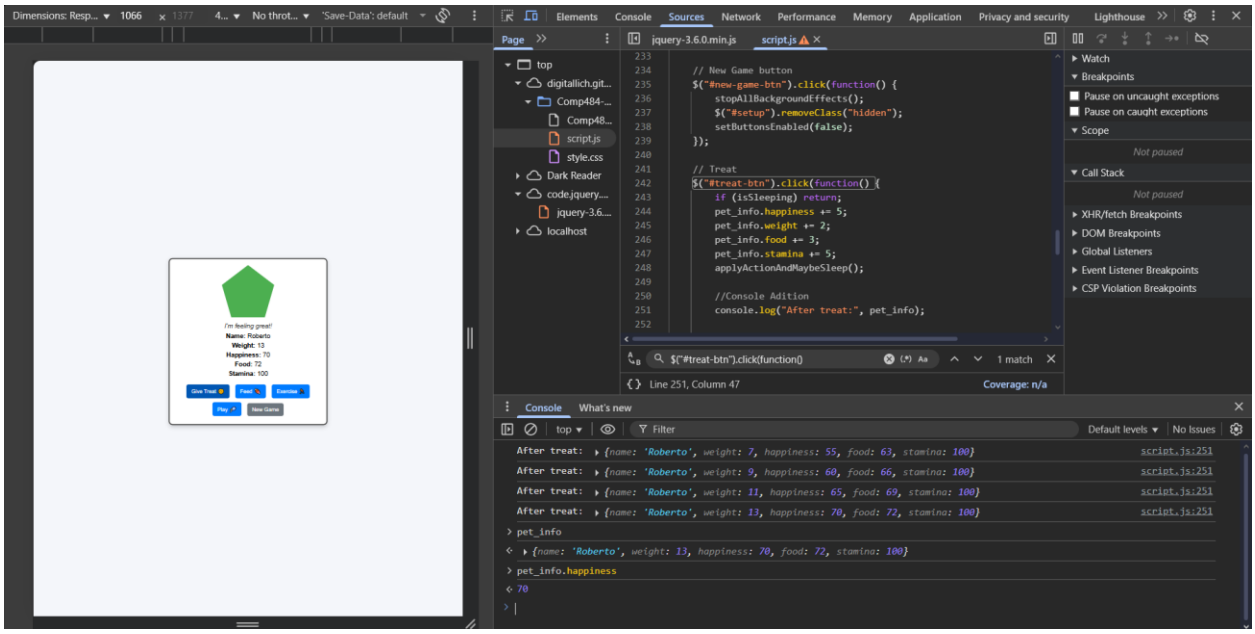
I set a breakpoint in the applyActionAndMaybeSleep() function in script.js. When I clicked the Exercise button, the execution paused before clamping the stats. In the Scope panel I inspected pet_info and saw stamina and happiness values change after each click. This helped me understand how the game decides when to put the pet to sleep.



I added a conditional breakpoint on the stamina check using pet_info.stamina <= 0. This made DevTools pause only when stamina dropped to zero. It let me debug the exact moment the pet falls asleep without stopping on every call, demonstrating how conditional breakpoints can focus debugging on specific states.

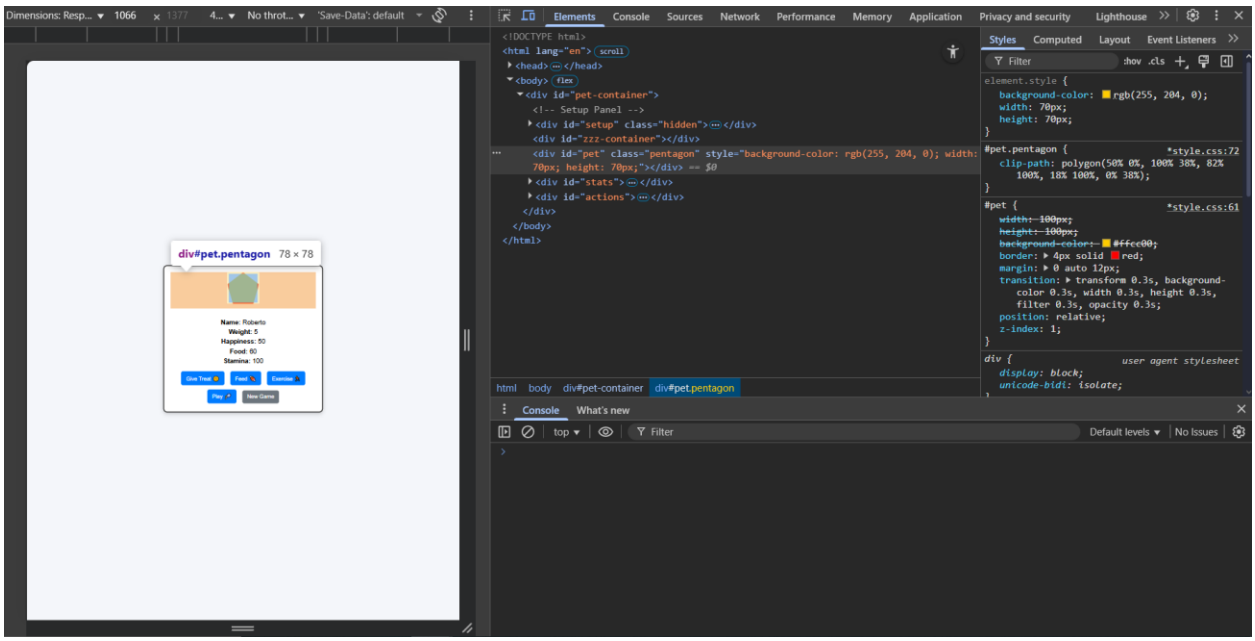


I added a console.log("After treat:", pet_info) to the Treat button handler. Every time I clicked the Treat button, the Console showed the updated pet_info. I also evaluated pet_info and pet_info.happiness directly in the Console to inspect the state of my game without adding more UI elements.

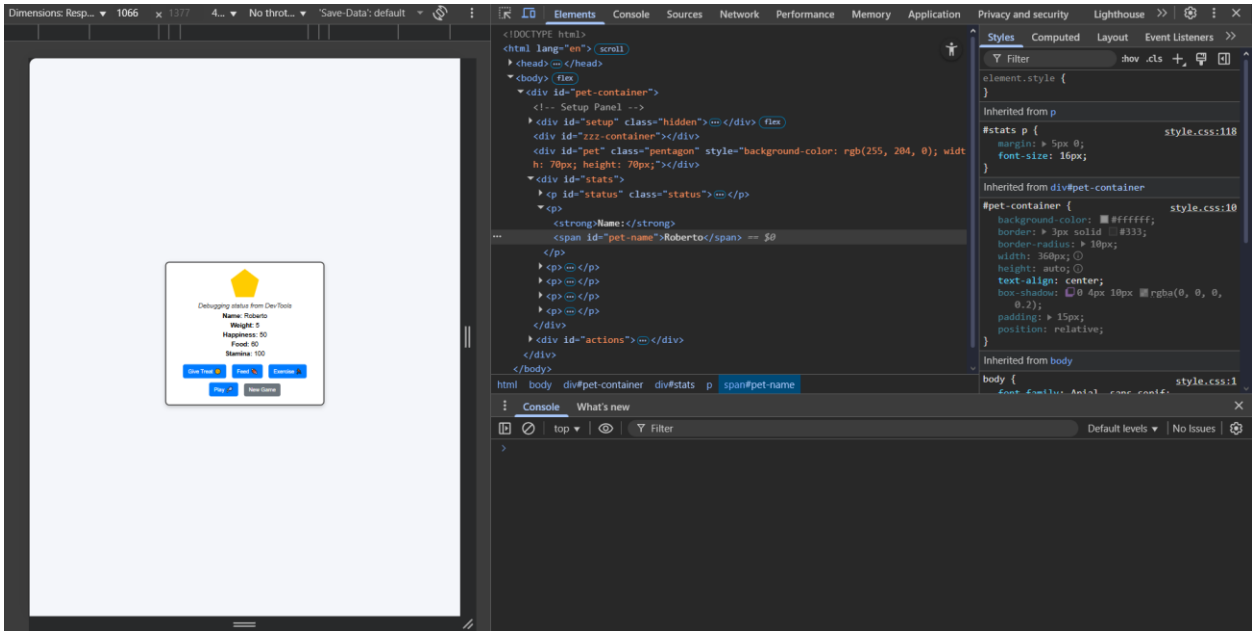


DOM

Using the Elements tab, I inspected the element and live-edited its styles. I changed the background color, size, and added a border. This showed how I can visually tweak and debug CSS in the browser before changing my actual stylesheet.



I directly edited the #status element using “Edit as HTML” in the Elements panel. The new text appeared immediately in the game UI, demonstrating how I can quickly test content and layout changes in the DOM.



I added a DOM breakpoint on the #pet element to break when its attributes change. When stamina reached zero, DevTools paused as the asleep class was added. This helped me connect the visual state (greyed-out pet) with the underlying JavaScript (startSleep and wakeUp functions).