

Case study Whack-a-mole



Overview

At our course here at SmartStudy we were to make a Whack-a-mole following the 30-day JavaScript30-challenge. We did not actually do 30 days of coding, but jumped straight to the last day, which was the Whack-a-mole game.

The setup is that we watch a video when a guy makes the game, then we code alongside the 14-minute-long video.

I made the game work and I'm happy with that.

Context and challenge

1. Project background and description

This was another game we were asked to make, on top of the others we made. It is a good way to learn JavaScript.

2. Problem/why

To better our developing skills.

3. Project-goal and objectives

Make a functional Whack-a-mole game.

Process and insight

This was a solo task. Everyone in class sat down and did their work at their own pace.

The Solution

I typed along the video, looked after to write it correctly to make it work.

The Result

The tutorial was very insightful and good explained. The game works perfectly. It is a bit hard, but I like the challenge.

```
<body>
  <h1>Whack-a-mole!<span class="score">0</span>
  <button class="button" onclick="startGame()">Start Game</button>
  <div class="game">
    <div class="hole hole1">
      <div class="mole"></div>
    </div>
    <div class="hole hole2">
      <div class="mole"></div>
    </div>
    <div class="hole hole3">
      <div class="mole"></div>
    </div>
    <div class="hole hole4">
      <div class="mole"></div>
    </div>
    <div class="hole hole5">
      <div class="mole"></div>
    </div>
    <div class="hole hole6">
      <div class="mole"></div>
    </div>
  </div>
  <script src="/javascripts/app.js"></script>
</body>
```

```
    setTimeout(() => (timeUp = true), 10000);
  }
  function bonk(e) {
    if (!e.isTrusted) return;
    score++;
    this.parentNode.classList.remove("up");
    scoreBoard.textContent = score;
  }
  moles.forEach(mole => mole.addEventListener("click", bonk));
```