views/js/main.js

```
/*
Welcome to the 60fps project! Your goal is to make Cam's
Pizzeria website run jank-free at 60 frames per second.

There are two major issues in this code that lead to
sub-60fps performance. Can you spot and fix both?


Built into the code, you'll find a few instances of the
User Timing API (window.performance), which will be
console.log()ing frame rate data into the browser console.
To learn more about User Timing API, check out:
http://www.html5rocks.com/en/tutorials/webperformance/usertiming/

Creator:
Cameron Pittman, Udacity Course Developer
cameron *at* udacity *dot* com
*/

// As you may have realized, this website randomly
// generates pizzas.
// Here are arrays of all possible pizza ingredients.
var pizzaIngredients = {};
pizzaIngredients.meats = [
  "Pepperoni",
  "Sausage",
  "Fennel Sausage",
  "Spicy Sausage",
  "Chicken",
  "BBQ Chicken",
  "Chorizo",
  "Chicken Andouille",
  "Salami",
  "Tofu",
  "Bacon",
  "Canadian Bacon",
  "Proscuitto",
  "Italian Sausage",
  "Ground Beef",
  "Anchovies",
  "Turkey",
  "Ham",
  "Venison",
```

```
  "Lamb",
  "Duck",
  "Soylent Green",
  "Carne Asada",
  "Soppressata Picante",
  "Coppa",
  "Pancetta",
  "Bresola",
  "Lox",
  "Guanciale",
  "Chili",
  "Beef Jerky",
  "Pastrami",
  "Kielbasa",
  "Scallops",
  "Filet Mignon"
];
pizzaIngredients.nonMeats = [
  "White Onions",
  "Red Onions",
  "Sauteed Onions",
  "Green Peppers",
  "Red Peppers",
  "Banana Peppers",
  "Ghost Peppers",
  "Habanero Peppers",
  "Jalapeno Peppers",
  "Stuffed Peppers",
  "Spinach",
  "Tomatoes",
  "Pineapple",
  "Pear Slices",
  "Apple Slices",
  "Mushrooms",
  "Arugula",
  "Basil",
  "Fennel",
  "Rosemary",
  "Cilantro",
  "Avocado",
  "Guacamole",
  "Salsa",
  "Swiss Chard",
  "Kale",
  "Sun Dried Tomatoes",
  "Walnuts",
```

```
  "Artichoke",
  "Asparagus",
  "Caramelized Onions",
  "Mango",
  "Garlic",
  "Olives",
  "Cauliflower",
  "Polenta",
  "Fried Egg",
  "Zucchini",
  "Hummus"
];
pizzaIngredients.cheeses = [
  "American Cheese",
  "Swiss Cheese",
  "Goat Cheese",
  "Mozzarella Cheese",
  "Parmesean Cheese",
  "Velveeta Cheese",
  "Gouda Cheese",
  "Muenster Cheese",
  "Applewood Cheese",
  "Asiago Cheese",
  "Bleu Cheese",
  "Boursin Cheese",
  "Brie Cheese",
  "Cheddar Cheese",
  "Chevre Cheese",
  "Havarti Cheese",
  "Jack Cheese",
  "Pepper Jack Cheese",
  "Gruyere Cheese",
  "Limberger Cheese",
  "Manchego Cheese",
  "Marscapone Cheese",
  "Pecorino Cheese",
  "Provolone Cheese",
  "Queso Cheese",
  "Roquefort Cheese",
  "Romano Cheese",
  "Ricotta Cheese",
  "Smoked Gouda"
];
pizzaIngredients.sauces = [
  "Red Sauce",
  "Marinara",
```

```
  "BBQ Sauce",
  "No Sauce",
  "Hot Sauce"
];
pizzaIngredients.crusts = [
  "White Crust",
  "Whole Wheat Crust",
  "Flatbread Crust",
  "Stuffed Crust"
];

// Name generator pulled from
// http://saturdaykid.com/usernames/generator.html
// Capitalizes first letter of each word
String.prototype.capitalize = function() {
  return this.charAt(0).toUpperCase() + this.slice(1);
};

// Pulls adjective out of array using random number
// sent from generator
function getAdj(x){
  switch(x) {
    case "dark":
      var dark = ["dark","morbid", "scary", "spooky",
                  "gothic", "deviant", "creepy", "sadistic",
                  "black", "dangerous", "dejected",
                  "haunted", "morose", "tragic",
                  "shattered", "broken", "sad",
                  "melancholy", "somber", "dark", "gloomy",
                  "homicidal", "murderous", "shady",
                  "misty", "dusky", "ghostly", "shadowy",
                  "demented", "cursed", "insane",
                  "possessed", "grotesque", "obsessed"];
      return dark;
    case "color":
      var colors = ["blue", "green", "purple", "grey",
                    "scarlet", "NeonGreen", "NeonBlue",
                    "NeonPink", "HotPink", "pink", "black",
                    "red", "maroon", "silver", "golden",
                    "yellow", "orange", "mustard", "plum",
                    "violet", "cerulean", "brown",
                    "lavender", "violet", "magenta",
                    "chestnut", "rosy", "copper", "crimson",
                    "teal", "indigo", "navy", "azure",
                    "periwinkle", "brassy", "verdigris",
                    "veridian", "tan", "raspberry", "beige",
```

```
                "sandy", "ElectricBlue", "white",
                "champagne", "coral", "cyan"];
  return colors;
case "whimsical":
  var whimsy = ["whimsical", "silly", "drunken",
                "goofy", "funny", "weird", "strange",
                "odd", "playful", "clever", "boastful",
                "breakdancing", "hilarious",
                "conceited", "happy", "comical",
                "curious", "peculiar", "quaint",
                "quirky", "fancy", "wayward", "fickle",
                "yawning", "sleepy", "cockeyed",
                "dizzy", "dancing", "absurd",
                "laughing", "hairy", "smiling",
                "perplexed", "baffled", "cockamamie",
                "vulgar", "hoodwinked", "brainwashed"];
  return whimsy;
case "shiny":
  var shiny = ["sapphire", "opal", "silver", "gold",
               "platinum", "ruby", "emerald", "topaz",
               "diamond", "amethyst", "turquoise",
               "starlit", "moonlit", "bronze", "metal",
               "jade", "amber", "garnet", "obsidian",
               "onyx", "pearl", "copper", "sunlit",
               "brass", "brassy", "metallic"];
  return shiny;
case "noisy":
  var noisy = ["untuned", "loud", "soft", "shrieking",
               "melodious", "musical", "operatic",
               "symphonic", "dancing", "lyrical",
               "harmonic", "orchestral", "noisy",
               "dissonant", "rhythmic", "hissing",
               "singing", "crooning", "shouting",
               "screaming", "wailing", "crying",
               "howling", "yelling", "hollering",
               "caterwauling", "bawling", "bellowing",
               "roaring", "squealing", "beeping",
               "knocking", "tapping", "rapping",
               "humming", "scatting", "whispered",
               "whispering", "rasping", "buzzing",
               "whirring", "whistling", "whistled"];
  return noisy;
case "apocalyptic":
  var apocalyptic = ["nuclear", "apocalyptic",
                     "desolate", "atomic", "zombie",
                     "collapsed", "grim", "fallen",
```

```
                        "collapsed", "cannibalistic",
                        "radioactive", "toxic",
                        "poisonous", "venomous",
                        "disastrous", "grimy", "dirty",
                        "undead", "bloodshot", "rusty",
                        "glowing", "decaying", "rotten",
                        "deadly", "plagued", "decimated",
                        "rotting", "putrid", "decayed",
                        "deserted", "acidic"];
    return apocalyptic;
  case "insulting":
    var insulting = ["stupid", "idiotic", "fat", "ugly",
                     "hideous", "grotesque", "dull",
                     "dumb", "lazy", "sluggish",
                     "brainless", "slow", "gullible",
                     "obtuse", "dense", "dim", "dazed",
                     "ridiculous", "witless", "daft",
                     "crazy", "vapid", "inane", "mundane",
                     "hollow", "vacuous", "boring",
                     "insipid", "tedious", "monotonous",
                     "weird", "bizarre", "backward",
                     "moronic", "ignorant",
                     "scatterbrained", "forgetful",
                     "careless", "lethargic", "insolent",
                     "indolent", "loitering", "gross",
                     "disgusting", "bland", "horrid",
                     "unseemly", "revolting", "homely",
                     "deformed", "disfigured",
                     "offensive", "cowardly", "weak",
                     "villainous", "fearful", "monstrous",
                     "unattractive", "unpleasant",
                     "nasty", "beastly", "snide",
                     "horrible", "syncophantic",
                     "unhelpful", "bootlicking"];
    return insulting;
  case "praise":
    var praise = ["beautiful", "intelligent", "smart",
                  "genius", "ingenious", "gorgeous",
                  "pretty", "witty", "angelic",
                  "handsome", "graceful", "talented",
                  "exquisite", "enchanting",
                  "fascinating", "interesting",
                  "divine", "alluring", "ravishing",
                  "wonderful", "magnificient", "marvelous",
                  "dazzling", "cute", "charming",
                  "attractive", "nifty", "delightful",
```

```
                    "superior", "amiable", "gentle",
                    "heroic", "courageous", "valiant",
                    "brave", "noble", "daring", "fearless",
                    "gallant", "adventurous", "cool",
                    "enthusiastic", "fierce", "awesome",
                    "radical", "tubular", "fearsome",
                    "majestic", "grand", "stunning"];
      return praise;
    case "scientific":
      var scientific = ["scientific", "technical",
                        "digital", "programming",
                        "calculating", "formulating",
                        "cyberpunk", "mechanical",
                        "technological", "innovative",
                        "brainy", "chemical", "quantum",
                        "astro", "space", "theoretical",
                        "atomic", "electronic", "gaseous",
                        "investigative", "solar",
                        "extinct", "galactic"];
      return scientific;
    default:
      var scientific_default = ["scientific", "technical",
                                "digital", "programming",
                                "calculating",
                                "formulating", "cyberpunk",
                                "mechanical",
                                "technological",
                                "innovative", "brainy",
                                "chemical", "quantum",
                                "astro", "space",
                                "theoretical", "atomic",
                                "electronic", "gaseous",
                                "investigative", "solar",
                                "extinct", "galactic"];
      return scientific_default;
  }
}

// Pulls noun out of array using random number sent
// from generator
function getNoun(y) {
  switch(y) {
    case "animals":
      var animals = ["flamingo", "hedgehog", "owl",
                     "elephant", "pussycat", "alligator",
                     "dachsund", "poodle", "beagle",
```

```
                    "crocodile", "kangaroo", "wallaby",
                    "woodpecker", "eagle", "falcon",
                    "canary", "parrot", "parakeet",
                    "hamster", "gerbil", "squirrel", "rat",
                    "dove", "toucan", "raccoon", "vulture",
                    "peacock", "goldfish", "rook", "koala",
                    "skunk", "goat", "rooster", "fox",
                    "porcupine", "llama", "grasshopper",
                    "gorilla", "monkey", "seahorse",
                    "wombat", "wolf", "giraffe", "badger",
                    "lion", "mouse", "beetle", "cricket",
                    "nightingale", "hawk", "trout",
                    "squid", "octopus", "sloth", "snail",
                    "locust", "baboon", "lemur", "meerkat",
                    "oyster", "frog", "toad", "jellyfish",
                    "butterfly", "caterpillar", "tiger",
                    "hyena", "zebra", "snail", "pig",
                    "weasel", "donkey", "penguin", "crane",
                    "buzzard", "vulture", "rhino",
                    "hippopotamus", "dolphin", "sparrow",
                    "beaver", "moose", "minnow", "otter",
                    "bat", "mongoose", "swan", "firefly",
                    "platypus"];
  return animals;
case "profession":
  var professions = ["doctor", "lawyer", "ninja",
                     "writer", "samurai", "surgeon",
                     "clerk", "artist", "actor",
                     "engineer", "mechanic", "comedian",
                     "fireman", "nurse", "RockStar",
                     "musician", "carpenter", "plumber",
                     "cashier", "electrician", "waiter",
                     "president", "governor", "senator",
                     "scientist", "programmer",
                     "singer", "dancer", "director",
                     "mayor", "merchant", "detective",
                     "investigator", "navigator",
                     "pilot", "priest", "cowboy",
                     "stagehand", "soldier",
                     "ambassador", "pirate", "miner",
                     "police"];
  return professions;
case "fantasy":
  var fantasy = ["centaur", "wizard", "gnome", "orc",
                 "troll", "sword", "fairy", "pegasus",
                 "halfling", "elf", "changeling",
```

```
                    "ghost", "knight", "squire",
                    "magician", "witch", "warlock",
                    "unicorn", "dragon", "wyvern",
                    "princess", "prince", "king", "queen",
                    "jester", "tower", "castle", "kraken",
                    "seamonster", "mermaid", "psychic",
                    "seer", "oracle"];
  return fantasy;
case "music":
  var music = ["violin", "flute", "bagpipe", "guitar",
               "symphony", "orchestra", "piano",
               "trombone", "tuba", "opera", "drums",
               "harpsichord", "harp", "harmonica",
               "accordion", "tenor", "soprano",
               "baritone", "cello", "viola", "piccolo",
               "ukelele", "woodwind", "saxophone",
               "bugle", "trumpet", "sousaphone",
               "cornet", "stradivarius", "marimbas",
               "bells", "timpani", "bongos", "clarinet",
               "recorder", "oboe", "conductor",
  "singer"];
  return music;
case "horror":
  var horror = ["murderer", "chainsaw", "knife",
                "sword", "murder", "devil", "killer",
                "psycho", "ghost", "monster",
                "godzilla", "werewolf", "vampire",
                "demon", "graveyard", "zombie",
                "mummy", "curse", "death", "grave",
                "tomb", "beast", "nightmare",
                "frankenstein", "specter",
                "poltergeist", "wraith", "corpse",
                "scream", "massacre", "cannibal",
                "skull", "bones", "undertaker",
                "zombie", "creature", "mask",
                "psychopath", "fiend", "satanist",
                "moon", "fullMoon"];
  return horror;
case "gross":
  var gross = ["slime", "bug", "roach", "fluid", "pus",
               "booger", "spit", "boil", "blister",
               "orifice", "secretion", "mucus",
               "phlegm", "centipede", "beetle", "fart",
               "snot", "crevice", "flatulence", "juice",
               "mold", "mildew", "germs", "discharge",
               "toilet", "udder", "odor", "substance",
```

```
                "fluid", "moisture", "garbage", "trash",
                "bug"];
  return gross;
case "everyday":
  var everyday = ["mirror", "knife", "fork", "spork",
                  "spoon", "tupperware", "minivan",
                  "suburb", "lamp", "desk", "stereo",
                  "television", "TV", "book", "car",
                  "truck", "soda", "door", "video",
                  "game", "computer", "calender",
                  "tree", "plant", "flower", "chimney",
                  "attic", "kitchen", "garden",
                  "school", "wallet", "bottle"];
  return everyday;
case "jewelry":
  var jewelry = ["earrings", "ring", "necklace",
                 "pendant", "choker", "brooch",
                 "bracelet", "cameo", "charm", "bauble",
                 "trinket", "jewelry", "anklet",
                 "bangle", "locket", "finery",
                 "crown", "tiara", "blingBling",
                 "chain", "rosary", "jewel", "gemstone",
                 "beads", "armband", "pin", "costume",
                 "ornament", "treasure"];
  return jewelry;
case "places":
  var places = ["swamp", "graveyard", "cemetery",
                "park", "building", "house", "river",
                "ocean", "sea", "field", "forest",
                "woods", "neighborhood", "city", "town",
                "suburb", "country", "meadow", "cliffs",
                "lake", "stream", "creek", "school",
                "college", "university", "library",
                "bakery", "shop", "store", "theater",
                "garden", "canyon", "highway",
                "restaurant", "cafe", "diner", "street",
                "road", "freeway", "alley"];
  return places;
case "scifi":
  var scifi = ["robot", "alien", "raygun", "spaceship",
               "UFO", "rocket", "phaser", "astronaut",
               "spaceman", "planet", "star", "galaxy",
               "computer", "future", "timeMachine",
               "wormHole", "timeTraveler", "scientist",
               "invention", "martian", "pluto",
               "jupiter", "saturn", "mars", "quasar",
```

```
                    "blackHole", "warpDrive", "laser",
                    "orbit", "gears", "molecule", "electron",
                    "neutrino", "proton", "experiment",
                    "photon", "apparatus", "universe",
                    "gravity", "darkMatter", "constellation",
                    "circuit", "asteroid"];
      return scifi;
    default:
      var scifi_default = ["robot", "alien", "raygun",
                           "spaceship", "UFO", "rocket",
                           "phaser", "astronaut",
                           "spaceman", "planet", "star",
                           "galaxy", "computer", "future",
                           "timeMachine", "wormHole",
                           "timeTraveler", "scientist",
                           "invention", "martian", "pluto",
                           "jupiter", "saturn", "mars",
                           "quasar", "blackHole",
                           "warpDrive", "laser", "orbit",
                           "gears", "molecule", "electron",
                           "neutrino", "proton",
                           "experiment", "photon",
                           "apparatus", "universe",
                           "gravity", "darkMatter",
                           "constellation", "circuit",
                           "asteroid"];
      return scifi_default;
  }
}

// types of adjectives for pizza titles
var adjectives = ["dark", "color", "whimsical", "shiny",
                  "noise", "apocalyptic", "insulting",
                  "praise", "scientific"];

// types of nouns for pizza titles
var nouns = ["animals", "everyday", "fantasy", "gross",
             "horror", "jewelry", "places", "scifi"];

// Generates random numbers for getAdj and getNoun
// functions and returns a new pizza name
function generator(adj, noun) {
  var adjectives = getAdj(adj);
  var nouns = getNoun(noun);
  var randomAdjective =
      parseInt(Math.random() * adjectives.length);
```

```
  var randomNoun = parseInt(Math.random() * nouns.length);
  var name = "The " +
      adjectives[randomAdjective].capitalize() + " " +
      nouns[randomNoun].capitalize();
  return name;
}

// Chooses random adjective and random noun
function randomName() {
  var randomNumberAdj = parseInt(Math.random() *
      adjectives.length);
  var randomNumberNoun = parseInt(Math.random() *
      nouns.length);
  return generator(adjectives[randomNumberAdj],
      nouns[randomNumberNoun]);
}

// These functions return a string of a random ingredient
// from each respective category of ingredients.
var selectRandomMeat = function() {
  var randomMeat = pizzaIngredients.meats[
      Math.floor((Math.random() *
      pizzaIngredients.meats.length))];
  return randomMeat;
};

var selectRandomNonMeat = function() {
  var randomNonMeat = pizzaIngredients.nonMeats[
      Math.floor((Math.random() *
      pizzaIngredients.nonMeats.length))];
  return randomNonMeat;
};

var selectRandomCheese = function() {
  var randomCheese = pizzaIngredients.cheeses[
      Math.floor((Math.random() *
      pizzaIngredients.cheeses.length))];
  return randomCheese;
};

var selectRandomSauce = function() {
  var randomSauce = pizzaIngredients.sauces[
      Math.floor((Math.random() *
      pizzaIngredients.sauces.length))];
  return randomSauce;
};
```

```
var selectRandomCrust = function() {
  var randomCrust = pizzaIngredients.crusts[
      Math.floor((Math.random() *
      pizzaIngredients.crusts.length))];
  return randomCrust;
};

var ingredientItemizer = function(string) {
  return "<li>" + string + "</li>";
};

// Returns a string with random pizza ingredients nested
// inside <li> tags
var makeRandomPizza = function() {
  var pizza = "";

  var numberOfMeats = Math.floor((Math.random() * 4));
  var numberOfNonMeats = Math.floor((Math.random() * 3));
  var numberOfCheeses = Math.floor((Math.random() * 2));

  for (var i = 0; i < numberOfMeats; i++) {
    pizza = pizza + ingredientItemizer(selectRandomMeat());
  }

  for (var j = 0; j < numberOfNonMeats; j++) {
    pizza = pizza +
        ingredientItemizer(selectRandomNonMeat());
  }

  for (var k = 0; k < numberOfCheeses; k++) {
    pizza = pizza + ingredientItemizer(selectRandomCheese());
  }

  pizza = pizza + ingredientItemizer(selectRandomSauce());
  pizza = pizza + ingredientItemizer(selectRandomCrust());

  return pizza;
};

// returns a DOM element for each pizza
var pizzaElementGenerator = function(i) {
  var
      // contains pizza title, image and list of ingredients
      pizzaContainer,
      // contains the pizza image
```

```
        pizzaImageContainer,
        // the pizza image itself
        pizzaImage,
        // contains the pizza title and list of ingredients
        pizzaDescriptionContainer,
        // the pizza name itself
        pizzaName,
        // the list of ingredients
        ul;

    pizzaContainer  = document.createElement("div");
    pizzaImageContainer = document.createElement("div");
    pizzaImage = document.createElement("img");
    pizzaDescriptionContainer = document.createElement("div");

    pizzaContainer.classList.add("randomPizzaContainer");
    pizzaContainer.style.width = "33.33%";
    pizzaContainer.style.height = "325px";

    // gives each pizza element a unique id
    pizzaContainer.id = "pizza" + i;
    pizzaImageContainer.classList.add("col-md-6");

    pizzaImage.src = "images/pizza.png";
    pizzaImage.classList.add("img-responsive");
    pizzaImageContainer.appendChild(pizzaImage);
    pizzaContainer.appendChild(pizzaImageContainer);


    pizzaDescriptionContainer.classList.add("col-md-6");

    pizzaName = document.createElement("h4");
    pizzaName.innerHTML = randomName();
    pizzaDescriptionContainer.appendChild(pizzaName);

    ul = document.createElement("ul");
    ul.innerHTML = makeRandomPizza();
    pizzaDescriptionContainer.appendChild(ul);
    pizzaContainer.appendChild(pizzaDescriptionContainer);

    return pizzaContainer;
};

// resizePizzas(size) is called when the slider in
// the "Our Pizzas" section of the website moves.
var resizePizzas = function(size) {
```

```
 // User Timing API function
window.performance.mark("mark_start_resize");

// Changes the value for the size of the pizza above
// the slider
function changeSliderLabel(size) {
  switch(size) {
    case "1":
      document.querySelector("#pizzaSize").innerHTML =
          "Small";
      return;
    case "2":
      document.querySelector("#pizzaSize").innerHTML =
          "Medium";
      return;
    case "3":
      document.querySelector("#pizzaSize").innerHTML =
          "Large";
      return;
    default:
      console.log("bug_in_changeSliderLabel");
  }
}

changeSliderLabel(size);

// Returns the size difference to change a pizza element
// from one size to another.
// Called by changePizzaSlices(size).
function determineDx (elem, size) {
  var oldwidth = elem.offsetWidth;
  var windowwidth =
      document.querySelector("#randomPizzas").offsetWidth;
  var oldsize = oldwidth / windowwidth;

  // TODO: change to 3 sizes? no more xl?
  // Changes the slider value to a percent width
  function sizeSwitcher (size) {
    switch(size) {
      case "1":
        return 0.25;
      case "2":
        return 0.3333;
      case "3":
        return 0.5;
      default:
```

```
            console.log("bug in sizeSwitcher");
        }
      }

      var newsize = sizeSwitcher(size);
      var dx = (newsize - oldsize) * windowwidth;

      return dx;
    }

    // Iterates through pizza elements on the page and
    // changes their widths
    function changePizzaSizes(size) {
      for (var i = 0;
          i < document.querySelectorAll(
              ".randomPizzaContainer").length; i++) {
          var dx =
              determineDx(document.querySelectorAll(
                  ".randomPizzaContainer")[i], size);
          var newwidth = (document.querySelectorAll(
                  ".randomPizzaContainer")[i].offsetWidth +
                  dx) + 'px';
          document.querySelectorAll(
              ".randomPizzaContainer")[i].style.width =
              newwidth;
      }
    }

    changePizzaSizes(size);

    // User Timing API is awesome
    window.performance.mark("mark_end_resize");
    window.performance.measure("measure_pizza_resize",
        "mark_start_resize", "mark_end_resize");
    var timeToResize = window.performance.getEntriesByName(
        "measure_pizza_resize");
    console.log("Time to resize pizzas: " +
        timeToResize[0].duration + "ms");
};

// collect timing data
window.performance.mark("mark_start_generating");

// This for-loop actually creates and appends all of the
// pizzas when the page loads
for (var i = 2; i < 100; i++) {
```

```
  var pizzasDiv = document.getElementById("randomPizzas");
  pizzasDiv.appendChild(pizzaElementGenerator(i));
}

// User Timing API again. These measurements tell you how
// long it took to generate the initial pizzas
window.performance.mark("mark_end_generating");
window.performance.measure("measure_pizza_generation",
    "mark_start_generating", "mark_end_generating");
var timeToGenerate =
    window.performance.getEntriesByName(
        "measure_pizza_generation");
console.log("Time to generate pizzas on load: " +
    timeToGenerate[0].duration + "ms");

// Iterator for number of times the pizzas in the
// background have scrolled.
// Used by updatePositions() to decide when to log the
// average time per frame
var frame = 0;

// Logs the average amount of time per 10 frames needed to
//  move the sliding background pizzas on scroll.
function logAverageFrame(times) {
   // times is the array of User Timing measurements from
   // updatePositions()
  var numberOfEntries = times.length;
  var sum = 0;
  for (var i = numberOfEntries - 1;
       i > numberOfEntries - 11; i--) {
    sum = sum + times[i].duration;
  }
  console.log("Average time to generate last 10 frames: " +
      sum / 10 + "ms");
}

// The following code for sliding background pizzas was
// pulled from Ilya's demo found at:
// https://www.igvita.com/slides/2012/
//    devtools-tips-and-tricks/jank-demo.html

// Moves the sliding background pizzas based on scroll
// position
function updatePositions() {
  frame++;
  window.performance.mark("mark_start_frame");
```

```
  var items = document.querySelectorAll('.mover');
  for (var i = 0; i < items.length; i++) {
    var phase = Math.sin((document.body.scrollTop / 1250) +
        (i % 5));
    items[i].style.left = items[i].basicLeft + 100 * phase +
        'px';
  }

  // User Timing API to the rescue again.
  // Seriously, it's worth learning.
  // Super easy to create custom metrics.
  window.performance.mark("mark_end_frame");
  window.performance.measure("measure_frame_duration",
      "mark_start_frame", "mark_end_frame");
  if (frame % 10 === 0) {
    var timesToUpdatePosition =
      window.performance.getEntriesByName(
          "measure_frame_duration");
    logAverageFrame(timesToUpdatePosition);
  }
}

// runs updatePositions on scroll
window.addEventListener('scroll', updatePositions);

// Generates the sliding pizzas when the page loads.
document.addEventListener('DOMContentLoaded', function() {
  var cols = 8;
  var s = 256;
  for (var i = 0; i < 200; i++) {
    var elem = document.createElement('img');
    elem.className = 'mover';
    elem.src = "images/pizza.png";
    elem.style.height = "100px";
    elem.style.width = "73.333px";
    elem.basicLeft = (i % cols) * s;
    elem.style.top = (Math.floor(i / cols) * s) + 'px';
    document.querySelector(
        "#movingPizzas1").appendChild(elem);
  }
  updatePositions();
});
```