```
'use strict';
/////////////// Lecture 82: Project #3: PIG GAME
  Introduction
/*

// Drawing Diagrams
If you would like to draw flowcharts like the one in
  the image go to
diagrams.net

*/
// Building Application - Step 1: Set scores to 0,
  make dice disappear.
// In this case, we will have to select elements
  using their unique ID,
// instead of their class name.
// Use the # for IDs when using querySelector. #str
  and .str are selectors.
// So when we use querySelector()  we have to use the
  #
const score0El = document.querySelector('#score--0');
// If you don't like the hash, you can do it this way
  instead:
const score1El = document.getElementById('score--1');
// select the dice element
const diceEl = document.querySelector('.dice');
// NOTE: This .getElementById() method is supposed to
  be a little bit faster
// than querySelector asymptotically. A difference is
  only noticeable if you
// select a thousand elements or something.
const current0El = document.getElementById('current--
  0');
const current1El = document.getElementById('current--
  1');

// Select the active class between the players:
const player0El = document.querySelector('.player--
  0');
const player1El = document.querySelector('.player--
  1');
//Set these scores to 0
```

```
30    score0El.textContent = score1El.textContent = 0; //
        note that we write here a number, but JS will
        implicitly convert them to strings.
31
32    // Create a hidden class for dice, and add it at the
        beginning.
33    diceEl.classList.add(`hidden`);
34
35    ///////////////// Lecture 83: Rolling the Dice
36    // Building Application - Step 2: Rolling the Dice:
        When user rolls a dice, we want to first generate a
        random dice roll, then display it, and then check
        whether it is a one or not. If it's not, we add
        that dice roll to the current score. If it is a 1,
        we go to the next player.
37    // Notice the flowchart is sort of breaking down a
        large problem into a subproblem, which may be
        really useful.
38
39    // We first want to react to clicking the roll dice
        button. Add an event listener to it:
40    const btnNewEl = document.querySelector('.btn--new');
41    const btnRollEl = document.querySelector('.btn--
        roll');
42    const btnHoldEl = document.querySelector('.btn--
        hold');
43
44    // Global variable to hold currentScore
45    const scores = new Array(0, 0);
46    let currentScore = 0; // See line 54
47    let activePlayer = 0; // See Lecture 83
48    let playing = true;
49    // Lect
50    //Rolling the dice functionality
51    btnRollEl.addEventListener(`click`, function () {
52      if (playing) {
53        // Generate random dice rolls
54        const dice = Math.trunc(Math.random() * 6) + 1;
55        // Display the dice
56        diceEl.classList.remove(`hidden`);
57        // Manipulate the source attribute of the <a> tag
            from our JS.
```

```
58        diceEl.src = `dice-${dice}.png`; // Remember your
            string literals!!
59        // For debugging:
60        console.log(dice);
61        // If its a 1, switch player, if not add to the
            current score.
62        if (dice !== 1) {
63          // Add to the current score
64          // Note that you SHOULD NOT only store the
              currentScore in the DOM.
65          // instead, we would also want the currentScore
              as a variable in our JS Code, which always
              holds the currentscore of this current round.
              So we need to declare a variable outside the
              scope of this function.
66          currentScore += dice;
67          // current0El.textContent = currentScore;
68          document.getElementById(
69            `current--${activePlayer}`
70          ).textContent = currentScore;
71        }
72        /////////// Lecture 83: Switching the Active
            player
73        // Switching from one active player to another
74        // We need to keep track which player is the
            active player the moment the dice was rolled.
            So we will create another variable that holds
            exactly this.
75        // We'll create a variable that toggles between 0
            and 1, and it will be 0 when the active player
            is 0, and 1 when the active player is 1.
76        // Since we start with the first player, we set
            it to 0.
77        // We'll also be storing the scores of the
            players in an array, remember arrays are zero-
            based, and the score of player number 1 will be
            here at index 0, and the score of player 2 will
            have the index 1.
78        // for now, we'll have to adjust our code in the
            last lecture to select the currentScore for the
            player that is active upon the dice roll.
79        else {
```

```
        changePlayer();
      }
    }
    // When the active player changes, we need to change the interface slightly from the inactive to the active class.}
  }); // We won't be reusing this function, so we can declare it anonymously.

  /////////////// Lecture 84: Holding Current score
  // Building Application Part 4: upon the click of the button, we want to add the current score to the total score. So let's see that in the demo version here: As we roll the dice, the currentscore will be transferred to the global score.
  // If the score is at least 50, then the current Player wins.
  btnHoldEl.addEventListener('click', function () {
    if (playing) {
      // Add current score to score of active player:
      scores[activePlayer] += currentScore;
      document.getElementById(`score--${activePlayer}`).textContent = scores[activePlayer];
      // Check if current score is at least 100,
      if (scores[activePlayer] >= 20) {
        playing = false;
        // Finish the Game
        document
          .querySelector(`.player--${activePlayer}`)
          .classList.add('player--winner');
        document
          .querySelector(`.player--${activePlayer}`)
          .classList.remove(`player--active`);
        diceEl.classList.add(`hidden`); // makes the dice disappear upon winning the game.
      } else {
        // If not , switch to the next player.
        changePlayer();
      }
    }
  });
```

```
112
113   // Utility function - Lecture 84
114   function changePlayer() {
115     document.getElementById(`current--
        ${activePlayer}`).textContent = 0;
116     currentScore = 0;
117     activePlayer = 1 - activePlayer; // Reassign the
        activePlayer
118     // the toggle() method in classList, if the class
        isn't there it will add it.
119     // if the class is there, it will remove it.
120     player0El.classList.toggle(`player--active`);
121     player1El.classList.toggle(`player--active`);
122   }
123   /////////////// CODING CHALLENGE #2: Lecture 85:
        Resetting the Game
124   btnNew.addEventListener();
125
```